

VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking

Aditya Sharma - 170050043

Suraj - 170050044

Rohan Abhishek - 170050078

Introduction

- Problem: Separating the voice of a target speaker from multi-speaker signals.
- There are 2 major methods:
 - First extract all audio signals(for each speaker) and then decide which output corresponds to the target speaker.
 - Directly use the d-vector of the speaker as input along with the Noisy Audio spectrogram to generate a mask. This mask is then applied to the Noisy Audio spectrogram to get the target audio -- Approach followed by us.
- d-vector: Speaker embedding vector, obtained using a reference audio of the target speaker.



Input-Output Behaviour

The input to our system will be 2 audio files: **reference_audio** and **noisy_audio**.

- **reference_audio** is used to determine a *d-vector* for the reference/target speaker whose audio is to be extracted from the **noisy_audio**.

The output will be the audio of the target speaker extracted from the *noisy_audio* after removing background noise and audio from other speakers, if any.

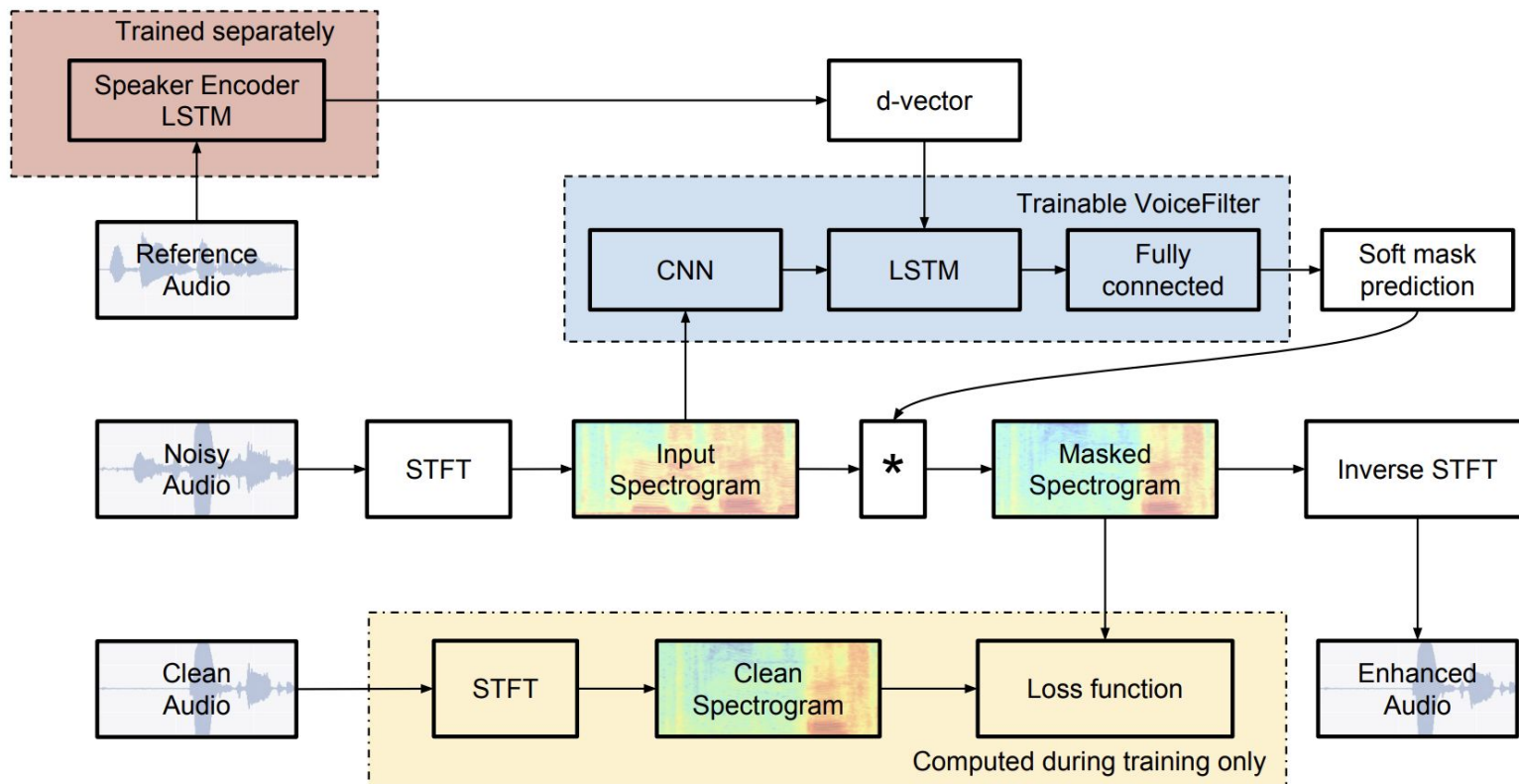
The main neural network is given 2 inputs:

1. a **d-vector** of the target speaker
2. **magnitude spectrogram** computed from the noisy audio

The neural network returns a **mask** to be applied to the noisy audio to obtain the enhanced audio.



System Architecture



Reference: <https://arxiv.org/pdf/1810.04826.pdf>

Methodology of VoiceFilter System

- A magnitude spectrogram is computed by applying STFT on the Noisy Audio.
- The VoiceFilter system neural network takes the D-vector of target speaker and spectrogram computed of the Noisy Audio. This network returns a mask to filter the audio.
- The mask is multiplied element-wise to the input spectrogram to obtain the masked spectrogram.
- The masked spectrogram undergoes Inverse STFT to produce the enhanced audio version of the target speaker.



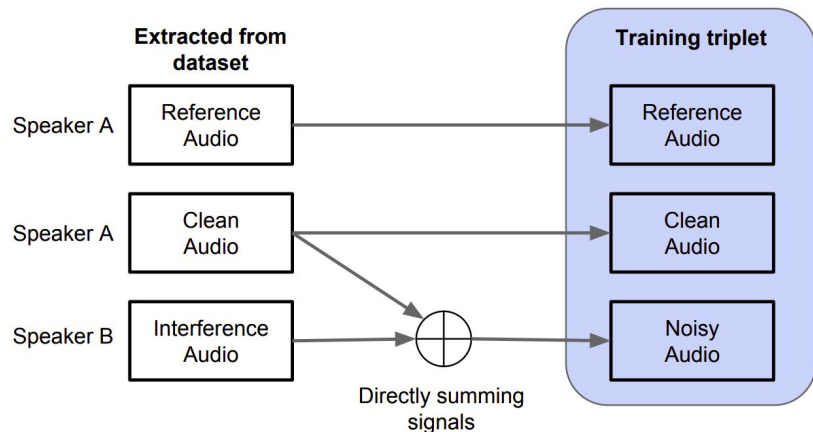
Evaluation Metric

Mainly, there will be 2 metrics to evaluate our output:

- **WER (Word Error Rate)**
 - Calculate WER for noisy_audio and output_audio wrt the test of the target speaker.
 - WER should be reduced in multi-signal scenarios, which means the output audio is good.
 - WER should remain same for single speaker scenarios, which means that the system is not degrading the performance and recognizing the speaker correctly.
- **SDR (Source to Distortion Ratio)**
 - SDR is a very common metric to evaluate source separation systems. It is an energy ratio between the energy of the target signal contained in the enhanced signal and the energy of the errors (coming from the interfering speakers and artifacts). **Thus, the higher it is, the better.**



Input Data Processing Workflow



This diagram represents the preprocessing done to obtain the Noisy Audio, Reference Audio and Clean Audio for the train, validation and test datasets.

We used random reference and clean audio from a speaker and an interference audio from a different speaker and generated 3 audios as shown in the diagram.

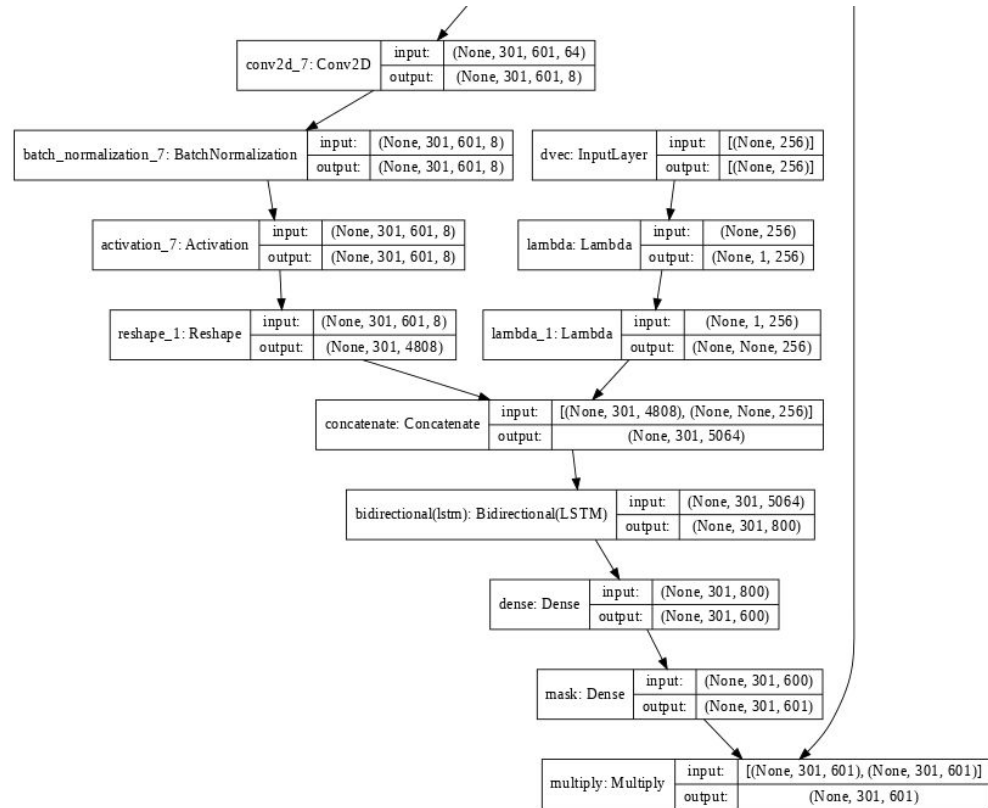
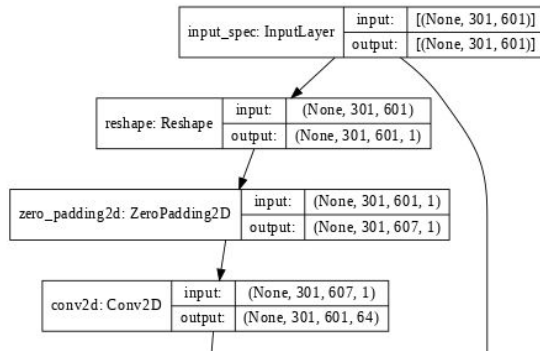
More statistics on train, test and validation datasets in upcoming slides.

Speaker Encoder Model

- We used a pre-trained model, which consists of the following details.
- The speaker encoder is a **3-layer LSTM** network trained with the **generalized end-to-end loss**.
- It takes as inputs log-mel filterbank energies extracted from windows of 1600 ms, and outputs speaker embeddings, called **d-vectors**, which have a fixed dimension of 256.



VoiceFilter Model



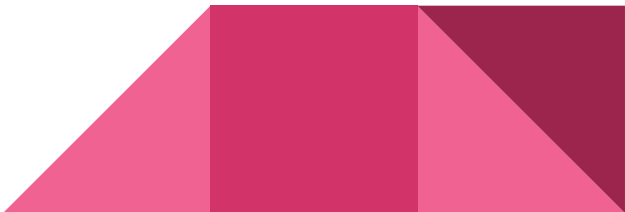
https://drive.google.com/file/d/1fcZiaAtbwbbavxX3YBfCG17Gy_lulyJrN/view?usp=sharing

Dataset Statistics

LibriSpeech Dataset

1. Train (6.3GB): 251 speakers
 - a. 16k + 7.6k samples prepared (45+21GB)
 - b. Took approximately 9 hours on Google Colab to prepare the dataset.
2. Validation(322MB): 40 speakers
 - a. 920 samples prepared
3. Test(331MB): 40 speakers
 - a. 949 samples prepared

Approximately 1.17%, 1.09%, 1.05% samples respectively with same primary and reference speech.



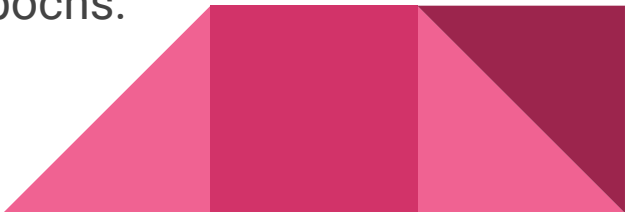
Model Training

Model 1: Obtaining d-vector from reference speech

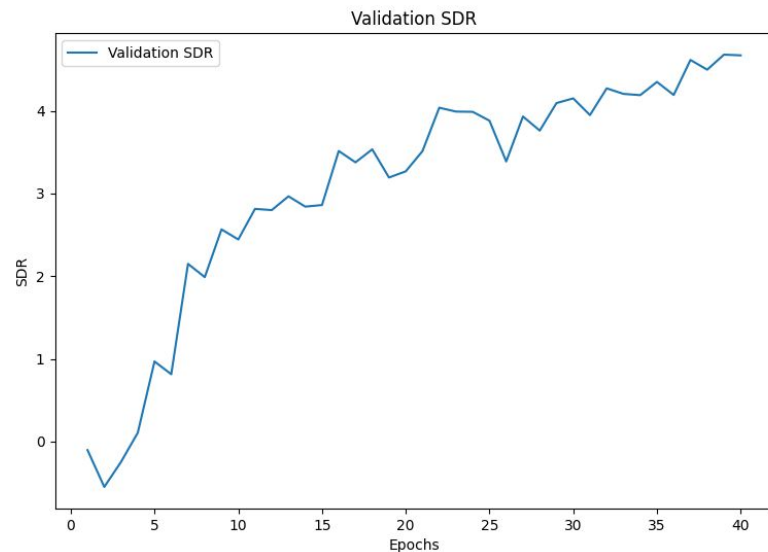
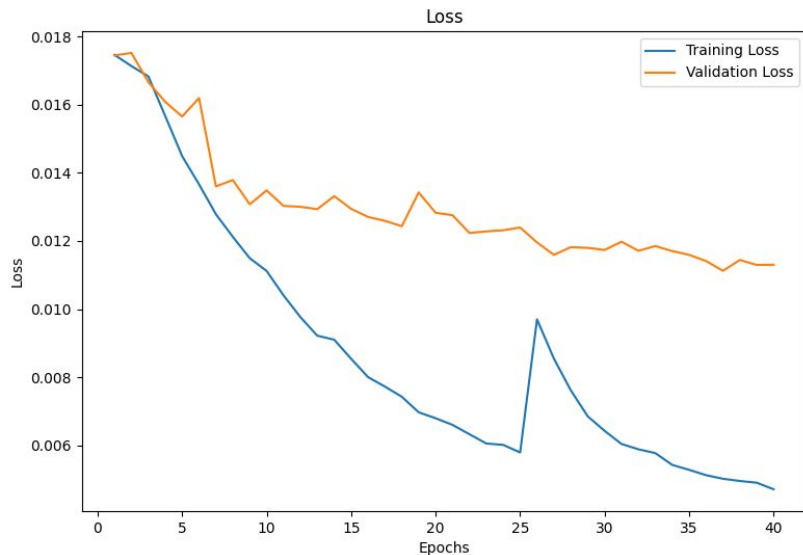
- Used pretrained model provided by <https://github.com/mindslab-ai/voicefilter>

Model 2: VoiceFilter

Trained the model on Google Colab Pro GPUs.

- Training set 1 (16k samples) used to train first 25 epochs.
 - Took approx. 13 hours (30 mins per epoch)
 - Training set 2 (7.6k samples) used to train next 15 epochs.
 - Took approx. 3 hours (12 minutes per epoch)
- 

Training and Validation results



Test results

Number of epochs	Test SDR
10	2.277
20	3.267
30	4.398
39	4.681
40	4.672

The model after 39 epochs performs the best on validation dataset.

Final Test SDR = 4.681







The paper mentions median SDR value of 12.6, but multiple unofficial implementations claim to reach only upto 6.

VoiceSplit: Best SDR 6.17 after 3k hour training.

This is because Google has used a private dataset with audios of over millions of speakers to train the model.



Demo: Case 1 (Mixed audio, Different speakers)

Reference audio	
Primary audio	
Secondary audio	
Mixed audio	
Output (Model 0)	
Output (Model 40)	

VoiceFilter should remove the secondary audio from the input by the mask and the output should be similar to the primary audio.

Demo: Case 2 (Single Audio, Same Reference Speaker)

Reference audio
Primary audio
Mixed audio
Output (Model 0)
Output (Model 40)



Ideally, the output should be same as the input as it consists only the reference speaker's audio.



Demo: Case 3 (Single Audio, Different reference Speaker)

Reference audio
Primary audio
Mixed audio
Output (Model 0)
Output (Model 40)



Ideally, there should be no voice in the output as the input audio does not have reference speaker's audio.



Trials and Tribulations

- Even though the key concept of the project was simple: To train a masking network, we had to go through some articles and other papers to get a more detailed understanding of the problem and the solution.
- Dataset Preparation -- Due to the size of the training dataset(16k files), we were not able to effectively use Google Drive as Colab gives error in case of large no, of small file transfers.
- Training using GPU -- Colab provides only 68 GB disk for GPU runtime(of which 38 GB is already used). So, we were not able to even load the dataset. Same goes for Kaggle and Paperspace. Finally, we checked the specification of Google Colab Pro and it was just sufficient for our needs.



Work Split

1. Aditya Sharma: Data Preparation + Understanding Paper + Report
2. Suraj: Understanding Paper + Implementation + Dataset Preparation + Experiments
3. Rohan Abhishek: Paper Reading + Implementation + Experiments

Report is not yet completed.



References

- [VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking](#)
Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John Hershey, Rif A. Saurous, Ron J. Weiss, Ye Jia, Ignacio Lopez Moreno
- <https://github.com/mindslab-ai/voicefilter>
- <https://github.com/jain-abhinav02/VoiceFilter>
- <https://github.com/Edresson/VoiceSplit>



The background is a solid pink color. In the top right corner, there is a decorative pattern of geometric shapes: a light pink triangle, a dark pink square, and a light pink triangle, all arranged in a way that suggests a larger square or rectangle being divided into smaller sections.

Thank You!