VoxCeleb Dataset Exploration and Audio Analysis

Chaojie Zhang
cz2064@nyu.edu

December 16, 2020

Abstract

VoxCeleb (Nagrani, Chung, Xie, and Zisserman (2020)) is a dataset consisting of short clips of human speech, extracted from interview videos uploaded to YouTube. Our project implements speaker verification and audio style transfer using VoxCeleb Dataset. We apply 1D-CNN, RNN, CNN-LSTM models in the speaker verification task, and apply neural style transfer (Gatys, Ecker, and Bethge (2015)) on audio signals. All codes and related material are available in https://github.com/ChaojieZhang-cz/TS-Project-VoxCeleb.

Keywords: Speaker Verification; Deep Neural Networks; Audio Style Transfer

1 Introduction

Deep neural networks have been widely used in classification and speech recognition.But there is not much research applying deep neural networks in speaker verification using raw audio signals. In our project, we build different deep neural network models to achieve speaker verification from raw audio data.

Style transfer as a computer vision technique, it allows us to recompose the content of an image in the style of another (Gatys et al. (2015)). It is widely used in image style transfer. In our project, we applied the principle of style transfer, to achieve audio style transfer.

2 Related Work

ResNet (He, Zhang, Ren, and Sun (2016)) and VGG (Simonyan and Zisserman (2014)) are powerful applications in classification tasks, the models can capture important features from images or signals. Kernel-based binary classifiersr (Lee, Tso, Chang, Wang, and Jeng (2014)) can achieve speaker identification using feature vectors from two audio files.

Neural style transfer (Gatys et al. (2015)) can recompose the content of an image in the style of another.

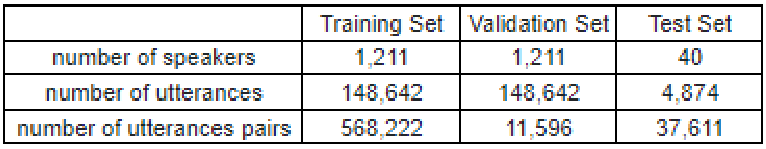
3 Methodology

3.1 Task

We have two tasks in this project. Task 1 is speaker verification, to determine whether two samples of speech are from the same person. Task 2 is audio style transfer, to transfer the style of a reference utterance to a target utterance content.

3.2 Data

VoxCeleb (Nagrani et al. (2020)) is a dataset consisting of short clips of human speech, extracted from intervew ivideos uploaded to YouTube. We use the audio data in VoxCeleb1 (Nagrani, Chung, and Zisserman (2017)) database. All data is splitted into a training set, validation set and test set. The speakers in the test set are not overlapping with any speakers in the training set nor validation set. We then generate random utterance pairs from utterances, the number of matching pairs (from the same speaker) is approximately equal to the number of unmatched pairs (from different speakers). Table 1 shows the number of speakers, utterances and utterance pairs in training, validation and test set.



	Training Set	Validation Set	Test Set
number of speakers	1,211	1,211	40
number of utterances	148,642	148,642	4,874
number of utterances pairs	568,222	11,596	37,611

Table 1: Train, validation, test split

3.3 Preprocess

We conduct two approaches in data preprocess. The first approach for 1D-CNN models is to convert audio files to 1D tensors, normalize the tensors with mean=2e-5, std= 0.05, crop the tensors to (1x65536) tensors. The second approach for the CNN-LSTM model is to convert the audio files to spectrogram, and apply log transform to the spectrogram. In the log transform, we add a small value to the spectrogram to avoid negative infinite output. We then analyze the audio data as spectrogram images.

3.4 Speaker Verification Models

The main structure of the models is shown in Figure 1. We apply 1D-CNN and CNN-LSTM models to generate two feature vectors from the utterance pair, then generate the b-vector (Lee et al. (2014)) using element-wise summation, subtraction, and multiplication, from two feature vectors, and connect with fully connected layers.

We use 1D-CNN and CNN-LSTM models to generate feature vectors from audio files. We use VGG (Simonyan and Zisserman (2014)) and ResNet (He et al. (2016)) as reference to design our 1D-CNN model. We change all 2D convolution and 2D max pooling to 1D convolution and 1D max pooling, and use similar structure in our models. The 1D-CNN model structures are shown in Table 2.

For the CNN-LSTM model (Figure 2), we use the ResNet18 (He et al. (2016)) to generate intermediate feature vectors from the spectrogram. Since the utterances from the dataset have different lengths, the converted spectrograms will have different widths. We remove the fully connected layer and replace the global average pooling layer with a 7x7 average pooling layer. We will generate a feature map from the ResNet model. The size of the feature map will be different according to the different length of the original audio file. In the generated feature map, each column is regarded as a feature vector of a certain part from the original audio file. We then use 2 layers LSTM to generate the final feature vector from all intermediate feature vectors.

In the training, we use the Adam optimization (Kingma and Ba (2014)) and set the learning rate to 1e-4, we use binary cross entropy loss and train all models for 100 epochs. In each epoch, we randomly select 50,000 utterance pairs from the training set. Because of the large numbers of utterance pairs in the training set, the accuracy of all models are still increasing in the final several epochs.

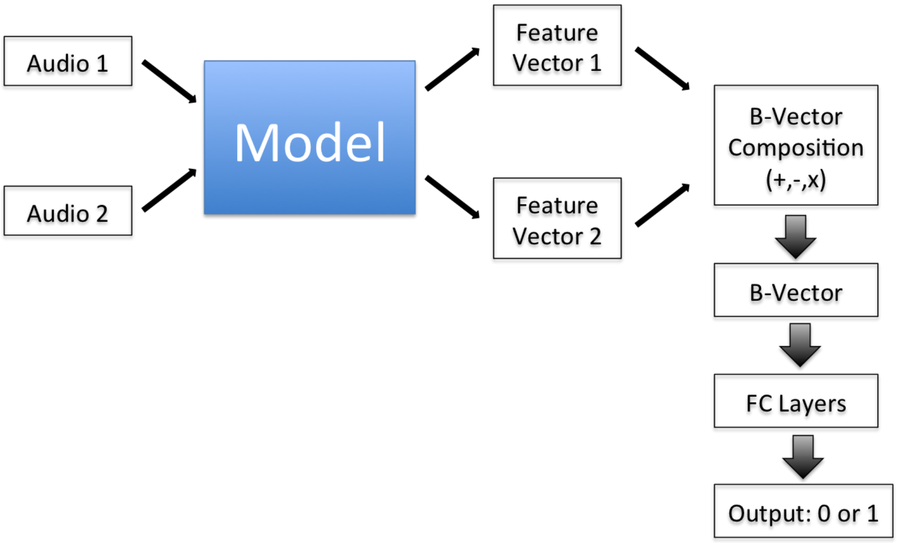


Figure 1: Model structure

3.5 Audio Style Transfer

In audio style transfer, we use the pre-trained VGG16 from the speaker verification task. Since the model is trained for speaker verification, it can capture important features of speech style from different speakers. In the audio style transfer, we apply the neural style transfer method from the work of Gatys et al (Gatys et al. (2015)), to combine the content loss and style loss and generate a new audio file. The new audio file will contain the content from the content audio, and also have the speech style from the style audio. We use the feature maps from the first layer for content loss, and the feature maps from the first five layers for style loss. We set content weight = 1, style weight = 1e10, and optimize for 10,000 steps.

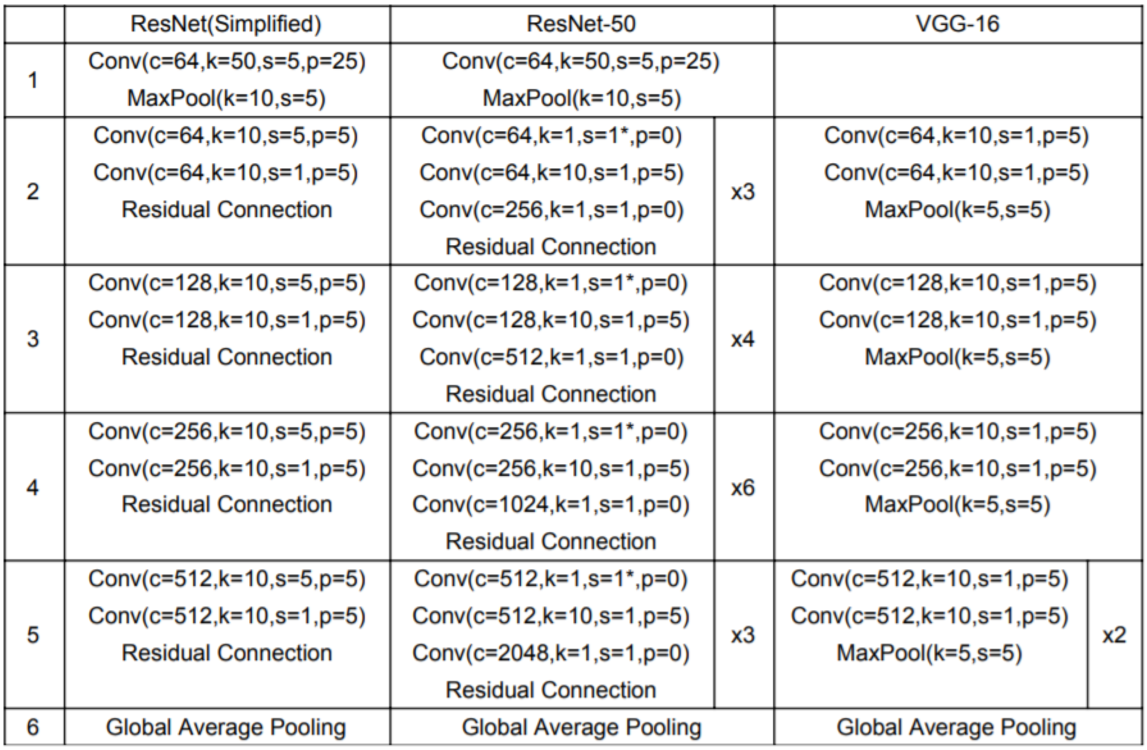
4 Evaluation

We use accuracy, area under the receiver operating characteristic curve (AUC) and Equal Error Rate (EER) to evaluate our model performance. EER is a rate used to determine the threshold value for a system when its false acceptance rate (FAR) and false rejection rate (FRR) are equal (Chung et al. (2019)). The evaluation result is shown in Table 3.

We didn't evaluate our audio style transfer result, some transfer examples are available on the shared slides in github.

5 Failed Attempts in Speaker Verification

We have implemented LSTM models independently to achieve speaker verification, but the accuracy didn't improve after 20 epochs training. We tried to convert the audio file to a 1xN tensor and a 64xN tensor, and apply two layers LSTM model to generate feature vectors from the audio (Figure 3). We also tried to convert the audio file to a spectrogram,



	ResNet(Simplified)	ResNet-50		VGG-16	

1	Conv(c=64,k=50,s=5,p=25)
	MaxPool(k=10,s=5)
		Conv(c=64,k=50,s=5,p=25)
		MaxPool(k=10,s=5)			

2	Conv(c=64,k=10,s=5,p=5)
	Conv(c=64,k=10,s=1,p=5)
	Residual Connection
		Conv(c=64,k=1,s=1*,p=0)
		Conv(c=64,k=10,s=1,p=5)
		Conv(c=256,k=1,s=1,p=0)
		Residual Connection
			x3	Conv(c=64,k=10,s=1,p=5)
				Conv(c=64,k=10,s=1,p=5)
				MaxPool(k=5,s=5)	

3	Conv(c=128,k=10,s=5,p=5)
	Conv(c=128,k=10,s=1,p=5)
	Residual Connection
		Conv(c=128,k=1,s=1*,p=0)
		Conv(c=128,k=10,s=1,p=5)
		Conv(c=512,k=1,s=1,p=0)
		Residual Connection
			x4	Conv(c=128,k=10,s=1,p=5)
				Conv(c=128,k=10,s=1,p=5)
				MaxPool(k=5,s=5)	

4	Conv(c=256,k=10,s=5,p=5)
	Conv(c=256,k=10,s=1,p=5)
	Residual Connection
		Conv(c=256,k=1,s=1*,p=0)
		Conv(c=256,k=10,s=1,p=5)
		Conv(c=1024,k=1,s=1,p=0)
		Residual Connection
			x6	Conv(c=256,k=10,s=1,p=5)
				Conv(c=256,k=10,s=1,p=5)
				MaxPool(k=5,s=5)	

5	Conv(c=512,k=10,s=5,p=5)
	Conv(c=512,k=10,s=1,p=5)
	Residual Connection
		Conv(c=512,k=1,s=1*,p=0)
		Conv(c=512,k=10,s=1,p=5)
		Conv(c=2048,k=1,s=1,p=0)
		Residual Connection
			x3	Conv(c=512,k=10,s=1,p=5)
				Conv(c=512,k=10,s=1,p=5)
				MaxPool(k=5,s=5)
					x2

6	Global Average Pooling	Global Average Pooling		Global Average Pooling	

Table 2: 1D CNN models (c:channels, k:kernel size, s:stride, p:padding, s=1* represents stride=5 in the first iteration and stride=1 in the rest iterations.)

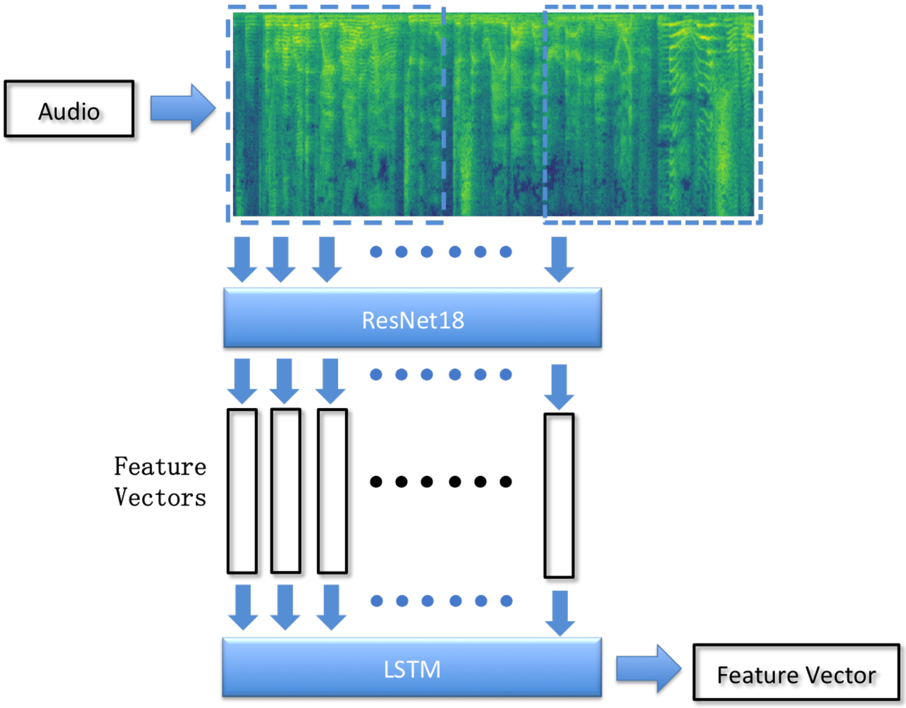


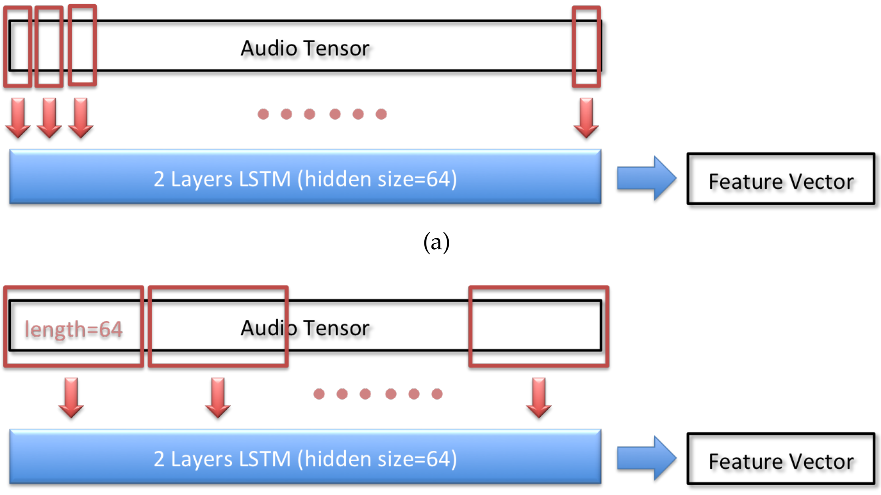
Figure 2: CNN-LSTM

	ResNet-10	ResNet-50	VGG16	CNN-LSTM
Accuracy	89.07%	87.44%	85.52%	91.68%
AUC	0.96	0.94	0.93	0.97
EER (Equal Error Rate)	0.11	0.13	0.15	0.09

Table 3: Speaker verification results

then take each column of the spectrogram as an intermediate feature vector, and apply two layers LSTM to generate feature vectors from all intermediate feature vectors (Figure 4).

All LSTM models failed to generate important feature vectors for speaker verification, there may be some reasons. Because speaker identification is not related to the order of the sentence, but related to voice characteristics. The CNN models can capture features of voice characteristics and give good predictions, while the LSTM models work more efficiently in capturing lag events. Also, the two layers LSTM model contains much fewer parameters than our simplest 1D-CNN model. There are only 199,168 parameters in the two layers LSTM model, but there are 5,431,040 our simplest 1D-CNN model. The 2-layers LSTM model may not be complex enough to capture advanced features from audio files.



(a)

(b)

Figure 3: LSTM models for 1D tensors

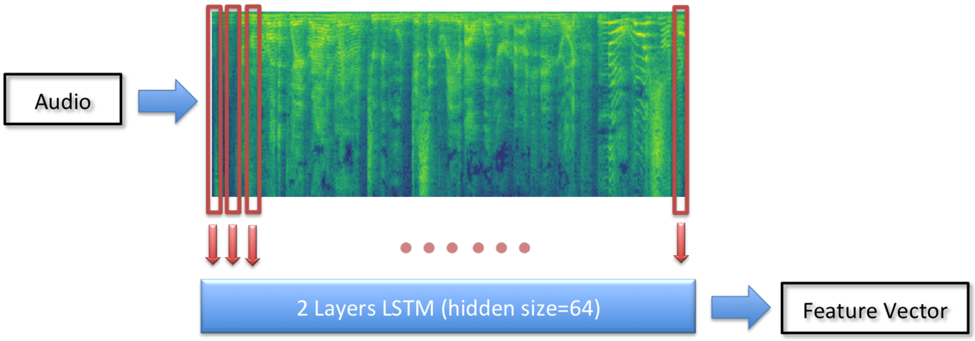


Figure 4: LSTM model for spectrogram

6 Discussion and Conclusions

In the speaker verification task, our 1D-CNN and CNN-LSTM models give good performance. The LSTM models failed to make right predictions, maybe because of wrong choice of RNN model and insufficient complexity. For all successful models, because of the large numbers of utterance pairs in our training set, the accuracy of all models are still increasing in the final several epochs. The models will give better performance with more training. Also, our training dataset only contains utterances from 1,211 speakers, it is not enough to generate important features for speaker identification. Training the models in larger datasets will give better results.

In the audio style transfer, the model can achieve flexible audio style transfer using only content utterance and style utterance. There is not much irrelevant noise in the generated audio, it implies that the model has captured the important features of speech style. Since our model was trained in a small dataset (1,211 speakers), training the model for more epochs and in larger datasets will improve the model performance.

So far, our audio style transfer result is not perfect. The generated audio did contain a new speech style from another person, but many of the audio outputs contain irrelevant noise, and some doesn't contain all contents from the original audio. A perfect transfer is difficult in this task, many of the utterances contain background noise or short utterances from other persons. The number of speakers in the training set are too small for feature extraction, the model may need more samples to extract important features of speech style. Also, evaluating the style transfer result is difficult. We can use our CNN-LSTM model from the speaker verification task to evaluate the transfer result. We also need to implement speech recognition models to identify the content loss in the generated audio.

References

Chung, J. S., Nagrani, A., Coto, E., Xie, W., McLaren, M., Reynolds, D. A., & Zisserman, A. (2019). Voxsrc 2019: The first voxceleb speaker recognition challenge. *arXiv preprint arXiv:1912.02522*.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

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

Lee, H.-S., Tso, Y., Chang, Y.-F., Wang, H.-M., & Jeng, S.-K. (2014). Speaker verification using kernel-based binary classifiers with binary operation derived features. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 1660–1664).

Nagrani, A., Chung, J. S., Xie, W., & Zisserman, A. (2020). Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, *60*, 101027.

Nagrani, A., Chung, J. S., & Zisserman, A. (2017). Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*.

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