Conditional GAN-based EEG Data Augmentation

Rishi Chandra
rchand18@jhu.edu

Abstract

The acquisition of high-quality neurological data is essential for advancing our understanding of brain function and developing effective diagnostic and therapeutic interventions for neurological disorders. However, obtaining sufficient amounts of such data can require expensive experimental trials, and yield imbalanced data results. Generative models, particularly Conditional Generative adversarial networks (cGANs), are recently highly successful in generative applications in both image and time-series data generation. In this work, I propose two cGAN architectures to create synthetic EEG data in 1-dimensional time-series form, and 2-dimensionsl spectrogram form. The generated data quality was evaluated using a trained classifier, and then assessed for its usability in data augmentation. Results showed that the generated data in 2-dimensions was effective in improving the performance of the classifier via data augmentation. This work demonstrates the potential of GAN-based data augmentation for improving the quality and quantity of EEG data, and for enhancing the performance of EEG-based classification models.

1 Introduction

Electroencephalogram (EEG) signals have been increasingly used in a variety of applications, including emotion recognition, cognitive assessment, and brain-computer interfaces. However, a major challenge in EEG research is the limited availability of quality data, which can hinder the development and performance of machine learning models. Data augmentation is a commonly used technique to address this issue, where synthetic data is generated from the original data to increase its quantity and diversity.

In recent years, generative adversarial networks (GANs) [1] have shown promising results in generating realistic synthetic data for various applications, including image and time-series data. Particularly, the conditional GAN (cGAN) [2] architecture demonstrates how GANs can be leveraged to produce controllable outputs with the addition of a conditioning variable. This architecture allows for the generation of class-specific synthetic EEG data, which can subsequently be used in classification tasks.

The experimental question for this paper is whether GAN-based data augmentation can effectively improve the performance of EEG-based emotional sentiment classification. I use a dataset of EEG signals collected during the viewing of emotional stimuli, which includes labeled data for 3 different emotional states. I train two cGAN architectures to generate synthetic EEG data as both 1-dimensional signals and 2-dimensional spectrograms, and evaluate the generated data for each method using a trained classifier. I also assess the usability of the generated data for data augmentation by evaluating the performance of a classifier prior to and after training with an augmented dataset.

2 Dataset and Features

I used the EEG Brainwave Dataset: Feeling Emotions [3]. The experiment presented subjects with visual stimuli in the form of movie clips to evoke different emotional valence. The data was collected from two people (1 male, 1 female) for 3 minutes per emotional state - positive, neutral, and negative. The data was collected via a Muse EEG headband which recorded the TP9, AF7, AF8 and TP10 EEG placements via dry electrodes, resampled with a variable frequency of 150Hz. The resulting data yielded 2132 EEG samples - 708 positive, 708 negative, and 716 neutral. Each sample included a signal of length 750 samples, along with 1798 manually extracted features, as detailed in the paper [4].

For 1-D signal generation and classification, the signals were shifted and normalized, and the full 2548-length vectors (1798 features + 750 sample signal) were classified and generated. For 2-D signal generation and classification, the signals were also shifted and normalized, and a spectrogram was generated from the 750-sample signal via Short-time Fourier Transform (STFT) to produce a 129×22 image. In the 2-D classification and generation experiment, the 1798 manually extracted features were omitted.

3 Methods

For both 1-D and 2-D generation, the training and evaluation followed a 6-step process, as outlined in **Fig 1**. First, a classification model was trained to predict the emotion class, and the best test accuracy was recorded. A 0.9/0.1 train-test split was used. The cGAN was

Reading the Mind: Computational Cognitive Neuroscience of Vision 2023, (LateX template borrowed from NeurIPS.)

then trained to create synthetic class-conditioned samples. To evaluate the representative quality of the synthetic data, the pretrained classification model was used for inference on the synthetic data. This process was repeated with varying cGAN hyperparameters and architectures until inference accuracy on the synthetic data was comparable with inference accuracy on the real dataset.

Once the cGAN produced samples of satisfactory quality as measured by the inference evaluation, it was then evaluated for usability in data augmentation. A classification model was trained on a small subset of the data, and the best test accuracy (evaluated on the rest of the data) was recorded to serve as a baseline. The cGAN was reinitialized and trained on the same subset of data. The classifier was then trained on synthetic data generated from the cGAN, and the best classification performance after synthetic data training was recorded.

3.1 1-D Signal Generation

An LSTM was used for 1-D signal classification. The architecture and hyperparameters are depicted in **Fig 2.** below. An LSTM-based cGAN was used for 1-D signal generation, as depicted in **Fig 3.** below. The LSTM architecture was used for its advantageous performance in evaluating time-series data [5] compared to the traditional RNN and for its straightforward implementation. The classifier was trained for 200 epochs and an early stopping patience of 20 epochs. The cGAN was trained for 500 epochs.

3.2 2-D Spectrogram Generation

A CNN was used for 2-D spectrogram classification. The architecture and hyperparameters are depicted in **Fig 4.** below. A deep convolutional GAN (DCGAN) [6] was used for 2-D spectrogram generation, as depicted in **Fig 5.** below. 2-D generation and classification was evaluated in addition to 1-D due to the improved efficiency of convolutional networks relative to recurrent networks, as well as the potential to learn relevant features via the spatial and temporal dimensions of EEG images. The model was trained for 200 epochs and an early stopping patience of 20 epochs. The cGAN was trained for 500 epochs.

4 Results

4.1 1-D Signal Generation

The results for initial evaluation of the LSTM-based GAN. as well as some generated samples, are shown in **Fig. 6** below. The LSTM classifier achieved a test accuracy of $\mathbf{97.18\%}$ on the full dataset. After training the LSTM-based cGAN on the full dataset, the trained LSTM classifier achieved a test accuracy of $\mathbf{90.12\%}$ across 1000 synthetic samples.

The results for data augmentation evaluation of the LSTM-based GAN are shown in **Fig. 7** below. A 25% training subset of the data was created, and both the LSTM classifier and LSTM-based GAN were reinitialized and trained on this subset. The LSTM classifier achieved $\mathbf{92.19\%}$ test accuracy after training on the subset. The accuracy remained at $\mathbf{92.19\%}$ after synthetic data training.

4.2 2-D Spectrogram Generation

The results for initial evaluation of the DCGAN. as well as some generated samples, are shown in **Fig. 8** below. The CNN classifier achieved a test accuracy of $\mathbf{90.16\%}$ on the full dataset. After training the DCGAN on the full dataset, the trained LSTM classifier achieved a test accuracy of $\mathbf{84.39\%}$ across 1000 synthetic samples.

The results for data augmentation evaluation of the DCGAN are shown in Fig. 9 below. A 50% training subset of the data was created, and both the CNN classifier and DCGAN were reinitialized and trained on this subset. The LSTM classifier achieved $\mathbf{86.24\%}$ test accuracy after training on the subset. The accuracy increased to $\mathbf{89.46\%}$ after synthetic data training.

5 Discussion

5.1 1-D Signal Generation

The LSTM classifier achieved a test accuracy of 97.18% on the full dataset. With 1000 synthetic siganls from an LSTM-based GAN, it achieved a test accuracy of 90.12%. The accuracy on the synthetic signals proved to be above 90%, demonstrating that the signals were consistently representative of the conditioned class. The classification accuracy did decrease on the synthetic data, which is understandable due to the potentially diminished quality of the synthetic data samples, and the difficulty in sequential data generation in a recurrent GAN architecture.

When evaluating the LSTM-based GAN for data augmentation usability using a 25% data subset, the LSTM classifier achieved 92.19% test accuracy before and after synthetic data training, showing no improvement with the introduction of data augmentation. This indicates that although the synthetic signals may have been representative of their class, there was not enough signal diversity to contribute meaningful data variance to the training set and thereby improve the test-time classification performance. This can also be attributed to the costly training of LSTM-based architectures, which meant that extensive hyperparameter testing was not possible. Additionally, the test accuracy was already quite high prior to any synthetic data training, so there was little margin for improvement.

5.2 2-D Spectrogram Generation

The CNN classifier achieved a test accuracy of 90.16% on the full dataset. With 1000 synthetic spectrograms from the DCGAN, it achieved a test accuracy of 84.39%. These accuracies are fairly consistent, and demonstrate that the spectrograms were consistently representative of the conditioned class.

When evaluating the DCGAN for data augmentation usability using a 50% data subset, the CNN classifier improved from 86.24% prior to synthetic data training to 89.46% after training with synthetic data. This $> 3\%$ increase demonstrates that the DCGAN was able to learn generalizable features from the limited data subset, producing images that provided meaningful variance to the training of the classifier that generalized to test-time performance.

5.3 Methodology and Improvements

In the 2-D CNN experiments, I chose to omit the 1798 manually selected features, which explains the lower overall accuracies across the experiments compared with the 1-D LSTM experiments. This is because test accuracies were already high without augmentation, and removing the manually selected features would produce a more realistic baseline metric to demonstrate the benefit of augmentation training, as it widens the margin for improvement. Additionally, I elected to use a 50% subset of the data rather than 25% as used with the LSTM. This is because a 25% subset did not contain enough data for the highly complex and much deeper DCGAN architecture to produce quality images.

While both the LSTM-based GAN and DCGAN demonstrated the ability to produce representative class-conditioned EEG data, only the DCGAN demonstrated the potential to produce samples with enough meaningful variance to be used for data augmentation purposes. This may be attributed to the advantages of convolutional architectures in computational efficiency and feature learning, where they do not suffer from the vanishing gradients and difficulty in long-range dependencies that are present in LSTM architectures. The DCGAN architecture was also much deeper and underwent more extensive hyperparameter testing. In addition, other research has demonstrated that 2-dimensional analysis of EEG data demonstrates improved performance relative to 1-dimensional analysis where topographical changes in brain activation are consistent across subjects [7].

Overall, this demonstrates the efficacy and usability of DCGANs for EEG data augmentation in future research and applications. Potential improvements include using the Wasserstein GAN (WGAN) [8] architecture, where the discriminator is replaced with a "critic", and the optimization aims to minimize the Wasserstein distance between the real and fake data distributions. This architecture results in more stable convergence, and encourages the generator to produce more diverse samples, which is ideal for data augmentation tasks. In other research, the WGAN has proven to be effective in EEG data augmentation [9]. Other techniques such as Weight Clipping and Gradient Penalty [10] are shown to improve GAN convergence and generative quality. Additionally, Probabilistic Diffusion [11] has demonstrated great success in high-quality conditional image generation, and could potential yield preferable results in EEG spectrogram generation tasks.

6 Code

https://github.com/rishic3/GenerateEEG

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[2] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[3] Jordan Bird, Aniko Ekart, Christopher Buckingham, and Diego Faria. Mental emotional sentiment classification with an eeg-based brain-machine interface. 04 2019.

[4] Jordan Bird, Diego Faria, Luis Manso, A. Ekárt, and Christopher Buckingham. A deep evolutionary approach to bioinspired classifier optimisation for brain-machine interaction, 08 2019.

[5] Ralf C. Staudemeyer and Eric Rothstein Morris. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 2019.

[6] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[7] Darshil Shah, Gopika Gopan K., and Neelam Sinha. An investigation of the multi-dimensional (1d vs. 2d vs. 3d) analyses of eeg signals using traditional methods and deep learning-based methods. *Frontiers in Signal Processing*, 2, 2022.

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[9] Kay Gregor Hartmann, Robin Tibor Schirrmeister, and Tonio Ball. Eeg-gan: Generative adversarial networks for electroencephalographic (eeg) brain signals, 2018.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

7 Figures

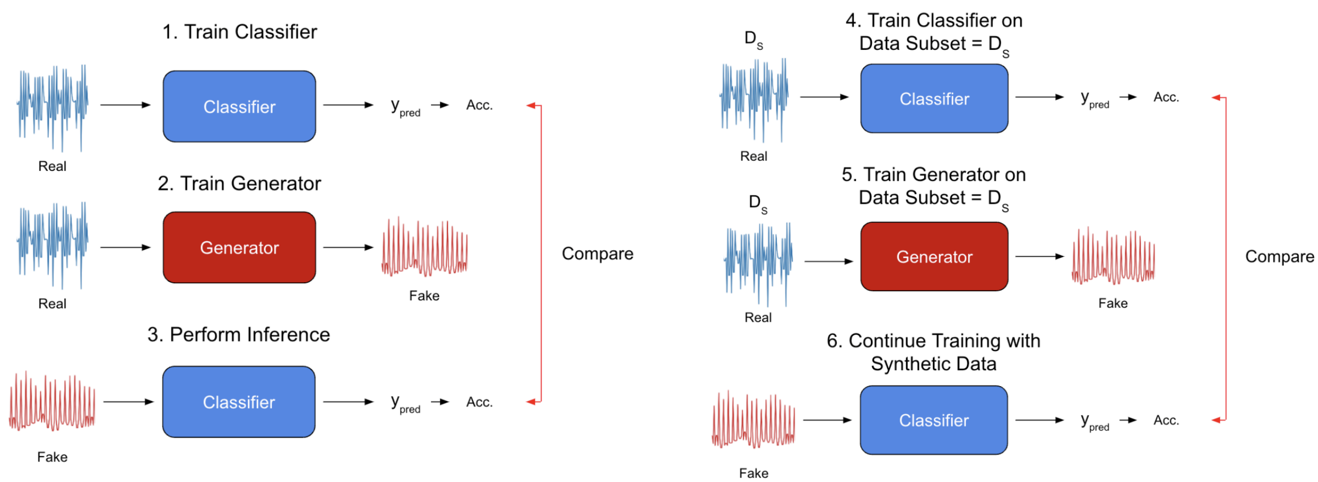

Figure 1: Training and evaluation process for the EEG cGAN. Both the classifier and generator are trained on the entire dataset. The classifier accuracy on the real and synthetic data is compared to evaluate the synthetic data quality.
Next, both the classifier and generator are reinitialized and trained on a small subset of the data. The classifier is then trained on a synthetic data produced by the generator, and the performance before and after synthetic data training is compared.

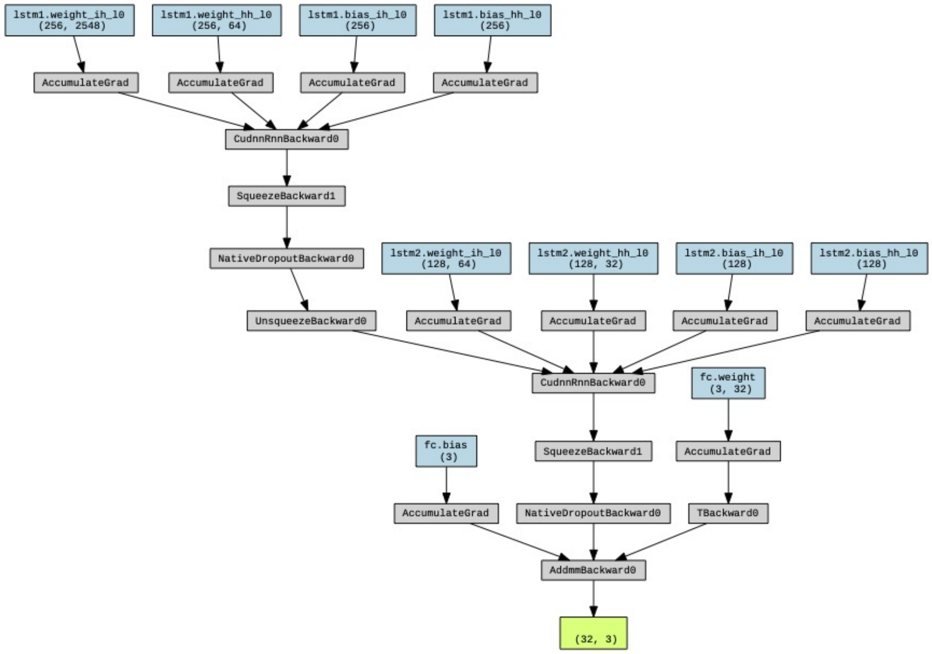

Figure 2: LSTM classifier architecture. Contains an LSTM layer with 64 hidden nodes, followed by a dropout layer with $p = 0.2$, a second LSTM layer with 32 hidden nodes, another dropout layer with $p = 0.2$, and a fully connected layer mapping to the 3-class prediction. The architecture was trained with all-zero initialization, Adam optimization [12], and a learning rate of $1e^{-3}$.

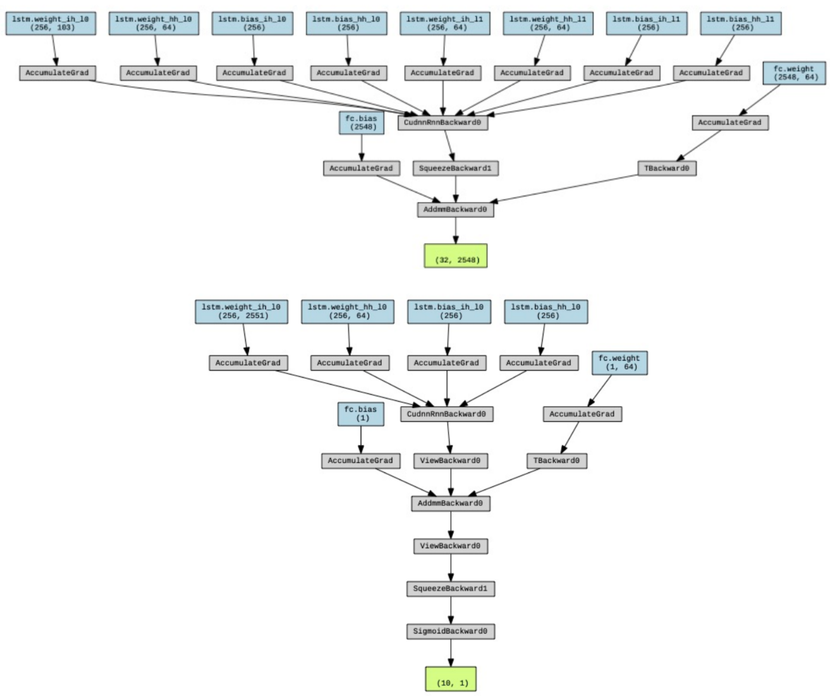

Figure 3: LSTM-based cGAN architecture. Both the generator and discriminator contain 2 LSTM layers each with 64 hidden nodes. The architecture was trained with all-zero initialization, Adam optimization, and a learning rate for both the generator and discriminator of $1e^{-4}$.

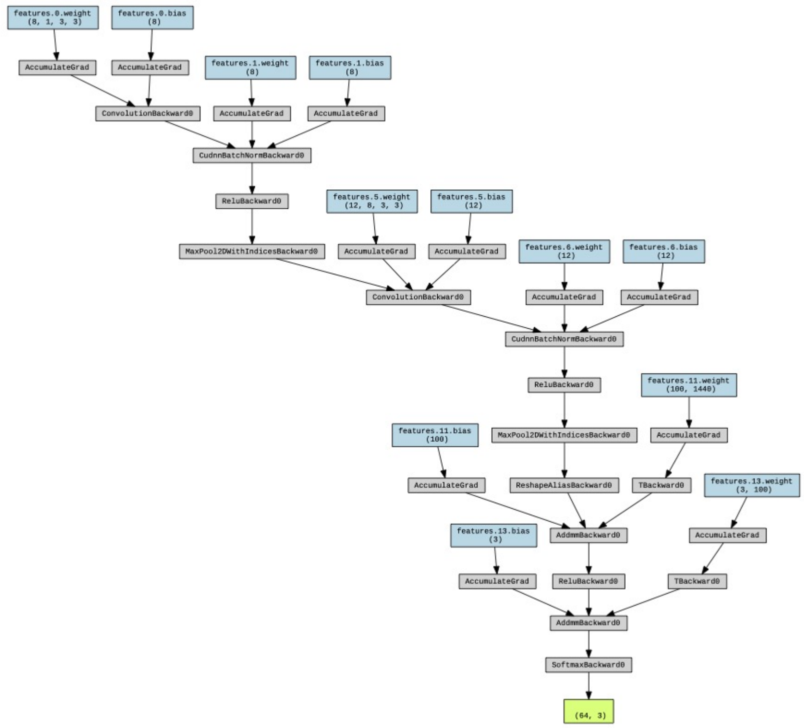

Figure 4: CNN classifier architecture. Contains 2 convolutional blocks, each with batch-norm and maxpooling layers, each with 3×3 kernels. This is followed by a dropout layer with $p = 0.25$, and two fully-connected layers to yield a class prediction. The architecture was trained with Kaiming initialization [13], Adam optimization, and a learning rate of $1e^{-3}$.

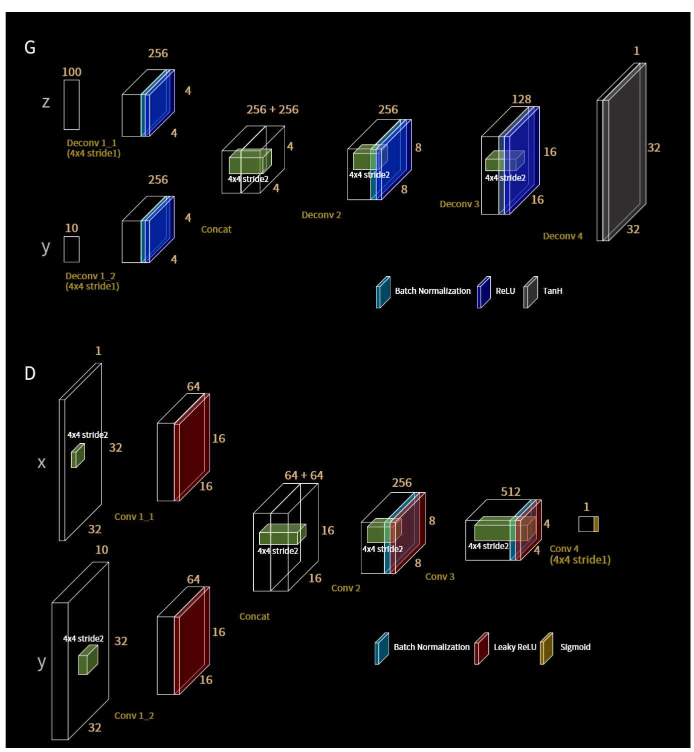

Figure 5: DCGAN architecture. The architecture is further detailed in the DCGAN paper [6]. An interpolation layer was included to map the image to a rectangular 129×22 spectrogram. The architecture was trained with Kaiming initialization, Adam optimization, and a learning rate for both the generator and discriminator of $1e^{-4}$.

LSTM Test Accuracy on Real Data	**97.18%**
LSTM Test Accuracy on Synthetic Data	**90.12%**

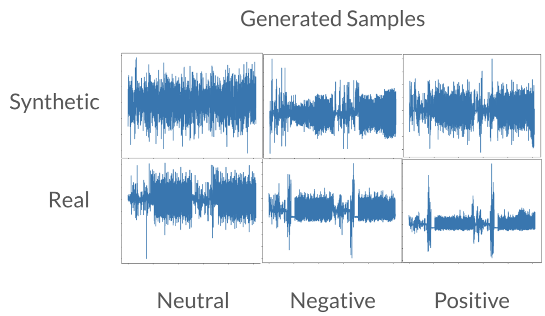

Figure 6: Left: LSTM-based GAN results, evaluated via inference from a trained LSTM classifier. Accuracy on the real dataset and 1000 synthetic samples were 97.18% and 90.12% respectively.
Right: Samples of generated signals compared to real signals and their corresponding classes.

LSTM Test Accuracy w/ 25% Data Training	**92.19%**
LSTM Test Accuracy w/ Synthetic Data Training	**92.19%**

Figure 7: LSTM-based GAN data augmentation results, evaluated via performance of an LSTM classifier before and after augmentation training. Accuracy on the data subset was 92.19% before and after the addition of 2000 synthetic samples in training.

CNN Test Accuracy on Real Data	**90.16%**
CNN Test Accuracy on Synthetic Data	**84.39%**

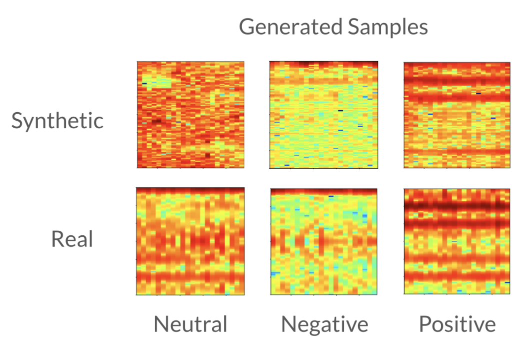

Figure 8: Left: DCGAN results, evaluated via inference from a trained CNN classifier. Accuracy on the real dataset and 1000 synthetic samples were 90.16% and 84.39% respectively.
Right: Samples of generated spectrograms compared to real spectrograms and their corresponding classes.

CNN Test Accuracy w/ 50% Data Training	**86.24%**
CNN Test Accuracy w/ Synthetic Data Training	**89.46%**

Figure 9: DCGAN data augmentation results, evaluated via performance of an CNN classifier before and after augmentation training. Accuracy on the data subset was 86.24% before and increased to 89.46% after the addition of 2000 synthetic samples in training.