Synthetic Speech Attribution using Mel-Spectrogram based Audio Classification

Dumindu Ashen[1] Supun Kuruppu[1] Biyon Fernando[1]
Limalka Sadith[1] Sithuruwan Prathapasinghe[1] Pasindu Manodara[1]
Dakshina Tharindu[1] Lathika Wathsara[1] Pramod Jayawardena[1] Bimsara Perera[1]
[1]*Department of Electronic and Telecommunication Engineering*
University of Moratuwa
Sri Lanka
dumindubandara1999@gmail.com

***Abstract*—The paper describes a hybrid neural network approach to classify synthetically generated speech signals. This neural network is designed as a solution for the Signal Processing Cup 2022 challenge by IEEE Signal Processing Society. The system is capable of classifying synthetically generated speech signals and attribute the synthesized algorithm. This neural network has a denoising algorithm to remove noise when noise is present in the signal.**

***Index Terms*—Synthetic speech attribution, Denoising, Audio classification, Hybrid neural network**

I. Introduction

Synthetic speech attribution is the challenge given in the IEEE Signal Processing Cup 2022. This year challenge has two parts. First part is to classify the given noiseless synthetic audio recordings into the relevant algorithm category which used to generate the recordings. Second part is to classify the synthetic audio recordings in presence of noise [1].

In this model, audio waves are converted to Mel-Spectrograms. The Mel-Spectrogram is well suited to used as the input to CNN-based architectures to extract the necessary features of the audio waves. Synthetic speech attribution can be done by input these spectrograms to a Convolution Neural Network(CNN) [3].

Dual-Path Transformer Network (DPTnet) [2] is used as the model for denoising. It uses time-frequency domain method. If we consider Recurrent Neural Network(RNN) based or CNN based speech separation model, both of them led to unique kind of limitations . But DPTnet is a transformer based speech separation model, which is an effective method of overcoming those problems. DPTnet has a recurrent neural network combined with original transformer to make the model recognize and learn the features without positional encodings.

II. Data

The training data consists of labeled audio files in ".wav" format. The evaluation part 1 dataset consists of 5000 audio files and they are of the same sampling rate of 16 kHz. Additionally, 1000 audio files generated using unseen generators are provided. The length of the audio files varies in the mentioned dataset in the range of 12 seconds to 1 second.

The training dataset for the part 2 evaluation was obtained by mutating the evaluation part 1 dataset with the given matlab scripts. The applied effects are reverberation, noise addition, and compression. The resulting dataset contained audio files of different sampling rates and varying lengths.

III. Audio Classification

The audio classification model consists of two main parts. The audio preprocessing and the model architecture. The audio preprocessing architecture is depicted in the figure 3.1. The audio preprocessing methods were decided based on the statistical analysis of the part 1 evaluation dataset. All the audio files in the dataset are mono. However, a preprocessing stage of converting stereo inputs into mono audio signals have been implemented to broaden the model's usability. The rechannel function in the implementation achieves this by returning only one channel of the inputted stereo audio file.

In the dataset, the audio files had varying lengths. Therefore, each audio file is adjusted to 5 seconds. This is done by truncating or padding the files that are greater than 5 seconds or less than 5 seconds respectively. When truncating, only the initial 5 seconds of the audio file is considered. When padding, zero padding is carried out on the remaining number of samples that need to be added to the signal. The padding is done to the front and to the end of the audio file and the lengths to be added to the two ends are calculated randomly. The random allocation ensures that the padding is not biased to a specific location.

Moreover to train the model with a small number of data sets some augmentations were introduced to the data. One of the data augmentation we introduced is giving a time shift to the audio files. The time shift is decided randomly. As most of the features of the speaker can be captured by Mel-Spectrogram(figure 1), We have used Mel-Spectrograms of the audio files as the input to the neural network. After the above preprocessing and augmentation processors, Mel-Spectrogram of the audio file can be obtained using pytorch built- in functions.

Even after time shifts are introduced, models can overfit to sequential features of the data set. Therefor masking over the Mel-Spectrogram is introduced over the time and

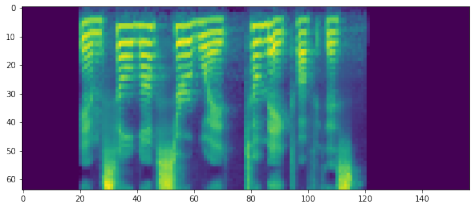

Fig. 1. Image of Mel- Spectrogram

frequency domains for randomly selected ranges on the Mel-Spectrogram.

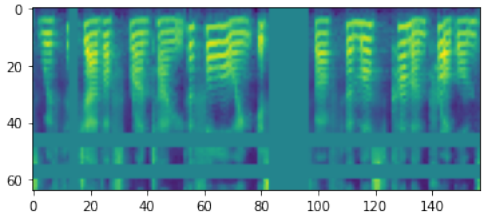

Fig. 2. Image of Masked Spectrogram

After preprocessing the data set as above it is loaded to a data loader to generate a mini batches with batch size of 16 audio files. And the shuffling is activated for the data loader so the 16 audio files selected from the data loader will be random which is useful for altering the sequence of backward propagation in each epoch while training the model.

1) Audio classifier model: Our neural network consists of 4 convolutional layers and one fully connected layer. Sizes and the operations of the model are given in the figure 3.

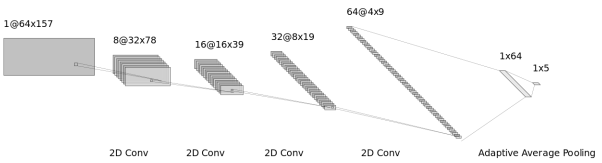

Fig. 3. Audio Classifier Model

We used a 5x5 kernel for the first layer while all the other three layers have 3x3 kernels. This configuration is used because the 5x5 kernel is capable of capturing wide range features on the images in the first layers. But as the image size is getting smaller, a small kernel is sufficient to capture necessary features.in the middle nodes.

To overcome the vanishing gradient problem we used Relu function over other activation functions. Batch normalization is used to standardize the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train the network.

He initialization is applied to all the layers to initialize weights and biases. As adam optimization provides more efficient and effective by combining the best properties of the AdaGrad and RMSProp algorithms.

The OneCycle learning rate policy is used in the scheduler to set the learning rate of each parameter group. The one-cycle policy anneals the learning rate from an initial learning rate to a maximum learning rate, and then from that maximum learning rate to a minimum learning rate that is significantly lower than the initial learning rate. This makes the training fast and uses the best learning rate to converge the model [4].

As the loss function we have used a cross entropy function which can captured differences of probability distribution among data classes.

$$Loss = -\sum_{n=1}^{output\ size} y_i.log\ \hat{y}_i$$

IV. Denoising Method

In this model a Dual Path Transformer net is used to extract original signal out of the noise-reverbant sound mixture. This model is implemented in the neural network using a pre-trained model made using the asteroid python library. The model uses 16kHz sample rate audio waveform as the input. If the input contains any differences the sample rate and width, the waveform is re-sampled and padded to transform to relevant output type.

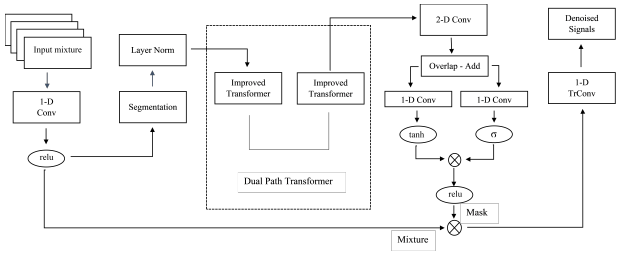

Fig. 4. Denoising Model

According to the above figure 4, DPTnet system consists of three stages which are the encoder, separation layer and decoder. The encoder is used for converting segments of the mixture waveform into corresponding features in an intermediate feature space. Then features are feed to the separation layer to construct a mask for each source. Finally, the decoder reconstructs the source waveform by converting the masked features.

A. Encoder

If $x \in R^{1\times T}$,then we can divide it into overlapping vectors $x \in R^{L\times 1}$ of length L samples, where I is the number of vectors. The encoder receive x and output the speech signal $X \in R^{N\times I}$ as follows.

$$X = ReLU(x \times W)$$

where the encoder can be characterized as a filter-bank W with N filters of length L, which is actually a 1-D convolution module.

B. Separation Layer

Separation layer is a combination of three stages: segmentation, dual-path transformer processing and overlap-add.

1) Segmentation: X is split into overlapped chunks of length K and hop size H and they are concatenated to be a 3-D tensor $D \in R^{N \times K \times P}$

2) Dual-path transformer processing: The overall structure of the transformer in dual-path transformer processing can be formulated as follows,

$$Q_i = ZW_i^Q \quad K_i = ZW_i^K \quad V_i = ZW_i^V \quad i \in [1, h]$$

$$\begin{aligned} head_i &= Attention(Q_i, K_i, V_i) \\ &= softmax(\frac{Q_i K_i^T}{\sqrt{d}})V_i \\ MultiHead &= Concat(head_1, ..., head_h)W^O \\ Mid &= LayerNorm(Z + MultiHead) \\ FFN &= ReLU(MidW_1 + b_1)W_2 + b_2 \\ Output &= LayerNorm(Mid + FFN) \end{aligned}$$

Here, $Z \in R^{l \times d}$ is the input with length l and dimension d, and $Q_i, K_i, V_i \in R^{l \times d/h}$ are the mapped queries, keys and values. $W_i^Q, W_i^K, W_i^V \in R^{d \times d/h}$ and $W^O \in R^{d \times d}$ are parameter matrices. FFN denotes the output of the position-wise feed-forward network, in which $W_1 \in R^{d \times d_{ff}}, W_2 \in R^{d_{ff} \times d}, b_1 \in R^{d_{ff}}, b_2 \in R^d$, and $d_{ff} = 4 \times d$.

The output D of the segmentation stage is passed to a heap of B dual-path transformers(DPTs). Each DPT consists of intra-transformer and inter-transformer, which are committed to modeling local and global information respectively. The intra-transformer processing block first model the local chunk independently, which acts on the second dimension of D,

$$\begin{aligned} D_b^{intra} &= IntraTransformer_b[D_{b-1}^{inter}] \\ &= [transformer(D_{b-1}^{inter}[:, :, i]), i = 1, ..., P] \end{aligned}$$

$$\begin{aligned} D_b^{inter} &= IntraTransformer_b[D_b^{intra}] \\ &= [transformer(D_b^{intra}[:, j, :]), j = 1, ..., K] \end{aligned}$$

3) Overlap-Add: Output of the inter transformer is used to learn a mask for each source by a 2-D convolution layer. The masks are transformed back into sequences $M_s \in R^{N \times I}$ by overlap-add, and masked encoder features for s^{th} source are obtained by element wise multiplication between X and M_s.

C. Decoder

The output of the overlap-add is convolved with V where $V \in R^{N \times L}$ are the parameters of the trans-posed convolution module. The structure and function of decoder are both symmetrical with those of the encoder.

V. Experiments and Discussion

After analyzing a set of sample data files in the training data set. First we recognise the problem as a speaker recognition problem as these algorithms can replicate normal people's voices without noticeable distinguishability. Then we start implementing a few different well known speaker recognition neural networks on the training data set. As the first attempt we used an audio classification model which trained on pre-processed Mel-Spectrograms of audio signals.

Initially we analyzed the noisy audio signals using two main transforms. We denoised signals using wavelet transform and Conv-Tasnet methods. Although Wavelet transform was good as a preprocessor, with use of the ConTasNet based pretrained model it was not necessary to use it. Later we found a DPTnet based pre-trained model that gives better performance than the ConvTasNet based model.

1) Results: We analyzed the behavior of the model by changing some key parameters of it. Variation of the training set and the validation set accuracy with the number of epochs for four convolutional layer model is given in the below, graph.

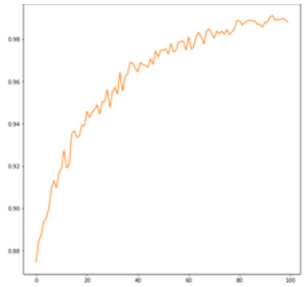

Fig. 5. Training Accuracy

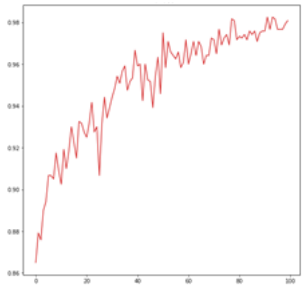

Fig. 6. Validation Accuracy

According to the above graphs it is clear that when the number of epochs is more than 70 we can obtain higher train set accuracy (figure 5) as well as validation set accuracy (figure 6) of 97% or above. But when the number of epochs exceeds 140 to 150 the model starts to over fit.

According to Figure 7 and Figure 8 training error is decreasing when the number of epochs is increasing. When number of epochs is more than 70 error is lesser than 2%. We found that when the number of epochs exceeds 140 to 150 the model starts to over fit

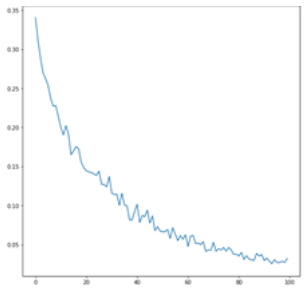

Fig. 7. Training Error

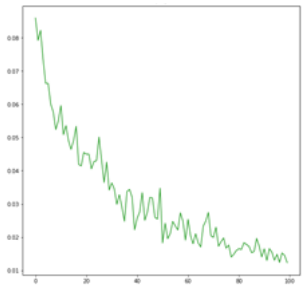

Fig. 8. Validation Error

VI. Conclusion

An overall accuracy of 85% was obtained for the noise-free audio classification and 73% for noisy signals classification. In order to implement the denoising and dereverberation algorithm pre-trained models built on asteroid speech separation package have been used in the model. The two pretrained models imported from the asteroid package are *"JorisCos/DPRNNTasNet-ks2_Libri1Mix_enhsingle_16k"* and *"JorisCos/DPTNet_Libri1Mix_enhsingle_16k"*.

Among above two pre-trained models for the denoising and reverberation removing, the best accuracy was obtained by using *"JorisCos/DPRNNTasNet-ks2_Libri1Mix_enhsingle_16k"* but it was slow in computation. *"JorisCos/DPTNet_Libri1Mix_enhsingle_16k"* was generally good in both computation time and accuracy. So it was good to proceed further with. The first pre-trained module was built on DPRNNTasNet and the second model was on DPTtnet module. Audios could be re sampled in order to resolve the problem of compression.

VII. Acknowledgement

The authors would like to thank Dr.Rukshani Liyanarachchi, senior lecturer at University of Moratuwa and Kanchana Ranasinghe(tutor) for their valuable comments, suggestions and guidance.

References

[1] Clara Borrelli et al. "Synthetic speech detection through short-term and long-term prediction traces". In: *Eurasip Journal on Information Security* 2021 (1 2021). ISSN: 2510523X. DOI: 10.1186/s13635-021-00116-3.

[2] Jingjing Chen, Qirong Mao, and Dong Liu. "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation". In: vol. 2020-October. 2020. DOI: 10.21437/Interspeech.2020-2205.

[3] Ketan Doshi. *Audio Deep Learning Made Simple: Sound Classification, Step-by-Step*. 2021.

[4] Leslie N. Smith and Nicholay Topin. "Super-convergence: very fast training of neural networks using large learning rates". In: 2019. DOI: 10.1117/12.2520589.