GENERATIVE ADVERSARIAL NETWORKS TO IMPROVE, TO EXPLAIN, AND TO INTERACT WITH ARTIFICIAL INTELLIGENCE

SILVAN MERTES

Universität
Augsburg
University

Dissertation zur Erreichung des akademischen Grades Dr. rer. nat.

University of Augsburg
Faculty of Applied Computer Science
Chair for Human-Centered Artificial Intelligence
April 2025

Erstgutachterin: Prof. Dr. Elisabeth André
Zweitgutachterin: Prof. Dr. Cristina Conati
Drittgutachter: Prof. Dr. Bernhard Bauer

Tag der mündlichen Prüfung: 19. Mai 2025

Silvan Mertes: *Generative Adversarial Networks to Improve, to Explain, and to Interact with Artificial Intelligence,* Dissertation zur Erreichung des akademischen Grades Dr. rer. nat., © April 2025

ABSTRACT

Generative Artificial Intelligence has become part of our everyday lives. Nowadays, it is difficult to *not* get in contact with generative technologies - be it in the form of writing assistants, image generation tools, or even personal assistants. However, Generative AI is not yet leveraged to its full potential. There are lots of directions that can benefit from such systems, but many of them are still under-researched. Generative Adversarial Networks (GANs) in particular have technically matured over the last years, while systems and approaches that use those models are still in their infancy. In this thesis, five topic areas are identified that can be substantially enhanced by GANs. Those areas are *Robustness of AI Systems*, *Explainability of AI Systems*, *Expressiveness of AI Systems*, *Feedback Synthesis* and *Interaction with AI*.

The thesis then introduces concepts, technical approaches and user studies that investigate how GANs can be used in those areas. Approaches for using GANs to augment datasets are introduced to improve the robustness of AI systems. It is proposed how GANs can be used to synthesize realistic visual explanations that make use of the principles of counterfactual reasoning in order to foster explainability of AI systems. It is presented how GANs can be used to make AI more expressive by synthesizing continuously conditioned images without the need for continuously labeled training data. Approaches are introduced that provide personalized visual and textual feedback for a job interview training system. Finally, it is shown how GANs can be used build interactive systems that can counteract stress. As such, the contributions of this thesis aim to showcase the potential of GANs in a wide range of research fields.

ZUSAMMENFASSUNG

Generative Künstliche Intelligenz ist mittlerweile ein fester Bestandteil unseres Alltags. Heutzutage ist es schwierig, *nicht* mit generativen Technologien in Kontakt zu kommen – sei es in Form von Schreibassistenten, Werkzeugen zur Bildgenerierung oder sogar persönlichen Assistenten. Dennoch wird das Potenzial solcher Technologien noch längst nicht vollständig ausgeschöpft. Es gibt viele Möglichkeiten, wie diverse Forschungsfelder von solchen Systemen profitieren könnten, aber vieles davon ist noch kaum erforscht.

Insbesondere *Generative Adversarial Networks* (GANs) haben sich in den letzten Jahren technisch stark weiterentwickelt, während Systeme und Ansätze, die diese Modelle nutzen, noch in den Kinderschuhen stecken. In dieser Arbeit werden fünf Themenbereiche identifiziert, die durch den Einsatz von GANs erheblich vorangebracht werden können. Diese Bereiche sind: Robustheit von KI-Systemen, Erklärbarkeit von KI-Systemen, Expressivität von KI-Systemen, Feedback-Generierung und Interaktivität in KI-Systemen.

Die Arbeit stellt Konzepte, technische Ansätze und Nutzerstudien vor, die untersuchen, wie GANs in diesen Bereichen eingesetzt werden können. Ansätze zur Nutzung von GANs zur Erweiterung von Datensätzen werden eingeführt, um die Robustheit von KI-Systemen zu verbessern. Es wird vorgeschlagen, wie GANs genutzt werden können, um realistische visuelle Erklärungen zu erzeugen, die auf den Prinzipien kontrafaktischen Denkens basieren, um die Erklärbarkeit von KI-Systemen zu fördern. Es wird gezeigt, wie GANs verwendet werden können, um KI-Systeme ausdrucksfähiger zu machen, indem kontinuierlich konditionierte Bilder generiert werden, ohne auf kontinuierlich annotierte Trainingsdaten angewiesen zu sein. Es werden Ansätze vorgestellt, die personalisiertes textuelles und visuelles Feedback für ein Bewerbungsgesprächs-Trainingssystem bereitstellen. Schließlich wird gezeigt, wie GANs genutzt werden können, um interaktive Anwendungen zur Stressbewältigung zu entwickeln.

Die Beiträge dieser Arbeit zielen darauf ab, das Potenzial von GANs in einer breiten Palette von Forschungsfeldern aufzuzeigen.

ACKNOWLEDGMENTS

This thesis would not have been possible without the help of many people. First, I want to thank Prof. Dr. Elisabeth André, who was the best supervisor I could have wished for. By always giving me the greatest possible freedom to do what I wanted, she enabled me to fully enjoy my time at the lab and become the researcher that I am today. I also want to thank Prof. Dr. Cristina Conati and Prof. Dr. Bernhard Bauer for taking the time to review this work.

I am deeply grateful for having had the opportunity to work with incredibly kind and talented colleagues. In particular, I would like to thank Prof. Dr. Hannes Ritschel for directing my interest towards the chair's research fields already during my time as a Bachelor's and Master's student - without him, I would never have started at the lab. I would like to thank Dr. Florian Lingenfelser for supporting me with his down-to-earth perspectives from the very first day of my PhD. Many thanks also to my office mate Thomas Kiderle, who has walked the entire journey with me from day one. Very special thanks to Ruben Schlagowski, Stina Klein, Daksitha Withanage Don, Dr. Matthias Kraus, and Alexander Heimerl for becoming great friends along the way. I would also like to thank my former colleagues Prof. Dr. Tobias Huber, Dominik Schiller, and Dr. Katharina Weitz, who have significantly enhanced my research through countless fruitful discussions. Thanks to my parents, who have always supported me.

Finally, a very special *thank you* to my wonderful partner Laura Braun for supporting me in all aspects of my life over the past 12 years - without you, none of this would have been possible.

EDITORIAL REMARKS

ACADEMIC VOICE

The author has employed the academic “we” rather than the first-person singular throughout this work. This stylistic choice reflects the interdisciplinary collaboration and collective research approach that underpins this work.

UTILIZATION OF DIGITAL TOOLS

Digital tools, such as GPT models (GPT-3.5, GPT-4, GPT-4o, GPT-4o mini), DeepL Write, and Grammarly, have been used for language refinement and stylistic improvement to enhance clarity, coherence, and overall quality.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ACGAN	Auxiliary Classifier GAN
ADA	Adaptive Discriminator Augmentation)
AdaIN	Adaptive Instance Normalization
AI	Artificial Intelligence
AUC	Area Under The ROC Curve
ASMR	Autonomous Sensory Meridian Response
BCE	Binary Crossentropy
CCC	Concordance Correlation Coefficient
CNN	Convolutional Neural Network
CORR	Correlation
CUB	Caltech-UCSD Birds Dataset
DCGAN	Deep Convolutional GAN
DEQ	Discrete Emotions Questionnaire
DL	Deep Learning
DS	Deep Spectrum
ES	Evolutionary Strategy
FID	Fréchet Inception Distance
GAN	Generative Adversarial Network
GUI	Graphical User Interface
IoU	Intersection over Union
IS	Inception Score
JS	Jensen-Shannon
KL	Kullback-Leibler
LIME	Local Interpretable Model-Agnostic Explanations
LRP	Layer-wise Relevance Propagation
LVE	Latent Variable Evolution

M	Mean
MCC	Matthews Correlation Coefficient
MOS	Mean Opinion Score
MSE	Mean Squared Error
PCA	Principal Component Analysis
ProGAN	Progressive Growing of GANs
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RMS	Root Mean Square
ROI	Region Of Interest
ROC	Receiver Operating Characteristic
RQ	Research Question
SAGR	Sign Agreement
SD	Standard Deviation
SLIC	Simple Linear Iterative Clustering
SSIM	Structural Similarity Index Measure
SVM	Support Vector Machine
TiA	Trust in Automation Questionnaire
UEQ	User Experience Questionnaire
VAE	Variational Autoencoder
VMM	Vector Manipulation Module
WGAN	Wasserstein GAN
WGAN-GP	Wasserstein GAN + Gradient Penalty
XAI	Explainable Artificial Intelligence

Part I

BACKGROUND

1

INTRODUCTION

1.1 MOTIVATION

One trait that makes humans a unique species is the ability to be creative. As such, humans have learned to create beautiful things over the course of thousands and thousands of years. Over time, human culture developed, giving birth to art, architecture, literature, and music. The creative mind sets human cognition apart from lower forms of intelligence - humans do not function as mere reactive, impulsive beings, but have the gift of thinking up new things, of crafting things and objects that have never been seen or heard before.

This circumstance was arguably one of the most common objections to the term *Intelligence* in *Artificial Intelligence* (AI) for a very long period of time - how could AI ever be compared to the human way of thinking, if those algorithms are not at all capable of *creating new things*? In fact, the entire spectrum of different AI algorithms has focused almost exclusively on discriminative tasks for a long time, e.g., classifying objects, determining an optimized way of doing something, or solving a regression problem.

However, a sudden paradigm shift occurred when the breakthrough of generative models was launched by Goodfellow et al. (2014). With them introducing their concept of *Generative Adversarial Networks* (GANs), a simple but effective approach came to light, which would change the whole AI research landscape. From then on, AI systems were capable of *being creative*, resulting in a completely new hype about Generative AI, that, due to continuous new innovations (e.g., rather recent developments such as *Diffusion Models* or *Large Language Models*), continues to this day.

Using Deep Learning (DL) for Generative AI has opened up a whole lot of new possibilities, and ways of using corresponding systems are being explored continuously.

However, the whole field of DL-based generative AI is still *relatively* new. Although the fast and steady progression of algorithms and approaches, application scenarios and use-cases that make use of Generative AI are still in their infancy. Respective applications are often one-sided, although from a *technical* point of view, certain approaches (in particular GANs) have matured over the last years and are ready to be used to solve actual problems.

Showing these directions, i.e., showcasing how Generative AI can be utilized in a variety of problem domains, is what this thesis is all about.

1.2 RESEARCH OBJECTIVES

DL-based Generative AI should not be seen as "the next step" that supersedes everything that has been done before, but rather as an entirely new direction that exists in parallel with "discriminative" AI. From this perspective, interesting symbioses of the two fields emerge - Generative AI and non-Generative AI can *complement* each other, leading to a diverse set of new possibilities. Similarly to how human creativity completes the human logical-thinking being, Generative AI can enrich non-generative AI systems in various meaningful ways. For example, using the generative capabilities of such systems to synthesize training data might directly improve the robustness of discriminative models. Also, generative AI can be used to synthesize visual explanations that give insights into how a non-generative AI makes a decision. Additionally, as generating content on the fly also allows AI systems to dynamically react to a user, it has potential to enhance the interaction between a user and AI systems.

In the remainder of this work, we will focus on Generative Adversarial Networks (GANs), as this subfield of generative AI is already established in the field, and mature and efficient architectures exist.

Overall, by demonstrating how GANs can be used to address such diverse scenarios, we try to answer the question:

> *How can Generative Adversarial Networks* be used to improve, to explain, and to interact with Artificial Intelligence?

Specifically, we identified five particular problem domains that can benefit from the use of GANs:

- Robustness of AI Systems
- Explainability of AI Systems
- Expressiveness of AI Systems
- Feedback Synthesis
- Interaction with AI

In this work, we will provide exemplary concepts, ideas, and frameworks to each of the five areas as follows.

1.2.1 *Robustness*

To achieve robustness of an AI system, several factors need to be addressed. These factors encompass both technical and non-technical considerations – from problem modelling to deployment of the actual system. However, especially in the context of Machine Learning, one of the key factors to the functionality of a system is the data on which the underlying algorithm was trained. Often, the development of an AI system fails simply because not enough training data is available to approximate a robust model. A common strategy is therefore to artificially inflate a training data set, which is also known as *Data Augmentation*. Conventional data augmentation usually implements diverse transformations on the initial data set, thereby creating new instances that preserve the fundamental features of the input while introducing authentic variations.

Traditional data augmentation, although proven effective in numerous scenarios, comes with constraints and potential drawbacks.

One notable limitation is variability. Traditional augmentation techniques often rely on rudimentary transformations, such as rotation, flipping, and scaling for the image domain, or phase shifting and noise injection for the audio domain. While these transformations induce variability to the data to some extent, their scope may fall short of delivering the full spectrum of potential variations encountered in real-world data.

Additionally, applying data augmentation in non-optimal ways can support *overfitting*. This occurs when the model becomes overly attuned to the augmented data, reducing its adaptability to real-world instances that deviate from the augmented samples.

Generative AI, especially GANs, can help here. As those architectures are modelling the context of data instances in a much more complex way than traditional data augmentation techniques, they can be used to synthesize a more diverse, comprehensive set of augmented training data.

However, major challenges remain when using GANs for Data Augmentation. Probably the most critical issue is that - as GANs model the distribution of an original dataset they are trained on - they are still not capable of augmenting datasets with data that falls completely out of the original distribution. Therefore, they only work well if the original data already spans a distribution that contains the majority of the problem domain. This is not necessarily a problem when working with large datasets. There, data augmentation often is applied to only "fill certain holes" in the data distribution - the broader *scope* of the underlying data distribution is mostly already defined, as long as a critical mass of training samples exist. For smaller datasets, this is more of a problem. Here, training data might not be even close to covering the boundaries of the data distribution - and as such, a generative model trained on that data might still not fully cover that distribution. Therefore, ways have to be found how to use GANs to break out of that original training data distribution - especially for small datasets. As such, in this thesis, we will try to answer the question:

> *RQ1: How can we use GANs to augment small datasets without being stuck in the original dataset distribution?*

1.2.2 *Explainability*

Artificial intelligence becomes increasingly integrated into diverse aspects of our daily lives. It is important to comprehend the decisions undertaken by AI models, as this understanding is foundational for building trust, accountability, and ensuring ethical development of these systems. A critical point here is transparency. Many state-of-the-art AI models, particularly those belonging to the category of *Deep Learning*, function as "black boxes". As such, they pose difficulties for users to grasp the rationale behind their predictions or decisions. Explainability serves as a tool to unravel these black boxes, offering stakeholders, including end-users, insights into the reasoning behind specific decisions. This transparency is particularly important in domains where the AI's decisions carry serious real-world consequences, such as healthcare, finance, and criminal justice.

Furthermore, explainability plays a crucial role in enhancing accountability. As AI systems influence decisions affecting individuals or communities, mechanisms for attributing responsibility become essential. Explainable AI enables stakeholders to trace the decision-making process, identify potential biases, and ensure alignment with ethical standards and regulatory requirements. This accountability is crucial not only for legal and regulatory compliance but also for establishing a sense of responsibility among AI developers and organizations deploying these systems.

Beyond transparency and accountability, explainability contributes significantly to user acceptance. Users are more likely to trust and adopt AI systems when they can comprehend the decision-making process leading to conclusions. Whether in applications like autonomous vehicles or medical diagnostics, user trust in AI decisions is important. Explainability fosters this trust by demystifying the AI's decision-making process, reducing uncertainty, and providing users the ability to validate and understand the system's outputs.

Moreover, explainability proves invaluable in identifying and mitigating bias within AI models. If AI systems base decisions on biased training data, they risk discriminating underprivileged individuals. By offering explanations for model decisions, stakeholders can detect and counteract bias in such models.

Common techniques in explainable artificial intelligence (XAI) are often based on feature attribution, a methodology aimed at revealing the importance of specific features in a model's decision-making process. While these approaches have gained popularity for shedding light on which features contribute significantly to predictions, they have limitations. Feature attribution techniques predominantly show the relevance of specific features, but mostly fall short in communicating the reason for that relevance. In essence, they provide a *what* rather than a *why* perspective, leaving a critical gap in our understanding of the models. XAI algorithms that are based on the paradigm of *Counterfactual Reasoning* (i.e., Factual Explanations like Counterfactual Explanations or Semifcatual Explanations), on the other hand, can enhance XAI by offering insights into *why* specific features are relevant. As such, they provide a more comprehensive understanding of the model's inner workings.

However, a major challenge in the field of factual explanations, especially in the image domain, is that it is hard to generate *realistic* factual explanations. Further, current research mainly almost exclusively focuses on carrying out information about *important* features. The fact that knowledge about features that are explicitly *irrelevant* to an AI system also highly contributes to the model understanding gets mostly neglected. As such, in this thesis, we will try to answer the questions:

> *RQ2: How can we use GANs to generate realistic counterfactual explanations for image classifiers?*
>
> and
>
> *RQ3: How can we use GANs to build explanation systems that communicate information about irrelevant features?*

1.2.3 *Expressiveness*

In many cases, we want to generate data with a high degree of expressiveness. This means that we want to be able to steer certain characteristics of the data in a detailed and continuous way. However, respective models mostly rely on datasets that contain information about the features that we want to control during inference. For instance, when we want to control a specific feature in a continuous way, we need to train a model with the help of a dataset that includes such continuous annotations for that feature. Often, those datasets are not available, but only datasets with coarser annotations - like categorical labels - exist.

A specific example is the synthesis of emotional face images. It remains a challenge to train adequate, application-specific GAN models that allow for a continuous depiction of emotional faces (i.e., faces that can show arbitrary valence and arousal values), as this would commonly imply the need for continuously labeled datasets (i.e., annotated regarding valence and arousal), which are time- and resource-intensive to build. On the other hand, discretely labeled datasets, i.e., datasets of face images that are annotated with discrete emotions (like *happiness*, *sadness* or *anger*), are widely available. Further, the latter are also much easier to build for specific applications, as discrete labeling can be done faster than continuous labeling. Therefore, in this thesis, we will use the synthesis of emotional faces as examplary use-case to answer the question:

> *RQ4: How can we use GANs to synthesize continuously conditioned images by using only discretely labeled training data?*

1.2.4 *Feedback Synthesis*

Using generative AI, we can design systems and applications that empower users and foster self-esteem. One particular class of applications that address these objectives are coaching and teaching systems. Here, AI systems can provide constructive feedback and encouragement to users. Systems that recognize and leverage individual strengths can contribute to a positive sense of self-worth. By synthesizing personalized feedback based on users' unique abilities, AI can help individuals to appreciate their strengths, while individual weaknesses can be pointed out in order to improve on them. Job interview training systems are an excellent example of how AI can support self-esteem. These systems create a safe environment for individuals to practice and refine their interview skills, providing constructive feedback and reducing anxiety associated with real interviews. By using such systems, individuals can gain confidence and get prepared for real-world interviews - which might positively impact their self-esteem and self-worth.

However, it remains a challenge to build job interview training systems that give personalized, but still realistic and comprehensive feedback, as most existing approaches focus on specific features that the interviewee showed, while ignoring the feature context, i.e., the interrelation between the various contributing features. As such, in this thesis, we will try to answer the question:

> *RQ5: How can we use GANs to enhance an AI based job interview training system in order to give personalized, realistic and comprehensible feedback?*

1.2.5 *Interaction with GANs*

While discriminative AI is mostly use to solve well-defined tasks like classification or regression problems, generative AI can create entirely new data - we even could get the impression that generative AI is able to be *creative*. As such, it is conceivable that we want to directly participate in such creative processes - we want to *interact* with the AI. GANs in particular are a promising class of generative models to enable such interactions, as they - once trained - are capable of synthesizing data in real-time. In this thesis, we give a specific example how an interaction with GANs can be made possible. Specifically, we aim to build an interactive system that is able to counteract stress. There already exist a variety of methods and techniques to hinder and prevent stress. One of those methods is the consumption of content inducing an *Autonomous Sensory Meridian Response* (ASMR). ASMR refers to a physical relaxing feeling that is induced by certain auditory, visual, or tactile stimuli. In recent years, ASMR became particularly popular through platforms like YouTube, where many content creators produce videos that are specifically designed to trigger the state of ASMR. In literature, ASMR is described as a *flow-like mental state* (Barratt and Davis, 2015). Up to this point, ASMR is limited to being a *passive* experience - users only passively consume ASMR content. Although the highly related state of flow usually requires some sort of *activity* (Csikszentmihalyi, 2000), it has never been researched if some sort of *active* component can also enhance the stress-hindering relaxation capabilities of ASMR. GANs have been proven capable of synthesizing high-quality sound, and also are fast enough for being parametrized and used in real-time applications. As such, GANs can be used to realize both active *real-time parametrization* and passive *sound synthesis* components - and as such, to build an *interactive* ASMR system.

However, employing ASMR in an interactive setting remains a major challenge, as the topic is rarely researched and adequate interaction techniques are not yet known. As such, in this thesis, we will try to answer the question:

> *RQ6: How can we use GANs to build an interactive ASMR experience?*

1.3 STRUCTURE OF THE THESIS

This thesis is outlined as follows:

Part 1, i.e., the remainder of the part that you are currently reading, will give an introduction into the theoretical and technical concepts that this thesis is built upon. Mainly, this will cover the ideas behind the broad palette of GAN architectures that were used to implement the concepts of this work.

Part 2 presents two technical approaches on how GANs can be used to artificially increase the amount of training data for specific problem domains - as such, contributing to the robustness of AI systems.

Part 3 presents a technical approach on how to generate highly realistic counterfactual explanations for image classifiers. Further, it introduces the novel concept of *Alterfactual Explanations*, an explanation mechanism that tries to communicate irrelevant features of an AI. Also, an implementation to generate such Alterfactual Explanations by using GANs is presented.

Part 4 presents a technical approach on how to use GANs for affective face interpolation without the need for continuously labeled training data.

Part 5 presents two technical approaches on how to use GANs for creating personalized feedback in a job interview training setting. The first approach aims to generate verbal feedback, while the second one aims to synthesize highly realistic images that visualize how the trainee could have behaved better.

Part 6 introduces an interactive GAN-based application to support well-being. Therefore, we employ an application for an interactive ASMR experience.

Part 7 summarizes the contributions of this thesis and gives a brief outlook into future work.

2

GENERATIVE ADVERSARIAL NETWORKS

2.1 DISCRIMINATIVE VERSUS GENERATIVE MODELS

Figure 1: The discriminative and the generative detective (i.e., the former forger), representing discriminative and generative models.

To get a basic understanding of what a generative model *is*, we have to compare it to its natural counterpart: a discriminative model. Here, we want to start with an analogy. Think of a scenario where an art museum is dealing with a problem: an art forger is creating fake paintings which are sold as genuine masterpieces to the museum. To solve this problem, the museum director hires a detective with a solid education in the art business. This detective represents the discriminative approach. The discriminative detective's job is to quickly and accurately identify whether a painting is real or fake. For each painting, he looks for specific details - those that he learned are relevant for the decision of if the painting is real or fake. These details could be unusual brushstrokes, inconsistencies in color, or even minor mistakes in signature placement. These are called the *features*. Based on the features he observes, the discriminative detective makes a decision: is the painting real, or was it faked by the art forger? The detective's goal is to estimate the boundary between real works and forgeries as clearly as possible. In mathematical terms, he learns a function that directly reflects the probability that a painting is fake given the features that he had observed. This can be expressed as:

$$P(\mathrm{Fake}|\mathrm{Features}) = f(\mathrm{Features}) \tag{1}$$

Here, $f(\mathrm{Features})$ is the discriminative model. It focuses solely on making the best classification decision based on the features. In the realm of Machine Learning (ML), we are not (necessarily) looking at detectives, but at technical models that are able to solve problems on a computational level. Here, we can implement such a concept through a variety of technical approaches, for example through mechanisms from the field of *Supervised Learning*. As such, the function f would be refined by looking at lots of examples of real and fake paintings. The strategy is adjusted to minimize mistakes, ensuring a fake is rarely misclassified as real (and vice versa).

Let's get to the generative approach. As the art museum still has huge trouble with the art forgeries, the museum director has a different idea. He hires a former art forger (i.e., the *generative* detective) - as such, that former art forger is not just interested in identifying fakes. The art forger isn't just looking at a finished painting. Rather, he thinks about how the painting was made - how the artist chose their colors, applied their brushstrokes, and composed the scene. The forger is trying to model the entire process of creating a painting, whether it's a real painting or a forgery. The forger develops two models in his mind. The *genuine painting model* (i.e., how a real artist might have created the painting) and the *fake painting model* (i.e., how a forger might try to replicate this process, and what small errors or differences might sneak in). Mathematically, the hired former art forger, when helping to identify forgeries, calculates the probability of seeing a particular painting given that it's a real piece $P(\mathrm{Features}|\mathrm{Real})$ and the probability of seeing it given that it's a fake $P(\mathrm{Features}|\mathrm{Fake})$. He then can combine these probabilities with his understanding of how often real and fake paintings appear - the *prior probability*. Using Bayes' theorem, he can calculate:

$$P(\mathrm{Fake}|\mathrm{Features}) = \frac{P(\mathrm{Features}|\mathrm{Fake}) * P(\mathrm{Fake})}{P(\mathrm{Features})} \tag{2}$$

This calculation gives him the likelihood that a painting is fake after considering both the process of creation and the features observed.

However, additionally to being able to identify fake paintings, the former art forger could *use this deep understanding to create forgeries by himself* - at least, if he were still in business.

In simple terms, the *discriminative detective* wants to classify the painting as real or fake only based on the features he observes. He does not worry about the process of how the painting was made, just about the end result. Discriminative *Models* work similar - they focus on assessing an observation, and not necessarily on a deeper understanding of how the observation came about. The *generative detective*,

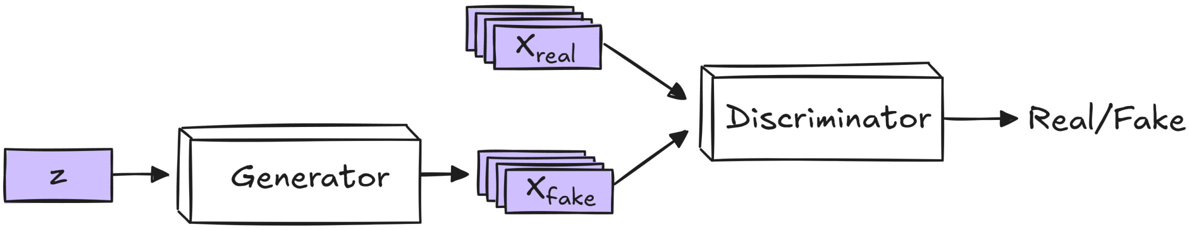


Figure 2: A schematic overview of the Vanilla GAN architecture.

however, is interested in the whole process - the story behind the painting's creation. He understands the techniques and methods used to produce both real and fake paintings, and uses this understanding to either make judgments, or create forgeries. The same goes for Generative *Models* - they try to model a deep understanding of how data comes about, i.e., they try to model the whole *Data Distribution*.

2.2 THE IDEA OF GANS

Now that we have seen the key differences between discriminative and generative approaches, let's try to understand how both paradigms are combined in the idea of *Generative Adversarial Networks*.

In our little forgery tale, the generative and the discriminative detective are again on their way to investigate a high-profile case of potential forgery. Unfortunately, they suffer a terrible car accident - which leaves them both with amnesia and erases the majority of their memories and skills. The generative detective, whom we simply will call the *Generator* from now on, and the discriminative detective, (from now on, simply the *Discriminator*), must now start from scratch. They have to relearn everything about their respective skills. However, they still both want to approach their tasks in their distinctive manner: the Generator aims to learn the process of faking art, while the Discriminator wants to assess if an existing painting is real or fake based on observations. This time, there's a new twist: during their detective work, they've become friends. As such, instead of learning their goals independently, they join forces and rebuild their expertise through a process of mutual learning - through *adversarial* interaction.

The Generator begins its work from zero, producing rudimentary paintings that are little more than abstract blobs of color. These early efforts are far from convincing and are more akin to random experiments than good forgeries. The Discriminator is equally bad. With no memory of his former training, he has lost his eye for detail and his ability to spot forgeries. His initial attempts at evaluating paintings are as unskilled as the Generator's attempts to create them.

Despite their difficulties, the Generator and Discriminator are committed to regaining their lost abilities. They begin to learn from each other in an *adversarial game*:

- ROUND 1. The Generator creates a painting that is more of a chaotic collection of shapes and colors than a recognizable piece of art. The Discriminator tries to judge whether the Generator's creation is real or fake. Also, he looks at real paintings and tries to assess their validity - but, as inexperienced as he is, he can only make random guesses.

- ROUND 2. The Generator, eager to improve, takes the Discriminator's feedback (even though it's not very accurate) and attempts to create a slightly more structured painting. The Discriminator, having a first impression of the domain now, begins to recognize very basic patterns and features that might indicate a fake.

- ONGOING ROUNDS. As they continue to interact, both the Generator and the Discriminator gradually regain their lost skills. The Generator's paintings become more sophisticated with each round, while the Discriminator sharpens his ability to spot flaws and inconsistencies. Over time, this adversarial process leads both of them to a level of skill that matches their pre-accident expertise.

This scenario, where both the Generator and Discriminator start from nothing and improve together, mirrors how GANs work. Now, let's look at the maths behind the approach.

THE GENERATOR'S OBJECTIVE. The Generator G starts by creating random samples $G(z)$ from noise z. These early outputs are not convincing, but with feedback from the Discriminator D, the Generator learns to produce more realistic images. The Generator's goal is to create samples $G(z)$ that make the Discriminator believe they are real, i.e., $D(G(z)) \approx 1$. As such, the Generator's objective is:

$$\min_G V(G) = \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{3}$$

$\mathbb{E}_{x \sim p}$ refers to the ***expected value*** *(or average) over the distribution p. It indicates averaging over all possible values of x, where each x is drawn according to the probability distribution p.*

THE DISCRIMINATOR'S OBJECTIVE. The Discriminator attempts to distinguish between real and fake data. Initially, it struggles, but as it sees more examples, it improves its ability to classify real versus fake data correctly. The Discriminator aims to maximize the probability of correctly identifying real data $D(x) \approx 1$ and rejecting fake data $D(G(z)) \approx 0$. We can formulate it as:

$$\max_{D} V(D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (4)$$

THE ADVERSARIAL MINMAX GAME. Both the Generator and the Discriminator engage in a minmax game. The Generator seeks to minimize the Discriminator's ability to distinguish between real and fake paintings, while the Discriminator aims to maximize its accuracy in identifying real versus fake paintings. As such, they both *compete* against each other, hence the term *adversarial*. The whole interaction can be mathematically formulated as follows:

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (5)$$

Here, $V(D, G)$ is the value function that both the Generator and the Discriminator are optimizing. There exists a multitude of different approaches based on GANs. However, this value function, used as *Adversarial Loss*, forms the basis of all of these methods. Figure 2 depicts a schematic overview of the GAN paradigm.

2.3 VANILLA-GAN

The basic concept of GANs - where a Generator creates data and a Discriminator evaluates it - provides a powerful framework for producing realistic synthetic data. To carry this concept over into the field of modern machine learning, particularly deep learning, we use neural networks to implement both the Generator and Discriminator. The original and most basic form of a GAN, introduced by Goodfellow et al. (2014), uses fully connected neural networks for both the Generator and Discriminator. In their original paper, they did not define *one* single architectural specification, they only set the rough scope of using *Multilayer Perceptrons*, i.e., simple fully connected neural networks. They presented the GAN idea more as a *framework* than a specific model architecture - as such, the original GAN idea is applicable to all kinds of architectures. More importantly, what they *did* define in detail is the exact approach of *training* the model. In a real-world implementation, deploying GANs requires a methodical, step-by-step computational strategy. Attempting to fully optimize the Discriminator within each training cycle is not only computationally expensive but also increases the risk of overfitting, especially when working with limited datasets. To circumvent this, the training process alternates between multiple optimization steps for the Discriminator and a single optimization step for the Generator. This alternating procedure ensures that the Discriminator remains nearly optimal, provided that the Generator's updates occur gradually. The training algorithm presented by Goodfellow et al. (2014) is presented in Algorithm 1.

Algorithm 1 Basic GAN training loop as presented by Goodfellow et al. (2014).

```
for number of training iterations do
    for $k$ steps do
        • Sample minibatch of $m$ noise samples $\{z^1, ..., z^m\}$ from noise prior $p_g(z)$.
        • Sample minibatch of $m$ examples $\{x^1, ..., x^m\}$ from data generating distribution $p_{data}(x)$.
        • Update the Discriminator by ascending its stochastic gradient:
          $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^i\right) + \log\left(1 - D\left(G\left(z^i\right)\right)\right)\right].$
    end for
    • Sample minibatch of $m$ noise samples $\{z^1, ..., z^m\}$ from noise prior $p_g(z)$.
    • Update the Generator by descending its stochastic gradient:
      $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^i\right)\right)\right).$
end for
```

2.4 Problems with the Vanilla GAN Framework

While the basic principles of the GAN framework have laid a solid foundation for generative modeling, the initial Vanilla GAN approach encountered several challenges and limitations. Understanding the most critical problems with the original idea is necessary for a good understanding of the idea itself. As such, in this section, we will introduce the three most common challenges that arose with Goodfellow's work: *Mode Collapse*, *Vanishing Gradients* and *Non-Convergence*.

2.4.1 *Mode Collapse*

Let's go back to our story about the two detectives, who recently lost the majority of their memory in the tragical accident and are now trying to re-learn their skills in a GAN-like manner. As their training continues, the generative detective (Generator), who initially struggled to produce anything even resembling a real painting, has made a breakthrough. One day, after many rounds of trial and error, the Generator realizes that painting vague, minimalist landscapes — a few hills, a sunset, and some clouds — reliably gets approval from the discriminative detective (Discriminator). As such, the Generator starts focusing only on this one style. Over and over, he paints only these familiar landscapes. Perhaps he adds a slight variation, like a few more clouds or a brighter hue in the sky, but he never moves far away from this approach. The Generator, once open to creativity, has *collapsed* into this one safe formula. As such, the cooperation between the two detectives begins to suffer. Without diversity in the artworks, the Generator fails to grow, and the Discriminator cannot get better because he is not confronted with new material.

In GANs, this is known as *Mode Collapse*. The Generator, instead of capturing the full diversity of real data, finds a *safe spot* – a subset of data that can repeatedly fool the Discriminator without needing to learn the range of features in the dataset. As a result, the Generator produces outputs that lack variety, generating only a limited subset of the data distribution, which limits the GAN's overall performance.

Mode Collapse can be mathematically understood by examining how the Generator's distribution $p_g(x)$ becomes overly focused on specific areas of the data distribution p_{data}, rather than converging to the full distribution. Mode Collapse is linked to the optimization dynamics. Instead of achieving *global* minima, the Generator finds *local* minima by generating a limited set of outputs. Because the Discriminator's success is localized, this local minimum doesn't penalize the Generator strongly enough for failing to capture the full distribution.

2.4.2 *Vanishing Gradients*

Back to our story: After weeks of learning, the discriminative detective's skills have reached an impressive level. Now, he can instantly spot the Generator's forgeries. No matter what the Generator produces, the Discriminator instantly declares it fake. However, his judgment is unhelpful, leaving the Generator frustrated. Without any specific feedback, the Generator can't tell what's working and what's not – it only knows that every attempt is vehemently rejected. The situation becomes a deadlock. The Generator, despite its best efforts, receives almost no constructive feedback, as the Discriminator's judgments offer no insight into what a more realistic painting might look like. Without constructive feedback, the generative detective feels lost.

In GANs, this problem is known as *Vanishing Gradients*, where the Discriminator's overly confident predictions lead to near-zero gradients, starving the Generator of the information needed to improve. As the Discriminator reaches nearly optimal performance, it approaches certainty (probability values close to 0 or 1), which in turn causes the Generator's gradient updates to diminish, which in turn leads to stalled training.

Vanishing Gradients are mathematically tied to the structure of the GAN's gradient updates, especially when D achieves near-perfect accuracy. The Generator seeks to minimize:

$$\mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]$$

For each z, the Generator's gradient is proportional to $\nabla_\theta \log(1 - D(G(z)))$.

When $D(G(z))$ is close to 0 (a near-certain "fake" judgment), the gradient $\nabla_\theta \log(1 - D(G(z)))$ approaches zero. This is because:

$$\frac{\partial}{\partial\theta}\log(1 - D(G(z))) \to 0 \quad \text{as} \quad D(G(z)) \to 0.$$

Consequently, the Generator stops receiving meaningful updates, making training stall.

2.4.3 *Non-Convergence*

One day, the generative detective decides to experiment more. He starts doing drastic changes very frequently, shifting its style from minimalist landscapes to hyper-detailed cityscapes to abstract art. The Discriminator recognizes that the styles change frequently, and also starts to frequently change its evaluation criteria more often. As they both jump from one approach to another without consistency, their interactions become chaotic. Each day, the Generator tries a completely new style, and each day the Discriminator uses a different metric to assess the painting's realism. This instability grows until they're no longer progressing. Both are stuck in a constant loop of trying to outdo one another without learning, unable to find a steady rhythm. The situation turns into an unpredictable, endless back-and-forth, with no convergence in sight.

In GANs, this *Non-Convergence* happens when the Generator and Discriminator fail to reach equilibrium. The loss objectives may shift in an uncoordinated way, causing both networks to oscillate in a manner that prevents stable training and convergence. This instability can prevent both the Generator and Discriminator from learning effectively. Often – but not always – Non-Convergence is a result of mismatching learning rates η of the Discriminator and the Generator. When $\eta_D \gg \eta_G$ (Discriminator's learning rate much larger than Generator's learning rate), D can adapt too quickly, causing the Generator's updates to be ineffective as the Discriminator shifts unpredictably. Vice versa, when $\eta_G \gg \eta_D$, the Generator may produce overly aggressive changes, causing D to be unable to adapt, leading to oscillations or a collapse in training.

2.5 COMMON MODIFICATIONS TO THE GAN TRAINING PROCESS

Fortunately, by the time, a variety of modifications have been introduced that address the aforementioned issues. The most important modifications will be explained in the following sections.

2.5.1 *WGAN*

One of the key advancements in tackling these issues came through the introduction of the *Wasserstein GANs* (WGAN), which make use of the *Wasserstein Distance*. The Wasserstein Distance (also known as *Earth Mover's Distance*) offers a more stable way of measuring how far apart two distributions are compared to the Jensen-Shannon Divergence (JS Divergence) used in the original GAN framework. This more advanced distance measure provides the foundation for WGAN, as introduced by Arjovsky, Chintala, and Bottou (2017).

The term "Earth Mover's Distance" comes from a different analogy: imagine two piles of soil representing different probability distributions. The Wasserstein Distance quantifies the minimum amount of work (or cost) needed to transform one pile into the other by "moving soil."

For the very last time, let's return to our detectives, which are still trying to improve themselves in their adversarial setting. In their effort, the Generator still produces a painting that it hopes will fool the Discriminator. However, instead of simply declaring the painting as *real* or *fake* (as he did before), the Discriminator now evaluates how much work would be required to transform the Generator's painting into a real painting - the discriminative detective now assesses how far the colors and brushstrokes of the generated artwork are from being an original. The Discriminator imagines a palette of paints laid out to create a painting. If the Generator's painting is off, the Discriminator provides feedback based on how many *paint strokes* of a certain paint need to be added or adjusted to achieve the ideal artwork. This more nuanced evaluation gives the Generator continuous feedback, much like an artist receiving specific suggestions on how to refine their work. This is analogously to measuring the *Wasserstein Distance*. To see why the Wasserstein Distance is more effective than the JS Divergence used in Vanilla GANs, let's look at the maths. In a Vanilla GAN, the objective is based on minimizing the JS Divergence between the real data distribution $p_{data}(x)$ and the Generator's distribution $p_g(x)$. This approach can struggle when there is little overlap between the two distributions as that leads to the aforementioned problem of vanishing gradients. In this scenario, the Generator receives vague feedback, much like an artist who is told their work is "not good enough" without knowing how to improve. The Wasserstein Distance, however, measures the minimum "cost" of transforming one distribution into another. As such, it quantifies the effort required to turn the Generator's painting into a real-looking piece. Mathematically, it can be expressed as:

$$W(p_g, p_{data}) = \inf_{\gamma \in \Pi(p_g, p_{data})} \mathbb{E}_{(x,y)\sim\gamma} [\|x - y\|]$$

In this equation, $\Pi(p_g, p_{data})$ represents all possible ways to pair points from the generated artwork with points from the real data distribution. The Wasserstein Distance finds the optimal pairing that minimizes the amount of "paint" that needs to be moved to transform the Generator's work into something that closely resembles an original. As such, in a GAN using the Wasserstein Distance (i.e., WGAN), the role of the Discriminator shifts slightly. As instead of simply classifying data as "real" or "fake," it assigns a score based on the realism of the Generator's forgeries, the Discriminator becomes a *Critic*. The Critic's job is to estimate the Wasserstein Distance between the real artworks p_{data} and the faked ones p_g. As a consequence, the WGAN training loop differs, too. The ultimate objective of the Critic is to provide a useful estimate of the Wasserstein distance. However, the Critic does not compute this distance directly as a single value - instead, it learns a function that approximates how far apart these two distributions are apart. The Critic's training involves maximizing the difference between its scores for real samples and its scores for generated samples. The training objective can be expressed as:

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_{data}}[D(x)] + \mathbb{E}_{z \sim p_z}[D(G(z))]$$

In this expression, the Critic is effectively trying to maximize the score difference. In other words, the Critic's training maximizes the scores it assigns to real samples while minimizing the scores for generated samples. This process leads to a better approximation of the Wasserstein distance between the two distributions over time.

In WGANs, ensuring that the Critic properly estimates the Wasserstein Distance necessitates certain constraints, primarily *Lipschitz continuity*. A function is said to be Lipschitz continuous if there exists a constant L such that for any two inputs x and y:

$$|f(x) - f(y)| \leqslant L\|x - y\|$$

Especially, when $L = 1$, the function is specifically *1-Lipschitz continuous*, which means that the change in the function's output is restricted to be no greater than the distance between inputs. This property ensures that the function does not change too rapidly, which is crucial for the stability of the training process. In the context of the Critic in WGANs, Lipschitz continuity ensures that small changes in the input (i.e., minor changes in the generated or real artworks) lead to bounded changes in the output score. To enforce this Lipschitz constraint, the original WGAN proposed a simple yet effective method: *weight clipping*. This involves limiting the weights of the Critic network to lie within a predefined range, typically [-0.01, 0.01]. By constraining the weights, the function represented by the Critic becomes more stable and is less likely to exhibit rapid fluctuations, which would lead to significant changes in the output score for small input variations.

2.5.2 *WGAN-GP*

The primary function of weight clipping in WGAN is to restrict the weights of the Critic network to a specific range, ensuring that the network adheres to the Lipschitz constraint. However, with overly constrained weights, the Critic may be unable to accurately learn complex decision boundaries or capture the nuances in data distributions. As a result, this can reduce the quality of the feedback it provides to the Generator, hindering the GAN's overall ability to generate high-quality synthetic data.

To address these limitations, the *Wasserstein GAN with Gradient Penalty (WGAN-GP)* was proposed as an alternative to weight clipping. Instead of enforcing the Lipschitz condition by constraining weights, WGAN-GP incorporates a *gradient penalty* in the Critic's loss function, which has shown to improve both training stability and performance.

The gradient penalty in WGAN-GP encourages the gradients of the Critic's output with respect to its input to have a norm of 1. This is achieved by modifying the Critic's loss function to include a penalty term that calculates the deviation of the gradient norm from 1. In other words, WGAN-GP aims to keep the gradient norm close to 1 at all points, enforcing the Lipschitz constraint without directly constraining the network's weights. The modified loss function for the Critic in WGAN-GP is given by:

$$\mathcal{L}_{\text{Critic}} = \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r}[D(x)] + \lambda \mathcal{L}_{\text{GP}}$$

where $\mathbb{P}_g$ and $\mathbb{P}_r$ denote the generated and real data distributions, respectively, and λ is a hyperparameter that controls the contribution of the gradient penalty term. The gradient penalty term, $\mathcal{L}_{\text{GP}}$, is calculated as follows:

$$\mathcal{L}_{\text{GP}} = \mathbb{E}_{\hat{x}} \left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right]$$

where $\hat{x}$ represents a set of points sampled uniformly along straight lines between pairs of real and generated samples. By sampling from these interpolations, the penalty encourages smooth transitions between real and generated data regions, ensuring that the Critic maintains the Lipschitz condition.

The gradient penalty approach provides a softer and more continuous enforcement of the Lipschitz condition compared to weight clipping. This results in a more stable training process, reducing instances of mode collapse and other training instabilities commonly seen in traditional GANs.

In practice, the gradient penalty coefficient λ plays a crucial role in the performance of WGAN-GP. While larger values of λ enforce the Lipschitz constraint more strictly, they may also slow down convergence. On the other hand, smaller values of λ may lead to insufficient regularization, resulting in instability. Empirically, $\lambda = 10$ is commonly used and has been shown to perform well across a range of applications (Gulrajani et al., 2017).

2.5.3 *Spectral Normalization*

While *WGAN-GP* addresses the limitations of weight clipping by introducing a gradient penalty, another alternative to enforcing the Lipschitz constraint in WGANs is *Spectral Normalization*. This method offers a simpler, computationally efficient way to control the Lipschitz continuity of the Critic while preserving its expressive power by enforcing the *1-Lipschitz continuity* condition across the network. As already stated, in the context of WGANs, the Critic function D is required to be *1-Lipschitz continuous* to provide valid Wasserstein distance estimates. While the gradient penalty, that was explained in the section before, helps maintain a stable Lipschitz constraint, it also involves additional gradient computations on interpolated samples, adding computational overhead.

Spectral Normalization offers a more efficient approach, enforcing the Lipschitz constraint by controlling the *spectral norm* of each layer's weight matrix. This spectral norm, the largest singular value of the weight matrix, bounds the layer's output change with respect to its input. By normalizing each layer's weights by their spectral norm, Spectral Normalization ensures that the Critic adheres to 1-Lipschitz continuity, allowing smooth gradient flow without constraining the individual weights directly.

In the following, we will explain how Spectral Normalization works. For any layer in the Critic network, let W be the weight matrix. The spectral norm $\sigma(W)$ represents the maximum *stretching* factor the matrix applies to any vector, computed as the largest singular value of W. Spectral normalization approximates this largest singular value using *power iteration*, an efficient iterative method for singular value estimation. There, first a random initial vector v_0 that will be iteratively refined is defined. This vector represents an arbitrary input direction for which we want to find how much the matrix W can stretch it:

$$v_0 \sim \mathcal{N}(0, I)$$

Normalize v_0:

$$v_0 = \frac{v_0}{\|v_0\|_2}$$

Then, for each iteration k, the following steps are performed:

1. Multiply the current vector by the weight matrix W:

$$w_k = W v_k$$

2. Normalize the result to prevent the vector from growing too large:

$$v_{k+1} = \frac{w_k}{\|w_k\|_2}$$

After several iterations, the normalized vector ν_k converges towards the direction of the eigenvector associated with the largest singular value (spectral norm) of W:

$$\sigma(W) \approx \frac{w_k^T \nu_k}{\|\nu_k\|_2}$$

The weight matrix W is then normalized by its estimated spectral norm:

$$\hat{W} = \frac{W}{\sigma(W)}$$

This normalized weight matrix $\hat{W}$ ensures that the Lipschitz constant for each layer remains close to 1, enforcing 1-Lipschitz continuity throughout the Critic network. During training, the Critic uses the normalized weight matrix $\hat{W}$, preserving the 1-Lipschitz condition across all layers. This enables the network to provide stable and reliable gradients to the Generator without the expressiveness limitations imposed by weight clipping.

Spectral Normalization has become widely adopted across different types of GANs due to its balance of efficiency and effectiveness. It is particularly useful in applications where computational resources are limited, or when training GANs with large Critic networks that would suffer from weight clipping constraints.

2.6 EVALUATION METRICS FOR GANS

As the applications of GANs continued to expand, the demand for robust evaluation metrics to assess their performance has become increasingly critical. Unlike traditional discriminative models, where evaluation is often straightforward using well-established metrics such as accuracy, precision, recall, F1 score, or area under the ROC curve (AUC), the evaluation of generative models presents unique challenges. Discriminative models are typically designed to classify or predict outcomes based on input data, allowing for clear and quantifiable performance assessments. For instance, accuracy measures the proportion of correct predictions, while precision and recall evaluate the relevance of positive predictions and the model's ability to identify all relevant instances, respectively. In contrast, evaluating GANs is inherently more complex due to the nature of their outputs, which are new data instances that can vary significantly in quality, fidelity, and realism - there is no simple *True* or *False*. This complexity arises from several factors: the generated data must be assessed against a target distribution, and evaluation must encompass not only the quality of individual instances but also the overall diversity of the generated samples. Fortunately, several commonly employed metrics exist for assessing GANs, and these will be explored in the subsequent sections.

2.6.1 *Inception Score*

Introduced by Salimans et al. (2016), the Inception Score (IS) has become a standard tool in the evaluation of generative models. It is designed to measure two essential properties of a set of generated images: their *quality* and *diversity*. Quality refers to how realistic and coherent the images appear, while diversity assesses whether the generated images cover a broad range of different visual classes. The score relies on a pretrained classifier — commonly the Inception v3 model (Szegedy et al., 2016), trained on the ImageNet dataset (Deng et al., 2009) — as a proxy for evaluating these two dimensions.

In order to do so, the IS calculates the *Kullback-Leibler (KL) divergence* between two separate distributions: (i) the conditional distribution $p(y|x)$, which represents the probabilities (e.g., the proxy classifier outputs) of different labels for a given image x, and (ii) the marginal distribution $p(y)$, which represents the overall distribution of classifier-assigned labels across all generated images. Here, an image of high quality should result in a conditional distribution $p(y|x)$ that majorly revolves around a single class. Such a peak would imply that the classifier is confident in assigning the synthesized image to a specific category - and certainly, if an image is of high quality, then the proxy classifier should have an easy time recognizing what is depicted in the image.

On the other hand, *diversity* in the image set ensures that the marginal distribution $p(y)$ is spread out across multiple classes. The KL-divergence measures how much information the conditional distribution $p(y|x)$ contributes to the marginal distribution $p(y)$. By averaging the KL-divergence across all images in the dataset, IS provides a single numerical score that models both quality and diversity.

Mathematically, the Inception Score is defined as:

$$\text{IS} = \exp\left(\mathbb{E}_{x\sim p(x)}\left[D_{\text{KL}}(p(y|x)||p(y))\right]\right),$$

where $D_{\text{KL}}(p(y|x)||p(y))$ is the KL-divergence, and $\mathbb{E}_{x\sim p(x)}$ denotes the expectation over all generated images x. The exponential ensures that the score remains positive and interpretable. A high score indicates that the model generates high-quality, diverse images, while a low score suggests issues with image quality or diversity.

The computation of IS involves several steps. First, the generative model is used to produce a large set of images. The typical recommendation is to generate at least 50,000 images to ensure stable results (Salimans et al., 2016). Each image is then passed through the Inception v3 model, which outputs a probability distribution over the ImageNet classes. These distributions are used to calculate both

$p(y|x)$ for individual images and $p(y)$, the overall marginal distribution. The KL-divergence is computed for each image, and the average KL-divergence is exponentiated to yield the final score.

The Inception Score has become a widely used metric in the evaluation of GANs due to its intuitive interpretation and computational efficiency. By using a pretrained classifier, it avoids the need for human labeling or manual evaluation, providing a quick and scalable method for assessing image quality. Additionally, because the Inception Score reflects both quality and diversity, it aligns well with the goals of most image-generation tasks. However, the metric is not without limitations. One major drawback is its reliance on the Inception v3 model, which was trained on the ImageNet dataset. This creates a bias toward datasets that share similar class distributions or visual characteristics with ImageNet. For instance, if the generated images fall outside the domain of ImageNet classes, the score may be misleading. If the GAN was trained on producing data of a completely different domain, another proxy model has to be trained - which, besides inducing a lot of effort - makes it hard to set the numerical IS results in context due to a lack of comparability.

Another limitation is its insensitivity to *mode collapse*. The Inception Score does not explicitly penalize such behavior, as it does not directly analyze the diversity *within* each class (only across different classes). Furthermore, IS assumes that the pretrained classifier serves as a proxy for human perception, which may not always hold true, particularly for tasks involving subjective or aesthetic judgment.

Despite these limitations, the Inception Score has seen widespread adoption and remains one of the most cited evaluation metrics in generative modeling research.

2.6.2 *Fréchet Inception Distance*

Introduced by Heusel et al. (2017), the Fréchet Inception Distance (FID) has become one of the most reliable and frequently used metrics for assessing the quality and diversity of generated images. Unlike the IS, which focuses on the conditional distribution of class probabilities, FID measures the similarity between the distributions of real and generated images in the feature space of a pretrained classifier. This approach provides a more direct and robust comparison of the statistical properties of real and generated data. The FID is grounded in the assumption that real and generated images can be represented as multivariate Gaussian distributions in a feature space. Typically, the feature space is defined by the activations of the penultimate layer of the Inception v3 network, again pretrained on the ImageNet dataset. The FID measures the Fréchet distance (also known as the *Wasserstein-2 distance*) between these two Gaussian distributions. Specifically, it quantifies how similar the means and covariances of the real and generated data distributions are, providing an assessment of both quality and diversity.

Mathematically, the FID is defined as:

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g}\right),$$

where:

- μ_r and Σ_r are the mean and covariance of the real data distribution in the feature space.
- μ_g and Σ_g are the mean and covariance of the generated data distribution in the feature space.
- $\|\mu_r - \mu_g\|_2^2$ is the squared Euclidean distance between the means of the two distributions.
- $\text{Tr}(\cdot)$ denotes the trace of a matrix, summing its diagonal elements.

The term involving the square root of the covariance matrices accounts for the interaction between the distributions and captures how the shapes of the two distributions align. Lower FID values indicate that the generated images are closer to the real images in the feature space, implying higher quality and diversity.

To compute the FID, several steps are followed. First, a set of real images and a corresponding set of generated images are passed through the Inception v3 model to extract their feature representations from the penultimate layer. The means and covariances of the feature representations are calculated for both datasets. These statistics are then used to compute the FID using the formula above. Like the Inception Score, the FID typically requires a large number of images - typically at least 10,000 samples — to produce stable and meaningful results. One of the key advantages of the FID over the Inception Score is its ability to detect *mode collapse*. By comparing the distributions of real and generated data, FID can identify discrepancies in diversity, even if the individual generated images are realistic. Additionally, FID is not restricted to datasets that align with the class distributions of ImageNet, as it does not rely on class labels or predictions. Instead, it uses the Inception model purely as a feature extractor, making it more versatile and applicable to a wide range of generative tasks - at least in the image domain.

However, FID is not without its limitations. Its reliance on the Inception v3 model means that it may not accurately capture perceptual quality for domains significantly different from ImageNet. For instance, datasets with highly abstract or artistic content may not align well with the feature space of the Inception network, potentially leading to misleading FID values. Furthermore, FID assumes that the feature distributions are Gaussian, an assumption that may not hold true in practice for complex datasets. This can result in inaccuracies when the real or generated data distributions deviate significantly from Gaussianity. Another limitation is the sensitivity of FID to the number of samples used for its computation - insufficient samples can lead to unstable estimates of the means and covariances - and as (in contrast to the IS) it also involves using *real* data, it can only be applied if a sufficiently large dataset is available at all.

Despite these limitations, the FID has become a gold standard metric in GAN research due to its robustness and interpretability. Variants of FID have been proposed to address its limitations, such as sensitivity to sample size and Gaussian assumptions (Jayasumana et al., 2024). Also, extensions to the FID exist that additionally account for conditional information in the synthesis process (Soloveitchik et al., 2021).

2.6.3 *Manual Evaluation*

Despite the widespread use of quantitative metrics like IS and FID, manual evaluation remains essential in many contexts, particularly when perceptual quality and subjective attributes such as artistic style or realism are important. Metrics like IS and FID, while useful, often fail to fully capture fine-grained details, semantic coherence, or aesthetic preferences. Therefore, manual evaluation is still used extensively. There, human evaluators judge the quality of the generated content - but can also assess other metrics. E.g., this is not only crucial for evaluating the quality, but also for evaluating semantic consistency in conditional GANs, ensuring that generated images align with specific input conditions. To achieve manual evaluation in a structured way, various methodologies exist. One common approach is A/B testing, where evaluators compare pairs of images generated by different models and select the one they find more realistic or task-appropriate. The easiest form of A/B testing is to let evaluators identify whether an image is real or fake. Another popular method is the *Mean Opinion Score (MOS)*. Here, evaluators rate images on a Likert scale based on criteria like realism or quality. However, manual evaluation is not only useful for assessing the quality or consistency of generated data. Also, it can be used to assess a variety of different characteristics that are specific to certain use-cases or scenarios. Here, user studies are often the measure of choice. For example, when evaluating explanation systems that are based on generative AI, we could use such user studies to evaluate if generated explanations are actually helping users to understand an AI system. In this thesis, we mostly use generative systems for specific purposes that go beyond generating high-quality data. As such, we mostly use this form of evaluation - we conduct user studies to evaluate if our systems actually do what we want them to do.

2.7 ADVANCED GAN ARCHITECTURES

While Vanilla GANs introduced an effective framework for generative modeling, they exhibit limitations when applied to more complex tasks, particularly in handling high-dimensional data like large, detailed images. The basic architecture lacks scalability and struggles to produce realistic results at higher resolutions. Additionally, without advanced mechanisms for feature extraction, such as convolutional layers, the generated outputs often fail to capture fine details and textures, resulting in lower-quality and less convincing outputs. Because of that, more complex architectures have been proposed in the past. In the following sections, we will briefly introduce the most important architectural milestones.

2.7.1 *Deep Convolutional GANs*

One of the first extensions that were made to Goodfellow's original GANs was the addition of convolutional layers to the architecture. By doing so, more complex feature maps could be learned, enabling to understand and handle more complex datasets. However, as we want to *generate* data, convolutional layers (e.g., dimensionality *reduction*) is not enough - at some point we also need to increase our dimensions again in order to produce the desired outputs. *Deep Convolutional GANs* (DCGANs), an architecture introduced by Radford, Metz, and Chintala (2015), do that by including *transposed* convolutional layers. Transposed convolution, also referred to as deconvolution, is a fundamental operation that enables DCGANs the generation of high-resolution outputs from lower-dimensional inputs. Despite the term *deconvolution*, this operation does not reverse the mathematical process of convolution. Instead, it performs a *learnable* upsampling. Its primary role is to progressively increase the spatial dimensions of feature maps in the generator, transforming a low-dimensional noise vector into a high-resolution image.

To understand transposed convolution, it is helpful to compare it to standard convolution. In standard convolution, the spatial dimensions of the input are often reduced through the application of a kernel that slides across the feature map. For an input of size $H_{\text{in}} \times W_{\text{in}}$, the output size after applying a kernel of size $k \times k$, with stride s and padding p, is given by:

$$H_{\text{out}} = \frac{H_{\text{in}} - k + 2p}{s} + 1, \tag{6}$$

and similarly for the width W_{out}. Transposed convolution, in contrast, *increases* the spatial dimensions of the input. It introduces gaps (zero-padding) between the elements of the input feature map and then applies a kernel, filling the expanded grid with new values based

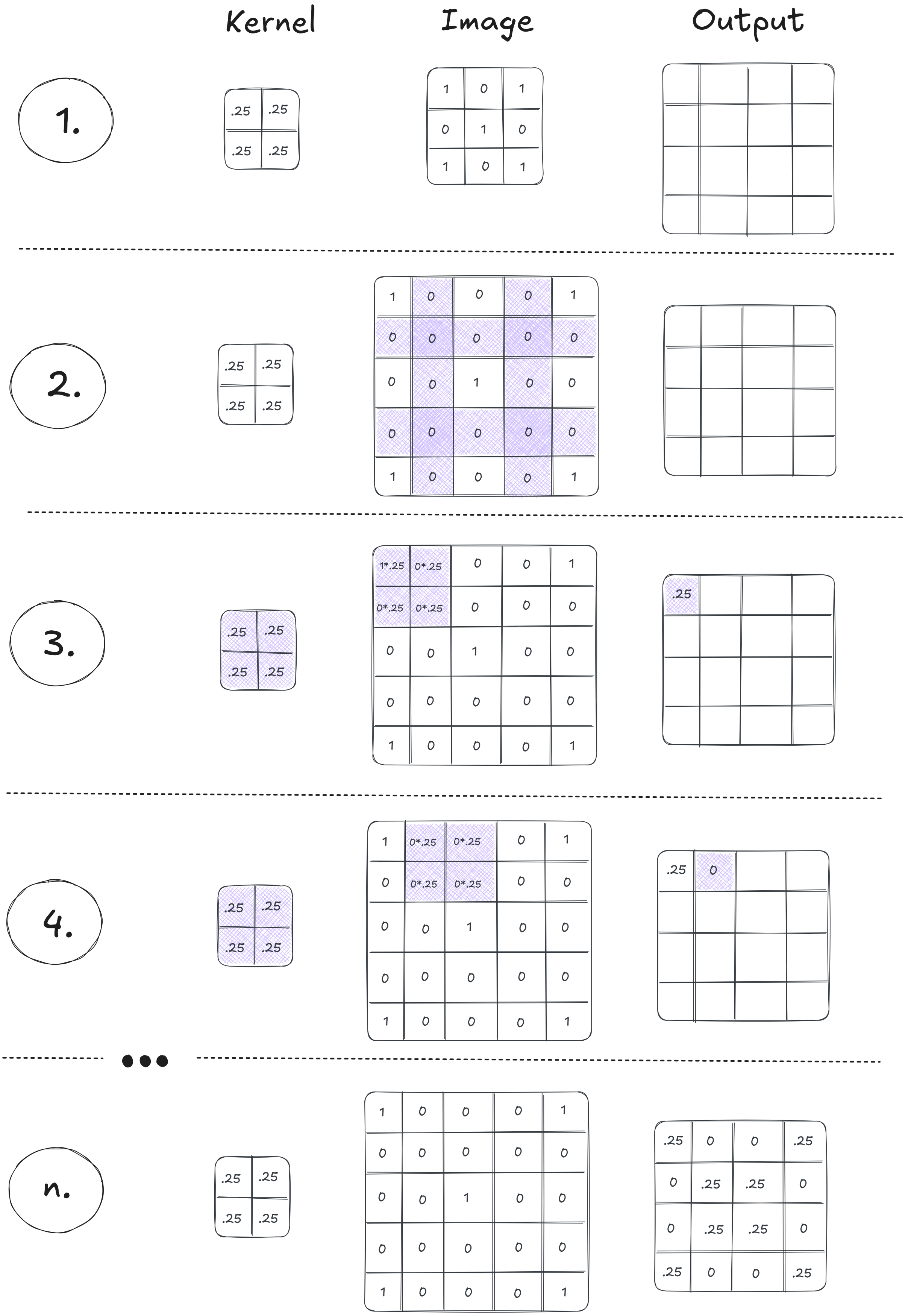


Figure 3: Transposed convolution with a stride of 2.

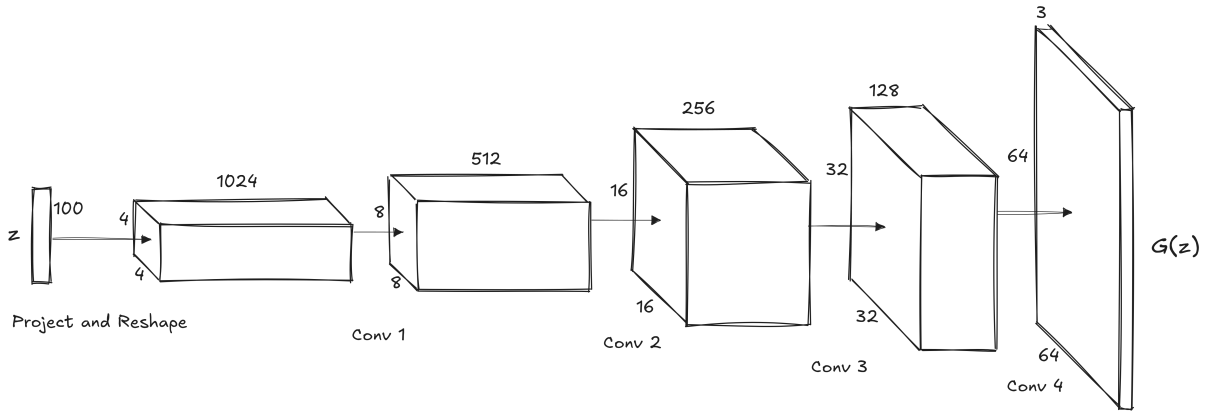


Figure 4: A schematic overview of the DCGAN generator. The numbers refer to the dimensions of the feature maps.

on learnable weights. The output size of a transposed convolution is calculated as:

$$H_{\text{out}} = s(H_{\text{in}} - 1) + k - 2p, \tag{7}$$

and similarly for the width, where s is the stride, k is the kernel size, and p is the padding.

Mathematically, transposed convolution can be represented as a matrix operation. Convolution is a linear transformation, expressed as $\mathbf{y} = W\mathbf{x}$, where W is the convolutional kernel and $\mathbf{x}$ is the input feature vector. Transposed convolution applies the transpose of this matrix, $W^\top$, to map the lower-dimensional $\mathbf{y}$ back to the higher-dimensional space of $\mathbf{x}$:

$$\mathbf{x}' = W^\top \mathbf{y}. \tag{8}$$

However, it is important to note that transposed convolution does not invert convolution in the sense of reconstructing the original input $\mathbf{x}$. Instead, it creates a new learned representation with higher resolution, guided by the structure of $W^\top$. Figure 3 illustrates the process of transposed convolution.

DCGANs extend the Vanilla GAN framework by incorporating convolutional and such transposed convolutional layers. By replacing fully connected layers with convolutional and transposed convolutional layers, DCGANs allow for the generation of high-quality images with improved stability and scalability.

The generator in DCGAN (see Figure 4) takes as input a noise vector $\mathbf{z} \sim \mathcal{N}(0, 1)$ or $\mathbf{z} \sim \mathcal{U}(-1, 1)$, sampled from a latent space. This vector is first transformed into a dense feature map via a fully connected layer and reshaped into a low-resolution tensor, such as $4 \times 4 \times n_{\text{features}}$, where n_{features} represents the depth of the feature map. The generator progressively upsamples this tensor using transposed convolutional layers. Each transposed convolution layer ap-

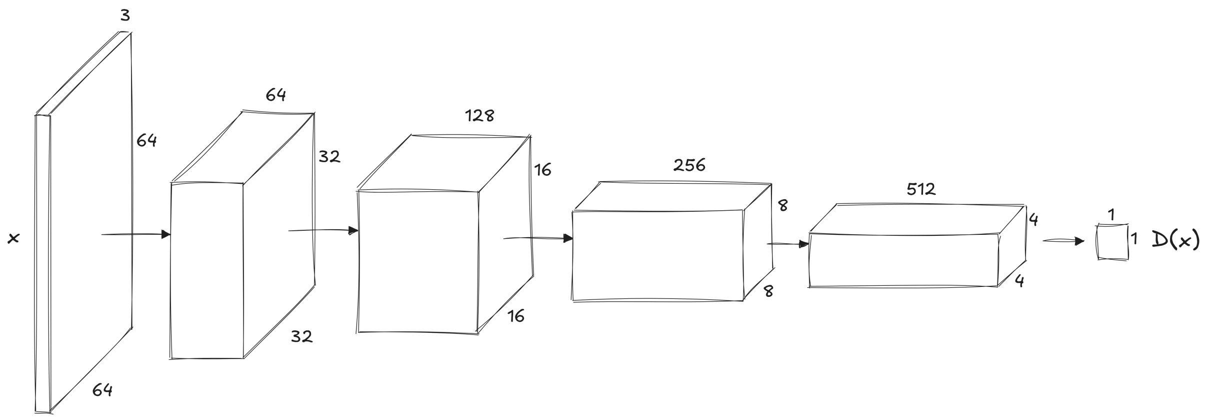


Figure 5: A schematic overview of the DCGAN discriminator. The numbers refer to the dimensions of the feature maps.

plies a learnable filter W to the input feature map $\mathbf{X}$, defined mathematically as:

$$\mathbf{Y}_{i,j} = \sum_{m,n} \mathbf{X}_{m,n} \cdot W_{(i-m),(j-n)}, \tag{9}$$

where $\mathbf{Y}$ represents the upsampled feature map, and (i,j) and (m,n) are spatial indices. These operations increase the spatial resolution at each layer. Batch normalization is defined as:

$$\text{BN}(\mathbf{X}) = \frac{\mathbf{X} - \mu}{\sqrt{\sigma^2 + \epsilon}}, \tag{10}$$

where μ and σ^2 are the mean and variance of the mini-batch. It is applied after each transposed convolution to stabilize training by normalizing feature maps. Rectified Linear Unit (ReLU) activation $\text{ReLU}(x) = \max(0,x)$ is used for all layers except the output layer, which employs the tanh activation $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ to scale outputs to the range $[-1,1]$, aligning with the normalized range of the training images.

The discriminator in DCGAN (see Figure 5) is a standard convolutional neural network that takes an image as input and classifies it as real or fake. The discriminator uses strided convolutions for downsampling, replacing traditional pooling layers, which allows the network to learn its own spatial hierarchy. Leaky Rectified Linear Unit (Leaky ReLU) activation is applied to all layers, defined as:

$$\text{LeakyReLU}(x) = \begin{cases} x, & x \geqslant 0, \\ \alpha x, & x < 0, \end{cases} \tag{11}$$

where α is a small positive slope for negative values (e.g., 0.2). Batch normalization is also used to stabilize feature maps, and the final layer employs a sigmoid activation $\sigma(x) = \frac{1}{1+e^{-x}}$ to output a probability score indicating the likelihood that the input image is real.

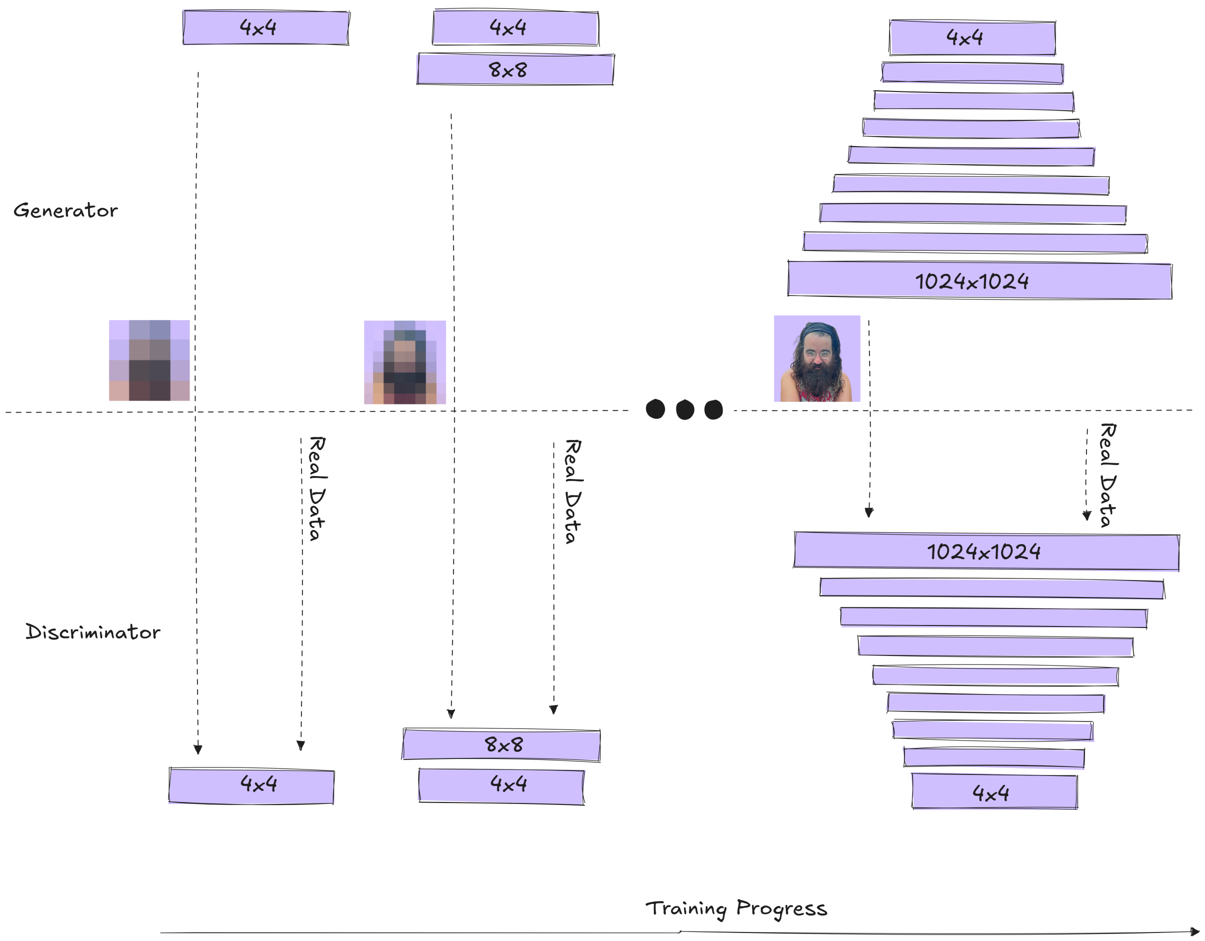


Figure 6: A schematic overview of ProGAN.

The loss functions in DCGAN are derived from the original GAN formulation - the discriminator still aims to maximize its ability to distinguish real images $x \sim p_{\text{data}}$ from generated images $\hat{x} = G(\mathbf{z})$, while the generator minimizes the discriminator's ability to distinguish them.

Note that - similarly to Vanilla GANs - there is not *one* single DCGAN architecture. DCGAN rather refers to the concept of using the mechanisms of convolution and transposed convolution. In this thesis, that concept was used in nearly all of the introduced technical approaches - even if the term DCGAN is not explicitly mentioned.

2.7.2 *Progressive Growing of GANs*

Progressive Growing of GANs (ProGAN) is an advanced generative adversarial network architecture introduced by Karras, Aila, et al. (2017). Conventional GANs - like Goodfellow's Vanilla GAN, and also DCGANs - attempt to train the generator and discriminator at the target resolution from the beginning. Contrary to that, ProGAN starts to learn low resolution images first. Then, during training, it gradually increases the resolution by adding layers to both the generator and discriminator. Therefore, at the beginning of the training, the model learns to capture the overall structures of the data - in later stages, the learned details become finer and finer.

The architecture of ProGAN (see Figure 6) starts with a minimal network that generates images at a low resolution, such as 4×4 pixels. New convolutional layers are then gradually added during training - to both the generator and discriminator. Those newly added layers incrementally increase the resolution of the generated images to 8×8, 16×16, and so on, until the desired resolution is reached.

At each resolution stage, the generator uses transposed convolutions to upsample the feature maps, while the discriminator applies strided convolutions to downsample the input. At each stage in the process, the outputs of the newly added layers are blended with the existing network outputs. In order to avoid abrupt changes (that could lead to mode collapse), this happens over a transition period (i.e., ProGAN uses a *fade-in* mechanism). That transition period can be formalized as follows:

$$\mathbf{X}_{\text{output}} = \alpha \cdot \mathbf{X}_{\text{new}} + (1 - \alpha) \cdot \mathbf{X}_{\text{old}}.$$

Here, $\alpha \in [0, 1]$ is used to weight the influence of new new versus old resolution stages.

Generally, the training in ProGAN follows the standard GAN objective. The training at each resolution proceeds until the generator and discriminator reach a stable adversarial balance. After that balance is achieved, the next resolution stage starts.

2.8 controlling a gan's output

The concepts explained in the previous sections are the foundation of GANs. Using them allows the synthesis of new data that never has been seen before. However, these methods largely focus on improving the quality and diversity of the generated outputs in a non-controlled way. Once the networks are trained, the user has no possibility to steer certain characteristics of the generated content. In their basic form, GANs generate samples from a latent space, where a noise vector $\mathbf{z}$ is randomly sampled from a predefined distribution, such as $\mathcal{N}(0, 1)$ or $\mathcal{U}(-1, 1)$. While this complete randomness ensures that the results are diverse, it also leads to the outputs being completely unpredictable. Therefore, traditional GANs are unsuitable for applications where specific attributes of the output need to be controlled. To address this, mechanisms are required to steer the output of GANs toward desired attributes or features. The most important ones will be explained in the following sections.

2.8.1 *Disentanglement and Latent Space Manipulation*

While GANs are powerful in generating realistic samples, the relationship between the dimensions of z and the attributes of x (e.g., think of characteristics like age or gender when generating face images) is typically entangled, meaning changes in z can affect multiple attributes simultaneously. To address this, various works have focused on disentangling the latent space to enable interpretable and controllable generation (X. Chen et al., 2016; Y. Shen, C. Yang, et al., 2020; Härkönen et al., 2020).

Disentanglement in the context of GANs refers to the alignment of specific latent dimensions or directions with distinct attributes in the generated data. This enables precise control over the output by manipulating the latent space.

For instance, in *InfoGAN* (X. Chen et al., 2016), mutual information is used to encourage interpretable latent representations by maximizing the shared information between a subset of latent variables (c) and the generated data (x). To make this optimization tractable, a mutual information term $I(c; x)$ is approximated using a variational lower bound, which introduces an auxiliary distribution $Q(c|x)$ to estimate the posterior $P(c|x)$. This encourages the generator to encode meaningful and interpretable information about c in the generated output $G(z)$. By maximizing mutual information between c and the generated data, the model ensures that c controls specific, disentangled features in the output. For example, c_1 might influence *age*, while c_2 might control *gender* in a generative model for facial images.

Another approach to disentanglement involves identifying interpretable directions in the latent space *post hoc* using techniques like Principal Component Analysis (PCA) or linear regression. For example, *GANSpace* (Härkönen et al., 2020) applies PCA to the intermediate feature spaces of pre-trained GANs, discovering directions that control interpretable attributes such as age or pose in face images.

Latent space manipulation is further refined in works like *InterfaceGAN* (Y. Shen, C. Yang, et al., 2020), which identifies hyperplanes in the latent space corresponding to binary attributes (e.g., *smiling* vs. *not smiling*). By training linear classifiers on latent vectors paired with attribute labels, directions v_k in the latent space can be determined for specific attributes. Manipulation is achieved by moving the latent vector z along these directions:

$$z' = z + \alpha v_k,$$

where α controls the magnitude of change in the attribute.

The disentanglement and control of latent spaces can be evaluated using metrics like disentanglement scores (Eastwood and Williams, 2018), which measure how well individual latent variables independently control distinct attributes.

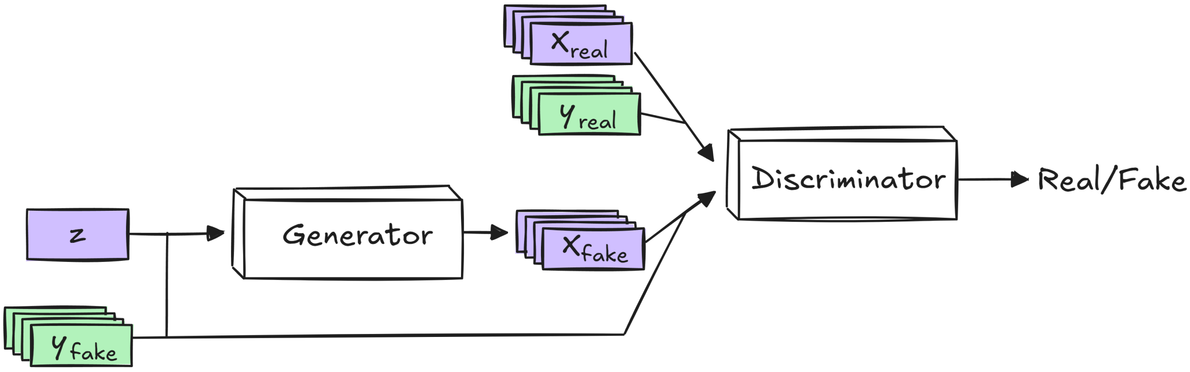


Figure 7: A schematic overview of the Conditional GAN architecture.

2.8.2 *Conditional GAN*

Conditional GANs (cGANs) are an extension of GANs that incorporate additional information (i.e., the *condition*), to guide the generation process. Introduced by Mirza and Osindero (2014), cGANs allows the generator and discriminator to condition their operations on an additional variable y. The primary objective of cGANs is to generate data that aligns with the given condition, enabling targeted and controlled synthesis.

In a cGAN, the generator $G(z|y)$ takes as input both the noise vector $z \sim p_z$ and the condition y, and maps these to the data space. Simultaneously, the discriminator $D(x|y)$ evaluates the likelihood that a given sample x is real, considering the condition y. The cGAN objective function modifies the standard GAN loss to incorporate this conditioning, becoming:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x|y)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z|y)|y))].$$

This formulation enforces that the generated data $G(z|y)$ is not only realistic but also aligned with the given condition y. The condition y can be integrated into the cGAN architecture in several ways. One straightforward method, as proposed in the original cGAN paper (Mirza and Osindero, 2014), is to concatenate y with the input noise z for the generator and with the input data x for the discriminator. For the generator, this means using an augmented input $z' = [z, y]$. Similarly, for the discriminator, $x' = [x, y]$, ensuring that the condition y is explicitly included. A schematic overview of the Conditional GAN architecture is shown in Figure 7.

If y is high-dimensional or categorical, it can be embedded into a lower-dimensional vector space $e_y = \text{Embedding}(y)$ before being concatenated, as demonstrated in text-to-image synthesis models like those by Reed et al. (2016).

Also, more sophisticated methods have been developed to incorporate y into the input space. For example, in the *Auxiliary Classifier*

GAN (ACGAN) proposed by Odena, Olah, and Shlens (2017), the discriminator is equipped with an auxiliary classifier $C(x)$ to predict the condition y. This introduces a classification loss in addition to the standard adversarial loss. Here, the classification loss encourages the generator to produce outputs that are consistent with the given condition y, as verified by the discriminator's auxiliary classifier.

2.8.3 *StyleGAN*

StyleGAN completely rethought the standard GAN architecture. It introduced a style-based generator that separates the latent space from the image synthesis process. This approach enables control over the generation process by disentangling features such as pose, texture, and color (Karras, Laine, and Aila, 2018). Therefore, it is closely related to the concepts already introduced in Section 2.8.1. The key difference to the Vanilla GAN architecture is the handling of the latent vector z. That latent vector is not directly mapped to the output. Instead, StyleGAN transforms z into an intermediate latent vector w using a mapping network:

$$w = f(z), \quad z \sim \mathcal{N}(0, I).$$

The vector w resides in a disentangled latent space $\mathcal{W}$, which allows the manipulation of features (Karras, Laine, and Aila, 2018).

To incorporate w into the generation process, StyleGAN uses *Adaptive Instance Normalization (AdaIN)* (X. Huang and Belongie, 2017). AdaIN normalizes each feature map x_i and re-scales it based on w:

$$\text{AdaIN}(x_i, y) = y_{s,i} \cdot \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i},$$

where: - x_i: The i-th feature map in the generator. - $\mu(x_i)$ and $\sigma(x_i)$: The mean and standard deviation of the feature map x_i, respectively. - $y_{s,i}$ and $y_{b,i}$: Learnable affine transformations derived from w, representing the scale and bias for each feature map.

This ensures that w can steer attributes (e.g., color or texture) without messing with the underlying spatial structure (X. Huang and Belongie, 2017; Karras, Laine, and Aila, 2018). For example, early layers in the generator affect global attributes such as pose and shape, while later layers influence fine details like skin texture or hair strands.

To further improve the disentanglement, StyleGAN introduces *style mixing regularization*. Here, two different latent codes w_1 and w_2 are randomly applied to different layers of the generator during training:

$$w_{\text{mixed}} = [w_1, \dots, w_k, w_2, \dots, w_L],$$

where L is the total number of layers, and k is a randomly chosen split point (Karras, Laine, and Aila, 2018). Doing so prevents the generator

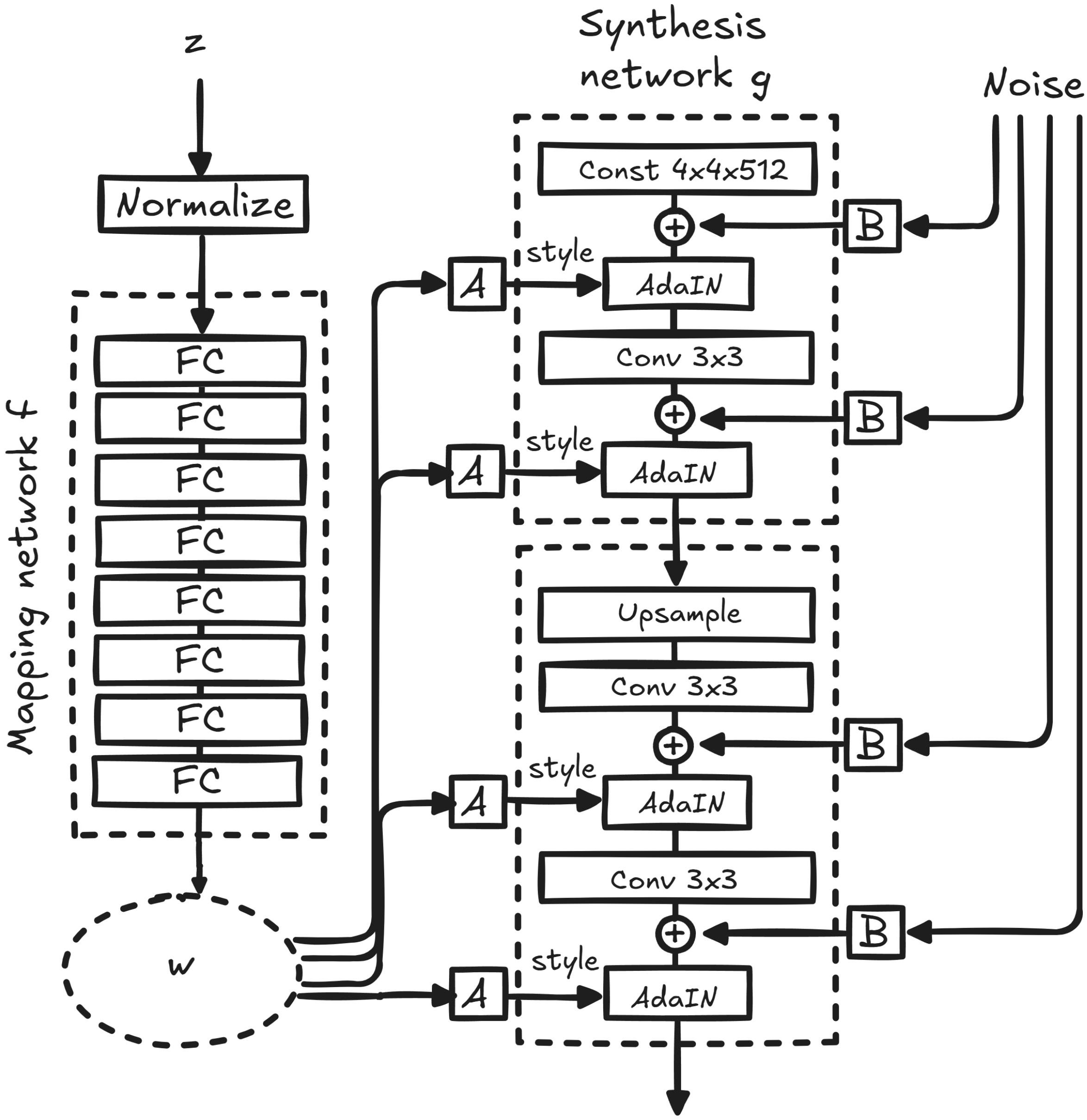


Figure 8: A schematic overview of StyleGAN. Image content adapted from (Karras, Laine, and Aila, 2018).

from relying on correlations between styles at different layers - which in turn makes sure that each layer independently contributes to the image.

Another mechanism that is incorporated by StyleGAN is the so-called *noise injection*. Noise Injection introduces random variations to the feature maps at each layer by adding Gaussian noise N to the feature maps. This stochastic element enables the generation of unique details, such as individual freckles, hair strands, or background patterns, while the style vector w determines the overall appearance.

A schematic overview of the StyleGAN architecture is shown in Figure 8.

2.8.3.1 *StyleGAN2*

StyleGAN achieved very good results in image synthesis. However, there were also limitations here - for example, spatial misalignments caused by AdaIN led to artifacts in the generated images. These problems were addressed in StyleGAN2. StyleGAN2 improved the original architecture by replacing AdaIN with *demodulation*, a mechanism that normalizes convolutional weights instead of feature maps. By ensuring consistent feature sizes during convolution, demodulation prevents distortions (Karras, Laine, Aittala, et al., 2020). The modulation mechanism can be formulated as follows:

$$\hat{w}_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_k w_{k,j}^2 + \epsilon}},$$

where $w_{i,j}$ represents the weights applied to the j-th feature map. This change resolved the artifact issues caused by AdaIN and ensured that features at different spatial resolutions remained aligned (Karras, Laine, Aittala, et al., 2020).

Additionally, StyleGAN2 introduced *path length regularization*, a mechanism to ensure smooth transitions in the latent space. It does so by penalizing abrupt changes in the output when small changes are made to w. Therefore, the generator is encouraged to maintain consistent mappings between the latent space and the data space (Karras, Laine, Aittala, et al., 2020). Also, additional noise was integrated into the synthesis process - instead of injecting noise only directly into the feature maps, StyleGAN2 additionally incorporates noise in a way that enhances fine details without disrupting the overall structure, which results in higher-quality images (Karras, Laine, Aittala, et al., 2020).

2.8.3.2 *StyleGAN-ADA*

When training GANs, a big challenge is the dependency on large data sets. With small datasets, the discriminator can become over-adapted as it learns too easily to distinguish real from generated samples.

StyleGAN-ADA (Adaptive Discriminator Augmentation) addresses this problem by dynamically augmenting the inputs to the discriminator during training (Karras, Aittala, Hellsten, et al., 2020).

Enhancements, such as flips, rotations and color variations, are applied to both real and generated images before they are passed to the discriminator. These augmentations ensure that the discriminator focuses on meaningful features rather than adapting too closely to the training patterns.

By dynamically adjusting the augmentation probability, StyleGAN-ADA enables robust training even with small datasets (Karras, Aittala, Hellsten, et al., 2020).

2.8.3.3 *StyleGAN3*

The focus of StyleGAN3, the newest generation of StyleGAN, is to resolve aliasing artifacts and improve geometric consistency in the generated images. Those aliasing artifacts occur when high-frequency components are improperly sampled, which in turn leads to distortions in features like edges or textures. To prevent this, StyleGAN3 introduced the so-called *alias-free* architecture. This means that it uses band-limited filters and modified convolutional layers. Through that technique, feature alignment is maintained across resolutions (Karras, Aittala, Laine, et al., 2021).

Additionally, StyleGAN3 puts focus on *equivariance*. Here, equivariance refers to the desire that transformations applied in the input space (e.g., a rotation) should be reflected in the output image without introducing distortions. Through these mechanisms, StyleGAN3 is particularly suited for tasks requiring geometric precision.

2.9 STYLE TRANSFER NETWORKS

Until now, we have focused on techniques that enable GANs to generate entirely new data - such as realistic images - by sampling from a latent space. These methods aim to generate *new* data, but there are many situations where the task at hand is not to generate something new, but rather to modify *existing* data. In particular, we often want to change the style or visual appearance of an image while preserving its underlying structure or content. This is where *Style Transfer Networks* come into play.

Style transfer addresses the problem of altering an image's style in a way that it resembles the style of another image - or the style of a whole set of images. For example, consider taking a photograph and rendering it in the painting style of a specific artist, like van Gogh or Monet. The content of the photograph - the layout, shapes, and objects - remains intact, but the appearance (e.g., colors, textures or brushstrokes) is transformed to match the desired style.

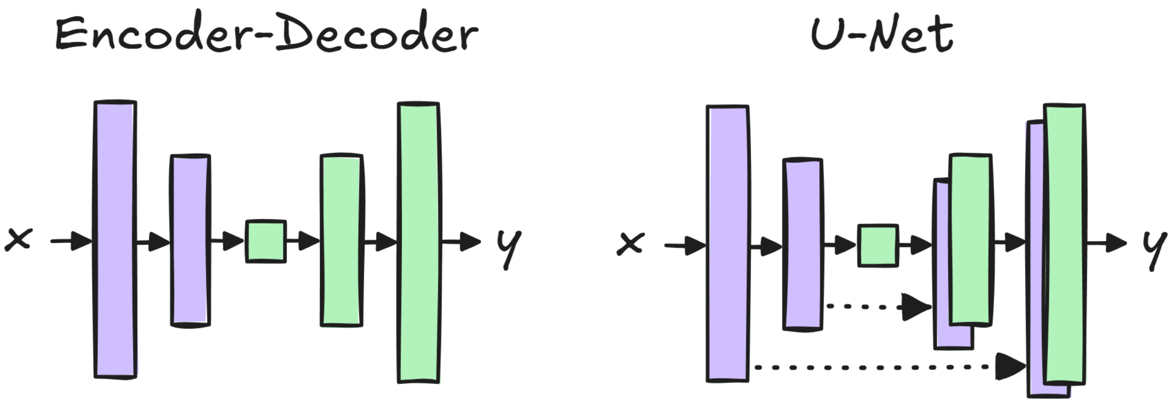


Figure 9: A schematic, simplified overview of an encoder-decoder versus U-Net.

Traditional style transfer techniques rely on optimization-based approaches that are computationally expensive, requiring multiple iterations to blend content and style (Gatys, 2015; C. Li and Wand, 2016; Luan et al., 2017). These methods, while effective, are unsuitable for real-time or large-scale applications. Style Transfer Networks, particularly those based on GANs, ease this process by learning to apply style transformations in a single forward pass, making the process both faster and more flexible.

In the following sections, we will introduce some specific GAN-based approaches to Style Transfer that turned out to be the most influential in that particular research field.

2.9.1 *Pix2Pix*

One of those approaches was presented by Isola et al. (2017). Their framework, which they called *Pix2Pix*, is based on cGANs. In contrast to the cGANs that we explained previously, where the generation process is conditioned on categorical labels or low-dimensional auxiliary information, Pix2Pix conditions the synthesis process on input images (see Figure 10). This formulation allows for the learning of complex mappings between two image domains, such as converting grayscale images to color, translating sketches to photorealistic renderings, or generating realistic images from segmentation maps.

The generator in Pix2Pix employs a *U-Net* architecture (Ronneberger, P. Fischer, and Brox, 2015), an architecture similar to an encoder-decoder structure, but with skip connections that link each layer in the encoder to its corresponding layer in the decoder (see Figure 9). The encoder progressively down-samples the input image to extract hierarchical features, while the decoder up-samples these features to reconstruct the output image. The skip connections ensure that high-resolution details from the input image are preserved and

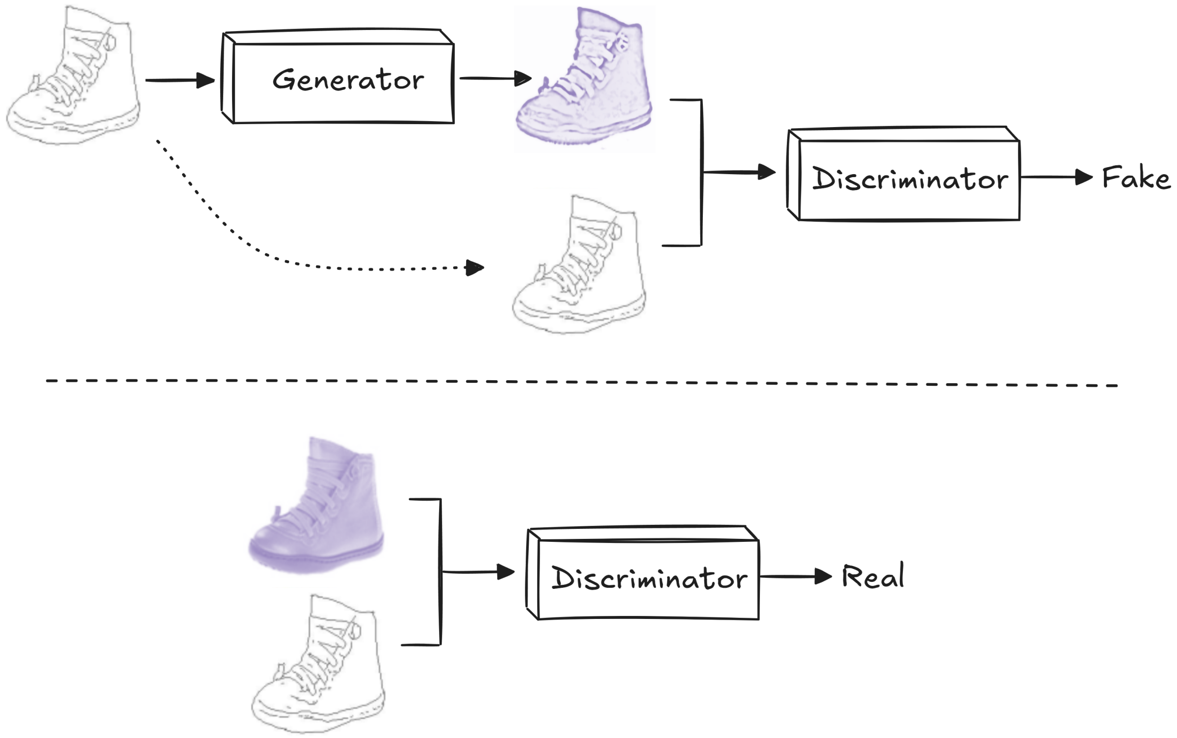


Figure 10: A schematic illustration of the Pix2Pix architecture's generator (top) and discriminator (bottom) training process (see (Isola et al., 2017)).

directly incorporated into the output, reducing the loss of spatial information inherent in the down-sampling process.

The discriminator in Pix2Pix, referred to as PatchGAN, evaluates the realism of a generated image not globally but at the scale of local patches (e.g., 70×70 pixels). This approach allows the discriminator to focus on fine-grained details, such as textures and local patterns, while maintaining computational efficiency. The PatchGAN discriminator predicts whether each patch in the image is real or generated. By doing so, it enforces local consistency in the generated output.

The training objective of Pix2Pix combines two loss functions: the standard adversarial loss and a pixel-wise reconstruction loss. Here, the reconstruction loss, formulated as an L_1 loss, ensures that the generated image $G(x, z)$ closely matches the ground truth image y:

$$\mathcal{L}_{L_1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]$$

The total objective function combines these two terms with a weighting factor λ, balancing realism and fidelity:

$$G^* = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda \mathcal{L}_{L_1}(G)$$

During training, the discriminator D is still optimized to distinguish between real image pairs (x, y) and generated pairs $(x, G(x, z))$,

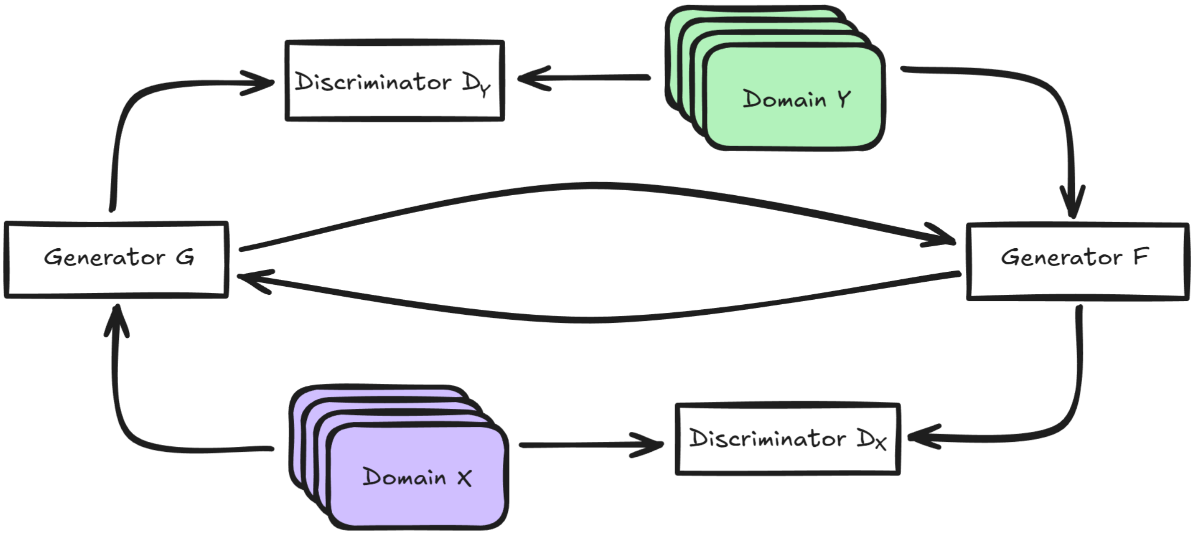


Figure 11: A very simplified schematic overview of the CycleGAN architecture.

while the generator G is optimized to minimize the combined adversarial and reconstruction losses. Importantly, Pix2Pix is limited to tasks where paired input-output image datasets are available, as it relies on the explicit alignment between the domains to learn the mapping.

2.9.2 *CycleGAN*

In many real-world scenarios, obtaining paired datasets for supervised image-to-image translation tasks is either impractical or impossible. For example, consider the task of transforming images of horses to images of zebras - *real* data pairs are simply not available here, as there has never been a real horse magically turning into a zebra. As such, a method is needed that can handle the task of *unpaired* Image-to-Image translation. This means, that instead of learning the relation between two styles (or *domains*) by looking at specific pairs of data, such a method should learn that relation by looking at whole datasets, where each dataset contains only data of one domain. CycleGAN, introduced by Zhu et al. (2017), does this by introducing the concept of *cycle consistency* that ensures structural coherence between original and translated images.

CycleGAN consists of two generators, $G : X \to Y$ and $F : Y \to X$, and two discriminators, D_X and D_Y. The generator G is trained to map images from domain X to domain Y such that the generated images $G(x)$ are indistinguishable from real images in domain Y, as judged by the discriminator D_Y. Similarly, F is trained to map images from domain Y to domain X, with the discriminator D_X ensuring that $F(y)$ resembles real images in domain X. The adversarial loss for G and D_Y

ensures that $G(x)$ is indistinguishable from real Y, and is, similarly to the original GAN adversarial loss, expressed as:

$$\begin{aligned}\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] \\ + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))].\end{aligned}$$

Analogously, the adversarial loss for F and D_X is defined as:

$$\begin{aligned}\mathcal{L}_{GAN}(F, D_X, Y, X) = \mathbb{E}_{x \sim p_{data}(x)}[\log D_X(x)] \\ + \mathbb{E}_{y \sim p_{data}(y)}[\log(1 - D_X(F(y)))].\end{aligned}$$

Since paired data is not available, CycleGAN introduces the cycle consistency loss to ensure that the learned mappings are meaningful and reversible. The idea is that if an image $x \in X$ is transformed to domain Y using $G(x)$ and then mapped back to domain X using $F(G(x))$, the resulting image should closely match the original x. This is expressed as:

$$\begin{aligned}\mathcal{L}_{cycle}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] \\ + \mathbb{E}_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1].\end{aligned}$$

This loss enforces that $F(G(x)) \approx x$ and $G(F(y)) \approx y$. By minimizing this loss, the model avoids generating arbitrary mappings that would discard critical information about the input images.

To further stabilize training and ensure that the transformations do not alter images unnecessarily, the identity loss is introduced. The identity loss ensures that if an image already belongs to the target domain, applying the generator should leave it unchanged. For example, if $x \in X$ is input to F (which maps $Y \to X$), the output $F(x)$ should be close to x. This loss is defined as:

$$\mathcal{L}_{identity}(G, F) = \mathbb{E}_{y \sim p_{data}(y)}[\|G(y) - y\|_1] + \mathbb{E}_{x \sim p_{data}(x)}[\|F(x) - x\|_1].$$

The total objective of CycleGAN is a weighted combination of the adversarial loss, cycle consistency loss, and identity loss. As such, the full objective is:

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) \\ + \lambda_{cycle}\mathcal{L}_{cycle}(G, F) + \lambda_{identity}\mathcal{L}_{identity}(G, F),\end{aligned}$$

where λ_{cycle} and $\lambda_{identity}$ control the relative importance of the cycle consistency and identity losses. During training, the generators G and F are optimized to minimize the combined loss, while the discriminators D_X and D_Y are trained to maximize their ability to distinguish real images from generated ones. Figure 11 shows an overview of the CycleGAN architecture.

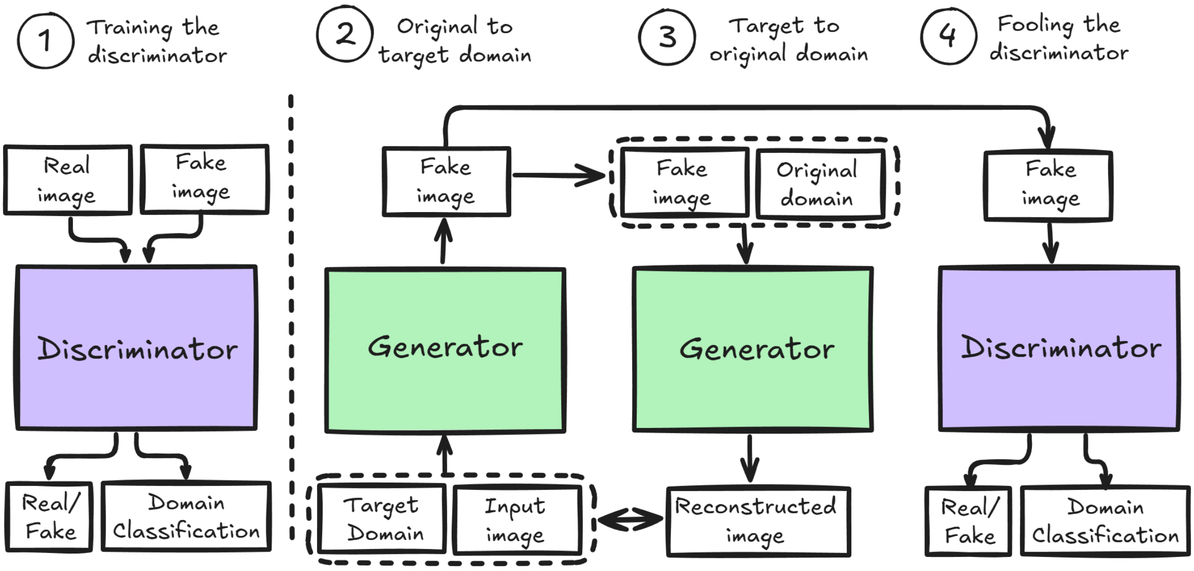


Figure 12: A schematic overview of the StarGAN training process. (1) Additionally to classyfing real vs fake, the discriminator learns a domain classification. (2) The generator generates an image based on an input image and a target domain label. (3) That generated image, together with the original domain (i.e., the domain that the input image stemmed from) gets fed into the generator to produce a reconstructed version of the input. (4) Additionally to that reconstruction loss, the generator tries to generate images that the discriminator thinks are real and belong to the desired target class. Image content adapted from (Y. Choi et al., 2018).

2.9.3 *StarGAN*

Although CycleGAN has demonstrated its applicability in various scenarios, it is inherently limited to tasks involving two domains, requiring separate models for every pair of domains. For tasks involving multiple domains, the approach becomes computationally inefficient and challenging to scale. Therefore, *StarGAN* was introduced by Y. Choi et al. (2018) as a framework for unified and scalable multi-domain image-to-image translation. StarGAN extends the CycleGAN architecture by allowing translation across multiple domains with a single model, rather than training separate models for each domain pair. StarGAN achieves this by conditioning the generator and discriminator on domain labels, enabling the model to generalize across all domains. The key idea is to incorporate domain-specific information into the translation process while ensuring that the transformations are consistent and reversible across domains.

StarGAN operates with a single generator G, which maps an input image x from domain d_x to an output image in a target domain d_y, and a single discriminator D, which classifies whether an image is real or fake and predicts its domain label. By conditioning both the generator and discriminator on domain labels, StarGAN generalizes the translation process to support multiple domains efficiently. For example, $G(x, d_y)$ generates an image from domain d_x that resembles images in domain d_y.

The adversarial loss ensures that the generator produces realistic images indistinguishable from real images in the target domain. The discriminator D classifies images as real or fake while also predicting their domain labels. The adversarial loss is defined as:

$$\begin{aligned}\mathcal{L}_{adv} = {} & \mathbb{E}_{x\sim p_{data}(x)}[\log D_{real}(x)] \\ & + \mathbb{E}_{x\sim p_{data}(x), d_y\sim p(d_y)}[\log(1 - D_{real}(G(x, d_y)))].\end{aligned}$$

Here, $D_{real}(x)$ is the discriminator's probability of x being real.

The discriminator also predicts the domain labels of real and generated images. This is captured by the domain classification loss, which ensures that: (i) the discriminator correctly predicts the domain labels of real images, and (ii) the generator produces images whose domain labels match the target domain.

The domain classification loss for real images is:

$$\mathcal{L}_{cls}^{real} = \mathbb{E}_{x\sim p_{data}(x)}[-\log D_{cls}(d_x|x)],$$

where $D_{cls}(d_x|x)$ is the discriminator's predicted probability of domain d_x given real image x.

For generated images, the domain classification loss is:

$$\mathcal{L}_{cls}^{fake} = \mathbb{E}_{x\sim p_{data}(x), d_y\sim p(d_y)}[-\log D_{cls}(d_y|G(x, d_y))].$$

This term ensures that the generator $G(x, d_y)$ generates images classified by D as belonging to the target domain d_y.

To ensure the transformations are reversible and retain content, StarGAN employs a reconstruction loss similar to CycleGAN's cycle consistency loss. If an image is transformed from its original domain d_x to a target domain d_y and back, the reconstructed image should resemble the original. This is expressed as:

$$\mathcal{L}_{reconstruction} = \mathbb{E}_{x\sim p_{data}(x), d_y\sim p(d_y)}[\|G(G(x, d_y), d_x) - x\|_1].$$

The full objective combines the adversarial, domain classification, and reconstruction losses. The overall loss is:

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda_{cls}^{real}\mathcal{L}_{cls}^{real} + \lambda_{cls}^{fake}\mathcal{L}_{cls}^{fake} + \lambda_{reconstruction}\mathcal{L}_{reconstruction},$$

where λ_{cls}^{real}, λ_{cls}^{fake}, and $\lambda_{reconstruction}$ are weights balancing the importance of the respective loss components.

Figure 12 depicts the StarGAN training process.

2.9.4 *StyleGAN Projection*

In Section 2.8.3, StyleGAN was introduced as a State-of-the-Art method for generating new image data. However, StyleGAN also includes a projection mechanism that allows for style conversion. Conceptually, that mechanism differs substantially from frameworks like Pix2Pix, CycleGAN, and StarGAN. While those models focus on learning direct mappings between domains, either with paired data (Pix2Pix) or unpaired data (CycleGAN, StarGAN), StyleGAN projection utilizes a pre-trained StyleGAN generator - and embeds images into its latent space for manipulation. Unlike those other domain-to-domain translation frameworks, which are designed to transform images directly from one style or domain to another, StyleGAN projection works by finding an image's latent representation within the generator's latent space, allowing for fine-grained style modifications. The method for using StyleGAN projection for image editing, as explained below, was introduced by Abdal, Qin, and Wonka (2019). This process involves finding a latent code $\mathbf{w}$ such that the generated image $G(\mathbf{w})$ closely reconstructs the input image I (i.e., the image that should be altered). Projection (i.e., finding the latent code of the input image) involves optimizing the latent code $\mathbf{w}$ such that:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}(I, G(\mathbf{w})),$$

where $\mathcal{L}$ is a loss function designed to measure the similarity between I and $G(\mathbf{w})$.

The loss function typically consists of two components. The first component, the so-called *Pixel-wise Loss*, measures the direct pixel-by-pixel difference between the input image and the reconstructed image:

$$\mathcal{L}_{\text{pixel}} = \|I - G(\mathbf{w})\|_2.$$

The second component, the *Perceptual Loss*, accounts for perceptual differences that the pixel-wise loss may overlook. Features are extracted from a pre-trained network ϕ (e.g., VGG (Simonyan and Zisserman, 2014)) and compared between the input and the reconstruction:

$$\mathcal{L}_{\text{perceptual}} = \|\phi(I) - \phi(G(\mathbf{w}))\|_2.$$

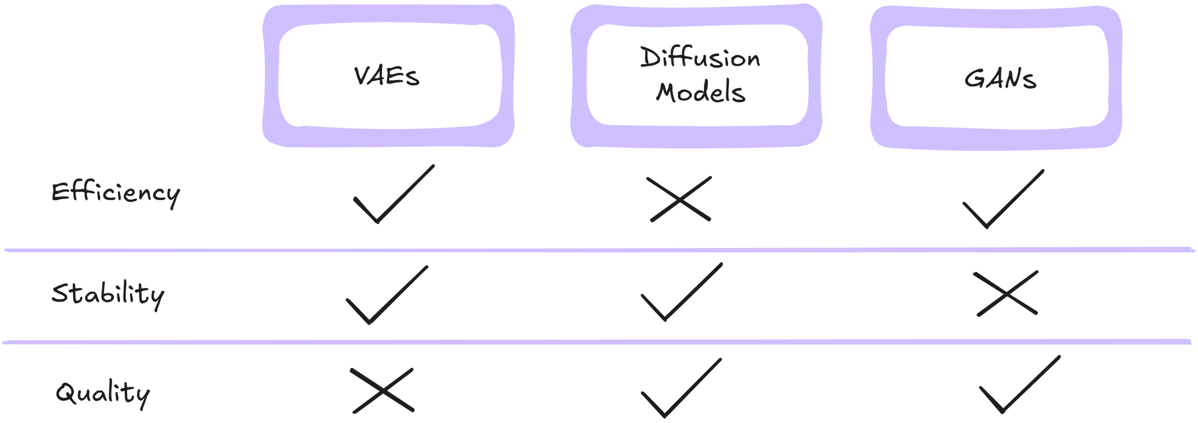
The combined loss function is:

$$\mathcal{L} = \lambda_{\text{pixel}}\mathcal{L}_{\text{pixel}} + \lambda_{\text{perceptual}}\mathcal{L}_{\text{perceptual}},$$

where λ_{pixel} and $\lambda_{\text{perceptual}}$ are weighting factors that balance the contributions of each loss term .

The optimization process minimizes $\mathcal{L}$ iteratively, often using gradient descent-based methods like Adam. The result is a latent code $\mathbf{w}^*$ that closely reconstructs the input image.

Once the input image is projected into the latent space, style conversion becomes a matter of manipulating the latent code $\mathbf{w}^*$. StyleGAN's disentangled latent space allows specific edits to attributes such as texture, color, or structure while preserving the underlying



	VAEs	Diffusion Models	GANs
Efficiency	✓	✗	✓
Stability	✓	✓	✗
Quality	✗	✓	✓

Figure 13: Advantages and disadvantages of VAEs, Diffusion Models and GANs.

content of the image. The edited latent code $\mathbf{w}'$ produces the altered image:

$$I' = G(\mathbf{w}').$$

2.10 CHOOSING THE RIGHT MODEL

In the preceding sections, we have layed the theoretical foundation for understanding the state-of-the-art in current GAN research - and thus, the remainder of this thesis. However, if it comes to choosing the right model for a specific task at hand, many factors have to be considered. Those considerations do not even start with the decision of a specific GAN model - first and foremost, it isn't even trivial to choose a GAN model *at all*. The main "competitiors" of GANs are arguably *Variational Autoencoders* (VAE) (Kingma, 2013) and *Diffusion Models* (Ho, A. Jain, and Abbeel, 2020). Although explaining both of them in detail would go beyond the scope of this work, we do not want to leave the decision between GANs, VAEs and Diffusion Models undiscussed. Very generally speaking, there are three main requirements to a generative model:

- It should yield results of high quality (*Quality*).
- When trained, results should be produced in a reasonable time (*Efficiency*).
- The model should be trainable in a stable way (*Stability*).

Without going into too much detail, we would like to emphasize that none of the three popular generative approaches sufficiently fulfills all of these three requirements at the same time (see Figure 13). While Diffusion Models can be trained in a very stable way and produce results that are of very high quality, it takes quite a long time to produce those outputs. VAEs - once the quite stable training has completed - are quite fast, but the outcomes tend to be blurry (in the image domain) or noisy (in the audio domain). GANs, on the other hand, produce results of high quality, and they (again, once trained) do that in a very fast way. However, as discussed in the preceding sections, they have many difficulties (e.g., mode collapse) that make the training rather unstable. Although mechanisms exist that counter those drawbacks to a certain degree, they still are not completely resolved.

As such, for a specific task, a decision must be made as to which requirement is most likely to be neglected: GANs are mostly appropriate for tasks where data of *high quality* is to be produced *fast*.

After deciding for a category of models - in this thesis, we focus exclusively on GANs - the specific architecture has to be chosen. Here, again, it depends on the specific use-case. For the main parts of this work's implementations, we used fairly "simple" architectures as GAN backbone models. More sophisticated architectures (e.g., StyleGANs (Karras, Laine, and Aila, 2018) or Progressive Growing GANs (Karras, Aila, et al., 2017)) - due to their complexity - require longer training and (possibly) more hyperparameter tuning. As such, it is rather difficult to train them appropriately with limited resources - and for this thesis, most models were trained with such limited resources. For instance, we mostly used consumer-level GPUs (e.g., Nvidia GTX 1060 Ti, Nvidia RTX 2070, Nvidia Titan X, etc.) for training. Also, the more complex the architecture, the more difficult it becomes to correctly assess which parts of the results can be attributed to the newly introduced concepts, and which parts result from specific model *configuration* choices. Therefore, it should be noted, that the results of this work mostly are to be seen as proof of concept of the introduced ideas. In a productive environment - with better hardware and more resources available - all the technical contributions of this work could easily be applied to state-of-the-art architectures to yield even better quality.

Part II

ROBUSTNESS

The key requirement for everything that we want to use AI for is trivial: The underlying AI model should work in a robust way. While this statement might be obvious, *achieving* that goal is difficult. There are a lot of factors that contribute to the robustness of DL-based AI approaches, be it model architectures, hyperparameters, hardware resources, and many others. However, when we talk specifically about AI systems that use mechanisms of Deep Learning, one of the most critical factors is the available training data. When we want to train such a model for a specific use-case, we often need a huge amount of data - and almost as often, we don't have that data. A common approach to dealing with that is *Data Augmentation*. That term refers to all kinds of techniques that augment existing datasets with new data in order to get a bigger training dataset that makes the training more robust and cover a larger portion of the problem domain distribution. There are a lot of established techniques for data augmentation. Many of them heavily contribute to making the training process more robust. However, as they often are based on simple data transformations, they lack in actually expanding the portion of the training *domain* that is covered by the training *dataset*, i.e., they do not generate training data that holds *new* information for the task at hand. Here, generative AI can help - and in the following chapter we will show *how*. Therefore, we will first introduce the basic concepts that are commonly used to augment data. Here, we will confine ourselves to image and audio data, as our research mainly revolved around those two domains. Then, we will propose two approaches on how GANs can be used to artificially increase datasets and evaluate if they are able to overcome the limitations of traditional data augmentation techniques.

3

TRADITIONAL DATA AUGMENTATION

3.1 IMAGE DOMAIN

In the image domain, traditional data augmentation mostly refers to simple transformation algorithms. Note that those algorithms primarily aim to enhance a dataset in a way that it enables the training of more *robust* models. Robustness, in this context, means that a model is able to maintain its performance even if the real-world data (i.e., the data that the model sees during inference) deviates from the training data. Deviations might be different orientations of objects, different illuminations, or other factors that slightly change in comparison to the information contained in the training data. In the following, we will go through the most popular techniques. Of course, there are many more - as such, for a more comprehensive overview on data augmentation techniques for the image domain, we refer to the survey by Shorten and Khoshgoftaar (2019).

3.1.1 *Flipping*

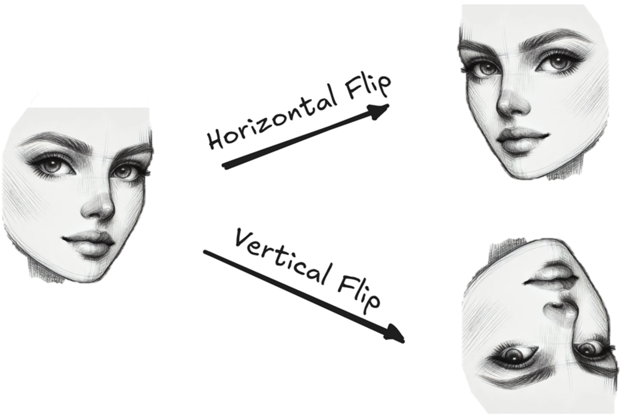


Figure 14: Horizontal and vertical flipping.

Flipping an image simply refers to *mirroring* it. It can be performed either horizontally or vertically. Horizontal flipping comes in handy for tasks like object segmentation or recognition. For example, imagine an image of an urban scenery - cars, houses, trees, etc. - if such a scenery is flipped horizontally, it still might *make sense*. As such, adding those images to the dataset can extend the robustness of models trained on respective datasets. Vertical flipping, on the other hand, is used more rarely. If you think of the urban scenery again, you can probably imagine why vertical flipping might not produce the most plausible images. In Figure 14, the process of flipping is depicted.

3.1.2 *Rotation*

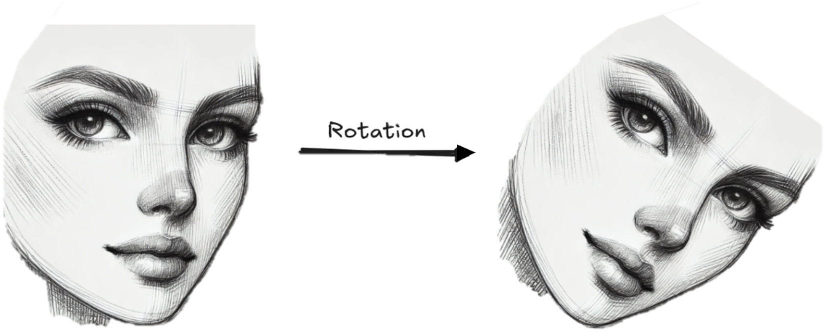


Figure 15: Rotation.

Often, objects in a computer vision task can have multiple orientations. For example, when analyzing an image of a human face with respect to certain traits like emotions, it should be irrelevant for a trained model whether the head is straigt or slightly tilted. In order to get robust against such rotations, we can simply add rotated versions of the existing training data to the training dataset (see Figure 15).

3.1.3 *Cropping*

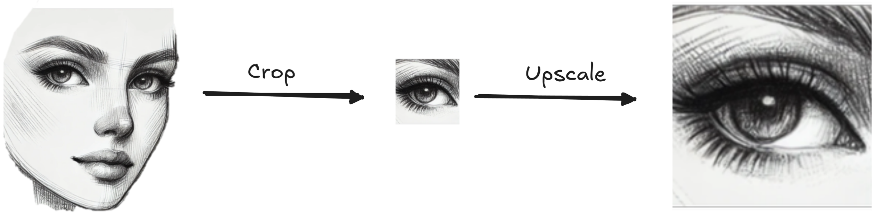


Figure 16: Cropping with subsequent upscaling.

A common technique for data augmentation is *Cropping*. Here, a sub-image is extracted from the original image. That sub-image is usually drawn from the exact center of the original image (i.e., center cropping), or from a random position (i.e., random cropping). Every pixel that is not part of that sub-image gets "thrown away". Often, after cropping, the image gets re-scaled (i.e., upscaled) to the original size (see Figure 16). It might appear counter-intuitive that dropping pixels is actually a means of data *augmentation*, as deletion and augmentation usually reflect contradictory concepts. However, when adding cropped data to a dataset, a machine learning model, for that instances, is forced to extract all relevant data from just that tiny excerpt. As such, the model becomes more robust. It learns to cope with data that does not contain *all* information - as it is often the case for real-world data.

3.1.4 *Color Jittering*

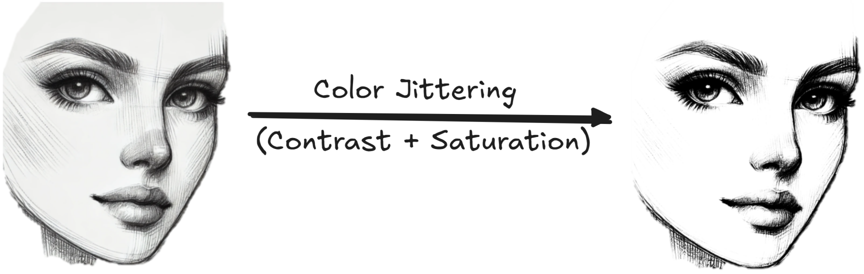


Figure 17: Color jittering. Specifically, contrast was increased and saturation decreased.

One important consideration in most computer vision tasks is illumination. Although not changing the overall spatial structure, illumination has a big impact on the pixel value distribution of an image. In order to become robust against those deviations, a training dataset should be augmented with images where various illumination settings are simulated. Here, techniques of *Color Jittering* come into play. Color jittering refers to a whole category of algorithms that all alter certain color characteristics of an image. For example, color jiterring can include changing the brightness, contrast, hue or saturation of an image. Augmenting images with these techniques comes close to having multiple illumination setups in a dataset (see Figure 17).

3.1.5 *Noise Injection*

In real-world applications, data might be imperfect. As such, data augmentation also strives for introducing imperfect data to the training dataset, so that the model learns to cope with flaws. A simple but efficient method is to add noise to an image. This can be done through various heuristics. Probably the most popular one is to simply add gaussian noise (see Figure 18). However, other techniques

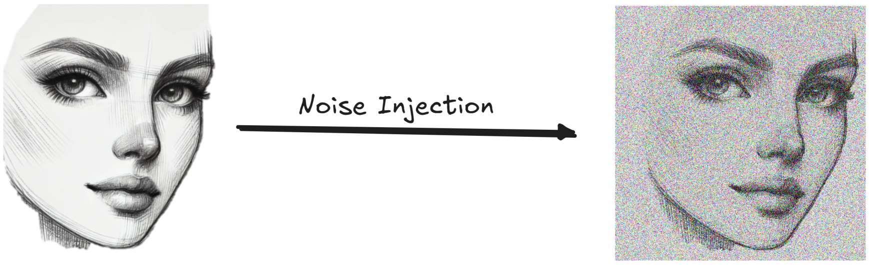


Figure 18: Noise injection with Gaussian Noise.

exist that are frequently used to simulate very specific imperfections. For example, *Salt and Pepper* can be used to mimic defects in camera sensors. In order to do so, some pixels are randomly replaced by either white (=*salt*) or black (= *pepper*) pixels.

3.2 AUDIO DOMAIN

When training machine learning models for the audio domain, data augmentation is also frequently used. Similarly to the image domain, traditional data augmentation here mostly makes use of rather simple techniques, which we will explain in the following. Note that the following compilation is not complete - there are a lots of different techniques for audio data augmentation which are partially very complex, and it would be impossible to cover all of them here. For a more complete overview on existing techniques, we would like to refer to Abayomi-Alli et al. (2022). Also, audio data augmentation techniques have to be applied with care - not all methods can be used for *every* use-case. Similarly to the image domain, where, for instance, vertically flipping images might produce data that is impossible to observe in reality, applying the wrong audio data augmentation techniques can also result in samples that are not contributing to the task.

3.2.1 *Slice and Shuffle*

Slicing is a technique that addresses similar problems as Cropping does for the image domain. Basically, it is the same approach of cutting out a subset of a sample - just not in the 2D space (as for images), but in the 1D audio space. Determining the size of the single slices, however, can become less trivial: the slices can be created with a fixed size, but they can also be chosen in a more sophisticated way. For example, when working with speech, using a fixed size for cropping would completely destroy the semantics of the spoken content. As such, here, it might make sense to use single words or utterances as slices. Often, in a subsequent step, those slices get shuffled - as such, the audio sequence is rearranged. By doing so, temporal variations in the data sample are introduced, which again contributes to the robustness of trained models.

3.2.2 *Pitch Shift*

Often, in the audio domain, we deal with problems that are invariant to pitch. A simple example would be speech recognition - a specific word, no matter if spoken by someone with a very high voice or very low voice, should always be recognized as the same word. To introduce pitch invariance into a dataset, increasing or decreasing the shift of data points is a popular technique for data augmentation. The pitch in an audio signal is reflected by the frequency. Just altering the frequency, however, would not only change the pitch, but also the length of the audio signal. As such, altering a pitch typically involves a resampling and/or time stretching step in order to maintain the sample length. However, there are also scenarios where augmenting the data with varying pitch might *not* make sense. For example, think of the training of a classifier that should learn to differentiate between male and female voices. Here, the pitch is inherently important for the classifier's decision, as female voices (on average) have a higher pitch than male voices (Oliveira, Gama, and Magalhães, 2021).

3.2.3 *Phase Inversion*

Audio signals can be thought of as a combination of multiple sinusoid signals, which add up to new waveforms. What we actually hear is the oscillation of those waveforms, as that oscillations are directly carried over to the eardrum. The sinusoids that form the sound are characterized not only by their amplitude and frequency, but also by the phase. An inversed phase, i.e., mirroring the sample points on the x axis, does not change the sound for us humans, as our eardrum only processes the absolute differences in amplitude changes. Contrary, for a machine learning model that relies on raw data, that makes a difference, as that model basically just "sees" the raw values. As such, augmenting datasets with phase inversion is commonly used.

3.2.4 *Reverb*

Reverb is a more complex technique for audio data augmentation. The reasoning behind using it is that in the real world, depending on the specific physical environment, sound is reflected in various manners. However, mostly we want to be invariant against the environment of the sound source. As such, reverb is used for data augmentation in order to enable the simulation of different environments.

3.2.5 *Scaling & Warping*

Scaling and Warping are two closely related concepts. They both can be performed on either the magnitude or the frequency of audio signals. While scaling refers to uniformly adjusting the amplitude or frequency of a signal, warping adjusts those characteristics non-uniformly. For example, time warping (i.e., non-uniformly changing the signal frequency) might include warping only certain slices of an audio signal with a subsequent speed adjustment of the whole sample. Magnitude sampling can include multiplicating a whole sample with a cubic spline curve.

3.2.6 *Noise Injection*

Analogously to the image domain, noise injection can be used to artificially decrease a sample's quality. In the audio domain, noise injection is mostly done by adding gaussian noise to a sample to simulate data imperfections that might occur in real-world data.

3.3 DATA AUGMENTATION WITH GANS

Contrary to traditional data augmentation, more recent approaches do not follow the idea of altering existing data - they try to generate completely new data. This means, they try to sample from a specific data distribution instead of modifying data that already was available. Further, as described in Chapter 2, GANs seem to be a perfect fit for generating new data. As such, it is not surprising that with the advent of GANs, many researchers have tried to use them for data augmentation (A. Mumuni and F. Mumuni, 2022; Frid-Adar et al., 2018; Mariani et al., 2018). The basic idea of most works that exist in that area is to train a GAN on the same dataset that will later be used for training the machine learning model to solve a specific task (for example, a classification or segmentation network). However, before training that final model, the dataset is augmented by many more sample points that are generated by the GAN. Although this might sound very promising, there is one big drawback: Although new data is generated, that data still stems from the same distribution as the data that we had in the first place - as this is exactly what the GAN was trained for. As such, in order to hold *new* information, we have to trust in the GAN to be able to model *other* information that the classification or segmentation model learns later on - and generally, there is no reason to assume so. This leads to a distribution problem - although we are capable of augmenting a dataset with new *data*, we are not necessarily enhancing it with new *information*, because we are still stuck in the same distribution. Because of that, we need to find workarounds that allow us to use GANs while escaping the initial data distribution - enabling to generate actual new information for a dataset. In the following chapters, we will introduce two concepts (one for a classification problem in the audio domain, and one for a segmentation problem in the image domain) that we developed. Those two approaches address exactly the aforementioned distribution problem: we want to find ways to use GANs for data augmentation that introduce new *information* to a dataset.

4

DATA AUGMENTATION WITH LATENT VECTOR EVOLUTION

Large parts of this chapter have already been published in the following publication:

> *Mertes, S., Baird, A., Schiller, D., Schuller, B. W., & André, E. (2020). An evolutionary-based generative approach for audio data augmentation. In 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP).* (Mertes. et al., 2020)

In this chapter, we introduce a novel two-step approach to address the aforementioned problem of GANs not being able to generate new *information* when using them for data augmentation. In the first step, we utilize a GAN framework to create highly realistic audio data. In the second step, we then apply an evolutionary algorithm to search the input-space of the generative model for vectors that result in samples that have specific predefined characteristics. These characteristics represent information that is lacking in the original source data of the respective classes. The concrete feature values that shall be exhibited by the new data are determined by analyzing samples that were previously classified wrong. This way, the GAN is employed to only generate training samples that are useful for a specific classification task.

To evaluate our system, we tackle the problem of soundscape classification. Thus, we are building a system that is able to create new audio samples of soundscapes in a controlled way to improve the training of a Support Vector Machine (SVM) whose task is to differentiate between different soundscapes.

4.1 RELATED WORK

Multiple variants of GANs have been used previously to generate highly realistic audio data (Donahue, McAuley, and Puckette, 2018; Engel et al., 2019; Chandna et al., 2019). Further modifications, such as conditional GANs, enable the generation of audio data that exhibits specific characteristics (e.g., (C. Y. Lee et al., 2018)). However, these systems require labeled training data for each desired target characteristic of the generated data. In return, this means that the network needs to be trained from scratch each time a change in the target properties of the data is required. Artificial data that was generated by GANs has been used for data augmentation predominantly in the field of image processing (e.g., (Bowles et al., 2018; Mariani et al., 2018)), but there is also recent work that makes use of GAN-based data augmentation for acoustic scene classification (Madhu and Kumaraswamy, 2019; J. H. Yang, N. K. Kim, and H. K. Kim, 2018; Mun et al., 2017) as well as emotional speech (Rizos et al., 2020; Baird, Amiriparian, and Schuller, 2019). Most of these approaches are not able to generate the augmented data in a controlled manner, but rather use the GANs to produce random new samples to enhance existing datasets. As could be shown in the respective publications, such GAN based augmentation techniques are a promising approach. However, existing experiments in the audio domain did only operate on rather big datasets and therefore leave open the question of whether uncontrolled data augmentation with GANs can also be applied to rather small datasets.

A recent approach to address the controllability of GANs relies on the application of evolutionary algorithms to search through the solution space of GANs and find appropriate samples that match the required characteristics, i.e., predefined feature values that shall be exhibited. Thus, the randomness of the generated samples can be overcome. This so called *Latent Variable Evolution* (LVE) has been successfully employed for tasks like fingerprint-based biometric systems (Bontrager et al., 2018), the creation of video games (Volz et al., 2018; Giacomello, Lanzi, and Loiacono, 2019; Schrum et al., 2020) or facial composite generation (Zaltron, Zurlo, and Risi, 2019). The ability to generate samples in a targeted way makes LVE a promising approach to enhance datasets with data that is actually meaningful for the respective classification task. To the best of our knowledge, there is no prior work that uses the principles of LVE for either raw audio or even the task of data augmentation.

4.2 APPROACH

Our proposed approach deals with the problem of augmenting raw audio datasets in a controlled manner, using generative adversarial networks in two steps. To this end, we generate artificial data samples which exhibit characteristics that are underrepresented in the original dataset. In the first step, we train a WaveGAN architecture to produce new samples of a certain class using random noise vectors as input. In the second step, we use an evolutionary algorithm to search the

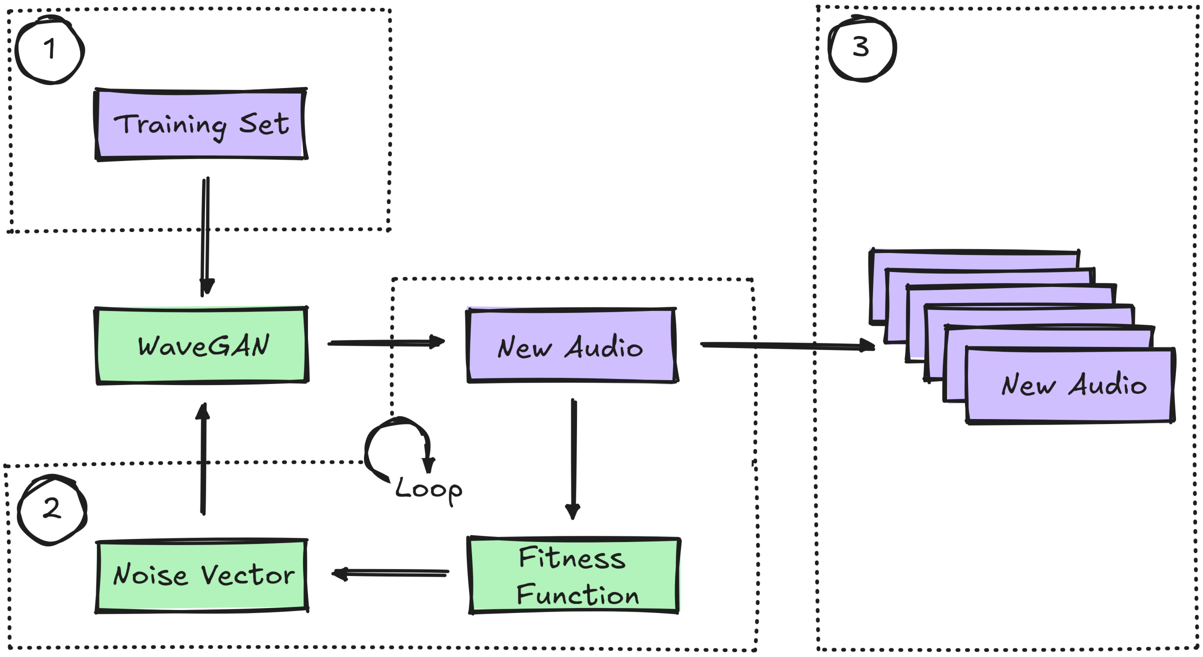


Figure 19: Overview of our approach. (1) A WaveGAN model is trained on our dataset. (2) An evolutionary algorithm is used to find appropriate noise vectors to create new audio samples that exhibit predefined characteristics. (3) Those new audio samples are collected and taken as augmented data to enhance the existing dataset.

input space of the WaveGAN for vectors that result in samples that show the desired feature values. An overview of the system is shown in Fig. 19. This section describes both the used WaveGAN architecture as well as the evolutionary algorithm.

4.2.1 *WaveGAN*

The WaveGAN architecture was first introduced by Donahue, McAuley, and Puckette (2018). The authors showed that WaveGAN is capable of generating realistic sounding audio data for tasks that are related to nature soundscapes, such as bird sounds. Its main concepts follow the basic idea of DCGANs. As described in Chapter 2, DCGANs are a modification to the initial GANs which enable the modelling of data with even higher complexity by including convolutional layers to both the generator and the discriminator network (Radford, Metz, and Chintala, 2015). As DCGAN was developed for image generation, multiple parts of it are slightly modified by WaveGAN to enable the handling of audio data. For example, the two-dimensional up- and downsampling filters are replaced by its one-dimensional equivalent (i.e., kernels of size $n\ times\ n$ become kernels with size $n * n$). For details, please refer to the original work about WaveGAN (Donahue, McAuley, and Puckette, 2018) and its official repository[1].

[1] https://github.com/chrisdonahue/wavegan

It is worth mentioning that our further approach is independent from the chosen GAN architecture. As a result, the underlying GAN architecture can be replaced for any other model, depending on the scope of the respective application.

4.2.2 *Evolutionary Algorithm*

After training a WaveGAN model on a specific domain, that model is able to transform random noise vectors to audio samples that follow the distribution of the training dataset. Thus, new audio samples can be generated that never had been heard before but sound as if they originated from the learned domain. To find audio samples that show certain feature characteristics that we want to control, we follow the idea of Latent Variable Evolution to search through the solution space of the trained WaveGAN model. First, we initialize a starting population of random noise vectors and feed them to the trained WaveGAN. Subsequently, we evaluate the resulting audio data by using a predefined fitness function that measures how appropriate the samples are with respect to the feature values that we want to have, i.e., feature values that add information to our training dataset for the classification stage, as is described in more detail in section 4.3. The noise vectors that performed best are then slightly mutated and recombined, whilst the other noise vectors are being discarded. The new noise vectors that originate by the mutation of the best prior noise vectors can then be fed to the trained WaveGAN again. This process is repeated until audio samples are found that show the desired feature value. The procedure is described more formally in the following.
Let x be the output of the generator, denoted as $x = G(z)$, where the function G represents the transformation learned by the generator that takes a latent space vector z as input. Further, we define a measurement function $f(x)$ that calculates the value of the feature that we want to control. Thus, $f(G(z))$ corresponds to the feature value that is achieved by feeding z to the generator. We denote the value that we want the feature to be as t and call this the target value. Thus, a *perfect* noise vector z for target value t would fulfill $f(G(z)) = t$.
As described above, we chose the best noise vectors to be mutated and recombined further by a fitness function. We can find such a fitness function $\mathrm{fitness}(z)$ easily by constructing the reciprocal of the distance between the shown feature value and the target value:

$$\mathrm{fitness}(z) = \frac{1}{|f(G(z)) - t|}$$

We use this fitness function to train an Evolutionary Algorithm. The term Evolutionary Algorithms denotes a class of optimization methods that are inspired by the evolution of natural living beings. In general, they work by iteratively generating a new set of data samples (*in-*

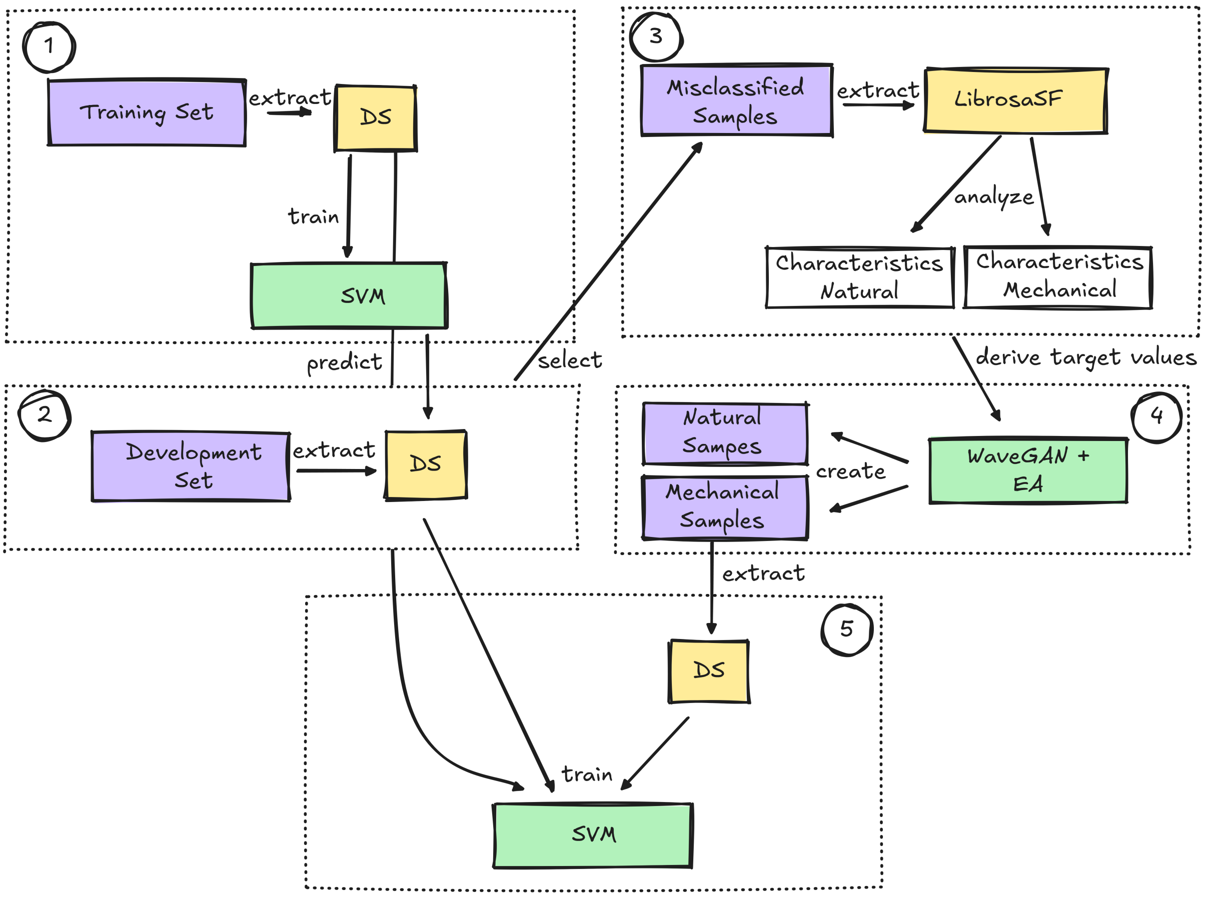


Figure 20: Illustration of the Augmentation Process. (1) The SVM is trained on the training partition of the original data using Deep Spectrum (DS) features. (2) The trained SVM is used to predict samples from the development partition. (3) The misclassified samples are analyzed regarding six standard spectral features extracted with the Librosa library. The mean and standard deviation per class and feature is calculated. (4) Augmentation samples are generated by the use of the Evolutionary Algorithm and the WaveGAN, using the calculated feature range as target values. (5) The final SVM is trained on the augmented data and the train and development partitions of the original data.

dividuals) - a so-called *population* - and choosing the best out of these samples (*selection*) regarding a predefined fitness function, and alter those samples in various ways to get a new population that is (hoped to be) better than the previous one (*Mutation and Recombination*). *Evolution Strategies* (ES) are a group of evolutionary algorithms that are mainly used for multidimensional, continuous problems (Beyer and Schwefel, 2002). This property makes them an ideal fit to operate on the latent vector that is used as input for the GAN. Specifically, we chose a $(\mu/p+\lambda)$-ES. This means, that the best μ individuals of the parent population generate λ new individuals, whereas the parent individuals are also included into the following generation of individuals. p represents the group size of the recombination, i.e., p individuals of the parent population are responsible for the creation of a new individual simultaneously. For our evaluation, we chose the following - empirically determined - parameters: $\mu = 50$, $p = 2$, $\lambda = 150$.

Thus, we used a population size of $\mu + \lambda = 200$. We configured our algorithm to create 50 new individuals by recombination and 100 new individuals by mutation, as this led to the best results in our experiments.

We chose *Uniform Crossover* as recombination method. This means, that for the generation of a new individual out of two parent individuals, there is a stochastic decision process for every element of the new vector to determine if the element is taken from either the one or the other parent. To formalize this, let z^1 and z^2 be the n-dimensional parent latent vectors with $z^1 = (z_1^1, z_2^1, ..., z_n^1)^T$ and $z^2 = (z_1^2, z_2^2, ..., z_n^2)^T$. Also, let $o = (o_1, o_2, ..., o_n)^T$ be the offspring individual that shall be derived from z^1 and z^2. Then, for every $i \in \{1, 2, ..., n\}$, it is randomly decided if either $o_i = z_i^1$ or $o_i = z_i^2$.

For the mutation operations, we made use of a Gaussian mutation operator. Given a parent vector $z^3 = (z_1^3, z_2^3, ..., z_n^3)^T$ that shall be mutated to an offspring $mo = (mo_1, mo_2, ..., mo_n)^T$, mo is determined as follows:

$$\forall i_{\in\{1,...,n\}} : mo_i = z_i^3 + \mathcal{N}(0, \sigma^2)$$

Here, $\mathcal{N}(0, \sigma^2)$ is the mutation value that is randomly sampled from a Gaussian Distribution, where the variance σ^2 is chosen according to the *1/5 Success Rule* (Beyer and Schwefel, 2002).

4.3 EXPERIMENTS

To test the validity of our approach, we chose to evaluate it on the task of soundscape classification. Due to the complex nature of soundscapes, which consist of a large variety of individual sounds, this task is very challenging.

To assess the impact of our data augmentation approach on the performance of a soundscape classification system, we perform multiple experiments in which we augment existing datasets with respect to specific, underrepresented characteristics, from which we expect that they contribute to improve the performance of a classification model. The following section describes our experimental setup as well as the methodology that we applied.

4.3.1 *Methodology*

As described in the previous section, our approach is able to generate new audio samples that exhibit predefined characteristics. We use this method to augment data in a controlled way to improve an SVM based soundscape classifier that predicts if a sample belongs to either of the two classes *mechanical* or *nature*. To this end, we train the classification system on three different datasets and compare the results. The first dataset (*dataset_orig*) contains only original data, while the second (*dataset_aug*) was enhanced with data that was randomly generated by feeding arbitrary noise vectors to a WaveGAN that was trained on the original data.[2] For the third dataset (*dataset_aug_ctrl*), we applied our approach to the same training procedure that was used for *dataset_aug*. All of the three datasets were partitioned into train, development and test, where the development and test partitions remain the same, and the train partition of *dataset_orig* was enhanced with different augmentation data for *dataset_aug* and *dataset_aug_ctrl*. We did not use traditional data augmentation techniques for any of the datasets, as we wanted to focus on the advantages of targeted augmented data over random GAN-based augmented data. The three datasets are discussed in detail in the following sections. The complete augmentation process for our experiments is depicted in Figure 20.

4.3.2 *Original Dataset*

For evaluation purposes, we chose to perform our experiments on a subset of the Emotional Soundscapes database (Fan, Thorogood, and Pasquier, 2017). The dataset contains audio files of certain soundscapes which are sorted by environment. We decided to consider only the two classes of *mechanical* and *natural* environments, as the samples of these classes, despite their fundamental differences, generally have a very noisy appearance, which makes them often hard to distinguish even for humans. For example, it can be very hard to differentiate between a waterfall and the background noise of a room full of different kinds of machines, since both sounds are similar in terms of their low frequency range and regular noisiness. The nature class has many samples that contain large parts of silence. As these samples would complicate the feature extraction as well as the WaveGAN training, we removed them from the dataset, resulting in an increase in mean RMS energy level of the nature class from $2.18 * 10^{-2}$ to $2.64 * 10^{-2}$. We split all audio files into samples of 1 second length as our WaveGAN architecture produces outputs of a fixed size. Our final first dataset contains 600 samples (10 minutes) for the mechanical class and 300 samples (5 minutes) for the nature class. As mentioned above, the dataset was split into train, development and test partitions. The train partition for this dataset contains 420 samples (7 minutes) for mechanical and 210 samples (3.5 minutes) for natural. The development partition contains 60 samples (1 minute) for mechanical and 30 samples (0.5 minutes) for natural, while the test partition contains

[2] Example output of the trained WaveGAN can be found at https://tinyurl.com/y83rhjbb

Table 1: Spectral Librosa features and respective mean and standard deviation values that were used for the evolutionary algorithm.

	Wrongly classified as mechanical		Wrongly classified as nature	
Feature name	Mean	Std. Deviation	Mean	Std. Deviation
Spectral Centroid	1555.69	484.31	2828.69	143.44
RMS	0.08	0.11	0.01	$5.91*10^{-3}$
Spectral Bandwidth	1828.53	323.13	2169.40	89.97
Spectral Contrast	21.65	0.76	21.10	0.21
Spectral Flatness	0.002	$2.15*10^{-2}$	0.005	$2.76*10^{-3}$
Spectral Rolloff	3512.36	1321.68	5469.17	351.83

120 samples (2 minutes) for mechanical and 60 samples (1 minute) for natural. Data augmentation, as described below, is only applied to the train partition.

4.3.3 *Untargeted Augmentation*

For our second dataset (*dataset_aug*), we train WaveGAN models as described in section 4.2.1 for both the mechanical as well as the natural class. As training sets for the WaveGAN, we take the respective classes of both the train and development partitions of our original dataset *dataset_orig*. The WaveGAN models were trained for 200,000 iterations before we used them to generate random new data of both classes. 420 data samples (7 minutes) are generated per class. Our complete training set contains both the original data as well as the randomly generated samples.

4.3.4 *Targeted Augmentation*

The third dataset (*dataset_aug_ctrl*) contains the original data from *dataset_orig* as well as audio samples that were generated in a targeted way by the use of our approach. As described in section 4.2.2, we make use of an evolutionary algorithm to find the samples that show the feature values that we want to have. Our assumption is the following: if a classifier is not able to classify certain samples in a correct way, then it might lack training data that shows similar feature values as the ones that were classified wrongly. We therefore aim to generate new training data that exhibits feature characteristics of the previously wrong classified data. To find appropriate target values for the evolutionary algorithm, we analyze the samples of the development set that were classified wrongly by the use of the SVM that was trained only on *dataset_orig*. Specifically, we look at six standard spectral features (*Spectral Centroid, RMS, Spectral Bandwidth, Spectral Contrast, Spectral Flatness and Spectral Rolloff*). It is noteworthy that this feature set is only used to select the specific audio samples, but not for the classification itself. To this end, we rely on the DEEP SPECTRUM feature set described in section 4.3.5.

We calculate the mean m and standard deviation s of these features over all misclassified samples of one class, as shown in Table 1. Based on those values, we determine a range $[m - s, m + s]$ for each feature and class that we want to generate new data for. We decided to generate five samples of audio for each of the six features per class, resulting in 30 augmented audio samples per class. As can be noted, we generated much less augmented data for this experiment than we did for *dataset_aug*. By doing so, we want to verify our assumption that small amounts of targeted augmented data are adding more information to the classification task than comparably high amounts of untargeted augmentation data. To get the five target values that we need for our evolutionary algorithm, we tried to cover the range that we found by analyzing the false classified samples. Let $m_{i,c}$ be the mean and $s_{i,c}$ the standard deviation of the feature i for class c. We calculated our target values $t^1_{i,c}, t^2_{i,c}, ...t^5_{i,c}$ for the respective feature and class as follows:

- $t^1_{i,c} = m_{i,c} - s_{i,c}$
- $t^2_{i,c} = m_{i,c} - 0.5 * s_{i,c}$
- $t^3_{i,c} = m_{i,c}$
- $t^4_{i,c} = m_{i,c} + 0.5 * s_{i,c}$
- $t^5_{i,c} = m_{i,c} + s_{i,c}$

With these target values, we are able to cover a big range of the feature values that were missing in the initially wrong classified data samples. We trained the evolutionary algorithm with the respective target function for each of the 30 samples per class, resulting in 60 runs of the evolutionary algorithm. The feature values of all individuals were calculated with the *Librosa* (McFee et al., 2015) library during training. Every training run was stopped after 100 iterations.

4.3.5 *Deep Spectrum Features*

As the nature of soundscapes is complex, we chose a spectrogram based approach for extracting features that are used as input for the classification stage. We assume that this would inherently capture a larger portion of temporal information as compared to other conventional acoustic features. To this end, we extracted a 4 096 dimensional feature set of deep data representations using the DEEP SPECTRUM toolkit (Amiriparian et al., 2017)[3]. DEEP SPECTRUM has shown

[3] https://github.com/DeepSpectrum/DeepSpectrum

success for similar audio tasks (Baird, Amiriparian, and Schuller, 2019), and extracts features from the audio data using pretrained convolutional neural networks. For this study, we extracted spectrograms using the default Deep Spectrum settings including a VGG16 pretrained network, extracting one feature vector per audio sample.

4.3.6 *Support Vector Machine*

For all machine learning experiments, we used a Support Vector Machine with a linear kernel. During the development phase, we trained a series of SVM models, optimizing the complexity parameters ($C \in 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1$) and evaluating their performance on the development partitions. We then re-trained the model with the concatenated train and development partitions and evaluated the performance on the test partition. This whole procedure was done for each of the three datasets. As a measure of accuracy we report Unweighted Average Recall (UAR), as we wanted to take class imbalance into account.

4.4 RESULTS

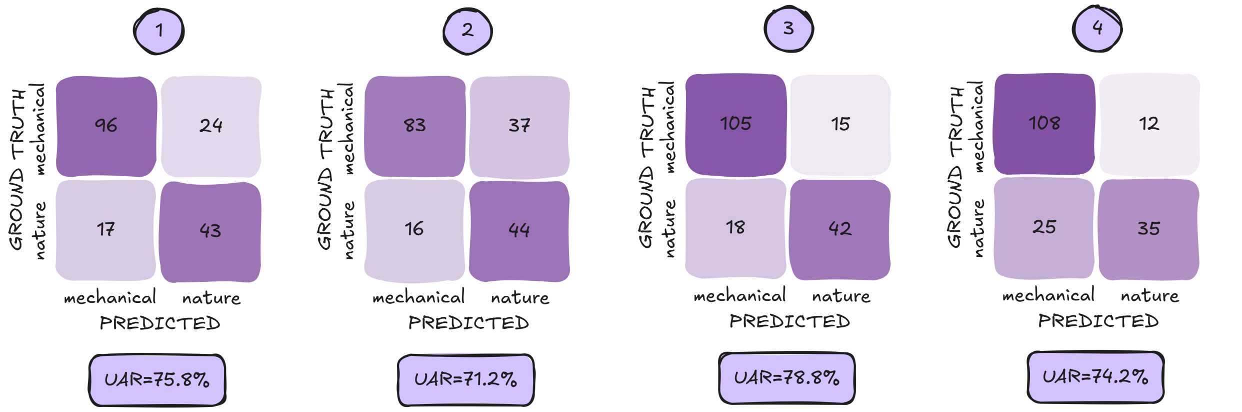


Figure 21: Confusion matrices for test partition of the SVMs that were trained on the different datasets. (1) Trained on original data only. (2) Trained on original data and randomly generated data. (3) Trained on original data and targeted augmented data, our approach. (4) Trained on targeted augmented data only.

After we trained the SVM on the three datasets, we evaluated all models on the test partition. In this section, we report our results for each dataset. As mentioned above, we report the UAR that each of our SVM models achieved with the respective optimal complexity parameter. As the task is to perform a binary classification, the chance level is represented by a UAR of 50.0. Our baseline model that was trained on the original data (*dataset_orig*) achieves an accuracy of 75.8% UAR. *Dataset_aug*, that contains randomly generated samples among the original data, results in a UAR of 71.2%. As can be seen, this value is remarkably below the first model. This leads to the deduction that the data that was generated by feeding random noise vectors to the GAN does not add meaningful information to the SVM during the training process, even making the training set worse. This shows the need for the generation in a controlled way, as applied in *dataset_aug_ctrl*. The model that was trained on the targeted augmented data achieves a UAR of 78.8%, thus outperforming both the classifiers that were trained on *dataset_orig* and *dataset_aug*. To evaluate this effect further, we trained a fourth model only on the targeted augmented data without the original data. This model achieves a UAR of 74.2%, thus being slightly below the baseline model. This is reasonable since the target values for the Evolutionary Algorithm were derived by analyzing the false classified samples from the classifier that was trained on *dataset_orig*, thus trying to specifically add information for value ranges that are not yet modelled in that dataset, but not claiming to be able to model the whole possible range of the features of the original data. The corresponding confusion matrices for each model are shown in Fig. 21.

4.5 DISCUSSION

When comparing the results of *dataset_orig* (that only contains the original audio samples) with *dataset_aug* (that adds randomly generated data to the training process), it can be seen that the performance of the trained SVM model considerably drops. This shows, that there is in fact a need for augmentation in a somewhat targeted way, although recent works could also achieve performance boosts while working with a random generation process (Madhu and Kumaraswamy, 2019). It is conceivable that the random generation in our problem domain is not sufficient due to the fact that our original dataset is very small compared to the datasets that were used in those previous works. However, the results that could be achieved with *dataset_aug_ctrl* outperform both the models from *dataset_orig* and *dataset_aug*. As *dataset_aug_ctrl* makes use of our controlled generation process, it is capable of adding augmentation data that is actually helping the classification task. Although the solution space that was learned by the WaveGAN has to be rather small - as we used only small amounts of data to train it - the Evolutionary Algorithm was able to find meaningful samples in that space. It is noteworthy that even considerably less training samples were generated for *dataset_aug_ctrl* than for *dataset_aug*. This shows that even small amounts of targeted augmentation data are better for the classification task than high amounts of randomly generated data.

4.6 CONCLUSION

In this chapter, we presented a new approach for augmenting training data for an audio classification problem in a targeted way. Therefore, we combined a GAN model - trained on the original training data - with an evolutionary algorithm. That evolutionary algorithm was used to steer the GAN into generating samples that actually are helping a task at hand. As the mutation strategy of our evolutionary algorithm was not restricted to stick to the latent distribution that the GAN was trained with, we successfully enabled the synthesis of new samples that actually hold information that the GAN did not synthesize when used without the evolutionary algorithm. Therefore, we showed that our approach has substantial advantages for our problem domain when comparing it to adversarial augmentation techniques that rely on a random class-wise augmentation.

We can summarize that in our chosen problem domain, the approach works reasonably well and shows potential to improve a broad range of classification problems that are existent in the current research community.

5

DATA AUGMENTATION THROUGH LABEL AUGMENTATION

Large parts of this chapter have already been published in the following publications:

> *Mertes, S., Margraf, A., Kommer, C., Geinitz, S., & André, E. (2020). Data augmentation for semantic segmentation in the context of carbon fiber defect detection using adversarial learning.* (Mertes, Margraf, Kommer, et al., 2020)

> *Mertes, S., Margraf, A., Geinitz, S., & André, E. (2023). Alternative data augmentation for industrial monitoring using adversarial learning. In International Conference on Deep Learning Theory and Applications - Revised and Selected Papers.* (Mertes, Margraf, Geinitz, et al., 2023)

In the last chapter, we focused on augmenting data for a classification problem in the audio domain. In this following chapter, another GAN-based data augmentation technique will be introduced. However, this time, we will address a *segmentation* problem in the *image* domain. Specifically, we will tackle a scenario that is important for the producing industry: AI-based visual inspection and defect detection. Visual inspection includes a wide repertoire of methodologies in industrial quality control. With the increasing level of automation and digitalisation in the manufacturing industry, automatic sensing technology has drawn much attention in the field through its potential to make time-intensive manual inspection of production processes obsolete. The field of online process monitoring primarily deals with imaging technology to detect changes, faults or potential anomalies in continuous production environments. Here, intelligent image processing is a key feature of monitoring systems.

In recent years, machine learning algorithms have progressively overtaken more traditional methods that were based on predefined filters (Cavigelli, Hager, and Benini, 2017; McCann, Jin, and Unser, 2017). In particular, Convolutional Neural Networks (CNNs) have become the state-of-the-art in online process monitoring. If large training datasets are available, they are flexible across domains and therefore can be applied to very different kinds of applications (Simonyan and Zisserman, 2014; K. He et al., 2016).

However, highly specialized industries are often confronted with incomplete or insufficient data. With increasing effort spent on data collection and preparation - tasks that require time and skilled personnel - deep learning models become inefficient, expensive and therefore unattractive. As such, online process monitoring, and industrial defect detection in particular, serves as perfect real-world scenario to study how GANs can help to augment such scarce datasets.

Specifically, in this chapter, we consider semantic segmentation of carbon fiber textiles with unique surface structures and heterogeneous anomalies. In practice, such anomalies need to be detected so that respective textiles can be removed from the remaining production process.

In contrast to image *classification* problems, where a class is assigned to a whole image, *semantic segmentation* refers to assigning a class to every single pixel of the image. As such, the labels (i.e., the ground-truths or label masks) are of much higher dimension than they are for classification tasks. In the simplest form - where only two classes exist - such a label has the form of a binary image. In the case of carbon fiber defect defection as we address here, a pixel that holds a 0 would indicate that this pixel does not belong to an area with a defect - while a 1 would indicate a defect area. Fig. 22 shows examples of defect images with corresponding label images.

To augment a dataset that contains such data, not only new image samples have to be generated for specific classes (like it is the case for augmenting data for classification tasks), but new *pairs* of images *and* label masks have to be created. Although this might seem difficult at first glance, it also offers a new possibility: Instead of using augmentation techniques to synthesize new image samples *directly*, we can synthesize label masks and *subsequently* transform them to real-looking image samples. Doing so has two advantages: First, the label mask space is not as complex as the image data space, which eases creating artificial ones. Second, and even more important: By augmenting the label space and successively transforming the labels to image data, the label synthesis mechanism does not have to be dependent on the distribution of the real image data. As such, it is promising to use that kind of *Label Augmentation* (with downstream style transformations to achieve image samples) in order to enhance the dataset with samples that are *not bound to the original data distribution*.

In this chapter, for the generation of such new labels, we propose two distinct concepts: the first approach is based on a handcrafted

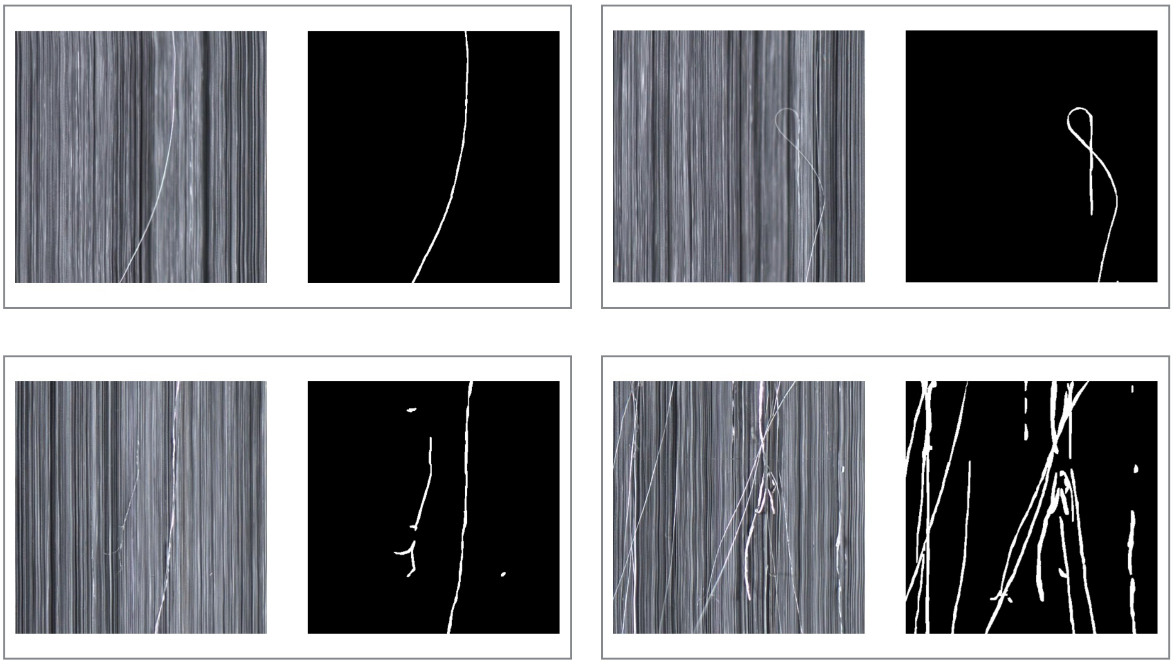


Figure 22: Examples of real image data pairs labeled by experts. The misaligned fibers are visible on top of the fiber carpet.

mathematical model precisely tailored to the application. Handcrafting such problem-specific models requires good domain understanding and further effort to model an algorithm and tune it to adjust to the given problem. The resulting function is transparent and human-readable which allows for better debugging and testing.

The second one uses a WGAN model trained on the original binary labels in order to allow the generation of synthetic labels. That proposed WGAN-based approach requires less manual effort than the former approach and automates a considerable part of the overall process.

In both cases, binary label images are generated and used for further processing, i.e., for generating image/label pairs by using an image-to-image translation system. That image-to-image translation again leverages a GAN-based architecture.

All in all, we present a novel approach to augment image data for semantic segmentation tasks by applying image-to-image translation on label masks that were created with either a domain-specific mathematical model or an approach entirely based on generative models. As such, our methods allow for artificially creating label and image pairs that are actually new and serve as training samples for semantic segmentation models. We test both approaches based on images containing carbon fiber surface defects and discuss the results.

5.1 RELATED WORK

5.1.1 *Industrial Defect Detection*

Several publications address industrial monitoring applications: Masci et al. (2012) used CNNs for classification of steel defects, and Soukup and Huber-Mörk (2014) used CNNs for photometric stereoscopic images. A region proposal network for real-time object detection was presented by Ren et al. (2015), while Ferguson et al. (2018) used CNNs and transfer learning to detect X-ray image defects. Furthermore, the use of CNNs for industrial surface anomaly inspection was explored by Staar, Lütjen, and Freitag (2019).

GANs have also already been used for anomaly detection in industrial use-cases. For instance, Schlegl et al. (2017) published a work in which GANs for marker detection were used for unsupervised anomaly detection. A survey exploring GANs for anomaly detection was presented by Di Mattia et al. (2019).

The identification of anomalies of carbon fibers in particular, e.g., the misalignment of textile surfaces, has also been discussed in various publications - ranging from studying appropriate hardware setups for monitoring the analyzed fibers (Geinitz, Margraf, et al., 2016; Geinitz, Wedel, and Margraf, 2016), over examining sophisticated feature modalities like thermography (K. Liu et al., 2022), to filter-based detection pipelines (Margraf et al., 2017; Margraf. et al., 2020; H. Wang et al., 2024) and DL-based methods (Szarski and Chauhan, 2022).

All these works show that the producing industry has a huge demand for automated mechanisms for defect detection. As such, to evaluate our novel approach for data augmentation, we chose a specific use-case from that field - and try to augment data for defect detection in carbon fibers.

5.1.2 *GAN-based Data Augmentation for Industrial Use-Cases*

As already introduced in Section 3.3, GAN-based data augmentation has been applied to various use-cases. This holds also true for industrial domains. There, GANs have been used to augment data in a variety of scenarios - also mostly in the context of defect detection for quality assessment - e.g., in the coffee industry (Y.-C. Chou et al., 2019), lifestyle industry (Rožanec et al., 2023), for machine fault diagnosis (Shao, P. Wang, and Yan, 2019; Ortego et al., 2020), surface defect detection (S. Jain et al., 2022) or wear prediction (Yuechi Jiang, Drescher, and Yuan, 2023). However, the problems mentioned in Section 3.3 remain: Using GANs to (more or less) randomly add new data to a dataset might have flaws, as it is not guaranteed that the data actually holds information that hasn't been already included in the initial dataset.

5.1.3 *Domain Adaptation with GANs*

A concept that is conceptually related to our approach is the so-called *Domain Adaptation*. Domain adaptation aims to adjust a model trained on a source domain so that it performs effectively on a target domain with a different data distribution. The objective is to reduce the performance disparity caused by domain shifts while utilizing knowledge from the source domain. Often, domain adaptation is approached with GANs. In particular - similar to this work - style conversion models can be used there. This has been shown for tasks like object detection (M. Zhang et al., 2022; Menke, Wenzel, and Schwung, 2022), object classification (Bejiga and Melgani, 2018), activity recognition (Sanabria, Zambonelli, and Ye, 2021), and semantic segmentation (S. Scherer et al., 2021; J. Choi, T. Kim, and C. Kim, 2019), but also for domain-specific scenarios like generating digital twins (Ulhas, Kanapiran, and Berman, 2024). However, in contrast to our work, domain adaptation aims to convert data that follows one specific distribution to data that follows another distribution, where both distributions have the same *syntactic* meaning (for example, both distributions might represent images). In our approach, the source distribution is artificially created *label* data, and our target domain is the actual data (in our *specific* case, images of carbon fibers).

5.2 APPROACH

The following sections explain the concepts that are introduced in this chapter. The main idea of our approach is that we are enhancing datasets with augmented data by artificially modeling label images, and after that convert those into real image data. By doing so, we get image/annotation pairs that are needed for the training of neural networks for semantic segmentation tasks. To create new label data, we propose two different methods. The first method is an algorithm specifically designed for our particular application at hand, i.e., defect detection in carbon fiber structures. It is based on a randomized label generator that uses a stochastically parametrized function to build segmentation masks. The second method is a more generic one. It uses a WGAN that is trained on raw segmentation masks. After training, the WGAN is capable of generating new label images that appear similar to the original labels. By applying this concept, we get rid of the engineering overhead that is necessary when using the first method. While the randomized label generator that is used by the first method has to be defined and optimized specifically for every new segmentation task, the WGAN should be able to learn the label structure of new tasks by itself.

The labels that are produced by either of the two methods are fed into a *pix2pix* network that was trained on an image-to-image conversion

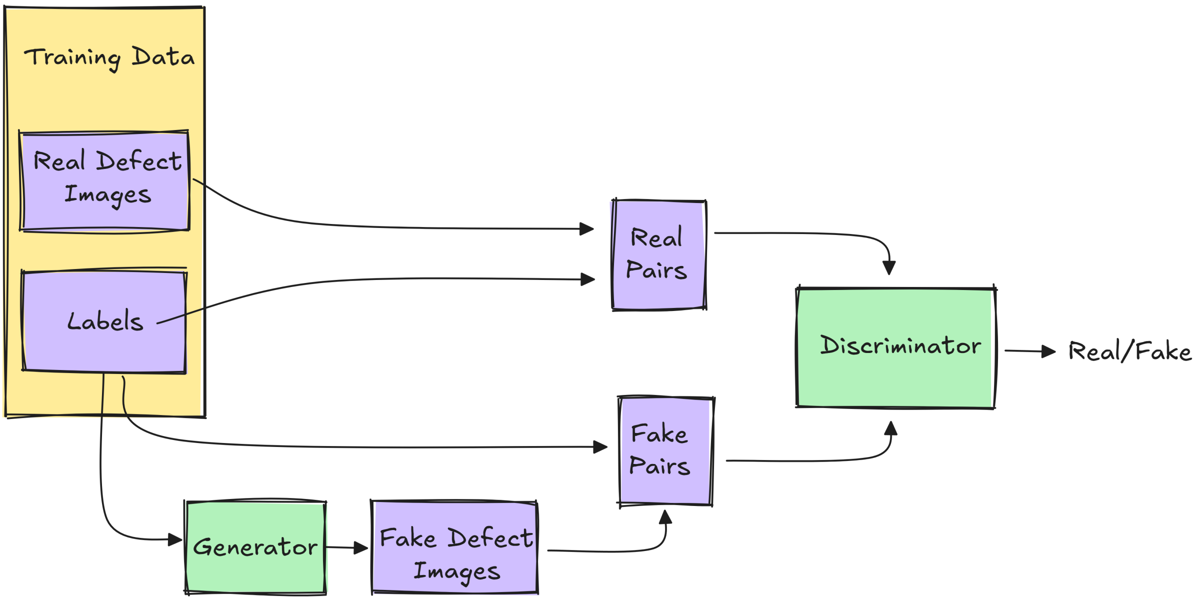


Figure 23: Training of a *pix2pix* network to perform image-to-image translation between labels and defect images (Step 1).

task, i.e., the network was trained to convert label mask images to real images that fit to the respective labels, thus resulting in image/label pairs that can be used to enlarge training datasets.
All in all, our approach can be seen as a three-folded system: first, we train a *pix2pix* network on an image-to-image translation task, so that it learns to perform a translation from labels of defect images to their corresponding image data. Second, one of the aforementioned methods is used to generate synthetic label data. At last, the synthetic label data is fed into the trained *pix2pix* network, resulting in new training pairs for further machine learning tasks. The following sections explain these steps in more detail.

5.2.1 *Label-to-Image Model*

In order to convert label masks to corresponding images, we trained a *pix2pix* model. For the training, a dataset of existing real defect images and manually labeled annotation masks was used. The basic scheme of the training process is depicted in Fig. 23. We used the *pix2pix* network architecture (Isola et al., 2017) as described in Section 2.9.1. We adapted the size of the input layer to fit the dimensions of our dataset. Other than that, we did not make any modifications to the originally proposed architecture.

5.2.2 *Synthetic Label Generation*

The idea of our approach is to feed new synthetic label data into the *pix2pix* network in order to obtain new pairs of defect images and label masks. As mentioned above, two different methods were applied for the stage of synthesizing new label data.

5.2.2.1 *Mathematical Modelling of Defects*

The first method is based on the observation that in many application scenarios, label masks have a common structure. The approach is illustrated in Fig. 24. The idea behind this first approach to generate fake label masks is to find a mathematical description of those structures for a specific use-case. In the application scenario that serves as an example for evaluation in this chapter - the mentioned defect detection on carbon fiber structures - label masks usually appear as mostly straight or curved lines. Those structures can be seen as a combination of multiple graphs with different rotations and varying thickness of the plots. Thus, the mathematical description of a single defect label could be approximated through a handcrafted function. By adding a stochastic factor to such a function, we can plot different graphs that can be considered as new, artificial label masks. For our specific task, we conducted several experiments that showed that the following function $f(x)$ can be used to cover a huge part of carbon fiber defect structures. We denote $f(x)$ as:

$$f(x) = a_1 \cdot \sin(a_2 \cdot x) + a_3 \cdot \sin(x) + a_4 \cdot \cos(a_5 \cdot x) + a_6 \cdot x + a_7 \cdot x^2$$

where the parameters a_n are chosen randomly within certain defined intervals. For our specific experiments, we found appropriate intervals by visual expection of carbon fiber defect images. By using those intervals (listed in Table 2), we ensure to cover a wide range of different defect structures. To that end, a sine function was tuned with a rather big amplitude to model the *global* structure of a defect, which is typically shaped in curves. For the structure on a more *local* level, we used another sine with a much smaller amplitude interval. Aperiodic curvings were modeled by the use of polynomic functions. We randomly set the variables and plotted the resulting graph for every fake label for $x \in [0, w]$ where w represents the width of the sample images. After creating those plots, they were rotated randomly. At last, we took a random subset of those graphs, randomized the thickness of the resulting lines, and overlapped those graphs to create images of labels with a realistic fiber-like appearance.

It has to be noted that this method is very specific to the given task at hand. A lot of engineering time and effort has to be spent to find sufficient mathematical models for different use-cases. However, similar approaches for defect-modeling have been successfully applied to similiar problems in previous work (Haselmann and Gruber, 2017).

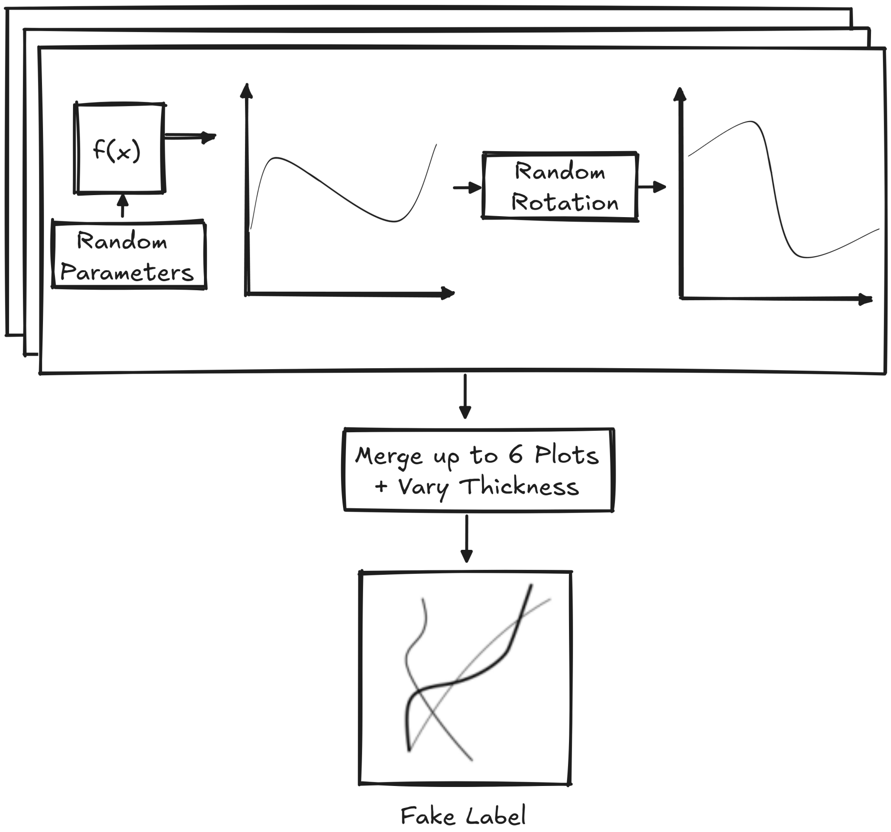


Figure 24: Heuristic to generate fake labels using the label generator (Step 2).

Table 2: Parameters for the fake label generator.

Parameter	Lower bound	Upper bound
a_1	15	30
a_2	0.02	0.03
a_3	1	50
a_4	-0.5	0.5
a_5	-0.5	0.5
a_6	-0.5	0.5
a_7	0.005	0.0095

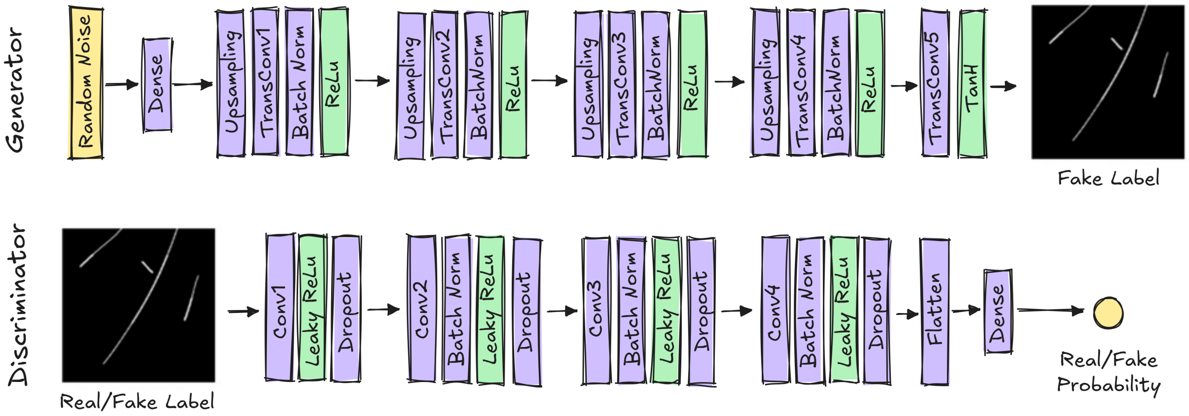


Figure 25: Architecture of both the generator and critic of the used WGAN network.

5.2.2.2 *Generative Modelling of Defects*

The fact that the approach of mathematically modeling label mask structures is coupled to a lot of engineering overhead led to the investigation of a more generic approach, which is described in this section.

The basic idea of this method is to use the capability of GANs to transform random noise vectors into data that looks similar to data of a given training set. While the pix2pix network that was described earlier in this work performs a transformation between different image domains, more traditional GANs are designed to generate completely new data. This property was used for data augmentation tasks in the past, not only in the image domain (Bowles et al., 2018), but also for audio classification tasks (Mertes. et al., 2020). However, instead of generating new image data of carbon fiber defect images, our approach uses such a rather traditional GAN to create new label segmentation masks, which then can be transformed to defect image data by feeding it to our pix2pix network, as will be described in

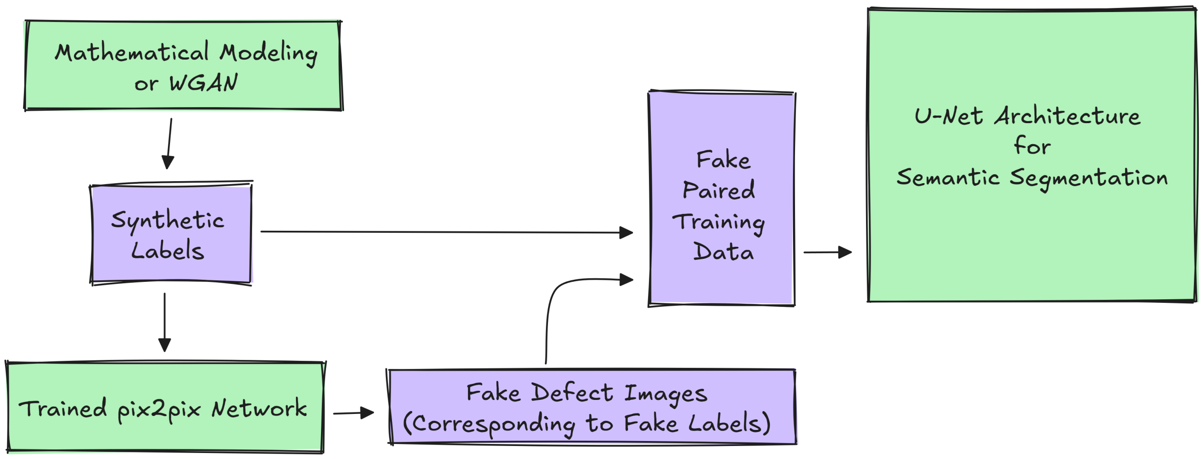

Mathematical Modeling or WGAN
Synthetic Labels
Trained pix2pix Network
Fake Defect Images (Corresponding to Fake Labels)
Fake Paired Training Data
U-Net Architecture for Semantic Segmentation

Figure 26: The generation and preparation of training data for U-Net using a trained *pix2pix* model and the fake label generator to create fake training pairs (Step 3).

the next section. To generate artificial label mask images, we made use of a convolutional GAN that operates on the *Wasserstein-Loss* as introduced by Arjovsky, Chintala, and Bottou (2017). The network architecture of both the generator and the critic is illustrated in Fig. 25. As a training dataset, we used real label masks. More specifically, the same label masks were used that were already part of the training pairs of the *pix2pix* network. Details regarding the training configuration can be found in Section 5.3.5.

5.2.3 *Finalizing the Training Data*

In the last step, the generated label data was used to create new corresponding image data. Thus, the label data that was produced by either mathematical modelling or by the WGAN was used as input to the trained *pix2pix* model. As such, the resulting data pairs of label/image data can be used to train further networks for the actual segmentation task. The whole system is shown in Fig. 26. For our experiments, we chose a U-Net architecture to perform this segmentation task. It has to be emphasized that the selection of this specific network was done for the purpose of evaluating and comparing our augmentation approaches, and that we don't claim that architecture to be the best choice for the respective task. However, U-Net could achieve promising results in related fields like biomedical image segmentation (Ronneberger, P. Fischer, and Brox, 2015).

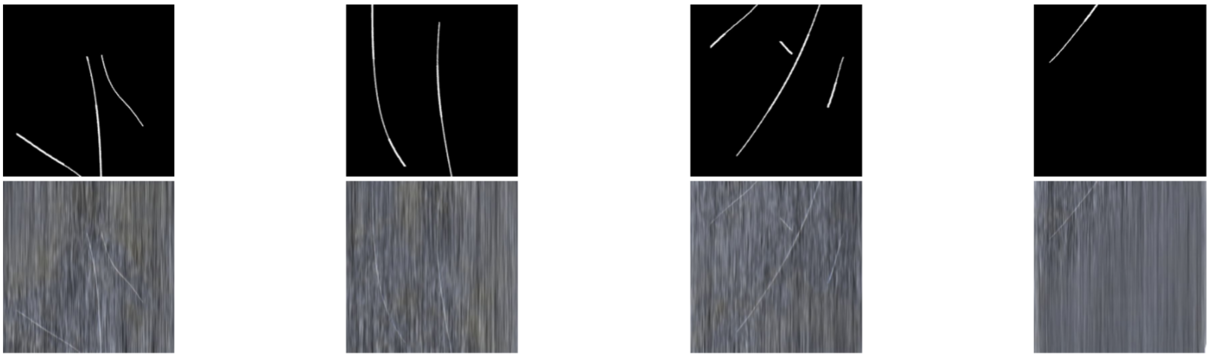


Figure 27: Samples of synthetic labels generated through handcrafted modeling (top row) and corresponding *pix2pix* outputs (bottom row) imitating misaligned fibers.

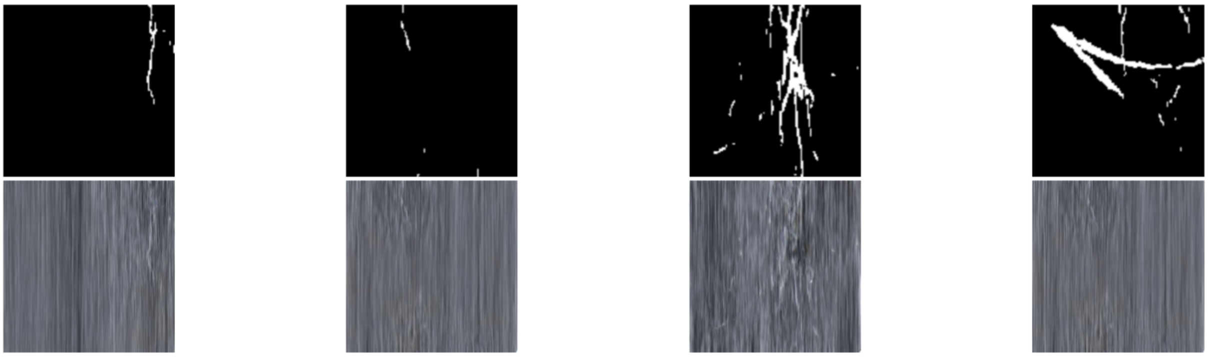


Figure 28: Samples of synthetic labels generated by WGAN (top row) and corresponding outputs (bottom row) using the same *pix2pix* model as for Fig. 27.

5.3 EXPERIMENTS

5.3.1 *Dataset Specifics*

Our system was evaluated in the context of an industrial application scenario. More precisely, the domain of carbon fiber defect monitoring was chosen for testing and evaluating of the proposed approaches. In images of fiber structures without recognizable defects, the single fibers are aligned in parallel and form a carpet of straight lines. During the production process, mechanical stress caused by spools in the transportation system can lead to damage of the fiber material, which usually can be recognized as misaligned fibers. The shape of those cracked fibers, as well as their position and size, vary heavily. Thus, there is no *template* for single defects. Given the different images of defective fiber material, a huge variety of defect structures can be observed. In this specific use case, we aim for the identification of defects on a carbon fiber carpet. To achieve this goal, a U-Net architecture is trained to perform a binary segmentation of pixels that contain defects. The environment and the design of the monitoring system that was used to acquire the image data for our experiments has been described in earlier publications (Geinitz, Wedel, and Margraf, 2016; Geinitz, Margraf, et al., 2016; Margraf et al., 2017).

5.3.2 *Experimental Setup*

For a meaningful evaluation, we ran several experiments to compare the two variants of our approach with conventional data augmentation methods. Thus, parts of our datasets that are described below were augmented with traditional data augmentation techniques. The following *simple* image transformations were applied to those artificially extended datasets:

- Randomised crop of squares of different size
- Horizontal and vertical flip
- Rotation
- Elastic transformation
- Grid distortion

We arranged the image data in six different sets and performed multiple trainings of a U-Net architecture. Then, we used the resulting models to make predictions on a test set. To ensure comparability, the same test set was used for every training set. Every training pair for the U-Net architecture consists of a real or fake defect image and a real or fake binary label image:

DATASET 1 contains 300 pairs of real defect data and corresponding binary label images. Thus, only original data without data augmentation was taken.

DATASET 2 contains the same 300 pairs of defect data and corresponding labels as *dataset 1*. Additionally, conventional data augmentation was applied as described above. For each image, some of those aforementioned transformation operations were performed with a predefined probability.

DATASET 3 contains 3000 pairs of defect data and corresponding labels. 2700 of the 3000 data pairs were generated by applying the *pix2pix* based data augmentation approach on *dataset 1*. For the creation of synthetic labels, our mathematical model was applied. Furthermore, the 300 original data pairs from *dataset 1* were taken.

DATASET 4 contains the same 3000 pairs of defect data and corresponding labels as *dataset 3*. Additionally, the same conventional stochastic data augmentation as for *dataset 2* was applied, i.e., each image was transformed with a predefined probability during training. Thus, *dataset 4* combines common data augmentation with our approach.

DATASET 5 contains 3000 pairs of defect data and corresponding labels. In this dataset, 2700 of the 3000 pairs were generated by the *pix2pix* network. This time, however, the input data for image-to-image-translation was not generated from a handcrafted function, but by training a *WGAN* model on image pairs of real sample data. The resulting model was then used to create new binary labels. The remaining 300 image pairs were taken from *dataset 1* as performed in the previous datasets.

DATASET 6 contains the same 3000 pairs of defect data and corresponding labels as *dataset 5*. Hereby, though, we altered the training of the U-Net by adding traditional data augmentation as for *dataset 4* and *dataset 2*. In this dataset, 2700 of the 3000 pairs were generated by the *pix2xpix* network. This was performed to examine how *regular* data augmentation will change the result on top of WGAN based label generation and image-to-image translation.

Each of the datasets was used to train a separate U-Net model for semantic segmentation. For testing and evaluation, a single, distinct dataset was used, containing real defect data and annotations that were provided by domain experts.

5.3.3 *Pix2Pix Configuration*

The configuration of the *pix2pix* model is given in Tab. 3. We stopped the training after 3200 epochs, as we could not observe any further improvement of the generated images by that time. Fig. 27 shows a selection of pairs of labels and images generated through application of the *pix2pix* model, where the labels where created by mathematical modeling. Fig. 28 shows label/image pairs where the labels were generated by our WGAN approach.

Table 3: Pix2Pix Configuration

Parameter	Value
Learning rate	0.0005
Batch Size	1
Epochs	3200
Loss Function	Mean Squared/Absolute Error

5.3.4 *U-Net Configuration*

The U-Net architecture was trained individually for every dataset. As described above, dataset 2, 4 and 6 were augmented with traditional data augmentation, i.e., conventional image transforms.

A stochastic component was added to the image transformations, i.e., all operations were performed with a given probability.

The randomized crop was given the probability $p = 0.25$ and a window size interval of $[400, 512]$ pixels. Furthermore, the probabilities for flipping, rotation, elastic transform and grid distortion were set to $p = 0.5$.

The U-Net model itself was slightly adapted to fit the dataset. The default size of the training images was 512x512, yet the default U-Net setting only accepts images of size 28x28. A ResNet-18 model is used as encoder by the U-Net. The architecture was adapted to fit the input size before applying the model. The training configuration of the U-Net is shown in Tab. 4.

Table 4: U-Net Configuration

Parameter	Value
Learning rate	0.0001
Batch Size	10
Epochs	200
Loss Function	Binary Cross Entropy / Dice Loss

5.3.5 *WGAN Configuration*

The configuration for the WGAN training is shown in Table 5. As the WGAN produces non-binary image data as output, we applied a binarization stage to the final output images in order to get binary label masks.

Table 5: WGAN Configuration

Parameter	Value
Learning rate	0.00005
Batch Size	64
Epochs	100,000

5.4 RESULTS

The metrics $accuracy$, MCC and $F_\beta - Score$ are less dependable for an objective evaluation of the segmentation model in our specific task. The proportion between background pixels (i.e., the non-defect pixels) and foreground pixels (i.e., the defect pixels) per image is thoroughly unbalanced, as the defects mostly consist of single fibers and therefore take much less space in the images. While $accuracy$ returns the proportion of true results among all data points examined, MCC and $F_\beta - Score$ aim to balance out true and false positives and negatives of the binary classification result. In contrast, the *Jaccard index* or *Intersection over Union (IoU)* is used to measure the similarity of two sets, i. e., the similarity of the ground truth and the prediction. This property makes the IoU the most appropriate for the task at hand. Thus, we focus on the IoU metric for our experiments in order to allow an objective and problem related evaluation methodology. However, all relevant scores are reported in Tab. 6 for the sake of completeness.

For all of the six datasets, the training was aborted after 200 epochs since it was observable that the models had converged. Respective loss graphs can be seen in Figure 29 and Figure 30

All training results are shown in Tab. 6.

The trained models were all evaluated on the test dataset. The model trained on dataset 1 reached an $accuracy$ of 0.985 and an IoU of 0.391 on the test set, while the model trained with dataset 2 reached an $accuracy$ of 0.992 and an IoU of 0.593. Furthermore, the IoU for the model based on dataset 3 reached an IoU of 0.579 and an $accuracy$ of 0.991, while training with dataset 4 achieved a value of 0.575 for the IoU and 0.991 for the $accuracy$. In addition, dataset 5 resulted in an IoU of 0.423 and $accuracy$ of 0.991, while training on dataset 6 let the IoU drop to 0.206 and $accuracy$ decrease to 0.980.

Table 6: U-Net results from test runs on the datasets 1 through 6 for batch size 5.

	PPV	TPR	IoU	ACC	MCC	F1	F2
Dataset 1	0.539169	0.586753	0.390778	0.985035	0.55487	0.561956	0.576576
Dataset 2	0.772803	0.718101	0.592925	0.991935	0.740872	0.744448	0.728413
Dataset 3	0.745926	0.721067	0.578888	0.991419	0.729034	0.733286	0.725905
Dataset 4	0.756767	0.705387	0.57502	0.991471	0.72631	0.730175	0.715098
Dataset 5	0.602418	0.585941	0.422541	0.990628	0.594065	0.543877	0.589383
Dataset 6	0.281570	0.433361	0.205801	0.980426	0.339847	0.341352	0.354881

Fig. 31 shows a sample selection of defect images taken from the test set with red overlays representing the Regions of Interest (ROIs), i.e., the regions that were predicted to contain defect areas, predicted by the U-Net model.

5.5 DISCUSSION

As can be seen, the U-Net model trained on dataset 3 substantially outperformed the model trained on dataset 1. This shows that our approach of mathematical defect modeling in combination with a *pix2pix* architecture could substantially improve the quality and diversity of the raw training set. When comparing the results of dataset 2 (conventional data augmentation) and dataset 3, it becomes apparent that our approach is slightly worse, however not substantially diverts from conventional data augmentation techniques as applied on dataset 2. The difference comprises within less then 0.02 for the *IoU*.

The combination of generating synthetic data using that first approach with subsequent conventional data augmentation as for dataset 4 did not lead to any improvement. The model trained on dataset 4 outperforms dataset 1, but leads to a slighty lower *IoU*, *accuracy* and *MCC* than dataset 2 and 3. However, the degradation ranges within less then 0.02 for the *IoU* and is therefore not substantial under the given circumstances. The last two rows of Tab. 6 show the results from the experiments that apply image-to-image translation on labels generated with the WGAN model. As can be seen, dataset 5 slightly outperforms dataset 1 since it results in a higher *IoU*, *accuracy* and *MCC*. However, it clearly proves inferior to datasets 2 through 4 which means regular data augmentation as well as the problem-tailored label generator clearly performs better in the given scenario. For the sake of completeness and transparency, we also did an experiment in which we extended the data of *dataset 5* with conventional data augmentation, resulting in *dataset 6*. The outcome,

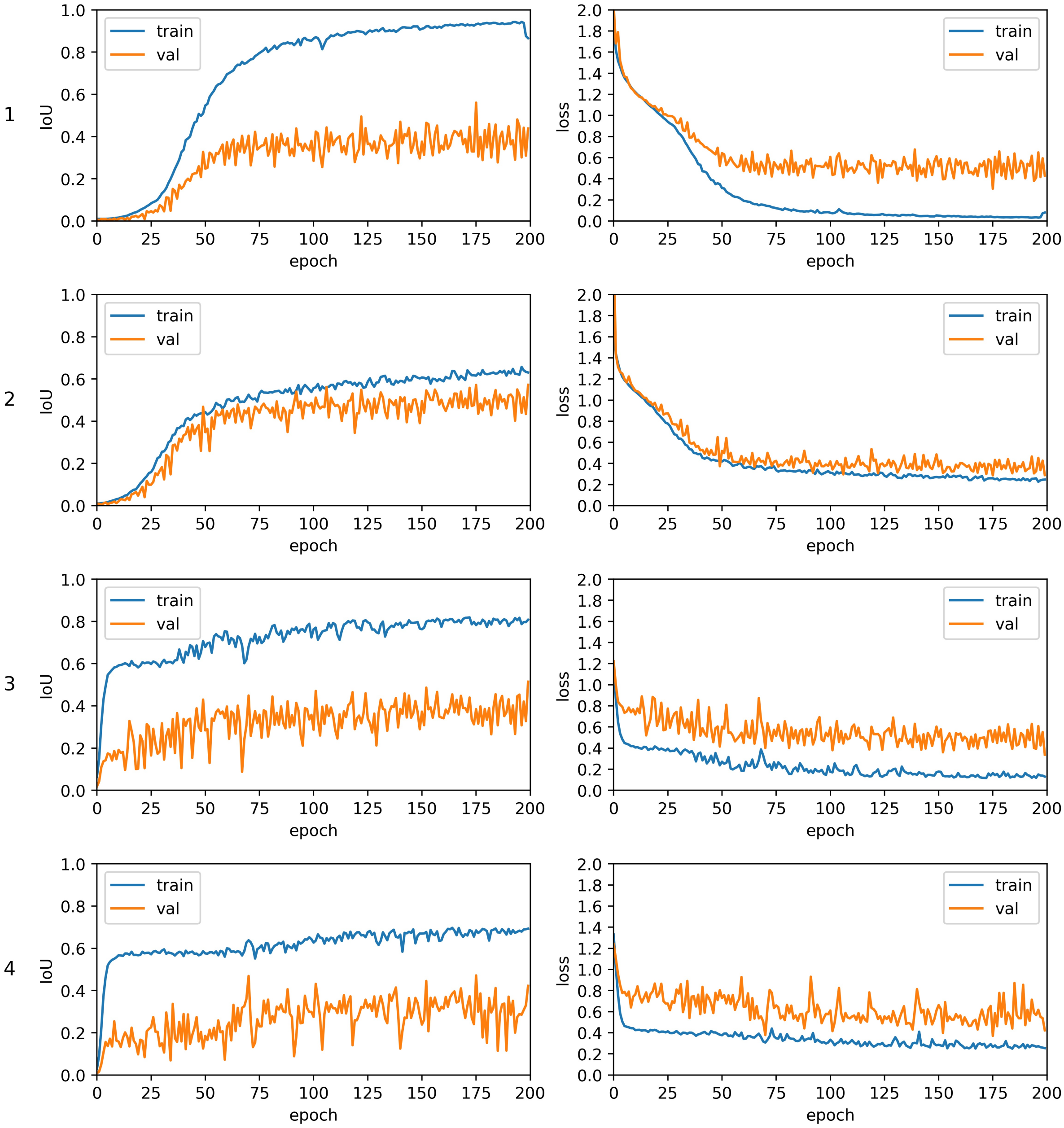


Figure 29: IoU Score and Loss during U-Net training for dataset 1 through 4 from top to bottom row.

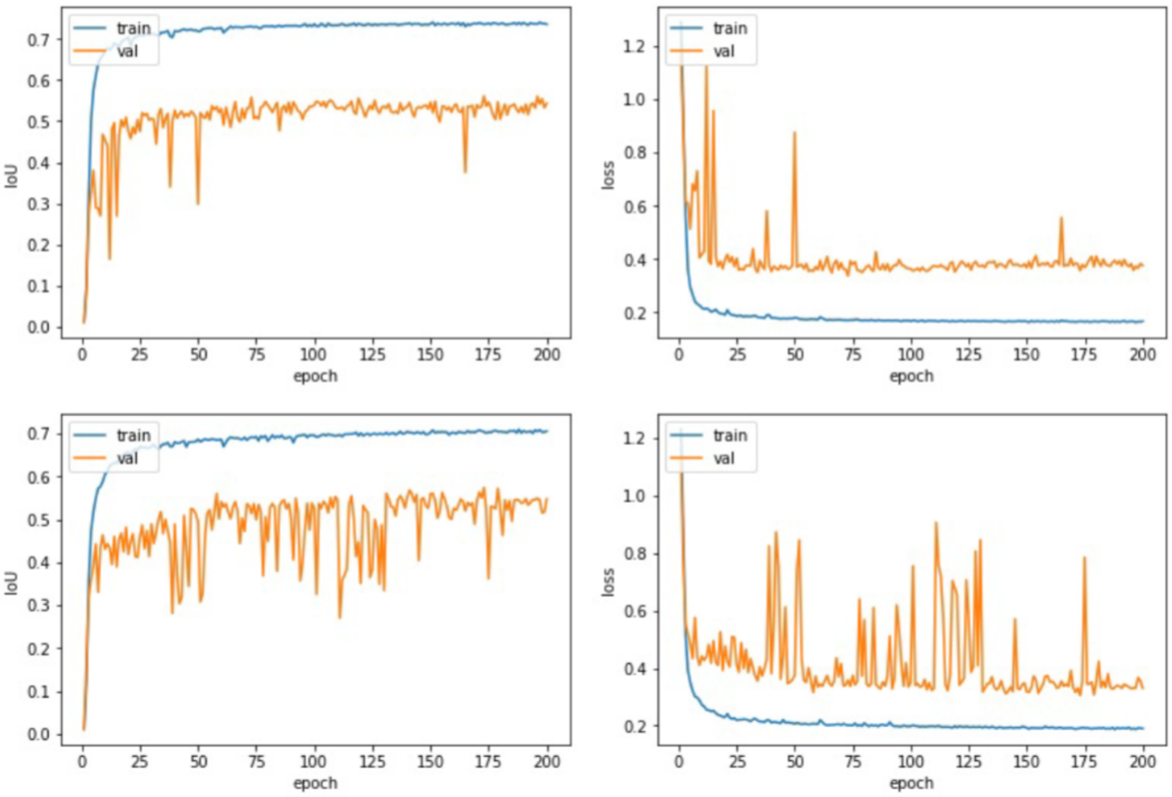


Figure 30: IoU Score and Loss during U-Net training for dataset 5 and 6 from top to bottom row.

though, indicates a clear deterioration all along the line. Every metric appears lower than for any other dataset (i. e., datasets 1 - 5 perform better).

As the results from Tab. 6 and the samples depicted in Fig. 31 suggest, the pairs of synthetic images and labels of carbon fiber defects were successfully used to replace traditional data augmentation for semantic segmentation network training - but they were not *better* than traditional data augmentation.

At first glance, it might look like our approach does not add anything to traditional data augmentation. However, we suggest to keep in mind that this study has only applied the workflow on images of carbon fiber surfaces with its very special types of defects and anomalies. It is conceivable that designing more complex and thought-through models for the defect modeling might yield even better results. At this point, it cannot be ruled out that the workflow functions differently under other circumstances that are considerably distinct to the present use case.

We consider both our *pix2pix* based image generation approaches a more realistic and application-oriented form of data augmentation. The experiments were conducted with and without traditional data augmentation in order to evaluate the effectiveness and expandability of our approach. Excessive use of traditional data augmentation

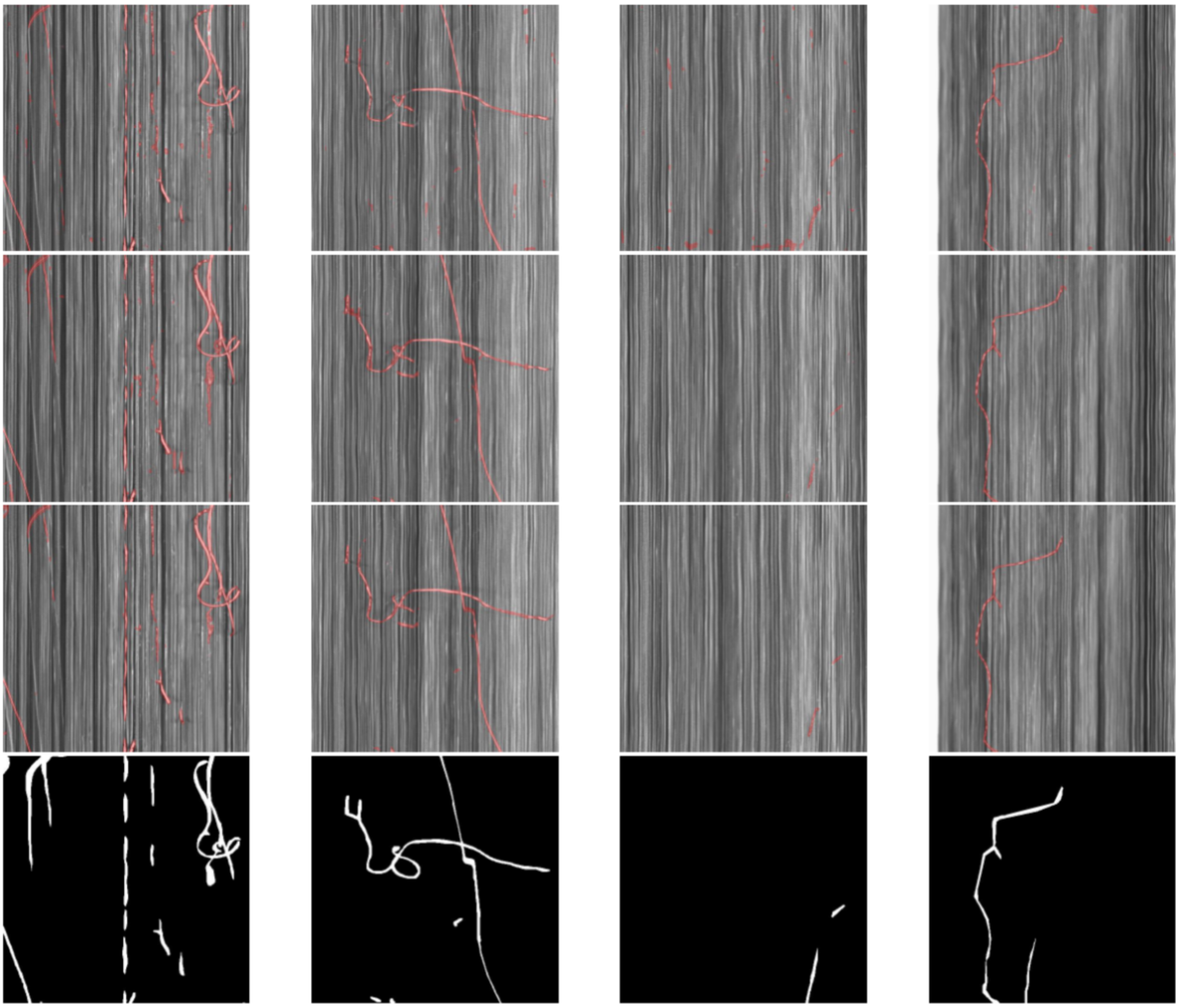


Figure 31: Real carbon fiber defects from the test set with red overlay from U-Net segmentation for *dataset 1* (top row), *dataset 3* (second row), *dataset 5* (third row) and the *ground truth* (bottom row).

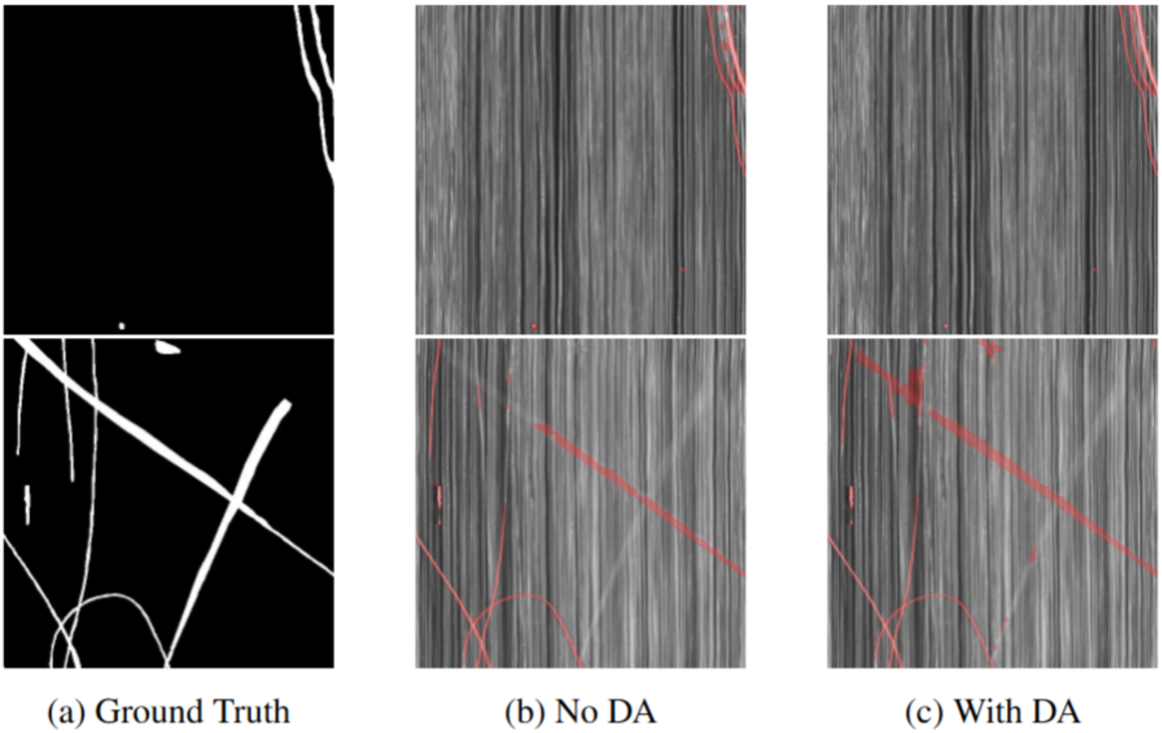


Figure 32: Comparison of noticable side effects in the classification of filaments based on the WGAN based approach with (*With DA*) and without (*No DA*) data augmentation and the corresponding ground truth.

might superimpose the 'real' data within the training set due to its low level form of manipulation, reproduction and reuse which raises the risk of overfitting during model training. GAN based data augmentation might be less prone to repetitive patterns since it tries to project the variation found in the original data to the synthetic data.

In summary, the approach as proposed in this chapter reveals a potential alternative to traditional, simple data augmentation. The generated data forms a representation of images that is substantially different from the sample data but still resembles the training domain. Furthermore, the suggested algorithms support training deep learning models for semantic segmentation with small sample datasets. This also applies if only few annotations are available.

In this chapter, two competing concepts were suggested and evaluated on industry data. While the first approach based on a handcrafted function represents a very problem-specific, handcrafted solution, the second concept entirely uses adversarial and deep learning for model training.

On the one hand, the approach based on WGAN, Pix2Pix and U-Net combines three different deep learning architectures and only requires parameter tuning to work for the given dataset. This workflow already offers a high level of automation since it reduces the effort for designing a defect detection system to providing a small sample set of annotated data. Of course, only a well-selected and sufficiently annotated test set allows for serious model validation and testing. On the other hand, the handcrafted function is transparent, human-readable and stable. Its output can be visualized and tested against tolerance criteria. It also allows to be tuned by setting limiting parameters, such as window size, orientation or line thickness. Also, auditing and requirements testing can be easily performed. However, the mathematical function lacks of flexibility in terms of domain transfer. In order to design a mathematical model specific to the problem, domain knowledge needs to be collected and translated into abstract dependencies. Thus, it qualifies for applications with a high need for transparency and stability, e. g., security in critical environments.

5.6 conclusion

In this chapter, we presented image-to-image translation as a means for data augmentation in the context of defect detection on textiles and carbon fiber in particular. Therefore, we discussed related GAN approaches and designed two variations of a novel concept for generating synthetic defects based on sparse labeled data using a *pix2pix* model.

Within our experiments on six different datasets we showed that a *pix2pix* based approach could substantially improve the pixel-based classification quality of U-Net models when using a problem-specific label generator compared to using no data augmentation at all. In general, the synthetic defects helped to augment the dataset so that segmentation quality improves on sparse data. However, the approach did not outperform regular data augmentation techniques.

Although WGAN proved inferior to competing techniques, it still helped to support semantic segmentation to a certain degree.

The suggested approach has been tested for the given setting but is not limited to textiles.

In conclusion, we could show the potential of our novel data augmentation technique, although it is still not mature enough to completely replace traditional data augmentation.

Part III

EXPLAINABILITY

With the rapid development of Machine Learning (ML) methods, black-box models powered by complex ML algorithms are increasingly making their way into high risk applications, such as healthcare (Stone et al., 2016). Systems used here must not only work, but also need to provide comprehensible and transparent information about their decisions. To support more transparent Artificial Intelligence (AI) applications, approaches for Explainable Artificial Intelligence (XAI) are an ongoing topic of high interest (Arrieta et al., 2020). A recent trend in XAI is to work with explanations that are based on *Counterfactual Reasoning*, i.e., explanations that show how an AI system would react to an alternative input. The most prominent example of such explanations are *Counterfactual Explanations*. They communicate information about relevant features by showing a modified version of the input that leads to a different decision of the AI to be explained. Creating such explanations is a difficult task - especially in the image domain, where explanations should still have high quality to appear reasonable. Here, GANs can help - not only due to their capabilities of generating high-quality data, but also as they are able to create data that still looks *consistent*. As such, in this part of the thesis, after a short overview on traditional XAI techniques and factual explanations is given in Chapter 6, we will present how GANs can be used to generate high-quality counterfactual explanations for image classifiers in Chapter 7. In some cases, it might not even be sufficient to communicate information about *relevant* features - *irrelevant* features might be similarly important for an AI's understanding. However, approaches that explicitly address such irrelevant features do not yet exist. Therefore, in Chapter 8, we introduce such a paradigm that we call *Alterfactual Explanations*. As that concept is completely new, we also present a user study to evaluate if such explanations can actually complement counterfactual explanations. After having validated that the *concept itself* is promising, we also introduce a technical framework to *generate* those explanations (see Chapter 9). Again, we show that GANs are perfectly suited for generating such explanations for an image classification task.

6

TRADITIONAL XAI TECHNIQUES

Some parts of this chapter have already been published in the following publications:

> *Mertes, S., Karle, C., Huber, T., Weitz, K., Schlagowski, R., & André, E. (2022). Alterfactual Explanations - The Relevance of Irrelevance for Explaining AI Systems. In IJCAI 2022 Workshop on XAI.* (Mertes, Karle, et al., 2022)

> *Mertes, S., Huber, T., Karle, C., Weitz, K., Schlagowski, R., Conati, C., & André, E. (2024). Relevant Irrelevance: Generating Alterfactual Explanations for Image Classifiers. In 33rd International Joint Conference on Artificial Intelligence (IJCAI) 2024.* (Mertes, Huber, Karle, et al., 2024)

Before introducing our new concepts and technical frameworks, first, a very brief overview over existing XAI approaches is given. Note that this overview only takes into account approaches that were used in the following chapters. For a more detailed overview, we would like to point the user to more comprehensive surveys, e.g., Tjoa and Guan (2020) or Das and Rad (2020).

6.1 FEATURE ATTRIBUTION

Feature Attribution refers to the concept of communicating *which* features are relevant for decisions - and often also *how* important they are.

A frequently-used representative for an XAI approach based on feature attribution is *Local Interpretable Model-agnostic Explanations* (LIME) (Ribeiro, Singh, and Guestrin, 2016). The basic idea of LIME is to approximate an interpretable model around the original model. As a consequence, it is possible to create explanations for various machine learning domains like text and image classification. Depending on the model to be explained, the explanations come in the form of textual or visual feedback. In the case of image classification, LIME is highlighting whole areas in the image that have been crucial for the prediction of a specific class.

Other mechanisms aim to produce so-called *Saliency Maps*, i.e., heatmaps that show how much each pixel of an input image contributed to a certain decision. Examples for such mechanisms that are particularly used in the image domain are *Layer-wise Relevance Propagation* (LRP) (Bach et al., 2015) or GradCAM (Selvaraju et al., 2020). There, LRP assigns a relevance value to each neuron in a neural network, measuring how relevant this neuron was for a particular prediction. In order to do so, LRP defines different rules, all of which are based on the intermediate outputs of the neural network during a forward pass. On the other hand, GradCAM focuses on the feature maps of a convolutional neural network's final convolutional layer. Therefore, gradients flowing into that layer are computed and visualized.

Note that besides those few mentioned approaches, there is a huge amount of other feature attribution mechanisms in the field of XAI, e.g., DeepLIFT (Shrikumar, Greenside, and Kundaje, 2017), SHAP (Lundberg and S.-I. Lee, 2017), or SmoothGrad (Smilkov et al., 2017). However, what all of them have in common is that they - in one way or the other - try to communicate *which* features of an input contribute to a model's decision *to what degree*.

6.2 COUNTERFACTUAL REASONING

In contrast to mechanisms from the field of Feature Attribution (i.e., those mechanisms that try to communicate *which features* are important for a decision), the paradigm of *Counterfactual Reasoning* refers to communicating how decisions could have turned out if *another input* would have been given. Figure 33 illustrates the difference between those concepts using exemplary explanations for a fictional AI that decides if a person is creditworthy or not. We will use that scenario as a running example of how the different explanation paradigms would answer the question of *"Why does the AI say that I am not creditworthy?"*.

6.2.1 *Factual Explanations*

> *"There was another female person that also had rather little money, and she also did not get the credit."*

Factual explanations are the traditional way of explaining by example, and often provide a similar instance from the underlying data set (adapted or not) for the input data point that is to be explained (Keane, Kenny, Temraz, et al., 2021). Other approaches do not choose an instance from the dataset, but generate new ones (Guidotti et al.,

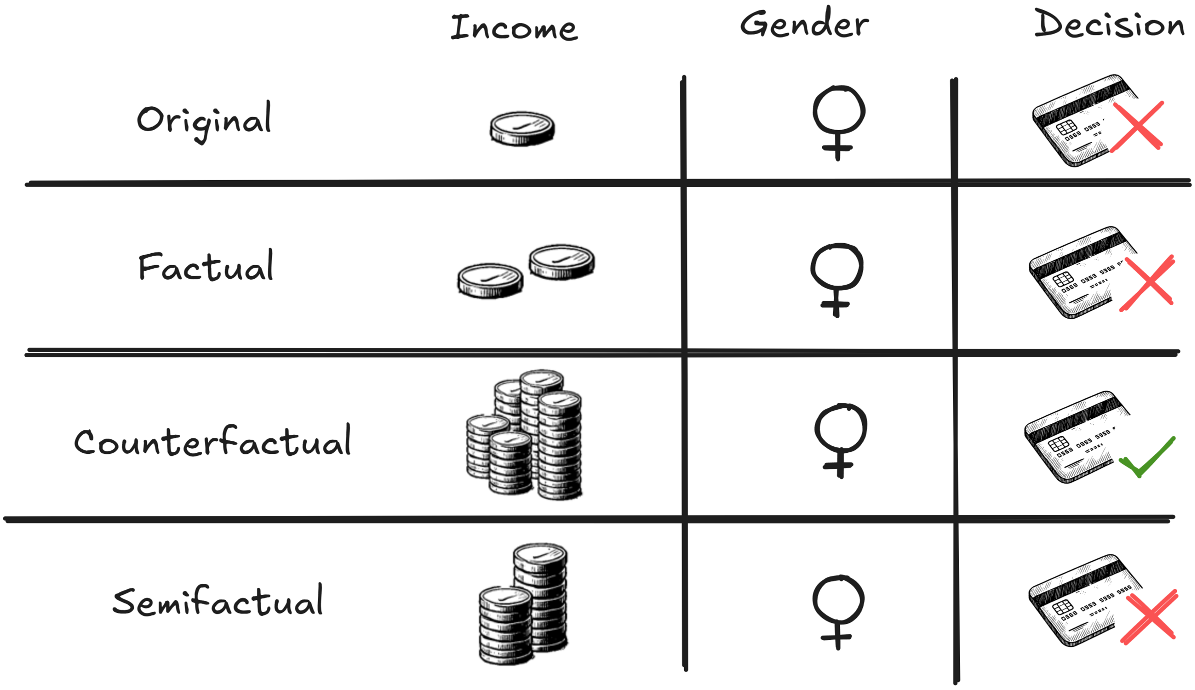


Figure 33: Explanation types that follow the principles of Counterfactual Reasoning. Input features to a fictional decision system to be explained are *Income* and *Gender*, whereas the former is relevant and the latter is irrelevant to the AI's decision on whether a credit is given or not.

2019). The idea behind factual explanations is that similar data instances lead to similar decisions, and the awareness of those similarities leads to a better understanding of the model. Further explanation mechanism that fall in this category are *Prototypical Explanations* and *Near Hits* (B. Kim, Khanna, and Koyejo, 2016; Herchenbach et al., 2022).

6.2.2 *Counterfactual Explanations*

> *"If you had that amount of money, you would get the credit."*

Counterfactual explanations are a common method humans naturally use when attempting to explain something and answer the question of *Why not ...?* (Miller, 2019; R. M. J. Byrne, 2019). In XAI, they do this by showing a modified version of an input to an AI system that results in a different decision than the original input. Counterfactual explanations should be minimal, which means they should change as little as possible in the original input (Keane, Kenny, Temraz, et al., 2021; Miller, 2021). Many researchers have emphasized that counterfactual explanations should be actionable and feasible, i.e., should provide a user with an example that is achievable and realistic in real life (Barocas, Selbst, and Raghavan, 2020; Ustun, Spangher, and Y. Liu, 2019). How to achieve this is an ongoing research topic, with open questions for example whether feasibility can be achieved by adhering to data distributions found in the training set (Laugel et al., 2019; Mahajan, C. Tan, and Sharma, 2019; Keane, Kenny, Delaney, et al., 2021). Wachter, Mittelstadt, and C. Russell (2017) name multiple advantages of counterfactual explanations, such as being able to detect biases in a model, providing insight without attempting to explain the complicated inner state of the model, and often being efficient to compute. For counterfactual explanations, a multitude of works exist that use GANs to automatically generate explanations for image classifiers (Nemirovsky et al., 2022; Van Looveren, Klaise, et al., 2021; Khorram and Fuxin, 2022).

6.2.3 *Semifactual Explanations*

> *"Even if you had that amount of money, you would still not get the credit."*

Similar to counterfactual explanations, semifactual explanations are an explanation type humans commonly use. They follow the pattern of *Even if X, still P.*, which means that even if the input was changed in a certain way, the prediction of the model would still not change to the foil (McCloy and R. M. Byrne, 2002). In an XAI context, this means that an example, based on the original input, is provided that modifies the input in such a way that moves it toward the decision boundary of the model, but stops just before crossing it (Kenny and Keane, 2020). Similar to counterfactual explanations, semifactual explanations can be used to guide a user's future action, possibly in a way to deter them from moving toward the decision boundary (Keane, Kenny, Temraz, et al., 2021).

7

COUNTERFACTUAL EXPLANATION GENERATION

Large parts of this chapter have already been published in the following publication:

> *Mertes, S., Huber, T., Weitz, K., Heimerl, A., & André, E. (2022). Ganterfactual —Counterfactual Explanations for Medical Non-Experts using Generative Adversarial Learning. In Frontiers in Artificial Intelligence, 5, 825565.* (Mertes, Huber, Weitz, et al., 2022)

Counterfactual explanations try to help to understand why the actual decision was made instead of another one by creating a slightly modified version of the input which results in another decision of the AI (Wachter, Mittelstadt, and C. Russell, 2017; R. M. J. Byrne, 2019). As they alter the original input, they directly show *how* the input could have looked like, such that another decision would have been made, instead of only showing *where* a modification of the input would make a difference in the classifiers outcome. Creating such a slightly modified input that changes the model's prediction is by no means a trivial task. In the visual domain, current counterfactual explanations often utilize images from the training data as basis for modified input images. This often leads to counterfactual images that are either distinct but similar images from the training data, or that are unrealistically modified versions of the input image. Humans, however, prefer counterfactuals that modify as little as necessary and are rooted in reality (R. M. J. Byrne, 2019). In this chapter, we present a novel counterfactual explanation approach that aims to tackle these current challenges by utilizing adversarial image-to-image translation techniques. Traditional generative adversarial networks for image-to-image translation do not take a model's decision into account and are therefore not suited for counterfactual generation. To this end, we propose to include the classifier's decision into the objective function of the generative networks. This allows for the creation of counterfactual explanations in a highly automated way, without the need for heavy engineering when adapting the system to different use cases. Further, although our approach is steered by a classifier model's *decision*, it is independent of that model's specific *architecture* - it is *model-agnostic*.

We evaluate our approach by a computational evaluation and a user study. Specifically, we use our system to create counterfactual explanations for a classifier that was trained on a classification task to predict if x-ray images of the human upper body are showing lungs that are suffering from pneumonia or not. In addition to being a highly relevant application for explanations, this scenario is suitable for evaluating explanations for non-experts since they are not expected to have in-depth knowledge of that domain, i.e., they are completely reliant on the explanation that the XAI system gives in order to follow the AI's decisions. Furthermore, pneumonia in x-ray images predominantly is reflected by opacity in the shown lungs. Opacity is a textural information that can not be explained sufficiently enough by the spatial information provided by common saliency map approaches. To validate our assumptions, we compare the performance of our approach against two established saliency map methods, namely *Local Interpretable Model-agnostic Explanations* (LIME) and *Layer-wise Relevance Propagation* (LRP).
With our work we make the following contributions:

- We present a novel approach for generating counterfactual explanations for image classifiers and evaluate it computationally.
- We evaluate our approach in a user study and gain insights in the applicability of counterfactual explanations for non-ML experts in an exemplary medical context.
- We compare counterfactual explanations against two state-of-the-art explanation systems that use saliency maps.

7.1 RELATED WORK

Counterfactual explanations describe an alternative reality that is contrastive towards the observed one (Molnar, 2019). This approach of generating explanations is in line with how humans explain things. Humans rarely ask why something happened, but rather why the current outcome is present instead of a different one (Miller, 2019). This similarity is one of the advantages over approaches that focus on feature importance.

7.1.1 *Generating Counterfactual Explanations*

Various approaches to generate counterfactual explanations have emerged. The first to introduce counterfactual explanations have been Wachter, Mittelstadt, and C. Russell (2017). They formulated the computation of counterfactuals as an optimization problem. Their goal was to identify a counterfactual that is the closest to the original input, by minimizing the distance between the input data and a potential counterfactual. One of the first technical approaches on generating counterfactual explanations - that is still widely used - was *DICE* (Mothilal, Sharma, and C. Tan, 2020). However, DICE was designed to work with numerical data, while in this chapter we focus on image data.

7.1.2 *Counterfactual Explanations for the Image Domain*

For the image domain, Van Looveren and Klaise (2019) proposed a model-agnostic approach to generate counterfactual explanations by using class prototypes to improve the search for interpretable counterfactuals. They evaluated their approach on the MNIST dataset, as well as the Breast Cancer Wisconsin (Diagnostic) dataset. Goyal et al. (2019) present an approach to create counterfactual explanations for an image classification task. They exchange a patch of the original image with a patch from a similar image from the training dataset which gets classified differently. They evaluated their approach on four different datasets, including MNIST, SHAPES, Omniglot and Caltech-UCSD Birds (CUB). Both the approaches, as they are not making use of state-of-the-art generative models, produce outcomes of rather low quality. Also, due to their specific counterfactual search procedures, it is not guaranteed that the counterfactuals only change the input images as much as needed.

7.1.3 *GAN-based Counterfactual Explanations*

Matthew L Olson et al. (2019) use a combination of a GAN and a Wasserstein Autoencoder to create counterfactual states to explain Deep Reinforcement Learning algorithms for Atari games. However, their approach is not applicable to classification or regression problems.

Nemirovsky et al. (2022) proposed *CounterGAN*, an architecture in which a generator learns to produce residuals that result in counterfactual images when added to an input image. However, their approach was only tested for low-dimensional image data. Also the resulting explanations were only computationally evaluated. W. Zhao, Oyama, and Kurihara (2021) propose an approach for generating counterfactual image explanations by using text descriptions of relevant features of an image to be explained. Those text descriptions are then analyzed regarding features that are not present in the counterfactual class. A counterfactual text description is built, which is subsequently transformed into a counterfactual image by using a text-to-image GAN architecture. However, the text descriptions have to be defined a priori, resulting in a lot of manual overhead.

7.1.4 *Counterfactual Explanations based on Style Transfer*

GAN architectures from the field of *Style Transfer* or *Image-to-Image Translation* - as explained in detail in Section 2.9 - enable the transformation of images between different image domains. There is existing work that uses those techniques of adversarial image-to-image translation for creating counterfactuals, but often the counterfactuals are not created for the purpose of explaining ML algorithms, but rather to improve such models by augmenting training datasets. For example, Neal et al. (2018) presented an algorithm to generate counterfactual images in order to augment data for an open set learning task, i.e., a task where not all classes are known during the training stage. C.-r. Wang et al. (2020) published an approach to create counterfactual images of breast images to improve the task of mammogram classification. To this end, they make use of the observation that healthy human breasts look symmetrical, allowing for a projection of a healthy breast to an unhealthy breast of the same person. While their results in theory could also be used as counterfactual explanations, their generation algorithm inherently relies on the symmetry of body parts, strongly limiting the generalization capabilities of their approach. Y. Zhao (2020) proposed to use a StarGAN (Y. Choi et al., 2018) architecture to create counterfactual explanation images. However, the system was only applied on binary images, i.e., images where each pixel is either black or white. The resulting counterfactuals were used to highlight the pixels which differ between original and counterfactual images.

All in all, our approach is the first that *automatically* generates counterfactual images that are of *high quality* and are meant to *explain* a classifier from the image domain. Additionally, we are the first to conduct an extensive user study to compare such an image counterfactual explanation mechanism to traditional feature attribution mechanisms.

7.2 APPROACH

In the following sections, we present a novel approach for generating counterfactual explanations for image classifiers using generative adversarial learning that addresses the problems that remain in existing work: the approach generates *realistic* counterfactual explanations while at the same time *taking an AI's decisions into account*.

7.2.1 *Counterfactual Explanations as an Image-to-Image Translation Problem*

As discussed by Wachter, Mittelstadt, and C. Russell (2017), one of the key concepts of counterfactual explanations is the concept of the *closest possible world*. Counterfactual explanations aim to show a slight variation of some object, where the change between the original object and its variation results in a different outcome. Transferred to the task of explaining image classifiers, counterfactual explanations should aim to answer the following question:

> *What minimal changes to the input image would lead the classifier to make another decision?*

This question implicates two major requirements to counterfactual images:

- The counterfactual image should look as similar to the original image as possible.
- The classifier should predict the counterfactual image as belonging to another class as the original image.

Looking at the second statement at a more abstract level, the predicted class of an image can be seen as some sort of top-level feature that describes a combination of several underlying features which the classifier considers to be relevant for the classification. Thus, the generation of counterfactual images can be broken down to a transformation of certain features that are relevant for the classification, while maintaining all other features which were not relevant for the classification. However, these two objectives are also defining the problem of *Image-to-Image Translation*. The goal of image-to-image translation is to transform features that are relevant for a certain image domain to features that lead to another image domain, while all other features have to be maintained. An example of such an image-to-image translation task are style-conversion problems, where each image domain represents a certain style. In this case, translating an image from one domain to another is equivalent to changing the style of the image. Viewing the problem of counterfactual creation from the perspective of image-to-image translation inevitably leads to the idea of borrowing techniques from that area for generating counterfactual images to explain image classifiers.

7.2.2 *Image-to-Image Translation with CycleGANs*

In this section, some important contents from Section 2.9 is very briefly refreshed. If you have paid close attention there, you can jump straight on to the next section (Section 7.2.3).

As already introduced in Section 2.9, there are various approaches for solving image-to-image translation problems which rely on the use of adversarial learning. The original GANs (Goodfellow et al., 2014) approximate a function that transforms random noise vectors to images which follow the same probability distribution as a training dataset (i.e., that appear similar to images from the training set which the GAN was trained on). They do this by combining a *generator network* G and a *discriminator network* D. During training the generator learns to create new images, while the discriminator learns to distinguish between images from the training set and images that were created by the generator. Thus, the two networks are improving each other in an adversarial manner. The objective of the two networks can be defined as follows:

$$\mathcal{L}_{original}(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log (1 - D(G(z)))], \tag{12}$$

where x are instances of image-like structures and z are random noise vectors. During training, the discriminator D maximizes that objective function, while the generator G tries to minimize it.

Various modified architectures have successfully been used to replace the random input noise vectors with images from another domain. Thus, those architectures are capable of transforming images from one domain to images of another domain. These approaches are commonly described as image-to-image translation networks. Common adversarial approaches for these kind of tasks rely on paired datasets (i.e., datasets that consist of pairs of images which only differ in the features that define the difference of the two image domains). As described above, in the context of counterfactual image generation for image classifiers, the aim is to transfer images from the domain of one class to the domain of another class. The aforementioned adversarial architectures are therefore not suited for the generation of counterfactual images since they could only be applied for classifiers that are trained on paired datasets. In practice, paired datasets for image classification are a rare occasion. A solution to the problem of paired datasets was posed by Zhu et al. (2017), who introduced the *CycleGAN* architecture. This architecture is based on the idea of combining two GANs, where one GAN learns to translate images of a certain domain X to images of another domain Y, while the other GAN learns to do the exact opposite: convert images of domain Y to images of domain X. The respective objective is defined as follows:

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))] \tag{13}$$

where G is the generator of the first GAN and D_Y the discriminator of the same GAN. Therefore, that first GAN learns the translation from images of domain X to images of domain Y. The objective of the second GAN, which consists of a generator F and a discriminator D_X, is defined analogously.

By feeding images x of domain X to G and subsequently feeding the resulting image $G(x)$ to F, the output of the second GAN $F(G(x))$ can be compared with the initial input x (and vice versa) to formulate a so-called *Cycle-consistency Loss*:

$$\mathcal{L}_{cycle}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1], \tag{14}$$

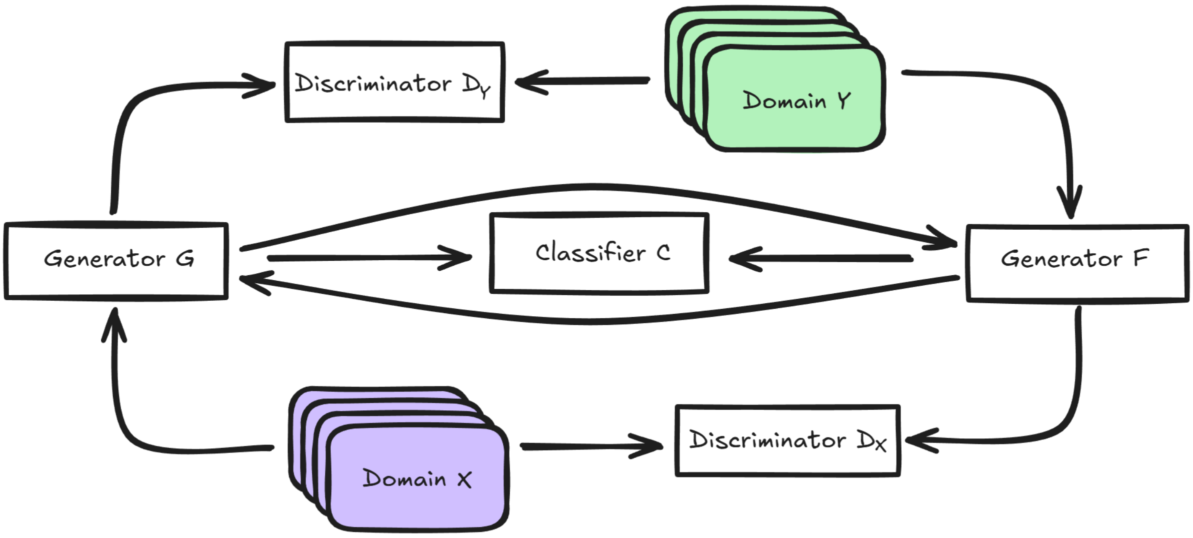


Figure 34: Schematic overview of our approach. A CycleGAN architecture is extended with the classifier that shall be explained. Both the generators of the CycleGAN include the classifier's decisions for the generated data into their loss function.

where $\|x_1\|$ represents the L1 norm. In combination with the adversarial losses given by Equation 13, the cycle-consistency loss can be minimized to solve image-to-image translation tasks that do not rely on a dataset of paired images. The full objective of such common CycleGANs is denoted as:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cycle}(G, F) \tag{15}$$

During training, the discriminators D_X and D_Y aim to maximize that objective function, while the generators G and F try to minimize it.

7.2.3 *Extending CycleGANs for Counterfactual Explanations*

Without loss of generality, we restrict ourselves to the generation of counterfactual explanations for a binary classifier (i.e., a classifier that only decides if an input image belongs to either one class or another). In theory, this can easily be extended to a multi-class classification problem by looking at each combination of classes as a separate binary problem. A naive approach to creating counterfactual images for a binary classifier would be to train a traditional CycleGAN architecture to transfer images between the two domains which are formed by the two classes of the training dataset of the classifier. This would lead to a system that is able to convert images from the domain of one class to images of the domain of the other class, while maintaining features that do not contribute to determining to which domain an image belongs to. If we now assume that the classifier, which we want to explain, is perfect and always predicts the correct class for every possible image in the two domains, then this would lead to counterfactual explanations: An input image, which was classified to belong to one of the two classes, can be fed into the trained CycleGAN to translate it into an image that is classified as the other class. However, this might not be an ideal explanation, since it is theoretically possible that the classifier does not use all the features which define a class (e.g., if some features are redundant). Moreover, the assumption of a perfect classifier is obviously wrong in the most cases. Thus, the resulting image can by no means be seen as a counterfactual *explanation* of a classifier, as the translation happens between two classes of the training dataset without considering the classifier's decision. To tackle that problem, a further constraint has to be added to the CycleGAN in order to take the actual decision of the classifier into account. To achieve this, we propose to incorporate an additional component to the CycleGAN's objective function, which we will describe below. Analogous to above, where x represented an image of domain X, let x now be an image that belongs to class X, while y belongs to class Y. Furthermore, consider a classifier C that for every input image img predicts either $C(img) = X$ or $C(img) = Y$. In this case, a *perfect* classifier would fulfill both of the following statements:

$$\forall x \in X : C(x) = X \quad \text{and} \quad \forall y \in Y : C(y) = Y \tag{16}$$

As of the objective functions that are used for the definition of the CycleGAN, G is responsible for the translation of images x from domain X to images that belong to Y, while F translates images from Y to X. As a counterfactual explanation should show images that the classifier would assign to another class as the original input images, the following statements should be fulfilled by G and F respectively:

$$\begin{gathered} C(img) = X \implies C(G(img)) = Y \\ \text{and} \\ C(img) = Y \implies C(F(img)) = X \end{gathered} \tag{17}$$

Most state-of-the-art classifiers do not simply output the actual class that was predicted. They rather use a softmax function to output a separate value for each class, representing the probability that the input actually belongs to the respective class. Thus, we extend the above formulation of our binary classifier to $C_2(img) = (p_X, p_Y)^T$, where p_X represents the probability of img belonging to X, while p_Y represents the probability of img belonging to Y. With this in mind, we can formulate a loss component for the counterfactual generation:

$$\begin{aligned} \mathcal{L}_{counter}(G, F, C) = \ & \mathbb{E}_{x \sim p_{data}(x)}[\|C_2(G(x)) - \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)\|_2^2] \\ & + \mathbb{E}_{y \sim p_{data}(y)}[\|C_2(F(y)) - \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)\|_2^2], \end{aligned} \tag{18}$$

where $\|\cdot\|_2^2$ is the squared L2 Norm (i.e., the squared error).

We chose the vector $(1,0)^T$ and $(0,1)^T$ since we wanted very expressive counterfactuals that are understandable by non-expert users. In theory one could also chose closer vectors like $(0.49, 0.51)$ to enforce counterfactual images that are closer to the decision boundary of the classifier.

Using our proposed counterfactual loss function allows to train a CycleGAN architecture for counterfactual image generation. During training, the generator networks of both GANs are getting punished for creating translated images that are not classified as belonging to the respective counterfactual class by the classifier.
Furthermore, as proposed by the authors of CycleGAN (Zhu et al., 2017), we add an *identity loss*, that forces input images to stay the same, if they already belong to the target domain:

$$\mathcal{L}_{identity}(G,F) = \mathbb{E}_{y\sim p_{data}(y)}[||G(y)-y||_1] + \mathbb{E}_{x\sim p_{data}(x)}[||F(x)-x||_1] \quad (19)$$

Thus, the complete objective function of our system is composed as follows:

$$\begin{aligned}\mathcal{L}(G,F,D_X,D_Y,C) = {} & \mathcal{L}_{GAN}(G,D_Y,X,Y) \\ & + \mathcal{L}_{GAN}(F,D_X,Y,X) \\ & + \lambda\mathcal{L}_{cycle}(G,F) \\ & + \mu\mathcal{L}_{identity}(G,F) \\ & + \gamma\mathcal{L}_{counter}(G,F,C) \end{aligned} \quad (20)$$

where μ is an *Identity Loss Weight* and γ is a *Counterfactual Loss Weight*. During training, the discriminators D_X and D_Y aim to maximize that objective function, while the generators G and F try to minimize it.
A schematic overview of our approach is depicted in Figure 34.

7.3 IMPLEMENTATION AND COMPUTATIONAL EVALUATION

The code of our implementation can be found online.[1]

7.3.1 *Use Case: Pneumonia Detection*

One major drawback of common XAI techniques such as LIME or LRP is that they highlight certain regions of interest, but they do not tell something about the semantics of that regions. Thus, when explaining a machine learning model, they give information about *where* to look for relevant things, but not explicitly *why* those things are relevant. Counterfactual explanation images tackle this problem. We argue that the advantages of such a counterfactual system stand out especially in explanation tasks where the users of the system do

[1] https://github.com/hcmlab/GANterfactual

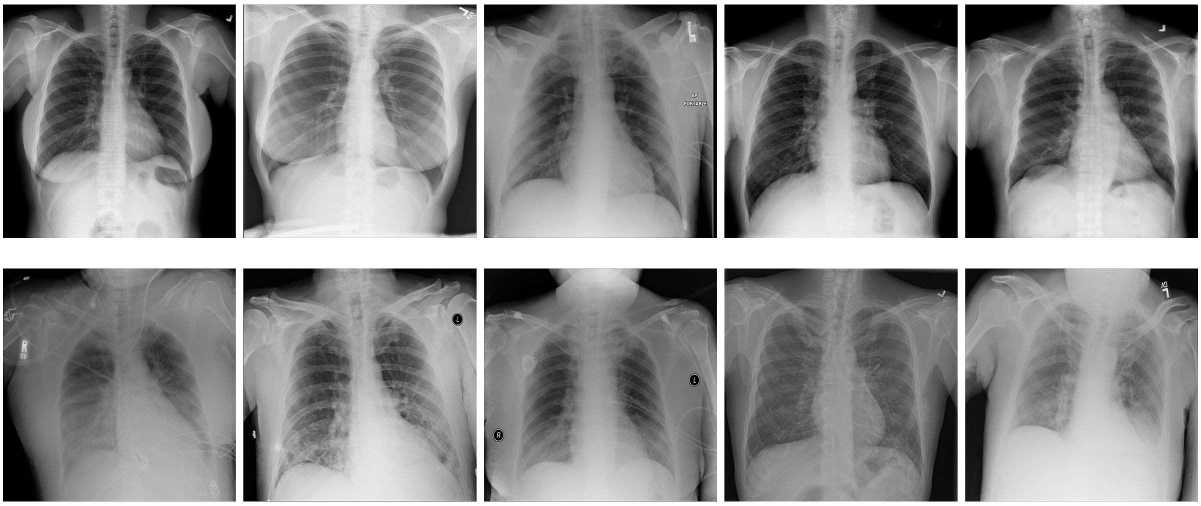


Figure 35: Example images of the used dataset. The top row shows images that are labeled as *Normal*, while the bottom row shows images labeled as *Lung Opacity*, indicating lungs that are suffering from pneumonia.

not have much prior knowledge about the respective problem area and thus are not able to interpret the semantics of the regions of relevance without assistance.

Therefore, to evaluate our approach, we chose the exemplary use-case of *Pneumonia Detection*. We trained a binary classification Convolutional Neural Network (CNN) to decide whether a given input of a human upper body's x-ray image shows a lung that suffers from pneumonia or not. Subsequently, we trained a CycleGAN that was modified with our proposed counterfactual loss function, incorporating the trained classifier.

Besides the importance of medical non-experts being able to understand diagnoses relating to them (Zucco et al., 2018), medical non-experts do not have a deeply formed mental model of the chosen domain. As such, we hypothesize that this leads to a lack of interpretability for common XAI techniques that only highlight areas of relevance.

7.3.1.1 *Classifier Training*

The aim of this section is to give an overview of the classifier that we want to be explained for our particular use case. However, we want to emphasize that our approach is not limited to this classifier's architecture. The only requirement for training our explanation network is a binary classifier C that is able to return a class probability vector $(p_X, p_Y)^T$ for an image that is fed as input.

To evaluate our system, we trained a CNN to decide if input images of x-rays are showing lungs that suffer from pneumonia or not. We used the dataset published for the *RSNA Pneumonia Detection Challenge*[2] by the Radiological Society of North America. The original dataset contains 29,700 frontal-view x-ray images of 26,600 patients. The training data is split into three classes: *Normal*, *Lung Opacity* and *No Lung Opacity / Not Normal*. We took only the classes *Normal* and *Lung Opacity*, as Franquet (2018) argue that opacity of lungs is a crucial indicator of lungs suffering from pneumonia, and we only wanted to learn the classifier to distinct between lungs suffering from pneumonia and healthy lungs. Other anomalies that do not result in opacities in the lungs are excluded from the training task to keep it a binary classification problem. All duplicates from the same patients were removed as well. For the sake of simplicity, we will refer to the class *Lung Opacity* as *Pneumonia* in the rest of this chapter. The resolution of the images was reduced to 512x512 pixels. Subsequently, we randomly split the remaining 14,863 images into three subsets: *train*, *validation*, and *test*. The distribution of the partitions is shown in Table 7.

Partition	Normal	Pneumonia	Total
Train (70%)	6195	4208	10403
Validation (10%)	886	602	1488
Test (20%)	1770	1202	2972
Total	8851	6012	14863

Table 7: Distribution of the images of the used dataset.

We trained an AlexNET architecture to solve the described task. For details about AlexNET, we want to point the interested reader to Krizhevsky, Sutskever, and G. E. Hinton (2017). We slightly modified the architecture to fit our needs. These modifications primarily include L2 regularization to avoid overfitting. Further, we replaced the loss function with an MSE loss, as this worked well for our classification task. A detailed description of the model that we used can be found in the appendix. The training configuration is shown in Table 8.

After training the classifier on the *train* partition for 1000 epochs, it achieved an accuracy of 91,7% on the *test* set (f1 score: 0.894; f2 score: 0.883). It should be noted that there exists a plethora of work that focuses on building classifiers that achieve a high classification performance on tasks that are similar to this one. Those classifiers achieve much better performance values than our classifier does. However, the aim of our work is to *explain* the decisions of a classifier. Explaining an AI model does not only include explaining decisions where the AI was right, but also cases where the AI was wrong, as a complete understanding of an AI also covers an understanding of cases

[2] https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/

Parameter	Value
Optimizer	Stochastic Gradient Descent
Learning Rate	0.0001
Momentum	0.9
Batch Size	32
Epochs	1000
Loss Function	Mean Squared Error

Table 8: Training configuration of the used AlexNET.

where the AI's decisions are incorrect. Thus, we found that a *perfect* classification model would not be an appropriate tool to measure the performance of an XAI system, resulting in our decision to not improve the classifier performance further (i.e., we did not conduct any hyperparameter tuning or model optimization).

7.3.1.2 *CycleGAN Training*

We trained a CycleGAN model with the objective function adapted as proposed in Section 7.2. As training dataset, we used the *train* partition of the same dataset that we used for our classifier. Our proposed counterfactual loss $\mathcal{L}_{\mathtt{counter}}$ was calculated using the trained classifier that was described in the previous subsection. The architecture of both the generators as well as both the discriminators where adopted from Zhu et al. (2017). As proposed by them, we additionaly used a modified version of the discriminator architecture called *PatchGAN*. This variant of the discriminator approximates validity values for different patches of the input instead of a single validity value for the whole input. Such a validity value estimates whether the input was generated by the generator or came from the training set. Further architectural details can be found in their publication. The training configuration parameters are listed in Table 9. Examples of counterfactual images that were produced by feeding images from the *test* partition into our trained generative model are shown in Figure 36. Here, the main structure and appearance of the lungs are maintained during the translation process, while the opacity of the lungs is altered. This was expected due to the pneumonia class of the used dataset being defined by lungs that show a certain degree of opacity. All in all, the visual inspection of the produced results already shows that our approach is promising.

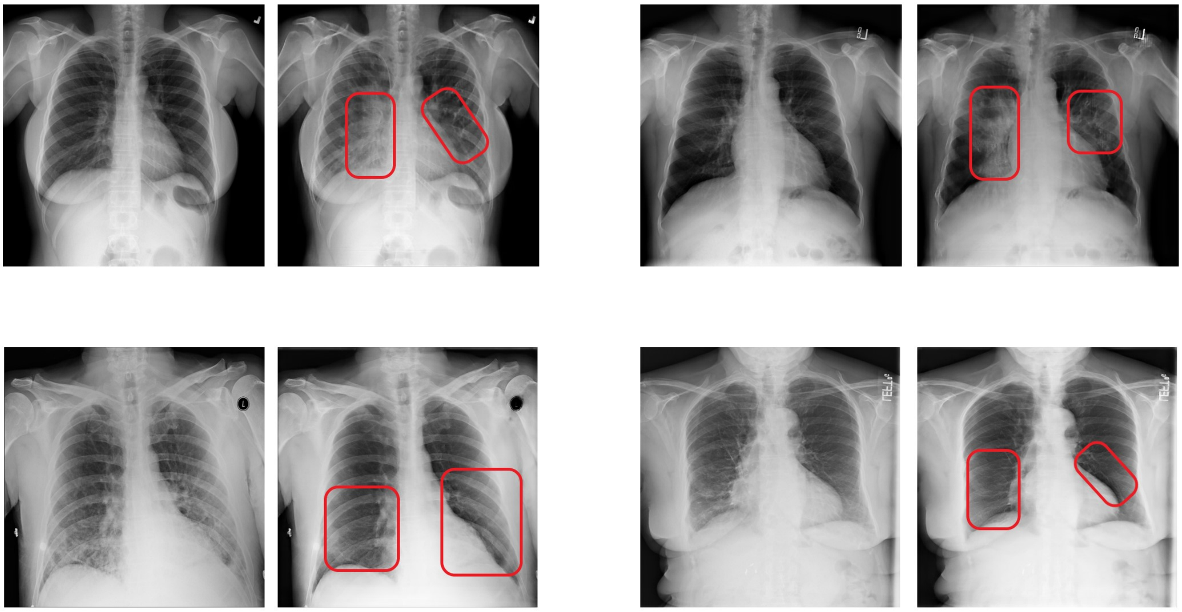


Figure 36: Examples of counterfactual images produced with our proposed approach. In each pair, the left image shows the original image, while the right image shows the corresponding counterfactual explanation. The red boxes were added manually to point the reader to the regions that were altered the most. The original images in the top row were classified as *normal*, while the original images in the bottom row were classified as *pneumonia*. The shown counterfactual images were all classified as the opposite as their respective counterpart.

Parameter	Value
Optimizer	Adam
Learning Rate	0.0002
Beta 1	0.5
Beta 2	0.999
Batch Size	1
Epochs	20
Cycle Consistency Loss Weight	10
Identity Loss Weight	1
Counterfactual Loss Weight	1

Table 9: Training configuration of the CycleGAN with our proposed counterfactual loss function.

7.3.2 *Computational Evaluation*

To see if the produced counterfactual images are classified differently than the original input images, we evaluated the system on the *test* partition. By doing so, we explicitly assess the *Validity* of the counterfactual explanations. To this end, we fed every image into the classi-

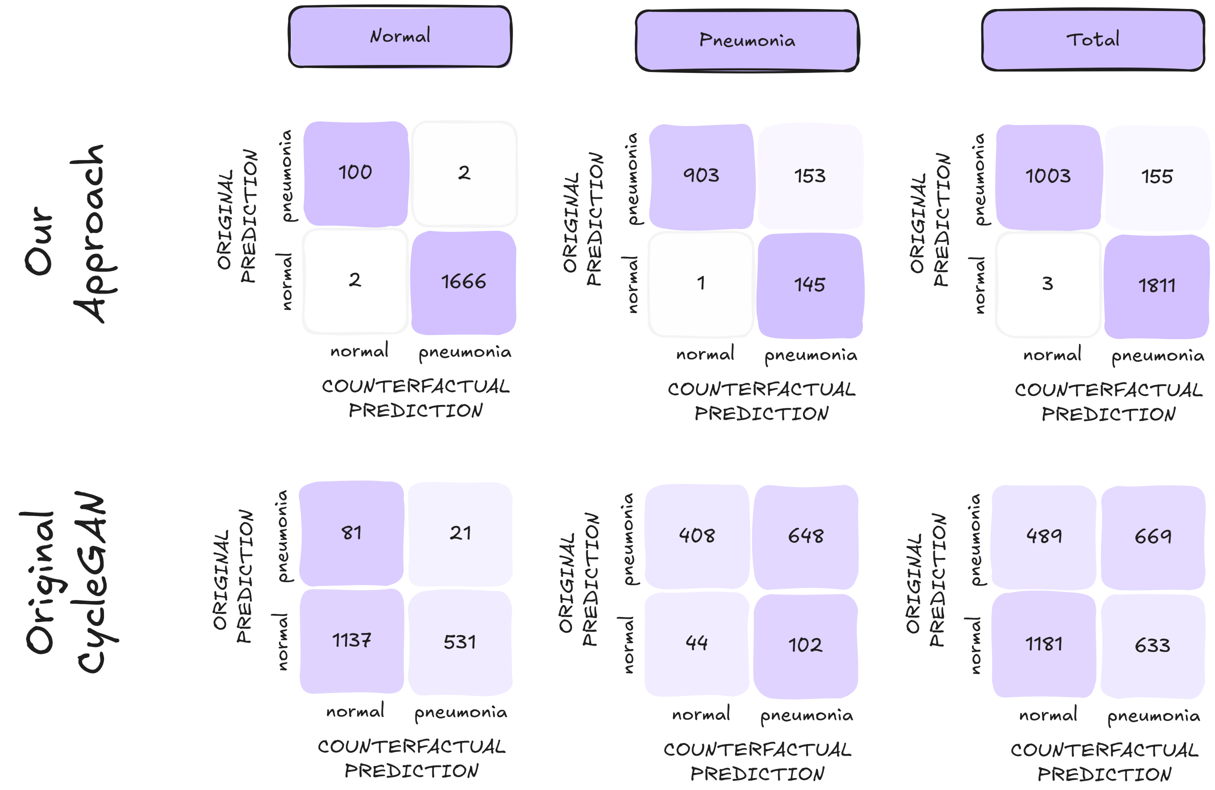


Figure 37: Computational evaluation results of the counterfactual image generation performance. The confusion matrices show the number of samples out of each subset (Normal, Pneumonia, Total) of the rsna dataset that the classifier predicted to be the respective class before (y-axis) and after (x-axis) the samples had been translated by either the original CycleGAN or by our approach.

fier, translated the image by the use of the respective generator network, and then classified the resulting counterfactual image. We did this separately for the images that originally were labeled as *normal*, as well as for those that were labeled as *Lung Opacity*. We performed this whole procedure for a CycleGAN that was modified with our approach, as well as for an original CycleGAN architecture that does not implement our proposed counterfactual loss function. It should be noted that this computational evaluation is not meant to assess the explanation performance per se, but rather it evaluates if our main modification to the CycleGAN (i.e., the addition of the counterfactual loss), indeed enhances the CycleGAN architecture with the capability to generate counterfactual images. To assess the explanation performance of our approach compared to traditional XAI techniques, we conducted a user study that will be described in Section 7.4.

Figure 37 shows the results of the computational evaluation. It can be seen that the counterfactual images generated by our approach were indeed classified as a different class than the original image in most of the cases. In total, our approach reaches an accuracy of 94.68%, where we understand the accuracy of a counterfactual image generator to be the percentage of counterfactuals that actually changed the classifier's prediction. For the images that were originally labeled as *normal*, the accuracy was 99.77%, while for the images that were labeled as *Lung Opacity* the accuracy reached 87.19%. Contrary, the original CycleGAN only reaches 37.75% accuracy in total (34.58% on *normal* lungs, 42.43% on *Lung Opacity* lungs). Those results indicate that the modification of the CycleGAN's objective with our additional counterfactual loss has a huge advantage over the original CycleGANs when aiming for the creation of counterfactual images. In conclusion, the counterfactual generation with our approach works sufficiently well, but it has a harder time when being confronted with images that actually show lungs suffering from pneumonia than in the case of processing images that show healthy lungs.

7.4 USER STUDY

To investigate the advantages and limitations of XAI methods, it is crucial to conduct human user studies.

In this section, we describe the user study we conducted to compare our proposed counterfactual approach with two state-of-the-art XAI approaches (LRP and LIME).

7.4.1 *Conditions*

We compare three independent variables by randomly assigning each participant to one of three conditions. The participants in each condition only interacted with a single visual explanation method. This between-subject design was chosen to avoid fatigue among the participants. Participants in the LRP condition were assisted by heatmaps generated through Layer-wise Relevance Propagation using the z-rule for fully connected layers and the $\alpha1\beta0$-rule for convolutional layers, as recommended by Montavon et al. (2019). The LIME condition contained highlighted Super-Pixels which were generated by LIME. Here, we chose the *Simple Linear Iterative Clustering* (SLIC) segmentation algorithm which Schallner et al. (2019) found to perform well in a similar medical use case. For the remaining hyperparameters, we used the default values and showed the five most important super-pixels. For both LIME and LRP, we omit the negative importance values since those were highly confusing to participants in our pilot study. Participants in the counterfactual condition were shown counterfactual images generated by our proposed approach (see section 7.2). The three different visualisations can be seen in Figure 38.

7.4.2 *Hypotheses*

All our hypotheses are targeting non-experts in healthcare and artificial intelligence. Since our aim is to evaluate our proposed counterfactual approach, we do not investigate differences between the saliency map conditions (LRP and LIME). For our user study we formulated the following hypotheses. Note that we did not specifically evaluate the *image quality* of the explanations, as the focus of our study was on the conceptual comparison between our method and feature attribution mechanisms and not on achieving the best possible quality.

- **Explanation Satisfaction**: Participants are more satisfied with the explanatory quality of counterfactuals compared to LIME and LRP.

- **Mental Models**: Counterfactuals helped participants to create more correct mental models about the AI than LIME and LRP.

- **Trust**: Participants have more trust in the AI system if it is explained with counterfactuals than if it is explained with LRP or LIME.

- **Emotions**: The intuitive and simple interpretation of counterfactuals makes participants feel happier, more relaxed and less angry compared to LRP and LIME.

- **Self-efficacy**: If counterfactuals are a more satisfying XAI method than LRP or LIME, participants feel also strengthened in their self-efficacy towards the AI system, compared to participants in the LRP and LIME conditions.

7.4.3 *Methodology*

To evaluate our hypotheses, we used the following Methods:

MENTAL MODELS We use two metrics to evaluate the mental models that the participants formed through our XAI methods. Quantitatively, we conduct a **prediction task**, as proposed by Hoffman et al. (2018), where the participants have to predict what the AI model will decide for a given x-ray image. For a more qualitative evaluation, we used a form of **task reflection**, also proposed by Hoffman et al. (2018). Here, the participants were asked to describe their understanding of the AI's reasoning after they completed the prediction task. For this, the participants were asked two questions about their mental model of the AI: "What do you think the AI pays attention to when it predicts pneumonia?" and "What do you think the AI pays attention to when it predicts healthy lungs?"

EXPLANATION SATISFACTION We used the Explanation Satisfaction Scale, proposed by Hoffman et al. (2018) to measure the participants' subjective satisfaction with the visual explanations (LRP, LIME, or counterfactuals) that we presented.

TRUST To evaluate the trust in the presented AI system, we used two items (i.e., "I trust the system" and "I can rely on the AI system") from the Trust in Automation (TiA) questionnaire proposed by Körber (2018). Körber points out that one or two items are sufficient to measure trust if people have no previous experience with the system, as is the case with our system.

EMOTIONS We used items for the subscales *anger*, *happiness*, and *relaxation* of the Discrete Emotions Questionnaire (DEQ) (C. Harmon-Jones, Bastian, and E. Harmon-Jones, 2016) to evaluate the participants' feelings after having solved the tasks.

SELF-EFFICACY We used one item to measure the self-efficacy towards the AI system. For this, we used a variation of one item proposed by Bernacki, Nokes-Malach, and Aleven (2015) (i.e., "How confident are you that you could detect pneumonia using the presented explanations in the future?").

7.4.4 *Participants*

In order to detect an effect of η_p^2 =0.04, with 80 % power in a one-way between subject MANOVA (three conditions, α=.05), the conducted a-priori power analysis suggested that we would need 37 participants in each condition (N = 111). In order to compensate for possible dropouts, we collected data of 122 participants using the Clickworker online platform[3].

To ensure a sufficient English level, participation was limited to users from the US, UK, Australia, or Canada whose native language is English. Since LRP and LIME are not designed with color blind people in mind, the participants were also asked if they were color blind and stopped from participating if they were.

To make sure that the participants understood the provided information about the task correctly, we used a quiz that they had to complete correctly to take part in the study. As incentive to diligently do the task, the participants received a bonus payment in addition to the base payment if they correctly predicted at least 2/3 of the AI model's prediction. In addition to these precautions, we subsequently excluded 4 participants due to the fact that they never looked at the XAI visualisations or their responses did not reflect a serious engagement with the study (e.g., free text answers which are not related to the question at all).

For our final analysis we used data from 118 participants between 18 and 67 years (*M* = 38.5, *SD* = 10.9). 63 of them were male, 53 female and 2 non-binary. The participants were randomly separated into the three XAI visualisation conditions. All in all, only 8 participants reported experience in healthcare. 43 participants stated that they had experience in AI. The level of AI and healthcare experience was evenly distributed between the three conditions.

7.4.5 *Procedure*

The entire study was web-based. After providing some demographic information, the participants received a short tutorial that explained the x-ray images and the XAI visualisations which they would interact with in the experiment. After the tutorial, each participant had to answer a quiz. Here, questions were asked to ensure that the participants carefully read the tutorial and understood how to interpret the x-ray images (e.g., "Which part of the body is marked in this picture?") and the XAI visualisations (e.g., "What do green areas in images tell you?" for the LIME and LRP conditions). Only participants who solved the quiz successfully were allowed to participate in the actual experiment.

The quiz was followed by the prediction task. Here, the participants were asked to predict the AI's diagnosis for 12 different images. To avoid cherry picking while still ensuring variety in the images, we randomly chose 12 images based on the following constraints: To make sure that the classifier equally makes false and correct predictions for each class, we wanted 3 true positives, 3 false positives, 3 true negatives, and 3 false negatives. Furthermore, inspired by Alqaraawi

[3] https://www.clickworker.com/clickworker/

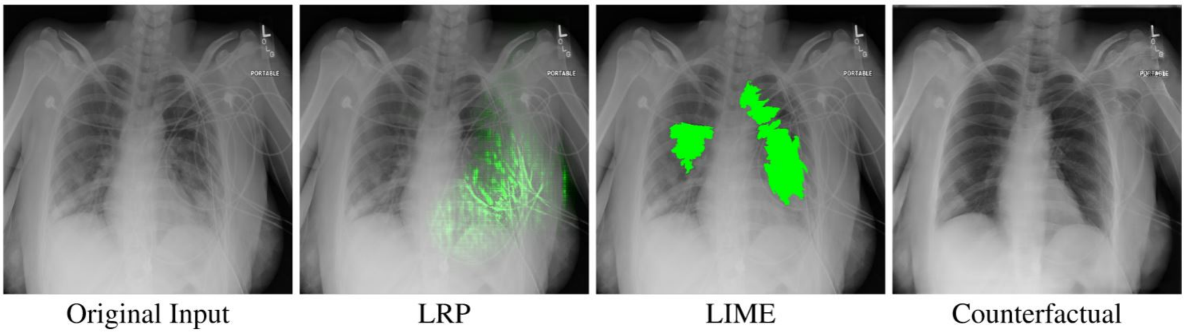


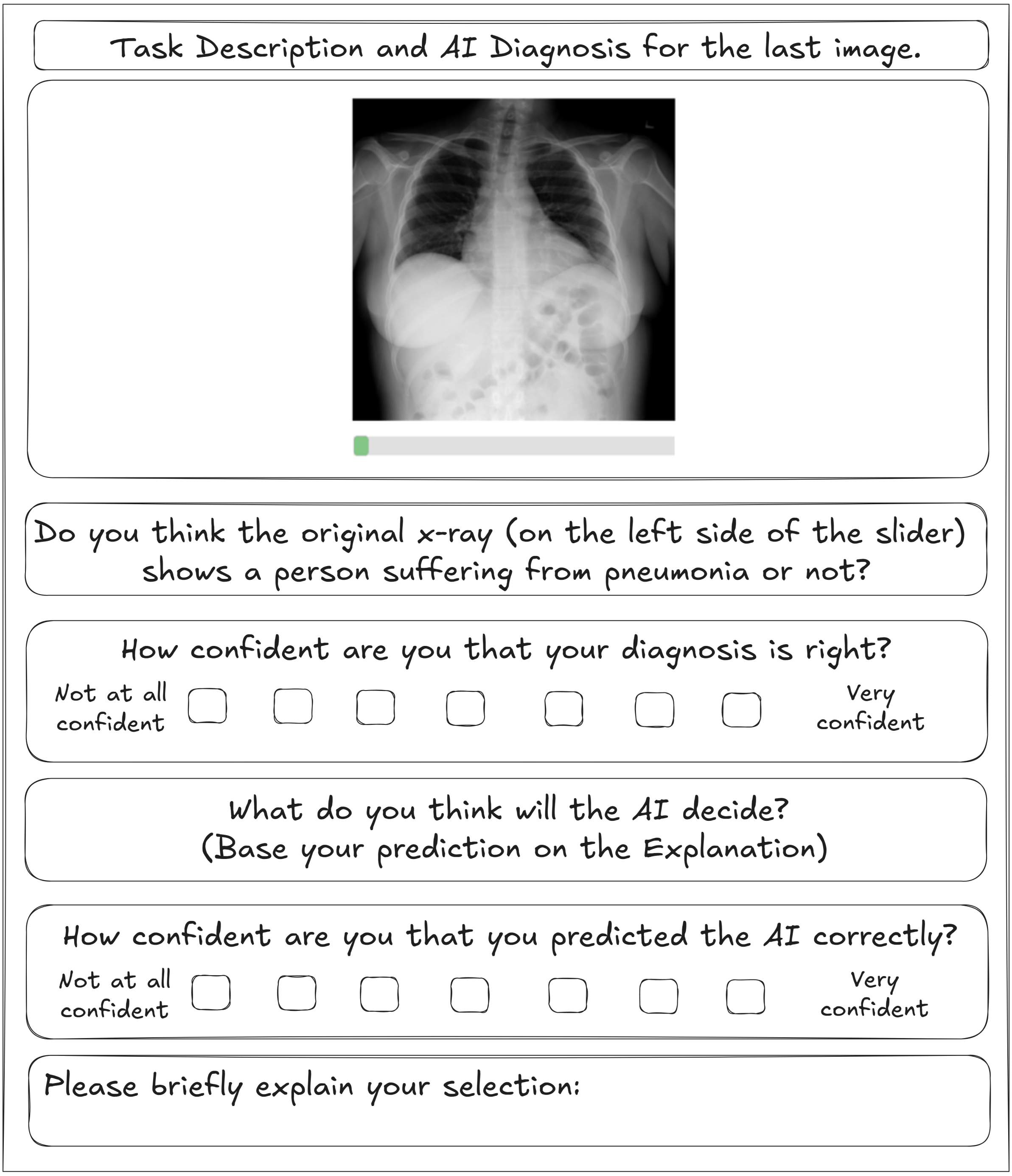
Figure 38: An example x-ray image classified as *Pneumonia*, as well as the different XAI visualizations used in our study when the slider is fully on the right side. Best viewed in color.

et al. (2020), we additionally used the AI model's confidence to ensure diversity in the images. Decisions where the model is certain are often easier to interpret than decisions where the AI model struggled. Since our prediction classifier mainly had confidence values between 0.8 and 1, we randomly chose one x-ray image with confidence values of 0.8, 0.9 and 1 (rounded) out of each of the sets of true positives, false positives, true negatives, and false negatives.

In addition to the original image, the participants were provided with a slider to interact with the XAI visualizations. Moving the slider to the right linearly interpolated the original image to either the counterfactual image or a version of the image that is augmented with a LRP or LIME heatmap, depending on the condition the user was in. Figure 38 shows an example of the 3 different XAI visualizations for one of the images used in our experiment. By tracking if the participants used the slider, we additionally know whether they looked at the XAI visualizations.

In our pilot study (N = 10), we found that participants often project their own reasoning to the AI. To mentally differentiate between their own diagnosis and the AI's diagnosis, the participants in the final study were asked whether they *themselves* would classify the given image as *pneumonia* or *not pneumonia* and how confident they are in this diagnosis on a Likert scale from 1 (not at all confident) to 7 (very confident). Then they were asked to predict whether *the AI* will classify the image as *pneumonia* or *not pneumonia*, based on the given XAI visualization. Again, they had to give a confidence rating in their prediction from 1 to 7. Finally, they could give a justification for their prediction if they wanted to. After each prediction they were told the actual decision of the AI for the last image. A schematic of the full task is shown in Figure 39.

After predicting the AI's decision for all 12 x-ray images, the task reflection followed. Here, participants had to describe their understanding of the AI's reasoning. Then the questionnaires about Explanation Satisfaction, Trust, Self-efficacy and Emotion were provided.



Task Description and AI Diagnosis for the last image.

Do you think the original x-ray (on the left side of the slider) shows a person suffering from pneumonia or not?

How confident are you that your diagnosis is right?

Not at all confident □ □ □ □ □ □ □ Very confident

What do you think will the AI decide?
(Base your prediction on the Explanation)

How confident are you that you predicted the AI correctly?

Not at all confident □ □ □ □ □ □ □ Very confident

Please briefly explain your selection:

Figure 39: A simplified schematic of our prediction task.

7.4.6 *Evaluation Methods*

QUANTITATIVE EVALUATION OF THE RESULTS We calculated the mean of the correct predictions of the AI and the participants confidences in their predictions of the AI. To make sure that we only use responses, where the participants at least saw the visual explanations, we excluded answers where the participant did not move the slider. If, for example, a participant did not use the slider 4 times then we only calculated the mean for the remaining 8 answers.

For the DEQ we calculated the mean for the emotion subscales happy, anger, and relaxation. For the TiA, we calculated an overall trust score from the two questions presented.

QUALITATIVE EVALUATION OF THE PARTICIPANTS' MENTAL MODEL OF THE AI Similar to Anderson et al. (2019) and Huber, Weitz, et al. (2020), we used a form of summative content analysis (Hsieh and Shannon, 2005) to qualitatively evaluate the participants' free text answers to the questions "What do you think the AI pays attention to when it predicts pneumonia?" and "What do you think the AI pays attention to when it predicts healthy lungs?". Our classifier was trained on a dataset consisting of x-ray images of normal lungs and x-ray images that contain lung opacity, which is a crucial indicator of lungs suffering from pneumonia. Since we only told the participants that our model classifies pneumonia, we can score their responses based on whether they correctly identified lung opacity as a key decision factor for our model. To this end, two annotators independently went through the answers and assigned concepts to each answer (e.g., *opacity*, *clarity*, *contrast* and *other organs than the lung*). Then, answers to the pneumonia question that contained at least one concept which related to opacity, like *opacity*, *white color in the x-ray* and *lung shadows*, received 1 point. Answers to the healthy lungs question that contained at least one concept related to clarity, like *clarity*, *black color in the x-ray* or *no lung shadows*, received 1 point. Answers for both questions that contained a concept related to contrast, like *contrast* or *clear edges*, received 0.5 points. All other answers received 0 points. For 21 out of all 236 responses, the two annotators differed in the given score. Here, a third annotator was asked to assign 0, 0.5 or 1 points to the answer and the final points were calculated by majority vote between the three annotators. By adding the points for those two questions, each participant was given a score between 0 and 2, approximating the correctness of their description of the AI.

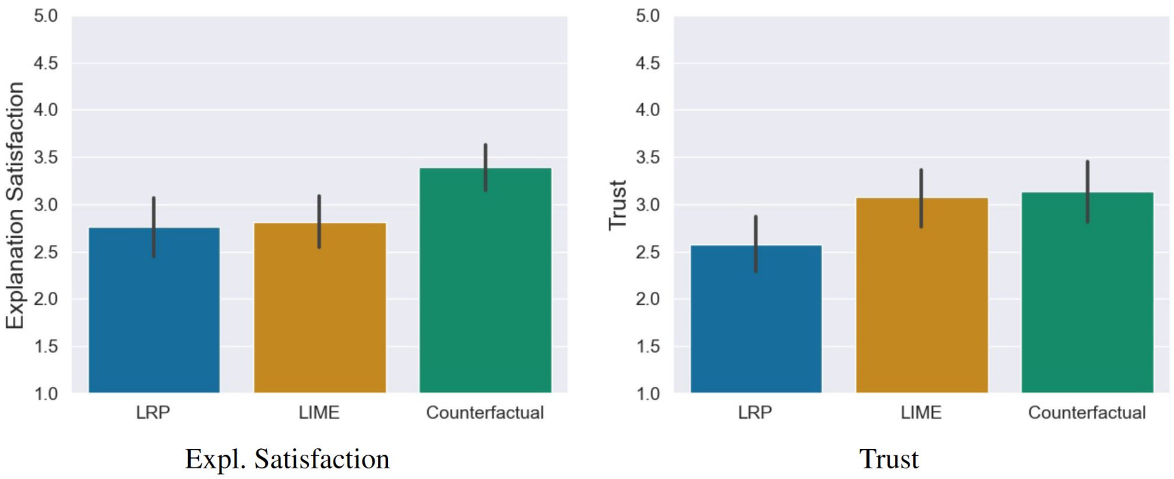


Figure 40: Results of the explanation satisfaction and trust questionnaires. Error bars represent the 95% Confidence Interval (CI).

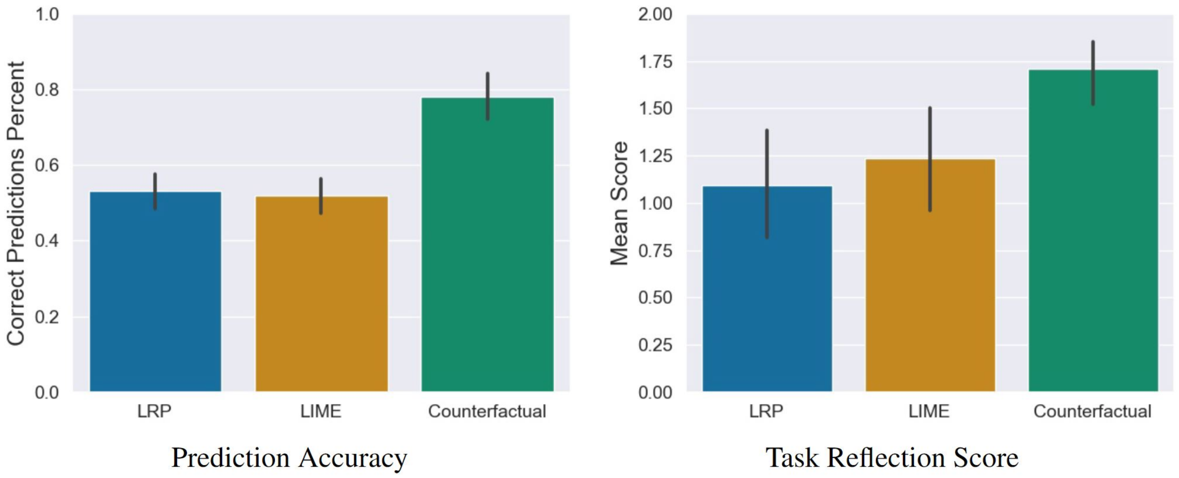


Figure 41: Results of the prediction task, and the task reflection questions. Error bars represent the 95% Confidence Interval (CI).

7.5 RESULTS

7.5.1 *Impact of XAI methods on Explanation Satisfaction, Trust, and Prediction Accuracy*

As a first impression of their mental models of the AI, the participants had to predict the decision of the neural network (pneumonia / no pneumonia). At the end of the study, they rated their trust in the AI as well as their explanation satisfaction. To evaluate these variables between the three conditions, we conducted a one-way MANOVA. Here, we found a significant statistical difference, Wilks' Lambda = 0.59, $F(6, 226) = 11.2$, $p < .001$. The following ANOVA revealed that all three variables showed significant differences between the conditions:

- **Prediction accuracy**: $F(2, 115) = 30.18$, $p = .001$,
- **Explanation satisfaction**: $F(2, 115) = 5.87$, $p = .004$,
- **Trust**: $F(2, 115) = 3.89$, $p = .02$,

To determine the direction of the differences between the three XAI method conditions, we used post-hoc comparisons for each variable[4]. The effect size d is calculated according to Cohen[5] (Cohen, 2013).

We found the following differences:

- **Prediction accuracy**: The participants' predictions of the AI's decisions were significantly more correct in the counterfactual condition compared to the LRP condition $t(115) = -6.48$, $p = .001$, $d = 1.47$ (large effect) as well as compared to the LIME conditions $t(115) = -6.92$, $p = .001$, $d = 1.55$ (see left sub-figure of Figure 41).
- **Explanation satisfaction**: Participants were significantly more satisfied with the explanation quality of the counterfactual explanations compared to the LRP saliency maps, $t(115) = -3.05$, $p = .008$, $d = 0.70$ (medium effect) and the LIME visualisations, $t(115) = -2.85$, $p = 0.01$, $d = 0.64$ (medium effect)(see Figure 40).
- **Trust**: The AI was rated as significantly more trustworthy in the counterfactual condition compared to the LRP condition, $t(115) = -2.56$, $p = .03$, $d = 0.58$ (medium effect) but not to the LIME condition, $t(115) = -0.29$, $p = .07$ (see Figure 40).

7.5.2 *Results of the qualitative Evaluation of the Users' Mental Models*

Subsequently to the significant differences in the prediction accuracy as a first impression of the mental model of the participants, we analysed the results of the content analysis of the task reflection responses. For this, we conducted a one-way ANOVA. Here we found a significant statistical difference, $F(2, 115) = 7.91$, $p < .001$. To determine the direction of the differences between the three conditions, we used post-hoc comparisons (see right sub-figure of Figure 41): Participants were asked to describe the AI's reasoning in three different conditions: counterfactual, LRP and LIME. Out of these, participants created correct descriptions significantly more often in the counterfactual condition compared to the LRP condition, $t(115) = -3.76$, $p < .001$, $d = 0.85$ (large effect) and the LIME condition, $t(115) = -2.97$, $p = .01$, $d = 0.66$ (medium effect).

[4] We used the Holm correction for multiple testing to adjust the p-values for all post-hoc tests we calculated.

[5] Interpretation of the effect size is: $d < .5$: small effect; d= 0.5-0.8 : medium effect; $d > 0.8$: large effect

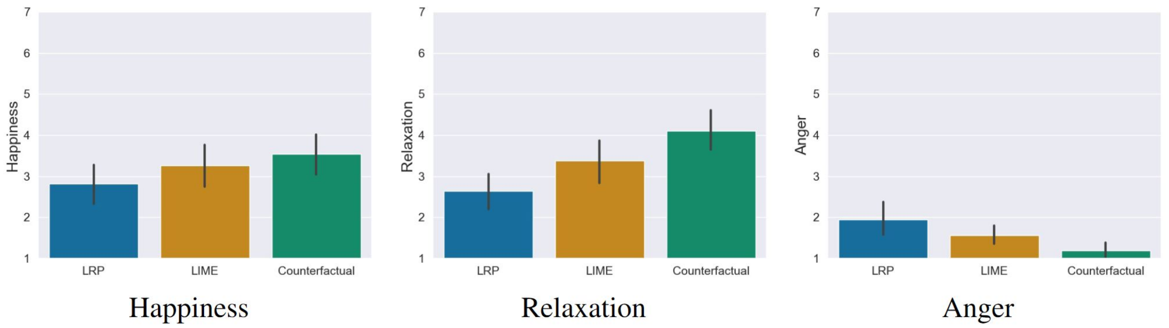


Figure 42: Results of the emotion questionnaires. Participants in the counterfactual condition felt significantly less angry and more relaxed compared to the LRP saliency map condition. For LIME, no significant differences were found. Error bars represent the 95% CI.

7.5.3 *Impact of XAI Methods on Users' Emotional State*

We also wanted to investigate whether working with the XAI methods had an influence on the emotional state of the participants. To analyse possible effects, we conducted a one-way MANOVA. Here we found a significant statistical difference, Pillai's' Trace = 0.20, $F(6, 228) = 4.26$, $p < .001$. The following ANOVA revealed that the emotion anger, $F(2, 115) = 6.75$, $p = .002$ and relaxation, $F(2, 115) = 9.07$, $p < .001$ showed significant differences between the conditions. Happy showed no significant differences between the conditions, $F(2, 115) = 2.06$, $p = .13$. The post-hoc comparisons[6] showed the following differences (see Figure 42):

- **Anger**: Participants in the counterfactual condition felt significantly less angry than in the LRP condition, $t(115) = 3.68$, $p = .001$, $d = 0.83$ (large effect). No differences were found for the LIME condition, $t(115) = 1.83$, $p = .12$.
- **Relaxation**: Participants in the counterfactual condition were significantly more relaxed than in the LRP condition, $t(115) = -4.26$, $p < .001.$, $d = 0.96$ (large effect). No differences were found for the LIME condition, $t(115) = -2.12$, $p < .06$[7]

7.5.4 *Impact of XAI Methods on Users' Self-Efficacy*

The analysis showed that (1) the quality of counterfactual explanations was rated significantly higher and (2) participants predicted the decisions of the AI significantly more accurate compared to LIME and LRP. Based on our last hypothesis, we therefore examined whether these positive assessments were also reflected in the self-efficacy and in the prediction confidence of the participants. For this purpose, we

[6] We used the Holm correction for multiple testing to adjust the p-values

[7] This p-value was no longer significant due to the Holm correction.

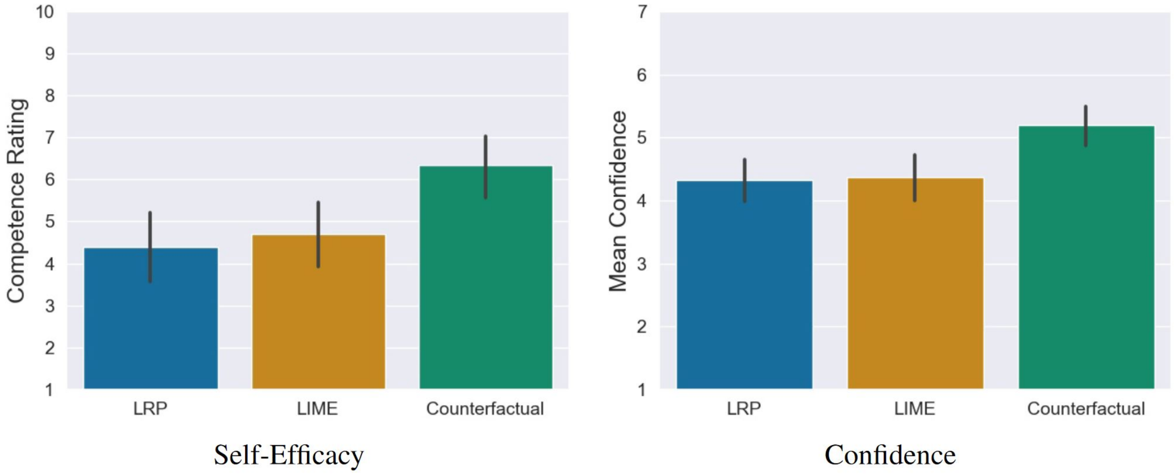


Figure 43: Significant differences regarding self-efficacy and general confidence of the participants in their predictions of the AI between the counterfactual condition and the saliency map conditions (LRP and LIME). Error bars represent the 95% CI.

conducted a one-way MANOVA. Here, we found a significant statistical difference, Pillai's Trace = 0.15, $F(4, 230) = 4.69$, $p = .001$. The following ANOVA revealed a statistical difference for self-efficacy $F(2, 115) = 6.93$, $p = .001$ and prediction confidence $F(2, 115) = 7.68$, $p < .001$ between the conditions. The post-hoc comparisons showed that counterfactuals lead to a significantly higher self-efficacy compared to LRP t = -3.44, $p = .002$, d = 0.78 (medium effect) as well as LIME, $t(115)$ = -2.94, $p = .01$, d = 0.66 (medium effect). The same pattern was found for the prediction confidence, where counterfactuals lead to a significantly higher prediction confidence compared to LRP $t(115)$ = -3.45, $p = .002$, d = 0.78 (medium effect) as well as LIME, $t(115)$ = -3.32, $p = .003$, d = 0.74 (medium effect) (see Figure 43). A closer look reveals that these significant differences stem from the confidence in the correct predictions and not the confidence in the incorrect ones (see Figure 44).

7.6 DISCUSSION

The study described in the previous sections was conducted with the aim to verify our hypotheses. With this in mind, we discuss our results in this section.

7.6.1 *Explanation Satisfaction*

As the results show, the counterfactual explanation images that were generated by the use of our novel approach provided the participants with significantly more satisfying explanations as both of the saliency map approaches. Saliency map methods like LIME and LRP only

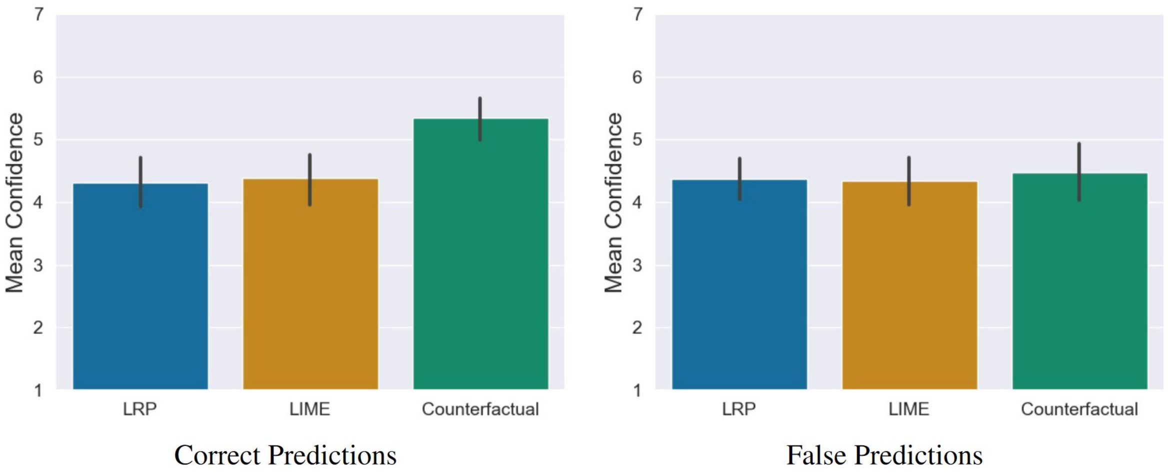


Figure 44: Confidence of the participants in correct and false predictions. The significant difference between the counterfactual condition and the saliency map conditions is based on the confidence in correct predictions, not in the incorrect ones. Error bars represent the 95% CI.

show which pixels were important for the AI's decision. The users are left alone with the task of building a bridge between the information of *where* the AI looked at, and *why* it looked there. Contrary, the counterfactual explanations generated by our system directly show *how* the input image would have to be modified to alter the AI's decision. Thus, the participants did not have to come up with an interpretation of the semantics of important areas by themselves. As the results of our study show, this difference plays a significant role in how satisfying the explanations are to non-expert users, validating our first hypothesis.

7.6.2 *Mental Models*

As described in section 7.4, two different methods were used to evaluate if the explanation systems allowed the participants to build up an appropriate mental model of the classifier. First, the participants had to do a prediction task of 12 images, where they had to decide if the AI would classify each of those images either as *Pneumonia* or *No Pneumonia*. Our results show that the participants were significantly better in performing those prediction tasks when they were shown counterfactual images created by our system than they were when provided with LIME or LRP saliency maps. Again, it could be argued that this advantage is caused by the fact that the counterfactual images give more than just a spatial information about the regions of importance. In fact, the actual decision of the AI was highly dependent on the blurriness of certain areas of the lung. A crucial thing to mention is that the absence of blurriness, i.e. the clarity of x-ray images that do not show lungs that are infected by pneumonia, obviously occurs at similar places where cloudy areas would appear in the case of pneumonia. Thus, the visual highlighting created by LIME or LRP predominantly shows where this distinction between opaque and not opaque lungs is made. However, the information is missing to which degree the AI actually thinks that there is an opacity in the lung. In contrast, the counterfactual images give this information by increasing or decreasing that opacity respectively. In general, we think that our counterfactual system has the most advantage in these kind of tasks, where the important regions are not distinct for different decisions. Specifically, we think that our approach excels in tasks where the AI's decision is being directed by different textural characteristics rather than by the position of certain objects in the image. The content analysis of the task reflection strengthens this assumption. Here, participants from the LRP and LIME conditions often referred to certain organs or regions in the image instead of focusing on the key decision factor of opacity. Examples for this are: "The AI pays attention not to just the lungs but the surrounding areas as well. The Abdomen seems to be an area of focus.", "From the heatmap I noticed the AI paying attention to the surrounding areas of the lungs, the spine, heart, abdomen, and the armpits often when it predicted pneumonia." and "I think the AI needs to see the green near the bottom of the chest to think healthy lungs."

7.6.3 *Trust*

Our results show that counterfactual explanations encouraged the participants to have more trust in the AI system. However, this only became apparent in comparison to LRP, but not to LIME. This result indicates that, on the one hand, the type of explanation (counterfactual explanation vs. feature importance/saliency maps) has an influence on the perceived trust of users. On the other hand, it also shows that even explanations of one type of XAI mechanisms (here: saliency map approaches) are perceived differently by users. This finding is important because it indicates that the type of visualisation (pixel-wise or superpixel-based) also has an influence on the users' trust rating. In our study we examined the general influences of three XAI methods on trust. Based on the results, further analyses are now necessary. For example, the question arises whether there is a correlation between the participants' predictions and the trust rating. One interesting observation in our results is that participants in the LIME condition trusted the system on a similar level as the participants in the counterfactual condition even though they performed significantly worse in the mental model evaluation. This indicates that their trust might not be justified. While this is interesting, the question of whether the trust of the participants in the AI system was actually justified needs to be examined more closely in the future.

7.6.4 *Emotions*

In our user study, we not only investigated the impact of XAI visualisations on trust and mental models, but also - for the first time - the emotional state of the participants. The results show that XAI not only influences users' understanding and trust, but also has an impact on users' affective states: Counterfactual explanations promote positive emotions (i.e., relaxation) and reduce negative emotions (i.e., anger). We suspect that this was because users might have found it easier to interpret the counterfactuals: users might have gotten frustrated while looking at explanations that they did not fully understand.

7.6.5 *Self-efficacy*

Our results show that participants were not only able to correctly assess the predictions of the AI with the help of the counterfactual explanations, but also that they were very confident in their judgements. Upon closer inspection we found that this boost in confidence only stems from the predictions which the participants got right. This indicates that they were not overconfident but justified in their confidence. While this is an interesting observation, it needs further investigation. The increase in confidence is also reflected in a significant increase in the self-efficacy of participants in the counterfactual condition, compared to LIME and LRP. Already A. Heimerl et al. (2020) assumed that the use of XAI could be a valuable support to improve self-efficacy towards AI. This assumption was empirically proven for the first time in our study and contributes to building more human-centered AI systems.

7.6.6 *Limitations*

It has to be investigated further how our proposed counterfactual generation method performs in other use cases. We believe that the advantage of our system in this pneumonia detection scenario to some degree results from the fact that the relevant information of the images is of a rather textural structure.

A further noteworthy observation is that, although the study showed that the produced counterfactuals lead to good results in our chosen non-expert task, our system modifies relevant features in a very strong way, i.e., features that are relevant for the classifier are modified to such a degree that the classifier is *sure* that the produced image belongs to the respective other class. As these strong image modifications point out the relevant features in a very emphasized way, they lead to satisfactory explanations for non-experts that are not familiar with fine details of the problem domain. However, those kind of explanations might not be optimal for expert users, as those could perceive the performed feature translation as an exaggerated modification of the original features. The adaption of our system for an expert system would demand for further modification of our proposed loss function to produce images that are closer to the classifier's decision boundary. We already propose a possible adjustment for this in section 7.2 but did not test this adjustment thoroughly yet. In our work, we presented a use case that was based on a binary classification problem. We want to emphasize that the proposed method can in theory easily be extended to a multi-class classification problem. In order to do so, multiple CycleGAN models have to be trained. When dealing with k classes $\{S_1, ..., S_k\}$, for every pair of classes (S_i, S_j), with $i \neq j$, a CycleGAN has to be trained to solve the translation task between domain S_i and S_j, resulting in $\frac{k!}{2(k-2)!}$ models. While there is conceptually not a problem with this, the training of a huge number of models in practice can become a challenge due to limited resources. Thus, we see the application of our approach rather in explaining classifiers that do not deal with too many different classes. A further question that arises when dealing with a multi-class problem is the choice of the classes for which a counterfactual image is generated. A straight-forward solution to this is to simply generate counterfactual explanations for all classes. Another way - that is more feasible for problems with a huge number of classes - is to pick the counterfactual classes according to the class probability that was attributed by the classifier.

In our chosen use case, relevant information is mainly contained in textural structures. Therefore, we cannot make a general statement about how the approach would perform in different scenarios where information is more dependent on non-textural information, e.g., occurrence or location of certain objects. However, we plan to address this question in future research by applying our approach to different scenarios.

Further, medical research often uses 3D data. Future work has to investigate if our GAN-based approach can be modified to cope with 3D structures (e.g., MRT data) in order to cover a wider range of practical scenarios.

7.7 CONCLUSION

In this chapter, we introduced a novel approach for generating counterfactual explanations for explaining image classifiers.

Our computational comparison between counterfactuals generated by an original CycleGAN and a CycleGAN that was modified by our approach showed that our introduced loss component forces the model to predominantly generate images that were classified in a different way than the original input, while the original CycleGAN performed very poorly in this respective task. Thus, the introduced modification had a substantially positive impact generating counterfactual images.

Furthermore, we conducted a user study to evaluate our approach and compare it to two state-of-the-art XAI approaches, namely LIME and LRP. As evaluation use case, we chose the explanation of a classifier that distinguishes between x-ray images of lungs that are infected by pneumonia and lungs that are not infected. In this particular use case, the counterfactual approach outperformed the common XAI techniques in various regards. Firstly, the counterfactual explanations that were generated by our system led to significantly more satisfying results as the two other systems that are based on saliency maps. Secondly, the participants formed significantly better mental models of the AI based on our counterfactual approach than on the two saliency map approaches. Also, participants had more trust in the AI after being confronted with the counterfactual explanations than with the LRP condition. Furthermore, users that were shown counterfactual images felt less angry and more relaxed than users that were shown LRP images.

All in all, we showed that our approach is very promising and shows great potential for being applied in similar domains.

However, it has to be investigated further how the system performs in other use cases and modalities. We believe that the advantage of our system in this specific scenario results from the relevant information of the images being of a rather textural structure, e.g., opacity. Thus, raw spatial information about important areas, as provided by LIME and LRP, does not carry enough information to understand the AI's decisions. Therefore, we recommend the application of our approach in similar use cases, where relevant class-defining features are expected to have a textural structure.

8

THE CONCEPT OF ALTERFACTUAL EXPLANATIONS

Large parts of this chapter have already been published in the following publication:

> *Mertes, S., Karle, C., Huber, T., Weitz, K., Schlagowski, R., & André, E. (2022). Alterfactual Explanations - The Relevance of Irrelevance for Explaining AI Systems. In IJCAI 2022 Workshop on XAI.* (Mertes, Karle, et al., 2022)

Counterfactual explanations show a version of the input data that is altered just enough to change an AI's decision. By doing so, the user is shown not only *which* features are relevant to the decision, but more importantly, *how* they would need to be changed to result in a different decision of the AI. Semifactual explanations follow a similar principle, but they modify the relevant features of the input data to an extent that the AI's decision does not change just yet.

Both methods have in common that they focus on the *important* features. However, awareness of irrelevant features can also contribute substantially to the complete understanding of a decision domain, as knowledge of the important features for the AI does not necessarily imply knowledge of the unimportant ones.

For example, if we want to investigate whether an AI system is subject to some bias regarding its predictions, we often want to know explicitly whether a particular feature is completely irrelevant to a classifier. As a concrete example, consider an AI system that assesses a person's creditworthiness based on various characteristics, and we want to study that system regarding its fairness. If that system was completely fair, a counterfactual explanation would be of the form: *If your income was higher, you would be creditworthy.* However, this explanation does not exclude the possibility that your skin color also influenced the AI's decision. It only shows that the income had a high impact on the AI. An explanation confined to the irrelevant features, on the other hand, might say *No matter what your skin color is, the decision would not change.* In this case, direct communication of irrelevant features ascertains that the system is fair with regards to skin color.

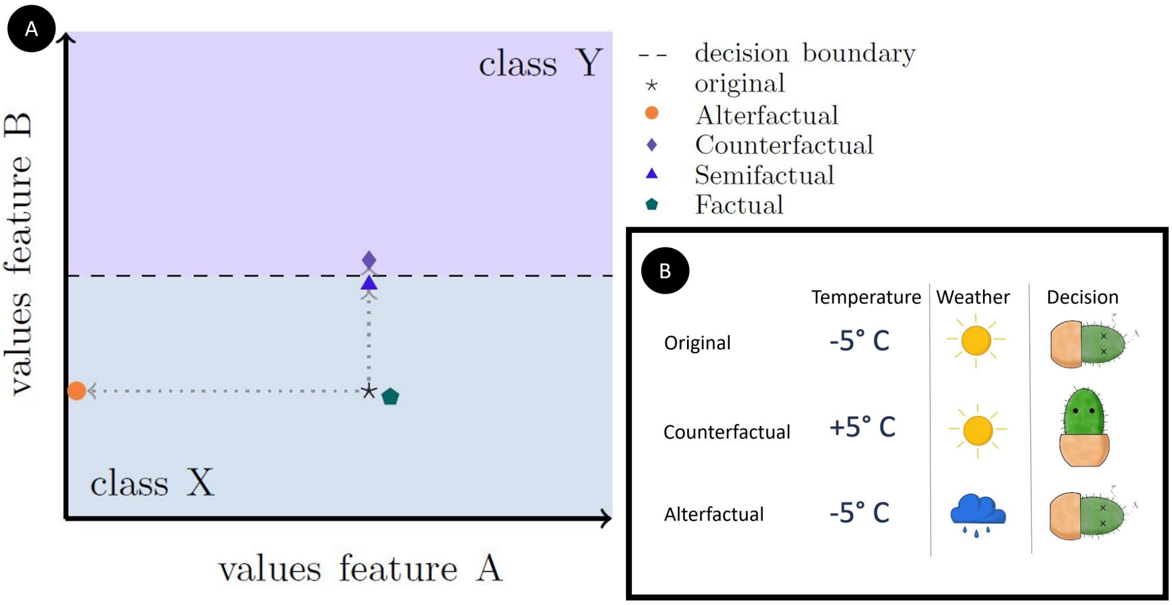


Figure 45: (A) Conceptual comparison of factual, counter-, semi-, and alterfactual explanations. The diagram shows the original input which is to be explained below the decision boundary belonging to class X. A factual explanation could be the nearest neighbor, located anywhere around the original input. A semifactual explanation would be located in minimal distance directly next to the decision boundary, but still below it. A counterfactual explanation would be above it in the region of class Y, but barely so. An alterfactual explanation would move in parallel to the decision boundary, indicating which feature values would not modify the model's decision. Note that this diagram is highly simplified - normally, there are more than two features, the decision boundary is more complex, etc. (B) Examples of a counterfactual and an alterfactual explanation. Input features to a fictional decision system to be explained are *temperature* and *weather*, whereas the former is relevant and the latter is irrelevant to the AI's decision on whether a cactus survives or not.

Conventional counterfactual thinking explanation paradigms do not provide this information directly.

To address this issue, this chapter introduces and evaluates a novel explanatory paradigm. We call explanations that follow this paradigm *Alterfactual Explanations*. This principle is based on showing the user of the XAI system an alternative reality that leads to the exact same decision of the AI, but where only irrelevant features change. All relevant features of the input data, on the other hand, remain the same. As this type of explanation conveys completely different information than common methods, we investigate whether the mental model that users have of the explained AI system is also formed in a different way, or can even be improved. We show that the communication of features unimportant to the decision contributes significantly to the understanding and formation of a mental model of AI systems.

Note, that in this chapter, we focus on the *concept* of these new explanation paradigm. *Generating* such alterfactual explanations will be the topic of the *next* chapter. By isolating concept and generation, we want to make sure that the findings of this chapter's study are not biased (either positively or negatively) by the specific generation approach.

8.1 IDEA

The basic idea of alterfactual explanations introduced in this chapter is to strengthen the user's mental model of an AI by showing irrelevant attributes of a predicted instance. Hereby, we understand irrelevance as the property that the corresponding feature, regardless of its value, does not contribute in any way to the decision of the AI model. When looking at models that are making decisions by mapping some sort of input data $x \in X$ to output data $y \in Y$, the so-called *decision boundary* describes the region in X which contains data points where the corresponding y that is calculated by the model is ambiguous, i.e., lies just between different instances of Y. Thus, irrelevant features can be thought of as features that do not contribute to a data point's distance to the decision boundary.

On the other hand, the information that is carried out by an explanation should be communicated as clearly as possible. As the information that is contained in an alterfactual explanation consists of the *irrelevance* of certain features, it should somehow be emphasized that these features can take *any* possible value. If we would change the respective features only to a small amount, the irrelevance is not clearly demonstrated to the user. Therefore, we argue that an alterfactual explanation should change the affected features to the maximum amount possible. By doing so, we communicate that the feature, *even* *if it is changed as much as it can change*, still does not influence the decision.

We take those two considerations as the base for the definition of an alterfactual explanation:

> Let $d : X \times X \to \mathbb{R}$ be a distance metric on the input space X. An *alterfactual explanation* for a model M is an altered version $a \in X$ of an original input data point $x \in X$, that maximizes the distance $d(x, a)$ whereas the distance to the decision boundary $B \subset X$ and the prediction of the model do not change: $d(x, B) = d(a, B)$ and $M(x) = M(a)$

Thus, the main difference between an alterfactual explanation and a counterfactual or semifactual explanation becomes clear: While the latter methods alter features resulting in a decreased distance to the decision boundary, the former method tries to keep that distance fixed. Further, while counterfactual explanations as well as semifactual explanations try to keep the overall change to the original input minimal (Keane, Kenny, Delaney, et al., 2021; Kenny and Keane, 2020), alterfactual explanations do exactly the opposite, which is depicted in Figure 45A. Figure 45B illustrates the difference between counterfactual and alterfactual explanations using a simple example.

8.2 USER STUDY

In order to validate if our approach of focusing only on irrelevant features for explaining an AI system helps users to form correct mental models of the system, we performed an online user study. Prior to the real study, a pilot study (n=14) was conducted to find out whether subjects could cope with the tasks. Note that in this study, we only focus on the *concept* of Alterfactual Explanations. The *generation* of such explanations will be subject of the next chapter. As our concept of communicating only irrelevant features of an AI's decision is entirely new, we wanted to study its validity unbiased from specific technical implementations.

8.2.1 *Hypotheses*

As the concept of alterfactual explanations can be considered a counterfactual thinking approach, we did not only want to validate the feasibility of alterfactual explanations per se. While traditional counterfactual explanations communicate information about *relevant* features, alterfactual explanations communicate information about *irrelevant* features. As such, we argue that counterfactual and alterfactual explanations might complement each other. Therefore, we also investigate how alterfactual explanations perform in comparison to such traditional counterfactual explanations. Furthermore, as we claim that counterfactual and alterfactual explanations convey different kinds of information, we also included the *combination* of the two approaches. We suspect that this combination could lead to even better explanations. To gain a decent understanding of the advantages and disadvantages of alterfactual explanations, we took different metrics into account that are commonly used to assess XAI systems. Concretely, we evaluated the explanation satisfaction as well as the mental model creation capabilities of the shown explanations. Thus, our hypotheses are as follows:

1. Mental Model Creation
   a) Alterfactual explanations lead to a more correct mental model of the AI than no explanations.
   b) Alterfactual explanations lead to similarly good mental models as counterfactual explanations.
   c) The combination of alterfactual and counterfactual explanations outperform both alterfactual as well as counterfactual explanations in terms of mental model creation.
2. Explanation Satisfaction
   a) Alterfactual explanations lead to a similarly good explanation satisfaction as counterfactual explanations (i.e., we do *not* expect to find significant differences here).
   b) The combination of alterfactual and counterfactual explanations outperform both alterfactual as well as counterfactual explanations in terms of explanation satisfaction.

8.2.2 *Methodology*

In order to test the hypotheses stated above, an online user study was conducted. We used a between-subject design with four conditions:

- **Alterfactual condition.** Participants in that condition were presented with original input features to an AI as well as alterfactual explanations.
- **Counterfactual condition.** Participants in that condition were presented with the original features as well as the counterfactual explanations.
- **Combination condition.** Participants in that condition were presented with the original features as well as both the alterfactual and the counterfactual explanations.
- **No Explanation condition.** Participants in that condition were presented only with the original features. No explanation was shown.

Similarly to our study presented in Chapter 7, a between-subject study design was chosen because we wanted to avoid order effects and mitigate the risk of fatigue. In the study, the participants were presented with an imaginary AI. The participants were told that the AI decides if hypothetical historical documents are forged or not. This specific scenario was chosen as it is not present in most people's everyday life, ensuring that the mental model of the AI that the participants develop is predominantly induced by the explanations that they are presented with during the study and do not stem from prior knowledge of the domain. The AI gets different inputs to work with. We designed the imaginary AI so that it follows a set of rules (unknown to the participants), where each input feature has a specific relevance to the AI. Those features are as follows:

- **Parchment Color.** The documents can be either of *light*, *medium* or *dark* parchment.
- **Word Count.** A single integer in the range $[1, 500]$.
- **Year of Creation.** The documents were created sometime between 200 BC and 200 AD.

The rules which the fictional AI uses to decide if a document is forged are:

- A document is forged if the word count is equal to or below 50.
- A document is forged if the word count is between 51 and 150 *and* the parchment color is light or medium.
- In all other cases, the document is considered to be authentic

Therefore, one attribute is always relevant (word count), one is relevant only in some cases (parchment color), and one is always irrelevant (year of creation). After answering some questions about their demographic background, the participants were given some general information about the data and AI used in the experiment. They were told that some historical documents had been found, and some of them had already been identified as forgeries. Futhermore, they were told that an AI had been trained to detect forgeries based on a short description of the documents containing the three attributes mentioned above. The three attributes were shown along with their value ranges. An exemplary input to the fictional AI was displayed in a table. Additionally, we explained which explanation type the participant was going to be shown during the study, and how the explanation type works. The participants were provided with example explanations that could be revealed by clicking a button. After using that button, the explanations were shown next to the original input. An example explanation is shown in Figure 46. Following this introduction, the participants were given two example inputs and corresponding explanations in order to familiarise themselves with the document descriptors and the mechanism to reveal the explanations. After that, each participant was quizzed about the information that was given up to that point. By doing so, we could exclude subjects who did not conscientiously participate in the study. After the quiz, a short training phase followed. In this phase, the participants were shown four exemplary document descriptors. Explanations for the AI's decisions, as well as the decisions itself were shown as well. The training phase was conducted to give the participants another chance to get comfortable with the explanation type and the domain itself. Subsequently, the study itself started. It was divided into three parts: For assessing the participants' mental model of the AI, we used *(i)* a prediction task and *(ii)* a questionnaire about the AI's rule set. To assess the participants' explanation satisfaction, we used *(iii)* an explanation satisfaction questionnaire.

8.2.2.1 *Mental Model Creation (i): Prediction Task*

The goal of the prediction task was to detect how well the participants could anticipate the classifier's decisions, which provides a quick window into how well they *understood* the AI (Hoffman et al., 2018). To this end, eight example inputs with explanations were shown in a random order. Four examples were classified as *forged* by the AI, whereas four examples were predicted as being *authentic*. As proposed by Hoffman et al. (2018), the decision of the AI was *not* shown, but had to be predicted by the participants. The idea of such a prediction task is that a good explanation should help to build a correct mental model of the AI, allowing to understand its decision process to an extent that those decisions can be predicted by the user. Additionally to the prediction of the AI's decision, participants had to choose how confident they were in their prediction on a 7-point Likert scale (0 = not at all confident, 6 = very confident). Furthermore, they had to justify their prediction in a free text form. Participants that were in the *No Explanation* condition did not see any explanations but had to rely on the original input data for their predictions. For every single prediction task, explanations had to be revealed by pressing the *Explain* button. By doing so, we were able to track if the participants really looked at the explanations.

8.2.2.2 *Mental Model Creation (ii): Understanding Questionnaire*

To assess if the participants developed a correct mental model of the AI's decision process, for each feature (i.e., parchment color, word count, year of creation), they were explicitly asked how much they agreed that it was relevant to the AI's decision on a 5-point Likert scale (0 = strongly disagree, 4 = strongly agree) after completing all predictions of the Prediction Task. Thus, while the Prediction Task can be seen as implicit measurement of the mental model's correctness, our Understanding Questionnaire directly measures if participants understood the relevance of different features.

8.2.2.3 *Explanation Satisfaction*

In order to validate hypotheses 2a and 2b, we used the Explanation Satisfaction Scale proposed by Hoffman et al. (2018) which consists of seven items, rated on a 5-point Likert scale (0 = strongly disagree, 5 = strongly agree).

Finally, the participants had the possibility to give free text feedback. The whole study was built using the *oTree* framework by D. L. Chen, Schonger, and Wickens (2016).

8.2.3 *Participants*

113 Participants between 24 and 71 years (M = 41.2, SD = 10.2) were recruited via Amazon MTurk. 62 of them were male, 48 female, 1 non-binary, and 2 preferred not to answer this question. Only participants with an *MTurk Masters Qualification* were allowed to participate, and subjects that did not pass the quiz were excluded from the study to minimize bias due to inattentive participants. The participants were randomly separated in the four conditions. Subjects of the three explanation conditions that did not look at a single explanation during the whole study were moved to the *No Explanation* condition for evaluation. Participants got paid a base reward of 5.00$ and another 0.50$ for each right prediction in the Prediction Task. By communicating that bonus payment before participation, we wanted to further motivate the participants to stay focused on the study. Only 5.3% of the participants had no experience with AI. Most of the participants (86.7%) have heard from AI in the media. In general, 79.7% of the participants were expecting a positive or extremely positive impact of AI systems in the future.

8.3 RESULTS

8.3.1 *Mental Model Creation*

To investigate the impact of the four different experimental conditions[1] on the (1) understanding and (2) prediction accuracy, we conducted a MANOVA. We found a significant difference, Pillai's Trace = 0.13, $F(6,218) = 2.52$, $p = .022$.

[1] (*Alterfactual* condition, *Counterfactual* condition, *Combination* condition, *No Explanation* condition)

The following ANOVA revealed that only the understanding of the participants showed significant differences between the conditions:

- *Understanding*: $F(3{,}109) = 3.90$, $p = .011$.
- *Prediction Accuracy*: $F(3{,}110) = 2.63$, $p = .217$.

As displayed in Figure 47, the post-hoc t-tests showed that the participants' understanding was significantly better in the *Alterfactual* condition compared to all other conditions. The effect size d is calculated according to Cohen (2013)[2]:

- **alterfactual vs. counterfactual**: $t(109) = 2.58$, $p = .011$, $d = 0.89$ (large effect).
- **alterfactual vs. combination**: $t(109) = 3.11$, $p = .002$, $d = 1.24$ (large effect).
- **alterfactual vs. no explanation**: $t(109) = 2.86$, $p = .005$, $d = 0.82$ (large effect).

The results indicate that alterfactual explanations help participants understand the relevant features more correctly than in all other conditions. Interestingly, the combination of alterfactual and counterfactual explanations leads to a worse performance and understanding by the participants (see Figure 47).

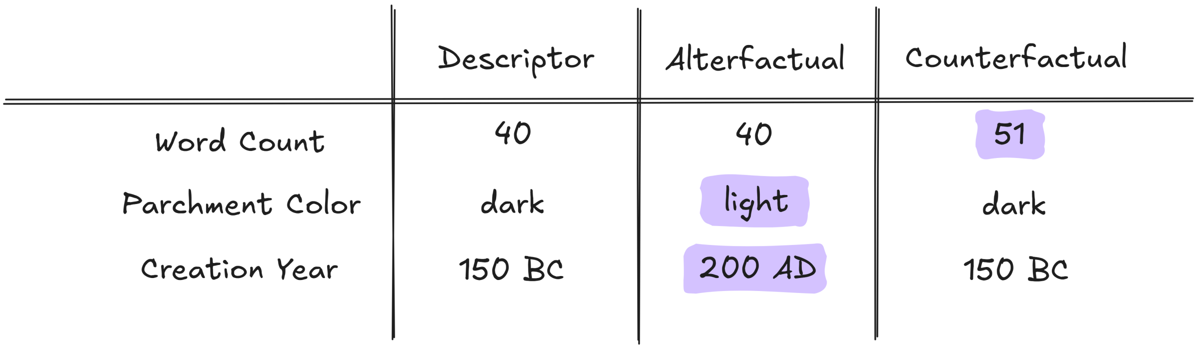
Therefore, hypothesis 1a holds, because alterfactual explanations outperformed the *No Explanation* condition as well as the *Combination* condition. Hypotheses 1b and 1c have to be rejected because alterfactuals explanations also outperfomed counterfactual explanations as well as the combination of both explanation types in the context of mental model creation.

Wondering about the results, especially about the fact that the *No Explanation* condition outperformed the *Combination* condition, we took a closer look, which of the features (i.e., word count, parchment color, year of creation) the participants did or did not understand in each condition. For this, we compared the amount of the correct features between the group, using a MANOVA. We found a significant difference, Pillai's Trace = 0.26, $F(9{,}327) = 3.49$, $p < .001$.

The following ANOVA revealed that only the feature *parchment color* showed significant differences between the conditions:

- *Parchment color*: $F(3{,}109) = 10.49$, $p < .001$.
- *Word count*: $F(3{,}109) = 0.03$, $p = .099$.
- *Creation year*: $F(3{,}109) = 1.22$, $p = .305$.

[2] Interpretation of the effect size is: $d < .5$: small effect; d = 0.5-0.8 : medium effect; $d > 0.8$: large effect



	Descriptor	Alterfactual	Counterfactual
Word Count	40	40	51
Parchment Color	dark	light	dark
Creation Year	150 BC	200 AD	150 BC

Figure 46: A sample document descriptor with explanations. In the *Combination* condition, both an alter- and a counterfactual explanation were shown. Subjects in the *Alterfactual* and *Counterfactual* conditions did not see the respective other explanation type. Subjects in the *No Explanation* condition did not see an explanation at all, but only the original document descriptor.

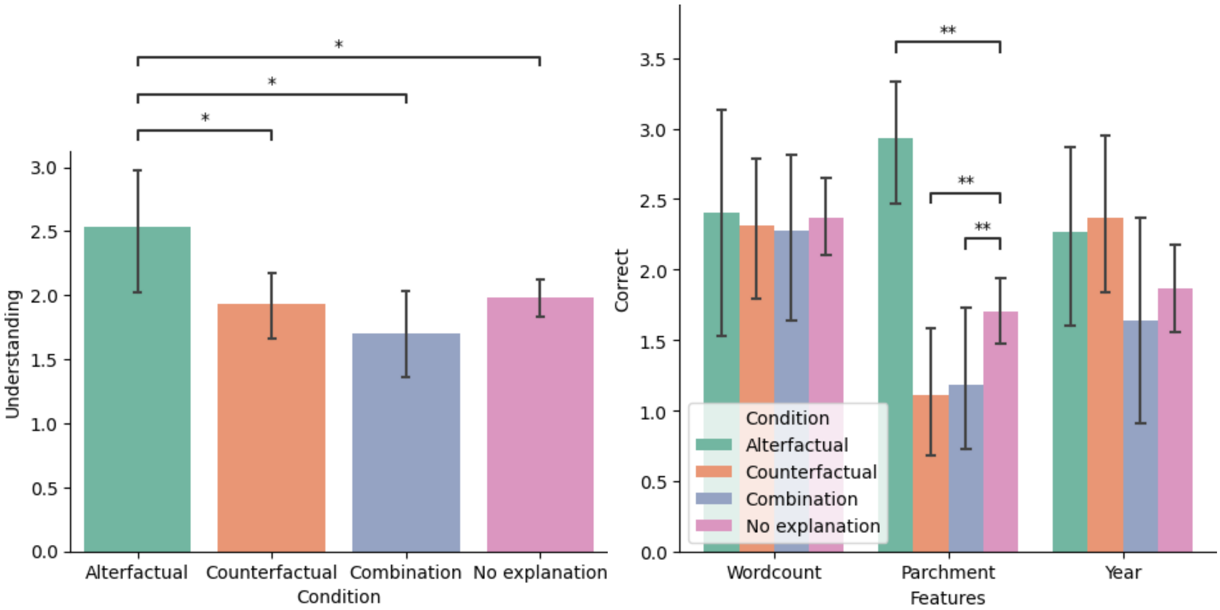


Figure 47: Impact of the four experimental conditions on the understanding of the relevant features of the AI. Alterfactual explanations outperformed all other conditions in helping participants to understand the relevant features of the AI system. Best viewed in color. Error bars represent the 95% CI. *$p < .05$, **$p < .001$.

As displayed in Figure 47, the post-hoc t-tests showed that the participants' correct understanding of the relevance of the parchment color feature were significant better in the *Alterfactual* condition, compared to all other conditions:

- **alterfactual vs. counterfactual**: $t(109) = 5.21$, $p < .001$, $d = 1.80$ (large effect).
- **alterfactual vs. combination**: $t(109) = 4.34$, $p < .001$, $d = 1.72$ (large effect).
- **alterfactual vs. no explanation**: $t(109) = 4.24$, $p < .001$, $d = 1.21$ (large effect).

8.3.2 *Explanation Satisfaction*

The ANOVA revealed that there were no significant differences between the three explanation conditions, $F(2,42) = 1.57$, $p = .219$, indicating that participants felt not specific satisfied by one of the explanation conditions.

Therefore, hypothesis 2b has to be rejected, as the combination of alterfactual and counterfactual explanations does not lead to a higher explanation satisfaction of the participants. Nevertheless, hypothesis 2a holds since the alterfactual explanations do not differ significantly compared to counterfactual explanations.

8.4 DISCUSSION

The results of our user study show novel insights into the explanatory performance of the different XAI approaches.

Alterfactual explanations support global understanding of users.

First of all, although not significantly differing from the other conditions in the Prediction Task, subjects that were provided with alterfactual explanations performed significantly better in the Understanding Task than all other participants. This indicates that direct communication of information about irrelevant features does indeed offer benefits. Contrary to our original assumption, the alterfactual explanations outperformed even the more traditional counterfactual explanations. Different from the Prediction Task, the Understanding Task directly surveys the users' mental models regarding the relevance of the input features. Thus, we argue that alterfactual explanations work better when it comes to the communication of how important different features are for a decision in general, although they do not convey a better understanding of which exact decision will be made when presented with a concrete input sample compared to counterfactual explanations. This suggests that alterfactual explanation could find application in scenarios where a global understanding of the AI system is important. Our investigation of the participants' feature-specific understanding strengthens this assumption: The alterfactual explanations' better performance in the Understanding Task mainly stems from users presented with alterfactual explanations having a significantly better understanding of the importance of the *parchment* feature. As that feature was relevant in some cases and irrelevant in others, understanding its relevance highly depends on global understanding of the model. However, future studies have to be conducted to assess the capability of alterfactual explanations to induce a global understanding of an AI's decision process in a broader scope.

Too many explanations can overstrain users.

Further, it seems very surprising that the combination of alterfactual and counterfactual explanations performs poorly, although they contain more information than any other condition. We assume that this stems from the fact that more information comes with higher demands on the users' *cognitive load*. We argue that the participants in the *Combination* condition were simply overwhelmed by the wealth of information. This finding is in line with cognitive load research that emphasized the fact that too much information can overwhelm users (Sweller, Van Merrienboer, and Paas, 1998). Future research has to find ways to communicate this vast amount of information without overburdening users.

Alterfactual explanations are equally satisfying as counterfactual explanations.

Lastly, we found no significant differences regarding Explanation Satisfaction between the three conditions that were presented with some kind of explanation. We argue again that the combination of counterfactual and alterfactual explanations could have overwhelmed the user. However, we see that alterfactual explanations lead to similarly good Explanation Satisfaction as the traditional counterfactual explanations, making them a viable approach for real-world XAI scenarios.

8.5 CONCLUSION

In this study, we presented a new XAI paradigm that we call *Alterfactual Explanations*. Our approach is based on only communicating information about features that are irrelevant to an AI's decision. A user study that we conducted showed that alterfactual explanations show huge potential for the field of XAI. In an Understanding Task measuring the capabilities of users to tell which features of an example input to an AI are important for its decision, alterfactual explanations significantly outperformed the more traditional counterfactual explanations as well as the combination of alterfactual and counterfactual explanations. Surprisingly, combining counterfactual and alterfactual explanations did not result in more correct mental models. We showed that alterfactual explanations lead to a similar good Explanation Satisfaction as counterfactual explanations.

9

ALTERFACTUAL EXPLANATION GENERATION

Large parts of this chapter have already been published in the following publication:

> *Mertes, S., Huber, T., Karle, C., Weitz, K., Schlagowski, R., Conati, C., & André, E. (2024). Relevant Irrelevance: Generating Alterfactual Explanations for Image Classifiers. In 33rd International Joint Conference on Artificial Intelligence (IJCAI) 2024.* (Mertes, Huber, Karle, et al., 2024)

As we argue that alterfactual and counterfactual explanations convey different information, we designed a generative approach that is capable of creating both types of explanations in order to explain an image classifier. For both, a set of requirements arises that need to be reflected in the objectives of our explanation generation approach.

1. The generated explanations should have high quality and look realistic.

2. The resulting explanation should be either classified as the same class as the original input (for alterfactual explanations), or as the opposite class (for counterfactual explanations).

3. For alterfactual explanations, the output image should change as much as possible, while for counterfactual explanations, it should change as little as possible.

4. For alterfactual explanations, only irrelevant features should change, i.e., the distance to the decision boundary should be maintained.

To address these objectives, different loss components (see next section) were used to steer a GAN-based architecture to generate the desired explanations. A GAN-based approach was chosen as similar concepts have successfully been applied to the task of counterfactual explanation generation in various existing works (Matthew L. Olson et al., 2021; Huber, Demmler, et al., 2023; Nemirovsky et al., 2022; Y. Zhao, 2020) - and also in the approach introduced in Chapter 7.

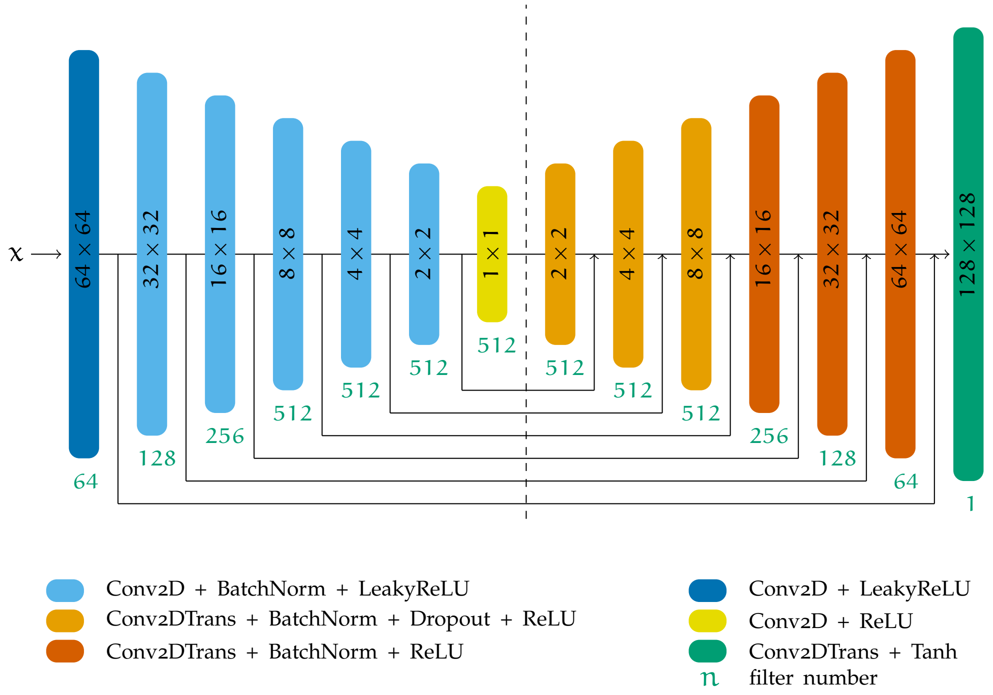


Figure 48: Architecture overview of the generator network.

In order to allow for a more focused and comprehensive user study design, in this work, we focus on explaining a binary image classifier.

A schematic overview of our architecture can be seen in Figures 48 and 49. For a more detailed description, we refer to Appendix D.2.

9.1 APPROACH

9.1.1 *Adversarial Component*

To address the first objective, an adversarial setting is used. Here, a generator network G is trained to take an original image x and a random noise vector z and transforms them into the respective explanation $\hat{x}$. As such, a mapping $\{x, z\} \rightarrow \hat{x}$ is learned by the generator. A discriminator network D is trained to identify the generated images as *fake* images in an adversarial manner.

Additionally, to partly target the second objective, we feed a target class label $\hat{y} \in \{0, 1\}$ to the discriminator. By doing so, the discriminator learns not only to assess if the produced images are real or fake, but also has the capability to decide if an explanation fits the data distribution of the class it is supposed to belong to. A somewhat similar idea was put forth by Sharmanska et al. (2020) within the context of fairness and yielded promising results there. During training, the discriminator is alternately fed with real and fake data. For real data, the

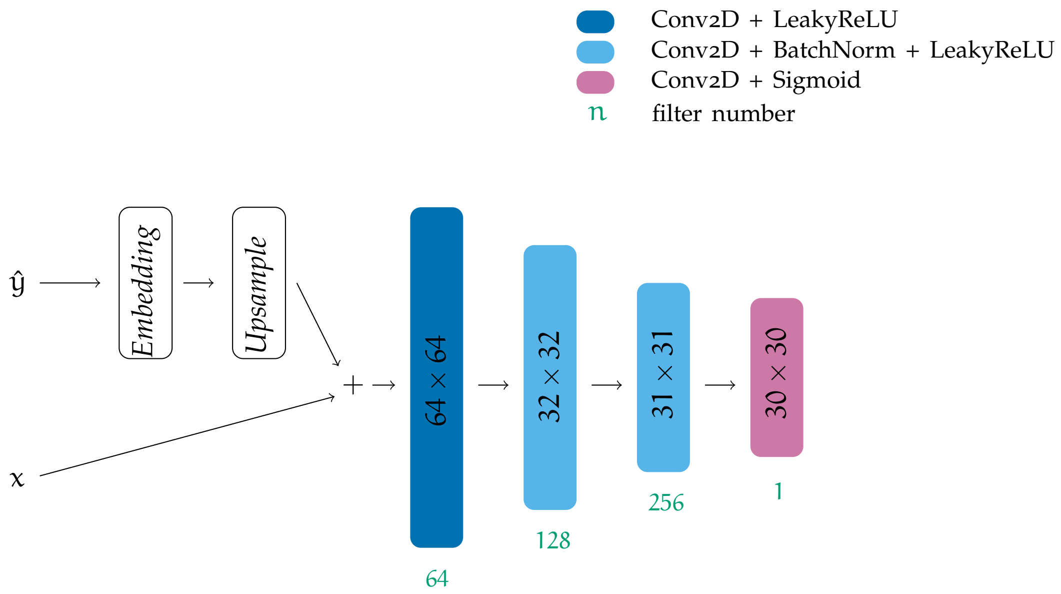


Figure 49: Architecture overview of the discriminator network.

target class label $\hat{y}$ reflects the class that the classifier to be explained assigns to the respective image x. For the generated explanations, the target class label $\hat{y}$ reflects either the class that was assigned to the original image x (for alterfactual explanations), or the opposite class (for counterfactual explanations).

By letting the generator and discriminator compete against each other during training, it is enforced that the resulting images look realistic and resemble the data distribution of the respective target classes. The objective function for the adversarial setting is formulated as follows:

$$\mathcal{L}_{adversarial} = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x, \hat{y})] + \mathbb{E}_{x \sim p_{data}(x), z \sim p_{noise}(z)}[\log(1 - D(G(x, z), \hat{y}))] \quad (21)$$

9.1.2 *Including Classifier Information*

The second objective is further addressed by incorporating the decisions of the classifier to be explained into the generator's loss function.

Let $C : X \to [0, 1]$ be a binary classifier with threshold 0.5. We define the classification target $\tilde{C}(x)$ as $\tilde{C}(x) := C(x)$ for alterfactual explanations and $\tilde{C}(x) := 1 - C(x)$ for counterfactual explanations. To measure the error between the actual classification of the generated explanation and the target classification, we used Binary Crossentropy (BCE) to define a classification loss $\mathcal{L}_C$:

$$\mathcal{L}_C = \mathbb{E}_{x,\hat{x}\sim p_{data}(x,\hat{x})}[\tilde{C}(x) \cdot \log C(\hat{x}) \\ + (1 - \tilde{C}(x)) \cdot \log(1 - C(\hat{x}))] \quad (22)$$

9.1.3 *SSIM Component*

The third objective was addressed by including a similarity component into the loss function. Explanations are meant for humans. Therefore, using the *Structural Similarity Index Measure* (SSIM) seemed to be an appropriate choice to measure image similarity for our approach, as it correlates with how humans are perceiving similarity in images (Z. Wang et al., 2004). The parameters for SSIM were chosen as recommended by Y. Wu et al. (2019).

As alterfactual explanations should change irrelevant features *as much as possible*, while counterfactual explanations should be *as close as possible* to the original image, the learning objective differs for both (low similarity for alterfactual explanations, high similarity for counterfactual explanations). With $[0, 1]$ as the range of SSIM, we designed the loss function as follows:

$$\mathcal{L}_{sim} = \begin{cases} \mathbb{E}_{x,\hat{x}\sim p_{data}(x,\hat{x})}\,[SSIM(x,\hat{x})] & \text{Alterfactual} \\ \mathbb{E}_{x,\hat{x}\sim p_{data}(x,\hat{x})}\,[1 - SSIM(x,\hat{x})] & \text{Counterfactual} \end{cases} \quad (23)$$

9.1.4 *Feature Relevance Component*

The fourth objective, i.e., forcing the network to only modify irrelevant features when generating alterfactual explanations, was addressed by using an auxiliary Support Vector Machine (SVM) classifier. Note that this loss is only applied when generating alterfactual explanations, not when generating counterfactual explanations. Y. Li, L. Ding, and Gao (2018) and Elsayed et al. (2018) have shown theoretically and empirically that the last weight layer of a Neural Network converges to an SVM trained on the data transformed up to this layer. An SVM's decision boundary can be calculated directly - unlike the one of a Neural Network (Yiding Jiang et al., 2018). As such, we use an SVM which was trained to predict the classifier's decision based on the activations of the classifier's penultimate layer as a way to approximate the classifier's decision boundary - if the generated alterfactual explanation has moved closer to the SVM's separating hyperplane, relevant features were most likely modified. Although an unchanged decision boundary distance does not necessarily guarantee that no relevant features were modified, it is a good indicator.

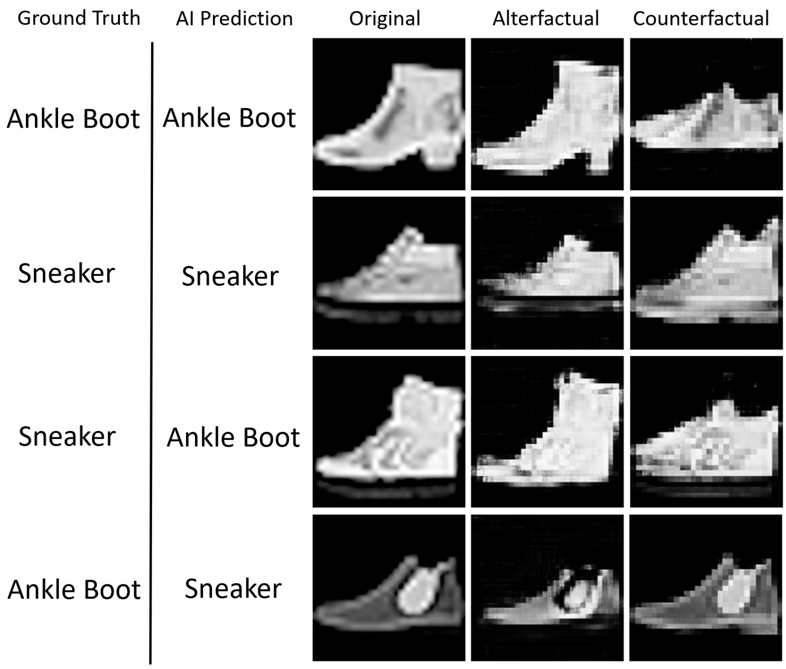


Figure 50: Example outputs of our system. It can be seen that alterfactual explanations change features that are irrelevant to the classifier, e.g., the color of the shoes or the width of the boot shaft, while counterfactual explanations change relevant features like the presence or absence of a boot shaft. From top to bottom the original images are a correctly classified ankle boot and sneaker, followed by two inputs incorrectly classified as ankle boot and sneaker.

The distance of x to the SVM's separating hyperplane f was defined as follows, with w as the SVM's weight vector:

$$SVM(x) = \left| \frac{f(x)}{||w||} \right| \tag{24}$$

The SVM loss is defined by the absolute difference in distance to the separating hyperplane between the original image and the generated alterfactual explanation:

$$\mathcal{L}_{SVM} = \mathbb{E}_{x \sim p_{data}(x), z \sim p_{\text{noise}}(z)} \left[|SVM(x) - SVM(\hat{x})| \right] \tag{25}$$

9.2 EVALUATION SCENARIO

To assess the performance of our approach, we applied it to the Fashion-MNIST data set (Xiao, Rasul, and Vollgraf, 2017). That data set contains 7,000 gray scale images for each of its ten categories of clothes, such as 'ankle boots' or 'pullover', splitted into *train* (6,000 images per class) and *test* (1,000 images per class) sets. The two classes we chose, 'ankle boots' and 'sneakers', were selected due to being somewhat similar in order not to oversimplify the classification task while still being distinct enough to be able to visually assess whether the generated explanations are clear. To create the classifier to be explained, we trained a relatively simple four-layer convolutional neural network, achieving an accuracy of 96.7% after 40 training epochs. The exact architecture and training configuration can be found in Appendix D.2.

Our explanation generation architecture was trained for 14 epochs, until visually no further improvement could be observed. For alterfactual explanations, we reached a validity (i.e., which portion of the explanations are classified as the correct target class by the classifier) of 96.20% and an average SSIM of 0.32 (here, lower is better), whereas the counterfactual explanations reached a validity of 87.70% and an average SSIM of 0.90 (here, higher is better). For more details refer to Appendix D.2. Exemplary generated explanations are shown in Figure 50.

9.3 USER STUDY

9.3.1 *Research Question and Hypotheses*

We conducted a user study to validate whether the counterfactual and alterfactual explanations generated by our approach help human users to form correct model understanding of an AI system. We designed our study similar to the the one presented in the previous chapter. Our hypotheses are as follows:

- Alterfactual and counterfactual explanations, as well as the combination of both, are more effective in enabling model understanding than no explanations.

- There is a difference in model understanding and explanation satisfaction between alterfactual and counterfactual explanations. However, we do not anticipate a specific direction since we see them as complementary concepts (i.e., alterfactual explanations focusing on irrelevant features, counterfactual explanations focusing on relevant features).

- Compared to the individual explanations, a combination of alterfactual and counterfactual explanations is a more effective way to enable a good model understanding and is more satisfying for users.

- There is a difference between conditions regarding the understanding of relevant and irrelevant features, where alterfactual explanations are more effective to identify irrelevant features while counterfactual explanations should help more with identifying relevant features.

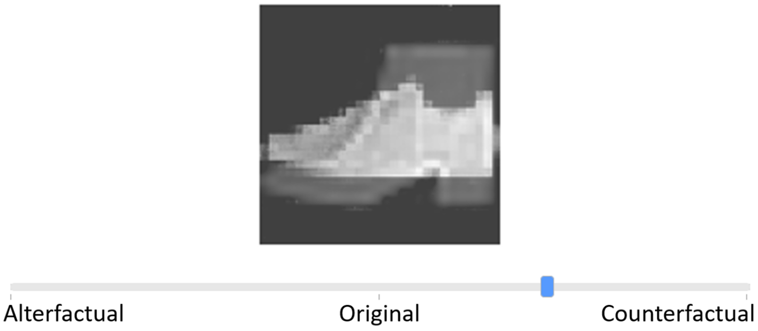


Figure 51: An example of how the explanations were presented in the *Combination* condition during the user study. By moving the slider to either side, the image is linearly interpolated into the counterfactual or alterfactual explanation. The order of the sides was randomized. In the *Alterfactual* and *Counterfactual* conditions, only one side of the slider was present and in the *Control* condition, there was no slider.

9.3.2 *Methodology*

9.3.2.1 *Conditions and Explanation Presentation*

We used a between-groups design with four conditions. Participants in the *Control* condition were presented only with the original input images to the AI. No explanation was shown. In the *Alterfactual* and *Counterfactual* conditions, participants were presented with the original input images and either alterfactual or counterfactual explanations. In the *Combination* condition, participants were presented with the original input images as well as both the alterfactual and the counterfactual explanations. Figure 51 shows how the explanations were presented in each condition.

9.3.2.2 *Procedure*

The whole study was built using the *oTree* framework by D. L. Chen, Schonger, and Wickens (2016). After answering questions about their demographic background, participants were given some general information about the data and their task during the prediction task. For the classifier, they were only told that an AI was trained to distinguish between ankle boots and sneakers. Two example images for each class (ankle boots and sneakers) were shown and some shoe specific terminology (e.g., "shaft") was introduced. Following this information, the participants were given an example input image for each class together with the classifier's prediction for this input image. In the explanation conditions, the participants were introduced to their corresponding explanation types (counterfactuals, alterfactuals or a combination) and could explore the explanations for those two images. After that, each participant answered a quiz about the information that was given up to that point, to make sure that they understood everything correctly. Subsequently, the study itself started. It was divided into three parts: For assessing the participants' understanding of the classifier, we used *(i)* a prediction task for assessing the local understanding, i.e., to assess if the participants understand why the AI makes a *specific* decision, and *(ii)* a questionnaire about the relevance of certain features for assessing the global understanding, i.e., to assess if the participants understand how the AI works *overall*. To assess the participants' explanation satisfaction, we used *(iii)* an explanation satisfaction questionnaire. The three phases of the experiment are described below.

9.3.2.3 *Local Model Understanding: Prediction Task*

To measure the local understanding of the classifier, we used a prediction task, which assesses the participants' ability to anticipate the AI classifier's decisions (Hoffman et al., 2018). Eight examples were shown, covering all possible classification outcomes (two correctly classified images for both sneakers and ankle boots, and two incorrectly classified images for both) to avoid bias. Figure 50 shows four of the images from the study. The example images were chosen randomly but we made sure that the alterfactual and counterfactual explanations generated by our model for those images were valid (i.e., the classifier predicted the same class as for the original image when fed with the alterfactual explanation, and the opposite class when fed with the counterfactual explanation). Participants had to predict the classifier's decision for each example image. Participants were additionally asked about their own opinion on which class the original shoe image belonged to. The answers to that particular question were not further analyzed - it was only added to help the participants distinguish between their own opinion and their understanding of the classifier. After predicting an example, they were told the correct label and the AI classifier's decision before moving on to the next example. The order of the examples was randomized.

9.3.2.4 *Global Model Understanding: Feature Relevance*

While the Prediction Task can be seen as *local* measurement of the users' understanding of the model in specific instances, we also wanted to investigate whether participants understood the *global* relevance of different features. To this end, we looked at two features that were relevant for our classifier ("presence/absence of a boot shaft" and "presence/absence of an elevated heel") as well as two features that were irrelevant for our classifier ("boot shaft width" and "the shoe's color and pattern on the surface area"). These features were chosen based on the authors' experience from training the classifier

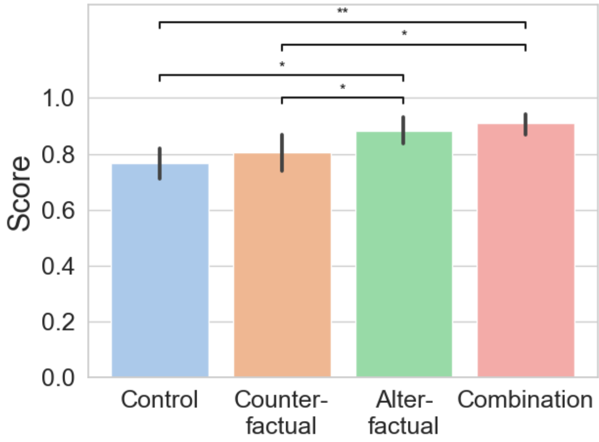


Figure 52: Mean participant prediction accuracy of the AI's prediction by condition. The conditions containing alterfactual explanations outperformed all other conditions. Error bars represent the 95% CI. $*p < .05$, $**p < .001$.

and a-priori explorations with the Feature Attribution explanation mechanisms LIME (Ribeiro, Singh, and Guestrin, 2016) and SHAP (Lundberg and S.-I. Lee, 2017). As such, after the participants went through the eight examples that were used for the prediction task, they were asked for each feature how much they agreed that it was relevant to the AI's decisions on a 5-point Likert scale (0 = strongly disagree, 4 = strongly agree). To aid them in their task, they were again shown the eight example images from the previous prediction task together with the classifier's decisions and the explanations corresponding to their condition.

9.3.2.5 *Explanation Satisfaction*

In order to measure the participants' subjective satisfaction, we used the Explanation Satisfaction Scale proposed by Hoffman et al. (2018) which consists of eight items rated on a 5-point Likert scale (0 = strongly disagree, 5 = strongly agree) that we averaged over all items. Since it does not apply to our use-case, we excluded the 5th question of the questionnaire. The seven remaining items address *confidence*, *predictability*, *reliability*, *safety*, *wariness*, *performance*, *likeability*. Finally, the participants had the possibility to give free text feedback.

9.3.3 *Participants*

Through a power analysis, we estimated a required sample size of at least 21 per condition for a MANOVA with 80% power and an alpha of 5%, based on the Pillai's Trace of 0.13 reported in the previous chapter's study. 131 Participants between 18 and 29 years (M = 22.2, SD = 2.44) were recruited at the University of *blinded for review*. 61

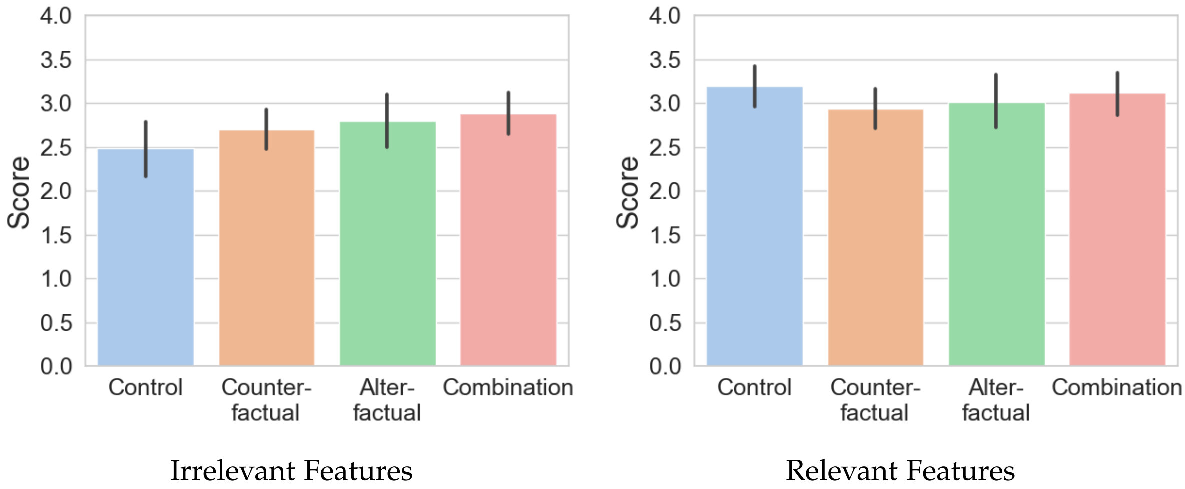


Figure 53: Mean understanding of the irrelevant and relevant features in our study. Error bars represent the 95% CI.

of them were male, 70 female. The participants were randomly separated into the four conditions (33 per condition and 32 in the Alterfactual condition). The highest level of education that most participants held (76.3%) was a high-school diploma. Only 11.5% of the participants had no experience with AI. Most of the participants (74%) have heard from AI in the media. Excluding participants that had no opinion on the subject, the participants expected a positive impact of AI systems in the future (M = 3.73 on a 5-point Likert Scale from 1 = "Extremely negative" to 5 = "Extremely positive"). There were no substantial differences in the demographics between conditions (see Appendix D.2).

9.4 RESULTS

9.4.1 *Model Understanding*

To investigate the impact of the four different experimental conditions on the (1) feature understanding and (2) prediction accuracy, we conducted a MANOVA. We found a significant difference, Wilks' Lambda = 0.859, $F(6,252) = 3.31$, $p = .004$.

The following ANOVA revealed that **only the prediction accuracy of the participants showed significant differences between the conditions**:

- *Feature Understanding*: $F(3,127) = 0.877$, $p = .455$.
- *Prediction Accuracy*: $F(3,127) = 6.578$, $p < .001$.

As displayed in Figure 52, the post-hoc t-tests showed that the participants' prediction accuracy was significantly better in the *Alter-*

factual and *Combination* conditions compared to the other conditions. The effect size d is calculated according to Cohen (2013):

- ***Alterfactual* vs. *Control***: $t(127) = 3.19$, $p = .002$, $d = 0.79$ (medium effect).
- ***Alterfactual* vs. *Counterfactual***: $t(127) = 2.06$, $p = .042$, $d = 0.51$ (medium effect).
- ***Combination* vs. *Control***: $t(127) = 3.93$, $p < .001$, $d = 0.97$ (large effect).
- ***Combination* vs. *Counterfactual***: $t(127) = 2.79$, $p = .006$, $d = 0.69$ (medium effect).

These results regarding the prediction task confirm our hypothesis that the **conditions with alterfactual explanations outperform the condition without explanations in the prediction task.** Further, **the combination of both explanation types did significantly outperform counterfactual explanations.** However, our hypothesis that the combination is more effective in terms of enabling a correct model understanding than alterfactual explanations has to be rejected.

9.4.2 *Relevant and Irrelevant Information*

As reported in the section above, we did not find a significant overall difference in the feature understanding task. However, in order to investigate our hypotheses about irrelevant vs. relevant features, we conducted another MANOVA between the conditions and the combined understanding values for the two relevant features and the two irrelevant features. This MANOVA did not find any significant differences, Wilks' Lambda = 0.951, $F(6,252) = 1.07$, $p = .379$. The mean understanding per condition can be seen in Figure 53.

9.4.3 *Explanation Satisfaction*

The ANOVA revealed that there were no significant differences in the subjective explanation satisfaction between the three explanation conditions, $F(2,95) = 0.34$, $p = .713$. The mean satisfaction values with standard deviation were: *Counterfactual* condition: 3.54 ± 0.53 ; *Alterfactual* condition: 3.65 ± 0.6; *Combination* condition: 3.58 ± 0.5.

9.5 DISCUSSION

With our proposed GAN-based approach, we demonstrated that it is possible to generate both counterfactual and alterfactual explanations for a black box image classifier. Using computational metrics, we showed that both of those generated explanations fulfill their respective requirements: The counterfactual explanations are very similar to the original images (i.e., 0.90 average SSIM) but change the classifiers prediction in 87.70% of the cases while alterfactual explanations are very different from the original image (i.e., 0.32 average SSIM), but do not change the classifier's prediction in 96.20% of the cases.

For the prediction task of our user study, alterfactual explanations and the combination of alterfactual and counterfactual explanations performed significantly better than the other two conditions. However, we did not observe a significant difference for the feature relevance understanding. This is highly interesting, as it contrasts with our previous study. There, a similar experimental design was employed for assessing the effect that alterfactual explanations have on users' mental models of a hard-coded classifier that assesses numerical feature descriptors for a fictional classification problem. In that scenario, alterfactual explanations led to a significantly better feature relevance understanding, while not having a substantial impact on the performance in a prediction task. A possible explanation for this is the fact that in this study, where we used an image classifier in the context of fashion classification, the users might already have had a quite distinctive mental model of the problem domain itself, not only because fashion holds a certain value in peoples' everyday lives, but also because images might be more accessible than numerical feature descriptors to end users. As such, the global understanding of the classifier might already be positively biased. This argument is supported by looking at the feature relevance understanding results of the control group - although not seeing any explanations, they already performed very well in identifying relevant features.

However, as can be seen by the significant performance improvement in the prediction task, the local understanding of the model does not benefit from this effect. As the classification model is imperfect, a global understanding of the use case itself does not necessarily imply an understanding of specific cases, e.g., when the classifier's decision does not correctly model reality.

Interestingly, we did not observe any significant differences in explanation satisfaction. This indicates that participants felt similarly satisfied by all explanation methods even though the alterfactual and combined explanations objectively helped more during the prediction task. The presentation of more information (i.e., in the combination condition) could have led to a higher cognitive load and influenced the subjective assessments of explanation satisfaction, resulting in the difference between objective measurement (i.e., model understanding) and subjective measurement (i.e., explanation satisfaction). Our results motivate future research to include measurements of the need for cognition and cognitive load when investigating counterfactual and alterfactual explanations.

9.6 CONCLUSION

In this chapter, we demonstrated the practical feasibility of *alterfactual explanations*. We show for the first time that it is possible to generate such explanations for black box models. Therefore, we used a DCGAN-based architecture that we enhanced with additional loss components to server our goal. Besides a computational evaluation, we conducted a user study which showed that our generated alterfactual explanations can complement counterfactual explanations.

In that study, we compared how users' model understanding of a binary image classifier changes when being confronted with counterfactual explanations, alterfactual explanations, or a combination of both. Further, a control group was assessed that did not see any explanations.

We found that in a prediction task, where the classifier's prediction had to be anticipated by looking at the explanations, users performed significantly better when they were provided with explanations that included alterfactual explanations compared to users that did not see alterfactual explanations, although we did not observe a significant difference in explanation satisfaction.

Overall, we showed that alterfactual explanations are a promising explanation method that can complement counterfactual explanations in future XAI systems.

Part IV

EXPRESSIVENESS

State-of-the-art generative models are capable of synthesizing new data of high quality. However, such models are data hungry, and training them requires datasets that cover large parts of the problem domain. Highly expressive models - e.g., models that are able to synthesize data with specific characteristics that can be controlled in a detailed way - demand data that is annotated with regards to those features that we want to control. In practice, this is a huge limitation. Although a lot of datasets for a variety of use-cases and scenarios exist, annotations are often only rudimentary. For instance, we often want to be able to *continuously* steer specific features of the generated data, while at the same time, datasets only include *discrete* annotations. In these cases, building expressive generative models poses a problem. In this chapter, it is shown how GANs can be leveraged to counteract this problem. Therefore, we propose an approach that we call *Label Interpolation*. That approach enables the use of GANs to generate continuously controllable outputs while only being trained on categorically annotated data. Therefore, we tackle the exemplary use-case of emotional face synthesis. As such, we show how we can use GANs that were trained on a categorically labeled emotional face dataset to synthesize face images that can be conditioned in a continuous valence/arousal space.

10

LABEL INTERPOLATION

Large parts of this chapter have already been published in the following publications:

> *Mertes, S., Lingenfelser, F., Kiderle, T., Dietz, M., Diab, L., & André, E. (2021). Continuous emotions: exploring label interpolation in conditional generative adversarial networks for face generation. In International Conference on Deep Learning Theory and Applications.* (Mertes, Lingenfelser, et al., 2021)

> *Mertes, S., Schiller, D., Lingenfelser, F., Kiderle, T., Kroner, V., Diab, L., & André, E. (2023). Intercategorical Label Interpolation for Emotional Face Generation with Conditional Generative Adversarial Networks. In International Conference on Deep Learning Theory and Applications - Revised and Selected Papers.* (Mertes, Schiller, et al., 2023)

During the course of this thesis, we have already seen examples for the capability of GANs to create high-quality images. Also, in Section 2.8.2, we have introduced how GANs can be controlled by using additional conditioning information. However, when training such models for a specific, it is mandatory that respective datasets that are available - datasets, where the desired features to be controlled are annotated. However, often we do not have such datasets. Specifically when aiming for a continuous controllability, i.e., when the scales of the features to be controlled are continuous and not discrete, respective datasets are rare - and building such datasets is time and resource intensive. On the other hand, datasets that were annotated with categorical labels are to be found more frequently - simply as annotating datasets in a categorical manner is a much easier and faster task.

In this chapter, we explore the applicability of *Label Interpolation* for Conditional GANs that were trained on categorical datasets. By doing so, we study the possibility to bypass the need for continuously labeled datasets when synthesizing images that should show continuously scaled traits. Since categorical labels are essentially binned versions of continuous labels, we assume that the samples belonging to a specific categorical label are covering a large spectrum of expressiveness. We believe that this information can be learned by a generative model and being exploited to create images on a continuous scale by interpolating through the conditioning space of a trained model. The goal of this chapter is therefore to answer the question of whether label interpolation can be a tool to overcome the drawbacks of categorical datasets for image synthesis. To explore the feasibility of our hypothesis, we first train cGANs on two datasets widely used for benchmarking various deep learning tasks, namely CIFAR-10 (Krizhevsky and G. Hinton, 2009) and Fashion-MNIST (Xiao, Rasul, and Vollgraf, 2017). Those datasets contain discrete class labels that we use for conditioning the GAN. We then examine the effects of interpolating between those discrete class labels by observing how a pre-trained classifier behaves when looking at continuously interpolated results. From the insights gained from these more generic datasets, we tackle the exemplary but concrete use-case of emotional face generation. Most datasets suitable for training such face generation GANs refer to categorical emotion models, i.e., they contain emotion labels that were annotated in a discrete way. This means, that the annotated emotions refer to emotional states like happy, angry or sad. However, for many real-world use cases, corresponding face images need to be generated in a more detailed manner to improve the credibility and anthropomorphism of the results. By applying our approach to that scenario, we enable the cGAN to generate faces showing emotional expressions that can be controlled in a continuous, dimensional way.

10.1 RELATED WORK

As already introduced in 2, *Conditional GANs* (cGAN) can be used to encode label information in the input vector, enabling the GAN to consider certain pre-defined features in the output. This property of cGANs was exploited by Y. Wang, Dantcheva, and Bremond (2018) and Gauthier (2014) to generate face image data with respect to specific features (e.g. *glasses*, *gender*, *age*, *mouth openness*). Similarly, Yi, Sun, and S. He (2018) made use of the cGAN conditioning mechanisms in order to augment emotional face image datasets. One problem of this approach is that they either use discretely labeled features, restricting the output to discrete categories, or they already use continuously labeled data during training which is rarely available in a plethora of scenarios.

A related task that GANs are frequently applied to is the task of *Style Conversion*, which in terms of facial expressions is also known as *Face Editing*. It intends to modify existing image data instead of generating entirely new data (Z. He et al., 2019; Royer et al., 2020; Y. Choi et al., 2018; M.-Y. Liu, Breuel, and Kautz, 2017; J. Lin et al., 2018). Using GANs, H. Ding, Sricharan, and Chellappa (2018) managed to develop a framework that allows to continuously adapt the emotional expressions of images. Although their approach is not explicitly based on continuously annotated data, the diversity of the intensity of emotions must be represented in the training set. Their system proved its capability of generating random new faces expressing a particular emotion. However, they didn't investigate the generation capabilities of their system according to common known dimensional emotion models like Russel's Valence-Arousal model (J. A. Russell and Barrett, 1999). The focus was rather to show that their face editing system is able to modify the intensity of discrete, categorical emotions.

In general, although interpolating through the *input* space of a GAN is common practice (see Chapter 2), interpolating through the *label* space of a cGAN is a quite under-explored mechanism.

To the best of our knowledge, there is no system that is trained on discrete emotion labels and outputs new face images that can be controlled in a continuous way.

10.2 APPROACH

In order to explore the applicability of label interpolation in cGANs, an appropriate framework had to be defined, which is presented in the following sections.

10.2.1 *Network Architecture*

The networks utilized in our experimental settings are largely founded on a *Deep Convolutional GAN* (DCGAN) (Radford, Metz, and Chintala, 2015). A detailed description of the original DCGAN architecture can be found in the respective publication. Additionally, to enable targeted image generation (which is not part of the original DCGAN), the architectures were extended with the principles of a *cGAN*.

Unlike conventional GANs, cGANs incorporate a conditioning mechanism consisting of an additional class input vector. This vector is used to control specific features of the output images by telling the generator network about the presence of certain features during training. Thus, this feature information must be given as labels while training the cGAN. As such, the input for a cGAN consists of a random noise component z (as in the original GAN framework) and a conditioning vector v. After the training process, the generator has learned to transform the random noise input into images that resemble the training domain, taking into account the conditioning information given by v in order to drive the outputs to show the desired features. In the context of emotional face generation, the random noise component is responsible for the face itself, while the conditioning information leads to specific emotions of the face. Therefore, two identical noises conditioned with different feature information should result in the same face showing different emotions.

In our implementation, the conditioning information is given to the network as one-hot encoded label vector, where each element represents a certain feature. Thus, the one-hot label vector v has the following form:

$$v = (v_1, v_2, ..., v_n) = \{0, 1\}^n \tag{26}$$

where n is the number of controlled features. The datasets that we used in our experiments are primarily designed for classification tasks. This implies that we consider a feature a class of the dataset. As in the scope of this work only datasets for single-class classification were considered, the following restriction holds true:

$$\sum_{i=1}^{n} v_i = 1 \tag{27}$$

10.2.2 *Interpolation*

After training, the definition of the condition part of the cGAN's input vector is changed to allow for a continuous interpolation between the originally discrete classes. Generally, this can simply be done by reformulating the conditioning vector v so that is not forced to a binary structure:

$$v = (v_1, v_2, ..., v_n) = [0, 1]^n \tag{28}$$

During our experiments, we found that keeping the restriction formulated in Equation 27 leads to better quality of interpolated results instead of picking the single elements of the vector arbitrarily in the interval $[0, 1]$. In other words, interpolation is done by subtracting some portion e from the input representative of one class and adding it to another class. Our hypothesis is that due to the differentiable function that is approximated by the cGAN model during the training process, those non-binary conditioning vectors lead to image outputs which are perceived as lying somewhere *between* the original, discrete classes. For our target context, the generation of face images with continuous emotional states, this would refer to images of faces that do not show the extreme, discrete emotions that are modeled in a categorical emotion system, but to more fine-grained emotional states as they are conventionally modeled by a dimensional emotion model as will be further elaborated on in Section 10.4.1.

10.3 FEASIBILITY STUDIES

To evaluate the feasibility of our approach, we decided to first apply it to two generic datasets, before finally addressing the problem of emotional human face generation.

10.3.1 *Datasets*

The Fashion-MNIST dataset (Xiao, Rasul, and Vollgraf, 2017) (which we already used in Chapter 9 for generating Alterfactual Explanations) encompasses a set of product pictures taken from the Zalando website, where each image belongs to one of 10 classes. Each of these contains 7,000 pictures. The images that we used are 8-bit grayscale versions with a resolution of 28x28 pixels. All in all, this results in a dataset of 70,000 fashion product pictures, whereas 60,000 are attributed to the training dataset and 10,000 to the test set. Examples for each class are shown in Figure 54.

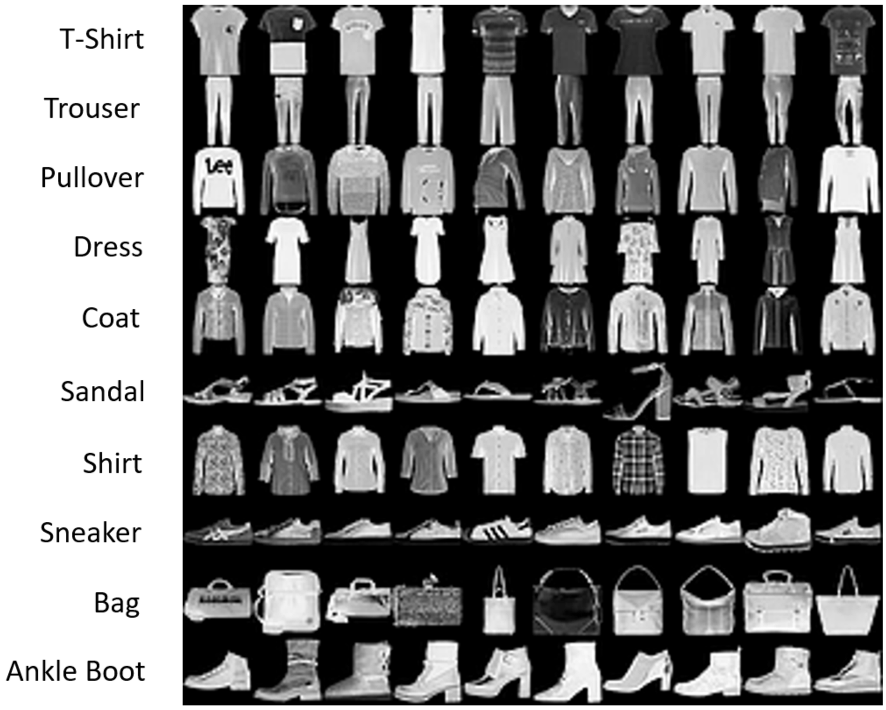


Figure 54: Fashion-MNIST categories and examples (Xiao, Rasul, and Vollgraf, 2017).

The CIFAR-10 and the CIFAR-100 datasets both are derived from the *80 million tiny images dataset* (Krizhevsky and G. Hinton, 2009). In contrast to the 100 classes of CIFAR-100, CIFAR-10 only contains a subset of 10 classes, whereas each class has 6,000 colored images of size 32x32. This results in a dataset of 60,000 images in total, where 50,000 belong to the training and 10,000 to the test set. The classes are mutually exclusive, even for narrow classes like trucks and cars. Figure 55 depicts example images for the corresponding 10 classes.

We decided to use the Fashion-MNIST dataset because it has originally been designed for measuring the performance of machine learning approaches. The pictures are gray-scaled and comparably small, making the dataset suitable for preliminary feasibility experiments.

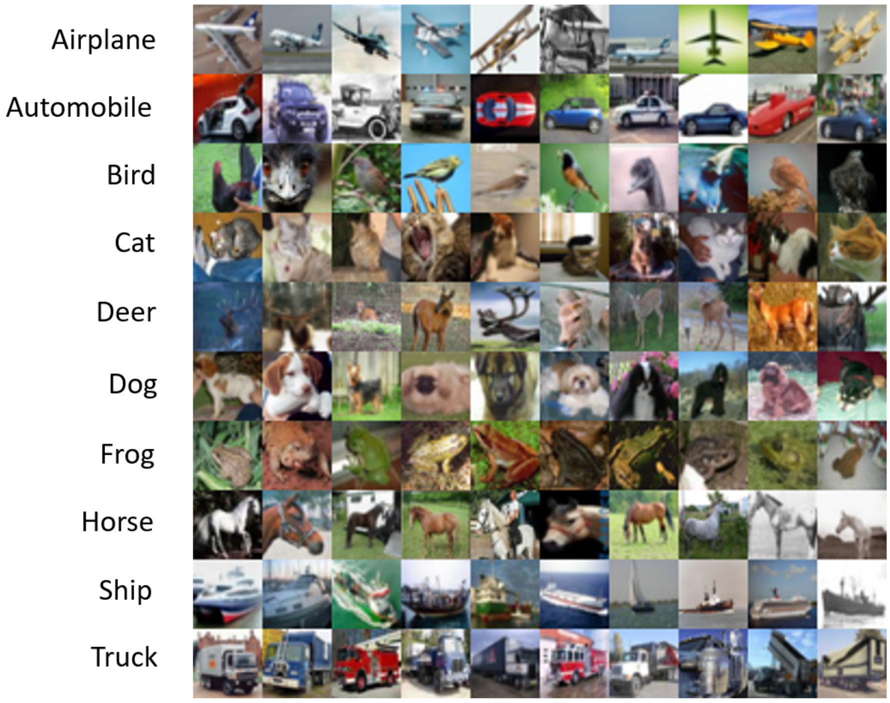


Figure 55: CIFAR-10 categories and examples. (Krizhevsky and G. Hinton, 2009)

To further test the viability of our approach, we aimed to increase the challenge gradually. Thus, we additionally chose to use the CIFAR-10 dataset. Although it also contains small pictures, the challenge is raised by the colorization and the slightly higher resolution.

10.3.2 *Methodology*

In order to evaluate if the interpolation algorithm creates smooth transitions between two arbitrary classes, we decided to perform a fine-grained analysis on the continuously generated outputs by the use of our approach. To this end, we used pre-trained classifiers that are able to accurately distinguish between the different discrete classes contained in the respective datasets. As the focus of this work is to gain insights into the question whether interpolating between discrete label information can be a promising tool for future applications, the discrete decisions of such classification models are not a good metric for our purposes. Instead, we want to explore if the interpolation mechanism is able to model the full bandwidth of transitional states that can occur *between* different classes. Thus, for evaluating if the interpolation mechanism works correctly, we assessed the confidence of the classification models that the interpolated result belongs to certain classes. Ideally, during interpolation, this confidence should continuously shift towards the class that is interpolated to.

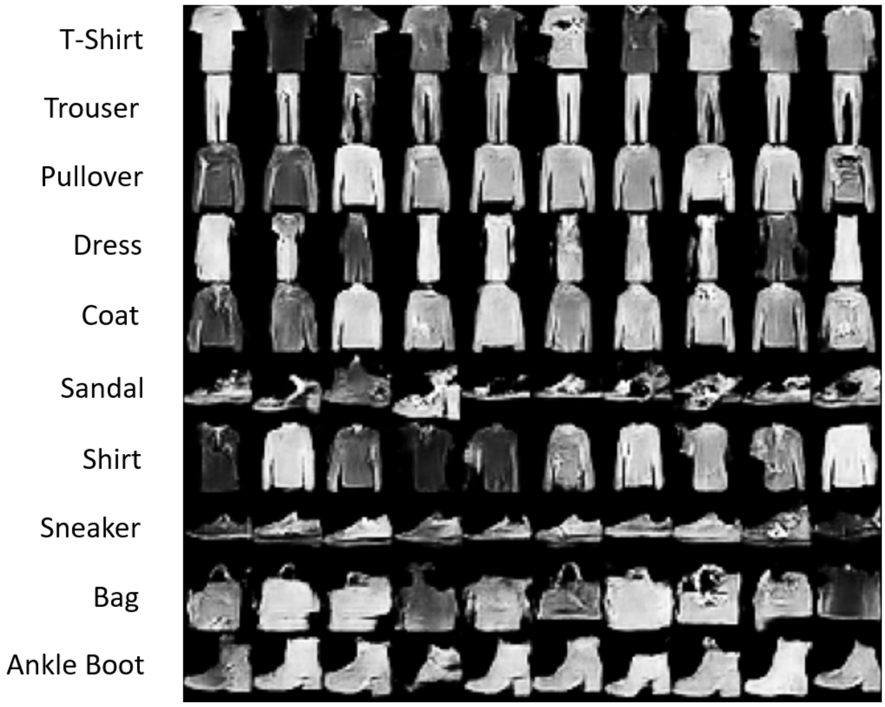


Figure 56: Exemplary outputs of the cGAN model trained on Fashion-MNIST.

10.3.3 *Training*

For both the datasets, we adapted the DCGAN architecture to fit the dataset. Slight changes to the architecture had to be made in order to produce reasonable outputs. Further, we enhanced both models with the conditioning mechanism as described in Sec. 10.2.1.

Fashion-MNIST. For this dataset, we trained the cGAN model for 20,000 random batches of size 32 on all of the 50,000 images of the *train* partition of the dataset using Adam optimizer with a learning rate of 0.0002 and β_1 of 0.5. Example outputs of the trained model can be seen in Figure 56, whereas example outputs of different interpolation steps are shown in Figure 58.

CIFAR-10. For this dataset, we trained the cGAN model for 30,000 random batches of size 32 on all of the 50,000 images of the *train* partition of the dataset, again using Adam optimizer with a learning rate of 0.0002 and β_1 of 0.5. Example outputs of the trained model can be seen in Figure 57, whereas example outputs of different interpolation steps are shown in Figure 59. In both images, it can be clearly seen that the chosen cGAN architecture apparently was not able to resemble the traing domain sufficiently enough. Results are blurry, and objects can only partially be recognized as the intended objects. However, we chose to continue with the validation of the interpolation as we were also interested in how label interpolation behaves when deal-

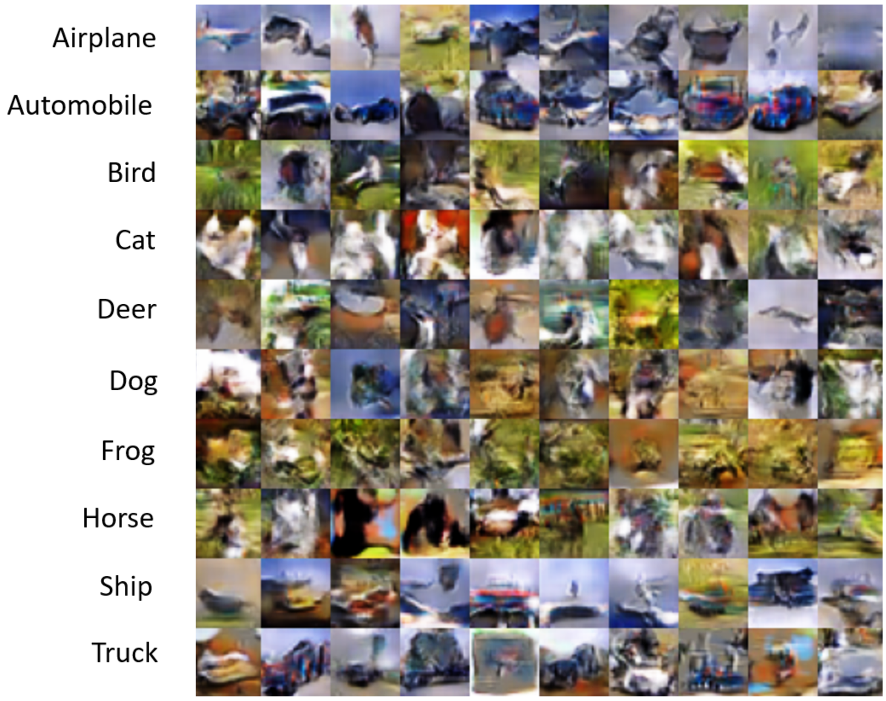


Figure 57: Exemplary outputs of the cGAN model trained on CIFAR-10.

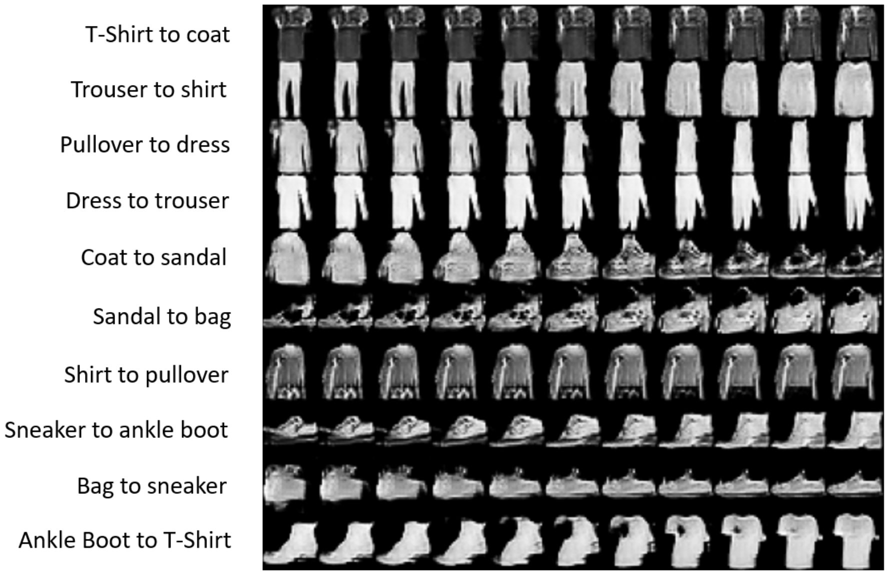


Figure 58: Exemplary outputs of the interpolation steps of the cGAN model trained on Fashion-MNIST.

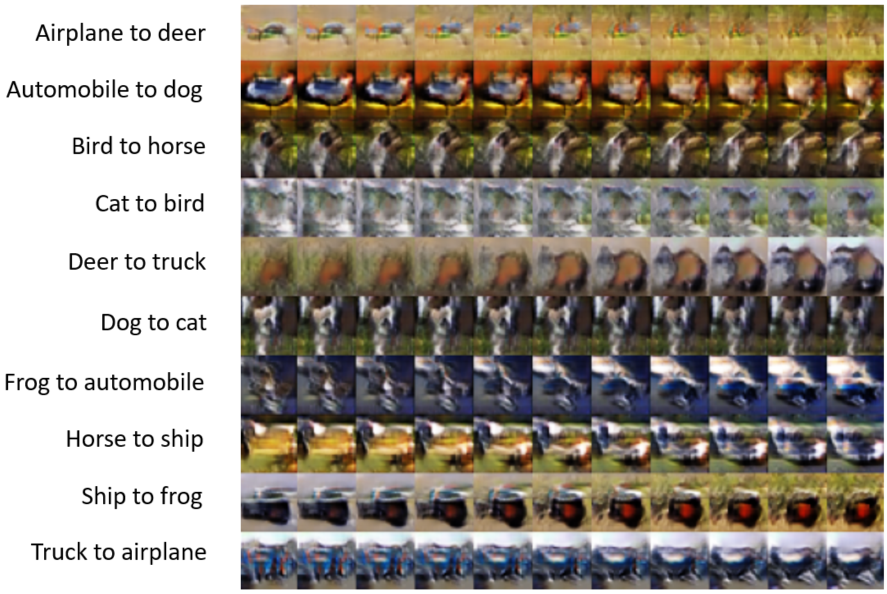


Figure 59: Exemplary outputs of the interpolation steps of the cGAN model trained on CIFAR-10.

ing with models that do not represent the respective training domain very well.

10.3.4 *Computational Evaluation*

In order to test the capability to interpolate between different classes, we used classifiers that we trained on the task of object classification. To this end, we used the EfficientNet-B0 architecture (M. Tan and Le, 2019), as these models turned out to achieve very high accuracy on both datasets (*Fashion-MNIST:* 0.9089, *CIFAR-10:* 0.9931). We used a softmax layer on top of the models, which produces an output vector $r \in \mathbb{R}^{+\ n}$ with $\sum_{i=1}^{n} r_i = 1$ where n is the number of classes. By interpreting this class probability vector r as confidence distribution over all the classes, we can assess the interpolation capabilities of the cGAN models by observing the change of r. To this end, 1,000 image sets were randomly generated for each class combination i, j in CIFAR-10 as well as Fashion-MNIST. Each of these images was conditioned on the respective source class i. Then, we performed interpolation steps for every source image as described in Section 10.2.2 with $\alpha = 0.1$, resulting in 10 interpolation steps until the target class was reached. For each interpolation steps, we fed all resulting images into the respective classifier model (i.e., either the Fashion-MNIST or the CIFAR-10 model). Results of the computational evaluation are plotted in Figure 60 and Figure 61.

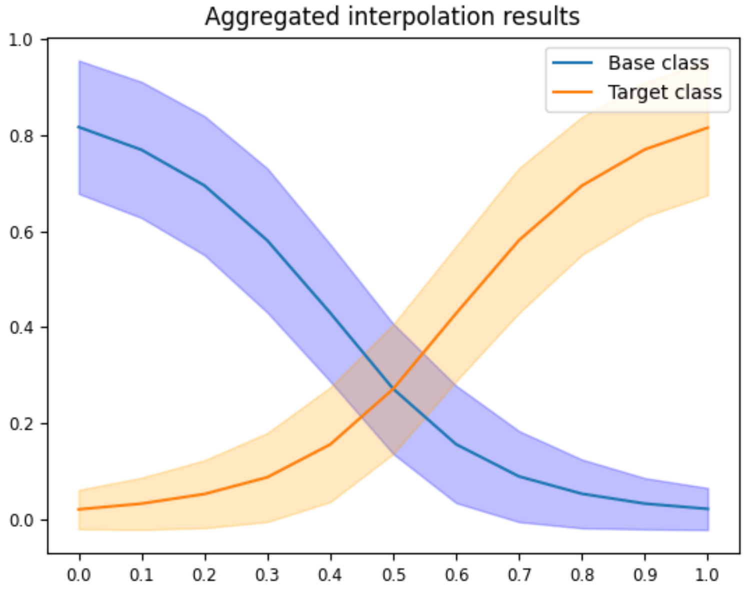


Figure 60: Results of the computational evaluation with Fashion-MNIST.

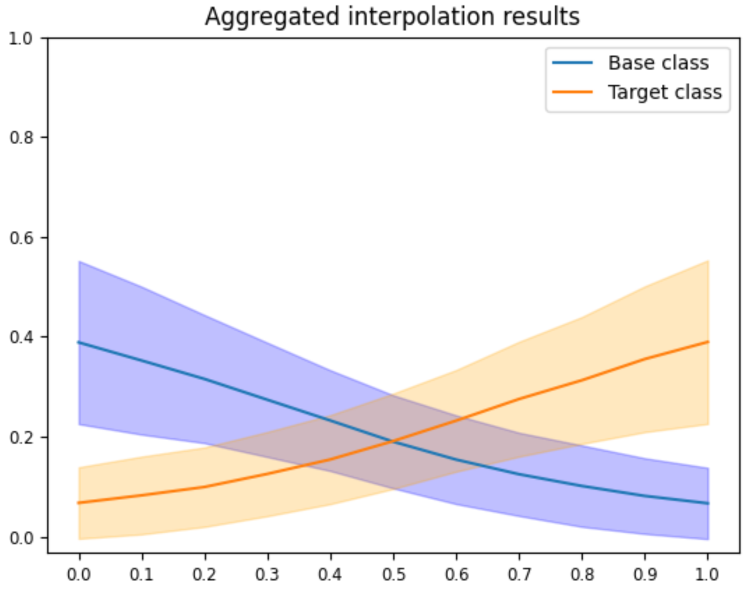


Figure 61: Results of the computational evaluation with CIFAR-10.

10.4 DIMENSIONAL FACE GENERATION

As our feasibility studies revealed that the mechanism of label interpolation shows promise when being used with more generic datasets, we apply it to the more sophisticated use-case of emotional face generation.

10.4.1 *Emotion Models*

Enabling algorithms to handle human emotion requires a discrete definition of affective states. Categorical and dimensional models are the two most prevalent approaches to conceptualize human emotions.
A categorical emotion model subsumes emotions under discrete categories like happiness, sadness, surprise or anger. There is a common understanding of these emotional labels, as terms describing the emotion classes are taken from common language. It is also for this reason, that categorical labels are the more common form of annotation found with datasets depicting emotional states. However, this (categorical) approach may be restricting, as many blended feelings and emotions cannot adequately be described by the chosen categories. Selection of some particular expressions can not be expected to cover a broad range of emotional states, especially not differing degrees of intensity.

An arguably more precise way of describing emotions is to attach the experienced stimuli to continuous scales within dimensional models. A. Mehrabian (1995) suggests to characterize emotions along three axes, which he defines as pleasure, arousal and dominance. Peter J. Lang, Margaret M. Bradley, and B. N. Cuthbert (1997) proposes the simplified axes of arousal and valence as measurements, resulting in the more commonly used dimensional emotion model. The valence scale describes the pleasantness of a given emotion. A positive valence value indicates an enjoyable emotion such as joy or pleasure. Negative values are associated with unpleasant emotions like sadness and fear. This designation is complemented by the arousal scale which measures the agitation level of an emotion (Figure 62). This representation is less intuitive but allows continuous blending between affective states.

Categorical as well as dimensional models are simplified, synthetic descriptions of human emotions and are not able to cover all of the included aspects. However, with our interpolation approach, we aim to cover the whole emotional range defined within the space of the dimensional valence-arousal model and enable a seamless transition between displayed emotions. As data collections featuring dimensional annotation for facial expressions are more sparse than the ones containing categorical labels (Section 10.4.2), being able to use emotional

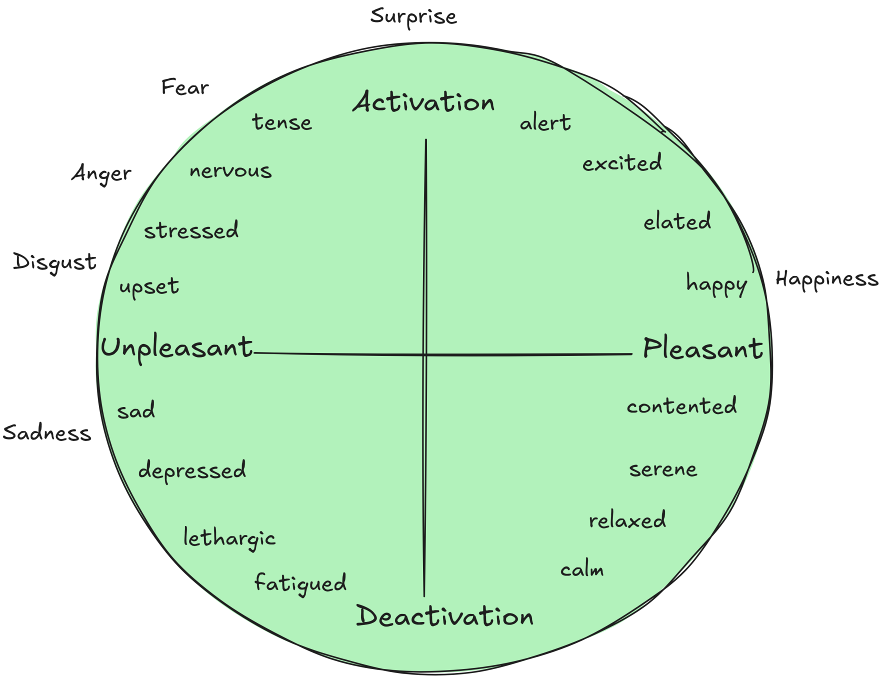


Figure 62: Russel's 2-dimensional valence arousal circumplex (J. A. Russell and Barrett, 1999).

labels in the training process is very beneficial. Goal of the following study is to use a cGAN that was conditioned on categorical emotions during training, and interpolate between those emotions in order to be able to create new images. Those newly generated face images show emotional states that are located in the continuous dimensional space of the valence/arousal model without having to correlate directly with discrete emotion categories.

To formally represent the valence and arousal of a face image I, we use a tuple $VA(I) = (v, a)$, where v refers to valence and a to arousal. Correlating with Russel's theory explained above, an image x with $VA(x) = (0, 0)$ is representing the center of the emotion space and thus show a neutral emotion. Emotions that are referred to in categorical emotion systems (e.g., *Happy*, *Sad*) are represented by valence/arousal states that show quite extreme values. When it comes to the interpolation of those dimensional emotional states, i.e., to create images with certain degrees of arousal or valence, we interpolate between the *neutral* emotion and the *extreme* emotional states. By the term *extreme emotion*, we refer to all categorical emotional states used except the *neutral* state, as this represents the center of the dimensional emotion model.

In our experiments, we stuck to performing interpolations between *Neutral* and a particular other emotion to preserve comparability between emotions. It should be noted that the approach could easily be extended to interpolate between two or even more categorical emo-

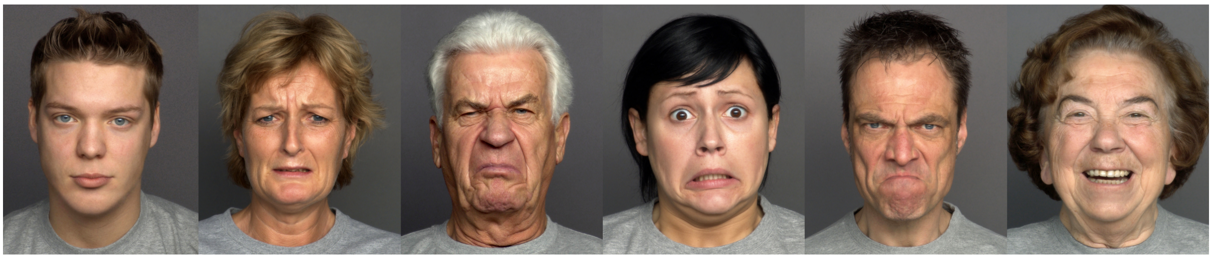


Figure 63: Exemplary data from FACES showing neutral, sad, disgust, fear, anger and happiness from left to right varying the age group (Ebner, Riediger, and Lindenberger, 2010).

tions. However, since we use only one categorical emotion and *Neutral* at a time, the following restriction must be added:

$$\exists i_{\in[2,6]} : v_1 + v_i = 1 \tag{29}$$

where v_1 represents the condition for *Neutral*.

To create an image that should show a specific degree of valence v or arousal a, where $0 \leqslant a, v \leqslant 1$, we use the one-hot element of the emotion that maximizes the specific value, for example *Happy* when it comes to valence, or *Angry* for arousal, and then decrease it to the desired degree. At the same time, we increase the one-hot element related to *Neutral* by the same amount, which allows us to create images showing valence/arousal values anywhere in Russel's emotion system, as opposed to the *extreme* values given during training.

10.4.2 *Dataset*

As previously mentioned, datasets labeled in terms of dimensional emotional models are scarce. Although there are a few datasets with continuous labeled information (e.g. *AffectNet* by Mollahosseini, Hasani, and Mahoor (2017) or AFEW-VA by Kossaifi et al. (2017)), they use to be gathered in the wild, resulting in miscellaneous data.

Data diversity usually is beneficial for deep learning tasks, however, in our specific use case of face generation with the focus on modeling certain emotional states in human faces, consistency in all non-relevant characteristics (i.e., characteristics not related to facial expressivity) is an advantage.

Thus, although a variety of categorically labeled datasets are available (Lucey et al., 2010; Matsumoto, 1988; Beaupré, Cheung, and Hess, 2000; Peter J Lang, Margaret M Bradley, Bruce N Cuthbert, et al., 1997; Van der Schalk, Hawk, and A. Fischer, 2009; Tottenham, 1998), we decided to use the FACES dataset (Ebner, Riediger, and Lindenberger, 2010) for our experiments, since it meets our requirements particularly well. In this dataset, all images are labeled in a discrete manner, and recorded with an identical uniformly colored background and an identical gray shirt. This is exemplified in Figure 63. To overcome the disadvantages of continuously labeled, but inconsistently recorded

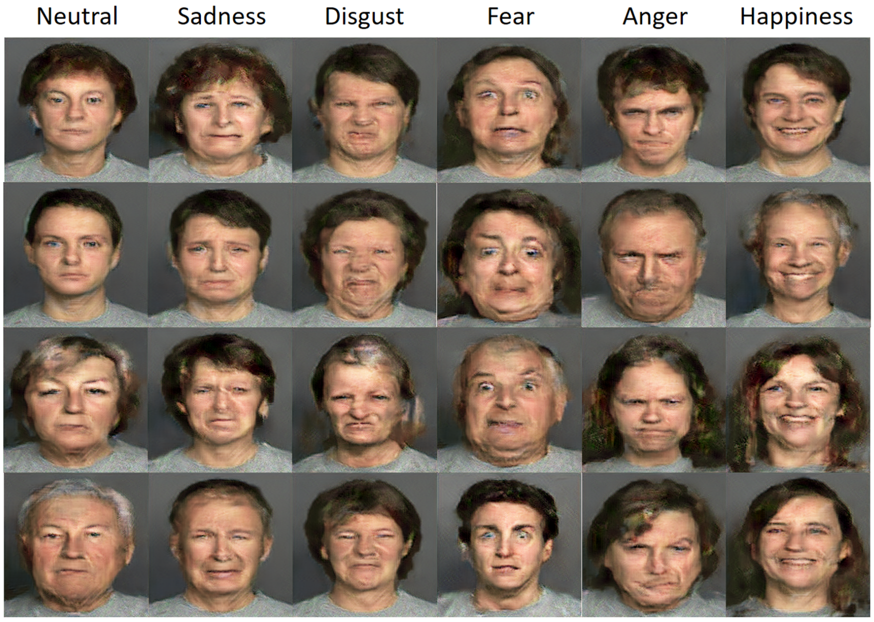


Figure 64: Example outputs of the trained cGAN model.

emotional face datasets, we explore the use of label interpolation with categorically labeled datasets.

Overall the FACES dataset consists of 2,052 emotional facial expression images, distributed over 171 men and women. 58 subjects are assigned to the group *Young*, 56 to *Middle-Aged* and 57 to *Old*, each showing 2 styles of the emotions *Neutral*, *Fear*, *Anger*, *Sadness*, *Disgust* and *Happiness*. For training, we resized the images to a target resolution of 256x256 pixels.

10.4.3 *Methodology*

As our feasibility study revealed, the interpolation approach has potential for creating transitions between different discrete states. However, it could be seen that the quality of the generated images, especially when dealing with the CIFAR-10 dataset, left room for improvement. To use the approach of label interpolation in a real world scenario like avatar generation or similar, such a poor image quality would be unacceptable. Thus, besides optimizing the cGAN model for our face generation use case even more, our evaluation process here is two-folded. First, we evaluate whether the cGAN is, before applying any interpolation, able to create images that are perceived correctly by human judgers. By doing so, we can assess if the cGAN model that we trained is capable of generating images with sufficient enough quality to express emotional states. Secondly, we conducted a computational evaluation analogously to the feasibility study.

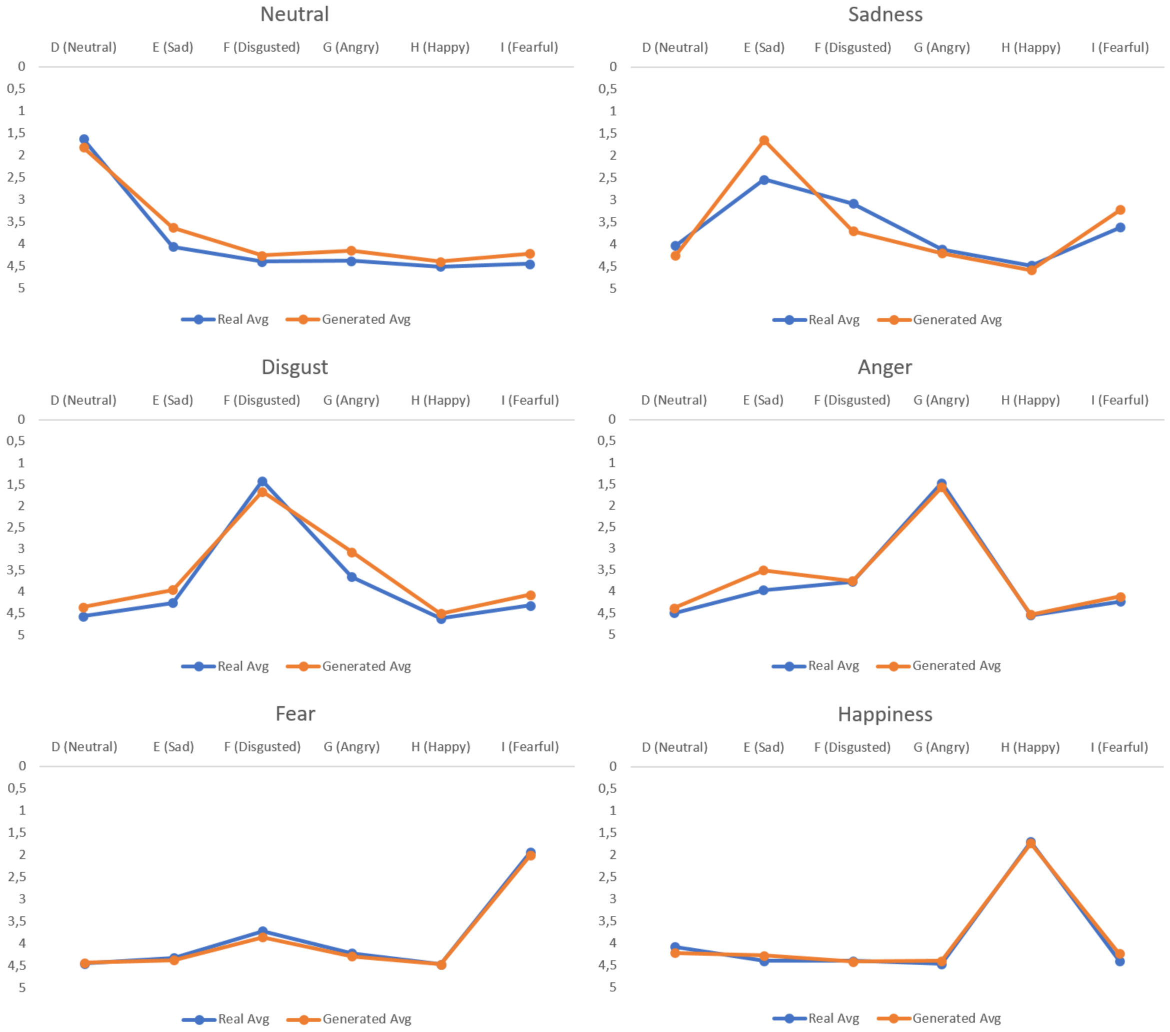


Figure 65: Results of the user study. Blue graphs show the perceived emotion of real images from the FACES dataset, while orange graphs show the perceived emotion of outputs of the cGAN conditioned on one-hot vectors. The y-axis represents the degree of the participant's agreement with the corresponding emotions that are represented by the x-axis.

10.4.4 *Training*

The model was trained for 10,000 epochs on all 2,052 images of the FACES dataset using *Adam* optimizer with a learning rate of 0.0001. Example outputs of the trained model, conditioned on one-hot vectors of all 6 used emotions, are shown in Figure 64.

10.4.5 *User Evaluation*

In our user study, we evaluated the cGAN's ability to produce images of discrete emotions generated with the respective one-hot vector encoding. In total, 20 probands of ages ranging from 22 to 31 years (M = 25.8, SD = 2,46, 40% male, 60% female) participated in the study.

During the survey, 36 images were shown to each of the participants. 18 of the images where original images taken from the FACES dataset, whereas the other 18 images were generated by the trained cGAN. All images were split evenly between all emotions, both for the original as well as for the generated images. To keep consistency with the images generated by the cGAN, the images taken from the FACES dataset were resized to 256x256 pixels. For each image, the participants were asked how much they agreed to the image showing a certain emotion. To mitigate confirmation bias, they were not told which emotion the image should show, but asked to provide their rating for each emotion. The ratings were collected by the use of a 5-point Likert scale (1 = strongly agree, 5 = strongly disagree). Results of the user study are shown in Figure 65.

As can be seen, the images that were generated by the cGAN were rated to show the respective target emotion in a similar convincing way as the original images taken from the FACES dataset. Each emotion is mostly recognized in the correct way by the study participants. One emotion, namely *Sadness*, even stands out as the artificially generated images were recognized even better than the original images, which were mistaken for *Disgust* more frequently. Considering these results, the trained cGAN model proves to be an appropriate basis for our interpolation experiments.

10.4.6 *Computational Evaluation*

Analogously to the computational evaluation in our feasibility studies, we verified if label interpolation can be used to enhance the cGAN network with the ability to generate images with continuous degrees of valence and arousal with the help of an auxiliary classifier. Again, 1,000 noise vectors per class were initially fed into the cGAN, where here, the classes were the five emotions *Sadness*, *Disgust*, *Fear*, *Anger* and *Happiness*. The conditioning vector was initially chosen to represent the neutral emotion. For each of the 5,000 noise vectors, 10 interpolation steps with step size $\epsilon = 0.1$ towards the respective extreme emotion were conducted. Thus, the last interpolation step results in a one-hot vector representing the respective extreme emotion. For evaluting the resulting valence/arousal values, we again used a pre-trained auxiliary classifier.

Figure 66 shows exemplary outputs of interpolation steps between *neutral* and the five used emotions.

The auxiliary classifier model was based on the MobileNetV2 architecture (Sandler et al., 2018). The model was trained on the AffectNet dataset for 100 epochs with *Adam* optimizer and a learning rate of 0.001, leading to a similar performance as the AffectNet baseline models, as can be seen in Table 10. We assessed the valence/arousal values for every interpolated output image of the cGAN and averaged

	AffectNet Baseline		Evaluation Model	
	Valence	Arousal	Valence	Arousal
RMSE	0.37	0.41	0.40	0.37
CORR	0.66	0.54	0.60	0.52
SAGR	0.74	0.65	0.73	0.75
CCC	0.60	0.34	0.57	0.44

Table 10: AffectNet performance comparison: Root Mean Square Error (RMSE), Correlation (CORR), Sign Agreement (SAGR) (Nicolaou, Gunes, and Pantic, 2011), and Concordance Correlation Coefficient (CCC).

them over the 1,000 samples per emotion, analogously to the feasibility studies described in Section 10.3.4. The results can be taken from Figure 67.

10.5 DISCUSSION

In our initial study we evaluated if our proposed approach can be used to seamlessly interpolate images between generic classes. To this end we relied on two widely used and publicly available datasets CIFAR-10 and Fashion-MNIST (see 10.3.1) to train our cGAN interpolation model. Figure 58 and Figure 59 are showing examples of the calculated interpolations between various classes on the Fashion-MNIST dataset and the CIFAR-10 dataset respectively. We can clearly see that the trained network was not able to capture the complexity of the input domain optimally. While the images from the Fashion-MNIST domain are showing slightly blurred contours, the generated images from the CIFAR-10 domain can only be partially recognized as the intended objects. However, when looking at the individual morphing steps between the classes, we can observe that the model is able to generate transitions that are generally smooth and continuous - two necessary prerequisites to apply the approach to interpolate between emotional expressions in human faces, in order to create meaningful results.

To further validate this observation we also employed a trained classifier for each dataset to predict the various interpolation steps between classes. Assuming a well calibrated classifier, we expected the distribution of the predicted class probabilities to continuously shift between the two interpolated classes along with the degree of interpolation.

Figure 60 and Figure 61 are showing the results for those classifiers as described in Section 10.3.4. In those plots, we averaged the class probabilities for both the base classes and the target classes for

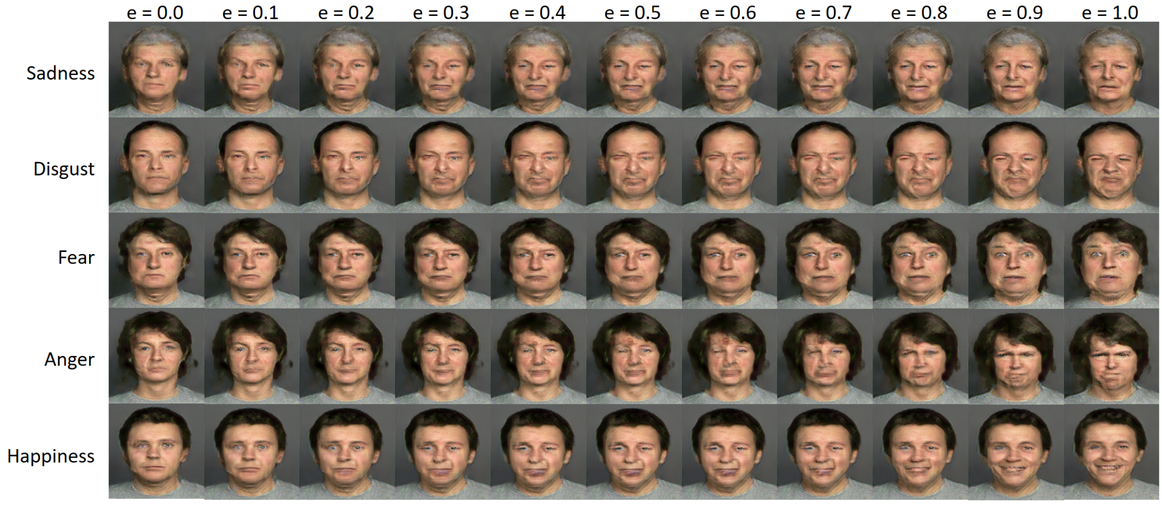


Figure 66: Example outputs of the interpolation mechanism. Each row shows a set of interpolation steps, where in each step, the emotion portion e was increased by 0.1, whereas the neutral portion was decreased by the same amount.

every interpolation step of all the assessed output images. It can be observed that, generally speaking, the interpolation mechanism led the network to generate transitions that are indeed perceived as *lying between the classes* by the classifier. Notably, the images produced by the cGAN that was trained on CIFAR-10 were generally classified with quite low confidence. This implies that the assumption that the cGAN model was not able to accurately resemble the dataset holds true. However, for both models, the confidence smoothly transitions between the two intended classes, indicating that label interpolation is a promising tool for further experiments.

We argue that those findings further substantiate the ability of our trained cGAN to generate continuous interpolations between images and therefore also the feasibility to further investigate if the approach is able to generate meaningful interpolations between different categorical emotions.

Upon visual inspection of the Fashion-MNIST and CIFAR-10 datasets, we found that the quality of the artificially generated images from the random noise vectors was substantially worse compared to the original samples from the respective areas.

We therefore firstly conducted a user study to assess the capabilities of our employed cGAN model to produce realistic images of people expressing clearly identifiable emotions (see Section 10.4.5. The results of this study, as depicted in Figure 65, show that participants generally recognize the expressed emotions in the artificially generated images similarly well as in the original images from the FACES dataset. The only exception being the emotion *Sadness*, which was even better identifiable from the artificially generated images than the original data, where participants confused the emotion more of-

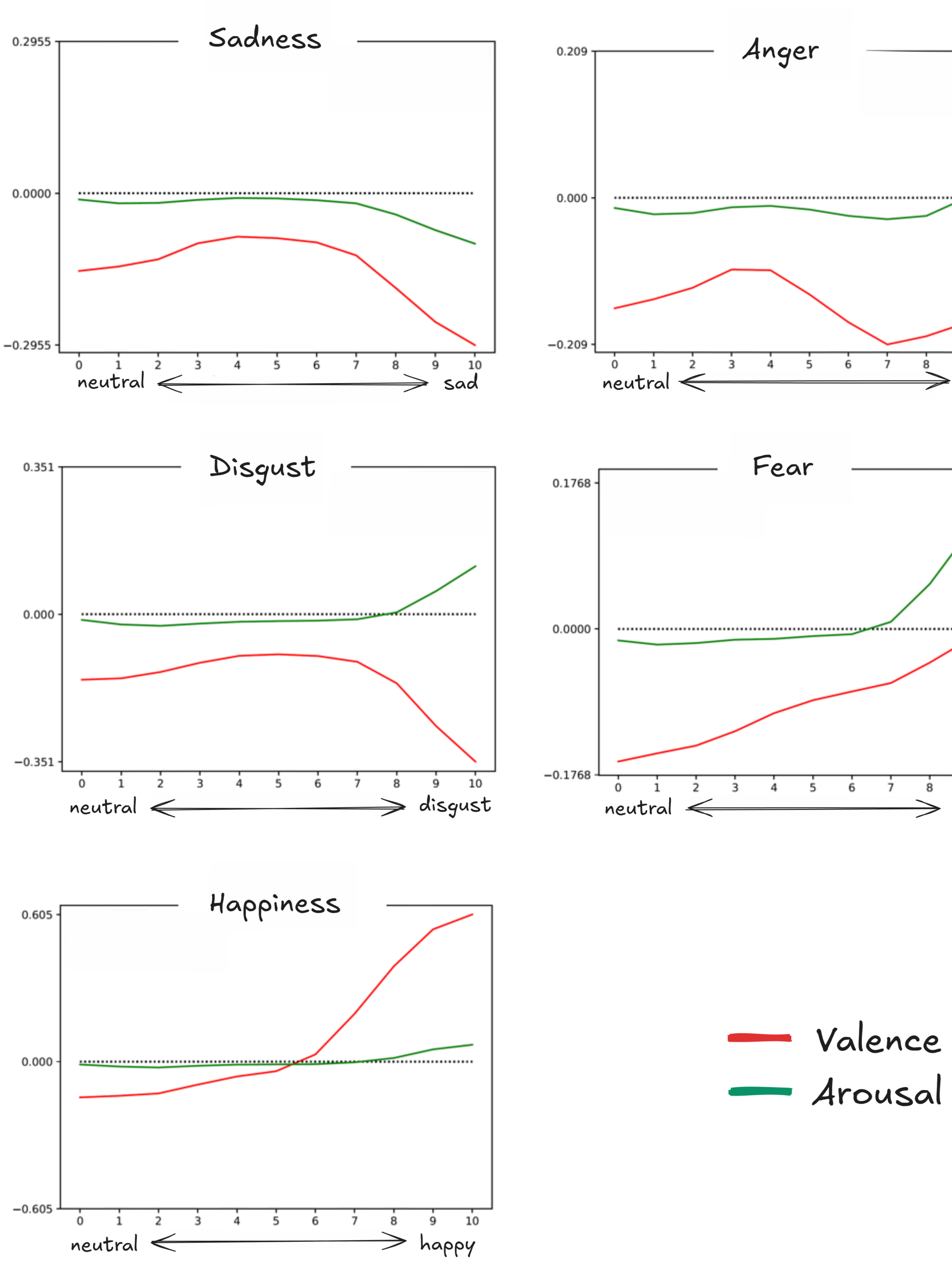


Figure 67: Computational Evaluation of our interpolation approach. Red graphs show valence, while green graphs show arousal. The x-axis represents the interpolation steps. Each interpolation step was performed by increasing the corresponding emotion vector element by 0.1, while decreasing the neutral vector element by 0.1.

ten with *Disgust*. Those results are leading us to the conclusion that our employed cGAN model is suitable to further explore interpolation between emotional classes.

The results of the computational evaluation are depicted in Figure 67 for each emotional class respectively. We can see that the interpolation mechanism is able to condition the cGAN to produce face images with various valence/arousal values. Upon further inspection we can observe that those values are mostly located in the value range between the start and end point of the interpolation, which indicates the general trend of the system to transition smoothly between emotional states. However, the plots also show that the interpolation function is not in all cases strictly monotonic. For example, for *Sadness* and *Disgust*, the valence value initially rises slightly before dropping towards the interpolation endpoint. Similarly for *Anger*, both valence and arousal values are first moving up and down before arriving at their initial starting point. This is a strong deviation to the position of anger in the circumplex model of emotions, where we would expect both valence and arousal to be notably higher when compared with the neutral position. Furthermore, we can see that the detected valence value is - in all cases - slightly below zero for the neutral emotion. Since all emotions have been correctly recognized by human raters, we attribute this behaviour to shortcomings in the valence/arousal regression model. Taking those human quality ratings and the predominantly correct trend lines of the interpolation into account, we argue that our approach can indeed be used to generate face images of continuous emotional states. The fact that the values are not evolving in a linear way, i.e., the plots appear rather as curves than as straight lines, does not take away much from the results, since the single interpolation step intervals can easily be modified to achieve a more even interpolation. E.g., instead of using the same step interval for every single interpolation step, higher intervals can be used in ranges where the target features are changing slower.

10.6 CONCLUSION

In this chapter, we examined the possibilities of continuous interpolation through a discrete label space of Conditional Generative Adversarial Networks. Therefore, we first conducted some feasibility studies to assess the general applicability of interpolating between discrete classes to a trained cGAN. We found that indeed the technique can be used to generate smooth transitions between classes, even in cases where the cGAN did not learn to model the training domain to a satisfactory level. Subsequently, we applied the label interpolation mechanism to the exemplary scenario of continuous emotional face generation. After ensuring that a cGAN trained on a dataset of categorical emotional face images learned to model that categorical emotional states by conducting a user study, we assessed the applicability of label interpolation in order to generate face images that show continuous emotional states. By using an auxiliary classifier for evaluating the cGAN outputs, we found that the algorithm was able to cover most of the valence/arousal ranges that are needed to cover the full dimensional emotion space. Although the performance of the approach shows to be highly dependent on the emotions that are used for interpolation, it demonstrates the potential of our novel approach of Label Interpolation.

Part V

FEEDBACK SYNTHESIS

A conceivable way of strengthening users' self-worth and self-esteem is to practice a specific skill with a user, while constantly making him or her aware of points to improve. By doing so, the user might get a more realistic impression of their own skill - which in turn could result in higher self-esteem. However, it is important that the feedback given to a user is actionable and feasible. The feedback should point the user in a direction that he or she can actually achieve. Therefore, counterfactual explanations are a promising medium of conveying such feedback, as they aim to show only the least possible change that has to be made in order to achieve a better outcome. However, synthesizing such counterfactual feedback is not a trivial task, as the resulting explanations should still be consistent. Here, GANs can help. In the following part, we will show how we can use GANs to build user feedback systems in order to reveal potential opportunities of improvement. Therefore, we will exemplarily address the scenario of a job interview training system.

11

VERBAL RECOMMENDATIONS FOR JOB INTERVIEW TRAINING SYSTEMS

Large parts of this chapter have already been published in the following publication:

> *Heimerl, A., Mertes, S., Schneeberger, T., Baur, T., Liu, A., Becker, L., Rohleder, N., Gebhard, P., & André, E. (2022). Generating personalized behavioral feedback for a virtual job interview training system through adversarial learning. In International Conference on Artificial Intelligence in Education (AIED).* (Alexander Heimerl et al., 2022b)

In stressful situations such as job interviews, many people tend to show nervous and uncontrolled behaviours. This circumstance most often affects their performance in a negative way. Especially in job interviews, the goal is to convince a recruiter of ones fit in a company by actively engaging in the conversation. Recruiters hereby consciously or unconsciously evaluate the candidate's social cues. The amount of positive engagement a candidate shows towards the interviewer may play a central role in deciding whether the candidate is suitable. Paulhus et al. (2013) found that active integration behaviors such as engagement, laughing, and humor led to better performance ratings and, therefore, to a higher chance of getting the job. In recent years, technology-based job interview training systems have been developed to improve the performance of candidates (e.g., (Baur, Damian, et al., 2013; Hoque et al., 2013; Takeuchi and Koda, 2021)). In terms of feedback generation, previous systems mostly rely on expected features that either have or have not been observed (Naim et al., 2018; Gebhard et al., 2019; Takeuchi and Koda, 2021).

This chapter extends previous approaches by not only measuring the occurrence of certain social signals, but also by answering the question *"what would have been a better behaviour?"*. In order to achieve this, our approach provides counterfactual explanations in the form of textual advises for improving the behaviour to be more fitting in the current situation. The counterfactual explanations are generated through a GAN and a subsequent template based system. A benefit of these created counterfactual explanations is that the generated recommendations change as much of the originally observed behaviour as needed, but as little as possible. Thus, recommendations are not drawn from highly exaggerated or oversimplified samples and therefore guarantee realistic and meaningful explanations. Compared to only giving descriptive feedback, providing counterfactual recommendations reduces possible errors in feedback interpretation. As behavioural assessment criteria, we use the user's engagement in the interview. As such, we try to give feedback that enables the trainee to appear more engaged in the conversion. Therefore, we present a feedback extension to an existing job interview training environment that uses a socially interactive agent as a recruiter and an engagement recognition component to enable the virtual agent to react and adapt to the user's behavior, and emotions (Baur, Mehlmann, et al., 2015). There, during a preparation phase, trainees were instructed to show certain behaviors in specific job interview training situations and got feedback from the virtual agent on whether they could perform these instructions correctly. This training aims to help improve social skills that are pertinent to job interviews. Our new feedback extension employs an XAI method based on counterfactual explanations for generating verbal feedback about observed social behavior. This approach allows communicating features (e.g., no eye contact, closed body posture) that weaken the overall job interview performance.

The introduced feedback extension is based on a deep learning classifier predicting the user engagement in job interview situations. As input, the classifier uses multimodal feature representations (e.g., gaze, body posture, or gestures) of the trainee. We exploit the concept of counterfactual explanations to show how the user would need to adapt his or her behaviour to appear more engaged. Therefore, a GAN-driven counterfactual explanation model is trained that transforms the shown feature representations to corresponding counterfactual explanations, i.e., the feature representations are changed in a way that the user would have appeared engaged. The explanation generation compares the counterfactual feature vectors with the original feature vectors to derive textual recommendations automatically. Finally, they are presented to the trainee by a socially interactive agent in the role of a job interview coach. Figure 68 shows a schematic overview of our approach.

11.1 RELATED WORK

Due to the complexity and importance of job interviews, automatic training approaches have been developed to improve the performance of the candidates. Multiple simulated training systems have been proposed over the years that combine social signal interpretation and virtual agents (Baur, Damian, et al., 2013; Hoque et al.,

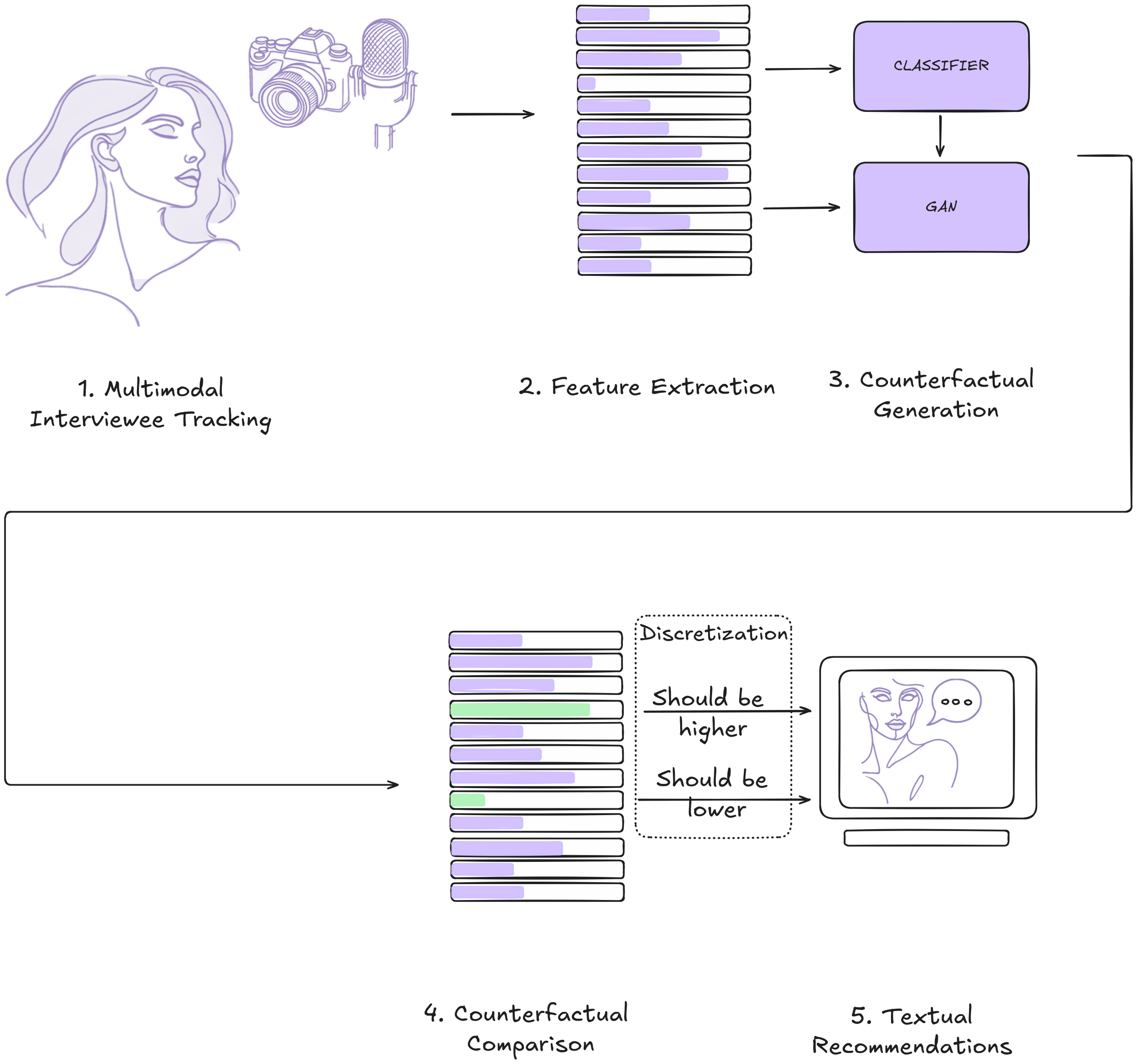

CLASSIFIER
GAN
1. Multimodal Interviewee Tracking
2. Feature Extraction
3. Counterfactual Generation
Discretization
Should be higher
Should be lower
4. Counterfactual Comparison
5. Textual Recommendations

Figure 68: Job interview training system with GAN-generated recommendations.

2013; Takeuchi and Koda, 2021). Another example by Gebhard et al. (2019) introduced a serious game simulation platform to train social skills. They showed that their training systems can be utilized to teach individuals how to display adequate socio-emotive reactions during job interviews. Naim et al. (2018) introduced a framework for the automatic assessment and analysis of job interview performance. Their proposed system is capable of reliably predicting ratings for interview features such as friendliness, excitement and engagement. Through analysis of the learned feature weights of their regression model they were able to derive general recommendations on how to behave during job interviews, e.g., use filler words less frequently. However, those recommendations are not specific to a situation but rather general guidelines. Takeuchi and Koda (2021) developed a job interview training system that provides automatically generated, situation-specific feedback by analyzing nonverbal behaviour and comparing it to a reference model of ideal nonverbal behaviour. The feedback generation was accomplished by defining weights for the shown improper nonverbal behavior in accordance with its importance during the interview.

Even though providing feedback or guidelines based on weight prioritization may produce satisfactory results, those approaches fail to take the interplay of different shown nonverbal behaviors into account, since each behavior is considered on its own. Imagine a job candidate that is appearing to be low engaged due to a closed body posture with crossed arms and additionally isn't giving his interlocutor much nonverbal feedback like nodding. For such a case it is not enough to consider each behavior or corresponding feature on its own. If we choose to recommend giving more nonverbal feedback, we need to be also aware of how the person is being perceived while changing one of his behaviors. In our case, this would result in a person nodding while still maintaining a closed body posture with crossed arms. Therefore, we argue that it is important to consider the interplay of features when generating personalized feedback and nonverbal behavior recommendations. By utilizing a counterfactual reasoning process we are able to generate feedback that models a holistic recommendation for nonverbal behavior adjustments. This reasoning process tries to answer the question of how should the person have behaved to be perceived as more engaged. For this purpose, the underlying GAN tries to change simultaneously as many features as needed while at the same time trying to change as few features as possible and therefore guaranteeing meaningful recommendations.

Furthermore, by basing our approach on counterfactual explanations, our system is able to give highly personalized feedback. Conati et al. (2021) pointed out the potential value of personalized XAI for intelligent tutoring systems - which is highly related to our problem domain.

11.2 RECOMMENDATION GENERATION

The next sections offer an overview of the different components we implemented to generate behavioral recommendations that point out how the user should have behaved to appear more engaged.

11.2.1 *Feature Extraction*

In order to train a model for engagement recognition and recommendation generation, we modeled a high-level engagement feature set that can be easily interpreted. The feature set consists of 18 metrics mapping facial behavior, body language and conversation dynamics.

During conversations, the face usually occupies most of the interlocutors' attention. A lot of important information regarding the level of engagement can be extracted from the face, respectively the head. In fact, there are multiple studies that found a correlation between head movement / gaze behaviour and conversational engagement (Ishii and Nakano, 2010; Bednarik, Eivazi, and Hradis, 2012; Ooko, Ishii, and Nakano, 2011). Inspired by those findings we defined several features that represent the overall movement of the head and gaze behavior. Moreover, we considered the valence of the face calculated from the facial action units that have been extracted with OpenFace (Baltrusaitis et al., 2018).

Another modality we take into account is the general body language of the job candidate. The alignment of the body and the limbs play an important role in broadcasting the state of engagement (Müller et al., 2013). Interlocutors that are engaged during a conversation align their bodies to each other in order to "create a frame of engagement" (Kidwell, 2013). We tried to cover the general behaviour of the body, as well as specific gestures or poses that are connected to engagement. We defined a group of features that are mainly inspired by the coding system introduced in (Dael, Mortillaro, and K. R. Scherer, 2012). It contains several metrics to map the orientation and movement of the joints. Those metrics represent - amongst others - the overall level of body openness. Besides that, we also calculate a cumulative value over all joints to measure the overall body movement. Lots of body movement may indicate restlessness, which can be an indication for low engagement (D'Mello, Chipman, and Graesser, 2007). In addition to that, we also considered the amount of gesticulation an individual performs, as that plays an important role in nonverbal communication (Albert Mehrabian, 2007; Dael, Mortillaro, and K. R. Scherer, 2012).

Finally, we also covered some form of conversation dynamics. Turntaking and vocal cues play an important role throughout a conversation (Knapp and J. Hall A., 1997). During a conversation, the interlocutors usually alternate their speaking turns. Therefore, we determine

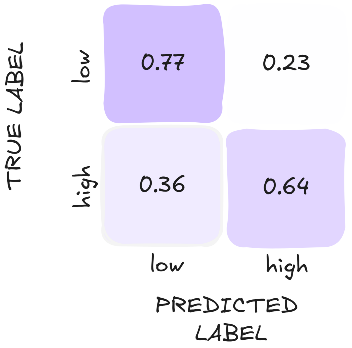


Figure 69: Confusion matrix of the neural network for the recognition of low and high conversational engagement (*Test* set).

the interlocutor that is currently holding the turn by considering the general voice activity of the interlocutors. This allows us to draw conclusions about the overall involvement of the individuals during the conversation. An overall low voice activity may imply a conversation with low engaged interlocutors.

11.2.2 *Engagement Model*

Based on the feature set introduced in Section 11.2.1, we trained a simple feedforward neural network with two dense layers for the recognition of low and high engagement. For training the network, we used the NoXi database (Cafaro et al., 2017). It provides dyadic novice-expert conversations. We decided on the NoXi corpus since it contains multi-modal multi-person interaction data and its transferability to social coaching scenarios. Moreover, the setup of the corpus allowed for both engaging as well as non-engaging interactions.

A total of 19 sessions of the NoXi corpus have been annotated regarding conversational engagement resulting in 10.5 hours of training data. The data has been randomly split into training and test sets, so that no sample of the same participant is present in the training and the test set. The training set included 13 sessions and contained 6.8 hours of data. The rest was allocated to the test set. Figure 69 displays the confusion matrix of the classifier for the test set.

11.2.3 *Counterfactual Features*

In a next step, to be able to give recommendations on how the user should have behaved to appear more engaged, we applied a counterfactual explanation generation algorithm, i.e., we aim to modify the input feature vectors that were classified as *low engaged* in a way that the classifier would change it's decision to *high engaged*. As described above, the recommendations that we aim for can be seen as counterfactual explanations for the engagement model presented in

Section 11.2.2. To generate these counterfactual feature vectors, we used an adversarial learning approach. In Chapter 7, we presented our *GANterfactual* architecture, which extended the CycleGAN framework (Zhu et al., 2017), which is an adversarial approach to domain translation, with further modifications that support the architecture in transforming original samples to counterfactual samples that are classified in a different way by a specific decision system to be explained. To this end, we incorporated the classifier into the training process of their CycleGAN-driven counterfactual explanation system via an additional loss function component. For this chapter's system, we built a network architecture adapted from the GANterfactual framework, which was originally implemented for generating counterfactual explanations in the image domain. The use of the GANterfactual framework has multiple benefits for the recommendation quality: Firstly, the cycle-consistency loss that is an integral part of CycleGANs forces that the learned transformation is minimal, i.e., only relevant features are changed. In the context of recommendation generation, this implies that the generated behavioral recommendations are highly personalized. Secondly, the adversarial loss component that is part of every GAN architecture leads to highly realistic results. Thus, recommendations are not drawn from highly exaggerated or oversimplified feature vectors. Thirdly, the counterfactual loss introduced in Chapter 7 enforces that the counterfactual explanations (in our case, the behavioral recommendations), are valid. As the engagement model that we used for our system works with feature vectors with no spatial relations between the single features, we replace the convolutional blocks of the original architecture with fully connected blocks. Further, the input layer was adapted to fit the feature representations that we also use for the engagement classifier. The rest of the architecture, as well as the training procedure, was taken from the original GANterfactual framework. For the GAN training, we relied on the NOXI dataset, which we also used for training the engagement classifier. Thus, the adversarial framework learns to convert feature vectors that show low engagement to feature vectors that show high engagement.

11.2.4 *Textual Recommendations*

After generating the counterfactual feature vectors, we compared them to the original feature vectors that represent the shown nonverbal behavior. Depending on the demanded detail of feedback, we return the features that had undergone the greatest value transformation. After identifying the most meaningful counterfactual features, we convert those into textual feedback. For this purpose, we discretize the features based on a defined textual template. For example, the feature representing the overall activity of the head gets translated into "try to keep your attention on your interlocutor" or "try to use more nonverbal feedback" depending on the present feature value. The amount of discrete classes varies for different features and can easily be adjusted depending on the given use case. The generated feedback is provided verbally to the user by the virtual coach inside the job interview training environment. An example of a recommendation provided by the virtual coach is displayed in Figure 70.

11.3 pilot study

The present pilot study's goal was to get preliminary insights about the assessment of a possible job interview training applying GAN driven recommendations. We used a mixed-methods design, combining questionnaires and a semi-structured interview. The study was conducted in January 2022.

11.3.1 *Method*

11.3.1.1 *Participants.*

We gathered data from 12 volunteering student participants (7 female, 5 male). Participants' age was between 21 and 29 years (*M* = 23.83, *SD* = 2.66). On average, participants attended 4.33 job interviews (*SD* = 2.74; *Min* = 1; *Max* = 10) prior to the study. Two of them had already experience with job interview trainings, three with virtual agents.

11.3.1.2 *Procedure and Material.*

In this pilot study, the experimenter and participant met in a video call. After agreeing to the consent form, the experimenter explained the background of the study and presented videos of our job interview training system. For the videos, we used a multi-modal job interview role-play dataset (Schneeberger et al., 2019) to create behavioral feedback. In that dataset, participants were confronted with a job interview conducted either by an interactive social agent or a human interviewer. Participants were recorded with the MS Kinect2. We used 5 sessions with the human interviewer as input to our job interview training system. The resulting recommendations were then rendered into a video (Fig. 70) that was shown to the participants. The participants saw the part of the job interview training in which the trainee gets the individual feedback from the virtual coach after having a mock job interview. The coach first presents the recorded part of the job interview and gives the recommendation afterwards verbally. Participants were asked to imagine that they were the trainees using the training to practice a job interview. Next, participants filled in the questionnaires. Then, the semi-structured interview was held. In the end, the experimenter thanked the participants for their participation. The whole procedure took around 25 minutes.

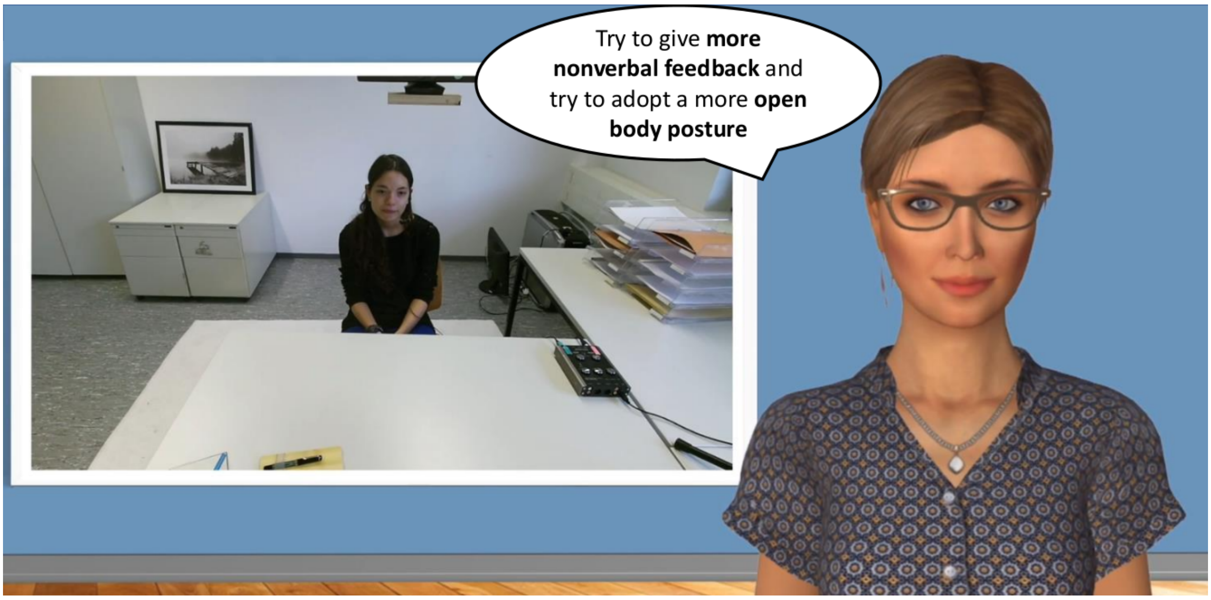

Try to give **more nonverbal feedback** and try to adopt a more **open body posture**

Figure 70: Coach giving the recommendation after the mock job interview.

11.3.1.3 *Measurements.*

Demographics included age, sex, job interview experience, and job interview training experience. *Usefulness* was measured with the usefulness scale of the MeCUE (Minge and Riedel, 2013). It contains three items. Cronbach's Alpha was .92. *Transfer motivation* was measured using four items adapted from (Rowold, Hochholdinger, and Schaper, 2008) covering whether training lessons learned will be useful in upcoming situations: "I believe that my performance in job interviews will improve if I apply the knowledge and skills I have acquired with training.", "It is unrealistic to believe that mastering the training content can improve my performance in job interviews. ", "I can apply skills and knowledge acquired from job interview training to my daily life.", "I feel like after the training I could apply the behavior very well. ". Cronbach's Alpha was .90. *Feedback Quality* was measured with four self constructed items: "I felt the feedback was accurate.", "I would have given similar feedback.", "I feel like the feedback is helpful.", "I don't think the computer can give me accurate feedback.". Cronbach's Alpha was .87. All questionnaire items were answered on a 7 point scale ranging from 1 (*strongly disagree*) to 7 (*strongly agree*).

The *Semi-structured interview* covered six areas: 1) general impression, 2) persona, 3) other possible use-cases, 4) suggestions for improvement, 5) intention for further use, and 6) added value.

11.3.2 *Results*

11.3.2.1 *Questionnaires*

In the three questionnaires, the following descriptive data was found: Usefulness ($M = 4.72$, $SD = 1.17$); Transfer motivation ($M = 4.92$, $SD = .94$); Feedback ($M = 4.60$, $SD = 1.26$).

11.3.2.2 *Semi-structured interviews*

The answers gathered in the semi-structured interview were analyzed and categorized for each of the six areas separately:
1) Regarding the *General impression*, participants mentioned six times that the recommendations were useful / feasible (e.g., "Simple tips that were easy to implement, but have a big impact.") or comprehensible (2x). Three participants mentioned that the recommendations were too unspecific. Once each was mentioned that the recommendations are not useful ("Would prefer feedback on the content of my answer. Job interview is too stressful for me such that I could focus on non-verbal behavior.") and too obvious ("If I saw myself in the video, I would have known that I have to improve the recommended behaviors.").
2) Participants described the *persona* as someone with a wish to improve (7 namings) that is open for new thing (3 namings), career oriented (2 namings), young (2 namings), self reflective (2 namings) or non-self reflective (1 naming).
3) As *other possible use-cases* participants named training to improve communication skills in general (8 namings) and for more specific groups, like patients with anxiety disorders or people with social phobias. The named also other possible situations like preparing for challenging employee appraisals, conflict resolution dialogs, or other high stakes situations. Another named use-case was public speaking (4 namings).
4) Participants mentioned seven times that they would like to have more specific recommendations, e.g. "The agent could say something like: Nonverbal feedback is nodding, for example." Moreover, they thought that recommendations based on the content of the answers would be helpful (2 namings). Also, some participants noted that the agent could be improved (3 namings), like using a more empathic voice. One participant noted that an interactive training mode, where you practice recommendations directly and get instant feedback would be helpful.
5) *Intention for further use* was indicated by 9 participants. Three could not imagine using the training.
6) The *added value* of the training was for most of the participants that the recommendations are given directly on a specific behavior shown in a specific situation during the job interview. Moreover, one

participant mentioned that the training was especially helpful as it gives a low-threshold possibility to practice job interviews that could be offered by agencies supporting people to find employment. One other participant said that having an agent instead of a human giving recommendations decreases the feeling of being judged for mistakes.

11.3.2.3 *Recommendation generation*

As described in Section 11.2.3, we incorporated a classifier for the recognition of engagement into the training process of the GAN via an additional loss function component. In order to verify the validity of our approach, we examined whether the counterfactuals generated by the GAN are modifying the features that the engagement classifier identified as important for the classification of low and high engagement. For this evaluation, we used five sessions of the multi-modal job interview role-play dataset (Schneeberger et al., 2019) that have also been used in Section 11.3 and extracted the importance scores of every feature in regard to the model's classification with LIME (Ribeiro, Singh, and Guestrin, 2016). Next, we calculated the absolute value change of how much each feature has been modified by the counterfactual transformation. Afterwards, we calculated the Pearson Correlation Coefficient between the importance scores of every feature and the absolute change of each feature, see Figure 71. High correlation scores indicate that the counterfactual feature transformation is in line with the corresponding importance of the feature. The more important a feature is for the classification of a sample the greater also should be the change of the feature in order to result in a different classification result. Seven features showed a strong positive correlation (GZ_DR, AM_CR, HD_TH, DIST_RW, YROT_LE, SDX_HD, SDXROT_HD), six features had a moderate positive correlation (HD_AC, YROT_RE, XROT_RE, TN_HD, CONT_MOV, EN_HA) and two features presented with a low positive correlation (DIST_LW, XROT_LE). Moreover, FO_RW had a strong negative correlation, VAL_F showed a moderate negative correlation and FO_LW had a weak negative correlation.

Moreover, we conducted a computational evaluation to investigate how well the generated counterfactual features change the decision of the engagement classifier. For this evaluation, we also used the multimodal job interview role-play dataset. We found that 96.49% of the generated counterfactual feature vectors led to a different decision of the engagement model as the original input features.

11.4 DISCUSSION

The results of our user study indicate that training with our system could be helpful to prepare for job interviews successfully. The recommendations given by the system were found to be helpful and

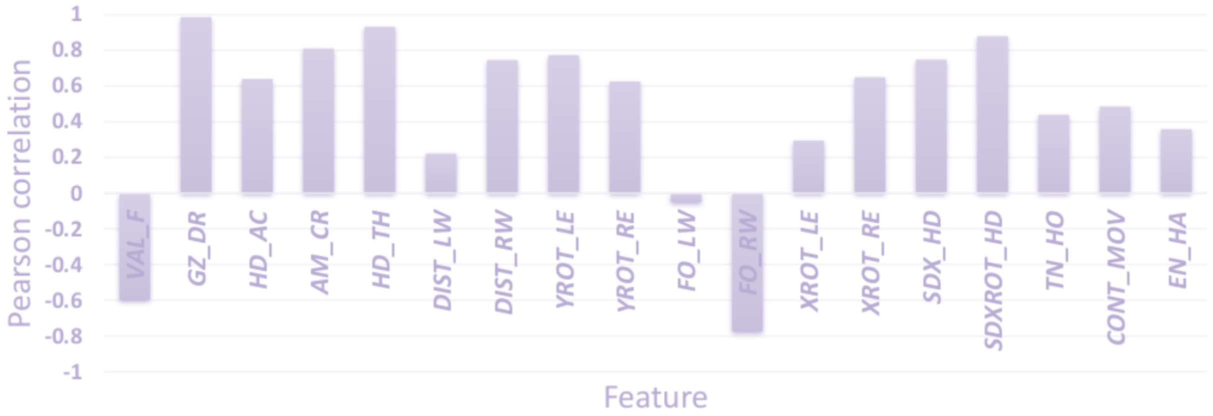


Figure 71: Pearson correlation between the absolute change of the feature values and the LIME classification relevance scores for every feature. The features are from left to right: *Valence Face, Gaze behavior, Head activity, Arms crossed, Head touch, X distance of left/right wrist and hip, Y rotation left/right elbow, Y distance of left/right wrist and hip, X rotation left/right elbow, Standard deviation head movement in X axis, Standard deviation Head X rotation, Turn hold, Continuous movement, Gesticulation*.

comprehensible, and transferable to other use cases. Moreover, most participants noted that the proposed approach adds additional value to the training by giving recommendations directly on a specific behavior in a specific situation. Part of the underlying training system automatically extracts situations that could be improved and displays them alongside the recommendation presented by the virtual coach. However, the pilot study also revealed that the recommendations should be more specific. Therefore, in future work, the template used for discretizing the counterfactuals should be extended to be more diverse and specific or use natural language processing to generate textual recommendations from counterfactuals directly. The latter would need additional annotation and training effort. Moreover, we examined the validity of our GAN-driven recommendation generation approach by calculating the Pearson correlation coefficient between the absolute changes of the feature values after counterfactual transformation and the importance of the features the classifier attributed to them regarding the classification result. We showed that most of the features (15 out of 18 features) had a moderate to strong correlation, which emphasizes the validity of the proposed approach. Only the two features corresponding to the relative position and movement of the left wrist and the feature representing the flexion of the left elbow presented a weak correlation. Further, it is interesting to point out that the feature representing the relative movement of the right wrist (FO_RW) has shown a strong negative correlation. This means that the counterfactual suggests decreasing the relative distance from the wrist to the rest of the body when the current feature value is an indication for low engagement. The opposite is the case when the current feature value indicates being highly engaged, here the relative distance should be increased. This indicates that for the given job interview data, the engagement classifier attributes a lower wrist distance towards the body as appearing higher engaged. A similar case presented itself for the valence of the face. For this feature, we found a moderate negative correlation. For the valence of the face, the classifier interprets lower valence values, meaning a more serious facial expression, as a sign for higher engagement. This interpretation is most likely related to the dataset used for training the classifier and the corresponding conversational engagement annotations. Therefore, extending the used training data for both the classifier and the GAN for future work makes sense. Especially the classifier might benefit from more training data as the accuracy scores leave room for improvement. Also, the current classifier only distinguishes between low and high engagement. It would also be interesting to investigate the resulting counterfactuals when using a more fine-grained representation for conversational engagement. Further, we also investigated how well the generated counterfactual features can change the decision of the engagement classifier. Overall, 96.49% of the counterfactual feature vectors led to a different decision of the engagement classifier as the original input features. This indicates that our GAN-driven approach enables to generate recommendations that, when being adopted, are consistently leading to a perception of high engagement. The computational evaluation, as well as the user study, indicate that the generated recommendations are valid and helpful in the context of job interview coaching scenarios.

11.5 CONCLUSION

In this chapter, we introduced a novel approach for generating textual nonverbal behavior recommendations in job interview training environments. We extended an interactive virtual job interview training system with a GAN-based approach that first detects behavioral weaknesses and subsequently generates personalized feedback. To evaluate the usefulness of the generated feedback, we conducted a mixed-methods pilot study using mock-ups from the job interview training system. The overall study results indicated that the GAN-based generated behavioral feedback is helpful. Moreover, participants assessed that the feedback would improve their job interview performance. All in all, the presented approach is a step towards personalized feedback systems to support self-esteem, and as such towards user-centered artificial intelligence.

12

VISUAL RECOMMENDATIONS FOR JOB INTERVIEW TRAINING SYSTEMS

In the last chapter, we aimed to generate verbal feedback, i.e., feedback that was represented as text. There, GANs were used to *find* decent counterfactual instances, while the *presentation* of the feedback was done procedurally. In this chapter, we give a proof-of-concept on how we could use GANs also for the representation of feedback. Therefore, this time, we use a procedural approach for finding the counterfactuals, while we use GANs for visualizing them.

Our approach uses tracked skeleton data of the user as intermediate representation – the actual counterfactual search is done on skeleton-level, i.e., for a specific point in time, we perform a procedural search for a skeleton data instance that is very close to a to the original one, but is perceived as more engaged. In order to obtain such data instances, we use manually labeled data of a publicly available dataset. The counterfactual skeleton is then used to synthesize an image of a real person using our GAN model. By using the skeleton representation as intermediate layer, we are able to synthesize data of arbitrary people, as long as we have enough data to train the GAN. One logical decision here would be to directly use tracked data of the trainee as training data – as such, the user could watch himself behaving in the correct way. However, here, training images of the user would have to be available. Although these could be recorded directly during the job interview training, it would still need some time to train the GAN before being able to show the feedback. A second option is to use a GAN that is already trained to synthesize images of another person (i.e., a job interview trainer) – and as such being able to give immediate feedback. In this chapter, we provide a proof-of-concept for the latter version, which, however, can technically be used analogously for the former one.

12.1 APPROACH

Our system is a pipeline of multiple processing steps. In a first stage, we track a video stream of the job candidate. From that video data,

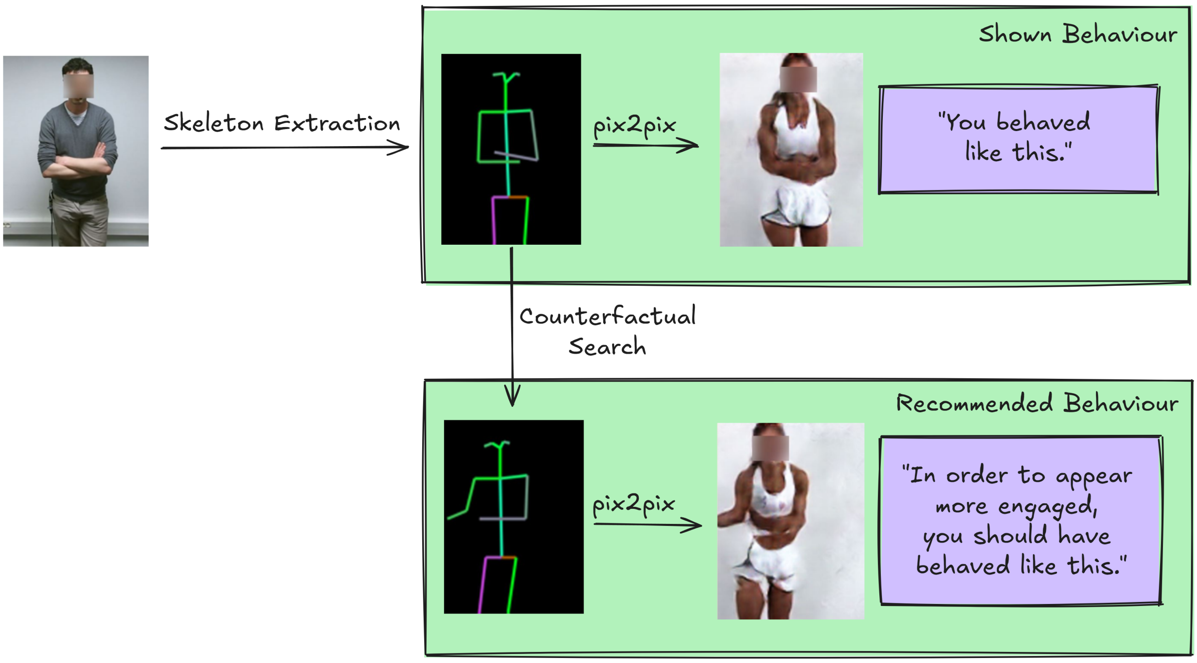


Figure 72: Exemplary output of our system. First, a job candidate is captured and its skeleton data is extracted. Then, a counterfactual skeleton is searched. Both the original and the counterfactual skeleton are converted to realistic images of a job interview trainer by using the pix2pix architecture. *Faces blurred for privacy protection.*

we extract pose skeleton data by the use of the OpenPose framework (Cao et al., 2019). In a next step, we convert that skeleton in a way that the pose would appear more engaged. To do so, we compare the extracted skeleton data with data samples of the Noxi Corpus (Cafaro et al., 2017). Through that comparison, we aim to find a skeleton sample that is as closely as possible to the candidate's skeleton, but appears to be highly engaged. Such a skeleton can be seen as a counterfactual example to the original one. After having found that counterfactual skeleton, we feed both the original and the counterfactual skeleton data into a style transfer GAN architecture that converts the skeletons data back into real looking images of a human.

By doing so, we have the possibility to present the candidate with a realistic looking visualisation of *how* his posture should change in order to appear more engaging.

12.1.1 *Counterfactual Skeleton Data*

The use of skeleton data as intermediate representation enables two crucial benefits for the application. First, using an abstract representation of the job candidate allows for the comparison of that representation with a broad variety of video recordings. In our case, we used the Noxi Corpus as a basis for finding a skeleton that is as closely to the original as possible, but shows high engagement. Second, as skeleton data holds a lot of information about the shown engagement, we keep the option open to use different counterfactual generation algorithms that do not rely on a comparison to existing data, but perform an automated conversion step from original to counterfactual data. E.g., in Chapter iii, we proposed a GAN-based generation system for counterfactual explanations. Although such end-to-end based systems could, in theory, be used in our application as well, we decided to use a more straight-forward approach which provides satisfying results in our domain. We calculated the Mean Squared Error (MSE) between the input skeleton and the skeleton coordinates of all samples of the Noxi Corpus that are labelled as *highly engaged*, i.e., that had an engagement annotation greater than 0.9. The skeleton with the smallest MSE was chosen as the counterfactual skeleton. The use of MSE as comparison metric enforces that (i) the difference between original and counterfactual skeleton is as small as possible, and (ii) if the original skeleton is already showing high engagement, the counterfactual skeleton will, due to the relatively large comparison dataset, most likely not change much. Such, it is ensured, that a job candidate that already appeared highly engaged does not get the advice to change his or her behaviour.

12.1.2 *From Skeleton Data to Visual Recommendations*

In order to allow for a visualized counterfactual recommendation, we used a pix2pix architecture (Isola et al., 2017) to convert the skeletons to photorealistic human data. For demonstration purposes, we trained the network on a video dataset cropped from YouTube. We used OpenPose to estimate skeleton data for each frame of the video, and then trained the network to convert skeleton data back to image data. Our pipeline allows our job interview coaching system to be retrained to produce images of job interview coaches that correspond to the personal preference of the job candidate. Further, it would also be possible to train the pix2pix network on images of the job candidate itself. By doing so, the coached person could see him- or herself performing the job interview while appearing more engaged. However, for a proof-of-concept, we have stayed with the YouTube data, as here we could use videos of a huge variety of different postures, allowing the pix2pix architecture to model a broad spectrum of skeleton data.

12.2 DISCUSSION

12.2.1 *Explanation Quality*

Following the proposed approach we were able to generate realistic images of poses from skeleton data. The conversion from skeleton data to images produced reasonable poses. Therefore, it is safe to assume that the underlying model has been able to identify the relevant information needed for a meaningful conversion.

However, our experiments also revealed room for improvement. While most of the skeleton data has been successfully mapped, modeling the hands and face has been inaccurate, i.e., the reconstructed hands and faces are quite blurry or missing at all. A variety of reasons could have potentially caused these results:

- One limitation is the small dataset used for training. For training the model, we used roughly one hour of YouTube video data. In order to have training data that contained a variety of different movement patterns, we selected a fitness video. As such, the resulting dataset consisted of samples mapping a broad range of joint positions and alignments. However, using that very compact data resulted in a quite low number of training samples.

- While fitness videos provide a great variety of poses, they come with the drawback of displaying repetitive motions (e.g., push-ups and lunges). As such, the diversity of the training data suffered.

- Moreover, we only used video data of one person. Therefore, the person-dependent characteristics of the skeleton may differ from the trainee, which can result in incorrect conversions.

These drawbacks can be overcome by extending the training dataset with more individuals and a variety of scenarios, as well as increasing the overall amount of sessions.

12.2.2 *Explanation Selection*

Our approach operates at the frame level, meaning skeletal data is extracted from individual images and transformed into counterfactuals accordingly. Given the video inputs of the job interview training, which consist of a large number of individual frames, the question arises as to which frames should indeed be transformed and displayed in order to provide the user with the most useful feedback. In principle, it is conceivable to transform the entire video stream frame by frame into counterfactuals, and by doing so, regaining a new video stream. However, this would require tweaks in the architecture, as the temporal component is not currently represented, which would lead to unsmooth videos with glitches and artifacts. Moreover, it is questionable whether an explanation is really needed for every point in time, or whether it makes more sense to select individual, particularly significant scenes/frames to showcase the feedback for a very conspicuous misbehavior. For instance, one could train a classifier to automatically detect the trainee's engagement and generate explanations for precisely those situations in which the trainee performed least effectively - as was done in the last previous chapter.

12.3 conclusion

Figure 72 shows an exemplary output of our system. As can be seen, the behavioural feedback produced by our system is able to produce outputs that are reasonable and can potentially help people with preparing for a real job interview. However, the image quality of the used pix2pix architecture has to be improved further. Also, although technically analogously to the work presented here, it has still to be validated if the approach can also generate feedback by using the recorded data of the trainee itself. All in all, we presented a proof-of-concept of our novel approach for automatically generating visual counterfactual recommendations for job interview training systems. Our outputs show that the approach is promising and shows great potential to be investigated further.

Part VI

INTERACTION WITH GANS

The real-time capabilities of GANs make them an ideal choice to enable the development of *interactive* systems. Such interactive systems could be used for a variety of use-cases. In this chapter, we give a specific example on how we can leverage the strengths of GANs to build interactive experiences - by using a GAN-based system to counteract stress. A popular method for stress relief is the consumption of content inducing a Autonomous Sensory Meridian Response (ASMR). ASMR is a sensory phenomenon involving pleasurable tingling sensations in response to stimuli such as whispering, tapping, and hair brushing. It is increasingly used to promote health and well-being, help with sleep, and reduce stress and anxiety. While ASMR triggers are both highly individual and of great variety. Consequently, finding or identifying suitable ASMR content, e.g., by searching online platforms, can take time and effort. Also, while ASMR is highly related to the concept of flow, which in turn always involves active involvement in a task, ASMR is - so far - always considered a *passive* experience. Here, we show how GANs can be used to elevate the consumption of ASMR to being an interactive experience. Therefore, we present a visual interface that allows to interact with a GAN model that was trained to synthesize ASMR sounds. By doing so, the sounds can be individualized in an interactive way. As such, we give an example of how GANs can be used to foster well-being by reducing stress in a novel and effective way.

13

USING GANS FOR INTERACTIVE ASMR TRIGGERS

Large parts of this chapter have already been published in the following publication:

> *Mertes, S., Strobl, M., Schlagowski, R., & André, E. (2023). ASMRcade: Interactive Audio Triggers for an Autonomous Sensory Meridian Response. In Proceedings of the 25th International Conference on Multimodal Interaction.* (Mertes, Strobl, et al., 2023)

In today's fast-paced world, stress is increasingly becoming a pervasive and challenging problem in people's daily lives (Steele, J. A. Hall, and Christofferson, 2020; Weinstein and Selman, 2016). Persistent stress can lead to a variety of physical and mental health issues, such as anxiety, depression, sleep disorders, heart disease, high blood pressure, headaches, and many other health problems (Kessler, 1997; Stults-Kolehmainen and Sinha, 2014; Halkos and Bousinakis, 2010; McEwen, 2008; Partinen, 1994; Gasperin et al., 2009). Therefore, it is very important for people to find an adequate balance to relax and thus counteract stress.

An effective method of relaxation that is becoming more and more popular is the consumption of content triggering an *Autonomous Sensory Meridian Response* (ASMR) (Barratt and Davis, 2015). ASMR is a term used to explain the tingling sensation some people experience in response to certain, mostly auditive, stimuli. Typically, this feeling is described as a delightful buzz that begins in the scalp and travels down the neck and back (Barratt, Spence, and Davis, 2017). Auditive triggers for ASMR can vary, including sounds such as whispering, soft speaking, tapping, or scratching. Additionally, visual or physical stimuli, like observing someone do a repetitive task or receiving a gentle touch, can also elicit ASMR. In recent years, ASMR has become increasingly popular, with numerous content creators producing videos and audio recordings intended to induce ASMR in their audience. While there is limited scientific research on the topic, many individuals find ASMR to be a relaxing and pleasurable experience (Barratt and Davis, 2015). However, it's important to note that not everyone experiences ASMR or is affected by sounds meant to induce ASMR in the same way.

ASMR is often considered a *passive* experience because it involves a state of relaxation and receptivity to certain stimuli rather than an active engagement with them. However, Barratt and Davis (2015) found a correlation between the number of ASMR responses and the *flow* state of participants, which is the psychological state of being immersed in a task (Csikszentmihalyi, 2000). Hence, as such tasks typically involve active participation or interaction with the environment, e.g., in sports or playing video games, we propose an *active* approach for ASMR triggers that interactively engages ASMR recipients.

To this end, this chapter introduces *ASMRcade*, an interactive application for ASMR sound synthesis, exploration, and customization. The application utilizes a GAN that we trained to generate high-quality audio samples that are aimed to trigger ASMR. By providing a graphical user interface, users of ASMRcade are able to interact with that model's latent space and, as such, directly influence the sounds that the GAN produces in real time.

The main contributions of this chapter are as follows:

- We trained a WaveGAN (Donahue, McAuley, and Puckette, 2018) model on a dataset of ASMR sounds. Specifically, for a proof-of-concept, we use *tapping* sounds, which have proven to induce ASMR for many people in the past (Barratt, Spence, and Davis, 2017). Such sounds are typically created by tapping on various objects, such as surfaces, or directly on a microphone.

- Using a web-based graphical user interface as introduced by Schlagowski, Mertes, and André (2021), we made the input latent space of the trained WaveGAN model interactive. Specifically, we represent parts of the latent model space of the WaveGAN through certain interface elements. By moving those elements, the user directly influences the sounds that the model produces.

- We present a first exploratory user study in which we let users interact with our system. We show that our system can induce ASMR and, as such, is a promising direction for future research in the field of ASMR.

To the best of our knowledge, this chapter is the first work introducing an interactive approach to the field of ASMR research.

13.1 RELATED WORK

13.1.1 *Autonomous Sensory Meridian Response*

Autonomous Sensory Meridian Response (ASMR) is a sensory phenomenon characterized by a pleasurable tingling sensation with its origin in the scalp, moving down the neck, and sometimes following the line of the spine down to other areas of the body in response to specific audio and visual stimuli. In recent years, there has been a growing interest in understanding the triggers that lead to ASMR experiences and the potential benefits of experiencing ASMR. As such, the research community has begun to embrace the potential of ASMR as a valuable source of inspiration for Human-Computer Interaction (Klefeker, Striegl, and Devendorf, 2020).

Barratt and Davis (2015) proposed that ASMR is a flow-like mental state, which refers to a state of intense focus and absorption in an activity. They suggested that ASMR experiences involve a deep immersion in the sensory stimuli, which can lead to a sense of timelessness and an altered state of consciousness. The researchers also found that people who experience ASMR report a range of positive effects, such as relaxation, stress relief, and improved mood.

In their later study, Barratt, Spence, and Davis (2017) investigated the sensory determinants of ASMR by surveying a large online community of individuals who experience ASMR. They found that the most common triggers were related to auditory stimuli, such as whispering, tapping, and scratching sounds. Other common triggers included visual stimuli, such as hand movements and personal attention, and touch-related stimuli, such as gentle touch and hair brushing. Interestingly, the researchers also found that individual factors, such as personality traits and mood, could influence the likelihood of experiencing ASMR.

Poerio et al. (2018) further explored the triggers of ASMR by conducting laboratory experiments in which participants listened to a range of audio and visual stimuli while their physiological responses were measured. They found that ASMR experiences were associated with a reduction in heart rate and an increase in skin conductance responses, indicating a relaxation response. The study also found that the most effective ASMR triggers were those that involved interpersonal closeness, such as whispering and soft-spoken voices.

In terms of content, people consume a variety of ASMR videos and audio recordings to experience the tingling sensation. Common types of content include role-playing scenarios, where the so-called *ASMRtist* portrays a particular role (such as a hairdresser or a doctor), soundscapes, where various sounds are played, and guided meditations or affirmations. There are many successful ASMRtists with millions of subscribers on YouTube, indicating the existence of a vast online community interested in the phenomenon. Overall, ASMR is a complex phenomenon that involves a combination of sensory and personal factors, and triggers can vary widely between individuals. However, research has shown that certain types of stimuli, such as gentle sounds and interpersonal closeness, are more likely to elicit an ASMR response. Furthermore, the flow-like mental state experienced during ASMR has been proposed to have potential benefits for mental health and well-being.

13.1.2 *Customizable Sound Generation*

The technical foundation to make an interactive ASMR experience possible is a method to synthesize artificial sounds in a controlled way. While early works in that field made use of procedural approaches like *Concatenative Sound Synthesis* (Schwarz, 2005) or *Modular Sound Synthesis* (Sueur, Aubin, and Simonis, 2008), mechanisms using machine learning are on the rise since several years. For example, approaches from the field of Evolutionary Computing are widely used for optimizing sound synthesis problems (Mitchell, 2012; Lai et al., 2006; Miranda and Al Biles, 2007) and were, although not in an interactive setting, also applied to ASMR sound synthesis (Nan and Fukumoto, 2022). One promising approach for sound generation is the use of Generative Adversarial Networks (GANs), which were originally developed for image synthesis (Goodfellow et al., 2014). Various modifications to the original GAN architecture were presented that address the generation of audio data, such as *GANSynth* (Engel et al., 2019) or WaveGAN (Donahue, McAuley, and Puckette, 2018).

In order to add the possibility for controlling the audio outputs generated by a GAN, mechanisms were proposed that incorporate directive features into the GAN training (C. Y. Lee et al., 2018; Dong et al., 2018), or directly search through the input space of an already trained GAN, e.g., using methods such as Latent Variable Evolution (LVE). In the context of ASMR, Fang et al. (2023) adapted the DCGAN architecture (Radford, Metz, and Chintala, 2015) to create random new ASMR sounds. A similar approach was followed by Oh et al. (2023), who created artificial ASMR sounds using the SpecVQGAN architecture (Iashin and Rahtu, 2021). However, although evaluations of both of these approaches indicated that GANs could create auditively pleasing ASMR sounds, they did not include mechanisms to engage with the sound synthesis process interactively.

13.1.3 *Interactive Approaches to Parameter Space Exploration*

The generation of ASMR audio content is only one part of our objective - another goal is to actively engage users in the ASMR sound generation process. For the ASMRcade application, we use the WaveGAN architecture for sound synthesis, which can transform an arbitrary, non-interpretable input vector to sound that resembles the data the model was trained on. Therefore, we have to enable the user to interact with the input parameter space of the WaveGAN. In the general field of audio synthesis, multiple approaches exist to interactively explore parameter spaces, such as mapping parameters to

2D-Interfaces using Hilbert curves (Tubb and Dixon, 2014), or using interactive evolutionary algorithms (Dahlstedt, 2001; Ritschel et al., 2019). In the context of music synthesis, interaction with parameter spaces was even used as musical instruments (Berndt, Al-Kassab, and Dachselt, 2015; Snyder and Ryan, 2014; Morris, Simon, and Basu, 2008; Kaliakatsos-Papakostas, Gkiokas, and Katsouros, 2018). Further, several works have focused on building interactive systems specifically for exploring a GAN's input parameter space. Here, a popular use case is to interactively control the output of GANs that were trained to generate drum sounds. For instance, Ramires et al. (2022) used *SeFa*, a closed-form factorization method (Y. Shen and Zhou, 2021), to find dimensionality-reduced directions of a GAN's parameter space, which they made accessible to the user through a Graphical User Interface (GUI). Further, Schlagowski, Mertes, and André (2021) and Schlagowski, Wildgrube, et al. (2022) aimed for interactive drum sound synthesis for a drum sequencer application. Similar to our work, they used the WaveGAN architecture and built a user interface to directly modify single latent vector elements of a GAN. As they observed positive user experience ratings in their user study, we decided to include certain GUI-elements of their system within our system.

13.2 THE ASMRCADE SYSTEM

Our system consists of two major parts. Firstly, a WaveGAN model that was trained to produce highly realistic and diverse ASMR tapping sounds, and secondly, a web-based graphical user interface, which includes visual representations of the WaveGAN's latent input vector elements. By interacting with those visual representations, i.e., changing their spatial position, the user can directly influence the WaveGAN's audio output in real-time. All in all, by using our system, the user can actively engage in the ASMR sound generation process instead of just passively listening to ASMR stimuli. The following sections give a more detailed overview of the ASMRcade system's single components.

13.2.1 *Audio Generation*

A modification of the original GAN framework was introduced by Donahue, McAuley, and Puckette (2018), who presented the WaveGAN architecture. While originally, GAN models were used for image synthesis, WaveGAN was specifically designed for audio synthesis. We selected the WaveGAN architecture for our system as it is fast and fulfills the required quality standards of the generated output for successful ASMR triggers.

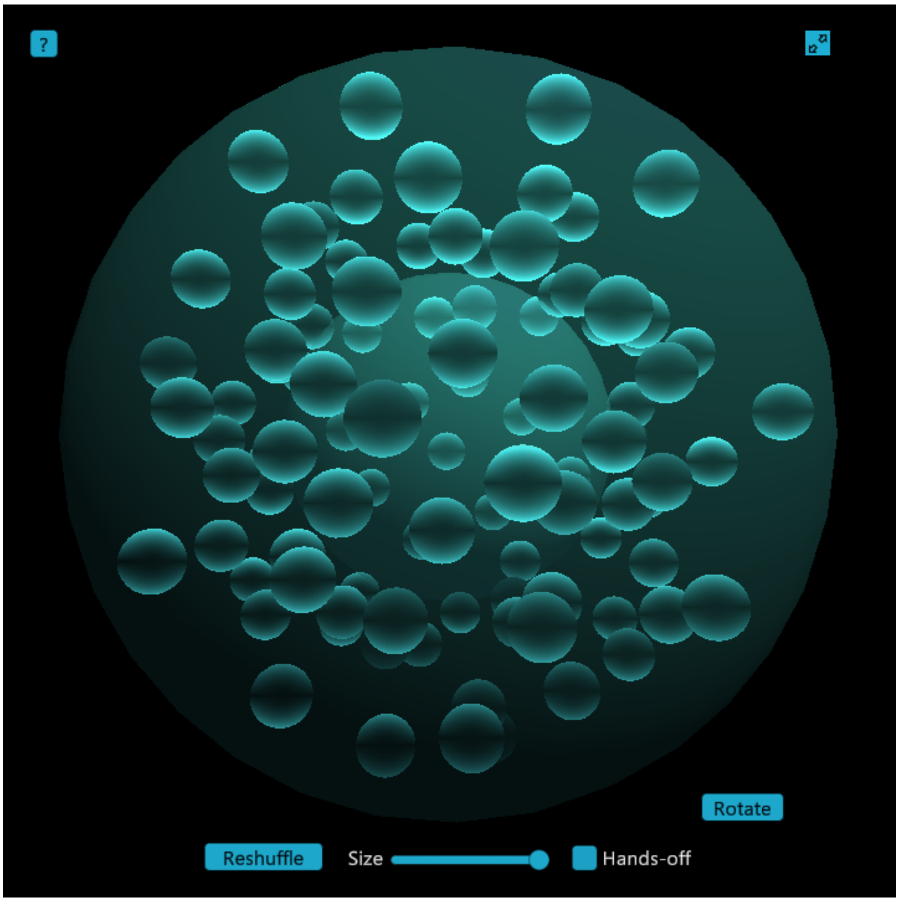


Figure 73: The ASMRcade user interface adopted from Schlagowski, Mertes, and André (2021).

13.2.1.1 *Dataset*

A necessary prerequisite for training a WaveGAN model is the existence of a suitable dataset. Datasets that can be used to train a GAN for high-quality content generation must be restricted to short, nonverbal, and quality-consistent data (K. Yang, B. Russell, and Salamon, 2020). As existing ASMR datasets did not fulfill these requirements, we collected data by processing suitable ASMR YouTube videos. For a proof-of-concept, we focused on *tapping* sounds, which were found to be a well-functioning ASMR trigger for many people (Barratt, Spence, and Davis, 2017). We used tapping-only videos of content creators well-known to the ASMR community, namely "ASMR Bakery"[1], "LottieLoves ASMR"[2], "Coromo Sara"[3] and "ricarda"[4]. We downsampled the audio to 16kHz mono using FFMPEG. One limitation of GANs is that they require a fixed output length definition. Therefore, we extracted audio segments of 1 second length using a threshold-based segmentation. Further, we applied *Fade-In* and *Fade-Out* effects to the segments. By doing so, we got rid of unwanted cracking or popping sounds at the start or end of the audio sample that the generator might otherwise reproduce. We divided the dataset into training, testing, and validation sets to ensure robust evaluation and validation of the GAN model.

13.2.1.2 *GAN Training*

We used the WaveGAN architecture presented by Donahue, McAuley, and Puckette (2018). WaveGAN is a modification of the DCGAN (Radford, Metz, and Chintala, 2015) architecture, which in turn extends the original GAN framework by replacing the fully connected layers by convolutional and deconvolutional layers. WaveGAN modifies the DCGAN to work well specifically with audio data, e.g., by replacing 2D kernels with 1D kernels and other adapatations. For a full architecture description, please refer to the WaveGAN publication (Donahue, McAuley, and Puckette, 2018). The model was trained using the training configuration recommended by Donahue, McAuley, and Puckette (2018). After 20k training steps, the generated audio samples resembled tapping sounds with various surfaces like glass and wood, and no further improvement could be observed. Minor artifacts were still present, but the overall results were satisfactory.

[1] https://www.youtube.com/watch?v=sIgkTYTWPz8

[2] https://www.youtube.com/watch?v=kMvGsOrpjNo

[3] https://www.youtube.com/watch?v=y03M_isyV3E

[4] https://www.youtube.com/watch?v=EGgKJsuM7Ns

13.2.2 *User Interface*

Schlagowski, Mertes, and André (2021) compared different graphical user interfaces, called *Vector Manipulation Modules* (VMMs), that make audio generated by WaveGAN customizable. Opting for a VMM design with high hedonistic UEQ ratings in *Attractiveness*, *Stimulation*, and *Novelty*, we adopted one of their designs representing elements of the latent input space in 3D spheres (see Fig. 73). The user can drag and drop these spheres to adjust their corresponding numerical value in the latent input space of the GAN and hear the resulting change of the generated audio sample in real-time. As the WaveGAN model that we trained transforms a 100-dimensional noise input vector to audio data resembling ASMR tapping sounds, there are 100 manipulatable spheres in the GUI. The numerical latent space value is proportional to the spatial distance of the dragged (smaller) sphere to an inner and outer sphere, representing the minimum and maximum thresholds of the corresponding latent space value (semitransparent spheres in Fig. 73). The initial 3D positions are calculated using the fibonacci lattice algorithm (Stanley, 1975).

Other application features include controls to add a *Reverb* effect, control the volume, and completely reshuffle the current input vector. Further, an additional *Size* slider is part of the system. By moving that slider, the number of controls for the input vector can be decreased so that the user can focus on only a few dimensions instead of all 100.

We embedded both the VMM and a Javascript-based version of WaveGAN into a webpage that uses the browser to display the VMM, run the GAN, and playback generated audio. Additional features of the resulting demonstrator system include:

1. Playback of the last generated ASMR sound in a loop.
2. An optional "Hands off" mode, in which the input vector automatically changes each time the sound is played within the loop.
3. A tutorial mode, which explains the system by pointing to certain AI elements and describing their features.

13.3 EVALUATION

To evaluate if our application is suitable to induce an ASMR experience, we conducted an explorative user study in which participants had to interact with the ASMRcade system. Therefore, we made ASMRcade available on a web server and invited online participants to test it.

13.3.1 *Study Group*

We wanted to ensure that a good portion of our participants were already familiar with and actively engaging with the concept of ASMR. Thus, we targeted the existing ASMR community by creating a post in the subreddit "r/ASMR", an online platform where people share ASMR-related experiences and information on a regular basis. There, we briefly explained the application and invited users to test the ASMRcade application and complete the associated questionnaires. As a result, we acquired 17 participants, which primarily consisted of users from the "r/ASMR" subreddit and others who discovered the Reddit post.

13.3.2 *Study Procedure*

During the study, the participants could first interact with the ASMRcade system for as long as they liked. After that, they had to complete an online questionnaire which consisted of three different parts:

1. **Personal information:** This part contained three questions regarding age, gender, and familiarity with ASMR to understand the demographic profile of the participants.
2. **User experience questionnaire:** We used the user experience questionnaire (UEQ) (Laugwitz, Held, and Schrepp, 2008) to assess how participants experienced the interaction with the system. The questionnaire contains 26 items measuring six dimensions of user experience: *Attractiveness*, *Perspicuity*, *Efficiency*, *Dependability*, *Stimulation*, and *Novelty*. Each item is represented by two opposing terms and a corresponding seven-point scale ranging from -3 (most negative) to 3 (most positive), with 0 indicating a neutral response. To maintain the integrity of the online questionnaire, we followed the UEQ Handbook's advice to remove suspicious data. As such, participant responses were considered suspicious if:
   - They gave vastly different answers for questions within the same scale, and this occurred in at least three scales.
   - They provided identical answers to more than 15 of the 26 questions.
3. **ASMR-specific questions:** The final section consisted of questions related to the participants' experience of ASMR while using ASMRcade. These questions aimed to gather information about whether participants experience ASMR in general, whether they experienced it while using ASMRcade, factors influencing the triggering or lack thereof, the time spent with the ASMRcade application, suggestions for improvement, likes and

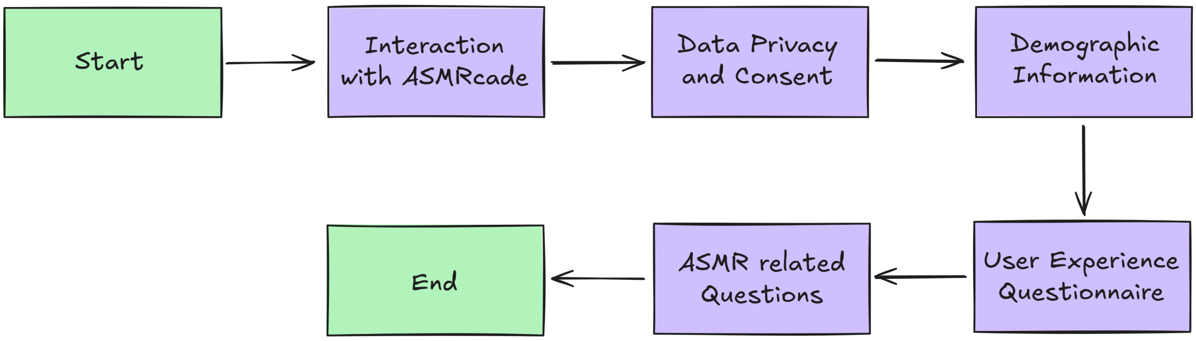


Figure 74: Flowchart of the user study.

dislikes, opinions on the generated sounds and the interaction with the system. Specifically, we asked the following questions:

- Q1: *Do you experience ASMR in general?* (Binary choice)
- Q2: *Have you experienced ASMR while using the ASMRcade?* (Binary choice). Depending on their answer, participants had to provide free-form feedback on how they interacted with the application when ASMR was triggered (Q2.1) or what they thought would have to change to experience ASMR (Q2.2).
- Q3: *How much time (in minutes) have you spent in the ASMRcade?* (Numerical-only free-form input)
- Q4: *What improvements would you suggest for a better ASMR experience?* (Free form input)
- Q5: *What did you like about the ASMRcade?* (Free form input)
- Q6: *What did you dislike about ASMRcade?* (Free form input)
- Q7: *How did you like the generated sounds?* (Free form input)
- Q8: *How did you like the interaction with the system?* (Free form input)

The procedure of the user study is shown in Figure 74.

13.4 RESULTS

In the following sections, we present the results of the online study. A total of 17 participants fully completed the survey. However, two participants reported spending 0 minutes with ASMRcade, leading to their removal before data analysis.

13.4.1 *Demographic Questions*

The demographic analysis of the remaining 15 participants revealed the following:

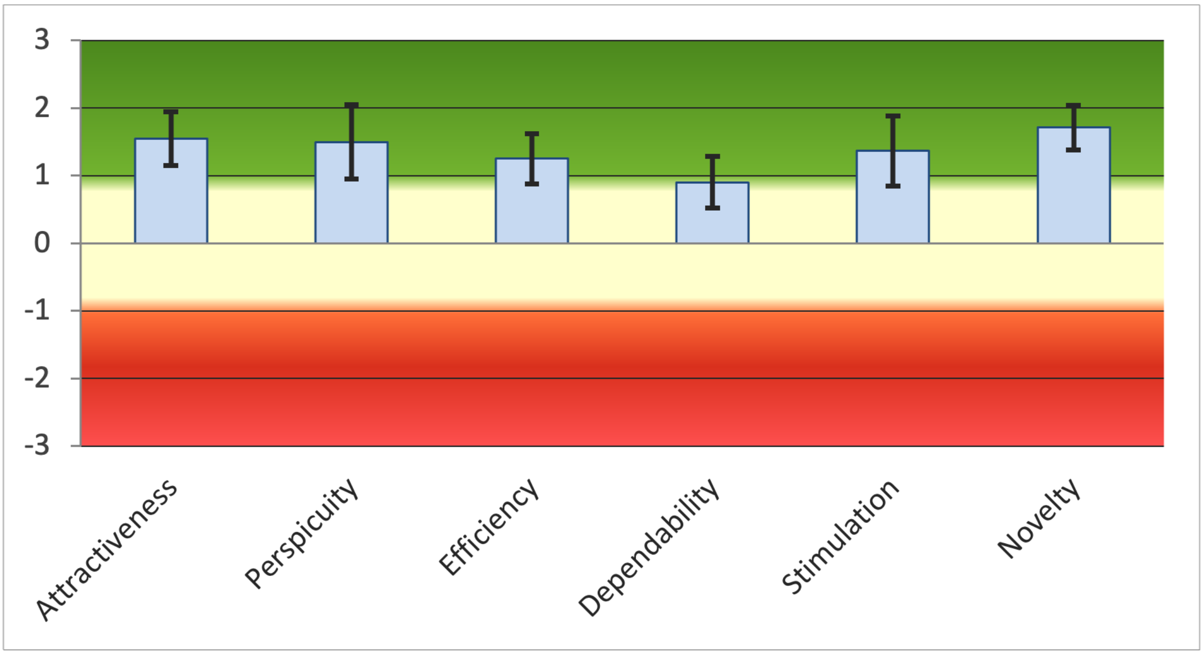


Figure 75: Mean values and variances for UEQ scales. The plot was created with the UEQ Data Analysis Tool, Version 12.

- Age: Participants ranged from 19 to 35 years old, with a mean age of 24.53 years.
- Gender: The sample included 10 males, 4 females, and 1 participant who preferred not to answer.
- Familiarity with ASMR:
  - ⋆ 1 participant never heard about it before
  - ⋆ 2 participants heard about it but never consumed ASMR content
  - ⋆ 11 participant consumed ASMR content before, but don't consume it occasionally or regularly
  - ⋆ 1 participant consumed ASMR content occasionally, but not regularly
  - ⋆ 0 participants consumed ASMR content regularly

13.4.2 *User Experience Questionnaire (UEQ)*

To maintain the integrity of the questionnaire, we removed suspicious data as suggested in the UEQ Handbook's guidelines. As such, two data entries were removed, leaving 13 valid samples for analysis. We used the analysis tool provided by Laugwitz, Held, and Schrepp (2008) to analyze participants' responses. The results for the UEQ scales are presented in Table 11 and depicted in Figure 75.

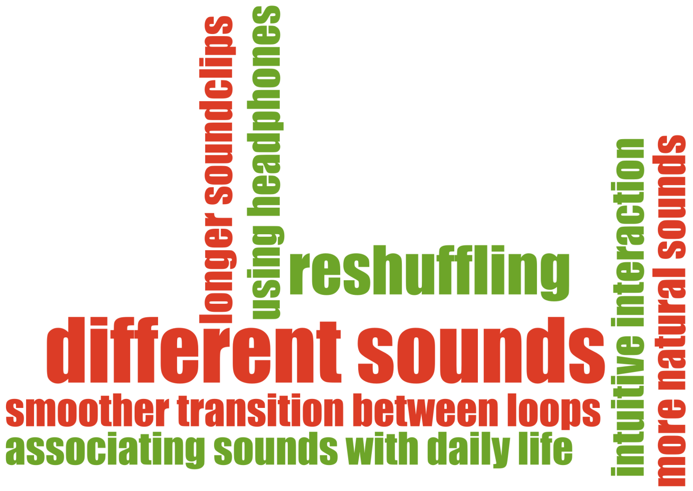


Figure 76: Word cloud resembling the responses to how the users were interacting with the application when ASMR was triggered (Q2.1, green), and what they think would need to change in order to experience ASMR (Q2.2, red).

The best-performing scale was Novelty (mean score of 1.712), followed by Attractiveness (1.551) and Perspicuity (1.5). Efficiency and Stimulation scored 1.25 and 1.365, respectively. The lowest-performing measure was Dependability, with a mean score of 0.904. However, according to the UEQ handbook, mean scores above 0.8 represent a *good evaluation*. Our application surpasses that value across all dimensions.

Scale	Mean	Variance
Attractiveness	1.551	0.54
Perspicuity	1.500	1.02
Efficiency	1.250	0.47
Dependability	0.904	0.50
Stimulation	1.365	0.90
Novelty	1.712	0.36

Table 11: UEQ Scales (Mean and Variance)

13.4.3 *ASMR related questions*

In the following, we present the results of the ASMR-specific questions from the online questionnaire. These questions aimed to assess participants' general ASMR experiences and their experiences with ASMRcade.

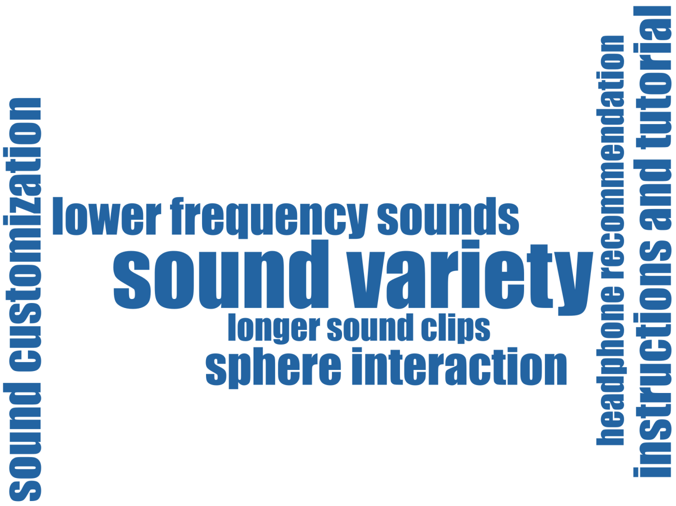


Figure 77: Improvement suggestions from participants (Q4).

- General ASMR Experience: 8 participants reported that they generally experience ASMR.
- ASMR Experience in ASMRcade: 7 participants indicated that they had experienced ASMR while interacting with ASMRcade.
- ASMR Experience Overlap: Among the 8 participants who reported general ASMR experiences, five also experienced ASMR while using ASMRcade. ASMRcade triggered an ASMR experience for two users that generally don't experience ASMR.
- Time Spent in ASMRcade: On average, participants spent 12.47 minutes interacting with ASMRcade. The minimum reported time was 3 minutes, and the maximum was 30 minutes.

To analyze the free-form feedback from participants, an inductive thematic analysis (Braun and Clarke, 2012) was conducted, which helps identify and report patterns within textual data. This qualitative method allows for understanding participants' experiences and opinions without imposing predetermined categories or theoretical perspectives and was previously used in similar studies (e.g., (Krauß et al., 2021; T. Zhang et al., 2020; Diethei et al., 2021)).

Code/word clouds were created using the themes and codes identified in the analysis to represent user feedback visually. Code clouds display words or phrases in varying font sizes, with larger sizes representing higher occurrence frequency within the data.

Figure 76 presents the code cloud to Q2.1 and Q2.2. Participants were shown one of those questions depending on how they answered regarding experiencing ASMR while interacting with ASMRcade. Green words represent feedback on how participants (that reported

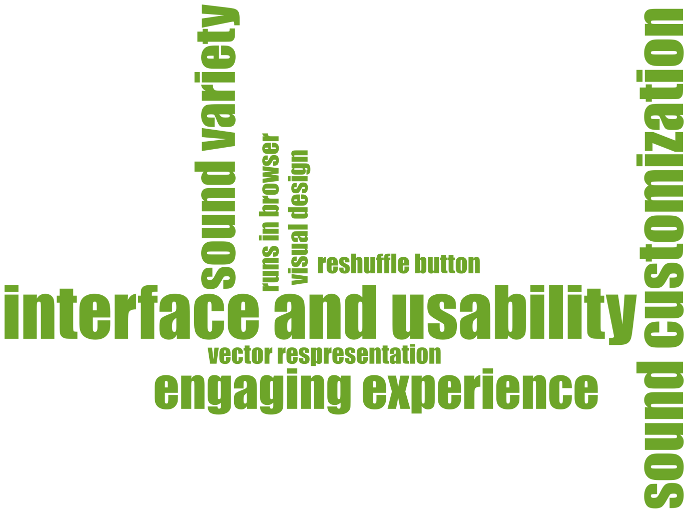


Figure 78: Word cloud resembling the responses to what users liked (Q5).

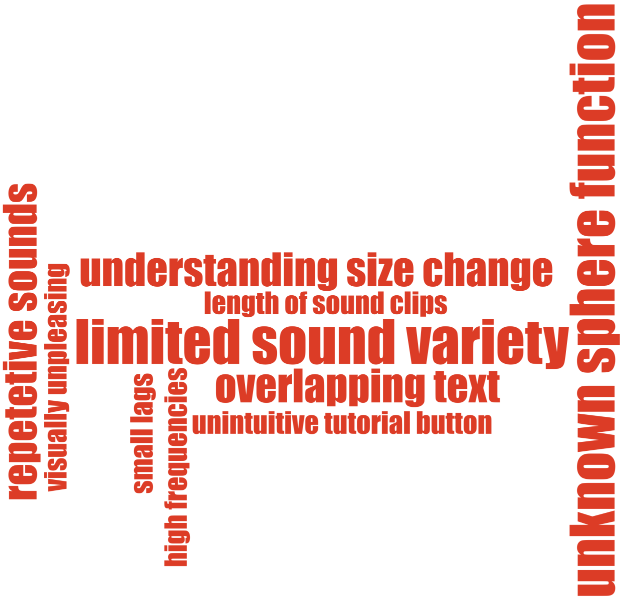


Figure 79: Word clouds resembling the responses to what users disliked (Q6)

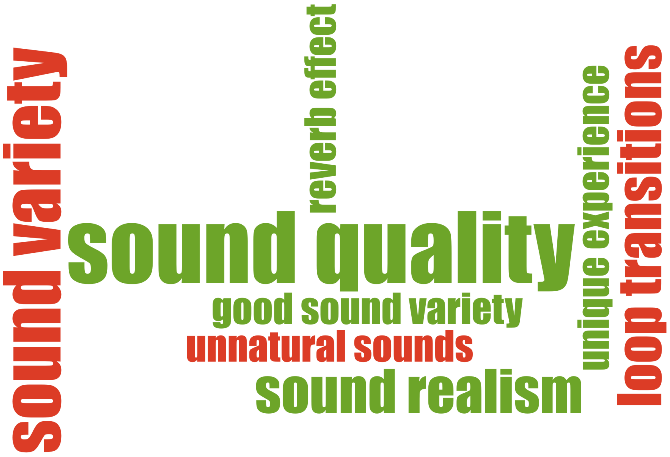


Figure 80: Feedback to the generated sounds (Q7).

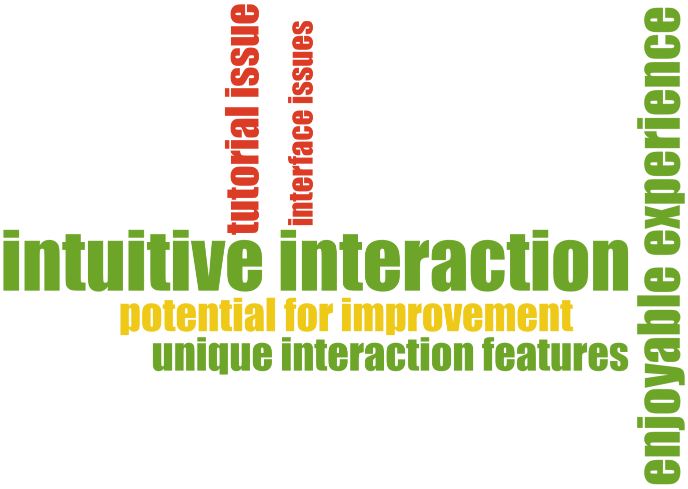


Figure 81: Feedback about the interaction experience (Q8).

experiencing ASMR while interacting with ASMRcade) interacted with ASMRcade when their ASMR was triggered. Red words represent participants' feedback that they hadn't experienced ASMR while interacting with ASMRcade about what they thought was missing or needed to change for them to experience ASMR.

The feedback by users that had experienced ASMR (green) includes aspects such as intuitive interaction (1 mention; e.g., "Changing the sounds became way more intuitive"), reshuffling (2 mentions; e.g., "Reshuffling until I experienced it"), associating sounds with daily life (1 mention; e.g., "My brain started to find references from my daily life from which the sound could be known and tried to find rhythm and movement that could be applied to the sound effects"), and using headphones (1 mention).

The feedback on what was missing to experience ASMR (red) encompasses comments on the need for longer sound clips (1 mention), different sounds (3 mentions; e.g., "Other sounds" and "Different sounds to choose from, not just tapping"), more natural sounds (1 mention; e.g., "I would like the sounds to be more natural, they felt kind of mechanical in the ASMRcade"), and smoother transition between loops (1 mention; e.g., "The transition between the loops was pretty noticeable which made it feel less natural to me").

Figure 77 displays the code cloud visualizing participants' suggestions for improvements to enhance the ASMR experience (i.e., Q4). The codes encompass various improvement suggestions, such as sound variety (5 mentions; e.g., "Different sounds," "Bigger variety of sounds," and "Diversify sound categories"), instructions and tutorial (2 mentions; e.g., "Clear instructions" and "Clickthrough tutorial"), sound customization (2 mentions; e.g., "Changing the speed"), longer sound clips (1 mention), lower frequency sounds (2 mentions; e.g., "Sound frequency adjustment" and "More profound bass sounds"), sphere interaction (2 mentions; e.g., "Sphere influence visualization" and "3D model interface improvement"), and headphone recommendation (1 mention).

Figure 78 presents the code cloud illustrating the aspects participants liked about ASMRcade (i.e., Q5). The codes highlight various positive aspects, including sound variety (4 mentions; e.g., "Endless sound possibilities," "Different patterns," and "Wide variety of sounds"), interface and usability (6 mentions; e.g., "Intuitive interaction," "Nice interface," "Minimalistic interface," and "Easy to use"), sound customization (5 mentions; e.g., "Volume and reverb control," "Realistic sounds," "Customizable sounds," and "User control over sounds"), reshuffle button (1 mention), browser compatibility (1 mention; e.g., "Runs in-browser"), vector representation (1 mention; e.g., "Vector representation was quite nice"), engaging experience (4 mentions; e.g., "Engaging and creative," "Interactive experience," and

"Unique experience"), and visual design (1 mention; e.g., "Visually pleasing").

Figure 79 presents the code cloud for user feedback regarding the aspects they disliked about ASMRcade (i.e., Q6). Unintuitive tutorial button (1 mention; e.g., "I disliked the question mark button as a Tutorial being very small."), Understanding size change (2 mentions; e.g., "I didn't understand the size change. I couldn't hear any difference with changing the size"), Unknown sphere function (3 mentions; e.g., "I didn't really understand what manipulating the spheres actually does", "fine-tuning is a little bit confusing"), Small lags (1 mention), Overlapping text (2 mentions; e.g., "bubble thingy laid over the text"), length of sound clips (1 mention), high frequencies (1 mention), Repetitive sounds (2 mentions; e.g., "Some sounds seemed to appear repetitively even though different orbs were manipulated", Limited sound variety (3 mentions; e.g., "too few different sounds", "it can only generate relatively unexciting sounds (i.e. tapping)"), Visually unpleasing (1 mention).

Figure 80 presents the code cloud generated based on participants' feedback on the generated sounds in ASMRcade (i.e., Q7). The positive feedback (green) includes aspects such as sound quality (4 mentions; e.g., "The sounds were pleasant" and "The tapping sounded convincing"), sound realism (2 mentions; e.g., "The knocking sounds were pleasant"), good sound variety (1 mention), reverb effect (1 mention) and unique experience (1 mention; e.g., "The loops added a unique and enjoyable aspect to the experience").

On the other hand, negative feedback (red) encompasses comments on the need for greater sound variety (3 mentions; e.g., "Some other sounds would've been nice" and "More variety"), loop transitions (2 mentions; e.g., "Regulate the length of the sounds" and "The transition between loops felt disruptive"), and unnatural sounds (1 mention; e.g., "They didn't feel very natural to me").

Figure 81 presents the code cloud for user feedback regarding their interaction with ASMRcade (i.e., Q8). It is composed of three colors: green for positive feedback, red for negative feedback, and yellow for neutral feedback. The codes highlight different aspects, including Enjoyable experience (3 mentions; e.g., "It was an enjoyable experience", "It was interesting and was fun to use"), Intuitive interaction (5 mentions; e.g., "The interaction was easy and intuitive", "It was mostly intuitive", "Easy to understand, learning by doing worked fast"), Tutorial issue (2 mentions; e.g., "the Tutorial problem", "Would have liked to be able to pause/click through the tutorial myself"), Unique interaction features (2 mentions; e.g., "I was pleasantly surprised that the system placed the balls at the maximum possible distance when I moved them too far", "Nice imaging of the different bubbles as an input vector for the neuronal network"), Interface issues (1 mention; e.g., "Bubbles partly laid over the text"), Potential for improvement (2 mentions; e.g., "More possibilities to customize and play around with different sounds would make it very enjoyable").

13.5 DISCUSSION

USER EXPERIENCE Overall, the User Experience Questionnaire (UEQ) results indicate a positive user experience with ASMRcade. The highest mean scores were observed for Novelty, Attractiveness, and Perspicuity, suggesting that ASMRcade was perceived as an innovative and appealing solution with a clear and understandable design. In the open feedback, one user stated, "It's a very interesting experience that I haven't had like that before. I really enjoyed customizing the sounds to what I wanted to hear, which is something I can't do when I'm watching an ASMR video." The relatively lower scores for Efficiency, Stimulation, and Dependability indicate room for improvement in these more pragmatic and usability-related scores. The efficiency and dependability of the application might be affected by the inherent unpredictability of WaveGAN. Users may need to manipulate multiple spheres to produce a pleasant sound, which might need to be more efficient. One user mentioned, "I felt like manipulating the spheres didn't really change much".

Overall, as the Pragmatic Quality scales (Perspicuity, Efficiency, Dependability) had lower mean scores than the Hedonic Quality scales (Stimulation, Originality), one can conclude that participants found ASMRcade to be more enjoyable and stimulating than efficient and dependable. As such, ASMRcade might not be reliable for generating desired sounds on every interaction because of the GANs' inherent unpredictability. This finding suggests that while the application offers an engaging and novel experience, its practical aspects may need further refinement to enhance user satisfaction. However, we also note that goal-oriented qualities may be less critical for an application designed for relaxation and de-stressing, such as ASMRcade. Furthermore, higher scores in Hedonic qualities may be attributed to the use case itself, as experiencing ASMR is a relaxing and immersive experience in itself (Barratt and Davis, 2015; Poerio et al., 2018). One participant's open feedback supports this notion: "It created curiosity and started to engage the creative mind."

ASMR EXPERIENCES More than half of the participants reported generally experiencing ASMR, with almost the same proportion experiencing ASMR while interacting with ASMRcade. The overlap between these groups demonstrates ASMRcade's potential to induce ASMR in users susceptible to the phenomenon. However, not all participants who generally experienced ASMR reported experiencing it with ASMRcade, suggesting that the application may need further improvements to cater to a broader range of ASMR triggers and preferences. On the other hand, certain participants were able to experience ASMR that did not experience ASMR before. This observation indicates that the interactive component we introduced through the ASMRcade application might have advantages over passive ASMR consumption. Interacting with an ASMR content generator can even be seen as a new category of ASMR triggers, which, like other categories, may work very well for some users but less for others. Here, future work has to dive more into detail on how exactly interactive and passive ASMR consumption differs in triggering ASMR experiences.

The open feedback collected from participants provided valuable insights into their experiences and opinions regarding ASMRcade, which complemented the quantitative data obtained from the User Experience Questionnaire (UEQ). By applying an inductive thematic analysis approach to the open feedback, several recurring themes were identified that shed light on the strengths and areas for improvement of the application.

One of the key strengths of ASMRcade highlighted by the participants was the intuitive interaction. Participants found navigating and manipulating the spheres to create different sounds easy. For example, one participant noted that "Changing the sounds became way more intuitive" once they got the hang of the system. This feedback corresponds with the above-average scores for Attractiveness and Perspicuity in the UEQ results. However, participants also mentioned that some aspects of ASMRcade's sphere functions needed clarification. One user stated, "I didn't really understand what manipulating the spheres actually does," while another mentioned, "Fine-tuning is a little bit confusing." These comments suggest that some users struggled to comprehend how the sphere manipulation impacted the generated sounds. Addressing this issue by providing clearer explanations about the function and limits of sphere interaction, visual cues, or tooltips could help users better understand the connection between sphere manipulation and the resulting sounds, thus enhancing their overall experience. One way to give visual cues would be to do a Vector Impact Analysis, like Schlagowski, Mertes, and André (2021) did for their drum sequencer . After analyzing their impact on the generated sounds, spheres can be sorted accordingly, and shaders could be used to indicate individual spheres' impact on different frequency bands.

The open feedback also revealed that participants appreciated the sound customization options available in ASMRcade but desired more advanced customization features. Users found the ability to manipulate volume and reverb controls helpful, as one participant noted, "I liked the option that you could separately change the volume and the Reverb" However, some users expressed a desire for even more control over sound characteristics, such as speed and frequency adjustments. For example, one participant suggested, "I would also like there to be a slider where I could manipulate the speed of the sound".

Further constructive feedback on areas that could be improved was provided. For instance, several users mentioned that the application would benefit from a more diverse selection of sounds. One participant suggested, "Diversify the sound categories". Another user mentioned that the sound transitions could be smoother, stating, "The transition between the loops was pretty noticeable, which made it feel less natural to me."

Furthermore, some participants found certain features to need to be clarified or to be more intuitive within ASMRcade. For example, one user noted that the tutorial button needed to be bigger and easier to notice. Another participant mentioned difficulties understanding the impact of changing the sphere size slider, which hides certain manipulatable spheres: "I didn't understand the size change; I couldn't hear any difference with changing the size." To address these issues, the application could be refined by improving the visibility and accessibility of the tutorial, as well as providing more precise explanations of what the size slider does and how it affects the generated sounds.

As the by far most recurring theme was the diversity of sounds, the logical next step should be to train the WaveGAN not only on tapping sounds but on a broad spectrum of different ASMR audio triggers that exist. These could include scratching, whispering, or other mouth-made sounds. Further improvements could address clarifying the sphere functions, refining sound transitions, refining confusing features, and providing clearer instructions or a more interactive tutorial for new users. By addressing these areas, ASMRcade could further enhance user satisfaction and provide a more engaging and enjoyable experience for users seeking relaxation and stress relief through ASMR.

13.6 CONCLUSION

In this chapter, we presented ASMRcade, an application for interactively exploring and generating personalized ASMR audio triggers. Using this application, users could change the input vector for a WaveGAN capable of transforming that input vector to ASMR-triggering tapping sounds by manipulating the synthesized sounds via a web-based user interface. A first explorative user study indicates that some users can benefit from the interactive approach and that it might be able to broaden accessibility for ASMR experiences as some users had their first successful ASMR triggers with our system.

From user experience questionnaires and the qualitative feedback, we conclude that users appreciated the intuitive interaction with the application, which was also perceived as innovative and appealing.

However, users wished to understand better how their input affected the generated sounds. Users especially enjoyed the sound customization options and requested further control over the speed and frequency of the generated sound. Furthermore, while the sound quality was good enough, many users asked for more variety in the generated sounds. As such, future work has to extend the system to a larger set of ASMR sound categories.

To conclude, the study is a proof of concept for a novel interactive approach for generating ASMR triggers. The substantial share of users that experienced ASMR and their interest in improving ASMRcade shows great potential for an interactive ASMR experience as a suitable alternative to conventional ASMR content.

Part VII

CONTRIBUTIONS & OUTLOOK

14

CONTRIBUTIONS

In the following, the contributions of this thesis are summarized - categorized into *conceptual*, *technical*, and *empirical* contributions. Thereby, they will be contextualized with regard to the research questions of this thesis (as defined in Chapter 1):

- *RQ1: How can we use GANs to augment small datasets without being stuck in the original dataset distribution?*

- *RQ2: How can we use GANs to generate realistic Counterfactual Explanations for image classifiers?*

- *RQ3: How can we use GANs to build explanation systems that communicate information about irrelevant features?*

- *RQ4: How can we use GANs to synthesize continuously conditioned images by using only discretely labeled training data?*

- *RQ5: How can we use GANs to enhance an AI based job interview training system in order to give personalized, realistic and comprehensible feedback?*

- *RQ6: How can we use GANs to build an interactive ASMR experience?*

14.1 CONCEPTUAL CONTRIBUTIONS

In this work, some new concepts were introduced:

- Existing approaches to augment datasets either rely on rather simple transformation algorithms or they are based on training generative models on the available data without constraints. Both those approaches have in common that the augmented data still follows the original data distribution quite strictly - which in turn hinders the information that is gained through the new data. In Part ii, to address RQ1, two approaches were introduced to overcome those limitations of being stuck in a dataset distribution when augmenting datasets.

The first approach (see Chapter 4) proposed to interpolate through the latent space of a GAN by using an evolutionary algorithm. That GAN was trained with the latent space sampled from a uniform noise distribution. As such, the *original* data is encoded in a latent space following that uniform distribution. Using an evolutionary algorithm to search for samples that help the training task at hand allows the latent input to break out of that uniform distribution - and as such is able to generate samples that actually hold new information.

The second approach (see Chapter 5) used a Style-Conversion GAN to map artificial labels to training data. Here, a paradigm shift was proposed: Instead of augmenting data itself, the labels (here in the form of segmentation masks) were augmented and successively mapped to new data points. By doing so, the original data distribution does not have to be approximated directly - allowing to synthesize data pairs that bring information outside of that distribution to the dataset.

- Counterfactual Explanations are a form of explanation that alter an input to an AI system in a way that the decision of that AI changes. However, generating such explanations is still a challenge. On the other hand, in recent years models for style transfer have become mature and efficient. As such, Chapter 7 proposes to consider a *decision* of an AI as a *style*. By doing so, style transfer models become applicable to the task of generating counterfactual explanation, addressing RQ2.

- Existing paradigms of explaining decisions of AI systems mostly rely on communicating information about *relevant* features. However, communicating information about *irrelevant* features might be similarly important for a comprehensive understanding of a model. As such, in Chapter 8, a completely new concept - named *Alterfactual Explanations* - was introduced. That new paradigm aims to directly communicate irrelevant information about an AI decision to the user. As the paradigm is based on showing an alternative reality where the decision of an AI does not change, it uses the concepts of factual explanations to effectively addressing RQ3.

- Although GANs are capable of generating highly realistic images, equipping them with the ability to be controlled in a fine-grained manner commonly is dependent on datasets that cover that granularity. As in many scenarios such data is not available, the concepts introduced in Chapter 10 propose to interpolate through the label space of a synthesis network that was trained on discretely labeled data. By doing so, the desired features can be controlled in a continuous way during the image synthesis process - without the need for continuously labeled training data. Therefore, RQ4 is effectively addressed.

- Feedback systems are a proper way to increase users' self-esteem. However, generating actionable feedback is not a trivial task. In Chapter 11 and Chapter 12, it is proposed to make use of concepts from the field of XAI in order to generate such feedback. Instead of explaining a classifier in order to make that classifier's decision transparent, we use counterfactual reasoning processes to explain a user's behavior. As such, generated counterfactual explanations for the behaviour serve as feedback for a training system - here, specifically a job interview training system. As such, RQ5 was addressed.

- ASMR is a popular method to reduce stress and as such contribute to well-being. ASMR is highly related to the state of *flow*. Although flow is coupled to being actively involved in a task, ASMR is commonly seen as a purely passive experience. So far, there is no research on the effects of incorporating *interactive* components to ASMR. Chapter 13 contributed to RQ6, as the concept of *Interactive ASMR* was introduced. There, instead of seeing ASMR only as a passive experience, the user is directly incorporated into the sound synthesis process.

14.2 TECHNICAL CONTRIBUTIONS

While technical implementations for all the approaches and experiments presented in this paper were needed and developed, some of them might have particularly high potential for being used in future research and development. Therefore, most of the implementations that were developed for answering the research questions of this thesis were open-sourced and made publicly available.[1] The main technical contributions are as follows:

- In Chapter 7 and Chapter 9, technical frameworks for generating both counterfactual explanations and alterfactual explanations for binary image classifiers were presented. Those frameworks contributed in answering RQ2. They allow to create explanations for various kinds of scenarios and are easy to adapt. Both frameworks leverage the idea of using mechanisms from

[1] The open-source repositories can be found on the GitHub page of the Chair for Human-Centered Artificial Intelligence: `https://github.com/hcmlab`

the field of style conversion for the synthesis of explanations. As such, they use established network architectures and extend them with additional loss components to achieve the desired explanation capabilities. Those additional loss components made sure that the classifier's decisions are taken into account when performing the style conversion step - steering the decision into a direction that satisfies the respective explanation mechanism's definition. They both work independently of the specific neural network architecture of the classifier that should be explained - the only requirement is that the model's gradients can be backpropagated through the model. As such, both the frameworks can easily be adapted to new use-cases and scenarios.

- In Chapter 4, a technical approach to data augmentation for the audio domain was presented. The approach includes the training of a GAN to model the initial data distribution of the dataset. Subsequently, an evolutionary algorithm was deployed to steer the GAN to synthesize new data that helps to improve the classification performance of a classifier model. The approach can easily be adapted to further use-cases, as it only relies on the definition of features to build an appropriate fitness function for the underlying evolutionary algorithm. Here, for our specific use-case, we made use of spectral features that can easily be calculated by proprietary libraries.

- Chapter 5 presented two methods for data augmentation in the image domain. While the first one incorporates a hand-crafted solution to model defects in carbon fibres, the second one is completely data-driven. As such, it can be adapted for further scenarios without the need for excessive engineering overhead. Both approaches aim to synthesize new label masks which subsequently can be processed by a GAN model in order to build new training pairs for semantic segmentation.

- Chapter 11 and Chapter 12 introduced applications for giving verbal and visual feedback for job interview training, effectively contributing to R5. For the first approach, we extended the counterfactual explanation generation framework that we already presented in Chapter 7, which was specifically modified to work with the feature representations that we used to assess an interviewee's performance. The second approach introduced a procedural approach to finding counterfactual feedback, which was complemented by a GAN-based method to convert counterfactual skeletons to realistic looking images of humans. Both applications can be used to conduct studies and also for real job interview trainings.

- In Chapter 13, an application for experiencing interactive ASMR was presented, addressing RQ6. Therefore, a WaveGAN model was trained to synthesize ASMR tapping sounds. The model was integrated into a user-friendly GUI, allowing for a seemless interaction. Further, the whole system was deployed as a web application, making it accessible to a broader audience. The application allows to further study the new concept of interactive ASMR. Also, by using it, everyone can experience that paradigm by themselves.

14.3 EMPIRICAL CONTRIBUTIONS

Besides the conceptual and technical contributions, in this thesis, results of various conducted user studies have been presented:

- In Chapter 7, a user study revealed that, in the chosen medical use-case, counterfactual explanations lead to significantly better results regarding mental models, explanation satisfaction, trust, emotions, and self-efficacy than two state-of-the art systems that work with saliency maps, namely LIME and LRP. As such, this study assessed if we successfully solved RQ2.

- With regard to RQ3, in Chapter 8, we presented a user study showing that alterfactual explanations are suited to convey an understanding of different aspects of the AI's reasoning than established counterfactual explanation methods.

- In the study in Chapter 9, we further demonstrated that in a prediction task, where the classifier's prediction had to be anticipated by looking at the explanations, users performed significantly better when they were provided with explanations that included alterfactual explanations compared to users that did not see alterfactual explanations. Again, this study assessed if we successfully could solve RQ2, while also contributing to RQ3.

- The study in Chapter 11 examined the usefulness of our job interview training system that generates verbal feedback. Here, besides approving that our system is helpful and comprehensible, it was found that recommendations for such use-cases should be highly specific, addressing RQ5.

- In Chapter 13, we presented a user study to answer RQ6. That study revealed that the concept of *interactive* ASMR has the potential to induce the state of ASMR for people that are generally not able to experience it when confronted with *passive* ASMR.

15

OUTLOOK ON FUTURE WORK

Research is never finished. As such, the following sections give an outlook on future work that might explore the topics of this thesis even further. Therefore, the main parts of this thesis are revisited.

15.1 ROBUSTNESS

Data augmentation techniques must be equipped with the ability to generate more diverse data.

In Part ii, approaches for data augmentation were introduced. While those approaches helped the AI to become better, they relied on rather complex mechanisms to control the networks' outputs to a high degree. The model that we used in Chapter 4 was steered by the fitness function of an evolutionary algorithm. In Chapter 5, we used a pix2pix model which takes artificially generated labels as input (either generated by a procedural algorithm or by another GAN). As such, in both approaches, mechanisms exist that slightly limit the diversity of the outputs. The evolutionary algorithm tries to find *specific* solutions for data points that are missing in the dataset, and the pix2pix architecture that we used does not use a random vector in its input (although the label synthesis steps have). However, the whole aim of conducting data augmentation is to *diversify* a dataset in order to make trained models more robust. Here, the approaches could be enhanced in future work. For the evolutionary-based approach (see Chapter 4), this could be achieved by adding more feature dimensions to the fitness function. As such, instead of only searching for data points that show *one* certain feature value, multiple feature combinations could be considered at a time. By doing so, multiple feature combinations could drastically increase the variety of the generated data. For the second approach (see Chapter 5), additionally to the dropout layers that are already included in our version of the model, a random input component could be added to the pix2pix input. In order to do so, solutions would have to be found that lead the GAN to not just ignoring that noise input (Isola et al., 2017).

15.2 EXPLAINABILITY

Appropriate metrics have to be found for evaluating alterfactual explanations.

In Part iii, the concept of *Alterfactual Explanations*, alongside with a technical approach to generate those explanations, was introduced. Those alterfactual explanations are, similar to counterfactual explanations, grounded in the concept of counterfactual reasoning. For counterfactual explanations, a variety of metrics have established which define *good* counterfactual explanations. The most mentioned ones in literature are *Proximity*, *Sparsity*, *Diversity*, *Feasibility* and *Plausibility* (Y.-L. Chou et al., 2022). For alterfactual explanations, on the other hand, appropriate metrics still have to be defined. Certainly, not all of the metrics apply that are use for counterfactual explanations. For example, achieving a high proximity (e.g., the distance of the explanation from the instance to be explained (Verma, Dickerson, and Hines, 2020)) or sparsity (e.g., focusing on as few features as possible (Keane and Smyth, 2020)) even contradicts the definition of an alterfactual explanation. Other metrics, like aiming for a high diversity in the explanations, seem also desirable for alterfactual explanations. However, there are metrics - like plausibility - where it is not so easy to tell if they are desirable for alterfactual explanations or not. Non-plausible explanations, for example, might include feature changes that are impossible to observe in reality. E.g., if an explanation changes an *immutable* feature - such as race - then it is not plausible (Y.-L. Chou et al., 2022). However, in an alterfactual explanation, changing the race implies that race is irrelevant for the AI to be explained. As such, even if the explanation might not be plausible, it directly communicates that the AI has no bias with regard to race. However, there might be scenarios where plausibility *is* desired. Future work has to dive more into these topics - it has to examine which metrics should be applied to the evaluation of alterfactual explanations.

15.3 EXPRESSIVENESS

Ethical considerations are important when generating human faces.

In Part iv, an approach was introduced that is able to generate face images the show emotions steerable in a dimensional emotion space. That work joins the ranks of recent trends and developments in the broader field of generative AI, where generating data with human visuals is an actively researched topic (M. Kim et al., 2023; Z. Huang et al., 2023; X. Shen et al., 2024; M. Liu et al., 2021). Especially since Diffusion Models came around in 2020 (Ho, A. Jain, and Abbeel, 2020) - a category of generative models that produces even more realistic outputs than GANs, with the drawback of taking substantially more time for inference, and thus limited to non-realtime applications - it has become possible to generate outputs that are truly indistinguishable from real data. While those advances come with great potential for making AI systems more empathic by equipping them with highly relatable, human-like visuals, they don't come without drawbacks. Specifically when generating human faces, attention has to be paid to ethical concerns. For example, when equipped with appropriate training data, generative models are able to generate convincing deepfakes of specific people. Those deepfakes can be misused for various malicious purposes - be it identity theft, faking legal evidences, or generating adult content. The only requirement that state-of-the-art algorithms have here is the availability of training data that covers the needed data distribution. However, when looking on our approach of generating human faces from a more conceptual view, it enables to generate realistic data more easily even if only parts of the data distribution is in the training set. In our case, the training set only consisted of categorical, *prototypical* emotions. With our label interpolation approach, we were able to generate those intermediate steps that were *not* in the training data. We restricted ourselves to generating emotional faces - however, future work has to find ways to prevent such algorithms, approaches and systems from generating malicious content.

15.4 FEEDBACK SYNTHESIS

Feedback systems should keep the human in the loop.

In Part V, we introduced approaches for generating verbal and visual counterfactual feedback for job interview training systems. There, the user was shown how he or she could have behaved better in order to appear more engaged in the interview. However, what that approach misses is an interactive component of *choosing* the explanation. Generally, in complex scenarios such as job interviews, there is not just one factor determining the outcome. As our approach (and also related works (Alexander Heimerl et al., 2022a; Alexander Heimerl et al., 2022b)) synthesizes a data point that alters the input just as much as needed for achieving a better outcome, it is limited to giving one single counterfactual state. However, especially in a training scenario, the user might want to decide to train different aspects than proposed by the counterfactual generation approach. For example, imagine that the counterfactual feedback proposes to keep the arms more open in order to appear more engaged - but the user has been in the gym all day, his arms are exhausted, and he only wants to focus on facial expressions in that training session. More generally speaking, when building feedback systems for end users, it is important that not only the *user* receives feedback, but also the *system* receives *feedback on how to give feedback*. As such, future work should try to build more comprehensive human-in-the-loop feedback systems that interactively give users more choice.

15.5 INTERACTING WITH GANS

Other modalities have to be explored to improve the accessibility of Interactive ASMR.

In Part vi, the concept of *Interactive ASMR* was introduced. Also, we presented a web application for experiencing that new concept. There, users could play around with a graphic visualization of a GANs latent input space. We found that the paradigm of introducing an interactive component to the usually passive experience of ASMR has great potential. In our work, we only focused on one specific interaction modality - playing around with a graphical user interface. However, in order to make the concept accessible to an even broader range of people, more interaction designs have to be evaluated for being used for interactive ASMR. For instance, ASMR is often used as a "tool" to fall asleep. McErlean and Banissy (2017) report that 41% of participants of a user study stated that they used ASMR to help them fall asleep. It is obvious that using a graphical user interface while trying to fall asleep might be hindering. However, there are modalities that still could be used here. For instance, physiological signals are a promising alternative to steer a system that synthesizes ASMR sounds, as it has been shown that the consumption of ASMR content has an effect on the heart rate of users (Engelbregt et al., 2022). This correlation could be exploited in order to build a system that uses the heart rate as input to an optimization model in order to build a perfectly personalized ASMR experience while trying to fall asleep. As the whole concept of interactively incorporating user input into the ASMR synthesis process is completely new, future work has infinite possibilities to extend the work presented in this thesis.

Part VIII

APPENDIX

A

OWN PUBLICATIONS

A.1 PUBLICATIONS RELEVANT FOR THIS DISSERTATION

Some ideas and figures have appeared previously in the following publications:

- Heimerl, Alexander, **Silvan Mertes**, Tanja Schneeberger, Tobias Baur, Ailin Liu, Linda Becker, Nicolas Rohleder, Patrick Gebhard, and Elisabeth André (2022). "Generating personalized behavioral feedback for a virtual job interview training system through adversarial learning." In: *International Conference on Artificial Intelligence in Education*. Springer, pp. 679–684.
- **Silvan Mertes**, Alice Baird, Dominik Schiller, Björn W Schuller, and Elisabeth André (2020). "An evolutionary-based generative approach for audio data augmentation." In: *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, pp. 1–6.
- **Silvan Mertes**, Tobias Huber, Christina Karle, Katharina Weitz, Ruben Schlagowski, Cristina Conati, and Elisabeth André (2024). "Relevant Irrelevance: Generating Alterfactual Explanations for Image Classifiers." In: *33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*. AAAI Press.
- **Silvan Mertes**, Tobias Huber, Katharina Weitz, Alexander Heimerl, and Elisabeth André (2022). "Ganterfactual—counterfactual explanations for medical non-experts using generative adversarial learning." In: *Frontiers in artificial intelligence* 5, p. 825565.
- **Silvan Mertes**, Christina Karle, Tobias Huber, Katharina Weitz, Ruben Schlagowski, and Elisabeth André (2022). "Alterfactual Explanations–The Relevance of Irrelevance for Explaining AI Systems." In: *IJCAI 2022 Workshop on XAI*.
- **Silvan Mertes**, Florian Lingenfelser, Thomas Kiderle, Michael Dietz, Lama Diab, and Elisabeth André (2021). "Continuous emotions: exploring label interpolation in conditional generative adversarial networks for face generation." In: *Deep Learning Theory and Applications*.

- **Silvan Mertes**, Andreas Margraf, Steffen Geinitz, and Elisabeth André (2020). "Alternative data augmentation for industrial monitoring using adversarial learning." In: *Deep Learning Theory and Applications*, pp. 1–23.
- **Silvan Mertes**, Andreas Margraf, Christoph Kommer, Steffen Geinitz, and Elisabeth André (2020). "Data augmentation for semantic segmentation in the context of carbon fiber defect detection using adversarial learning." In: *International Conference on Deep Learning Theory and Applications*. Springer.
- **Silvan Mertes**, Dominik Schiller, Florian Lingenfelser, Thomas Kiderle, Valentin Kroner, Lama Diab, and Elisabeth André (2020). "Intercategorical Label Interpolation for Emotional Face Generation with Conditional Generative Adversarial Networks." In: *International Conference on Deep Learning Theory and Applications*. Springer, pp. 67–87.
- **Silvan Mertes**, Marcel Strobl, Ruben Schlagowski, and Elisabeth André (2023). "ASMRcade: Interactive Audio Triggers for an Autonomous Sensory Meridian Response." In: *Proceedings of the 25th International Conference on Multimodal Interaction*, pp. 70–78.

A.2 OTHER PUBLICATIONS

Besides the publications that I used in this thesis, I have (co-)authored the following peer-reviewed works:

- Baird, Alice, **Silvan Mertes**, Manuel Milling, Lukas Stappen, Thomas Wiest, Elisabeth André, and Björn W Schuller (2021). "A Prototypical Network Approach for Evaluating Generated Emotional Speech." In: *Proc. Interspeech 2021*, pp. 3161–3165.
- Boulogne, Luuk H., Julian Lorenz, Daniel Kienzle, Robin Schön, Katja Ludwig, Rainer Lienhart, Simon Jégou, Guang Li, Cong Chen, Qi Wang, Derik Shi, Mayug Maniparambil, Dominik Müller, **Silvan Mertes**, Niklas Schröter, Fabio Hellmann, Miriam Elia, Ine Dirks, Matías Nicolás Bossa, Abel Díaz Berenguer, Tanmoy Mukherjee, Jef Vandemeulebroucke, Hichem Sahli, Nikos Deligiannis, Panagiotis Gonidakis, Ngoc Dung Huynh, Imran Razzak, Mohamed Reda Bouadjenek, Mario Verdicchio, Pasquale Borrelli, Marco Aiello, James A. Meakin, Alexander Lemm, Christoph Russ, Razvan Ionasec, Nikos Paragios, Bram van Ginneken, and Marie-Pierre Revel (2024). "The STOIC2021 COVID-19 AI challenge: Applying reusable training methodologies to private data." In: *Medical Image Anal.* 97, p. 103230.
- Hallmen, Tobias, **Silvan Mertes**, Dominik Schiller, and Elisabeth André (2022). "An efficient multitask learning architecture for affective vocal burst analysis." In: *The ACII Affective Vocal Bursts (A-VB) Workshop & Competition*.

- Hallmen, Tobias, **Silvan Mertes**, Dominik Schiller, Florian Lingenfelser, and Elisabeth André (2023). "Phoneme-Based Multi-task Assessment of Affective Vocal Bursts." In: *International Conference on Deep Learning Theory and Applications*. Springer, pp. 209–222.
- Heimerl, Alexander, Pooja Prajod, **Silvan Mertes**, Tobias Baur, Matthias Kraus, Ailin Liu, Helen Risack, Nicolas Rohleder, Elisabeth André, and Linda Becker (2024). "The ForDigitStress Dataset: A Multi-Modal Dataset for Automatic Stress Recognition." In: *IEEE transactions on affective computing*.
- Hellmann, Fabio, Elisabeth André, Mohamed Benouis, Benedikt Buchner, and **Silvan Mertes** (2024). "Anonymization of Faces: Technical and Legal Perspectives." In: *Datenschutz und Datensicherheit-DuD* 48.6, pp. 364–367.
- Hellmann, Fabio, **Silvan Mertes**, Mohamed Benouis, Alexander Hustinx, Tzung-Chien Hsieh, Cristina Conati, Peter Krawitz, and Elisabeth André (2024). "Ganonymization: A gan-based face anonymization framework for preserving emotional expressions." In: *ACM Transactions on Multimedia Computing, Communications and Applications*.
- Holzinger, Johanna, Alexander Heimerl, Ruben Schlagowski, Elisabeth André, and **Silvan Mertes** (2024). "A Machine Learning-Driven Interactive Training System for Extreme Vocal Techniques." In: *Proceedings of the 19th International Audio Mostly Conference: Explorations in Sonic Cultures*, pp. 348–354.
- Huber, Tobias, Maximilian Demmler, **Silvan Mertes**, Matthew L Olson, and Elisabeth André (2023). "Ganterfactual-rl: Understanding reinforcement learning agents' strategies through visual counterfactual explanations." In: *AAMAS*.
- Huber, Tobias, **Silvan Mertes**, Stanislava Rangelova, Simon Flutura, and Elisabeth André (2021). "Dynamic difficulty adjustment in virtual reality exergames through experience-driven procedural content generation." In: *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, pp. 1–8.
- Kiderle, Thomas, Hannes Ritschel, Kathrin Janowski, **Silvan Mertes**, Florian Lingenfelser, and Elisabeth André (2021). "Socially-aware personality adaptation." In: *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE, pp. 1–8.
- Kiderle, Thomas, Hannes Ritschel, **Silvan Mertes**, and Elisabeth André (2023a). "6 Multimodal humor in human-robot interaction." In: *Interactional Humor: Multimodal Design and Negotiation* 10, p. 169.
- Kiderle, Thomas, Hannes Ritschel, **Silvan Mertes**, and Elisabeth André (2023b). "Multimodal Irony for Virtual Characters." In: *Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents*, pp. 1–4.

- Kirchhoff, Aron, Alexander Hustinx, Behnam Javanmardi, Tzung-Chien Hsieh, Fabian Brand, Fabio Hellmann, **Silvan Mertes**, Elisabeth André, Shahida Moosa, Thomas Schultz, et al. (2025). "GestaltGAN: Synthetic photorealistic portraits of individuals with rare genetic disorders." In: *European Journal of Human Genetics*, pp. 1–6.
- Kuch, Johanna Magdalena, Jauweiria Nasir, **Silvan Mertes**, Ruben Schlagowski, Christian Becker-Asano, and Elisabeth André (2024). "Evaluating Gender Ambiguity, Novelty and Anthropomorphism in Humming and Talking Voices for Robots." In: *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE.
- Müller, Dominik, **Silvan Mertes**, Niklas Schroeter, Fabio Hellmann, Miriam Elia, Bernhard Bauer, Wolfgang Reif, Elisabeth André, and Frank Kramer (2023). "Towards automated COVID-19 presence and severity classification." In: *Caring is Sharing–Exploiting the Value in Data for Health and Innovation*. IOS Press, pp. 917–921.
- Rijn, Pol van, **Silvan Mertes**, Kathrin Janowski, Katharina Weitz, Nori Jacoby, and Elisabeth André (2024). "Giving robots a voice: Human-in-the-loop voice creation and open-ended labeling." In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–34.
- Rijn, Pol van, **Silvan Mertes**, Dominik Schiller, Piotr Dura, Hubert Siuzdak, Peter Harrison, Elisabeth André, and Nori Jacoby (2022). "VoiceMe: Personalized voice generation in TTS." In: *Proc. Interspeech 2022*.
- Ritschel, Hannes, Ilhan Aslan, **Silvan Mertes**, Andreas Seiderer, and Elisabeth André (2019). "Personalized synthesis of intentional and emotional non-verbal sounds for social robots." In: *2019 8th International conference on affective computing and intelligent interaction (ACII)*. IEEE, pp. 1–7.
- Ritschel, Hannes, **Silvan Mertes**, Florian Lingenfelser, Thomas Kiderle, and Elisabeth André (2023). "The Affective Bar Piano." In: *Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents*, pp. 1–3.
- Schiller, Dominik, **Silvan Mertes**, Marcel Achzet, Fabio Hellmann, Ruben Schlagowski, and Elisabeth André (2024). "More Than Noise: Assessing the Viscosity of Food Products Based on Sound Emission." In: *International Conference on Deep Learning Theory and Applications*. Springer, pp. 210–218.
- Schiller, Dominik, **Silvan Mertes**, Pol van Rijn, and Elisabeth André (2022). "Bridging the Gap: End-to-End Domain Adaptation for Emotional Vocalization Classification using Adversarial Learning." In: *Proceedings of the 3rd International on Multimodal Sentiment Analysis Workshop and Challenge*, pp. 95–100.

- Schiller, Dominik, **Silvan Mertes**, Pol van Rijn, and Elisabeth André (2021). "Analysis by synthesis: Using an expressive tts model as feature extractor for paralinguistic speech classification." In: *Proc. Interspeech 2021*.
- Schlagowski, Ruben, Kunal Gupta, **Silvan Mertes**, Mark Billinghurst, Susanne Metzner, and Elisabeth André (2022). "Jamming in MR: towards real-time music collaboration in mixed reality." In: *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, pp. 854–855.
- Schlagowski, Ruben, Dariia Nazarenko, Yekta Can, Kunal Gupta, **Silvan Mertes**, Mark Billinghurst, and Elisabeth André (2023). "Wish you were here: Mental and physiological effects of remote music collaboration in mixed reality." In: *Proceedings of the 2023 CHI conference on human factors in computing systems*, pp. 1–16.
- Schlagowski, Ruben, **Silvan Mertes**, and Elisabeth André (2021). "Taming the chaos: exploring graphical input vector manipulation user interfaces for GANs in a musical context." In: *Proceedings of the 16th International Audio Mostly Conference*, pp. 216–223.
- Schlagowski, Ruben, **Silvan Mertes**, Dariia Nazarenko, Alexander Dauber, and Elisabeth André (2024). "XR composition in the wild: the impact of user environments on creativity, UX and flow during music production in augmented reality." In: *Proceedings of the 19th International Audio Mostly Conference: Explorations in Sonic Cultures*.
- Schlagowski, Ruben, **Silvan Mertes**, Dominik Schiller, Yekta Said Can, and Elisabeth André (2024). "Exploring Physiology-Based Classification of Flow During Musical Improvisation in Mixed Reality." In: *International Conference on Deep Learning Theory and Applications*. Springer, pp. 296–309.
- Schlagowski, Ruben, Maurizio Volanti, Katharina Weitz, **Silvan Mertes**, Johanna Kuch, and Elisabeth André (2024). "The feeling of being classified: raising empathy and awareness for AI bias through perspective-taking in VR." In: *Frontiers in Virtual Reality* 5, p. 1340250.
- Schlagowski, Ruben, Fabian Wildgrube, **Silvan Mertes**, Ceenu George, and Elisabeth André (2022). "Flow with the beat! Human-centered design of virtual environments for musical creativity support in VR." In: *Proceedings of the 14th Conference on Creativity and Cognition*, pp. 428–442.
- **Silvan Mertes**, Thomas Kiderle, Ruben Schlagowski, Florian Lingenfelser, and Elisabeth Andre (2021). "On the potential of modular voice conversion for virtual agents." In: *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE, pp. 1–7.
- **Silvan Mertes**, Dominik Schiller, Michael Dietz, Elisabeth André, and Florian Lingenfelser (2024). "The AffectToolbox: Affect Anal-

ysis for Everyone." In: *2024 12th International conference on affective computing and intelligent interaction (ACII)*. IEEE.

- Triantafyllopoulos, Andreas, Björn W Schuller, Gökçe İymen, Metin Sezgin, Xiangheng He, Zijiang Yang, Panagiotis Tzirakis, Shuo Liu, **Silvan Mertes**, Elisabeth André, et al. (2023). "An overview of affective speech synthesis and conversion in the deep learning era." In: *Proceedings of the IEEE* 111.10, pp. 1355–1381.
- Van Rijn, Pol, **Silvan Mertes**, Dominik Schiller, Peter Harrison, Pauline Larrouy-Maestri, Elisabeth André, and Nori Jacoby (2021). "Exploring emotional prototypes in a high dimensional TTS latent space." In: *Proc. Interspeech 2021*.
- Withanage Don, Daksitha, Thomas Kiderle, **Silvan Mertes**, Dominik Schiller, Hannes Ritschel, and Elisabeth André (2025). "MeLaX: Conversations with Generative AI in Socially Interactive Agents." In: *Companion Proceedings of the 30th International Conference on Intelligent User Interfaces*, pp. 163–166.
- Withanage Don, Daksitha, Dominik Schiller, Mitho Müller, Tobias Hallmen, **Silvan Mertes**, Tobias Baur, Florian Lingenfelser, Lea Kaubisch, Corinna Reck, and Elisabeth André (2024). "Towards Automated Annotation of Infant-Caregiver Engagement Phases with Multimodal Foundation Models." In: *Proceedings of the 26th International Conference on Multimodal Interaction*.

B

TEACHING

B.1 LECTURES & PRACTICAL COURSES

- Winter Semester 2024/25
  - ⋆ Lecturer *Generative AI for Human-Computer Interaction* (Master)
  - ⋆ Lecturer *Praktikum Spieleprogrammierung* (Master)
  - ⋆ Lecturer *Seminar Grundlagen der Generativen Künstlichen Intelligenz* (Bachelor)
  - ⋆ Lecturer *Seminar Generative Künstliche Intelligenz* (Master)
- Summer Semester 2024
  - ⋆ Lecturer *Einführung in die Spieleprogrammierung* (Master)
  - ⋆ Lecturer *Seminar Grundlagen der Generativen Künstlichen Intelligenz* (Bachelor)
  - ⋆ Lecturer *Seminar Generative Künstliche Intelligenz* (Master)
  - ⋆ Guest Lecturer *Grundlagen der Human-Computer Interaktion* (Bachelor)
- Winter Semester 2023/24
  - ⋆ Lecturer *Seminar Generative Künstliche Intelligenz* (Master)
- Summer Semester 2023
  - ⋆ Lecturer *Seminar Generative Künstliche Intelligenz* (Master)
- Winter Semester 2022/23
  - ⋆ Lecturer *Praktikum Interactive Machine Learning* (Master)
- Winter Semester 2020/21
  - ⋆ Guest Lecturer *Interaction Design and Engineering* (Bachelor)
- Summer Semester 2020
  - ⋆ Lecturer *Human-Computer Interaction* (Master)

B.2 SUPERVISED BACHELOR'S THESES

- Conditional Human Image Synthesis with Generative Adversarial Networks (2020)
- Implementation of a Classification Model for Rhythmic Attunement in Music Therapy Sessions (2022, Co-Supervision)
- Exploring Tangible User Interfaces for Latent Space Manipulation of Generative Adversarial networks (2022, Co-Supervision)
- Generating Audio Triggers for an Autonomous Sensory Meridian Response with Generative Adversarial Networks (2023)
- GradCam zur Analyse von GAN-Trainingsprozessen (2024)
- Computer-assisted Feedback for Javelin Throw (2024, Co-Supervision)
- Entwicklung eines interaktiven, durch maschinelles Lernen gestützten Trainingssystems für extreme Gesangstechniken (2024, Co-Supervision)
- Konzeption und Implementierung einer nutzerfreundlichen grafischen Oberfläche für multimodale Emotionserkennung (2024, Co-Supervision)
- Gezielte Manipulation von Umgebung und Darstellung virtueller Charaktere in Bildern durch Diffusion Models (2024)
- Automatische Kolorierung von Mangas mithilfe von Deep Learning (2024)
- Automatische Generierung von Soundkulissen mit Hilfe von Deep Learning (2024)
- Generating Personalized Counterfactual Feedback for Javelin Throw Technique Improvement (2024, Co-Supervision)

B.3 SUPERVISED MASTER'S THESES

- Konträre Chatbotpersonas im internen Businessumfeld: Entwicklung und Präferenzanalyse (2021)
- Reinforcement Learning Techniques as Enhancement of frame-level Speech Emotion Recognition (2021, Co-Supervision)
- Exploring Opportunities for Musical Creativity Support in VR through Human-Computer-Interfaces and Interaction Design (2021, Co-Supervision)
- Alterfactuals as a Novel Explanation Method for Image Classifiers (2021)

- Generating Counterfactual Explanations for Atari Agents via Generative Adversarial Networks (2022, Co-Supervision)
- Using GANs for Combining Counterfactual Explanations and Feature Attribution (2023)
- Computational Generation and Adaption of Climbing Routes through Adversarial Learning (2023, Co-Supervision)
- Using CycleGAN to Learn Image-to-Image Translation for Unpaired Facial Expression Data (2023, Co-Supervision)
- Dynamische Texturgenerierung von Videospielen mit Diffusion Models (2023)

B.4 supervised project modules

- Evaluating GAN-based Alterfactual Explanation Generation (2023)
- Diffusion-based Counterfactual Explanation Generation for Facial Emotion Recognition (2023)
- Texture Editing with Diffusion Models (2024)

C

ACADEMIC ACTIVITIES

C.1 PEER REVIEWS

I have served as a peer reviewer for the following venues:

- Transactions on Affective Computing
- ACM Conference on Human Factors in Computing Systems (CHI)
- IEEE Signal Processing Magazine
- International Conference on Autonomous Agents and Multiagent Systems (AAMAS)
- ACM Conference on Intelligent User Interfaces (IUI)
- International Conference on Multimodal Interaction (ICMI)
- Transactions on Audio, Speech and Language Processing
- Applied Artificial Intelligence
- XAI2023 (XAI@IJCAI)
- European Conference on Artificial Intelligence (ECAI)
- IEEE Robotics and Automation Letters
- Elsevier Expert Systems with Applications
- International Conference on Affective Computing & Intelligent Interaction (ACII)
- Audio Mostly Conference
- PeerJ Computer Science

C.2 ROLES

During the past years, I was given the opportunity to fulfill the following roles:

- Scientific Coordinator *Human-Centered Production Technologies* at the AI Production Network at University of Augsburg
- Organizing Committee member (Proceedings Chair) ACM International Conference on Intelligent Virtual Agents (IVA) 2025
- Program Committee member ACM Conference on Intelligent User Interfaces (IUI) 2025
- Program Committee member International Conference on Autonomous Agents and Multiagent Systems (AAMAS) 2025
- Scientific Committee member Audiomostly 2024
- Organizing Committee member Interdisciplinary Tutorshop on Interactions with Embodied Virtual Agents at IVA 2024
- Session Chair 5th International Conference on Deep Learning Theory and Applications (DeLTA'24)
- Program Committee member Trustworthy Sequential Decicion-making and Optimization Workshop at ECAI 2024
- Program Committee member International Conference on Affective Computing & Intelligent Interaction (ACII) 2024
- Program Committee member Workshop on Explainable Artificial Intelligence at IJCAI 2023
- Session Chair 2nd International Conference on Deep Learning Theory and Applications (DeLTA'21)
- Program Committee member International Conference on Multimodal Interaction (ICMI) 2021-2023

C.3 AWARDS

I received the following awards:

- Best Paper Award at Conference on Deep Learning Theories and Applications (DeLTA 2020)
- Honorable Mention at IEEE Virtual Reality (IEEEVR 2022)
- Honorable Mention at Creativity & Cognition (C&C 2022)
- 1st Place at the A-VB Challenge held at the International Conference on Affective Computing & Intelligent Interaction (ACII2022)
- 2nd Place at the ComParE Challenge (Subchallenge *Escalation Detection*) at Interspeech 2021
- 4th Place at the STOIC2021 COVID-19 AI Challenge
- Best Poster Award at the International Conference on Deep Learning Theories and Applications (DeLTA 2024)

D

TECHNICAL & EXPERIMENTAL DETAILS

D.1 COUNTERFACTUAL EXPLANATION GENERATION

Table 12 shows the classifier architecture that we used in Chapter 7.

Layer	Description	Number of Filters	Size	Stride	Dropout Probability
1	Conv2D	96	11 x 11	4	-
2	MaxPooling2D	-	2 x 2	2	-
3	Batch Normalization	-	-	-	-
4	Conv2D	256	11 x 11	1	-
5	MaxPooling2D	-	2 x 2	2	-
6	Batch Normalization	-	-	-	-
7	Conv2D	384	3 x 3	1	-
8	Batch Normalization	-	-	-	-
9	Conv2D	384	3 x 3	1	-
10	Batch Normalization	-	-	-	-
11	Conv2D	256	3 x 3	1	-
12	MaxPooling2D	-	2 x 2	2	-
13	Batch Normalization	-	-	-	-
14	Flatten	-	-	-	-
15	Dense	-	4096	-	-
16	Dropout	-	-	-	0.4
17	Batch Normalization	-	-	-	-
18	Dense	-	4096	-	-
19	Dropout	-	-	-	0.4
20	Batch Normalization	-	-	-	-
21	Dense	-	1000	-	-
22	Dropout	-	-	-	0.4
23	Batch Normalization	-	-	-	-
24	Dense	-	2	-	-

Table 12: L2 bias and kernel regularization with a regularization factor of 0.001 was applied to all convolutional and dense layers except layer 25.

Layer	Description	# Filter	Size	Stride	Dropout	BatchNorm	Activation
1	Conv2D	64	4x4	2	-	no	LeakyReLU (0.2
2	Conv2D	128	4x4	2	-	yes	LeakyReLU (0.2
3	Conv2D	256	4x4	2	-	yes	LeakyReLU (0.2
4	Conv2D	512	4x4	2	-	yes	LeakyReLU (0.2
5	Conv2D	512	4x4	2	-	yes	LeakyReLU (0.2
6	Conv2D	512	4x4	2	-	yes	LeakyReLU (0.2
7	Conv2D	512	4x4	2	-	no	ReLU
8	Conv2DTranspose	512	4x4	2	0.5	yes	ReLU
9	Conv2DTranspose	512	4x4	2	0.5	yes	ReLU
10	Conv2DTranspose	512	4x4	2	0.5	yes	ReLU
11	Conv2DTranspose	256	4x4	2	-	yes	ReLU
12	Conv2DTranspose	128	4x4	2	-	yes	ReLU
13	Conv2DTranspose	64	4x4	2	-	yes	ReLU
14	Conv2DTranspose	1	4x4	2	-	no	Tanh

Table 13: Generator Architecture used in our evaluation scenario. The architecture is adapted from Y. Wu et al. (2019). Where BatchNorm, Dropout, or Activation function occurred together, the order applied was BatchNorm - Dropout - Activation.

D.2 ALTERFACTUAL EXPLANATION GENERATION

D.2.1 *GAN Architecture and Training*

D.2.1.1 *Generator Model*

The GAN's generator architecture is listed in Table 13.

D.2.1.2 *Discriminator Model*

The GAN's discriminator architecture is listed in Table 14.

D.2.1.3 *Training Configuration and Hyperparameters*

The training configuration and hyperparamters are shown in Table 15. The Adam optimizer was configured with $\beta_1 = 0.5$, $\beta_2 = 0.999$, $\epsilon = 1e-8$.

Further, the Support Vector Machine (SVM) that was included in the loss function (see main paper) was trained with the parameters listed in Table 16.

D.2.2 *Classifier Architecture and Training*

In Table 17, the model architecture for the classifier that we used in our evaluation scenario is described. The training configuration

Layer	Description	# Filter	Size	Stride	BatchNorm	Activation
0a	Embedding	-	8x8	-	no	-
0b	Upsample	-	128x128	-	no	-
1	Conv2D	64	4x4	2	no	LeakyReLU (0.2)
2	Conv2D	128	4x4	2	yes	LeakyReLU (0.2)
3	Conv2D	256	4x4	2	yes	LeakyReLU (0.2)
4	Conv2D	1	4x4	2	no	Sigmoid

Table 14: Discriminator Architecture used in our evaluation scenario. Where BatchNorm and Activation function occurred together, BatchNorm preceded the activation function. The first two layers, marked as '0a' and '0b' were used to upsample the label information to the size of the input image. The label and image were passed together to layer 1. The architecture is adapted from Y. Wu et al. (2019).

Batch Size	1
Epochs	14
Learning Rate Generator	1e-4
Learning Rate Discriminator	1e-4
Optimizer	Adam

Table 15: The setting used to train the GAN.

C (Regularisation)	10
Kernel	linear
Iterations	5000

Table 16: The setting used to train the SVM.

Layer	Description	# Filter	Size	Stride	BatchNorm	Activation
1	Conv2D	32	3x3	1	yes	ReLU
2	Conv2D	32	3x3	1	yes	ReLU
3	MaxPool2D	-	2x2	2	no	-
4	Conv2D	64	3x3	1	yes	ReLU
5	Conv2D	64	3x3	1	yes	ReLU
6	GAP	-	-	-	no	-
7	Dense	-	2	-	no	Softmax

Table 17: Classifier architecture used to train the classifier for the MNIST-Fashion dataset (classes *Sneaker* and *Ankle Boot*). Where BatchNorm and Activation function occurred together, BatchNorm preceded the activation function.

Batch Size	32
Epochs	40
Learning Rate	1e-3
Optimizer	Adam
Loss Function	Binary Cross Entropy

Table 18: The setting used to train the Fashion-MNIST classifier.

and hyperparamters are shown in Table 18. The Adam optimizer was configured with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$.

D.2.3 *Additional Dataset Experiments*

In order to demonstrate that our alterfactual generation approach is generalizable to different datasets, we additionally trained models for three other datasets. Here, we omitted the Feature Relevance component. As for that component an additional SVM has to be trained on the penultimate layer of the classifier layer, it takes away the model-agnostic property from the alterfactual generation network. By performing these additional experiments, we show that the approach can simply be adapted to be model-agnostic, although that may negatively affect the outcomes of the results - it is not specifically forced that *only* irrelevant features change. For the classifiers, we used the same architecture as for the Fashion-MNIST dataset, although batch size and epochs were modified to fit the hardware that we used.

D.2.3.1 *MNIST*

As the MNIST datasets has more than two classes (each class contains hand-drawn images of one specific digit), we picked the two digits that are most likely to be confused by deep learning classifiers: *Three* and *Eight*. The MNIST classifier was trained for 9 epochs with batch size 32. Besides not using the Feature Relevance component and increasing the epoch number to 42, the GAN network was trained with the same parameter settings as for the Fashion-MNIST dataset. We reached a validity of 95.92% and an average SSIM of 0.425. Example outputs are shown in Figure 82.

D.2.3.2 *MaskedFace-Net*

The MaskedFace-Net dataset contains images of people wearing face masks. Binary labels are provided, indicating that on the respective image the mask is worn correctly or incorrectly. The classifier was trained for 2 epochs with batch size 128. Besides not using the Feature Relevance component and decreasing the epoch number to 11, the GAN network was trained with the same parameter settings as for

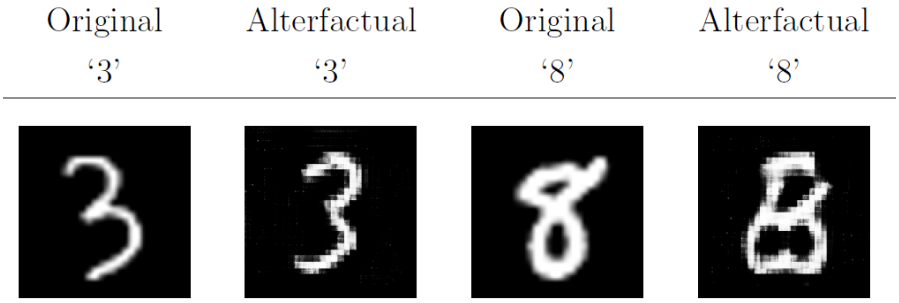


Figure 82: Examplary alterfactual outputs for the MNIST dataset.

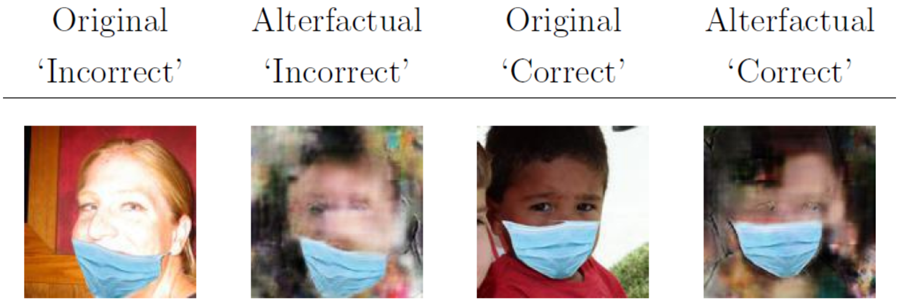


Figure 83: Examplary alterfactual outputs for the MaskedFace-Net dataset.

the Fashion-MNIST dataset. We reached a validity of 84.27% and an average SSIM of 0.091. Example outputs are shown in Figure 83.

D.2.3.3 *MaskedFace-Net (Gray Scale)*

Here, we also used the MaskedFace-Net dataset, but converted it to gray scale, demonstrating that our approach also works with gray scale data. The classifier was trained for 1 epoch with batch size 128. Besides not using the Feature Relevance component and decreasing the epoch number to 6, the GAN network was trained with the same parameter settings as for the Fashion-MNIST dataset. We reached a validity of 48.89% and an average SSIM of 0.002. Example outputs are shown in Figure 84.

D.2.4 *User Study*

D.2.4.1 *Demographic Details*

For the AI experience and Attitude we adapted a description of AI from B. Zhang and Dafoe (2019) and S. Russell and Norvig (2016) to "The following questions ask about Artificial Intelligence (AI). Colloquially, the term 'artificial intelligence' is often used to describe machines (or computers) that mimic 'cognitive' functions that humans associate with the human mind, such as 'learning' and 'problem solv-

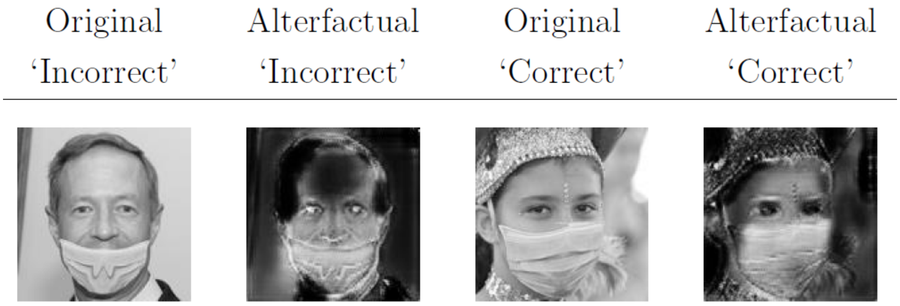


Figure 84: Examplary alterfactual outputs for the gray scale version of the MaskedFace-Net dataset. It can be clearly seen that the only part of the image that gets unchanged is the mask itself - indicating that everything else is irrelevant.

ing'." After this description, participants had to select one or more item describing their experience with AI. The distribution of the items for each condition is shown in Fig. 85. Following this we adapted a question from B. Zhang and Dafoe (2019) to measure the participants' attitude towards AI. We asked them to rate their answer to the question "Suppose that AI agents would achieve high-level performance in more areas one day. How positive or negative do you expect the overall impact of such AI agents to be on humanity in the long run?" on a 5-point Likert scale from "Extremely negative" to "Extremely positive". The participants also had the option to answer "I do not know" here, which would exclude them from the evaluation of this question.

D.2.4.2 *Additional Post-Hoc Results*

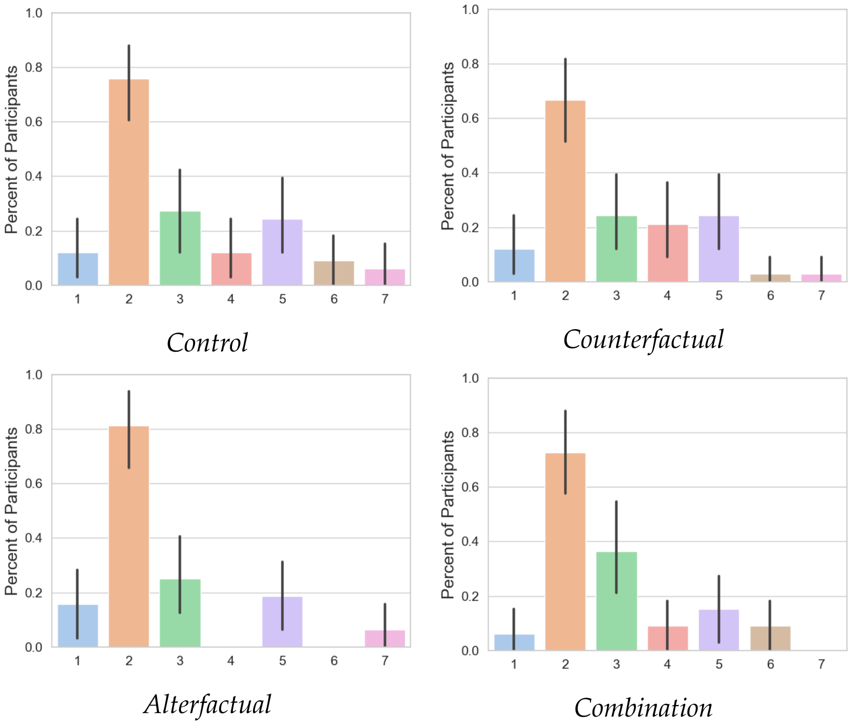
For completeness, we also report the results of the post-hoc t-tests on the participants' prediction accuracy that were not significant. The effect size d is calculated according to Cohen (2013):

- ***Counterfactual* vs. *Control***: $t(127) = 1.14$, $p = .258$, $d = 0.28$
- ***Combination* vs. *Alterfactual***: $t(127) = 0.71$, $p < .478$, $d = 0.18$.

For feature understanding and explanation satisfaction we did not calculate post-hoc tests since the ANOVA was not significant.

D.2.4.3 *Explanation Satisfaction Scale*

For evaluating explanation satisfaction, we used the Explanation Satisfaction scale by Hoffman (Hoffman et al., 2018) except one item that did not apply to our use case. The items that we used were as follows, where each item was rated on a 5-point likert scale (1 = strongly disagree, 5 = strongly agree):



Control

Counterfactual

Alterfactual

Combination

Figure 85: Distribution of the chosen AI experience items for each condition. The x-axis depicts the following items: 1 - I do not have any experience in AI related topics; 2 - I know AI from the media; 3 - I use AI technology in my private life; 4 - I use AI technology in my work; 5 - I have taken at least one AI related course; 6 - I do research on AI-related topics; 7 - Other:

- From the explanations, I **understand** how the AI makes its decision.
- The explanations of how the AI makes its decision are **satisfying**.
- The explanations of how the AI makes its decision have **sufficient detail**.
- The explanations of how the AI makes its decision seem **complete**.
- The explanations of how the AI makes its decision are **useful** to predict the AI's decision.
- The explanations of how the AI makes its decision show me how **accurate** the AI is.
- The explanations let me judge when I should **trust and not trust** the AI.

BIBLIOGRAPHY

Abayomi-Alli, Olusola O, Robertas Damaševičius, Atika Qazi, Mariam Adedoyin-Olowe, and Sanjay Misra (2022). "Data augmentation and deep learning methods in sound classification: A systematic review." In: *Electronics* 11.22, p. 3795.

Abdal, Rameen, Yipeng Qin, and Peter Wonka (2019). "Image2stylegan: How to embed images into the stylegan latent space?" In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4432–4441.

Alqaraawi, Ahmed, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze (2020). "Evaluating saliency map explanations for convolutional neural networks: a user study." In: *IUI '20: 25th International Conference on Intelligent User Interfaces, Cagliari, Italy, March 17-20, 2020*, pp. 275–285. URL: https://doi.org/10.1145/3377325.3377519.

Amiriparian, Shahin, Maurice Gerczuk, Sandra Ottl, Nicholas Cummins, Michael Freitag, Sergey Pugachevskiy, Alice Baird, and Björn Schuller (2017). "Snore sound classification using image-based deep spectrum features." In.

Anderson, Andrew, Jonathan Dodge, Amrita Sadarangani, Zoe Juozapaitis, Evan Newman, Jed Irvine, Souti Chattopadhyay, Alan Fern, and Margaret Burnett (July 2019). "Explaining Reinforcement Learning to Mere Mortals: An Empirical Study." In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pp. 1328–1334. DOI: 10.24963/ijcai.2019/184. URL: https://doi.org/10.24963/ijcai.2019/184.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). "Wasserstein generative adversarial networks." In: *International conference on machine learning*. PMLR, pp. 214–223.

Arrieta, Alejandro Barredo, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. (2020). "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." In: *Information Fusion* 58, pp. 82–115.

Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek (July 2015). "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation." In: *PLOS ONE* 10.7. Ed. by Oscar Deniz Suarez. ISSN: 1932-6203. (Visited on 12/18/2018).

Baird, Alice, Shahin Amiriparian, and Björn Schuller (2019). "Can deep generative audio be emotional? Towards an approach for personalised emotional audio generation." In: *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, pp. 1–5.

Baltrusaitis, Tadas, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency (2018). "OpenFace 2.0: Facial Behavior Analysis Toolkit." In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pp. 59–66. DOI: 10.1109/FG.2018.00019.

Barocas, Solon, Andrew D Selbst, and Manish Raghavan (2020). "The hidden assumptions behind counterfactual explanations and principal reasons." In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 80–89.

Barratt, Emma L and Nick J Davis (2015). "Autonomous Sensory Meridian Response (ASMR): a flow-like mental state." In: *PeerJ* 3, e851.

Barratt, Emma L, Charles Spence, and Nick J Davis (2017). "Sensory determinants of the autonomous sensory meridian response (ASMR): understanding the triggers." In: *PeerJ* 5, e3846.

Baur, Tobias, Ionut Damian, Patrick Gebhard, Kaska Porayska-Pomsta, and Elisabeth Andre (2013). "A Job Interview Simulation: Social Cue-Based Interaction with a Virtual Character." In.

Baur, Tobias, Gregor Mehlmann, Ionut Damian, Florian Lingenfelser, Johannes Wagner, Birgit Lugrin, Elisabeth André, and Patrick Gebhard (June 2015). "Context-Aware Automated Analysis and Annotation of Social Human–Agent Interactions." In: *ACM Trans. Interact. Intell. Syst.* 5.2. ISSN: 2160-6455.

Beaupré, MG, N Cheung, and U Hess (2000). "The Montreal set of facial displays of emotion." In: *Montreal, Quebec, Canada*.

Bednarik, Roman, Shahram Eivazi, and Michal Hradis (2012). "Gaze and Conversational Engagement in Multiparty Video Conversation: An Annotation Scheme and Classification of High and Low Levels of Engagement." In: *Proceedings of the 4th Workshop on Eye Gaze in Intelligent Human Machine Interaction*. Gaze-In '12. Santa Monica, California: ACM, 10:1–10:6. ISBN: 978-1-4503-1516-6.

Bejiga, Mesay Belete and Farid Melgani (2018). "Gan-based domain adaptation for object classification." In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, pp. 1264–1267.

Bernacki, Matthew L, Timothy J Nokes-Malach, and Vincent Aleven (2015). "Examining self-efficacy during learning: variability and relations to behavior, performance, and learning." In: *Metacognition and Learning* 10.1, pp. 99–117.

Berndt, Axel, Nadia Al-Kassab, and Raimund Dachselt (2015). "TouchNoise: A New Multitouch Interface for Creative Work with

Noise." In: *Proceedings of the Audio Mostly 2015 on Interaction With Sound*, pp. 1–8.

Beyer, Hans-Georg and Hans-Paul Schwefel (2002). "Evolution strategies–A comprehensive introduction." In: *Natural computing* 1.1, pp. 3–52.

Bontrager, Philip, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross (2018). "Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution." In: *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, pp. 1–9.

Bowles, Christopher, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger Gunn, Alexander Hammers, David Alexander Dickie, Maria Valdés Hernández, Joanna Wardlaw, and Daniel Rueckert (2018). "Gan augmentation: Augmenting training data using generative adversarial networks." In: *arXiv preprint arXiv:1810.10863*.

Braun, Virginia and Victoria Clarke (2012). *Thematic analysis.* American Psychological Association.

Byrne, Ruth M. J. (July 2019). "Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning." In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pp. 6276–6282. DOI: 10.24963/ijcai.2019/876. URL: https://doi.org/10.24963/ijcai.2019/876.

Cafaro, Angelo, Johannes Wagner, Tobias Baur, Soumia Dermouche, Mercedes Torres Torres, Catherine Pelachaud, Elisabeth André, and Michel Valstar (Nov. 2017). "The NoXi Database: Multimodal Recordings of Mediated Novice-Expert Interactions." In: *ICMI'17*.

Cao, Z., G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh (2019). "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Cavigelli, Lukas, Pascal Hager, and Luca Benini (May 2017). "CAS-CNN: A deep convolutional neural network for image compression artifact suppression." In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 752–759. DOI: 10.1109/ijcnn.2017.7965927.

Chandna, Pritish, Merlijn Blaauw, Jordi Bonada, and Emilia Gómez (2019). "Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan." In: *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 1–5.

Chen, Daniel L, Martin Schonger, and Chris Wickens (2016). "oTree—An open-source platform for laboratory, online, and field experiments." In: *Journal of Behavioral and Experimental Finance* 9, pp. 88–97.

Chen, Xi, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel (2016). "Infogan: Interpretable representation learning by information maximizing generative adversarial nets." In: *Advances in neural information processing systems* 29.

Choi, Jaehoon, Taekyung Kim, and Changick Kim (2019). "Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6830–6840.

Choi, Yunjey, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo (2018). "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8789–8797.

Chou, Yu-Liang, Catarina Moreira, Peter Bruza, Chun Ouyang, and Joaquim Jorge (2022). "Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications." In: *Information Fusion* 81, pp. 59–83.

Chou, Yung-Chien, Cheng-Ju Kuo, Tzu-Ting Chen, Gwo-Jiun Horng, Mao-Yuan Pai, Mu-En Wu, Yu-Chuan Lin, Min-Hsiung Hung, Wei-Tsung Su, Yi-Chung Chen, et al. (2019). "Deep-learning-based defective bean inspection with GAN-structured automated labeled data augmentation in coffee industry." In: *Applied Sciences* 9.19, p. 4166.

Cohen, Jacob (2013). *Statistical power analysis for the behavioral sciences*. Academic press.

Conati, Cristina, Oswald Barral, Vanessa Putnam, and Lea Rieger (2021). "Toward personalized XAI: A case study in intelligent tutoring systems." In: *Artificial Intelligence* 298, p. 103503. DOI: 10.1016/j.artint.2021.103503.

Csikszentmihalyi, Mihaly (2000). *Beyond boredom and anxiety*. Jossey-Bass.

D'Mello, Sidney S, Patrick Chipman, and Art Graesser (2007). "Posture as a predictor of learner's affective engagement." In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 29.

Dael, Nele, Marcello Mortillaro, and Klaus R. Scherer (June 2012). "The Body Action and Posture Coding System (BAP): Development and Reliability." In: *Journal of Nonverbal Behavior* 36.2, pp. 97–121. ISSN: 1573-3653.

Dahlstedt, Palle (2001). "Creating and exploring huge parameter spaces: Interactive evolution as a tool for sound generation." In: *ICMC*. Citeseer.

Das, Arun and Paul Rad (2020). "Opportunities and challenges in explainable artificial intelligence (xai): A survey." In: *arXiv preprint arXiv:2006.11371*.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database." In:

2009 IEEE conference on computer vision and pattern recognition. Ieee, pp. 248–255.

Di Mattia, Federico, Paolo Galeone, Michele De Simoni, and Emanuele Ghelfi (2019). "A survey on gans for anomaly detection." In: *arXiv preprint arXiv:1906.11632*. eprint: 1906.11632.

Diethei, Daniel, Jasmin Niess, Carolin Stellmacher, Evropi Stefanidi, and Johannes Schöning (2021). "Sharing heartbeats: motivations of citizen scientists in times of crises." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15.

Ding, Hui, Kumar Sricharan, and Rama Chellappa (2018). "Exprgan: Facial expression editing with controllable expression intensity." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Donahue, Chris, Julian McAuley, and Miller Puckette (2018). "Adversarial audio synthesis." In: *arXiv preprint arXiv:1802.04208*.

Dong, Hao-Wen, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang (2018). "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Eastwood, Cian and Christopher KI Williams (2018). "A framework for the quantitative evaluation of disentangled representations." In: *6th International Conference on Learning Representations*.

Ebner, Natalie C, Michaela Riediger, and Ulman Lindenberger (2010). "FACES—A database of facial expressions in young, middle-aged, and older women and men: Development and validation." In: *Behavior research methods* 42.1, pp. 351–362.

Elsayed, Gamaleldin, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio (2018). "Large margin deep networks for classification." In: *Advances in neural information processing systems* 31.

Engel, Jesse, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts (2019). "Gansynth: Adversarial neural audio synthesis." In: *arXiv preprint arXiv:1902.08710*.

Engelbregt, HJ, Kim Brinkman, CCE Van Geest, Mona Irrmischer, and Jan Berend Deijen (2022). "The effects of autonomous sensory meridian response (ASMR) on mood, attention, heart rate, skin conductance and EEG in healthy young adults." In: *Experimental Brain Research* 240.6, pp. 1727–1742.

Fan, Jianyu, Miles Thorogood, and Philippe Pasquier (2017). "Emosoundscapes: A dataset for soundscape emotion recognition." In: *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, pp. 196–201.

Fang, Zexin, Bin Han, C Clark Cao, and Hans D Schotten (2023). "Artificial ASMR: A Cyber-Psychological Approach." In: *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, pp. 1–6.

Ferguson, Max K, Ak Ronay, Yung-Tsun Tina Lee, and Kincho H Law (2018). "Detection and Segmentation of Manufacturing Defects with Convolutional Neural Networks and Transfer Learning." In: *Smart and sustainable manufacturing systems* 2.

Franquet, Tomás (2018). "Imaging of community-acquired pneumonia." In: *Journal of thoracic imaging* 33.5, pp. 282–294.

Frid-Adar, Maayan, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan (2018). "Synthetic data augmentation using GAN for improved liver lesion classification." In: *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, pp. 289–293.

Gasperin, Daniela, Gopalakrishnan Netuveli, Juvenal Soares Dias-da-Costa, and Marcos Pascoal Pattussi (2009). "Effect of psychological stress on blood pressure increase: a meta-analysis of cohort studies." In: *Cadernos de saude publica* 25.4, pp. 715–726.

Gatys, Leon A (2015). "A neural algorithm of artistic style." In: *arXiv preprint arXiv:1508.06576*.

Gauthier, Jon (2014). "Conditional generative adversarial nets for convolutional face generation." In: *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester* 2014.5, p. 2.

Gebhard, P., T. Schneeberger, E. André, T. Baur, I. Damian, G. Mehlmann, C. König, and M. Langer (2019). "Serious Games for Training Social Skills in Job Interviews." In: *IEEE Transactions on Games* 11.4, pp. 340–351.

Geinitz, Steffen, Andreas Margraf, André Wedel, Sebastian Witthus, and Klaus Drechsler (2016). "Detection of filament misalignment in carbon fiber production using a stereovision line scan camera system." In: *Proc. of 19th World Conference on Non-Destructive Testing*.

Geinitz, Steffen, André Wedel, and Andreas Margraf (2016). "Online Detection and Categorisation of Defects along Carbon Fiber Production using a High Resolution, High Width Line Scan Vision System." In: *Proceedings of the 17th European Conference on Composite Materials ECCM17, Munich*. European Society for Composite Materials.

Giacomello, Edoardo, Pier Luca Lanzi, and Daniele Loiacono (2019). "Searching the Latent Space of a Generative Adversarial Network to Generate DOOM Levels." In: *2019 IEEE Conference on Games (CoG)*. IEEE, pp. 1–8.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial nets." In: *Advances in neural information processing systems*, pp. 2672–2680.

Goyal, Yash, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). "Counterfactual visual explanations." In: *arXiv preprint arXiv:1904.07451*.

Guidotti, Riccardo, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini (2019). "Factual and counterfactual explanations for black box decision making." In: *IEEE Intelligent Systems* 34.6, pp. 14–23.

Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville (2017). "Improved training of wasserstein gans." In: *Advances in neural information processing systems* 30.

Halkos, George and Dimitrios Bousinakis (2010). "The effect of stress and satisfaction on productivity." In: *International Journal of Productivity and Performance Management* 59.5, pp. 415–431.

Härkönen, Erik, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris (2020). "Ganspace: Discovering interpretable gan controls." In: *arXiv preprint arXiv:2004.02546*.

Harmon-Jones, Cindy, Brock Bastian, and Eddie Harmon-Jones (2016). "The discrete emotions questionnaire: A new tool for measuring state self-reported emotions." In: *PloS one* 11.8, e0159915.

Haselmann, Matthias and Dieter Gruber (2017). "Supervised machine learning based surface inspection by synthetizing artificial defects." In: *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, pp. 390–395.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, Zhenliang, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen (2019). "Attgan: Facial attribute editing by only changing what you want." In: *IEEE Transactions on Image Processing* 28.11, pp. 5464–5478.

Heimerl, A., K. Weitz, T. Baur, and E. Andre (2020). "Unraveling ML Models of Emotion with NOVA: Multi-Level Explainable AI for Non-Experts." In: *IEEE Transactions on Affective Computing* 01, pp. 1–1. ISSN: 1949-3045. DOI: 10.1109/TAFFC.2020.3043603.

Heimerl, Alexander, Silvan Mertes, Tanja Schneeberger, Tobias Baur, Ailin Liu, Linda Becker, Nicolas Rohleder, Patrick Gebhard, and Elisabeth André (2022a). "" GAN I hire you?"–A System for Personalized Virtual Job Interview Training." In: *arXiv preprint arXiv:2206.03869*.

Heimerl, Alexander, Silvan Mertes, Tanja Schneeberger, Tobias Baur, Ailin Liu, Linda Becker, Nicolas Rohleder, Patrick Gebhard, and Elisabeth André (2022b). "Generating personalized behavioral feedback for a virtual job interview training system through adversarial learning." In: *International Conference on Artificial Intelligence in Education*. Springer, pp. 679–684.

Herchenbach, Marvin, Dennis Müller, Stephan Scheele, and Ute Schmid (2022). "Explaining image classifications with near misses, near hits and prototypes: Supporting domain experts in understanding decision boundaries." In: *International Conference on Pattern Recognition and Artificial Intelligence*. Springer, pp. 419–430.

Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (2017). "Gans trained by a two time-scale update rule converge to a local nash equilibrium." In: *Advances in neural information processing systems* 30.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models." In: *Advances in neural information processing systems* 33, pp. 6840–6851.

Hoffman, Robert R, Shane T Mueller, Gary Klein, and Jordan Litman (2018). "Metrics for explainable AI: Challenges and prospects." In: *arXiv preprint arXiv:1812.04608*.

Hoque, Ehsan, Matthieu Courgeon, Jean-claude Martin, Bilge Mutlu, and Rosalind W. Picard (2013). "MACH: my automated conversation coach." In: *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*.

Hsieh, Hsiu-Fang and Sarah E Shannon (2005). "Three approaches to qualitative content analysis." In: *Qualitative health research* 15.9, pp. 1277–1288.

Huang, Xun and Serge Belongie (2017). "Arbitrary style transfer in real-time with adaptive instance normalization." In: *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510.

Huang, Ziqi, Kelvin CK Chan, Yuming Jiang, and Ziwei Liu (2023). "Collaborative diffusion for multi-modal face generation and editing." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6080–6090.

Huber, Tobias, Maximilian Demmler, Silvan Mertes, Matthew L. Olson, and Elisabeth André (2023). "GANterfactual-RL: Understanding Reinforcement Learning Agents' Strategies through Visual Counterfactual Explanations." In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*. Ed. by Noa Agmon, Bo An, Alessandro Ricci, and William Yeoh. ACM, pp. 1097–1106. DOI: 10.5555/3545946.3598751. URL: https://dl.acm.org/doi/10.5555/3545946.3598751.

Huber, Tobias, Katharina Weitz, Elisabeth André, and Ofra Amir (2020). "Local and Global Explanations of Agent Behavior: Integrating Strategy Summaries with Saliency Maps." In: *CoRR* abs/2005.08874. arXiv: 2005.08874. URL: https://arxiv.org/abs/2005.08874.

Iashin, Vladimir and Esa Rahtu (2021). "Taming visually guided sound generation." In: *arXiv preprint arXiv:2110.08791*.

Ishii, Ryo and Yukiko I. Nakano (2010). "An Empirical Study of Eye-gaze Behaviors: Towards the Estimation of Conversational Engagement in Human-agent Communication." In: *Proceedings of the 2010 Workshop on Eye Gaze in Intelligent Human Machine Interaction*. EGIHMI '10. Hong Kong, China: ACM, pp. 33–40. ISBN: 978-1-60558-999-2.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros (2017). "Image-to-image translation with conditional adversarial networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134.

Jain, Saksham, Gautam Seth, Arpit Paruthi, Umang Soni, and Girish Kumar (2022). "Synthetic data augmentation for surface defect detection and classification using deep learning." In: *Journal of Intelligent Manufacturing*, pp. 1–14.

Jayasumana, Sadeep, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar (2024). "Rethinking fid: Towards a better evaluation metric for image generation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9307–9315.

Jiang, Yiding, Dilip Krishnan, Hossein Mobahi, and Samy Bengio (2018). "Predicting the generalization gap in deep networks with margin distributions." In: *arXiv preprint arXiv:1810.00113*.

Jiang, Yuechi, Benny Drescher, and Guoguang Yuan (2023). "A GAN-based multi-sensor data augmentation technique for CNC machine tool wear prediction." In: *IEEE Access*.

Kaliakatsos-Papakostas, Maximos, Aggelos Gkiokas, and Vassilis Katsouros (2018). "Interactive control of explicit musical features in generative lstm-based systems." In: *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, pp. 1–7.

Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen (2017). "Progressive growing of gans for improved quality, stability, and variation." In: *arXiv preprint arXiv:1710.10196*.

Karras, Tero, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila (2020). "Training generative adversarial networks with limited data." In: *Advances in neural information processing systems* 33, pp. 12104–12114.

Karras, Tero, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila (2021). "Alias-free generative adversarial networks." In: *Advances in neural information processing systems* 34, pp. 852–863.

Karras, Tero, Samuli Laine, and Timo Aila (2018). "A Style-Based Generator Architecture for Generative Adversarial Networks." In: *CoRR* abs/1812.04948. arXiv: 1812.04948. URL: http://arxiv.org/abs/1812.04948.

Karras, Tero, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila (2020). "Analyzing and improving the image quality of stylegan." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119.

Keane, Mark T, Eoin M Kenny, Eoin Delaney, and Barry Smyth (2021). "If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques." In: *arXiv preprint arXiv:2103.01035*.

Keane, Mark T, Eoin M Kenny, Mohammed Temraz, Derek Greene, and Barry Smyth (2021). "Twin Systems for DeepCBR: A Menagerie of Deep Learning and Case-Based Reasoning Pairings for Explanation and Data Augmentation." In: *arXiv preprint arXiv:2104.14461*.

Keane, Mark T and Barry Smyth (2020). "Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI)." In: *Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020, Proceedings 28*. Springer, pp. 163–178.

Kenny, Eoin M and Mark T Keane (2020). "On generating plausible counterfactual and semi-factual explanations for deep learning." In: *arXiv preprint arXiv:2009.06399*.

Kessler, Ronald C (1997). "The effects of stressful life events on depression." In: *Annual review of psychology* 48.1, pp. 191–214.

Khorram, Saeed and Li Fuxin (2022). "Cycle-consistent counterfactuals by latent transformations." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10203–10212.

Kidwell, Mardi (2013). "Framing, grounding, and coordinating conversational interaction: Posture, gaze, facial expression, and movement in space." In: *Body - Language - Communication. An International Handbook on Multimodality in Human Interaction*. De Gruyter Mouton, pp. 100–113.

Kim, Been, Rajiv Khanna, and Oluwasanmi O Koyejo (2016). "Examples are not enough, learn to criticize! criticism for interpretability." In: *Advances in neural information processing systems* 29, pp. 2280–2288.

Kim, Minchul, Feng Liu, Anil Jain, and Xiaoming Liu (2023). "Dcface: Synthetic face generation with dual condition diffusion model." In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 12715–12725.

Kingma, Diederik P (2013). "Auto-encoding variational bayes." In: *arXiv preprint arXiv:1312.6114*.

Klefeker, Josephine, Libi Striegl, and Laura Devendorf (2020). "What HCI can learn from ASMR: Becoming enchanted with the mundane." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–12.

Knapp L., Mark and Judith Hall A. (1997). *Nonverbal Communication in Human Interaction*. Harcourt Brace.

Körber, Moritz (2018). "Theoretical considerations and development of a questionnaire to measure trust in automation." In: *Congress of the International Ergonomics Association*. Springer, pp. 13–30.

Kossaifi, Jean, Georgios Tzimiropoulos, Sinisa Todorovic, and Maja Pantic (2017). "AFEW-VA database for valence and arousal estimation in-the-wild." In: *Image and Vision Computing* 65, pp. 23–36.

Krauß, Veronika, Alexander Boden, Leif Oppermann, and René Reiners (2021). "Current practices, challenges, and design implications for collaborative AR/VR application development." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15.

Krizhevsky, Alex and Geoffrey Hinton (2009). "Learning multiple layers of features from tiny images (Technical Report)." In: *University of Toronto*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2017). "Imagenet classification with deep convolutional neural networks." In: *Communications of the ACM* 60.6, pp. 84–90.

Lai, Yuyo, Shyh-Kang Jeng, Der-Tzung Liu, and Yo-Chung Liu (2006). "Automated optimization of parameters for FM sound synthesis with genetic algorithms." In: *International Workshop on Computer Music and Audio Technology*. Citeseer, p. 205.

Lang, Peter J, Margaret M Bradley, Bruce N Cuthbert, et al. (1997). "International affective picture system (IAPS): Technical manual and affective ratings." In: *NIMH Center for the Study of Emotion and Attention* 1, pp. 39–58.

Lang, Peter J., Margaret M. Bradley, and B. N. Cuthbert (1997). "Motivated attention: Affect, activation, and action." In: *Attention and orienting: Sensory and motivational processes*. Ed. by P. J. Lang, R. F. Simons, and M. T. Balaban. Psychology Press, pp. 97–135.

Laugel, Thibault, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki (2019). "The dangers of post-hoc interpretability: Unjustified counterfactual explanations." In: *arXiv preprint arXiv:1907.09294*.

Laugwitz, Bettina, Theo Held, and Martin Schrepp (2008). "Construction and evaluation of a user experience questionnaire." In: *Symposium of the Austrian HCI and usability engineering group*. Springer, pp. 63–76.

Lee, Chae Young, Anoop Toffy, Gue Jun Jung, and Woo-Jin Han (2018). "Conditional WaveGAN." In: *arXiv preprint arXiv:1809.10636*.

Li, Chuan and Michael Wand (2016). "Combining markov random fields and convolutional neural networks for image synthesis." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2479–2486.

Li, Yu, Lizhong Ding, and Xin Gao (2018). "On the decision boundary of deep neural networks." In: *arXiv preprint arXiv:1808.05385*.

Lin, Jianxin, Yingce Xia, Tao Qin, Zhibo Chen, and Tie-Yan Liu (2018). "Conditional image-to-image translation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5524–5532.

Liu, Kaixin, Mingkai Zheng, Yi Liu, Jianguo Yang, and Yuan Yao (2022). "Deep autoencoder thermography for defect detection of carbon fiber composites." In: *IEEE Transactions on Industrial Informatics* 19.5, pp. 6429–6438.

Liu, Ming-Yu, Thomas Breuel, and Jan Kautz (2017). "Unsupervised image-to-image translation networks." In: *arXiv preprint arXiv:1703.00848*.

Liu, Mingcong, Qiang Li, Zekui Qin, Guoxin Zhang, Pengfei Wan, and Wen Zheng (2021). "Blendgan: Implicitly gan blending for arbitrary stylized face generation." In: *Advances in Neural Information Processing Systems* 34, pp. 29710–29722.

Luan, Fujun, Sylvain Paris, Eli Shechtman, and Kavita Bala (2017). "Deep photo style transfer." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4990–4998.

Lucey, Patrick, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews (2010). "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression." In: *2010 ieee computer society conference on computer vision and pattern recognition-workshops*. IEEE, pp. 94–101.

Lundberg, Scott M and Su-In Lee (2017). "A Unified Approach to Interpreting Model Predictions." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Madhu, Aswathy and Suresh Kumaraswamy (2019). "Data Augmentation Using Generative Adversarial Network for Environmental Sound Classification." In: *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 1–5.

Mahajan, Divyat, Chenhao Tan, and Amit Sharma (2019). "Preserving causal constraints in counterfactual explanations for machine learning classifiers." In: *arXiv preprint arXiv:1912.03277*.

Margraf, Andreas, Anthony Stein, Leonhard Engstler, Steffen Geinitz, and Jörg Hähner (Dec. 2017). "An Evolutionary Learning Approach to Self-configuring Image Pipelines in the Context of Carbon Fiber Fault Detection." In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE.

Margraf., Andreas, Jörg Hähner., Philipp Braml., and Steffen Geinitz. (2020). "Towards Self-adaptive Defect Classification in Industrial Monitoring." In: *Proceedings of the 9th International Conference on Data Science, Technology and Applications - Volume 1: DATA*, IN-

STICC. SciTePress, pp. 318–327. ISBN: 978-989-758-440-4. DOI: 10.5220/0009893003180327.

Mariani, Giovanni, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi (2018). "Bagan: Data augmentation with balancing gan." In: *arXiv preprint arXiv:1803.09655*.

Masci, Jonathan, Ueli Meier, Dan Ciresan, Jürgen Schmidhuber, and Gabriel Fricout (2012). "Steel defect classification with max-pooling convolutional neural networks." In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–6.

Matsumoto, David Ricky (1988). *Japanese and Caucasian facial expressions of emotion (JACFEE)*. University of California.

McCann, Michael T., Kyong Hwan Jin, and Michael Unser (Nov. 2017). "Convolutional Neural Networks for Inverse Problems in Imaging: A Review." In: *IEEE Signal Processing Magazine* 34.6, pp. 85–95. DOI: 10.1109/msp.2017.2739299.

McCloy, Rachel and Ruth MJ Byrne (2002). "Semifactual "even if" thinking." In: *Thinking & Reasoning* 8.1, pp. 41–67.

McErlean, Agnieszka B Janik and Michael J Banissy (2017). "Assessing individual variation in personality and empathy traits in self-reported autonomous sensory meridian response." In: *Multisensory Research* 30.6, pp. 601–613.

McEwen, Bruce S (2008). "Central effects of stress hormones in health and disease: Understanding the protective and damaging effects of stress and stress mediators." In: *European journal of pharmacology* 583.2-3, pp. 174–185.

McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto (2015). "librosa: Audio and music signal analysis in python." In: *Proceedings of the 14th python in science conference*. Vol. 8.

Mehrabian, A. (Aug. 1995). "Framework for a comprehensive description and measurement of emotional states." In: *Genetic, social, and general psychology monographs* 121.3, pp. 339–361. ISSN: 8756-7547.

Mehrabian, Albert (2007). *Nonverbal Communication*. AldineTransaction.

Menke, Maximilian, Thomas Wenzel, and Andreas Schwung (2022). "Improving gan-based domain adaptation for object detection." In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 3880–3885.

Mertes, Silvan, Tobias Huber, Christina Karle, Katharina Weitz, Ruben Schlagowski, Cristina Conati, and Elisabeth André (2024). "Relevant Irrelevance: Generating Alterfactual Explanations for Image Classifiers." In: *33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*. AAAI Press.

Mertes, Silvan, Tobias Huber, Katharina Weitz, Alexander Heimerl, and Elisabeth André (2022). "GANterfactual—Counterfactual Explanations for Medical Non-experts Using Generative Adversarial Learning." In: *Frontiers in artificial intelligence* 5.

Mertes, Silvan, Christina Karle, Tobias Huber, Katharina Weitz, Ruben Schlagowski, and Elisabeth André (2022). "Alterfactual Explanations–The Relevance of Irrelevance for Explaining AI Systems." In: *IJCAI 2022 Workshop on XAI*.

Mertes, Silvan, Florian Lingenfelser, Thomas Kiderle, Michael Dietz, Lama Diab, and Elisabeth André (2021). "Continuous Emotions: Exploring Label Interpolation in Conditional Generative Adversarial Networks for Face Generation." In: *Proceedings of the 2nd International Conference on Deep Learning Theory and Applications, DeLTA 2021, Online Streaming, July 7-9, 2021*. Ed. by Ana L. N. Fred, Carlo Sansone, and Kurosh Madani. SCITEPRESS, pp. 132–139. DOI: 10.5220/0010549401320139. URL: https://doi.org/10.5220/0010549401320139.

Mertes, Silvan, Andreas Margraf, Steffen Geinitz, and Elisabeth André (2023). "Alternative data augmentation for industrial monitoring using adversarial learning." In: *Deep Learning Theory and Applications*, pp. 1–23.

Mertes, Silvan, Andreas Margraf, Christoph Kommer, Steffen Geinitz, and Elisabeth André (2020). "Data Augmentation for Semantic Segmentation in the Context of Carbon Fiber Defect Detection using Adversarial Learning." In: *Proceedings of the 1st International Conference on Deep Learning Theory and Applications, DeLTA 2020, Lieusaint, Paris, France, July 8-10, 2020*. Ed. by Ana L. N. Fred and Kurosh Madani. ScitePress, pp. 59–67. DOI: 10.5220/0009823500590067. URL: https://doi.org/10.5220/0009823500590067.

Mertes, Silvan, Dominik Schiller, Florian Lingenfelser, Thomas Kiderle, Valentin Kroner, Lama Diab, and Elisabeth André (2023). "Intercategorical Label Interpolation for Emotional Face Generation with Conditional Generative Adversarial Networks." In: *International Conference on Deep Learning Theory and Applications*. Springer, pp. 67–87.

Mertes, Silvan, Marcel Strobl, Ruben Schlagowski, and Elisabeth André (2023). "ASMRcade: Interactive Audio Triggers for an Autonomous Sensory Meridian Response." In: *Proceedings of the 25th International Conference on Multimodal Interaction*, pp. 70–78.

Mertes., Silvan, Alice Baird., Dominik Schiller., Björn Schuller., and Elisabeth André. (2020). "An Evolutionary-Based Generative Approach for Audio Data Augmentation." In: *Proceedings of the 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE.

Miller, Tim (2019). "Explanation in artificial intelligence: Insights from the social sciences." In: *Artificial Intelligence* 267, pp. 1–38.

Miller, Tim (2021). "Contrastive explanation: A structural-model approach." In: *The Knowledge Engineering Review* 36.

Minge, Michael and Laura Riedel (2013). "meCUE–Ein modularer Fragebogen zur Erfassung des Nutzungserlebens." In: *Mensch & Computer 2013–Tagungsband*. Oldenbourg Wissenschaftsverlag, pp. 89–98.

Miranda, Eduardo Reck and John Al Biles (2007). *Evolutionary computer music*. Springer.

Mirza, Mehdi and Simon Osindero (2014). "Conditional generative adversarial nets." In: *arXiv preprint arXiv:1411.1784*.

Mitchell, Thomas (2012). "Automated evolutionary synthesis matching: Advanced evolutionary algorithms for difficult sound matching problems." In: *Soft Computing* 16, pp. 2057–2070.

Mollahosseini, Ali, Behzad Hasani, and Mohammad H Mahoor (2017). "Affectnet: A database for facial expression, valence, and arousal computing in the wild." In: *IEEE Transactions on Affective Computing* 10.1, pp. 18–31.

Molnar, Christoph (2019). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. Christoph Molnar. ISBN: 9780244768522.

Montavon, Grégoire, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller (2019). "Layer-Wise Relevance Propagation: An Overview." In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 193–209.

Morris, Dan, Ian Simon, and Sumit Basu (2008). "Exposing parameters of a trained dynamic model for interactive music creation." In.

Mothilal, Ramaravind K, Amit Sharma, and Chenhao Tan (2020). "Explaining machine learning classifiers through diverse counterfactual explanations." In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617.

Müller, Cornelia, Alan Cienki, Ellen Fricke, Silva Ladewig, David McNeill, and Sedinha Tessendorf (2013). *Body - Language - Communication. An International Handbook on Multimodality in Human Interaction*. De Gruyter Mouton.

Mumuni, Alhassan and Fuseini Mumuni (2022). "Data augmentation: A comprehensive survey of modern approaches." In: *Array* 16, p. 100258.

Mun, Seongkyu, Sangwook Park, David K Han, and Hanseok Ko (2017). "Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane." In: *Proc. DCASE*, pp. 93–97.

Naim, Iftekhar, Md. Iftekhar Tanveer, Daniel Gildea, and Ehsan Hoque (2018). "Automated Analysis and Prediction of Job Interview Performance." In: *IEEE Transactions on Affective Computing* 9, pp. 191–204.

Nan, Zhen and Makoto Fukumoto (2022). "ASMR Sound Generation Simulating the Sounds Heard by a Fetus Using Interactive Evolutionary Computation." In: *2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS)*. IEEE, pp. 1–4.

Neal, Lawrence, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li (2018). "Open set learning with counterfactual images." In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 613–628.

Nemirovsky, Daniel, Nicolas Thiebaut, Ye Xu, and Abhishek Gupta (2022). "CounteRGAN: Generating counterfactuals for real-time recourse and interpretability using residual GANs." In: *Uncertainty in Artificial Intelligence*. PMLR, pp. 1488–1497.

Nicolaou, Mihalis A, Hatice Gunes, and Maja Pantic (2011). "Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space." In: *IEEE Transactions on Affective Computing* 2.2, pp. 92–105.

Odena, Augustus, Christopher Olah, and Jonathon Shlens (2017). "Conditional image synthesis with auxiliary classifier gans." In: *International conference on machine learning*. PMLR, pp. 2642–2651.

Oh, Ji Yeon, Daun Kim, Jae-Yeop Jeong, Jin-Woo Jeong, and Elizaveta Lukianova (2023). "Tingle Just for You: A Preliminary Study of AI-based Customized ASMR Content Generation." In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–7.

Oliveira, Rafaella Cristina, Ana CC Gama, and Max DC Magalhães (2021). "Fundamental voice frequency: acoustic, electroglottographic, and accelerometer measurement in individuals with and without vocal alteration." In: *Journal of Voice* 35.2, pp. 174–180.

Olson, Matthew L, Lawrence Neal, Fuxin Li, and Weng-Keen Wong (2019). "Counterfactual states for atari agents via generative deep learning." In: *arXiv preprint arXiv:1909.12969*.

Olson, Matthew L., Roli Khanna, Lawrence Neal, Fuxin Li, and Weng-Keen Wong (2021). "Counterfactual state explanations for reinforcement learning agents via generative deep learning." In: *Artif. Intell.* 295, p. 103455. DOI: 10.1016/j.artint.2021.103455.

Ooko, Ryota, Ryo Ishii, and Yukiko I. Nakano (2011). "Estimating a User's Conversational Engagement Based on Head Pose Information." In: *Intelligent Virtual Agents*. Ed. by Hannes Högni Vilhjálmsson, Stefan Kopp, Stacy Marsella, and Kristinn R. Thórisson. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 262–268. ISBN: 978-3-642-23974-8.

Ortego, Patxi, Alberto Diez-Olivan, Javier Del Ser, and Basilio Sierra (2020). "Data augmentation for industrial prognosis using generative adversarial networks." In: *Intelligent Data Engineering and Automated Learning–IDEAL 2020: 21st International Conference, Guimaraes, Portugal, November 4–6, 2020, Proceedings, Part II 21*. Springer, pp. 113–122.

Partinen, Markku (1994). "Sleep disorders and stress." In: *Journal of Psychosomatic Research*.

Paulhus, Delroy L., Bryce G. Westlake, Stryker S. Calvez, and P. D. Harms (2013). "Self-presentation style in job interviews: the role of personality and culture." In: *Journal of Applied Social Psychology* 43.10, pp. 2042–2059.

Poerio, Giulia Lara, Emma Blakey, Thomas J Hostler, and Theresa Veltri (2018). "More than a feeling: Autonomous sensory meridian response (ASMR) is characterized by reliable changes in affect and physiology." In: *PloS one* 13.6, e0196645.

Radford, Alec, Luke Metz, and Soumith Chintala (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." In: *arXiv preprint arXiv:1511.06434*.

Ramires, António, Jordan Juras, Julian D Parker, and Xavier Serra (2022). "A study of control methods for percussive sound synthesis based on GANs." In: *Proceedings of the International Conference on Digital Audio Effects (DAFX)*, pp. 224–31.

Reed, Scott, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee (2016). "Generative adversarial text to image synthesis." In: *International conference on machine learning*. PMLR, pp. 1060–1069.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., pp. 91–99. URL: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier." In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144.

Ritschel, Hannes, Ilhan Aslan, Silvan Mertes, Andreas Seiderer, and Elisabeth André (2019). "Personalized synthesis of intentional and emotional non-verbal sounds for social robots." In: *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, pp. 1–7.

Rizos, Georgios, Alice Baird, Max Elliott, and Björn Schuller (2020). "Stargan for Emotional Speech Conversion: Validated by Data Augmentation of End-To-End Emotion Recognition." In: *ICASSP*

2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 3502–3506.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-net: Convolutional networks for biomedical image segmentation.” In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, pp. 234–241.

Rowold, Jens, Sabine Hochholdinger, and N Schaper (2008). *Evaluation und Transfersicherung betrieblicher Trainings: Modelle, Methoden und Befunde*. Hogrefe.

Royer, Amélie, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole, and Kevin Murphy (2020). “Xgan: Unsupervised image-to-image translation for many-to-many mappings.” In: *Domain Adaptation for Visual Understanding*. Springer, pp. 33–49.

Rožanec, Jože M, Patrik Zajec, Spyros Theodoropoulos, Erik Koehorst, Blaž Fortuna, and Dunja Mladenić (2023). “Synthetic data augmentation using GAN for improved automated visual inspection.” In: *Ifac-Papersonline* 56.2, pp. 11094–11099.

Russell, James A and Lisa Feldman Barrett (1999). “Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant.” In: *Journal of personality and social psychology* 76.5, p. 805.

Russell, Stuart and Peter Norvig (2016). “Artificial Intelligence: A Modern Approach Global Edition.” In: *Pearson*.

Salimans, Tim, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen (2016). “Improved techniques for training gans.” In: *Advances in neural information processing systems* 29.

Sanabria, Andrea Rosales, Franco Zambonelli, and Juan Ye (2021). “Unsupervised domain adaptation in activity recognition: A GAN-based approach.” In: *IEEE Access* 9, pp. 19421–19438.

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L. Chen (2018). “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

Schallner, Ludwig, Johannes Rabold, Oliver Scholz, and Ute Schmid (2019). “Effect of Superpixel Aggregation on Explanations in LIME–A Case Study with Biological Data.” In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 147–158.

Scherer, Sebastian, Robin Schön, Katja Ludwig, and Rainer Lienhart (2021). “Unsupervised domain extension for nighttime semantic segmentation in urban scenes.” In.

Schlagowski, Ruben, Silvan Mertes, and Elisabeth André (2021). “Taming the chaos: exploring graphical input vector manipulation user interfaces for GANs in a musical context." In: *Proceedings of the 16th International Audio Mostly Conference*, pp. 216–223.

Schlagowski, Ruben, Fabian Wildgrube, Silvan Mertes, Ceenu George, and Elisabeth André (2022). "Flow with the beat! Human-centered design of virtual environments for musical creativity support in VR." In: *Proceedings of the 14th Conference on Creativity and Cognition*, pp. 428–442.

Schlegl, Thomas, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs (2017). "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery." In: *International conference on information processing in medical imaging*. Springer, pp. 146–157.

Schneeberger, Tanja, Mirella Scholtes, Bernhard Hilpert, Markus Langer, and Patrick Gebhard (2019). "Can social agents elicit shame as humans do?" In: *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, pp. 164–170.

Schrum, Jacob, Jake Gutierrez, Vanessa Volz, Jialin Liu, Simon Lucas, and Sebastian Risi (2020). "Interactive evolution and exploration within latent level-design space of generative adversarial networks." In: *arXiv preprint arXiv:2004.00151*.

Schwarz, Diemo (2005). "Current research in concatenative sound synthesis." In: *International Computer Music Conference (ICMC)*, pp. 1–1.

Selvaraju, Ramprasaath R, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra (2020). "Grad-CAM: visual explanations from deep networks via gradient-based localization." In: *International journal of computer vision* 128, pp. 336–359.

Shao, Siyu, Pu Wang, and Ruqiang Yan (2019). "Generative adversarial networks for data augmentation in machine fault diagnosis." In: *Computers in Industry* 106, pp. 85–93.

Sharmanska, Viktoriia, Lisa Anne Hendricks, Trevor Darrell, and Novi Quadrianto (2020). "Contrastive examples for addressing the tyranny of the majority." In: *arXiv preprint arXiv:2004.06524*.

Shen, Xiaolong, Jianxin Ma, Chang Zhou, and Zongxin Yang (2024). "Controllable 3d face generation with conditional style code diffusion." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 5, pp. 4811–4819.

Shen, Yujun, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou (2020). "Interfacegan: Interpreting the disentangled face representation learned by gans." In: *IEEE transactions on pattern analysis and machine intelligence* 44.4, pp. 2004–2018.

Shen, Yujun and Bolei Zhou (2021). "Closed-form factorization of latent semantics in gans." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1532–1540.

Shorten, Connor and Taghi M Khoshgoftaar (2019). "A survey on image data augmentation for deep learning." In: *Journal of Big Data* 6.1, p. 60.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (2017). "Learning important features through propagating activation differences." In: *International conference on machine learning*. PMlR, pp. 3145–3153.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition." In: *arXiv preprint arXiv:1409.1556*.

Smilkov, Daniel, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg (2017). "Smoothgrad: removing noise by adding noise." In: *arXiv preprint arXiv:1706.03825*.

Snyder, Jeff and Danny Ryan (2014). "The Birl: An Electronic Wind Instrument Based on an Artificial Neural Network Parameter Mapping Structure." In: *NIME*, pp. 585–588.

Soloveitchik, Michael, Tzvi Diskin, Efrat Morin, and Ami Wiesel (2021). "Conditional frechet inception distance." In: *arXiv preprint arXiv:2103.11521*.

Soukup, Daniel and Reinhold Huber-Mörk (2014). "Convolutional neural networks for steel surface defect detection from photometric stereo images." In: *International Symposium on Visual Computing*. Springer, pp. 668–677.

Staar, Benjamin, Michael Lütjen, and Michael Freitag (2019). "Anomaly detection with convolutional neural networks for industrial surface inspection." In: *Procedia CIRP* 79, pp. 484–489.

Stanley, Richard P (1975). "The Fibonacci lattice." In: *Fibonacci Quart* 13.3, pp. 215–232.

Steele, Ric G, Jeffrey A Hall, and Jennifer L Christofferson (2020). "Conceptualizing digital stress in adolescents and young adults: Toward the development of an empirically based model." In: *Clinical Child and Family Psychology Review* 23, pp. 15–26.

Stone, Peter, Rodney Brooks, Erik Brynjolfsson, Ryan Calo, Oren Etzioni, Greg Hager, Julia Hirschberg, Shivaram Kalyanakrishnan, Ece Kamar, Sarit Kraus, Kevin Leyton-Brown, David Parkes, Press William, Saxenian AnnaLee, Shah Julie, Tambe Milind, and Teller Astro (2016). "Artificial intelligence and life in 2030." In: *One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel*.

Stults-Kolehmainen, Matthew A and Rajita Sinha (2014). "The effects of stress on physical activity and exercise." In: *Sports medicine* 44, pp. 81–121.

Sueur, Jérôme, Thierry Aubin, and Caroline Simonis (2008). "Seewave, a free modular tool for sound analysis and synthesis." In: *Bioacoustics* 18.2, pp. 213–226.

Sweller, John, Jeroen JG Van Merrienboer, and Fred GWC Paas (1998). "Cognitive architecture and instructional design." In: *Educational Psychology Review* 10.3, pp. 251–296. DOI: 10.1023/A:1022193728205.

Szarski, Martin and Sunita Chauhan (2022). "An unsupervised defect detection model for a dry carbon fiber textile." In: *Journal of Intelligent Manufacturing* 33.7, pp. 2075–2092.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna (2016). "Rethinking the inception architecture for computer vision." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Takeuchi, Nao and Tomoko Koda (2021). "Initial Assessment of Job Interview Training System Using Multimodal Behavior Analysis." In: *Proceedings of the 9th International Conference on Human-Agent Interaction*. HAI '21. Virtual Event, Japan: Association for Computing Machinery, pp. 407–411. ISBN: 9781450386203.

Tan, Mingxing and Quoc Le (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks." In: *International Conference on Machine Learning*. PMLR, pp. 6105–6114.

Tjoa, Erico and Cuntai Guan (2020). "A survey on explainable artificial intelligence (xai): Toward medical xai." In: *IEEE transactions on neural networks and learning systems* 32.11, pp. 4793–4813.

Tottenham, N (1998). "MacBrain Face Stimulus Set." In: *John D. and Catherine T. MacArthur Foundation Research Network on Early Experience and Brain Development*.

Tubb, Robert and Simon Dixon (2014). "The Divergent Interface: Supporting Creative Exploration of Parameter Spaces." In: *NIME*, pp. 227–232.

Ulhas, Sangeet Sankaramangalam, Shenbagaraj Kannapiran, and Spring Berman (2024). "GAN-Based Domain Adaptation for Creating Digital Twins of Small-Scale Driving Testbeds: Opportunities and Challenges." In: *2024 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 137–143.

Ustun, Berk, Alexander Spangher, and Yang Liu (2019). "Actionable recourse in linear classification." In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 10–19.

Van der Schalk, J, ST Hawk, and AH Fischer (2009). "Validation of the Amsterdam Dynamic Facial Expression Set (ADFES)." In: *Poster for the International Society for Research on Emotions (ISRE), Leuven, Belgium*.

Van Looveren, Arnaud and Janis Klaise (2019). "Interpretable counterfactual explanations guided by prototypes." In: *arXiv preprint arXiv:1907.02584*.

Van Looveren, Arnaud, Janis Klaise, Giovanni Vacanti, and Oliver Cobb (2021). "Conditional generative models for counterfactual explanations." In: *arXiv preprint arXiv:2101.10123*.

Verma, Sahil, John Dickerson, and Keegan Hines (2020). "Counterfactual explanations for machine learning: A review." In: *arXiv preprint arXiv:2010.10596*.

Volz, Vanessa, Jacob Schrum, Jialin Liu, Simon M Lucas, Adam Smith, and Sebastian Risi (2018). "Evolving mario levels in the latent space of a deep convolutional generative adversarial network." In: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 221–228.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell (2017). "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." In: *Harv. JL & Tech.* 31, p. 841.

Wang, Chu-ran, Jing Li, Fandong Zhang, Xinwei Sun, Hao Dong, Yizhou Yu, and Yizhou Wang (2020). "Bilateral Asymmetry Guided Counterfactual Generating Network for Mammogram Classification." In: *arXiv preprint arXiv:2009.14406*.

Wang, Hongshuai, Hangyuan Luo, Xianjie Zhang, Zhiyong Zhao, Junbiao Wang, and Yujun Li (2024). "Automatic defect detection of carbon fiber woven fabrics using machine vision." In: *Mechanics of Advanced Materials and Structures* 31.28, pp. 10921–10934.

Wang, Yaohui, Antitza Dantcheva, and Francois Bremond (2018). "From attributes to faces: a conditional generative network for face generation." In: *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, pp. 1–5.

Wang, Zhou, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli (2004). "Image quality assessment: from error visibility to structural similarity." In: *IEEE transactions on image processing* 13.4, pp. 600–612.

Weinstein, Emily C and Robert L Selman (2016). "Digital stress: Adolescents' personal accounts." In: *new media & society* 18.3, pp. 391–409.

Wu, Yifan, Fan Yang, Yong Xu, and Haibin Ling (2019). "Privacy-protective-GAN for privacy preserving face de-identification." In: *Journal of Computer Science and Technology* 34, pp. 47–60.

Xiao, Han, Kashif Rasul, and Roland Vollgraf (2017). "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." In: *arXiv preprint arXiv:1708.07747*.

Yang, Jeong Hyeon, Nam Kyun Kim, and Hong Kook Kim (2018). "Se-Resnet with Gan-Based Data Augmentation Applied to Acoustic Scene Classification." In: *DCASE 2018 workshop*.

Yang, Karren, Bryan Russell, and Justin Salamon (2020). "Telling left from right: Learning spatial correspondence of sight and sound." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9932–9941.

Yi, Wei, Yaoran Sun, and Sailing He (2018). "Data augmentation using conditional GANs for facial emotion recognition." In: *2018 Progress in Electromagnetics Research Symposium (PIERS-Toyama)*. IEEE, pp. 710–714.

Zaltron, Nicola, Luisa Zurlo, and Sebastian Risi (2019). "CG-GAN: An Interactive Evolutionary GAN-based Approach for Facial Composite Generation." In: *arXiv preprint arXiv:1912.05020*.

Zhang, Baobao and Allan Dafoe (2019). "Artificial intelligence: American attitudes and trends." In: *Available at SSRN 3312874*.

Zhang, Mingxu, Hongxia Wang, Peisong He, Asad Malik, and Hanqing Liu (2022). "Exposing unseen GAN-generated image using unsupervised domain adaptation." In: *Knowledge-Based Systems* 257, p. 109905.

Zhang, Tianyi, Björn Hartmann, Miryung Kim, and Elena L Glassman (2020). "Enabling data-driven api design with community usage data: A need-finding study." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13.

Zhao, Wenqi, Satoshi Oyama, and Masahito Kurihara (2021). "Generating natural counterfactual visual explanations." In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 5204–5205.

Zhao, Yunxia (2020). "Fast Real-time Counterfactual Explanations." In: *arXiv preprint arXiv:2007.05684*.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A Efros (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks." In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

Zucco, Chiara, Huizhi Liang, Giuseppe Di Fatta, and Mario Cannataro (2018). "Explainable sentiment analysis with applications in medicine." In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 1740–1747.

Final Version as of August 26, 2025