OTTO-FRIEDRICH-UNIVERSITY BAMBERG

MASTER'S THESIS

Applying Explainable Artificial Intelligence for Deep Learning Networks to Decode Facial Expressions of Pain and Emotions

Author:
Katharina Blandina WEITZ
Matrikelnr: 1706341

Assessor:
Prof. Dr. Ute SCHMID
Supervisor:
Teena HASSAN

A thesis submitted in partial fulfillment of the requirements
for the degree of
Master of Science in Computing in the Humanities

in the
Cognitive Systems Group, Faculty of Information Systems and Applied Computer Sciences, Otto-Friedrich-University Bamberg
in cooperation with
Intelligent Systems Group, Fraunhofer IIS

August 31, 2018

"If you can't explain it simply, you don't understand it well enough."

Albert Einstein

OTTO-FRIEDRICH-UNIVERSITY BAMBERG

Abstract

Cognitive Systems Group, Faculty of Information Systems and Applied Computer Sciences,
Otto-Friedrich-University Bamberg
Intelligent Systems Group, Fraunhofer IIS

Master of Science in Computing in the Humanities

Applying Explainable Artificial Intelligence for Deep Learning Networks to Decode Facial Expressions of Pain and Emotions

by Katharina Blandina WEITZ

Deep learning networks are successfully used for object and face recognition in images and videos. In order to be able to apply such networks in practice, for example in hospitals as a pain recognition tool, the current procedures are only suitable to a limited extent. The advantage of deep learning methods is that they can learn complex non-linear relationships between raw data and target classes without limiting themselves to a set of hand-crafted features provided by humans. However, the disadvantage is that due to the complexity of these networks, it is not possible to interpret the knowledge that is stored inside the network. It is a black-box learning procedure. Explainable Artificial Intelligence (XAI) approaches mitigate this problem by extracting explanations for decisions and representing them in a human-interpretable form. The aim of this master's thesis is to investigate different XAI methods and apply them to explain how a deep learning network distinguishes facial expressions of pain from facial expressions of emotions such as happiness and disgust. The results show that the CNN has problems to distinguish between pain and happiness. By the usage of XAI it can be shown that the CNN discovers features for happiness in painful images, when the person shows no typical pain related facial expressions. Furthermore, the results show that the learned features of the network are dataset-independent. It can be concluded that model-specific XAI approaches seem to be a promising base to make the learned features visible for humans. This is on the one hand the first step to improve CNNs and on the other hand, to increase the comprehensibility of such black box systems.

Contents

Abstract ii

1 Introduction 1

2 Background and Theory 2
2.1 Psychological Constructs of Pain and Emotions . . . 2
2.1.1 Emotions . . . 2
2.1.2 Pain . . . 2
2.1.3 Measurement of Pain and Emotions . . . 3
2.2 Machine Learning for Facial Expression Analysis . . . 5
2.2.1 Convolutional Neural Networks . . . 6
2.2.2 VGG Face . . . 9
2.2.3 Data Augmentation . . . 10
2.3 Explainable Artificial Intelligence for Deep Learning Networks . . . 12
2.3.1 Deconvnet . . . 13
2.3.2 Backpropagation . . . 15
2.3.3 Guided Backpropagation . . . 16
2.3.4 CAM and (Guided) Grad-CAM . . . 16
2.3.5 Layer-wise Relevance Propagation . . . 18
2.3.6 Local Interpretable Model-Agnostic Explanations . . . 21

3 Research Questions 23

4 Material & Procedure 24
4.1 Material . . . 24
4.1.1 BioVid Dataset . . . 24
4.1.2 Data Preparation . . . 26
4.2 Implementation of CNN . . . 26
4.3 Implementation of Explainable AI Methods . . . 27

5 Results 28
5.1 Classification Results . . . 28
5.2 Results from Explainable AI Methods . . . 32
5.2.1 Deconvnet . . . 33
5.2.2 Backpropagation . . . 33
5.2.3 Guided Backpropagation . . . 34
5.2.4 Grad-CAM . . . 36
5.2.5 Guided Grad-CAM . . . 37
5.2.6 LRP . . . 38
5.2.7 LIME . . . 38
5.3 Generalization . . . 38

6 Discussion **52**
6.1 Predictive Performance 52
6.2 Decision Interpretation & Feature Explanation 52
6.3 Limitations 54
6.4 Future Research 55

Bibliography **57**

A CNN Architectures **63**
A.1 CNN Architecture With Early Stopping 63
A.2 CNN Architectures Without Early Stopping 63

Declaration of Authorship **66**

List of Figures

2.1 Different faces of pain . . . 4
2.2 Convolution in CNNs . . . 7
2.3 Max pooling in CNNs . . . 8
2.4 Dropout as regularization technique . . . 10
2.5 VGG face architecture . . . 11
2.6 Different data augmentation techniques . . . 12
2.7 Structure of deconvnet: Overview . . . 13
2.8 Structure of deconvnet: Forward and backward pass . . . 14
2.9 Structure of deconvnet: Switches . . . 14
2.10 Example for XAI methods: Backpropagation . . . 16
2.11 Different ReLU functions of XAI methods . . . 17
2.12 Example for XAI methods: CAM . . . 18
2.13 Example for XAI methods: (Guided) Grad-CAM vs. guided backpropagation . . . 19
2.14 Illustration of LRP for a multilayer neural network architecture . . . 20
2.15 Illustration of XAI method: LIME . . . 22

4.1 Overview of the procedure followed in the master's thesis . . . 25
4.2 Example of a cropped image from BioVid database . . . 26

5.1 CNN: Accuracy and loss of fold 5 . . . 29
5.2 CNN: Confusion matrix of fold 5 . . . 30
5.3 CNN: Normalized confusion matrix of fold 5 . . . 30
5.4 Visualization of the misclassification of the CNN using XAI methods (1) 31
5.5 Visualization of the misclassification of the CNN using XAI methods (2) 31
5.6 Raw BioVid images used for XAI . . . 32
5.7 Deconvnet: Saliency maps of BioVid images . . . 33
5.8 Backpropagation: Saliency maps of BioVid images . . . 34
5.9 Guided backpropgation: Saliency maps of BioVid images . . . 35
5.10 Grad-CAM: Heatmaps of BioVid images . . . 36
5.11 Guided Grad-CAM: Salieny maps of BioVid images . . . 37
5.12 LRP-Z: Heatmaps of BioVid images . . . 39
5.13 LRP-PresetAFlat: Heatmaps of BioVid images . . . 40
5.14 LRP-PresetBFlat: Heatmaps of BioVid images . . . 41
5.15 LRP-Epsilon: Heatmaps of BioVid images . . . 42
5.16 LIME: Heatmaps of BioVid images . . . 43
5.17 LIME with positive and negative super-pixels: Heatmaps of BioVid images . . . 44
5.18 Raw Actorstudy and UNBC-McMaster shoulder pain expression archive database images used for XAI . . . 46
5.19 Grad-CAM: Visualizing images for generalization . . . 47
5.20 Guided Grad-CAM: Visualizing images for generalization . . . 48
5.21 LRP: Visualizing images for generalization . . . 49

5.22 LIME: Visualizing images for generalization 50
5.23 LIME with positive and negative super-pixels: Visualizing images for generalization 51

List of Tables

2.1 Facial expressions of the six basic emotions with corresponding AUs (Ekman & Friesen, 2003; Friesen & Ekman, 1983). 3

4.1 Extracted BioVid data before balancing and data cleaning steps. 25
4.2 Extracted BioVid data after balancing and data cleaning steps. 26

5.1 Results of the 5-fold cross-validation of the best performing CNN. . . . 28
5.2 Results of the confusion matrix of fold 5. 29

A.1 Results of the 5-fold cross-validation using early stopping and a dropout of 0.5. 63
A.2 Results of the 5-fold cross-validation using a fixed epoch size of 7 and a dropout of 0.5 64
A.3 Results of the 5-fold cross-validation using a fixed epoch size of 7 and a L2 regularization of 0.0001. 64
A.4 Results of the 5-fold cross-validation using a fixed epoch size of 3 and a dropout of 0.5. 64
A.5 Results of the 5-fold cross-validation using a fixed epoch size of 3 and a L2 regularization of 0.0001. 65

List of Abbreviations

AU	**A**ction **U**nit
Adam	**Ada**ptive **M**oments estimation
BioVid	**BioVid** Heat Pain dataset
CAM	**C**lass **A**ctivation **M**apping
CNNs	**C**onvolutional **N**eural **N**etworks
IAPS	**I**nternational **A**ffective **P**icture **S**ystems
LFW	**L**abeled **F**aces in the **W**ild
LIME	**L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations
LRP	**L**ayer-wise **R**elevance **P**ropagation
LSTM	**L**ong **S**hort-**T**erm **M**emory
ReLU	**Re**ctified **L**inear **U**nit
SGD	**S**tocastic **G**radient **D**escent
YTF	**Y**ou **T**ube **F**aces in the Wild
XAI	**E**xplainable **A**rtificial **I**ntelligence

Chapter 1

Introduction

Facial expressions are one of the most important human nonverbal signals in interacting with other people and thus contribute to the emergence and maintenance of social relationships (Frith, 2009). One of the tasks of facial expressions is to communicate emotions (Ambady & Rosenthal, 1992; Ekman & Rosenberg, 1997). Emotions like happiness, anger, sadness, disgust, surprise, and fear are universal, which means that the same facial expressions can be associated with these emotions across different cultures (Ekman & Friesen, 1971). Especially when people are unable to express themselves verbally (e. g., through illness, accidents or congenital disabilities, or due infancy), facial expression is often the only way for these people to express emotions. For this reason, nursing staff in clinics and care facilities in particular are required to observe patients closely in order to be able to read their emotions and take action, if necessary. Due to the already significantly increased number of patients, especially in nursing homes, and the prognosis that more and more people will be cared for in such facilities in the future (Statistisches Bundesamt, 2015), a patient's facial expressions that are only monitored by people will not be manageable in the long term. Additionally, humans often have problems in differentiating between pain and other facial expressions (Aviezer, Trope, & Todorov, 2012; Brahnam, Chuang, Shih, & Slack, 2006). Therefore, in addition to the (classical) exploration of emotions in a psychological context, research into a technical solution for distinguishing emotions and pain has gained greater importance in the last decade. A system, which uses explainable artificial intelligence (XAI) to describe how pain differs from emotions, can also be used to train nursing staff to improve their ability to recognise pain correctly.

Chapter 2

Background and Theory

In this chapter, the theoretical background relevant for this master's thesis is described. First, the psychological constructs of emotions and pain are considered. This is followed by a description of deep learning networks. At the end of this chapter, the theoretical principles of XAI methods used in this master's thesis are explained.

2.1 Psychological Constructs of Pain and Emotions

In the following section the constructs of pain and emotions will be presented and distinguished from each other. Then, a short overview about measurements of pain and emotions in research and practical settings is provided.

2.1.1 Emotions

Plutchik (1982) defined emotion as an inferring complex sequence of reactions to a stimulus including cognitive evaluation, subjective changes, autonomous and neuronal arousal, impulses for action and behaviour. This has an effect on the stimulus that initiated the complex sequence. Emotions are one of the key characteristics for human experience (Vytal & Hamann, 2010). Emotional experiences permeate every area of (mental) life (Kassam, Markey, Cherkassky, Loewenstein, & Just, 2013). They have influence on the content and type of thoughts (Clore & Huntsinger, 2007), on decisions and actions (Damasio, 1994; Overskeid, 2000), and on memory and perception (Phelps, 2004; Phelps, Ling, & Carrasco, 2006; Scott et al., 1997). Therefore it is necessary and important to characterize the structure of emotional experience (Vytal & Hamann, 2010). The discrete emotion theories based on the work of Darwin (1873). His ideas were later taken up, expanded and made empirically accessible by Ekman (1971) and represent one possibility of the characterization of emotions. Their approach is based on a set of emotions that is universally valid, meaning that these emotions can be expressed and interpreted identically all over the world (Ekman & Friesen, 1971). These emotions are called basic emotions: happiness, anger, sadness, disgust, surprise, and fear (Ekman & Friesen, 1971). The facial expressions associated with these emotions (Ekman & Friesen, 2003; Friesen & Ekman, 1983) are listed in Table 2.1.

2.1.2 Pain

Merskey and Bogduk (2012, p. 209) describe pain as "an unpleasant sensory and emotional experience associated with actual or potential tissue damage, or described in terms of such damage." Pain has the function of demanding attention and thus stimulating and maintaining escape, recovery and healing (Williams, 2002). Without

TABLE 2.1: Facial expressions of the six basic emotions with corresponding AUs (Ekman & Friesen, 2003; Friesen & Ekman, 1983).

Emotion	**Action units**	**Description**
Happiness	9+12	Nose wrinkler, lip corner puller
Sadness	1+4+15	Inner brow raiser, brow lowerer, lip corner depressor
Surprise	1+2+5+26	Inner brow raiser, outer brow raiser, upper lid raiser, jaw drop
Fear	1+2+4+5+7+20+26	Inner brow raiser, outer brow raiser, brow lowerer, upper lid raiser, lid tightener, lip stretcher, jaw drop
Anger	4+5+7+23	Brow lowerer, upper lid raiser, lid tightener, lip tightener
Disgust	9+15+16	Nose wrinkler, lip corner depressor, lower lip depressor

pain, human life would be significantly shorter (Wall, 1999) or, as Damasio (1994) concluded: Pain increases the probability of survival. Facial expressions serve as a person's behavioural resource to express pain and at the same time can be perceived as being in pain by other people (Prkachin, 2009). Pain therefore has a very important social component, as the expression of pain triggers social reactions such as empathy, care, and nursing (Williams, 2002). Moreover, almost the same Action Units (AUs) are involved in the facial expression of pain as in that of disgust (Kunz, Peter, Huster, & Lautenbacher, 2013). Despite this, the study by Kunz et al. (2013) showed that people are able to judge whether the person shown expresses pain or disgust on the basis of facial expressions in pictures even without contextual information. In contrast to this, Aviezer et al. (2012) and Brahnam et al. (2006) found out that the abilities of humans to distinguish between pain and other facial expressions without context information is quite bad. One explanation for these different results is that unlike emotions such as happiness, where certain AUs are activated, pain in highly inter-individual. These inter-individual clusters of pain are displayed in Figure 2.1. This results in various facial expressions of pain (Kunz & Lautenbacher, 2014).

2.1.3 Measurement of Pain and Emotions

Emotions can be represented by the facial expression of a person (Duchenne, 1990). These facial expressions represent an important measure for the study of emotions, social interactions, communication, personality and development of people, especially children (Ekman, Huang, Sejnowski, & Hager, 1993; Ekman & Rosenberg, 1997; Ekman & Oster, 1979). In the expression of emotions through a person's facial expressions, there are various distinctions to make. Thus emotions can be represented by facial expressions, although the person does not feel the emotion at all (Ekman, 1993). The French anatomist Duchenne de Boulogne was the first to describe the differences in facial expressions with real and played joy (Ekman, Davidson, & Friesen, 1990). The Duchenne smile named after him (Ekman, 1989) differs from a played smile in that besides the contraction of the zygomaticus major muscle, which is needed to pull up the corners of the mouth to a smiling mouth, the

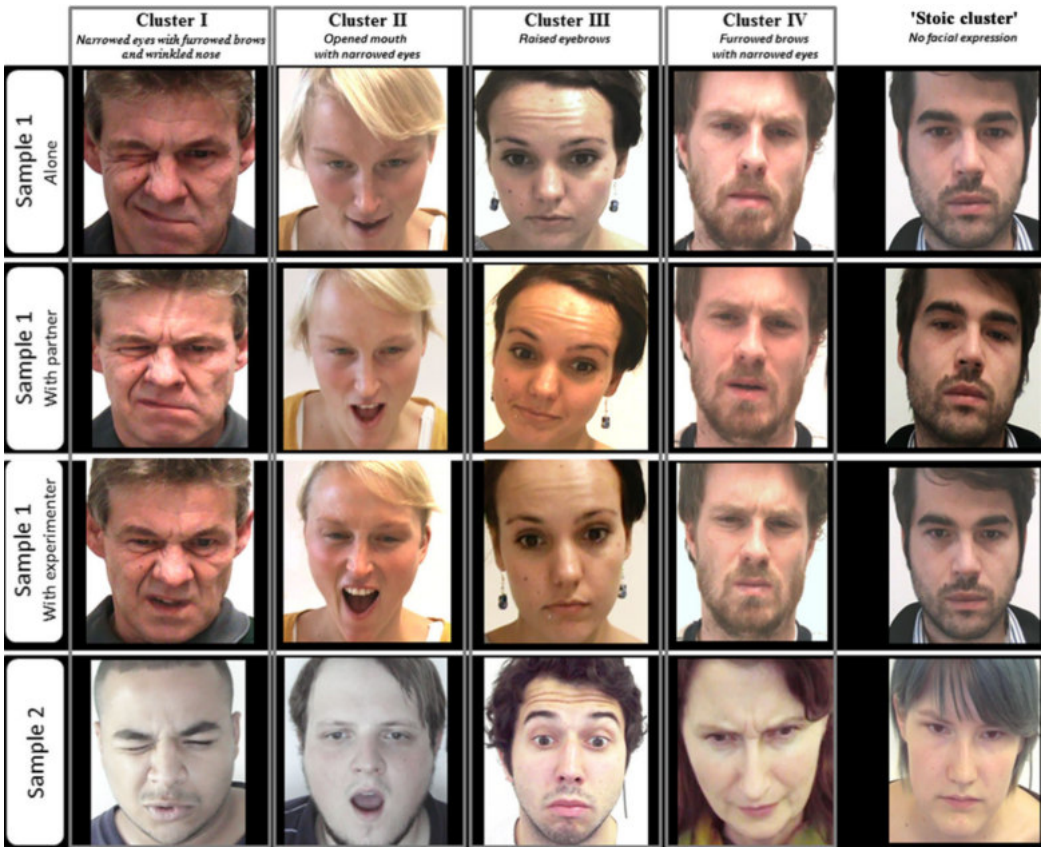


FIGURE 2.1: Different expression of pain, depending on inter-individual differences (Graphic from Kunz and Lautenbacher (2014)).

orbicularis oculi muscle, which reduces the eye opening, is also activated. Besides the facial pretence of emotions, there is also the possibility that people feel emotions but do not express them through facial expressions (Ekman, 1993). There are also inter-individual differences in the facial expression of emotions (Ekman, 1993). Besides, muscle movements in the face are more or less intense depending on the emotion and strength of the emotion (Aviezer et al., 2012). In research, the Facial Action Coding System (Friesen & Ekman, 1978) is a quantitative approach to encode these differences in facial expressions. For this, AUs are used, which describe specific muscle movements in the face. Other ways to objectively measure emotions are through the use of videos, functional magnetic resonance imaging or electroencephalography (Bartlett, Hager, Ekman, & Sejnowski, 1999).

Four types of measurements are commonly used in research in assessing pain: psychophysical methods, rating scale methods, magnitude estimation procedures, and in the behavioural way, the measurement of performance in different tasks. The Facial Action Coding System can also be used to measure the facial expression of pain (Craig, Prkachin, & Grunau, 1992). In clinical settings, self-report in the form of questionnaires (e.g., Visual Analogue Scale for Pain (McCormack, David, & Sheather, 1988), Numeric Rating Scale (Downie et al., 1978)), pain diaries (e.g, de Wit et al. (1999)) or verbal descriptions (e.g., Gracely, McGrath, and Dubner (1978)) has prevailed to measure pain (Werner et al., 2014).

2.2 Machine Learning for Facial Expression Analysis

In addition to the classical objective measurement methods in pain and emotion research, which often require complex human analysis, the use of machine learning methods has been researched and continuously improved in recent years to an automatic and thus time-saving alternative to the classical methods (Bartlett et al., 1999). In machine learning, emotion recognition using facial expressions is a subfield of social signal processing (Pitaloka, Wulandari, Basaruddin, & Liliana, 2017). Using machine learning, classifiers can learn by induction (Sebastiani, 2002). This means that the classifier learns patterns using examples. A great advantage of machine learning is that the classifier learns automatically, thereby a complex manual generation of a classifier by human experts is not necessary (Sebastiani, 2002). To define learning as a description about the improvements a computer program can make by itself, Mitchell (1997, p. 2) says that

> "A computer program is said to **learn** from experience E with respect to some class of task T and performance measure P, if its performance at tasks in T as measured by P, improves with experience E."

One type of task T is classification. Here the computer program has to solve the problem to which of k categories some input data belongs (Goodfellow, Bengio, & Courville, 2016). For this, the learning algorithm should produce a function $f : \mathbb{R}^n \rightarrow \{1, ..., k\}$. When using $y = f(x)$, the program assigns an input, which is described by the vector x, to a category y (Goodfellow et al., 2016). The category y is described by a numeric value (e.g., 0 =car, 1 =house). The experience E, a computer program can make depends on the approach for learning task. One approach is supervised learning (Goodfellow et al., 2016). Here, the datasets which are used by the computer program for training, are labeled. This means that the computer program is supervised through the information to which class the data belongs. Classification tasks are often found in object recognition (Goodfellow et al.,

2016). Object recognition (Krizhevsky, Sutskever, & Hinton, 2012; LeCun et al., 1990; Ioffe & Szegedy, 2015) and one subcategory of it, face recognition (Parkhi, Vedaldi, & Zisserman, 2015) is nowadays very successfully implemented using deep learning approaches (Krizhevsky et al., 2012; Matsugu, Mori, Mitari, & Kaneda, 2003). Deep learning represents a specific approach of machine learning (Goodfellow et al., 2016). It belongs to the so-called 'representation-learning methods', which means that these systems are fed with raw data and they automatically and independently learn the representations necessary for classification (LeCun, Bengio, & Hinton, 2015). This distinguishes deep learning from conventional machine learning methods, which require careful and elaborate feature extraction to bring the raw data into a suitable format for learning (LeCun et al., 2015). In many machine learning applications like image analysis, the input usually consists of multidimensional data arrays and the kernel is usually a multidimensional array of parameters (Goodfellow et al., 2016). The performance measure P as described by Mitchell (1997) can be, for example, the measurement of accuracy.

Machine learning, especially deep learning, was able to improve greatly in the recent years due to technological improvements such as the availability of high-speed GPUs (Samek, Wiegand, & Müller, 2017), the availability of large amount of data, and the development of open software frameworks such as Caffe (Jia et al., 2014) or Tensorflow (Abadi et al., 2016). Therefore, deep learning approaches like Convolutional Neural Networks (CNNs) have a large impact in the field of computer vision research (Parkhi et al., 2015). Since 2009, deep learning networks won many international contests in pattern recognition (Schmidhuber, 2015).

2.2.1 Convolutional Neural Networks

LeCun (1989) described a foldable artificial neural network, a Convolutional Neural Network, for the first time in the late 1980s. CNNs belong to the feedforward neural networks (Goodfellow et al., 2016). To understand CNNs, first a short overview about feedforward neural networks is given. After that, the specific assumptions for CNNs are explained.

Feedforward neural networks approximate a function F^*. As an example, a classifier $y = f^*(x)$ is given, which assigns an input x to a category y (Goodfellow et al., 2016). Feedforward thus describes the information flow through network (Goodfellow et al., 2016). In this example, a feedforward network defines a mapping $y = f(x;\theta)$ and learns the values of the parameter θ, which represent an approximation of the best function. Feedforward neural networks are able to distinguish data which are not linear separable (Dreiseitl & Ohno-Machado, 2002). They use a transformed input $\phi(x)$, where ϕ represents a non-linear transformation of the data x. When using deep learning, a neural network contains not only one $f(x)$ function, but several functions that are built in layers. For example, a function $f(x)$ with three layers consists of the form $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$ (Goodfellow et al., 2016). The more layers, the deeper the network.

In contrast to classical (deep) feedforward neural networks, which use matrix multiplication as the basis for their calculations, CNN uses convolutions (Goodfellow et al., 2016). The components which are specific for a CNN are convolutional layers, pooling layers, and fully-connected layers (LeCun et al., 2015). In the convolutional layers, calculation is done using convolution (see Figure 2.2). This is achieved by using a filter (the notation 'kernel' or 'feature detector' is also used), which scans over a given image. In doing so, matrix multiplication is used and the results are written into a feature map (LeCun et al., 2015). Convolutional layers can

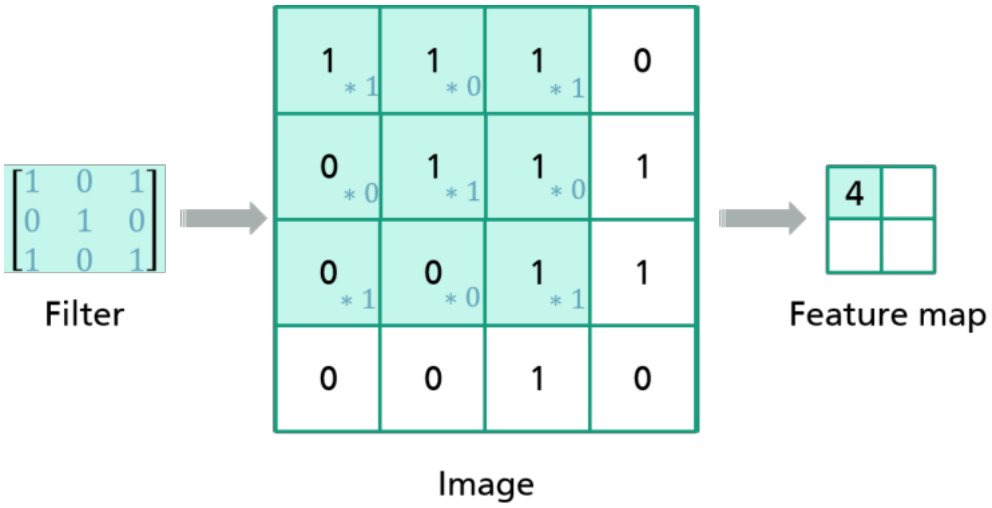


FIGURE 2.2: Convolution: A filter matrix (left) is multiplied with the pixel values of a binary image (middle). The results of the multiplication is stored into a feature map (right) (Graphic inspired by [1]).

be seen as feature extractors (Lin, Chen, & Yan, 2014) which makes it possible to have an end-to-end system which detects the features automatically and trains a classifier using these features.

In the example displayed in Figure 2.2, the operation is done in 2D. Using RGB images, three 2D arrays are used, including width and height informations for every colour channel (LeCun et al., 2015).

The resulting output of a convolutional layer is passed through a activation function (LeCun et al., 2015). Nowadays, the rectified linear unit (ReLU) activation function is mostly applied (Goodfellow et al., 2016; Jarrett, Kavukcuoglu, Ranzato, & LeCun, 2009). Krizhevsky et al. (2012) describe this activation function as a non-linear, non-saturating function in the form of

$$f(x) = max(0, x), \tag{2.1}$$

where f(x) returns zero when $x < 0$ and $f(x)$ returns x when $x \geq 0$. Deep neural networks using ReLUs have significantly shorter training times than saturating non-linearities (Krizhevsky et al., 2012). Another activation function is the softmax activation. It is often used in the last layer of a CNN (Goodfellow et al., 2016) to reflect the probability distribution of n classes. The softmax activation is formulated as

$$softmax(z)_i = \frac{exp(z_i)}{\sum_j exp(z_j)}, \tag{2.2}$$

where z is a vector of the inputs to the last layer (output layer) and i indexes the inputs of the vector z.

A convolution layer is followed by a pooling layer. Pooling layers are used in CNNs to reduce the dimensionality and therefore the number of parameters in the network. This leads to shorter training time and helps to reduce overfitting. Overfitting can be seen as "memorizing the training cases" (Dreiseitl & Ohno-Machado, 2002, p. 254). One of the used pooling methods is max pooling (Y. Zhou & Chellappa,

[1] http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

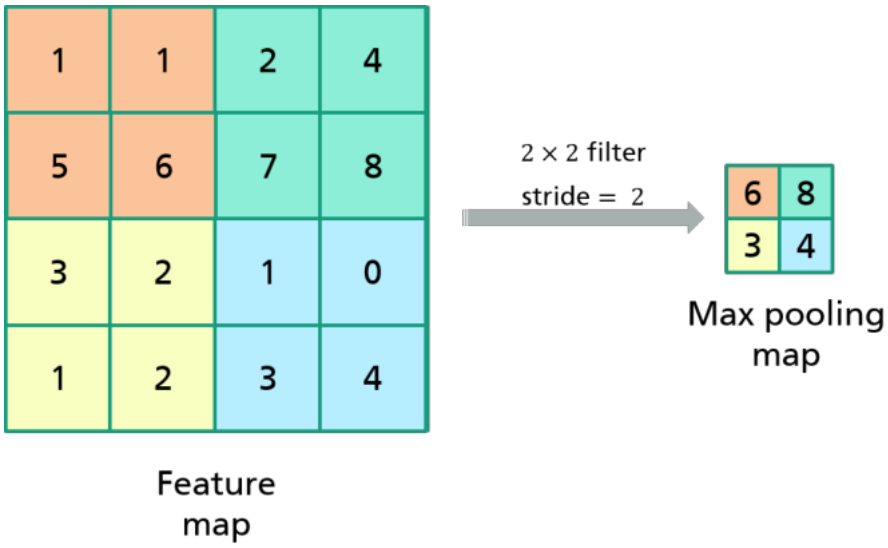


FIGURE 2.3: Max pooling: Using a 2×2 pooling filter with stride of 2 leads on the feature map (left) to a reduced max pooling map (right). The highest value for each region is used as input for the max pooling map (Graphic inspired by [2]).

1988). Here, the max values of different regions of the feature map are extracted and written into a max pooling map. The extracting of the max values is done by a filter which does not overlap regions. To guarantee a non-overlapping filter of size $z \times z$, a stride which is defined as $s = z$ is used (Krizhevsky et al., 2012). For example, to scan the feature map with a filter size of 2×2, a stride of 2 is necessary to ensure no overlapping regions (see Figure 2.3).

In a classical (deep) feedforward neural network, all layers are fully connected, which means that each neuron of the previous layer is connected to the following layer. In contrast, only the last layers in a CNN are fully connected layers. These layers are used to calculate the class score in a classification task.

Beside the architecture of a CNN, there are some relevant techniques which are important to make the network learn. According to Goodfellow et al. (2016), four things are essential to build a deep learning algorithm DL: specification of a dataset d, a cost function $cost\_func$, an optimization procedure opt, and a network model m (e.g., CNN). In a semi-formalized description it can be said:

$$DL(x) = d(x) + m(x) + cost\_func(m(x)) + opt(cost\_func(m(x))), \tag{2.3}$$

where the cost function is applied to the model and the optimization is used to minimize the cost function.

A cost function (sometimes referred to as 'error function' or 'loss function') is a function that quantifies the difference between the expected and actual outputs of a CNN. One cost function, used in deep learning for multi-class classification task is cross-entropy, formulated as

$$H(P, Q) = -\mathbb{E}_{x \sim P} \log Q(x) = -\sum_{x} P(x) \log Q(x) \tag{2.4}$$

[2] https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks

where P stands for the true distribution and Q stands for the distribution predicted by the model. The expectation of $f(x)$ with respect to $P(x)$ is denoted as $\mathbb{E}_{x\sim P}$.

To minimize the output of the cost function, optimization is needed. To optimize the layer weights during the training phase of a CNN and therefore to reduce the output of the cost function, backpropagation is used (Simonyan, Vedaldi, & Zisserman, 2014). The idea of backpropagation is not specific for CNNs and can also be used for calculation of other functions (Goodfellow et al., 2016). The choice of the backpropagation method depends on the used cost function and the used network model (LeCun, Bottou, Orr, & Müller, 1998). Backpropagation is used to calculate a gradient (Goodfellow et al., 2016). The gradient represents the rate at which the costs C change with respect to weights and biases [3]. Therefore, the gradient represents the direction of steepest change $\nabla C(x, y)$. The gradient is needed to apply a gradient-based optimization function to update the weights. In general, optimisation methods can be categorised based on whether fixed or adaptive learning rates are used. As an optimization method with fixed learning rates, stochastic gradient-descent (SGD) is nowadays often used (Goodfellow et al., 2016). A common approach using adaptive learning rate is the adaptive moment estimation (Adam), introduced by Kingma and Ba (2014).

Besides using pooling layers to reduce the danger of overfitting, regularization techniques are used to prevent the network model from adapting itself overly on the training set and performing poorly on unseen data. Goodfellow et al. (2016, p. 117) defined regularization as

> "any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error."

Two of the practically used regularization techniques are L_2 and dropout. L_2 is defined as

$$\lambda \sum_{i=1}^{k} w_i^2, \tag{2.5}$$

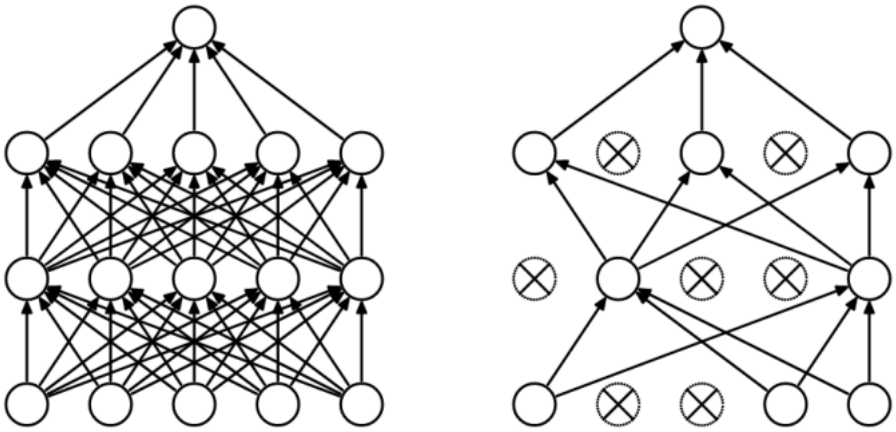
i.e. the sum of the squared weights w. λ serves in this formula as regularization rate and is therefore also called 'weight decay'. It reduces the weight vector by a constant factor (Goodfellow et al., 2016). If the values for λ are very high, the effect of regularization is very small. This leads to too much weight w being added, resulting in underfitting. When if λ is very small, the coefficients will shrink towards zero. This formula is added as a penalty to the cost function. Another approach to prevent overfitting is to use dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). The idea here is to drop out (randomly) neurons in a network, i.e. to remove the neuron with all of its incoming and outgoing connections (see Figure 2.4).

2.2.2 VGG Face

The VGG Face architecture, described by Parkhi et al. (2015), is a CNN which is specialized in face recognition. It is based on the architecture of the VGG (named after the research group which invented it: Visual Geometry Group[4]), which was first described by Simonyan and Zisserman (2014). VGG Face was trained on a collected

[3] The backpropagation algorithm is often misunderstood, because backpropagation does not represent the entire learning algorithm for the CNN. Instead, the gradient-based optimization method uses the gradient calculated by backpropagation for learning (Goodfellow et al., 2016).

[4] https://www.robots.ox.ac.uk/~vgg/



(a) Standard Neural Net (b) After applying dropout.

FIGURE 2.4: (a) shows a neural net consisting of two hidden layers with fully connected neurons. (b) shows the same network after using dropout as regularization technique which leads to a reduction of the complexity of the network (Graphic from Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov (2014)).

dataset of face images of celebrities. This dataset was created by using the Internet Movie Data Base to get a ranked celebrity list of actors. In the end, 1,635,159 images of 2622 actors were collected. The images contained profile shots as well as frontal shots. For testing, the Labelled Faces in the Wild (LFW) and YouTube Faces in the Wild (YTF) (Wolf, Hassner, & Maoz, 2011) dataset were used. LFW contains 13,233 images of 5749 persons. The YTF dataset contains 3425 videos of 1595 persons. The videos are from the video-sharing platform YouTube [5]. The VGG Face architecture consists of six different stages (see Figure 2.5). On the first stage, the input of the face images have a size of 224×224 pixels. The image size is reduced from stage to stage to in the end 7×7 pixels. The last stage of the VGG Face architecture includes three fully connected layers. The last fully connected layer consists of 2622 neurons. The high number of neurons in the last layer is explained by the fact that 2622 different prominent persons should be classified in the original use of the VGG Face model. The number of neurons in the last layer can be adjusted for the respective classification task. The activation function of the last layer is 'softmax' to scale the output values in a range between 0 and 1. In the other fully connected layers and in the convolutional layers, ReLU activation function is used.

2.2.3 Data Augmentation

One reason for the great success of CNNs lies in the efficient use of GPUs and ReLUs and the use of regularization techniques such as dropout (LeCun et al., 2015). Additional to these aspects, the larger the dataset, the more effective is the model training (Perez & Wang, 2017). Therefore, techniques to enlarge datasets and then use them for training the deep learning models also make an important contribution (LeCun et al., 2015). This techniques are summarized under the term data augmentation. By data augmentation, methods to change images using transformations are meant (Chatfield, Simonyan, Vedaldi, & Zisserman, 2014). These transformations do not change the underlying class and can therefore be used as a way to extend

[5] https://www.youtube.com

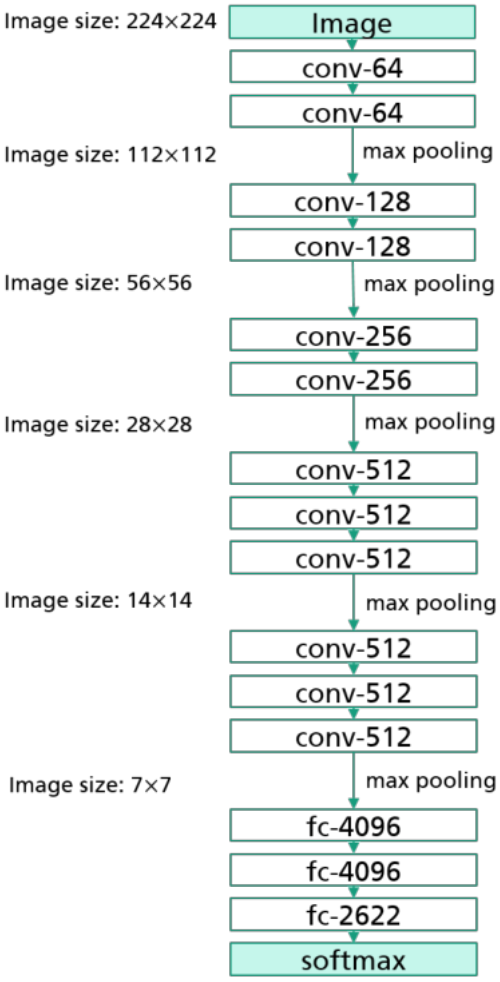


FIGURE 2.5: VGG Face architecture as described in Parkhi, Vedaldi, and Zisserman (2015).

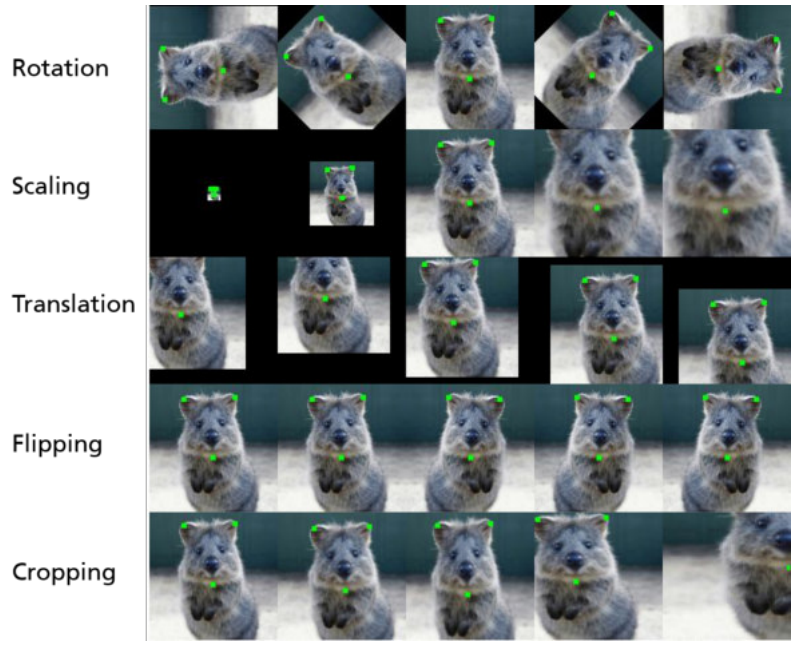


FIGURE 2.6: Influence of different data augmentation techniques on images (Graphic inspired by [6]).

datasets by adding further examples (Chatfield et al., 2014). Some of the common data augmentation techniques used in deep learning are rotation, scaling, translation, flipping and cropping (Chatfield et al., 2014; Hauberg, Freifeld, Larsen, Fisher, & Hansen, 2016) (see Figure 2.6).

2.3 Explainable Artificial Intelligence for Deep Learning Networks

Although deep learning models are used very successfully in various domains including image classification, one disadvantage of the methodology remains: due to the non-linear structure, deep learning is a black box, i.e. it is not comprehensible to people, especially laymen, how the trained network makes its decisions (Samek et al., 2017). This trade-off between accuracy and interpretability is a fundamental theme of all machine learning approaches. For example, it is shown that rule-based systems or expert systems are easy for humans to interpret, but are often not very accurate (Selvaraju et al., 2016). When CNNs are used, this interpretability is abandoned in favour of a higher accuracy, which is achieved by a stronger abstraction. Stronger abstraction means the addition of more than one layer to the network (Selvaraju et al., 2016). To compensate for the loss of interpretability, there is the possibility of using post-hoc methods for CNNs after a network has been trained, explaining "why they predict what they do" (Selvaraju et al., 2016, p. 2). Post-hoc methods focus more on the understanding of the results of a machine learning approach than to the understanding how a model works (Lipton, 2017). Six of these post-hoc approaches are presented in more detail in this master's thesis. First, six model-specific

[6] https://github.com/aleju/imgaug

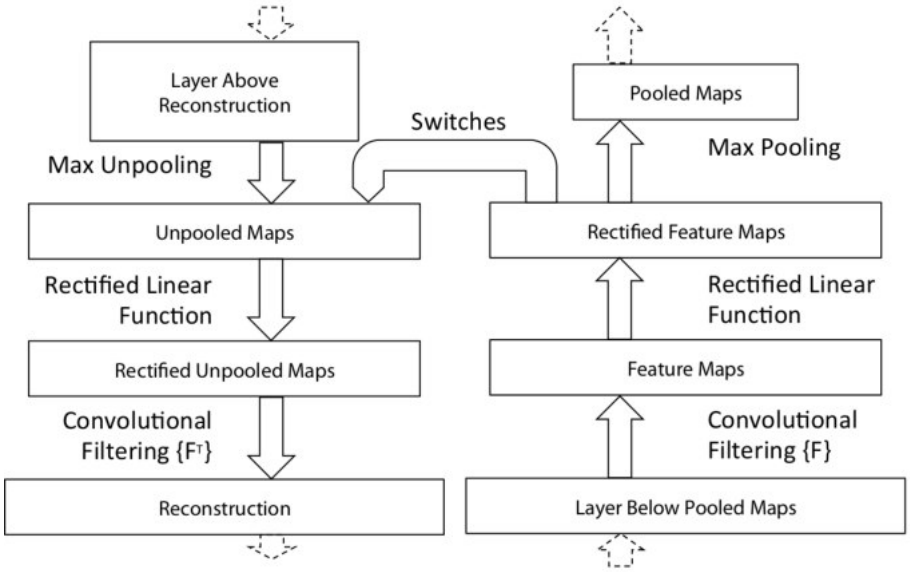


FIGURE 2.7: Structure of the deconvnet (left), which inverts the given structure of a CNN (right) (Graphic from Zeiler and Fergus (2014)).

approaches for deep learning networks: deconvnet, backpropagation, guided backpropagation, Grad-CAM, guided Grad-CAM, and Layer-wise relevance propagation (LRP) are presented. Following this, one model-agnostic approach, Local Interpretable Model-agnostic Explanations (LIME) is presented.

2.3.1 Deconvnet

The deconvnet approach was first used to perform supervised learning (Zeiler, Taylor, & Fergus, 2011). Zeiler and Fergus (2014) describe another way of usage of the deconvnet approach: to visualize the given activations of a feature map of CNNs. Here, deconvnet inverts the direction flow of a CNN (Springenberg, Dosovitskiy, Brox, & Riedmiller, 2014) (see Figure 2.7). For this, an input image is presented to the CNN to compute the features. To reconstruct a specific activation of a specific neuron in a specific layer, all other activations are set to zero in this layer of the CNN (see Figure 2.8). The resulting feature map is then given to the deconvnet. Three procedures, namely unpooling, rectification and filtering are applied on this given feature map. Unpooling is done using switches (see Figure 2.9). During the forward pass of the CNN, in the max pooling step the information about the location of the highest value is getting lost. The switches prevent this information loss by storing the locations of the max values. These switches are used by deconvnet to reconstruct these locations. As the CNN uses ReLU activations, in the step of rectification in the deconvnet the reconstructed map is also passed through a ReLU activation. This is formulated as

$$R_n = R_{n+1} \cdot (R_{n+1} > 0), \tag{2.6}$$

where R_n stands for the approximate feature map construction (Simonyan et al., 2014). In the last step of each deconvnet, filtering is applied. In the CNN, filters are used in the convolutional layers. To reconstruct the information using deconvnet, a transposed filter (flipping each filter vertically and horizontally) is used to 'unconvolve' the feature maps constructed by the CNN. The transposed filter is applied on the rectified maps constructed by the rectification step before. The deconvnet

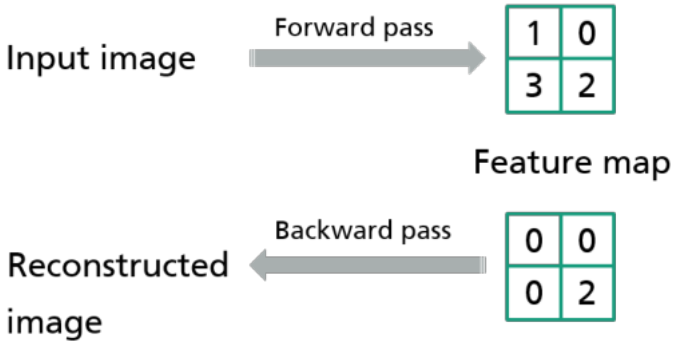


FIGURE 2.8: In the forward pass of a CNN a feature map is generated. To reconstruct the activation of a neuron, all neurons except one are set to zero before the backward pass using deconvnet is done (Graphic inspired by Springenberg, Dosovitskiy, Brox, and Riedmiller (2014)).

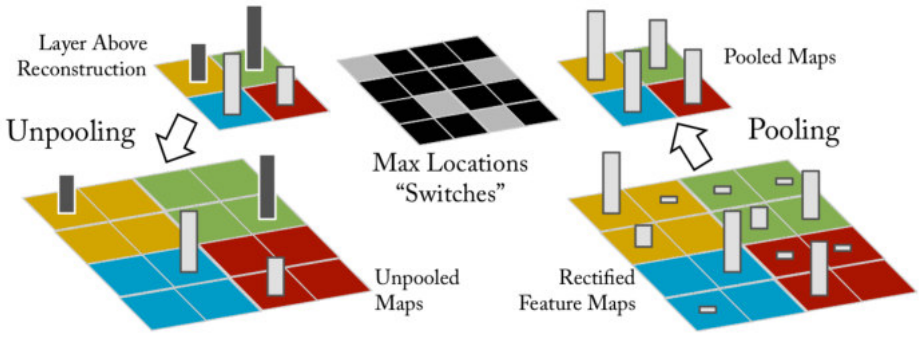


FIGURE 2.9: Before creating pooled maps in the CNN (right), switch variables including the location of the max values before the pooling are saved (middle). These switches are used by deconvnet to reconstruct the locations in the unpooling step (left) (Graphic from Zeiler and Fergus (2014)).

method is applied to every layer of in the CNN until the input pixels of the images are reached. The resulting map represents the parts of the image which activate the non-zero neuron at most.

2.3.2 Backpropagation

As Simonyan et al. (2014) stated, their backpropagation approach to visualize neuron activations in CNNs can be seen as a generalization of the deconvnet approach. In their approach, Simonyan et al. (2014) describe that two types of visualizations are possible by calculating the gradient of a class:

1. Generating an image that maximizes the class values
2. Generating a saliency map that refers to a specific image of a specific class

For this master's thesis, the second kind of visualization is used. Here, Simonyan et al. (2014) create saliency maps which are specific to the image, meaning that the saliency map highlights the areas of the given image that are discriminative for the given class. This is achieved by ranking the pixels of an image I_0. The ranking is based on the influence of the pixels on the class score function $S_c(I)$ of a class c of a CNN. The class score function $S_c(I)$ is a non-linear function of image I. To approximate the non-linear function $S_c(I)$ locally for I_0, a first-order Taylor expansion is performed:

$$S_c(I) \approx w^T I + b. \tag{2.7}$$

w represents the derivative of S_c, with respect to image I at point I_0:

$$w = \frac{\partial S_c}{\partial I}\bigg|_{I_0}. \tag{2.8}$$

Simonyan et al. (2014) mentioned that the derivative of the class score can be interpreted as the magnitude in which pixels needs to be changed to affect the class score $S_c(I)$ the most. These pixels can then be considered relevant for the object localization in the image. The procedure is very similar to the deconvnet approach (see Figure 2.7). The difference to deconvnet is the use of the ReLU function. For the ReLU rectification layers $X_{n+1} = max(X_n, 0)$, the sub-gradient looks like this:

$$R_n = R_{n+1} \cdot (X_{n+1} > 0), \tag{2.9}$$

where R_{n+1} is $\frac{\partial f}{\partial X_{n+1}}$. To represent the saliency map $M \in R^{m \times n}$ (m rows and n columns) for an image I_0 of a class c, the derivative w is found using backpropagation. Then the saliency map is generated by rearranging the elements of the vector w. When using RGB images, as this master's thesis does, the maximum magnitude of w across all colour channels is used. Formally, this is described by

$$M_{ij} = max|w_{h(i,j,c)}|, \tag{2.10}$$

where c stands for the colour channel of the pixel (i, j) of an image I. The index $h(i, j, c)$ refers to the element of w that corresponds to colour channel c of pixel (i, j). The saliency map is visualized for the highest scoring class (Simonyan et al., 2014) (see Figure 2.10). The computation to create an image-specific saliency map is not time consuming, because only one backpropagation pass is required (Simonyan et al., 2014).

FIGURE 2.10: Backpropagation: Image specific saliency map for the highest scoring class (Graphic from Simonyan, Vedaldi, and Zisserman (2014)).

2.3.3 Guided Backpropagation

An alternative approach that combines the idea of backpropagation and deconvnet is called guided backpropagation (Springenberg et al., 2014). Springenberg et al. (2014) experimented with new CNN structures without max pooling. To get an impression of how their new CNN worked, they used deconvnet to visualize the feature maps of different layers. When doing so, they determined that the deconvnet approach does not work well on CNNs without max pooling, because one important step in the deconvnet process is the unpooling of the feature map. Therefore, Springenberg et al. (2014) worked out another approach to visualize the predictions of a CNN which works without max pooling. Guided backpropagation also uses a ReLU function. Here, the negative gradients from the top layer (as used in deconvnet) and the negative gradients of the bottom layer (as used in backpropagation) are used by guided backpropagation to mask out all these negative values. This combination is formulated as:

$$R_n = R_{n+1} \cdot (X_{n+1} > 0) \cdot (R_{n+1} > 0). \tag{2.11}$$

Therefore, the guided backpropagation method only uses positive values for positive activations. An overview about the differences of the ReLU functions of the backward step used in the approaches of deconvnet, backpropagation, and guided backpropagation is given in Figure 2.11.

2.3.4 CAM and (Guided) Grad-CAM

Whereas the deconvnet, backpropagation, and guided backpropagation methods (Simonyan et al., 2014; Springenberg et al., 2014; Zeiler & Fergus, 2014) ignore the fully-connected layer of a deep learning network, CAM uses also these layers and thus give a complete view of the whole CNN (B. Zhou, Khosla, Lapedriza, Oliva, & Torralba, 2016). CAM stands for Class Activation Mapping (B. Zhou et al., 2016). CAM uses global average pooling, which was described by Lin et al. (2014), who uses them in their novel deep learning structure. The global average pooling allows

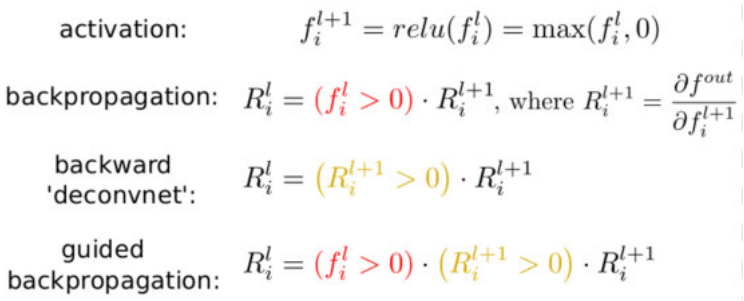


FIGURE 2.11: Different ReLU functions used by deconvnet, backpropagation and guided backpropagation method (Graphic from Springenberg, Dosovitskiy, Brox, and Riedmiller (2014)).

it to identify exactly the regions of an image which are important for class discrimination (B. Zhou et al., 2016). The main technique used by CAM is called class activation mapping (B. Zhou et al., 2016). Here, global average pooling is used on the last convolutional feature map. The features generated this way are the input for the fully connected layers which then create the classification output. In the case of a softmax activation in the last fully connected layer of a CNN, the activation of unit k is represented as $f_k(x, y)$ at a spatial location (x, y). For unit k, the result of the global average pooling F_k is $\sum_{x,y} f_k(x, y)$. The input for a class c in the softmax activation S_c is $\sum_k w_k^c F_k$, where w_k^c is the weight for class c in unit k. This weight is important, because it is the indicator for the importance of F_k for class c. The bias term is ignored in this formula. Comparable to Equation 2.2, the output P_c of the softmax activation for class c is

$$P_c = \frac{exp(S_c)}{\sum_c exp(S_c)}. \tag{2.12}$$

Now the CAM M_c for class c is a combination of the global average pooling F_k and the input to the softmax S_c and can defined for each spatial element as the following:

$$M_c(x, y) = \sum_k w_k^c f_k(x, y). \tag{2.13}$$

S_c can be seen as $\sum_{x,y} M_c(x, y)$. Therefore, $M_c(x, y)$ can be interpreted as the importance of the activation at spatial grid (x, y) which leads to the classification c. In other words, the CAM is a weighted linear sum of visual patterns at different spatial locations (for a resulting example visualization see Figure 2.12). Grad-CAM is a generalization of the CAM approach (Selvaraju et al., 2016). While the use of CAM is limited to a few specific CNNs, Grad-CAM can be applied to any CNN model. In summary, the calculation of the score y^c for each class for CAM can be formulated as follows (Selvaraju et al., 2016):

$$y^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z} \sum_i \sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}}. \tag{2.14}$$

This results in the feature map $L_{CAM}^c \in \mathbb{R}^{u \times v}$ (width u and height v) for a class c:

$$L_{\text{CAM}}^c = \sum_k w_k^c A^k \tag{2.15}$$

FIGURE 2.12: Highlighting of the important discriminative image regions to classify a briard using CAM (Graphic from B. Zhou, Khosla, Lapedriza, Oliva, and Torralba (2016)).

For Grad-CAM, the weights α_k^c are calculated as follows:

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backpropagation, i.e: } y_c} . \tag{2.16}$$

This changes the feature map generated by Grad-CAM to:

$$L^c_{\text{Grad-CAM}} = ReLU\left(\sum_k \alpha_k^c A^k\right) \tag{2.17}$$

Different to deconvnet (Zeiler & Fergus, 2014), backpropagation (Simonyan et al., 2014), and guided backpropagation (Springenberg et al., 2014), the approach of guided Grad-CAM (Selvaraju et al., 2016) not only highlights fine-grained details in the image but are also class discriminative (see Figure 2.13). To achieve this, the heatmaps generated with Grad-CAM and backpropagation are multiplied pointwise (Selvaraju et al., 2016).

2.3.5 Layer-wise Relevance Propagation

The LRP method, introduced by Bach et al. (2015) is another model-specific XAI approach for deep learning networks. It uses pixel-wise decomposition as its main concept, combined with layer-wise relevance propagation (Bach et al., 2015). The general idea of pixel-wise decomposition is to look at the impact of each input pixel $x_{(d)}$ of an input image x to the prediction $f(x)$. One possibility to do that is to segment (=decompose) the prediction $f(x)$ as the sum of the terms of the input dimensions, notated as:

$$f(x) \approx \sum_{d=1}^{V} R_d. \tag{2.18}$$

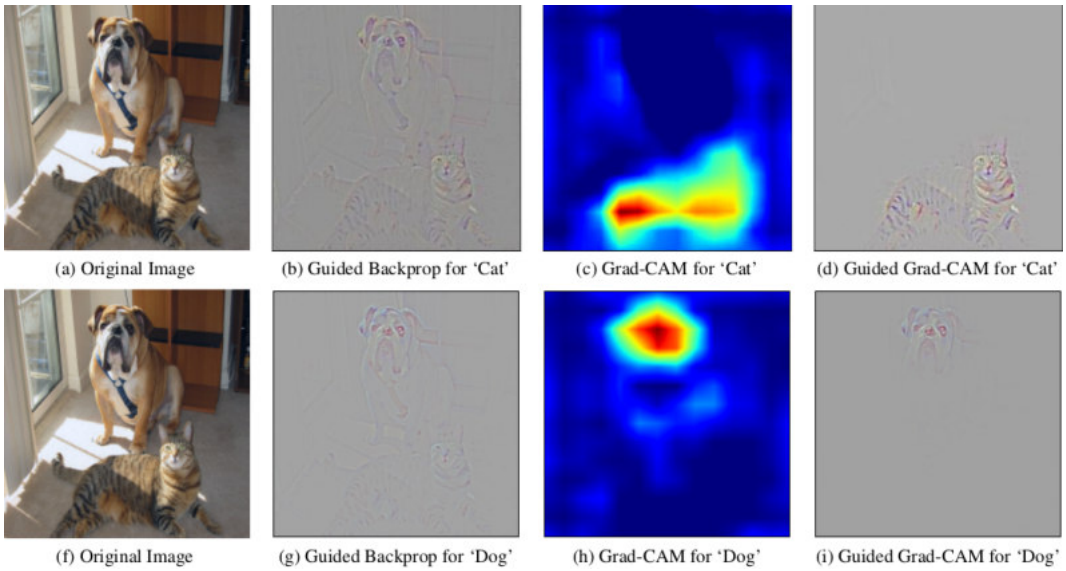


FIGURE 2.13: The first row shows the visualization for the class 'cat', the second row the visualizations for the class 'dog'. Original image (first column). Improvement in class discrimination using Grad-CAM (third column) instead of guided backpropagation (second column). Improvement of class discrimination and resolution using Guided Grad-CAM (fourth column) (Graphic from Simonyan, Vedaldi, and Zisserman (2014)).

$R_d < 0$ can be interpreted as evidence against the structure which should be classified, and $R_d > 0$ otherwise. The resulting Relevance R_d for each input pixel $x_{(d)}$ can be visualized in a heatmap by mapping every R_d to a colour space (Bach et al., 2015). LRP is an approach to achieve a pixel-wise decomposition as denoted in Equation 2.18. Here it is important to mention that LRP is not an algorithm which can be applied on neural networks. Instead, LRP defines constraints which have to be fulfilled when trying to calculate the importance of pixels to a classification result of a neural network (Bach et al., 2015). These constraints are described in Equation 2.23 and Equation 2.24. Before the LRP approach for the entire network is explained, the function of LRP on a single neuron j is described (Lapuschkin, Binder, Müller, & Samek, 2017): A neuron j gets a relevance score R_j from the higher layer. This relevance score is distributed proportionally to the contribution of the input neurons i of the neuron j. The distribution to i is based on the contribution of the i neurons in the forward pass:

$$R_{i \leftarrow j} = \frac{z_{ij}}{z_j} R_j. \tag{2.19}$$

z_{ij} is measuring the contribution of neuron i to the activation of neuron j. z_j represents the aggregation of all forward messages z_{ij} over i at j. The relevance value R_i is defined by all incoming relevance values, $R_{i \leftarrow j}$ of the neurons j in which i is involved:

$$R_i = \sum_j R_{i \leftarrow j}. \tag{2.20}$$

Therefore, the following local conservation property is given:

$$R_i = \sum_j R_{i \leftarrow j} \quad \text{and} \sum_i R_{i \leftarrow j} = R_j. \tag{2.21}$$

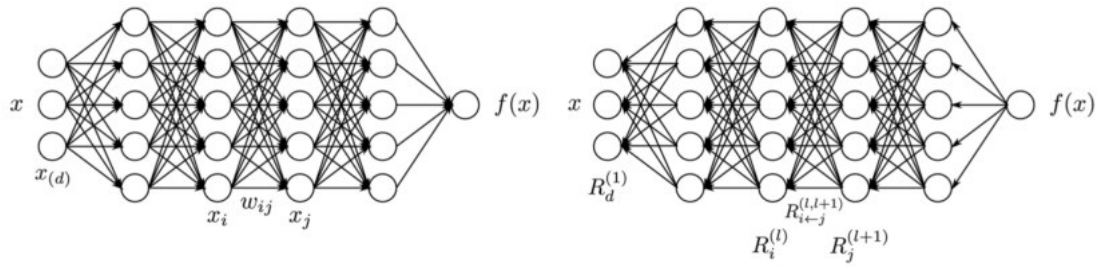


FIGURE 2.14: Forward pass in a multilayer neural network which results in a classification $f(x)$ (left). Calculating the layerwise relevance $R_{i\leftarrow j}^{(l,l+1)}$ between a neuron i and a neuron j using the sink neuron $R_j^{(l+1)}$ and a source neuron $R_i^{(l)}$ (right) (Graphic from Bach et al. (2015)).

With these formulas, it is possible to calculate the importance of pixels to a classification result of a neural network (Bach et al., 2015):

$$f(x) = \cdots = \sum_{d\in l+1} R_d^{(l+1)} = \sum_{d\in l} R_d^{(l)} = \cdots = \sum_{d\in l} R_d^{(1)}, \tag{2.22}$$

where $R_d^{(l+1)}$ stands for the relevance score for each dimension $z_d^{(l+1)}$ of the layer $l+1$, modeled by the vector z. The last layer is represented as $f(x)$ and the first layer of the network as $R_d^{(1)}$. The relevance of each neuron of the network except the last neurons (output neurons) is the first constraint of LRP. This first constraint is defined as:

$$R_i^{(l)} = \sum_{\text{k: i is input for neuron k}} R_{i\leftarrow k}^{(l,l+1)}. \tag{2.23}$$

It should be noted that the term 'input' refers to the direction during classification, i.e. from a previous layer to a subsequent layer. The second constraint for LRP is defined as:

$$R_k^{(l+1)} = \sum_{\text{i: i is input for neuron k}} R_{i\leftarrow k}^{(l,l+1)}, \tag{2.24}$$

which represents the sum over the sources at layer l for a fixed neuron k at layer $l+1$. In comparison, Equation 2.23 represents the sum over the sinks at layer $l+1$ for a fixed neuron i at a layer l. A visualization of the important components of LRP is displayed in Figure 2.14. The neuron activation of x_j represents a non-linear function of z_j. The pre-activations z_{ij} measure the relative contribution of each neuron x_i to R_j. The relevance decomposition, based on the local and global pre-activations is denoted as:

$$R_{i\leftarrow j}^{(l,l+1)} = \frac{z_{ij}}{z_j} \cdot R_j^{(l+1)} \tag{2.25}$$

A disadvantage of equation 2.25 is that for small z_j, relevance values $R_{i\leftarrow j}$ can take on unbounded values. To counteract this, a stabilizer $\varepsilon \geq 0$ can be used. The Equation 2.25 can be adjusted as follows:

$$R_{i\leftarrow j}^{(l,l+1)} = \begin{cases} \frac{z_{ij}}{z_j+\varepsilon} \cdot R_j^{(l+1)} & z_j \geq 0 \\ \frac{z_{ij}}{z_j-\varepsilon} \cdot R_j^{(l+1)} & z_j < 0 \end{cases} \tag{2.26}$$

In practice, the LRP method is often stabilized using a $\alpha\beta$-rule. It is possible to define specific values for α and β. The Equation 2.25 then changes to:

$$R_{i\leftarrow j}^{(l,l+1)} = R_j^{(l+1)} \cdot \left(\alpha \cdot \frac{z_{ij}^+}{z_j^+} + \beta \cdot \frac{z_{ij}^-}{z_j^-} \right) \tag{2.27}$$

The usage of α and β have the advantage through the stabilizing effect that they make it possible to visualize not only positive but also negative activations of pixels. The strength of the influence of negative and positive portions can be controlled with the choice of the respective α and β value (Bach et al., 2015; Montavon, Samek, & Müller, 2017). Besides these parameters, Kohlbrenner (2017) showed that a 'preset' variant of the LRP algorithm achieves optimal results in the calculation of relevance maps. Using the preset approach, the relevance scores R_j for all neurons of the lowest (first) layer are uniformly distributed to the input neuron instead of using the $\alpha\beta$ values (Lapuschkin et al., 2017). To control the resolution of the heatmaps generated by LRP, Bach, Binder, Müller, and Samek (2016) describe an approach of a 'mapping influence cut-off point'. This point describes the moment from which the forward mapping function of the classifier no longer influences relevance propagation, since only the receptive field of the classifier is relevant. The cut-off at this point is called the 'flat' rule.

2.3.6 Local Interpretable Model-Agnostic Explanations

The LIME method (Ribeiro, Singh, & Guestrin, 2016) uses local predictions to learn an interpretable model. The approach of Ribeiro et al. (2016) is a model-agnostic approach, which means it can provide explanations for the predictions of any classifier (e.g., decision trees, CNNs, linear models). With LIME and a variation of it, SP-LIME, explanations for the prediction of a model can be generated for specific images and the focus can also be laid on the model as a whole. In this master's thesis, the focus lays on the explanation of specific images. In the following paragraphs, the LIME approach for that case will be described in more detail.

To get an interpretable representation of the data, a simplification is needed. For the simplification in image classification, the original representation of an instance which should be explained, denoted as $x \in \mathbb{R}$ is represented as an interpretable representation in the form of a binary vector $x' \in \{0,1\}^{d'}$. Ribeiro et al. (2016) describe an explanation as a model $g \in G$, where G represents different kinds of interpretable models. In general, the explanation calculated by LIME looks like the following:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \{ \mathcal{L}(f, g, \pi_x) + \Omega(g) \}, \tag{2.28}$$

where $\Omega(g)$ stands for the complexity of the explanation $g \in G$. For example, using CNNs, the complexity measure is the number of non-zero weights. The complexity measure is the counterpart of interpretability, meaning the more complex an explanation is, the less interpretable it is by humans. The model being explained is denoted as $f : \mathbb{R}^d \rightarrow \mathbb{R}$. For multiple classification tasks, $f(x)$ represents the probability that x belongs to the relevant class. $\pi_x(z)$ serves as a proximity measure between a distance z and x and represents the locality. $\mathcal{L}(f, g, \pi_x)$ expresses the unfaithfulness of g in the approximation of f depending on the locality, given by π_x. The focus lays on two parts: to minimize $\mathcal{L}(f, g, \pi_x)$ to guarantee a local fidelity and to hold $\Omega(g)$ low to get a result which is still interpretable by humans. $\mathcal{L}(f, g, \pi_x)$ is approximated using samples which are weighted by π_x.

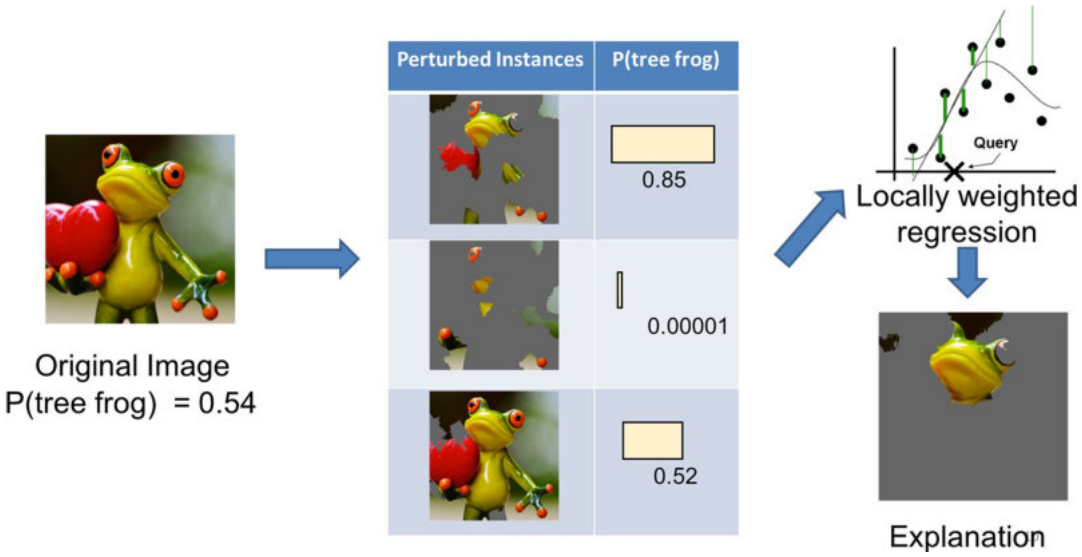


FIGURE 2.15: Different steps of the LIME procedure to get an interpretable model of the complex CNN image classifier.[7]

When using LIME for CNN image classifiers, the following steps are passed (see Figure 2.15):

1. The original image which is used for the prediction by the CNN is divided into super-pixels.

2. After that, the original image is perturbed into sample instances by switching some super-pixels off. The L2 distance between the original image and the perturbed image is calculated and used later as weights π_x for the explanation model.

3. The created sample instances are used as input for the CNN image classifier. The classifier then calculates a prediction for each of the perturbed images.

4. Extraction of K features (super-pixels) of the CNN image classifier which creating the maximum likelihood for the class which was predicted by the CNN. The selection of the K features is done using a variant (Efron, Hastie, Johnstone, & Tibshirani, 2004) of the Lasso algorithm from Tibshirani (1996). K stands for the amount of features which should be extracted. Here, any number can be used, but a higher value means more complexity (Efron et al., 2004). The weights for the K features are then learned using least squares method. The combination of Lasso with K features is named K-LASSO by Ribeiro et al. (2016).

5. The resulting relevant super-pixels can then be displayed on the image. The unrelevant super-pixels are grayed out.

[7] https://www.oreilly.com/learning/introduction-to-local-interpretable-model-agnostic-explanations-lime

Chapter 3

Research Questions

The meta-analysis of Lench, Flores, and Bench (2011) showed that people find it difficult to distinguish between different negative emotions, while the distinction between positive and negative emotions is easier for humans. In the recognition of emotions and pain by humans, the context, for example in the form of body movements, plays a role (Aviezer et al., 2012). In addition, indirect (e.g., rubbing a hurted part of the body) and direct expression (e.g., vocalization) of pain behaviour is combined to communicate pain (Keefe & Wren, 2013). If the context and the direct or indirect behaviour is omitted, it is very difficult for people to recognize emotions (Aviezer et al., 2012; Brahnam et al., 2006). The omission can be due to reduced ability of movement and expression due to illness or due to limited verbal communication skills, as for example in children. Video-based pain recognition could serve as a supplement to self-reported measurements of pain in these scenarios (Kunz et al., 2017). In this master's thesis, the video-based material of the BioVid dataset (Walter et al., 2013; Werner et al., 2013) is analysed using a fine tuned VGG Face CNN. On this model, different XAI methods are applied to create a post-hoc interpretability (Montavon et al., 2017). Referring to Montavon et al. (2017), there a distinction to make between the words 'interpretation' and 'explanation'. Montavon et al. (2017, p. 2) define interpretation as "the mapping of an abstract concept (e.g., a predicted class) into a domain that the human can make sense of." In this master's thesis, the interpretable domain are the images. Explanation is definded by Montavon et al. (2017, p. 2) as "the collection of features of the interpretable domain, that have contributed for a given example to produce a decision (e.g., classification)." In this master's thesis, an explanation is given by the heatmaps of the images, generated by the different XAI methods. In general, the goal is to find and visualize good explanations for the decisions of a network.

The three central questions to which this master's thesis would like to provide answers, are:

1. **Predictive performance**: How well can facial expressions of pain be automatically distinguished from those of disgust and happiness using self-learned spatial features?

2. **Decision interpretation**: How can the decisions made by the model be presented to people in a comprehensible and transparent way?

3. **Feature explanation**: How do the self-learned features differ for the facial expressions of pain and those of disgust and happiness?

Chapter 4

Material & Procedure

The process of carrying out this master's thesis consisted of three steps: the selection and preparation of the dataset, the training of the CNN network and the application of various XAI techniques for the trained CNN (see Figure 4.1). These steps are described in detail in this chapter.

4.1 Material

4.1.1 BioVid Dataset

For training, validating, and testing the deep learning model, the BioVid Heat Pain Dataset (BioVid) (Walter et al., 2013; Werner et al., 2013) was used. It contains data from 90 participants from three age groups (18-35, 36-50 and 51-65 years). Each of these age groups consisted of 15 women and 15 men. The dataset has five parts [1]:

1. Part A (Pain stimulation without facial EMG - short time windows)
2. Part B (Pain stimulation with facial EMG - partially occluded face, short time windows)
3. Part C (Pain stimulation without facial EMG - long videos)
4. Part D (Posed pain & basic emotions)
5. Part E (Emotion elicitation with video clips)

Part A consisted of 5.5 seconds long videos each. Part D consisted of 1 minute videos each. In the videos of part A, pain elicitation was done by induce heat using a thermode at the right arm (Walter et al., 2013). An individual pain threshold was determined and from it four pain intensities were derived (Walter et al., 2013). In the videos of part D, the emotions were induced (Walter et al., 2013) by showing people images from the International Affective Picture System (IAPS) (Lang, Bradley, & Cuthbert, 1997). For the emotion induction seven images with positive or negative valence and with high or low arousal were used. Therefore, 28 images were used in total.

Extracted pain images with a pain intensity of 3 and 4 from part A and extracted disgust and happiness images from part D were used for this master's thesis (see Table 4.1).

[1] http://www.iikt.ovgu.de/BioVid.html

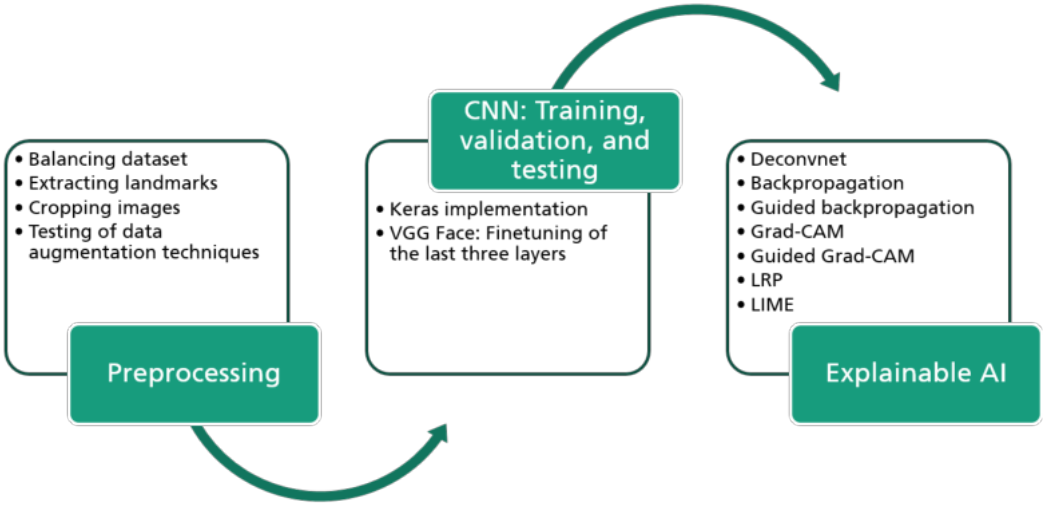


FIGURE 4.1: Overview of the procedure followed in this master's thesis, starting with preprocessing the BioVid dataset, followed by the training, validation, and testing of the CNN and then the application of different explainable AI techniques to visualize and explain the predictions of the CNN.

TABLE 4.1: Extracted BioVid data before balancing and data cleaning steps.

Part	**Name**	**Subjects**	**Frames**
Part A	Pain intensity 3	87	12,006
	Pain intensity 4	87	12,006
Part D	Disgust	76	114,076
	Happiness	75	112,575

TABLE 4.2: Extracted BioVid data after balancing and data cleaning steps.

Part	Name	Subjects	Frames
Part A	Pain intensity 3	87	12,006
	Pain intensity 4	87	12,006
Part D	Disgust	75	24,075
	Happiness	75	24,075

FIGURE 4.2: Example image of the BioVid dataset: image before cropping (left) and image after cropping (right) using 68 landmarks.

4.1.2 Data Preparation

Since the dataset was unbalanced between happiness, disgust, and pain, it was balanced by manually selecting 3×107 frames per category per subject in which the subject showed the emotions disgust and happiness. To ensure the variability of the selected frames, the 1 minute videos were divided into three time slots and 107 frames were extracted per time slot. During the manual selection of the frames, it was found that the subject 08311Zw55 moved her head massively in the disgust video. She turned away from the camera and talked to the study leaders and showed no disgust expression. This subject was therefore excluded from the dataset. After the balancing and data cleaning step, 24,012 frames for pain, 24,075 frames for disgust, and 24,075 frames for happiness (see Table 4.2) remained. After balancing the datatset, the BioVid images were cropped using 68 landmarks (see Figure 4.2). The landmarks were generated using the Dlib C++ library (King, 2009). The images were saved with a resolution of 256×256 pixels. As a preprocessing step, the data augmentation technique of flipping was used. This enlarged the dataset and was applied randomly during the training and validating phases of the CNN.

4.2 Implementation of CNN

To train the CNN, a VGG Face network was created with the deep learning library Keras (version 2.2.0). The implementation was realized in Python (version 3.5) and Tensorflow (version 1.5.0). The network was fine-tuned on BioVid dataset for the 3-class classification problem. Thereby, only the weights in the three final fully connected layers were updated. In each of the 5 runs, a different fold was reserved for testing and the remaining 4 folds were used for training. 20% of the training data were used for validation. In order to improve the performance of the CNN, various parameter optimizations were tried out. More details can be found in the appendix (see Appendix A).

4.3 Implementation of Explainable AI Methods

Altogether, seven different XAI approaches were implemented using Python 3.5 and different libraries for TensorFlow 1.8 and Keras 2.2.0. These approaches were: deconvnet, backpropagation, guided backpropagation, Grad-CAM, guided Grad-CAM, LRP, and LIME. For deconvnet, backpropagation and LRP, the Git Hub code from Lapuschkin, Alber, Hägele, Schütt, and Binder (2018) was used and adapted for the requirements of this master's thesis. For guided backpropagation, Grad-CAM and guided Grad-CAM, the Git Hub code from Petsiuk (2018) was adapted. For LIME, the Git Hub code of Ribeiro, Sameer, and Guestrin (2017) was used.

Chapter 5

Results

In this chapter, the results of this master's thesis are presented. First, the performance results of the CNN are presented, followed by the visualizations and a detailed description of different XAI methods.

5.1 Classification Results

For the XAI methods, the best performing VGG Face model (fold 5) of the 5-fold cross-validation was used (see Table 5.1). The loss and accuracy shown in Table 5.1 are rounded to two and three decimal places, respectively. The 5-fold cross-validation has an overall accuracy of 0.593 and an overall loss of 2.24 on the testing set. In the best model, the image pixel values were rescaled between 0-1 and an average-centering for each colour channel was applied. As hyperparameters, ADAM optimizer was used with a learning rate of 0.00001, categorical entropy was used as the loss function and L2 regularization with a regularization constant of 0.0001 was used to reduce overfitting. The end of the learning process was determined by the early stopping method on the validation set. If the loss stopped improving during an epoch, i.e., stopped going down any further, the training stopped after this epoch. The number of epochs required is displayed in Table 5.1. For testing the fold 5 model, 14,322 images were used. 4692 of these images belong to the class 'pain', 4815 images belong to the class 'disgust' and 4815 images belong to the class 'happiness' (see Table 5.2). The loss and accuracy values during training are visualized in Figure 5.1. As displayed in the confusion matrix (see Figure 5.2), the model could classify disgust images most often correctly, but it has especially problems in classifying happy images as happy. For 1356 happy images, happiness was wrongly classified as pain. In other words: only 57% of the happy images were classified as

TABLE 5.1: Results of the 5-fold cross-validation of the best performing CNN.

Fold	**Training**		**Validation**		**Testing**		**Epochs**
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	
1	0.86	0.998	2.08	0.623	2.34	0.537	2
2	0.85	0.998	2.11	0.608	2.17	0.610	2
3	0.85	0.998	1.99	0.637	2.28	0.592	2
4	0.86	0.997	2.18	0.600	2.85	0.562	2
5	0.55	0.999	1.81	0.625	1.58	0.665	4
Average					*2.24*	*0.593*	

TABLE 5.2: Results of the confusion matrix of fold 5.

	Precision	**Recall**	**F1-score**	**Images**
Pain	0.62	0.69	0.66	4692
Disgust	0.70	0.73	0.71	4815
Happiness	0.67	0.57	0.62	4815
Average/Total	0.67	0.66	0.66	14322

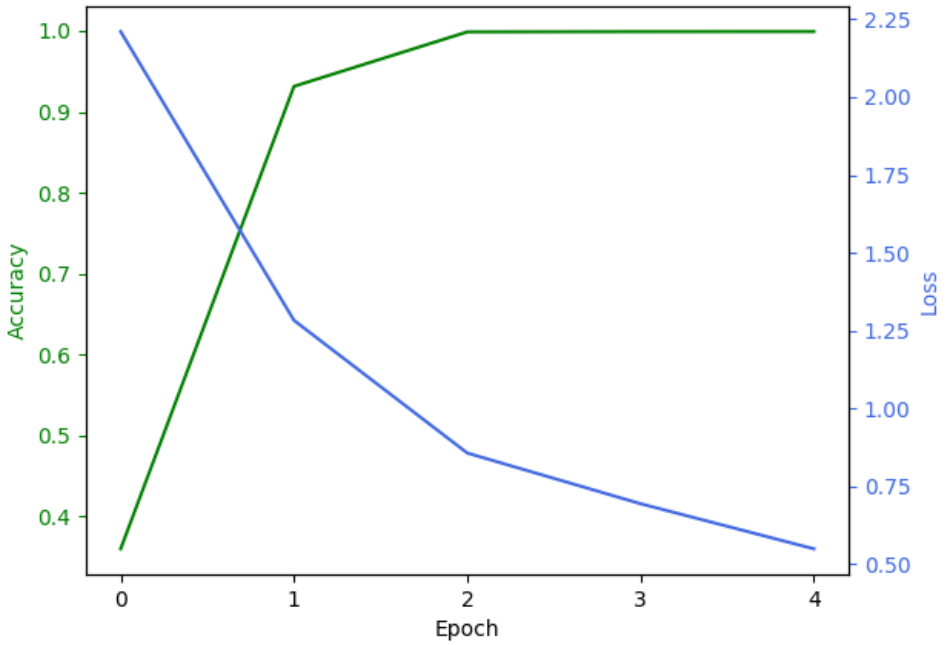


FIGURE 5.1: Training accuracy and training loss of fold 5.

happy (see Figure 5.3). The second most misclassified images were pain. Here, 17% of the pain images were classified as disgust (see Figure 5.3). In Figure 5.4 and Figure 5.5, two examples from the test fold are displayed. The first label above the images refers to the true class, the second label refers to the predicted class. In the original images (see Figure 5.4, (top, left)), the facial reaction to the pain intensities 3 and 4 are expressed. While the classifier wrongly predicts 'happiness' at pain intensity 3, it classifies 'pain' correctly at pain intensity 4. The visualizations by the XAI methods show that the difference between features relevant for happiness and pain are really small. It seems that especially small changes in the part around the eyes and nose are important for the classifier. When predicting pain, the part between the eyes, above the nose are important. For happiness prediction, the focus shifts towards the eyes.

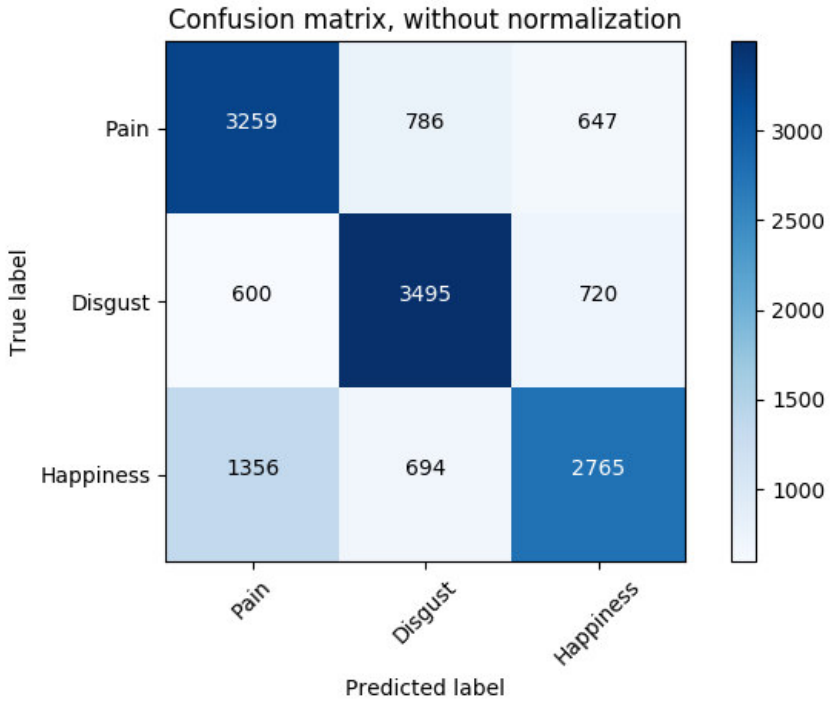


FIGURE 5.2: Confusion matrix of fold 5.

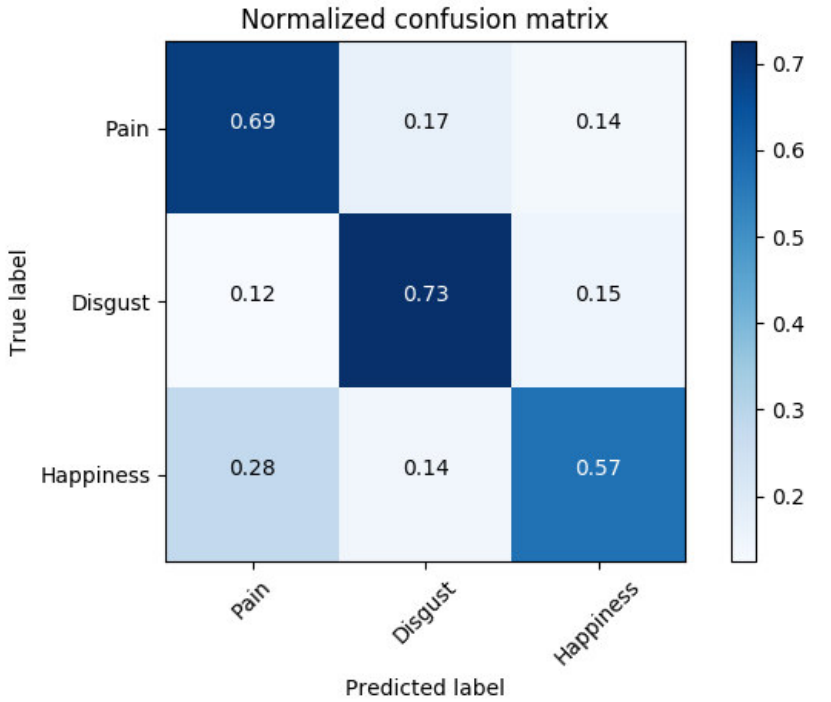


FIGURE 5.3: Normalized confusion matrix of fold 5.

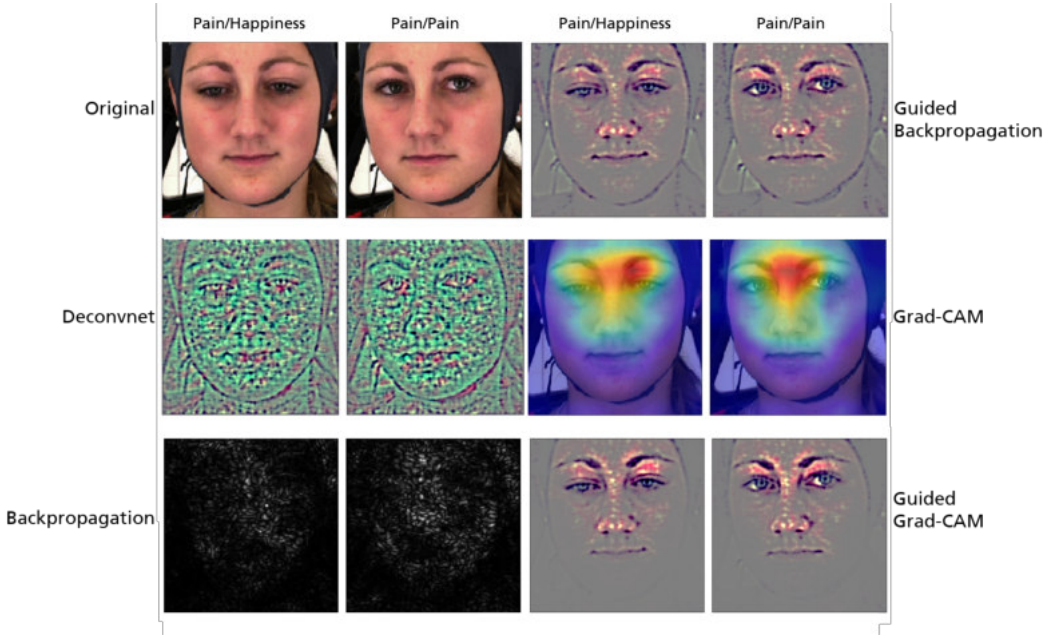


FIGURE 5.4: Misclassification of happiness in pain related images. Features and decisions explained using the methods deconvnet, (guided) backpropagation, and (guided) Grad-CAM. Original images with pain intensity 3 (top row, first image from left), and pain intensity 4 (top row, second image from left).

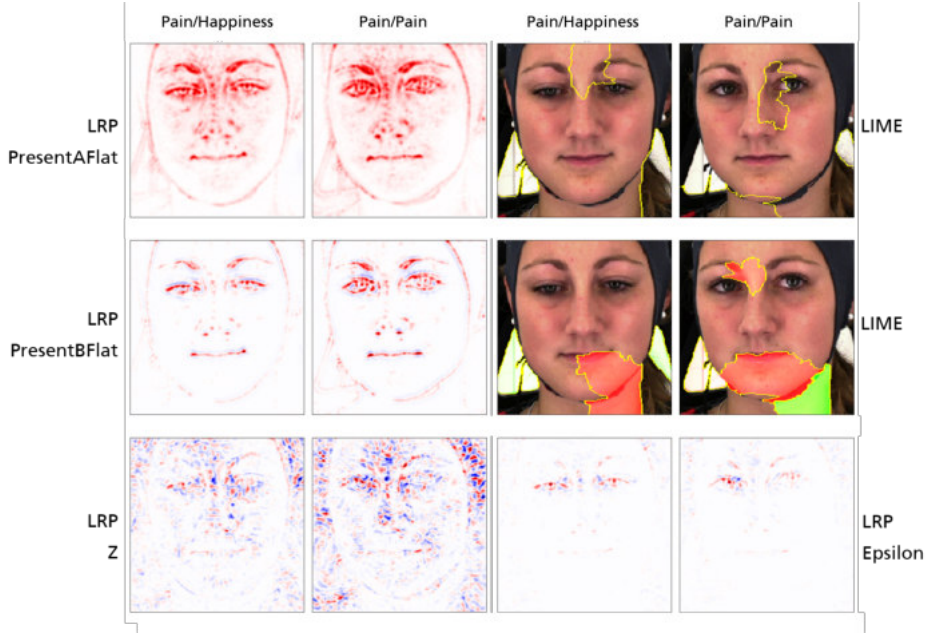


FIGURE 5.5: Misclassification of pain as happiness in pain related images. Features and decisions explained using the methods LRP and LIME. Original images as in Figure 5.4.

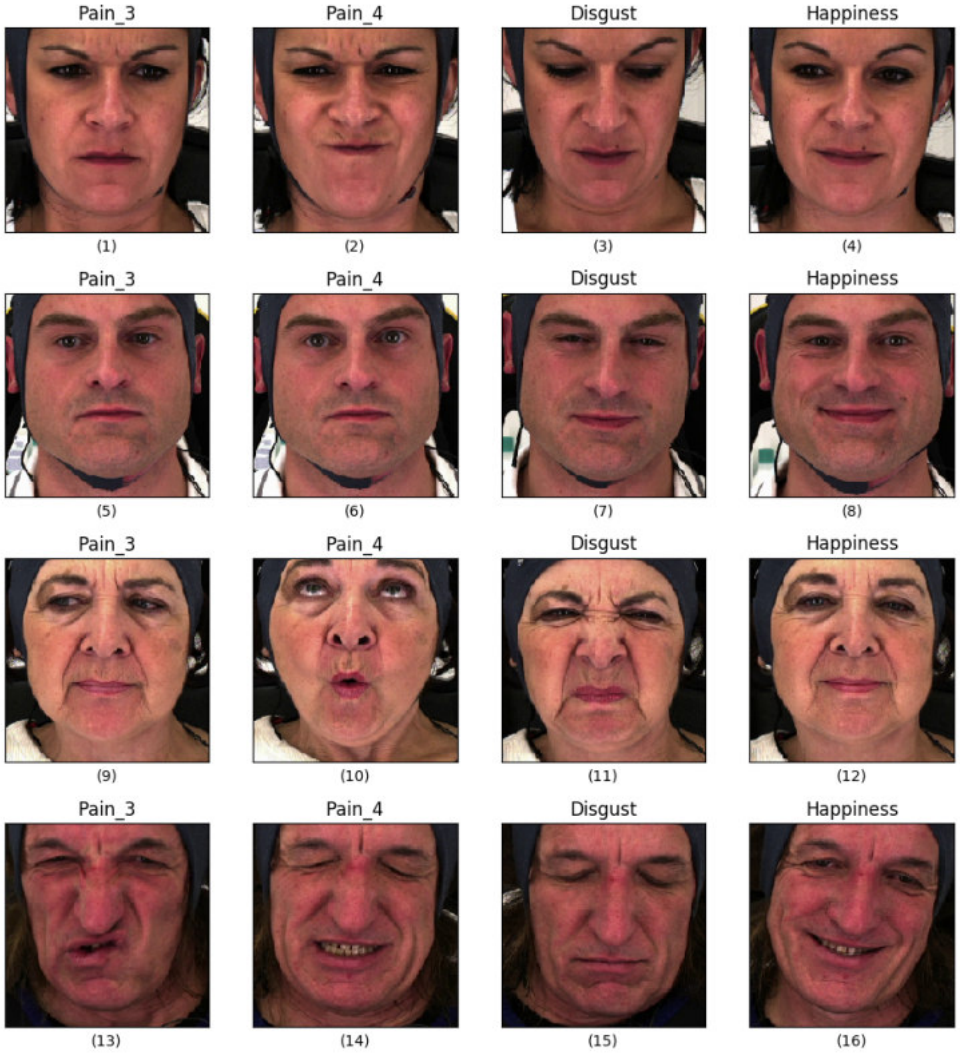


FIGURE 5.6: Images of 4 subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness.

5.2 Results from Explainable AI Methods

Four images of 4 subjects (2 women, 2 men) from the test fold are used to analyse the BioVid images with XAI (see Figure 5.6). The description above the images describes the true class label. The distinction between the two pain intensities (e.g., image (1) and (2) of Figure 5.6) is shown only for purposes of traceability. In the classification task, these two intensities are combined under a single class 'pain'. In the following subsections, the results of the XAI approaches decovnet, backpropagation, guided backpropagation, guided Grad-CAM, LRP, and LIME are presented. The order of the images shown in each of the generated XAI salieny maps and heatmaps corresponds to the order shown in Figure 5.6. Above each image the true class label is displayed, followed by the class label predicted by the CNN. Images (1), (3), (8), (9), (11), and (15) were classified correctly, while the remaining pictures were misclassified.

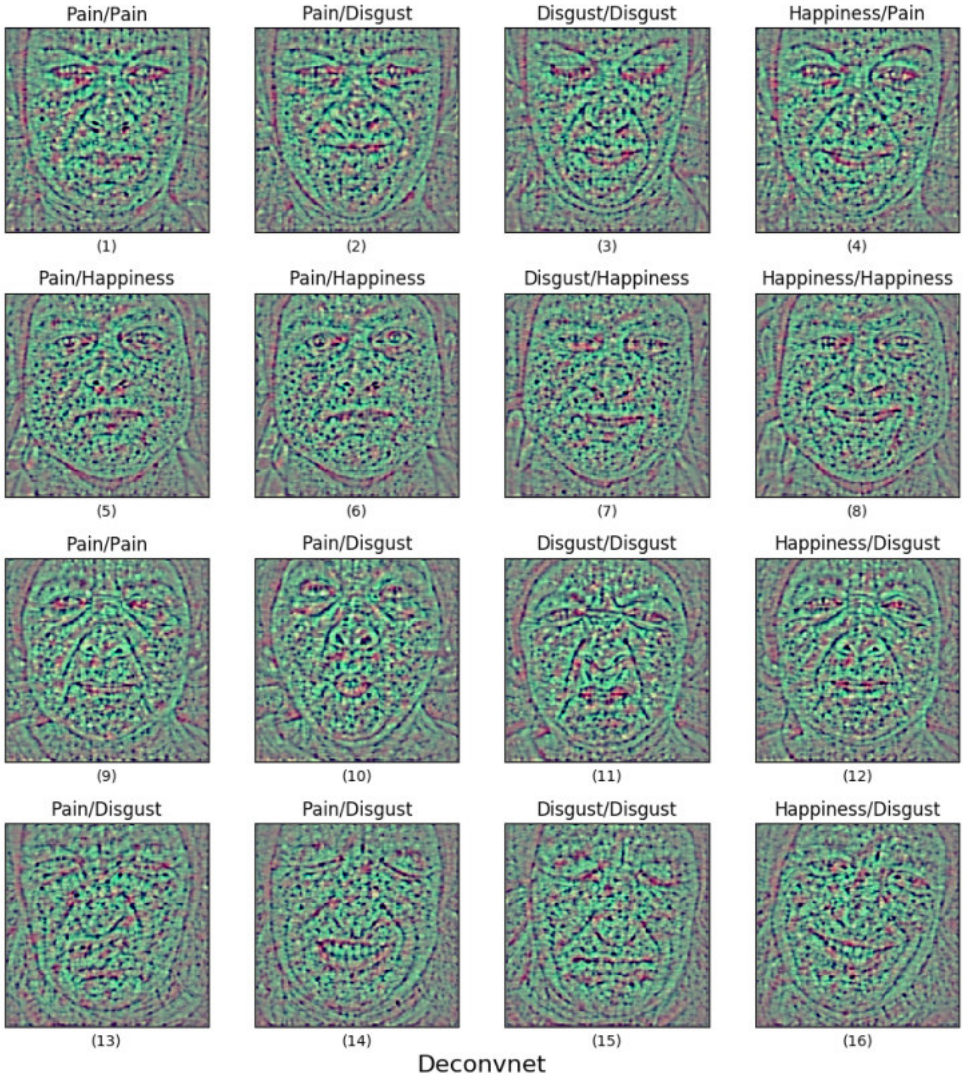


FIGURE 5.7: Deconvnet: Saliency maps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6

5.2.1 Deconvnet

In Figure 5.7 the saliency maps created with the deconvnet approach are shown. With the visualizations generated by the deconvnet method, the original faces can be perceived with the human eye. These are displayed in very fine granular form. A distinction of facial regions important for the prediction is not visible.

5.2.2 Backpropagation

In Figure 5.8 the saliency maps created with the backpropagation approach are shown. Here the relevant pixels are displayed as white dots, the rest is displayed in black. Although a fine granularity can be determined for the created images, it is not clearly visible to the human eye which parts of the face were relevant for the classifier for its decision. There are hardly any contours of a face or facial regions (e.g., nose,

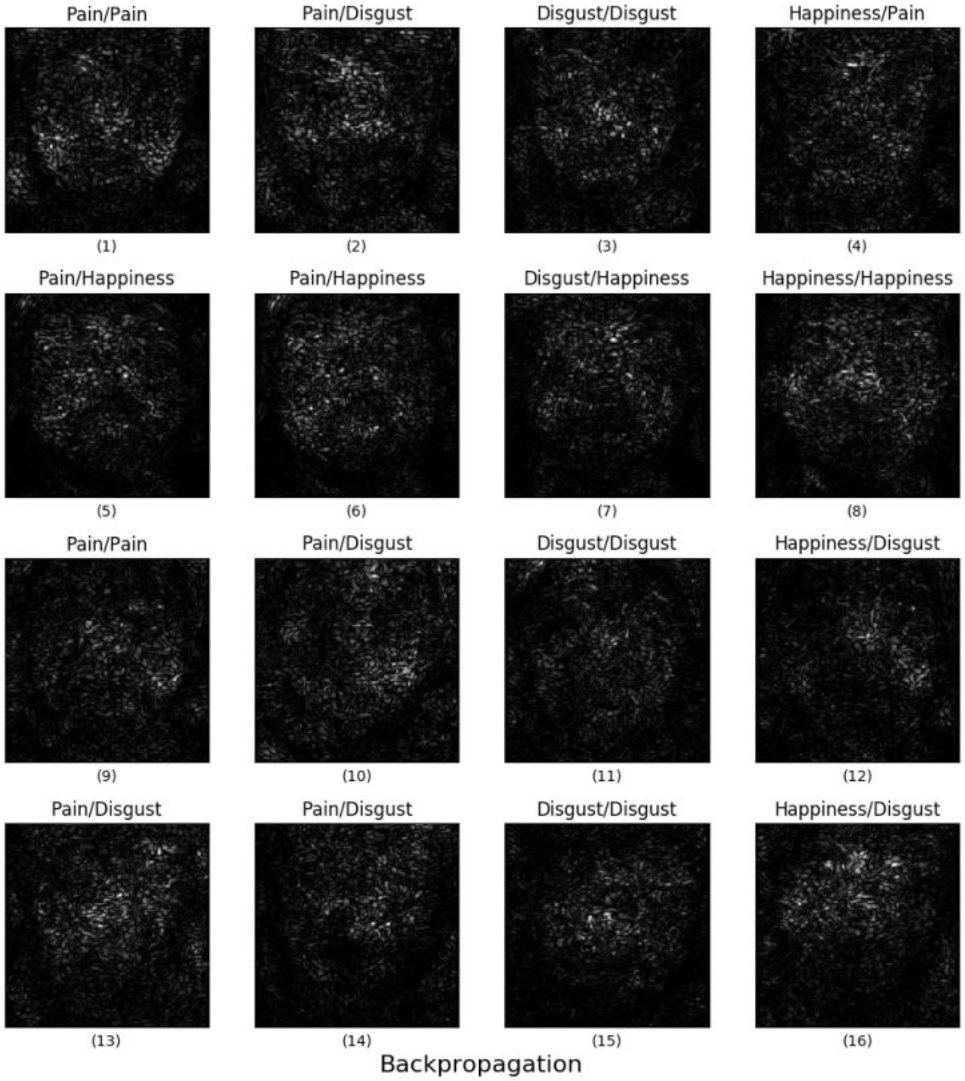


FIGURE 5.8: Backpropagation: Saliency maps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6.

mouth) to be recognized. This makes an interpretation almost impossible. It is also not visually possible to differentiate between the classifications.

5.2.3 Guided Backpropagation

The results for the guided backpropagation approach is displayed in Figure 5.9. For guided backpropagation, the results of the last convolution layer of the CNN were used. Similar to the backpropagation method, bright pixels indicate a contribution to the predicted class, while darker pixels are not important for the prediction. The result of the guided backpropagation method allows faces to be recognized by the human eye to see which regions were relevant for the classification. Despite visual improvements compared to the backpropagation method, differences between the individual classifications are not clearly visible.

FIGURE 5.9: Guided backpropagation: Saliency maps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness.

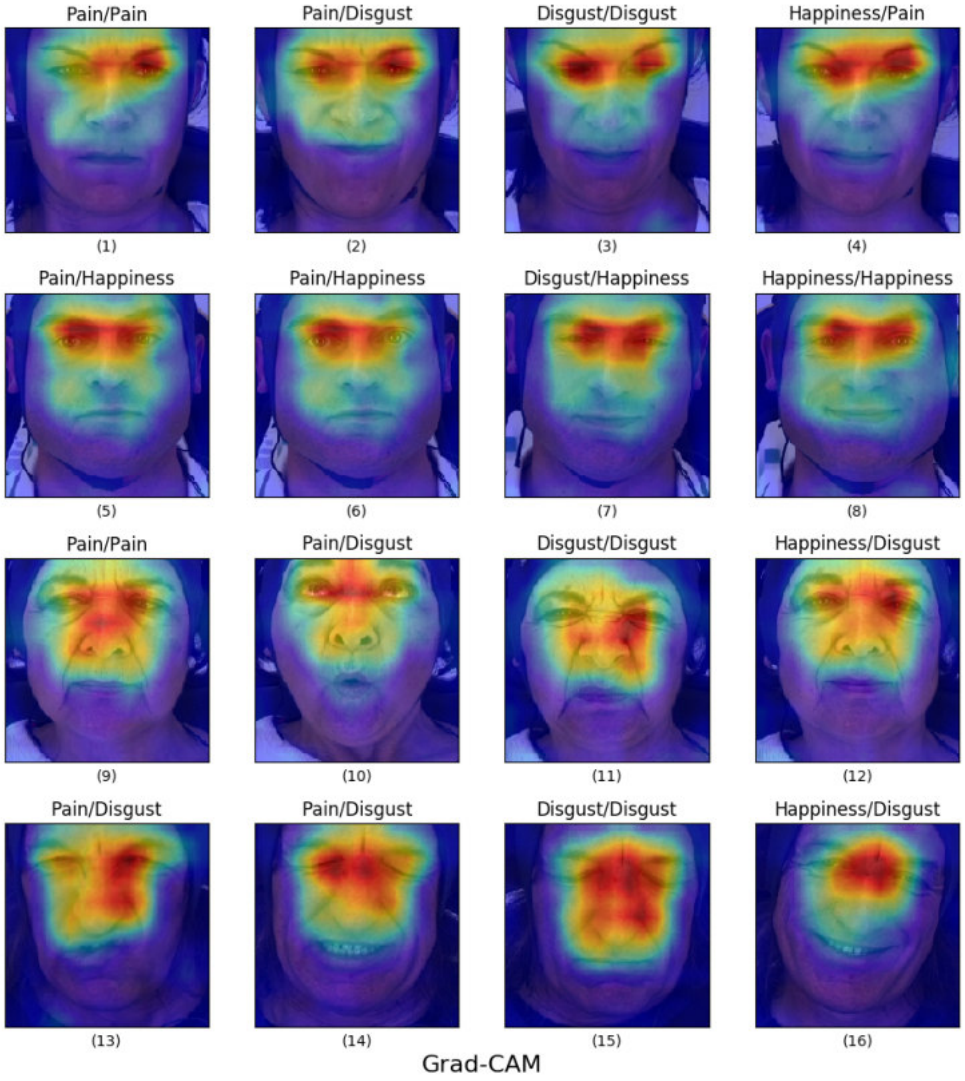


FIGURE 5.10: Grad-CAM: Heatmaps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6

5.2.4 Grad-CAM

In Figure 5.10 the results of the Grad-CAM approach are visualized. Here, the fully connected layers of the CNN are also involved in the analysis. Red areas indicate a high relevance of these areas for the classification and blue areas indicate a low relevance. Visualization using the Grad-CAM approach shows that in the classification of pain the eye area seems to be mainly and almost exclusively relevant for the classifier. With disgust, besides the high relevance of the eyes, the nose also shows up as an important region (see Subfigures (11) and (15)). For Happiness, the eyes are again relevant for classification, but here the area extends to the mouth region (see Subfigures (5), (6), (7), and (8)). Although there are visual differences, they are not sufficient to distinguish between the different classifications. Especially when looking at happiness and pain images, there are hardly any visual differences in the heatmaps.

FIGURE 5.11: Guided Grad-CAM: Saliency maps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6.

5.2.5 Guided Grad-CAM

In Figure 5.11 the results of the guided Grad-CAM approach are displayed. The method of guided Grad-CAM combines the fine granularity of the guided backpropagation approach with the ability of class discrimination of the Grad-CAM approach. Compared to the Grad-CAM approach, finer differences become clear here. Not relevant pixels are grayed out. In the case of disgust, the areas around the nostrils and the upper side of the nose are particularly relevant (see Subfigures (2), (3), (10), (11), (12), (13), (14), (15), and (16)). The images of happiness show that the upper side of the nose and the area of the cheeks, the region enclosed from the corners of the mouth to the eyes is an important area for classification (see Subfigures (5), (6), (7), (8)). In pain, the areas above the lashes and nose seem to be important (see Subfigures (1), (4), and (9)).

5.2.6 LRP

The LRP approach gives results with different focuses, depending on the used parameters. The basic LRP approach is applied in Figure 5.12 (LRP-Z), where no stabilizers are used. In the visualization, negative as well as positive values are represented in every picture. In Figure 5.13 (LRP-PresetAFlat), an α value of 1, a β value of 0, and an ε value of 1e-1 are used. In Figure 5.14 (LRP-PresetBFlat), an α value of 2, a β value of 1, and an ε value of 1e-1 are used. Comparing these two visualizations with the different α and β weights, it becomes apparent that with the increase of the α value the positive pixel values become more prominent, whereas this is not the case with the increase of the β values for the negative pixel values. In Figure 5.15, an ε value of 1e-7 is used. In comparison to the basic LRP approach, much less noise is represented here. For disgust, the folds on the sides of the nose are important. For happiness, the eyes and the lips are important. For pain, nose and eyes are important. It can also be seen that some pixels in these regions also make a negative contribution to classification. If one want a closer look at the positive pixels, Figure 5.13 should be viewed. Here, the focus lays more on the positive pixels. As in Figure 5.15, especially the eyes are important for classifying happiness. The nose gets important when detecting disgust. For pain, areas around the nose and the eyes are important.

5.2.7 LIME

LIME was the only model-agnostic approach tested in this master's thesis. The results for the up to three most important regions detected by LIME are displayed in Figure 5.16. Similar interpretations can be derived as with the methods described earlier, but due to the size of the superpixels, they are not quite as detailed as with the model-specific approaches such as guided Grad-CAM or LRP. The LIME method shows that the area around the nose and the upper cheek area starting from the nose are particularly relevant for the classification of disgust (see Subfigures (2), (10), (11), (13), and (14)). In addition to the nose and cheek area, the eye area also seems to be significant for the detection of the emotion happiness (see Subfigures (5), (6), (7), (8)). In the case of pain, the forehead region, the mouth and parts of the eye area are relevant for the classification (see Subfigures (1), (4), and (9)), depending on the input image. In Figure 5.17, the areas that are relevant for the classification (green) as well as the areas that are not relevant for this classification (red) are shown. In general, areas not directly related to the face (e.g., see Subfigures (4) and (7)) are not helpful for the classification of pain and happiness. Depending on the image, the forehead, the corners of the mouth, and the background are not relevant for the classification (see Subfigures (3), (14), and (16)).

5.3 Generalization

In addition to classifying the images from the test folder, 4 images (2 for disgust, 2 for happiness) of 2 subjects from the Actorstudy dataset[1] were used (see Subfigures (1) to (4) of Figure 5.18). Additionally, 2 images of 2 subjects from the UNBC-McMaster shoulder pain expression archive database (Lucey, Cohn, Prkachin, Solomon, & Matthews, 2011) showing a facial expression of pain were used (see Subfigures (5) and (6) of Figure 5.18). Like the BioVid datatset, the images of Actorstudy and

[1] Unpublished dataset from Intelligent Systems Group, Fraunhofer IIS, Erlangen

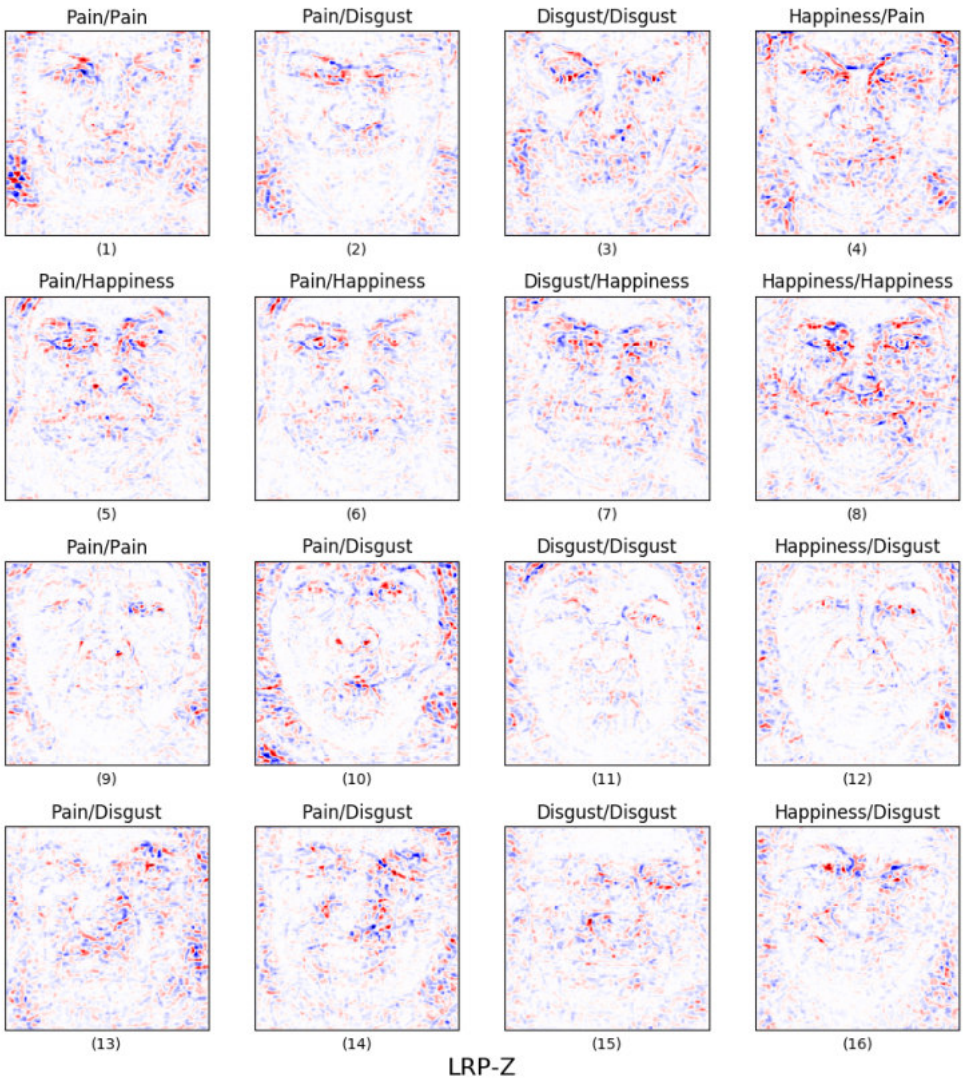


FIGURE 5.12: LRP-Z: Heatmaps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6.

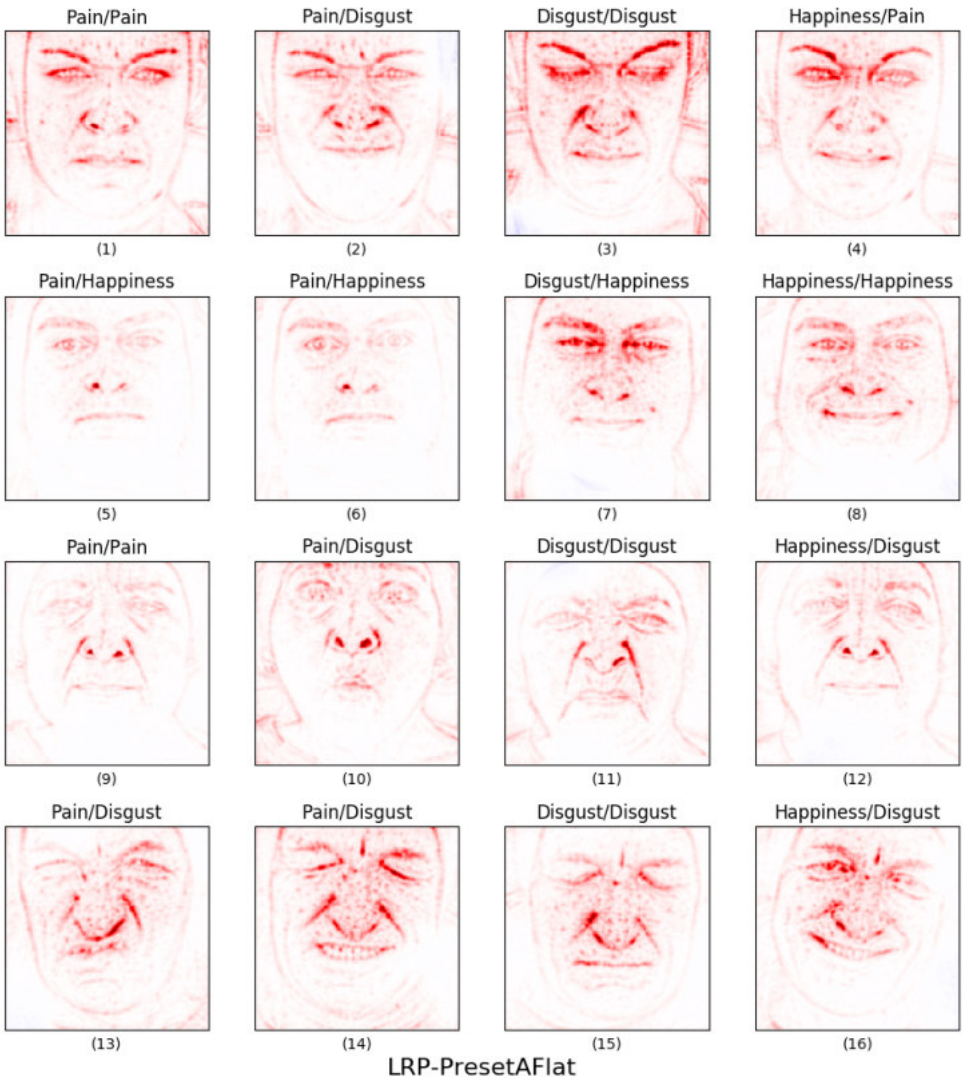


FIGURE 5.13: LRP-PresetAFlat: Heatmaps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6.

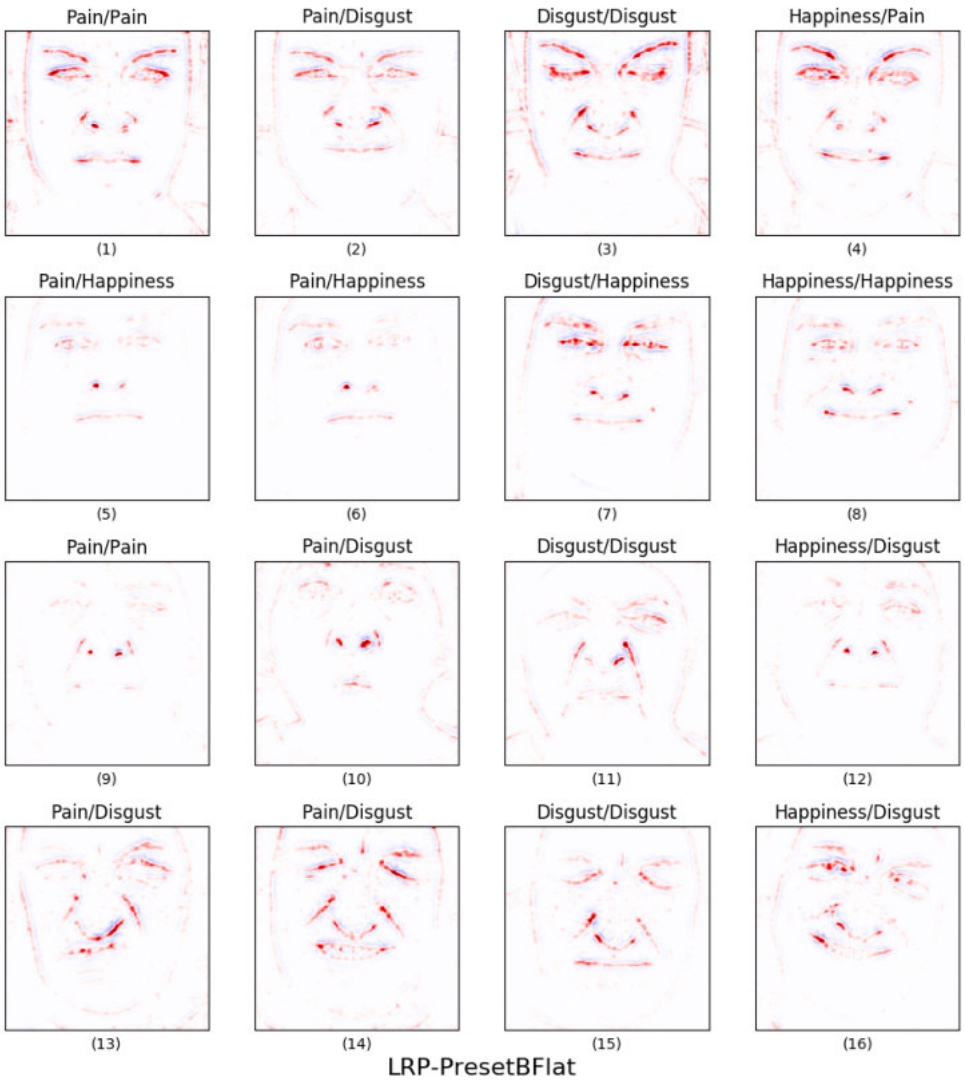


FIGURE 5.14: LRP-PresetBFlat: Heatmaps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6.

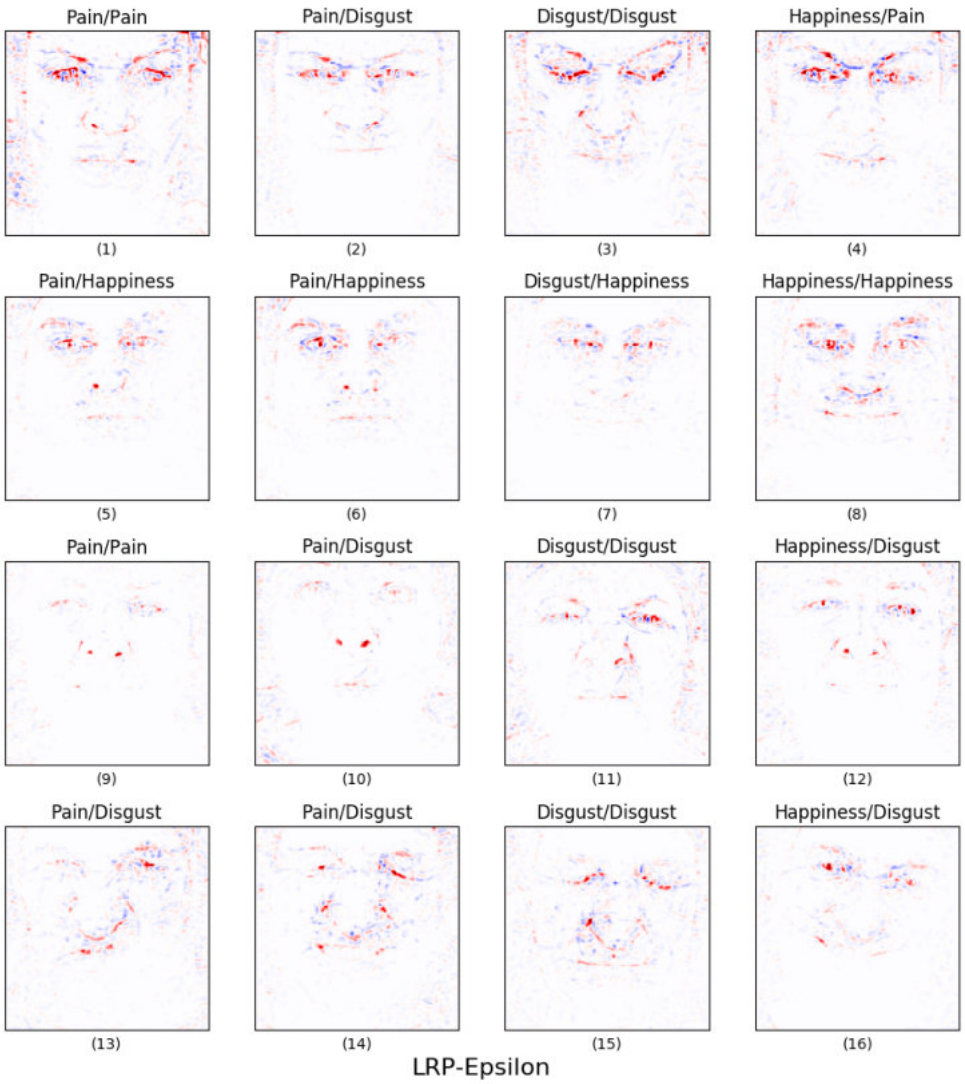


FIGURE 5.15: LRP-Epsilon: Heatmaps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6.

FIGURE 5.16: LIME: Heatmaps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6.

FIGURE 5.17: LIME: Heatmaps of the 4 selected subjects of the BioVid dataset, expressing pain (intensity 3 and 4), disgust and happiness. The order of the images is the same as in Figure 5.6. In Comparison to Figure 5.16, not only the positive features as also the negative features are highlighted.

UNBC-McMaster shoulder pain expression archive database were cropped using the landmarks generated by using the DLib C++ library (King, 2009). The purpose of using these images is to visualize whether the CNN had learned dataset-specific features or whether the features are dataset-independent. The XAI methods Grad-CAM, guided Grad-CAM, LRP-PresetAFlat, and LIME were used for visualization. The predictions for the images which displays emotions were correctly classified by the CNN. The network could not classify correctly one of the two pain pictures. The visualizations generated by the XAI approaches show that the features do not differ much from the images in the test folder. Already with the Grad-CAM approach the importance of the region of the nose at eye level becomes visible in the classification of disgust (see Figure 5.10). Further details cannot be worked out due to the lack of detail. The visualizations of guided Grad-CAM (see Figure 5.20) and LRP (see Figure 5.21) reveal finer details: in happiness, the mouth is an indicator for the classification, while this is less relevant for the classification of disgust. As with the visualizations of the images in the test folder, the results of LIME only provide a rough orientation, due to the size of the super-pixels. Regions around the eyes are important to classify disgust (see Subfigures (1) and (6) of Figure 5.22). For happiness, parts of the mouth are relevant (see Subfigures (2) and (4) of Figure 5.22). Figure 5.23 shows not only positive super-pixels but also negative super-pixels. Here it can be seen that aspects of the background make the classification worse. Additionally, for the classification of pain, the classifier detect many face related super-pixel as not relevant (see Subfigure (5) of Figure 5.23).

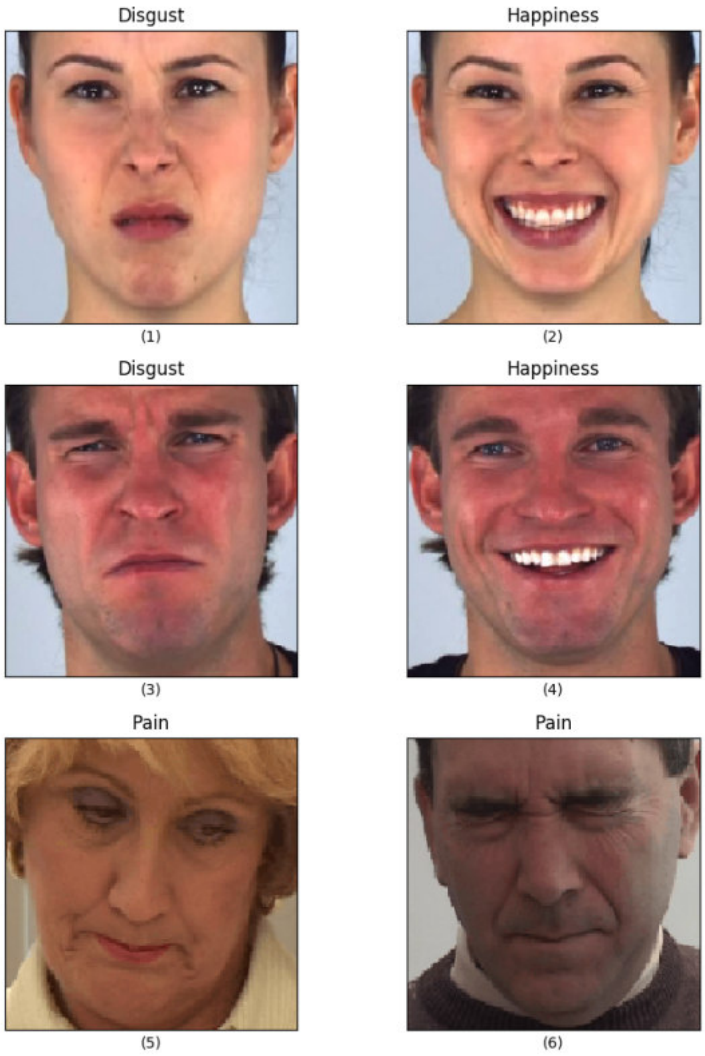


FIGURE 5.18: Six images of different datasets are used to verify the feature-generalization of the CNN. Images (1) to (4) are from the Actorstudy dataset (© Fraunhofer IIS). Images (5) and (6) are from the UNBC-McMaster shoulder pain expression archive database (© Jeffrey Cohn).

FIGURE 5.19: Grad-CAM: Heatmaps of the 3 subjects of the Actorstudy dataset and UNBC-McMaster shoulder pain expression archive database, expressing pain, disgust and happiness. The order of the images is the same as in Figure 5.18.

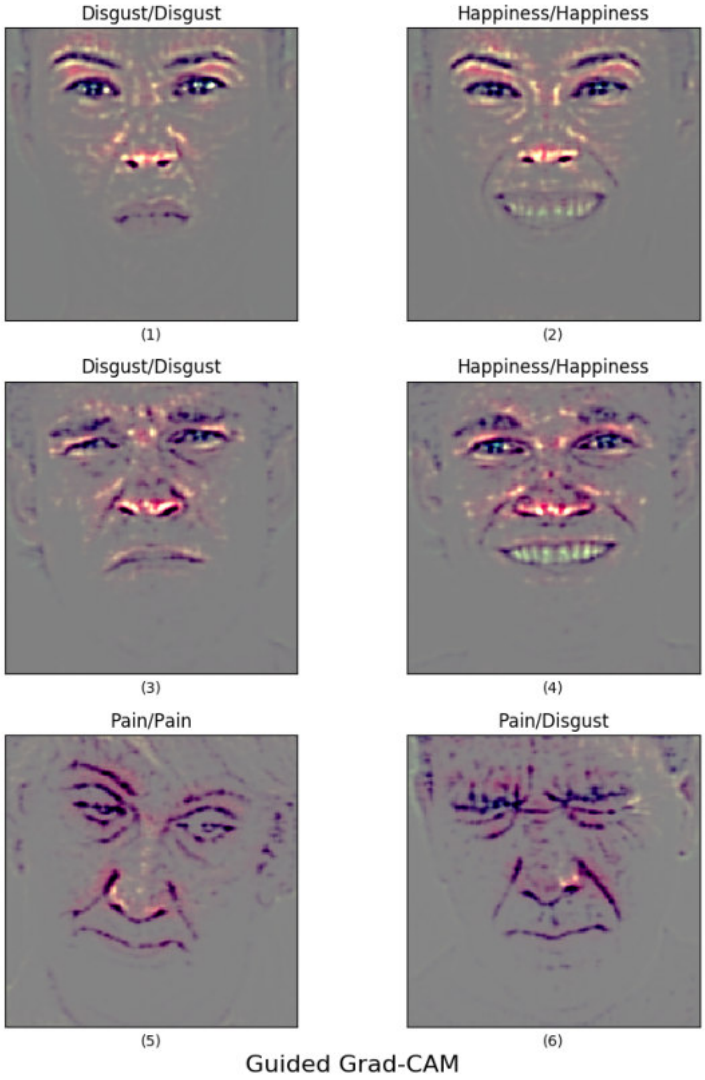


FIGURE 5.20: Guided Grad-CAM: Heatmaps of the 3 subjects of the Actorstudy dataset and UNBC-McMaster shoulder pain expression archive database, expressing pain, disgust and happiness. The order of the images is the same as in Figure 5.18.

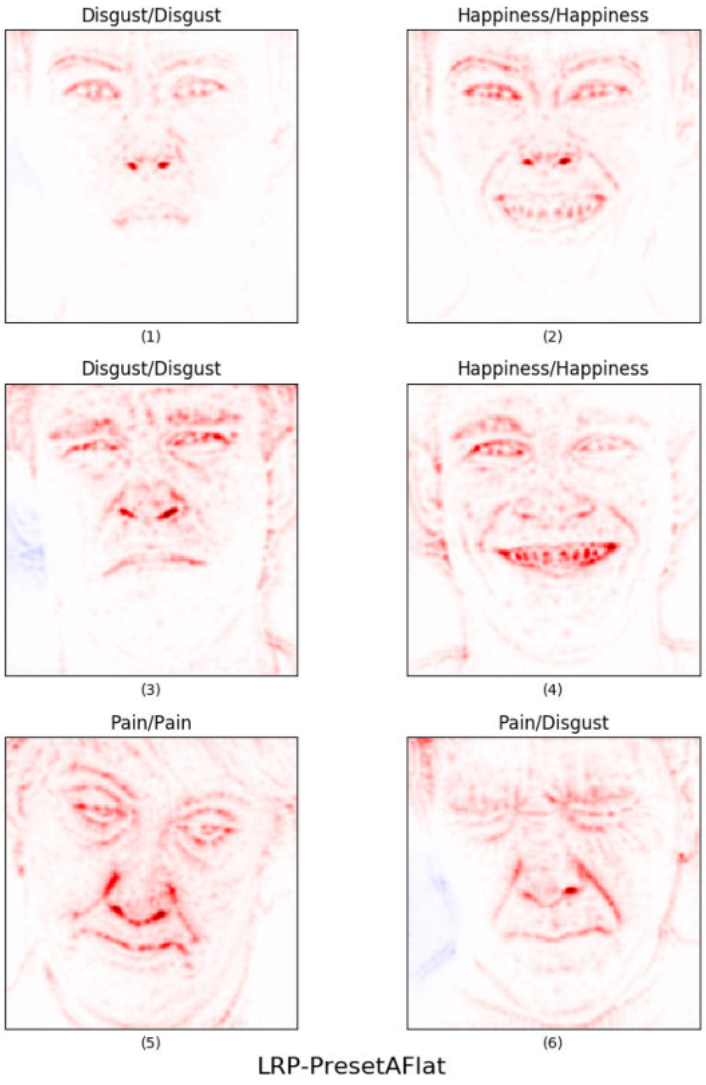


FIGURE 5.21: LRP: Heatmaps of the 3 subjects of the Actorstudy dataset and UNBC-McMaster shoulder pain expression archive database, expressing pain, disgust and happiness. The order of the images is the same as in Figure 5.18. The heatmaps include positive (red) and negative (blue) pixels.

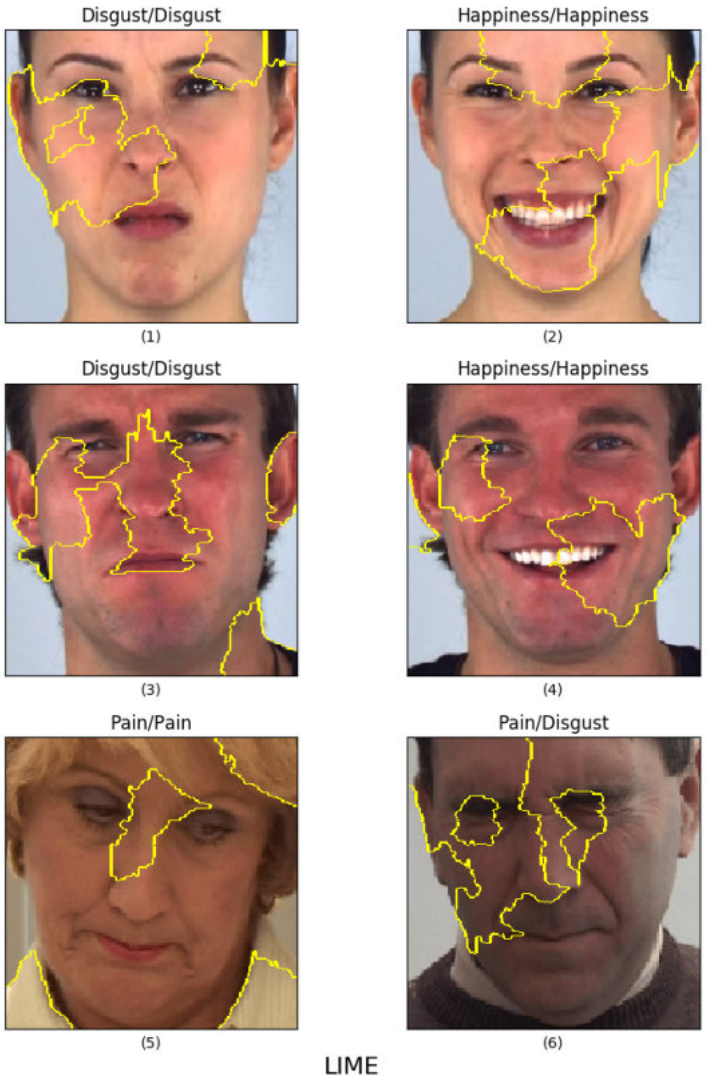


FIGURE 5.22: LIME: Heatmaps of the 3 subjects of the Actorstudy dataset and UNBC-McMaster shoulder pain expression archive database, expressing pain, disgust and happiness. The order of the images is the same as in Figure 5.18. The heatmaps only display positive super-pixels.

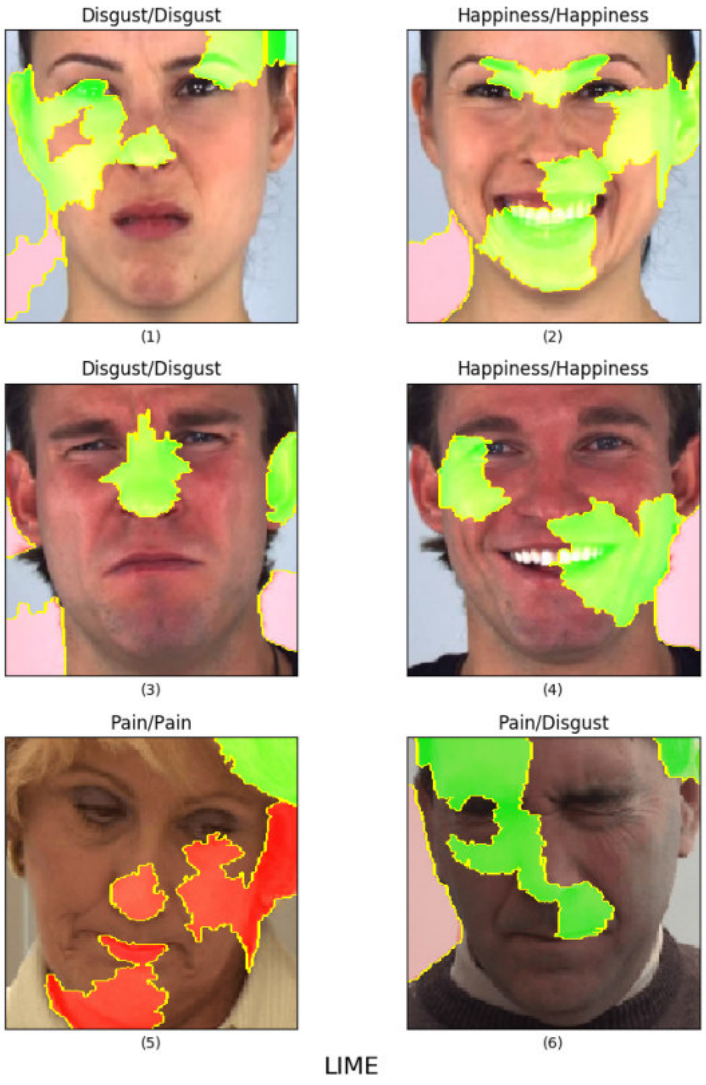


FIGURE 5.23: LIME: Heatmaps of the 3 subjects of the Actorstudy dataset and UNBC-McMaster shoulder pain expression archive database, expressing pain, disgust and happiness. The order of the images is the same as in Figure 5.18. The heatmaps displays negative (red) and positive (green) super-pixels.

Chapter 6

Discussion

The three questions to which this master's thesis would like to provide answers are summed up as **predictive performance**, **decision interpretation**, and **feature explanation**. In the first section, the question about **predictive performance** is answered, followed by the second section, which answers the research questions about **decision interpretation** and **feature explanation**. Subsequently, in a further section, the limitations of this master's thesis will be discussed. Finally, in the last section, an outlook on future research questions will be given.

6.1 Predictive Performance

The analysis of CNN revealed that the network had difficulty classifying happiness correctly. Above all, happy faces were often misclassified as faces of pain. An explanation for this can be the diversity of the dataset. Since the BioVid dataset does not consist of actors, the prototypical representation of the pain expression is not given. As already described in Kunz and Lautenbacher (2014), one can assume different facial expressions for pain. Another explanation can be alexithymia. Alexithymia describes the lack of facial expression of feelings. This can arise especially in stressful situations (Dinges et al., 2005). Since the induction of pain can be a stressful situation for people, this can be an explanation. By using XAI methods for a prototypical case, the relevant facial characteristics for this misclassification could be made visible. Images which were labelled as pain often show facial expressions that reminded more of a smile than a painful face. The CNN has not much problems distinguishing disgust and pain. By using two additional datasets, the Actorystudy dataset and the UNBC-McMaster shoulder pain expression archive database, selected XAI methods were used to demonstrate that the CNN has learned to record dataset-independent features. This is an important point to ensure correct detection in practical applications. Whereas the emotions in the BioVid dataset were induced, the emotions in the Actorstudy dataset were posed. The classification of the six example images of the two datasets showed 5 correct predictions. Motley and Camden (1988) found out that posed facial expressions of emotions are easier to identify by humans than spontaneous expressions of emotions. Even if the CNN is not a human, this could be the reason that the classification of the posed emotions in the generalization image examples is often correct.

6.2 Decision Interpretation & Feature Explanation

The objective of the master's thesis is to use XAI methods to make the black-box behaviour of CNNs interpretable for humans. To interpret the outcome of a CNN, an explanation is given by the saliency maps or heatmaps, generated for the input

images by the different XAI methods. In general, the goal was to find and visualize good explanations for the behaviour of a network. But what does 'a good explanation' actually mean? According to Selvaraju et al. (2016), two aspects are important for it:

1. The visual explanation should be class discriminative. This means that one should be able to perceive visual differences between different classes.

2. The visualization should have a good resolution, so that even fine-granular differences can be perceived.

The presented approach of (guided) backpropagation (Simonyan et al., 2014; Springenberg et al., 2014) and deconvnet (Zeiler & Fergus, 2014) mainly address aspect 2. When using saliency maps generated by (guided) backpropagation, a very detailed resolution with pixel accuracy is possible. As already mentioned in Selvaraju et al. (2016), the results of this master's thesis show that the generated saliency maps using (guided) backpropagation and deconvnet are not class discriminative (aspect 1), since the differences between classes are barely perceptible to the human eye. It is important to note that deconvnet does not directly visualize the learned features, but is 'conditioned' on an image (Springenberg et al., 2014). This is due to the fact that switches are used for reconstruction, which store the information about the location of the max values. In addition, deconvnet has difficulties visualizing an interpretable image structure for higher layers. This is because more invariant representations are learned in higher layers. This results in a single image not being able to activate these neurons at most (Springenberg et al., 2014). As Selvaraju et al. (2016) stated, the pixel-space gradient visualization approaches like (guided) backpropagation (Simonyan et al., 2014; Springenberg et al., 2014) and deconvnet (Zeiler & Fergus, 2014) are outperformed by guided Grad-CAM. The Grad-CAM approach (Selvaraju et al., 2016) fulfils aspect 1. Visualisations computed by Grad-CAM are highly class discriminative, but are not fine granular. The guided Grad-CAM approach (Selvaraju et al., 2016) can do both: the visualizations created by this approach are fine granular as well as class discriminative. These can be observed also in the results of this master's thesis. Fine granular details can be seen in the images and the unimportant parts of the images are grayed out.

As Bach et al. (2015) describe, the LRP approach differs from Simonyan et al. (2014) in the interpretation of the visualisations itself. The backpropagation approach answers the question 'What makes the painful face more or less a painful face?' In contrast, the LRP approach answers the question 'What makes the painful face a painful face?'. LRP offers many settings and parameters that can be changed to achieve the best possible result. The large number of parameters also has a disadvantage: correct assumptions are required as to what exactly one wants to make visible, and how negative and positive pixels could be connected. These assumptions can only be made to a limited extent in advance. Montavon et al. (2017) point out that different combinations of the default values for α, ε, and β must be tried to see how the visualizations change. This means that the fine-tuning for LRP takes some time due to trial and error. The results of this master's thesis are therefore only a first step towards finding the 'best' visualization.

The approach of Ribeiro et al. (2017), namely LIME, as well as the LRP approach of Bach et al. (2015) were able to display positive and negative (super-) pixels for classification. In contrast to the LRP method, the resolution of LIME was not very detailed, due to the size of the super-pixels. LIME uses quickshift (Vedaldi & Soatto, 2008) as segmentation method to create super-pixels. The quickshift algorithm uses

information about colour and pixel location to calculate super-pixels. Because every image of the used datasets have different colours, the generated super-pixels are different for every image. The different colouring depends on the skin tone, hair colour, make up, background and eye colour of the subjects. Therefore, the generated super-pixels are different, i.e., varying in size for every image. This makes the explanation for disgust, happiness and pain very blurred. In contrast to the LRP and guided backpropagation methods, LIME cannot display fine granular details (e.g., the influence of the nostrils). Another limitation of LIME has already been described by Ribeiro et al. (2017). They point out that LIME, as the name implies, creates local explanations. This makes it possible to display non-linear models linearly. If the best model is very strongly non-linear, even in the range of linear prediction then LIME cannot provide a good explanation (Ribeiro et al., 2017).

In the given examples the features for pain, happiness, and disgust are not always distinguishable from one another. This is partly because the visualizations refer to the specific image and not to the features generally learned by the CNN. Another reason is that the facial expressions for the respective examples of a class are very differently represented. In the generalization examples using the Actorstudy dataset, better predictions can be seen due to the posed emotions, which are also reflected in a more clearly interpretable visualization.

To summarize, it can be said that model-specific as well as model-agnostic XAI approaches are suitable for making the classifications of a CNN visible for interpretation. However, the interpretability of the different visualizations depends on the granularity and the power for class discrimination. It is also important to cleary highlight relevant regions so that the human eye can easily detect important segments in the face. Guided Grad-CAM, LRP and LIME seem to be the best interpretable approaches for this purpose.

6.3 Limitations

The BioVid dataset used in this master's thesis showed some problems. Due to the unbalanced classes, a manual frame selection for the emotions disgust and happiness had to be done. This manual selection is a highly subjective process. Therefore, it cannot be guaranteed that the selected frames correspond in intensity and clarity to those of the category pain. In addition, it must be noted that the pain recordings were real pain experienced but the emotions were induced by showing IASP images (Walter et al., 2013). The question arises whether the emotion induction with the help of IAPS images was sufficient or whether a stronger emotional expression could have been induced by a combination of other methods like auditory and visual stimuli. The study by Baumgartner, Esslen, and Jäncke (2006) showed that a combination of images and music can significantly increase the emotional experience. In the meta-analysis of Westermann, Spies, Stahl, and Hesse (1996) similar findings appeared. Emotion induction with the help of films (in which the combination of image and sound can also be found) produced the strongest effect. Westermann et al. (1996) note that the effects are less pronounced if the subjects are not informed about the purpose of the experiment. Stronger facial expressions might have made it easier for CNN to distinguish happiness from pain. In the work of Walter et al. (2013), no information is given regarding whether the subject was informed about the purpose of the BioVid experiment. Furthermore, no more additional details about the emotion sequences of the BioVid dataset are given. Therefore, it remains unclear whether frames of all possible valence and arousal conditions have made it into the training,

validation, and test sets. In addition, this possible broad spectrum of positive and negative emotions makes clear prototypical classification difficult. There was no information about an objective or subjective rating of the respective emotions, as it was the case with pain, where a person-specific pain threshold was determined and used (Walter et al., 2013).

While applying the different XAI methods, a practical disadvantage of LIME became apparent: LIME needed an average of 8 minutes to generate a visualization for an image. In comparison, the model-specific approaches took only a few seconds (e.g., LRP needed 1-5 seconds per image). LIME needs too much time to be useful for real-time applications in the field. Here, optimization of the LIME method and the usage of parallel processing (if possible), are some feasible solutions.

The informative value of the visualizations generated by the XAI methods can be further improved. Depending on the approach, it was very difficult to identify features used for classification. There are three possible reasons for this:

1. The choice of XAI method: For example, the class discrimination abilities of deconvnet and backpropagation are poor (Selvaraju et al., 2016).

2. The configuration of XAI method: The LRP approach, in particular, provides numerous possibilities to adjust the visualizations through the parameters α, β and ε, and through the flat and present versions.

3. The representation of CNN performance: The visualizations generated by the XAI approaches provide only a partial view of the classification capabilities of the CNN. Classification accuracy must be taken into account when looking at the visualizations.

The last point in particular must not be forgotten when reading the results of this master's thesis. The accuracy of the used folder of the VGG Face network was 66%. Although this is higher than the probability of guessing, it can certainly be improved.

6.4 Future Research

The classification performance of the network used in this master's thesis leaves much room for improvement. Optimization by adapting various hyperparameters is conceivable. For example, the analysis and adjustment of the threshold with which the network makes its decision for a classification would have to be considered. XAI methods can also be used to improve and optimize the network. Montavon et al. (2017) describe that the application of XAI methods is not only limited to the analysis of the output of a network, but also for the analysis of scientific data (Montavon et al., 2017). In addition, XAI methods can be used to analyse and validate the network itself. Already in the works of Selvaraju et al. (2016), Springenberg et al. (2014), Zeiler et al. (2011), Zeiler and Fergus (2014) XAI methods were used to visualize the learned features of different layers. XAI is therefore not only able to visualize the final result of a CNN, but also the learned features in the previous layers. This allows a previously non-existent view into the interior of a network. This view can be helpful in improving and optimizing CNNs. The visualisation itself can be optimized as well. For the LRP approach, adjustments of the α, β and ε values are possible to control the importance of the influence of positive and negative pixels (Bach et al., 2015). Moreover, Montavon et al. (2017) describe some practical recommendations

to improve the visualisations generated by the LRP method: using dropout as regularization technique, preferring sum pooling instead of max pooling and not to use too many fully connected layers in the network (whereas no definition is given what is meant by 'many').

The XAI methods presented in this master's thesis are not exhaustive. There are other approaches, such as DeepFaceLIFT (Liu, Peng, Shea, & Picard, 2017). In this approach, the landmarks used for face detection are coloured according to their significance for the classification decision.

In this master's thesis, a frame-based approach was followed to train and test the CNN. In the work of Ashraf et al. (2009), sequence-based approaches present themselves as a possible alternative. Although not the same hit rates are achieved as with frame-based approaches, sequence-based approaches are interesting because it allows to encode temporal characteristics. Ashraf et al. (2009) use a Support Vector Machine for feature extraction. For a sequence-based analysis, Long Short-Term Memory (LSTM) can be used (Hochreiter & Schmidhuber, 1997). The work of Rodriguez et al. (2018) already shows promising results in using LSTMs to include temporal information of a painful sequence to improve the network's performance.

Only two of the six basic emotions were used in this master's thesis to train the CNN. For use in practice, a pain recognition system should be able to distinguish pain from other emotional states. An expanded network that has been trained on all six basic emotions and pain would be useful to explore. Alternatively, the use of AUs (Craig et al., 1992; Ekman & Friesen, 2003) could be considered. The use of AUs would have the advantage that the results of the visualizations could be interpreted even better by comparing them with previous findings in emotion and pain research. In addition, Lucey et al. (2009) could show that the results are better when using AUs instead of features directly extracted from the images.

The application of different XAI methods, as presented in this master's thesis, represents only a first step into the analysis of the classification results of a CNN in the field of pain and emotions. The qualitative interpretation of the results is very subjective. An analysis of the interpretability of the visual XAI results still has to be empirically verified. An evaluation with a larger sample of laypersons and medical experts seems appropriate to obtain meaningful results on interpretability.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … Zheng, X. (2016). Tensorflow: large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 1–21.

Ambady, N. & Rosenthal, R. (1992). Thin slices of expressive behavior as predictors of interpersonal consequences: a meta-analysis. *Psychological Bulletin*, *111*(2), 256–274.

Ashraf, A. B., Lucey, S., Cohn, J. F., Chen, T., Ambadar, Z., Prkachin, K. M., & Solomon, P. E. (2009). The painful face–pain expression recognition using active appearance models. *Image and Vision Computing*, *27*(12), 1788–1796. doi:10.1016/j.imavis.2009.05.007

Aviezer, H., Trope, Y., & Todorov, A. (2012). Body cues, not facial expressions, discriminate between intense positive and negative emotions. *Science*, *338*(6111), 1225–1229. doi:10.1126/science.1224313

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10*(7), e0130140. doi:10.1371/journal.pone.0130140

Bach, S., Binder, A., Müller, K.-R., & Samek, W. (2016). Controlling explanatory heatmap resolution and semantics via decomposition depth. In *Proceedings of the international conference on image processing* (pp. 2271–2275). IEEE.

Bartlett, M. S., Hager, J. C., Ekman, P., & Sejnowski, T. J. (1999). Measuring facial expressions by computer image analysis. *Psychophysiology*, *36*(2), 253–263. doi:10.1017/S0048577299971664

Baumgartner, T., Esslen, M., & Jäncke, L. (2006). From emotion perception to emotion experience: emotions evoked by pictures and classical music. *International Journal of Psychophysiology*, *60*(1), 34–43. doi:10.1016/j.ijpsycho.2005.04.007

Brahnam, S., Chuang, C.-F., Shih, F. Y., & Slack, M. R. (2006). Machine recognition and representation of neonatal facial displays of acute pain. *Artificial Intelligence in Medicine*, *36*(3), 211–222. doi:10.1016/j.artmed.2004.12.003

Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

Clore, G. L. & Huntsinger, J. R. (2007). How emotions inform judgment and regulate thought. *Trends in Cognitive Sciences*, *11*(9), 393–399. doi:10.1016/j.tics.2007.08.005

Craig, K. D., Prkachin, K. M., & Grunau, R. V. (1992). The facial expression of pain. *Handbook of pain assessment*, *2*, 153–169.

Damasio, A. R. (1994). *Descartes' Irrtum: Fühlen, Denken und das menschliche Gehirn*. List Verlag, München.

Darwin, C. (1873). *The expression of the emotions in man and animals*. John Murray, London.

de Wit, R., van Dam, F., Hanneman, M., Zandbelt, L., van Buuren, A., van der Heijden, K., … Abu-Saad, H. H. (1999). Evaluation of the use of a pain diary in

chronic cancer pain patients at home. *Pain*, *79*(1), 89–99. doi:10.1016/S0304-3959(98)00158-4

Dinges, D. F., Rider, R. L., Dorrian, J., McGlinchey, E. L., Rogers, N. L., Cizman, Z., ... Metaxas, D. N. (2005). Optical computer recognition of facial expressions associated with stress induced by performance demands. *Aviation, space, and environmental medicine*, *76*(6), B172–B182.

Downie, W., Leatham, P., Rhind, V., Wright, V., Branco, J., & Anderson, J. (1978). Studies with pain rating scales. *Annals of the rheumatic diseases*, *37*(4), 378–381.

Dreiseitl, S. & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics*, *35*(5-6), 352–359. doi:10.1016/S1532-0464(03)00034-0

Duchenne, G.-B., de Boulogne. (1990). *The mechanism of human facial expression*. Cambridge University Press.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, *32*(2), 407–499.

Ekman, P. (1971). Universals and cultural differences in facial expressions of emotion. In *Nebraska symposium on motivation* (pp. 207–283). University of Nebraska Press.

Ekman, P. (1989). The argument and evidence about universals in facial expressions. *Handbook of Social Psychophysiology*, 143–164.

Ekman, P. (1993). Facial expression and emotion. *American Psychologist*, *48*(4), 384–392. doi:10.1037/0003-066X.48.4.384

Ekman, P., Davidson, R. J., & Friesen, W. V. (1990). The duchenne smile: emotional expression and brain physiology: ii. *Journal of Personality and Social Psychology*, *58*(2), 342–353. doi:10.1037/0022-3514.58.2.342

Ekman, P. & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, *17*(2), 124–129. doi:10.1037/h0030377

Ekman, P. & Friesen, W. V. (2003). *Unmasking the face: a guide to recognizing emotions from facial clues*. Malor Books.

Ekman, P., Huang, T. S., Sejnowski, T. J., & Hager, J. C. (1993). Final report to nsf of the planning workshop on facial expression understanding. *Human Interaction Laboratory*, *378*, 1–95.

Ekman, P. & Oster, H. (1979). Facial expressions of emotion. *Annual Review of Psychology*, *30*(1), 527–554. doi:10.1146/annurev.ps.30.020179.002523

Ekman, P. & Rosenberg, E. L. (1997). *What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system (facs)*. Oxford University Press, USA.

Friesen, W. V. & Ekman, P. (1978). Facial action coding system: a technique for the measurement of facial movement. *Palo Alto*.

Friesen, W. V. & Ekman, P. (1983). *Emfacs-7: emotional facial action coding system*.

Frith, C. (2009). Role of facial expressions in social interactions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1535), 3453–3458. doi:10.1098/rstb.2009.0142

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. http://www.deeplearningbook.org. MIT Press.

Gracely, R. H., McGrath, P., & Dubner, R. (1978). Ratio scales of sensory and affective verbal pain descriptors. *Pain*, *5*(1), 5–18. doi:10.1016/0304-3959(78)90020-9

Hauberg, S., Freifeld, O., Larsen, A. B. L., Fisher, J., & Hansen, L. (2016). Dreaming more data: class-dependent distributions over diffeomorphisms for learned data augmentation. In *Proceedings of the 19th international conference on artificial intelligence and statistics* (pp. 342–350). JMLR: W&CP.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735

Ioffe, S. & Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *Proceedings of the 12th international conference on computer vision* (pp. 2146–2153). IEEE. doi:10.1109/ICCV.2009.5459469

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014). Caffe: convolutional architecture for fast feature embedding. In *Proceedings of the 22nd acm international conference on multimedia* (pp. 675–678). ACM. doi:10.1145/2647868.2654889

Kassam, K. S., Markey, A. R., Cherkassky, V. L., Loewenstein, G., & Just, M. A. (2013). Identifying emotions on the basis of neural activation. *PloS One*, *8*(6), e66032. doi:10.1371/journal.pone.0066032

Keefe, F. J. & Wren, A. A. (2013). Assessment of pain behaviors. In *Encyclopedia of pain* (pp. 224–227). Springer. doi:10.1007/978-3-642-28753-4_302

King, D. E. (2009). Dlib-ml: a machine learning toolkit. *Journal of Machine Learning Research*, *10*, 1755–1758.

Kingma, D. P. & Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1–15.

Kohlbrenner, M. H. (2017, April). On the stability of neural network explanations. Bachelor's Thesis.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th conference on advances in neural information processing systems* (pp. 1097–1105).

Kunz, M. & Lautenbacher, S. (2014). The faces of pain: a cluster analysis of individual differences in facial activity patterns of pain. *European Journal of Pain*, *18*(6), 813–823. doi:10.1002/j.1532-2149.2013.00421.x

Kunz, M., Peter, J., Huster, S., & Lautenbacher, S. (2013). Pain and disgust: the facial signaling of two aversive bodily experiences. *PloS one*, *8*(12), e83277. doi:10.1371/journal.pone.0083277

Kunz, M., Seuss, D., Hassan, T., Garbas, J. U., Siebers, M., Schmid, U., … Lautenbacher, S. (2017). Problems of video-based pain detection in patients with dementia: a road map to an interdisciplinary solution. *BMC geriatrics*, *17*(1), 33. doi:10.1186/s12877-017-0427-2

Lang, P. J., Bradley, M. M., & Cuthbert, B. N. (1997). International affective picture system (iaps): technical manual and affective ratings. *NIMH Center for the Study of Emotion and Attention*, 39–58.

Lapuschkin, S., Alber, M., Hägele, M., Schütt, K., & Binder, A. (2018). Lrp. https://github.com/albermax/innvestigate. GitHub.

Lapuschkin, S., Binder, A., Müller, K.-R., & Samek, W. (2017). Understanding and comparing deep neural networks for age and gender classification. In *Proceedings of the international conference on computer vision* (pp. 1629–1638).

LeCun, Y. et al. (1989). Generalization and network design strategies. *Connectionism in Perspective*, 143–155.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. doi:10.1038/nature14539

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems* (pp. 396–404).

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (1998). Efficient backprop. In *Neural networks: tricks of the trade* (pp. 9–50). Springer. doi:10.1007/3-540-49430-8_2

Lench, H. C., Flores, S. A., & Bench, S. W. (2011). Discrete emotions predict changes in cognition, judgment, experience, behavior, and physiology: a meta-analysis of experimental emotion elicitations. *Psychological bulletin*, *137*(5), 834–855. doi:10.1037/a0024244

Lin, M., Chen, Q., & Yan, S. (2014). Network in network. *arXiv preprint arXiv:1312.4400*, 1–10.

Lipton, Z. C. (2017). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 1–9.

Liu, D., Peng, F., Shea, A., & Picard, R. (2017). Deepfacelift: interpretable personalized models for automatic estimation of self-reported pain. *arXiv preprint arXiv:1708.04670*, 1–16.

Lucey, P., Cohn, J. F., Prkachin, K. M., Solomon, P. E., & Matthews, I. (2011). Painful data: the unbc-mcmaster shoulder pain expression archive database. In *Proceedings of the international conference on automatic face & gesture recognition and workshops* (pp. 57–64). IEEE. doi:10.1109/FG.2011.5771462

Lucey, P., Cohn, J., Lucey, S., Matthews, I., Sridharan, S., & Prkachin, K. M. (2009). Automatically detecting pain using facial actions. In *Proceeedings of the 3rd international conference on affective computing and intelligent interaction and workshops* (pp. 1–8). IEEE. doi:10.1109/ACII.2009.5349321

Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, *16*(5-6), 555–559. doi:10.1016/S0893-6080(03)00115-1

McCormack, H. M., David, J. d. L., & Sheather, S. (1988). Clinical applications of visual analogue scales: a critical review. *Psychological medicine*, *18*(4), 1007–1019. doi:10.1017/S0033291700009934

Merskey, H. & Bogduk, N. (Eds.). (2012). *Part iii pain terms, a current list with definitions and notes on usage*. IASP Press. Retrieved from http://www.iasp-pain.org/PublicationsNews/Content.aspx?ItemNumber=1673&navItemNumber=677

Mitchell, T. (1997). *Machine learning*. McGraw-Hill international editions - computer science series. McGraw-Hill Education. Retrieved from https://books.google.de/books?id=xOGAngEACAAJ

Montavon, G., Samek, W., & Müller, K.-R. (2017). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, *73*, 1–15. doi:10.1016/j.dsp.2017.10.011

Motley, M. T. & Camden, C. T. (1988). Facial expression of emotion: a comparison of posed expressions versus spontaneous expressions in an interpersonal communication setting. *Western Journal of Communication (includes Communication Reports)*, *52*(1), 1–22. doi:10.1080/10570318809389622

Overskeid, G. (2000). The slave of the passions: experiencing problems and selecting solutions. *Review of General Psychology*, *4*(3), 284–309. doi:10.1037/1089-2680.4.3.284

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. In *Bmvc* (Vol. 1, *3*, pp. 1–12).

Perez, L. & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 1–8.

Petsiuk, V. (2018). Guided backpropagation, grad-cam, and guided grad-cam. https://github.com/eclique/keras-gradcam. GitHub.

Phelps, E. A. (2004). Human emotion and memory: interactions of the amygdala and hippocampal complex. *Current Opinion in Neurobiology*, *14*(2), 198–202. doi:10.1016/j.conb.2004.03.015

Phelps, E. A., Ling, S., & Carrasco, M. (2006). Emotion facilitates perception and potentiates the perceptual benefits of attention. *Psychological Science*, *17*(4), 292–299. doi:10.1111/j.1467-9280.2006.01701.x

Pitaloka, D. A., Wulandari, A., Basaruddin, T., & Liliana, D. Y. (2017). Enhancing cnn with preprocessing stage in automatic emotion recognition. *Procedia Computer Science*, *116*, 523–529. doi:10.1016/j.procs.2017.10.038

Plutchik, R. (1982). A psychoevolutionary theory of emotions. *21*, 529–553. doi:10.1177/053901882021004003

Prkachin, K. M. (2009). Assessing pain by facial expression: facial expression as nexus. *Pain Research and Management*, *14*(1), 53–58. doi:10.1155/2009/542964

Ribeiro, M. T., Sameer, S., & Guestrin, C. (2017). Lime. https://github.com/marcotcr/lime/. GitHub.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?: explaining the predictions of any classifier. In *Proceedings of the 22nd international conference on knowledge discovery and data mining* (pp. 1135–1144). ACM.

Rodriguez, P., Cucurull, G., Gonzalez, J., Gonfaus, J. M., Nasrollahi, K., Moeslund, T. B., & Roca, F. X. (2018). Deep pain: exploiting long short-term memory networks for facial expression classification. (99, pp. 1–11). IEEE. doi:10.1109/TCYB.2017.2662199

Samek, W., Wiegand, T., & Müller, K.-R. (2017). Explainable artificial intelligence: understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 1–8.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural networks*, *61*, 85–117. doi:10.1016/j.neunet.2014.09.003

Scott, S. K., Young, A. W., Calder, A. J., Hellawell, D. J., Aggleton, J. P., & Johnsons, M. (1997). Impaired auditory recognition of fear and anger following bilateral amygdala lesions. *Nature*, *385*, 254–257. doi:10.1038/385254a0

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, *34*(1), 1–47. doi:10.1145/505282.505283

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., & Batra, D. (2016). Grad-cam: why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1611.07450*, 1–17.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 1–8.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 1–14.

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: the all convolutional net. *arXiv preprint arXiv:1412.6806*, 1–14.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Statistisches Bundesamt (Ed.). (2015). *Ergebnisse der 13. koordinierten Bevölkerungsvorausberechnung*. Statistisches Bundesamt. Retrieved from https://www.destatis.de/DE/Publikationen/Thematisch/Bevoelkerung/VorausberechnungBevoelkerung/BevoelkerungDeutschland2060Presse5124204159004.pdf?__blob=publicationFile

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*, 267–288.

Vedaldi, A. & Soatto, S. (2008). Quick shift and kernel methods for mode seeking. In *Proceedings of european conference on computer vision* (pp. 705–718). Springer. doi:10.1007/978-3-540-88693-8_52

Vytal, K. & Hamann, S. (2010). Neuroimaging support for discrete neural correlates of basic emotions: a voxel-based meta-analysis. *Journal of Cognitive Neuroscience*, *22*(12), 2864–2885. doi:10.1162/jocn.2009.21366

Wall, P. D. (1999). Introduction to the fourth edition. In P. D. Wall & R. Melzack (Eds.), *The textbook of pain* (pp. 1–8). Harcourt Publishers.

Walter, S., Gruss, S., Ehleiter, H., Tan, J., Traue, H. C., Werner, P., ... da Silva, G. M. (2013). The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system. In *Proceedings of the international conference on cybernetics* (pp. 128–131). IEEE. doi:10.1109/CYBConf.2013.6617456

Werner, P., Al-Hamadi, A., Niese, R., Walter, S., Gruss, S., & Traue, H. C. (2013). Towards pain monitoring: facial expression, head pose, a new database, an automatic system and remaining challenges. In *Proceedings of the british machine vision conference* (pp. 1–13). doi:10.5244/C.27.119

Werner, P., Al-Hamadi, A., Niese, R., Walter, S., Gruss, S., & Traue, H. C. (2014). Automatic pain recognition from video and biomedical signals. In *Proceeedings of the 22nd international conference on pattern recognition* (pp. 4582–4587). IEEE. doi:10.1109/ICPR.2014.784

Westermann, R., Spies, K., Stahl, G., & Hesse, F. W. (1996). Relative effectiveness and validity of mood induction procedures: a meta-analysis. *European Journal of social psychology*, *26*(4), 557–580. doi:10.1002/(SICI)1099-0992(199607)26:4<557::AID-EJSP769>3.0.CO;2-4

Williams, A. C. d. C. (2002). Facial expression of pain: an evolutionary account. *Behavioral and brain sciences*, *25*(4), 439–455. doi:10.1017/S0140525X02000080

Wolf, L., Hassner, T., & Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *Proceedings of the conference on computer vision and pattern recognition* (pp. 529–534). IEEE. doi:10.1109/CVPR.2011.5995566

Zeiler, M. D. & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Proceedings of the european conference on computer vision* (pp. 818–833). Springer.

Zeiler, M. D., Taylor, G. W., & Fergus, R. (2011). Adaptive deconvolutional networks for mid and high level feature learning. In *Proceedings of the international conference on computer vision* (pp. 2018–2025). IEEE. doi:10.1109/ICCV.2011.6126474

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the conference on computer vision and pattern recognition* (pp. 2921–2929).

Zhou, Y. & Chellappa, R. (1988). Computation of optical flow using a neural network. In *Proceedings of the international conference on neural networks* (Vol. 1998, pp. 71–78).

Appendix A

CNN Architectures

A.1 CNN Architecture With Early Stopping

Compared to the CNN architecture used in the main part of this master thesis, another model with the same setting but with the regularization technique dropout of 0.5 was used instead of the L2 regularization (see Table A.1). Like the model in the main part, the values are rounded to two and three decimal places, respectively. The end of the learning process was determined by the early stopping method on the validation set.

A.2 CNN Architectures Without Early Stopping

Before early stopping, models with dropout (0.5) or L2 regularization (0.0001) were used. With these settings, the CNN was trained for a fixed number of epochs. In Table A.2 and Table A.3 the results of the CNNs using dropout or L2 regularization for a fixed epoch size of 7 are displayed. In Table A.4 (dropout) and Table A.5 (L2 regularization) the epoch size has the fixed size of 3.

TABLE A.1: Results of the 5-fold cross-validation using early stopping and a dropout of 0.5.

Fold	**Training**		**Validation**		**Testing**		**Epochs**
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	
1	0.04	0.996	1.22	0.639	1.45	0.532	2
2	0.04	0.996	1.26	0.601	1.31	0.599	2
3	0.03	0.997	1.14	0.650	1.49	0.592	2
4	0.04	0.995	1.36	0.599	1.96	0.556	2
5	0.04	0.997	1.78	0.625	1.02	0.662	2
Average					*1.45*	*0.588*	

TABLE A.2: Results of the 5-fold cross-validation using a fixed epoch size of 7 and a dropout of 0.5

Fold	**Training**		**Validation**		**Testing**		**Epochs**
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	
1	0.01	0.998	1.36	0.630	1.41	0.562	7
2	0.01	0.998	1.44	0.606	1.54	0.594	7
3	0.01	0.998	1.33	0.643	1.79	0.572	7
4	0.01	0.997	1.47	0.610	2.21	0.553	7
5	0.01	0.999	1.20	0.637	1.08	0.665	7
Average					*1.61*	*0.589*	

TABLE A.3: Results of the 5-fold cross-validation using a fixed epoch size of 7 and a L2 regularization of 0.0001.

Fold	**Training**		**Validation**		**Testing**		**Epochs**
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	
1	0.67	0.999	1.97	0.633	2.15	0.558	7
2	0.67	0.998	2.03	0.620	2.08	0.599	7
3	0.65	0.998	1.89	0.634	2.22	0.590	7
4	0.67	0.998	2.08	0.607	2.76	0.565	7
5	0.67	0.999	1.82	0.629	1.74	0.664	7
Average					*2.19*	*0.613*	

TABLE A.4: Results of the 5-fold cross-validation using a fixed epoch size of 3 and a dropout of 0.5.

Fold	**Training**		**Validation**		**Testing**		**Epochs**
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	
1	0.04	0.997	1.21	0.573	1.23	0.552	3
2	0.03	0.997	1.34	0.603	1.10	0.610	3
3	0.03	0.997	1.22	0.589	1.41	0.571	3
4	0.03	0.996	1.43	0.554	1.77	0.536	3
5	0.03	0.998	1.10	0.642	0.93	0.676	3
Average					*1.29*	*0.589*	

TABLE A.5: Results of the 5-fold cross-validation using a fixed epoch size of 3 and a L2 regularization of 0.0001.

Fold	**Training**		**Validation**		**Testing**		**Epochs**
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	
1	0.85	0.998	2.04	0.593	2.10	0.562	3
2	0.85	0.998	2.21	0.609	1.90	0.612	3
3	0.84	0.998	2.09	0.576	2.24	0.584	3
4	0.85	0.998	2.27	0.560	2.68	0.536	3
5	0.85	0.999	1.94	0.641	1.80	0.665	3
Average					*2.14*	*0.592*	

Declaration of Authorship

Ich erkläre hiermit gemäß § 17 Abs. 2 APO, dass ich die vorstehende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Bamberg, den 31.08.2018 Unterschrift: _____