An Evolutionary-based Generative Approach for Audio Data Augmentation

Silvan Mertes
HCM, University of Augsburg,
Germany
mertes@hcm-lab.de

Alice Baird
EIHW, University of Augsburg,
Germany
alice.baird@informatik.uni-augsburg.de

Dominik Schiller
HCM, University of Augsburg,
Germany
schiller@hcm-lab.de

Björn W. Schuller
EIHW, University of Augsburg, Germany
GLAM, Imperial College London, UK
schuller@informatik.uni-augsburg.de

Elisabeth André
HCM, University of Augsburg,
Germany
andre@hcm-lab.de

Abstract—In this paper, we introduce a novel framework to augment raw audio data for machine learning classification tasks. For the first part of our framework, we employ a generative adversarial network (GAN) to create new variants of the audio samples that are already existing in our source dataset for the classification task. In the second step, we then utilize an evolutionary algorithm to search the input domain space of the previously trained GAN, with respect to predefined characteristics of the generated audio. This way we are able to generate audio in a controlled manner that contributes to an improvement in classification performance of the original task. To validate our approach, we chose to test it on the task of soundscape classification. We show that our approach leads to a substantial improvement in classification results when compared to a training routine without data augmentation and training with uncontrolled data augmentation with GANs.

Index Terms—sound generation, data augmentation, evolutionary computing, latent vector evolution, generative adversarial networks

I. Introduction

Many current trends in the area of audio signal processing are relying on data driven machine learning approaches to achieve state of the art results [1]–[3]. However, the achieved performance for a task is heavily dependent on the quantity and the quality of available data. Depending on the specific task, such data can often be hard to obtain and costly to label particularly in the audio domain. As a consequence, researchers often have to deal with datasets of insufficient size or quality. A solution to this problem is posed by data augmentation, a process that artificially creates new input data from existing samples that are altered in a way that they differ from the original sample while still maintaining the information that is relevant for the respective task at hand. However, the complex sequential structure of audio data makes this creation of artificial samples a challenging task by itself. While common approaches to create augmented audio data rely on altering the already existing data with techniques such as pitch shifting, noise injection or time stretching [4], recent research has shown the feasibility of *Generative Adversarial Networks* (GANs) to create realistic new artificial data.

However, the drawback of conventional GANs is that the output is predominantly generated randomly. This leads to an uncontrolled enlargement of training data, which may not have any impact on the training of a classifier, or even amplifying its weaknesses when dealing with small datasets, as we will show later in this work.

In this paper, we propose a novel two-step approach to address this problem. In the first step, we utilize a GAN framework to create highly realistic audio data. In the second step, we then apply an evolutionary algorithm to search the input-space of the generative model for vectors that result in samples that have specific predefined characteristics. These characteristics represent information that is lacking in the original source data of the respective classes. The concrete feature values that shall be exhibited by the new data are determined by analyzing samples that were previously classified wrong. This way, the GAN is employed to only generate training samples that are useful for a specific classification task.

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

II. Related Work

Multiple variants of GANs have been used previously to generate highly realistic audio data [5]–[7]. Further modifications, such as, conditional GANs, enable the generation of audio data that exhibits specific characteristics (e.g., [8]). However, these systems require labelled training data for each desired target characteristic of the generated data. In return,
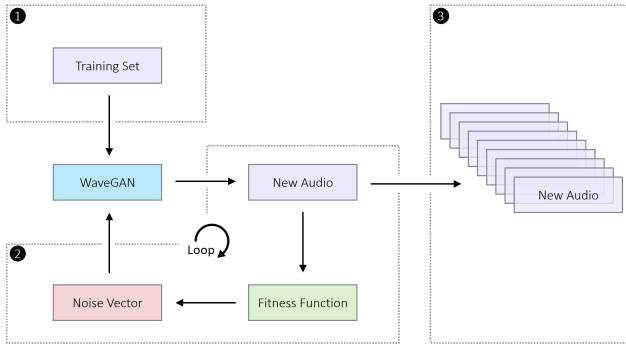
Fig. 1. Overview of our approach. (1) A WaveGAN model is trained on our dataset. (2) An evolutionary algorithm is used to find appropriate noise vectors to create new audio samples that exhibit predefined characteristics. (3) Those new audio samples are collected and taken as augmented data to enhance the existing dataset.

this means that the network needs to be trained from scratch each time a change in the target properties of the data is required. Artificial data that was generated by GANs has been used for data augmentation predominantly in the field of image processing (e.g., [9]–[11]), but there is also recent work that makes use of GAN-based data augmentation for acoustic scene classification [12]–[14] as well as emotional speech [15], [16]. Most of these approaches are not able to generate the augmented data in a controlled manner, but rather use the GANs to produce random new samples to enhance existing datasets. As could be shown in the respective publications, such GAN based augmentation techniques are a promising approach. However, existing experiments in the audio domain did only operate on rather big datasets and therefore leave open the question of whether uncontrolled data augmentation with GANs can also be applied to rather small datasets.
A recent approach to address the controllability of GANs relies on the application of evolutionary algorithms to search through the solution space of GANs and find appropriate samples that match the required characteristics, i.e., predefined feature values that shall be exhibited. Thus, the randomness of the generated samples can be overcome. This so called *Latent Vector Evolution* (LVE) has been successfully employed for tasks like fingerprint-based biometric systems, creation of video games or facial composite generation [17]–[21]. The ability to generate samples in a targeted way makes LVE a promising approach to enhance datasets with data that is actually meaningful for the respective classification task. To the best of our knowledge, there is no prior work that uses the principles of LVE for either raw audio or even the task of data augmentation.

III. APPROACH

Our proposed approach deals with the problem of augmenting raw audio datasets in a controlled manner, using generative adversarial networks in two steps. To this end, we generate artificial data samples which exhibit characteristics that are underrepresented in the original dataset. In the first step, we

train a WaveGAN architecture to produce new samples of a certain class using random noise vectors as input. In the second step, we use an evolutionary algorithm to search the input space of the WaveGAN for vectors that result in samples that show the desired feature values. An overview of the system is shown in Fig. 1. This section describes both the used WaveGAN architecture as well as the evolutionary algorithm.

A. GAN

The basic idea of GANs is to combine two competing networks, that improve each other. One network, the *generator* learns to transform any vector that follows a given distribution function, e.g., a uniform distribution, to an output sample that follows the distribution of a given training domain. The second network, the *discriminator* learns to distinguish between real training data and the samples produced by the generator [22]. During the training stage, the generator tries to fool the discriminator. Therefore, both networks compete against each other, hence, the *adversarial* part in the name.
The WaveGAN architecture was first introduced by Donahue et al. [5]. The authors showed that the system is capable of generating realistic sounding audio data for tasks that are related to nature soundscapes, such as bird sounds. Its main concepts follow the basic idea of *Deep Convolutional GANs* (DCGANs), that are a modification to the initial GANs which enable the modelling of data with even higher complexity by including convolutional layers to both the generator and the discriminator network [23]. As DCGAN was developed for image generation, multiple parts of it are slightly modified by WaveGAN to enable the handling of audio data. For example, the two-dimensional up- and downsampling filters are replaced by its one-dimensional equivalent (i.e., kernels of size $n \ times n$ become kernels with size $n * n$). For details, please refer to the original work about WaveGAN [5] and its official repository[1].
It is worth mentioning that our further approach is independent from the chosen GAN architecture. As a result, the underlying GAN architecture of the first step that is responsible for generating new audio files, can be replaced for any other model, depending on the scope of the respective application.

B. Evolutionary Algorithm

After training a WaveGAN model on a specific domain that model is able to transform random noise vectors to audio samples that follow the distribution of the training dataset. Thus, new audio samples can be generated that never had been heard before but sound as if they originated from the learnt domain. To find audio samples that show certain feature characteristics that we want to control, we follow the idea of Latent Vector Evolution to search through the solution space of the trained WaveGAN model. First, we initialize a starting population of random noise vectors and feed them to the trained WaveGAN. Subsequently, we evaluate the resulting audio data by using a predefined fitness function that measures

---

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

how appropriate the samples are with respect to the feature values that we want to have, i.e., feature values that add information to our training dataset for the classification stage, as is described in more detail in section IV. The noise vectors that performed best are then slightly mutated and recombined, whilst the other noise vectors are being discarded. The new noise vectors that originate by the mutation of the best prior noise vectors can then be fed to the trained WaveGAN again. This process is repeated until audio samples are found that show the desired feature value. The procedure is described more formally in the following.

Let x be the output of the generator, denoted as $x = G(z)$, where the function G represents the transformation learnt by the generator that takes a latent space vector z as input. Further, we define a measurement function $f(x)$ that calculates the value of the feature that we want to control. Thus, $f(G(z))$ corresponds to the feature value that is achieved by feeding z to the generator. We denote the value that we want the feature to be as t and call this the target value. Thus, a *perfect* noise vector z for target value t would fulfill $f(G(z)) = t$.

As described above, we chose the best noise vectors to be mutated and recombined further by a fitness function. We can find such a fitness function $fitness(z)$ easily by constructing the reciprocal of the distance between the shown feature value and the target value:

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

We use this fitness function to train an Evolutionary Algorithm. The term Evolutionary Algorithms denotes a class of optimization methods that are inspired by the evolution of natural living beings. In general, they work by iteratively generating a new set of data samples (*individuals*), a so called *population*, and choosing the best out of these samples (*selection*) regarding a predefined fitness function, and alter those samples in various ways to get a new population that is (hoped to be) better than the previous one (*Mutation and Recombination*). *Evolution Strategies* (ES) are a group of evolutionary algorithms that are mainly used for multidimensional, continuous problems [24]. This property makes them an ideal fit to operate on the latent vector that is used as input for the GAN. Specifically, we chose a $(\mu/p + \lambda)$-ES. This means, that the best μ individuals of the parent population generate λ new individuals, whereas the parent individuals are also included into the following generation of individuals. p represents the group size of the recombination, i.e., p individuals of the parent population are responsible for the creation of a new individual simultaneously. For our evaluation, we choose the following, empirically determined, parameters: $\mu = 50$, $p = 2$, $\lambda = 150$.

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

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

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

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

Here, $\mathcal{N}(0, \sigma^2)$ is the mutation value that is randomly sampled from a Gaussian Distribution, where the variance σ^2 is chosen according to the *1/5 Success Rule* [24].

IV. EXPERIMENTS

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

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

A. Methodology

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

---

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

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

augmented data over random GAN-based augmented data. The three datasets are discussed in detail in the following sections.

1) Original Dataset (dataset_orig): For evaluation purposes, we chose to perform our experiments on a subset of the Emotional Soundscapes database [25]. The dataset contains audio files of certain soundscapes which are sorted by environment. We decided to consider only the two classes of *mechanical* and *natural* environments, as the samples of these classes, despite their fundamental differences, generally have a very noisy appearance, which makes them often hard to distinguish even for humans. For example, it can be very hard to differentiate between a waterfall and the background noise of a room full of different kinds of machines, since both sounds are similar in terms of their low frequency range and regular noisiness. The nature class has many samples that contain large parts of silence. As these samples would complicate the feature extraction as well as the WaveGAN training, we removed them from the dataset, resulting in an increase in mean RMS energy level of the nature class from $2.18*10^{-2}$ to $2.64*10^{-2}$. We split all audio files into samples of 1 second length as our WaveGAN architecture produces outputs of a fixed size. Our final first dataset contains 600 samples (10 minutes) for the mechanical class and 300 samples (5 minutes) for the nature class. As mentioned above, the dataset was split into train, development and test partitions. The train partition for this dataset contains 420 samples (7 minutes) for mechanical and 210 samples (3.5 minutes) for natural. The development partition contains 60 samples (1 minute) for mechanical and 30 samples (0.5 minutes) for natural, while the test partition contains 120 samples (2 minutes) for mechanical and 60 samples (1 minute) for natural. Data augmentation, as described below, is only applied to the train partition.

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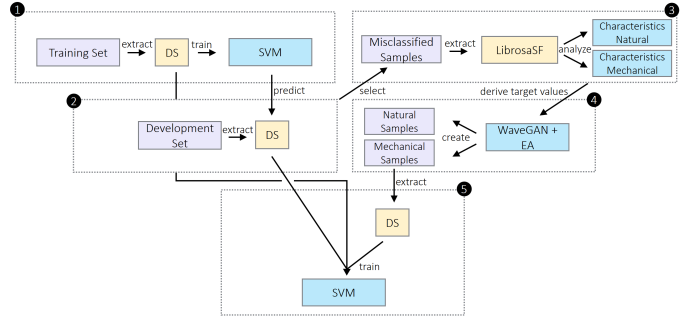

Fig. 2. Illustration of the Augmentation Process. (1) The SVM is trained on the training partition of the original data using Deep Spectrum (DS) features. (2) The trained SVM is used to predict samples from the development partition. (3) The misclassified samples are analyzed regarding six standard spectral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

3) Targeted Augmentation (dataset_aug_ctrl): The third dataset contains the original data from *dataset_orig* as well as audio samples that were generated in a targeted way by the use of our approach. As described in section III-B, we make use of an evolutionary algorithm to find the samples that show the feature values that we want to have. Our assumption is the following: if a classifier is not able to classify certain samples in a correct way, then it might lack training data that shows similar feature values as the ones that were classified wrongly. We therefore aim to generate new training data that exhibits feature characteristics of the previously wrong classified data. To find appropriate target values for the evolutionary algorithm, we analyze the samples of the development set that were classified wrongly by the use of the SVM that was trained only on *dataset_orig*. Specifically, we look at six standard spectral features (*Spectral Centroid, RMS, Spectral Bandwidth, Spectral Contrast, Spectral Flatness and Spectral Rolloff*). It is noteworthy that this feature set is only used to select the specific audio samples, but not for the classification itself. To this end, we rely on the DEEP SPECTRUM feature set described in section IV-A4.

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

be the mean and $s_{i,c}$ the standard deviation of the feature i for class c. We calculated our target values $t_{i,c}^1, t_{i,c}^2, ... t_{i,c}^5$ for the respective feature and class as follows:

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

With these target values, we are able to cover a big range of the feature values that were missing in the initially wrong classified data samples. We trained the evolutionary algorithm with the respective target function for each of the 30 samples per class, resulting in 60 runs of the evolutionary algorithm. The feature values of all individuals were calculated with the *librosa* [26] library during training. Every training run was stopped after 100 iterations.

4) Deep Spectrum Features: As the nature of soundscapes is complex, we choose a spectrogram based approach for extracting features that are used as input for the classification stage. We assume that this would inherently capture a larger portion of temporal information as compared to other conventional acoustic features. For this, we extract a 4 096 dimensional feature set of deep data-representations using the DEEP SPECTRUM toolkit [27][3]. DEEP SPECTRUM has shown success for similar audio tasks [16], and extracts features from the audio data using pretrained convolutional neural networks. For this study, we extract spectrograms using the default DEEP SPECTRUM settings including a VGG16 pretrained network, extracting one feature vector per audio sample.

5) Support Vector Machine: For all machine learning experiments, we use a Support Vector Machine with a linear kernel. During the development phase, we trained a series of SVM models, optimizing the complexity parameters ($C \in 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1$) and evaluating their performance on the development partitions. We then re-train the model with the concatenated train and development partitions and evaluate the performance on the test partition. This whole procedure is made for each of the three datasets. As a measure of accuracy we report Unweighted Average Recall (UAR) as this considers the class imbalance that is present in the data.

V. RESULTS

After we trained the SVM on the three datasets, we evaluated all models on the test partition. In this section, we report our results for each dataset. As mentioned above, we report the UAR that each of our SVM models achieved with the respective optimal complexity parameter. As the task is to perform a binary classification, the chance level is represented by a UAR of 50.0. Our baseline model that was trained on the original data (*dataset_orig*) achieves an accuracy of 75.8% UAR. *Dataset_aug*, that contains randomly generated samples among the original data, results in a UAR of 71.2%. As can be seen, this value is remarkably below the first model. This leads to the deduction that the data that was generated by feeding
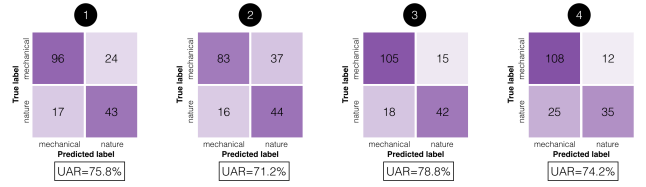
[3]https://github.com/DeepSpectrum/DeepSpectrum

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

random noise vectors to the GAN does not add meaningful information to the SVM during the training process, even making the training set worse. This shows the need for the generation in a controlled way, as applied in *dataset_aug_ctrl*. The model that was trained on the targeted augmented data achieves a UAR of 78.8%, thus outperforming both the classifiers that were trained on *dataset_orig* and *dataset_aug*. To evaluate this effect further, we trained a fourth model only on the targeted augmented data without the original data. This model achieves a UAR of 74.2%, thus being slightly below the baseline model. This is reasonable since the target values for the Evolutionary Algorithm were derived by analyzing the false classified samples from the classifier that was trained on *dataset_orig*, thus trying to specifically add information for value ranges that are not yet modelled in that dataset, but not claiming to be able to model the whole possible range of the features of the original data. The corresponding confusion matrices for each model are shown in Fig. 3.

VI. DISCUSSION

When comparing the results of *dataset_orig* that only contains the original audio samples, with *dataset_aug* that adds randomly generated data to the training process, it can be seen that the performance of the trained SVM model considerably drops. This shows, that there is in fact a need for augmentation in a somewhat targeted way, although recent works could also achieve performance boosts while working with a random generation process [12]. It is conceivable that the random generation in our problem domain is not sufficient due to the fact that our original dataset is very small compared to the datasets that were used in those previous works. However, the results that could be achieved with *dataset_aug_ctrl* outperform both the models from *dataset_orig* and *dataset_aug*. As *dataset_aug_ctrl* makes use of our controlled generation process, it is capable of adding augmentation data that is actually helping the classification task. Although the solution space that was learnt by the WaveGAN has to be rather small as we used only small amounts of data to train it, the Evolutionary Algorithm was able to find meaningful samples in that solution space. It is noteworthy that even considerably less training samples were generated for *dataset_aug_ctrl* than for *dataset_aug*. This shows that even small amounts of targeted augmentation data are better for the classification task than high amounts of randomly generated data.

VII. Conclusions and Outlook

In this paper, we presented a new approach for augmenting training data for an audio classification problem in a targeted way. We showed that our approach has substantial advantages for our problem domain when comparing it to adversarial augmentation techniques that rely on a random class-wise augmentation.

In the future, we plan to investigate the applicability of our approach to other datasets and problem domains to draw further conclusions about the generality of our approach. Furthermore, it would be interesting to see the performance of the approach in a multi-class problem. It is also worth to mention that this work focused on the comparison between random augmented data and targeted augmented data. For future work, it could be interesting to also compare the performance of traditional augmentation techniques and other state-of-the art augmentation techniques to our approach. Thus, we plan to do a bigger study that also takes other augmentation methods, bigger datasets and even different classifiers like neural networks into account. Another topic that has to be investigated is the possibility of iteratively repeating the proposed approach by using the results of the targeted augmented training set as input for further evolutionary algorithm steps. Besides that, future work should examine the optimal amount of targeted augmented data. We can summarize that in our chosen problem domain, the approach works reasonably well and shows potential to improve a broad range of classification problems that are existent in the current research community.

VIII. Acknowledgements

References

[1] J. Wagner, D. Schiller, A. Seiderer, and E. André, "Deep learning in paralinguistic recognition tasks: Are hand-crafted features still relevant?" in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana, Ed. ISCA, 2018, pp. 147–151.

[2] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2721–2725.

[3] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, p. 150, 2018.

[4] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[5] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," *arXiv preprint arXiv:1802.04208*, 2018.

[6] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," *arXiv preprint arXiv:1902.08710*, 2019.

[7] P. Chandna, M. Blaauw, J. Bonada, and E. Gómez, "Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[8] C. Y. Lee, A. Toffy, G. J. Jung, and W.-J. Han, "Conditional wavegan," *arXiv preprint arXiv:1809.10636*, 2018.

[9] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw, and D. Rueckert, "Gan augmentation: Augmenting training data using generative adversarial networks," *arXiv preprint arXiv:1810.10863*, 2018.

[10] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "Bagan: Data augmentation with balancing gan," *arXiv preprint arXiv:1803.09655*, 2018.

[11] S. Mertes, A. Margraf, C. Kommer, S. Geinitz, and E. André, "Data augmentation for semantic segmentation in the context of carbon fiber defect detection using adversarial learning," in *Proceedings of DeLTA 2020: 1st International Conference on Deep Learning Theory and Applications*, 2020.

[12] A. Madhu and S. Kumaraswamy, "Data augmentation using generative adversarial network for environmental sound classification," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[13] J. H. Yang, N. K. Kim, and H. K. Kim, "Se-resnet with gan-based data augmentation applied to acoustic scene classification," in *DCASE 2018 workshop*, 2018.

[14] S. Mun, S. Park, D. K. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using svm hyper-plane," *Proc. DCASE*, pp. 93–97, 2017.

[15] G. Rizos, A. Baird, M. Elliott, and B. Schuller, "Stargan for emotional speech conversion: Validated by data augmentation of end-to-end emotion recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3502–3506.

[16] A. Baird, S. Amiriparian, and B. Schuller, "Can deep generative audio be emotional? towards an approach for personalised emotional audio generation," in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2019, pp. 1–5.

[17] P. Bontrager, A. Roy, J. Togelius, N. Memon, and A. Ross, "Deepmaster-prints: Generating masterprints for dictionary attacks via latent variable evolution," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2018, pp. 1–9.

[18] V. Volz, J. Schrum, J. Liu, S. M. Lucas, A. Smith, and S. Risi, "Evolving mario levels in the latent space of a deep convolutional generative adversarial network," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 221–228.

[19] E. Giacomello, P. L. Lanzi, and D. Loiacono, "Searching the latent space of a generative adversarial network to generate doom levels," in *2019 IEEE Conference on Games (CoG)*. IEEE, 2019, pp. 1–8.

[20] J. Schrum, J. Gutierrez, V. Volz, J. Liu, S. Lucas, and S. Risi, "Interactive evolution and exploration within latent level-design space of generative adversarial networks," *arXiv preprint arXiv:2004.00151*, 2020.

[21] N. Zaltron, L. Zurlo, and S. Risi, "Cg-gan: An interactive evolutionary gan-based approach for facial composite generation," *arXiv preprint arXiv:1912.05020*, 2019.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[23] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[24] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies–a comprehensive introduction," *Natural computing*, vol. 1, no. 1, pp. 3–52, 2002.

[25] J. Fan, M. Thorogood, and P. Pasquier, "Emo-soundscapes: A dataset for soundscape emotion recognition," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2017, pp. 196–201.

[26] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[27] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, and B. Schuller, "Snore sound classification using image-based deep spectrum features," in *Proc. of INTERSPEECH 2017*. Stockholm, Sweden: ISCA, August 2017, pp. 3512–3516.