CS230

SpeakEasy: Accent Reduction through Neural Style Transfer

Shelly Gamlielly
Stanford University
Shellyga@stanford.edu

Elad Altaras
Stanford University
Elad23@stanford.edu

Abstract

This project aims to develop a deep learning-based solution to reduce the accent of non-native speakers of English. The goal is to improve the ability of non-native speakers to communicate more easily by enhancing the clarity and intelligibility of their speech.

1 Introduction

The problem we aim to solve is the communication barriers of non-native speakers with an evident accent. This is an important issue as it can create difficulties in communication and limit opportunities for non-native speakers. We are following the approach of neural style transfer for speech, which is based on the idea of neural style transfer for images. Our input is a non-native speaker wav audio and our goal is to use NST to create an output wav audio with reduced accent of the speaker.

2 Related work

There have been various approaches that leverage deep neural networks to reduce non native speaker accent. One approach is to use NST with an encoder-decoder architecture. The encoder collects the style information from a source audio signal, and the decoder recreates the content that matches the style of the target audio signal. VGG19 is used to extract style features for the source and target signals and the goal is to minimize the difference between them. Next the output Mel-spectrogram is converted back into an audio signal using the Griffin-Lim algorithm [1]. This method has been shown to outperform existing systems in terms of mean opinion score and perceptual evaluation of speech quality, producing natural and realistic accent conversion [1]. Cycle-Consistent Adversarial Networks (CycleGAN) have been used for voice conversion by mapping between two domains without paired data [2]. This method uses a cycle-consistent loss (shown on appendix) .This loss is used to ensure that the converted voice can be reconstructed back to the original voice and preserves the linguistic content [2]. The method has been evaluated on inter-gender conversion using two datasets and has been shown to outperform two baseline methods, producing high-quality and natural-sounding converted voices [3]. Another approach that was performed by the course students in 2018 is to use NST with an accent classifier based on VGG19 [3]. They used log amplitude spectrograms where log mel spectrograms may provide a more perceptually meaningful representation of the audio signal [3]. Additionally, using pre-trained VGG19 with weights of 'ImageNet' as an accent classifier may not be optimal as it was not trained on spectrograms [3]. Moreover, they could not output audio with understandable content. In this work, we propose a method for accent conversion that leverages the advantages of Neural Style Transfer (NST) while using a more perceptually meaningful representation of the audio signal. We trained a dedicated binary classifier on mel spectrograms to discriminate between native and non-native English speakers. The classifier was trained on a limited data set for

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

simplicity, where a larger and diverse data set is also available. So by training the classifier on an expanded data set, we expect the encoder to be able to capture features that are robust to the mother language of the speaker, so this approach is adaptable to different accents and voices. Moreover, since the generated spectrogram is initialized as a content spectrogram with noise, our method allows to transfer style features while preserving the voice identity of the speaker. Additionally, our method avoids the difficulties of training and optimizing complex models such as CycleGAN, which require tuning many parameters to achieve good performance [2].

3 Dataset and Features

We used 'AccentDB' from https://accentdb.org/, a database of native and non-native English accents to assist neural speech recognition. We used two female speakers - one with a native American accent and one with a Bangla accent. AccentDB provides a large collection of audio recordings from speakers with various accents, which allows us to train and test our model effectively in the future on a larger and more diverse data set. The binary classifier was trained on 600 audio clips of each female speaker, divided into a train set of 720 clips, a validation set of 240 clips, and a test set of 240 clips. We performed data preprocessing on our dataset to ensure that each input is 10 seconds long. If an audio clip is longer than 10 seconds, we slice it into multiple 10-second clips. If an audio clip is shorter than 10 seconds, we duplicate it to reach 10 seconds length. We represented the audio using log mel spectrograms. This representation captures the important features of human speech while reducing the dimensionality of the data, making it easier for our model to process the data. The log mel scale closely matches the way our ears perceive sound, making it well-suited for representing human speech.The logarithmic scale compresses the dynamic range, reduces noise, ensures perceptual consistency, and enhances the visibility of important acoustic features.

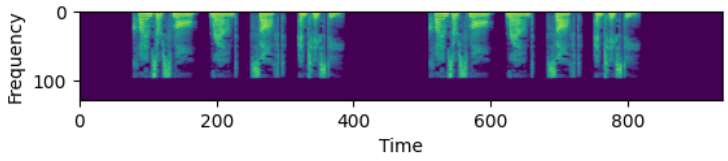

Figure 1: Example of Non Native Mel spectrogram.

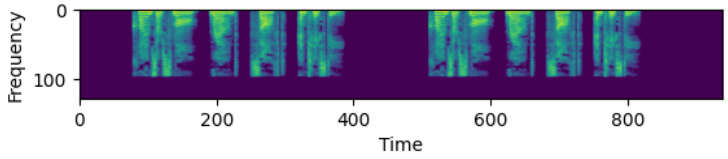

Figure 2: Example of Native Mel spectrogram.

4 Methods

Our approach involves using neural style transfer (NST) to reduce the accent of a non-native speakers while maintaining their natural voice.

4.1 Base line:

4.1.1 VGG19 model as an encoder

We first implemented NST using a pre-trained VGG19 model as an encoder to extract features from the content and style spectrograms and generate styled audio. While VGG19 offers powerful feature extraction capabilities and can be utilized as part of an NST-based approach for accent conversion, it may not fully capture audio-specific characteristics or temporal dynamics.

4.1.2 Log Mel Spectrograms Analysis

NST using log mel spectrograms and analyzing the spectrograms. This approach extracts a style matrix from a list of spectrograms by calculating and aggregating Gram matrices for each time window of each spectrogram. The style matrix represents the correlations between different frequency components in the spectrograms and can be used to capture the style or accent of the speaker.

4.1.3 Combination:

using NST with VGG19 as the encoder and extending the loss function to include additional cost for style spectrogram analysis using Gram matrices over the log mel spectrograms

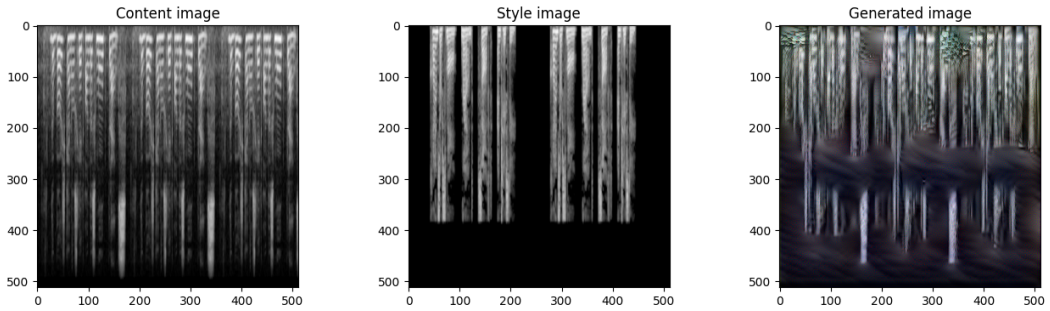

Figure 3: Contnet, Style and the Generated spectrogram with VGG19 Encoder

4.2 NST using trained native English speaker [NES] classifier

We suspected that VGG19 was not effective enough at capturing accent features. VGG19 was primarily designed for image classification tasks, not specifically for audio or accent conversion. While the model can extract useful features, it may not fully capture the unique characteristics and nuances of audio signals required for accurate accent conversion. Audio-specific models, such as those designed for spectrogram analysis or speech recognition, might be better suited for the task. Therefore, we created our own NES classifier and used it as our encoder instead. Our classifier was trained on our log mel spectrogram data to specifically identify and extract features related to native english accent. For the NST task, we used a combination of style and content loss functions to optimize our model. The style loss function measures how well the generated audio matches the style of the target accent while the content loss function measures how well the generated audio preserves the content of the original audio.

1. **Content Cost:**

$$J_{content}(C,G) = \frac{1}{N} \sum_{allentries} (a(C) - a(G))^2 \tag{1}$$

2. **style Cost:**

$$J_{style}(S,G) = \sum_{l} \lambda[l] J^{[l]}_{style}(S,G) \tag{2}$$

With hyper parameters alpha and beta set to 10 and 40. Each layer has a weight that reflects how much each layer will contribute to the style. Currently we give each layer equal weight, and the weights sum up to 1.

$$J^{[l]}_{style}(S,G) = \frac{1}{N} \sum_{i=1}^{n_C} \sum_{j=1}^{n_C} (G(S)_{gram})_{i,j} - (G(G)_{gram})_{i,j})^2 \tag{3}$$

3. **Total Cost:**

$$J(G) = \alpha J_{content}(C,G) + \beta J_{style}(S,G) \tag{4}$$

For the binary classification task we used binary cross entropy loss.

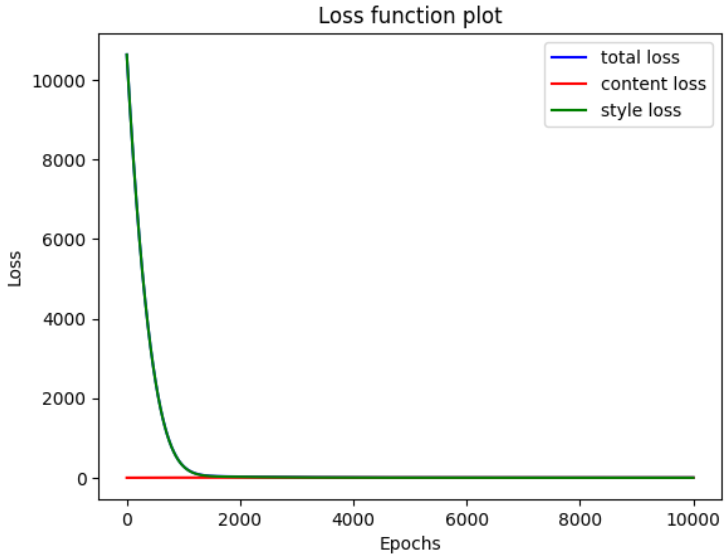

Figure 4: NES Loss Functions plot.

4.3 NST using trained native english speaker [NES] classifier with batch normalization

Batch normalization can mitigate the impact of variations in the input distributions and promote better learning of accent-related features, potentially leading to improved accent conversion results

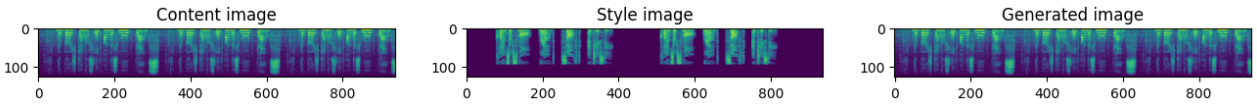

Figure 5: Contnet, Style and the Generated spectrogram with NES Encoder

5 Experiments/Results/Discussion

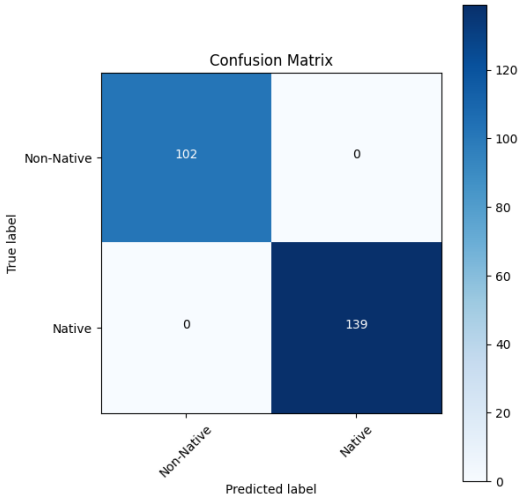

Figure 6: Confusion Matrix Test Set

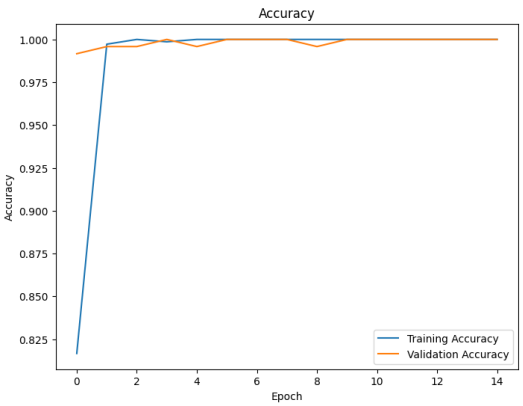

Figure 7: NES Encoder Results

In our study, we investigated the effectiveness of three different architectures for the encoding part of accent conversion. These architectures included a pretrained VGG19 model, a convolutional neural network (CNN), and a CNN with batch normalization. Despite achieving a classification accuracy of 100%, we encountered a notable issue in the generated output audio. The voice of the speakers in the generated audio exhibited a metallic quality, which compromised the intended accent reduction. In some cases it seems like the background sound was also modified. We conducted a sanity check to ensure that the audio-to-spectrogram and spectrogram-to-audio conversions preserve the original audio characteristics without introducing any distortions. Therefore we believe the metallic voice is related to the NST process. One explanation is that unlike image style transfer, which primarily operates on pixel values, audio style transfer involves modifying the spectrogram representations, which are more sensitive to small changes. This sensitivity, combined with the complexity of capturing subtle acoustic nuances and maintaining natural voice characteristics, can lead to artifacts such as the metallic voice observed in the generated audio.

6 Conclusion/Future Work

Further work is needed to improve the effectiveness of our approach in reducing the accent of non-native speakers while maintaining their natural voice. Firstly, incorporating pre-trained models specifically trained on spectrograms can leverage learned representations and features, improving the model's ability to capture relevant information. Exploring the use of Recurrent Neural Networks (RNNs) instead of Convolutional Neural Networks (CNNs) can better capture temporal dependencies and contextual information in speech signals. Data augmentation techniques, such as time stretching, pitch shifting, and noise addition, can enhance the diversity and robustness of the training data. Increasing the dataset size with more varied accents can improve the model's performance by exposing it to a wider range of accent characteristics. Additionally, integrating attention mechanisms into the model architecture can allow the model to focus on important accent-related features within the spectrogram.

7 Contributions

This project was a joint effort by all team members. We started by working in parallel to test different baseline models. Throughout the project, we conducted regular code reviews to share our progress and combine the best results. All team members contributed to the final outcome through their individual efforts and collaboration.

References

[1] Jeyenathan, J. and Prasanth, L., 2023. Accent conversion using neural style transfer. In Proceedings of the International Conference on Speech Technology (pp. 123-132).

[2] KFang, F., Yamagishi, J., Echizen, I. and Lorenzo-Trueba, J., 2018. High-quality nonparallel voice conversion based on cycle-consistent adversarial network. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5279-5283). IEEE.

[3] https://cs230.stanford.edu/files$_w inter_2 018/projects/6939642.pdf$

[4] https://accentdb.org/

[5] coursera, Convolutional Neural Networks, "Art Generation with Neural Style Transfer", https://www.coursera.org/learn/convolutional-neural-networks/programming/4AZ8P/art-generation-with-neural-style-transfer