# ASR Project Report

## A General Framework for Self Supervised Sound Localization and Separation

Arjit Jain 170050010
Aditya Vavre 170050089

November 2019

## 1 Introduction

Recently, there has been a lot of amazing work on Self Supervised Learning methods, using a rich collection of unlabelled videos, for Sound Localization and Separation tasks ranging from Audio Video Correspondence to Cocktail Party Problem. Even though these studies have been conducted independently and the tasks are different, the way these networks are constructed and trained is similar. We tried to exploit the underlying structure to come up with a generic network and training algorithm which can then be instantiated to work in different settings.
The motivation for this comes from the Deep Q-Network (DQN). We know that DQN can be used out of the box for a whole range of tasks, i.e. you can train the network on your setting and can get reasonably well results without any hyper-parameter tuning.

## 2 Methodology

The paper "Looking to Listen at the Cocktail Party:A Speaker-Independent Audio-Visual Model for Speech Separation" [1] describes a model which, given the facial features of people in a scene along with the audio stream, can filter out the speech corresponding to each person.
We can generalize the above idea by abstracting facial features to entities, so that this idea can be applied to other settings. For instance, Sound Localization can be modelled by filtering out the sound made by all objects in the scene and outputting the object which has the highest "similarity" with the original audio.

## 3 Prior Work

This work was not directly building upon any prior work. However, there were many papers following similar approach which were release at the same time as this. Here is a list of some of those papers.

- Audio-Visual Scene Analysis with Self-Supervised Multisensory Features
- Learning to Localize Sound Source in Visual Scenes
- Objects that Sound
- The Sound of Pixels

# 4 Network Architecture

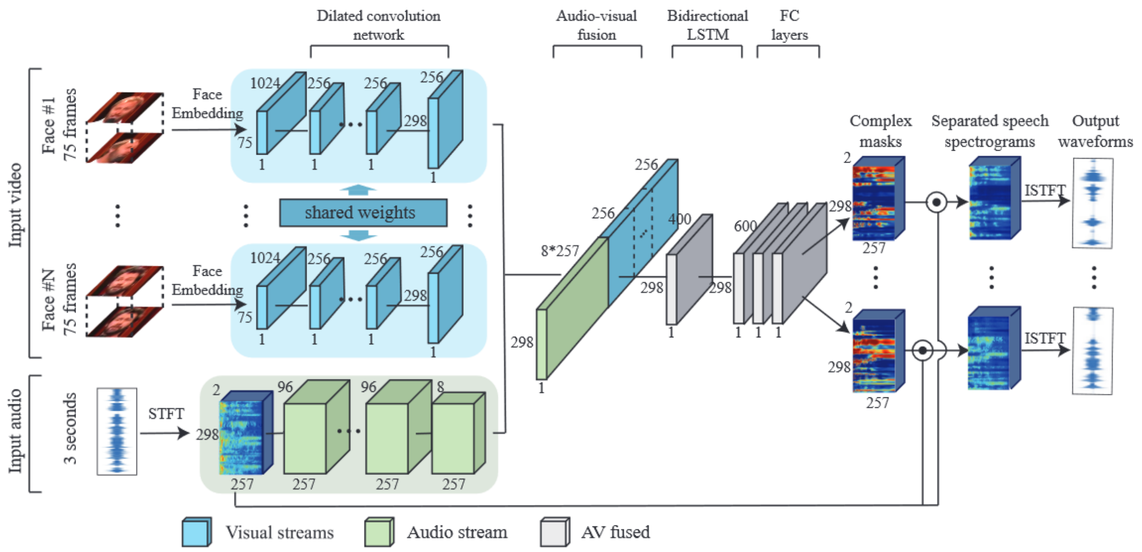


Figure 1: Model's Network

Figure 1 provides a high-level overview of the various modules in our network, which we will now describe in detail.

**Audio and visual streams**. The audio stream part of the model consists of dilated convolutional layers. The visual stream of the model is used to process the input face embeddings, and consists of dilated convolutions. To generalize to entities beyond faces, only the input to the visual stream would change, say features of objects obtained through a pre-trained object detection network.

**AV fusion**. The audio and visual streams are combined by concatenating the feature maps of each stream, which are subsequently fed into a BLSTM followed by three FC layers. The final output consists of a complex mask (two-channels, real and imaginary) for each of the input speakers. The corresponding spectrograms are computed by complex multiplication of the noisy input spectrogram and the output masks. The loss given by

$$L = \frac{1}{2}\sum_{t=1}^{T}(||y_{1_t} - \tilde{y}_{1_t}||^2 + ||y_{2_t} - \tilde{y}_{2_t}||^2 - \gamma||y_{1_t} - \tilde{y}_{2_t}||^2 - \gamma||y_{2_t} - \tilde{y}_{1_t}||^2) \ [2]$$

between the power-law com-pressed clean spectrogram and the enhanced spectrogram is used as a loss function to train the network. The final output wave forms are obtained using ISTFT.

## 4.1 Implementation Details

Our network is implemented in PyTorch, and its included operations are used for performing waveform and STFT transformations. ReLU activations follow all network layers except for last, where a sigmoid is applied. Batch normalization is performed after all convolutional layers. We use a batch size of 1 sample(due to memory constraints) and train with Adam optimizer for 10 epochs with a learning rate of $1 * 10^{-4}$.

## 4.2 Data Pre-Processing

Only 3 seconds of audio from each video is being used. All audio is re-sampled to 16kHz, and stereo audio is converted to mono by taking only the left channel. STFT is computed using a Hann window of length 25ms, hop length of 10ms, and FFT size of 512, resulting in an input audio feature

of 257x298x2 scalars.
We re-sample the face embeddings from all videos to 25 frames-per-second (FPS) before training. This results in an input visual stream of 75 face embeddings. The facial embeddings are obtained from FaceNet [3].

# 5 Experiments and Discussion

## 5.1 Experiment no. 0

The goal of this experiment was to check whether our network really works or not. We approached this by trying to over-fit on a very small training dataset size of 5 samples and then making predictions on the same samples. And indeed the network was able to separate speech.

## 5.2 Experiment no. 1

The goal of this experiment was to check the effect of power-law compression on the network's performance. We perform an ablation analysis for this. We created a training dataset from 100 randomly sampled examples, and trained it for 10 epochs on two networks, which were exactly the same except one used power-law compression and the other did not. Contrary to our intuition, we found that the network which used power-law compression seemed to produce noisy and distorted outputs and the one which did not use it produced more stable outputs.

## 5.3 Experiment no. 2

While listening to some outputs of the model, we found that the model was able to do a much better job of separating speech when the speakers were speaking different languages. The basis of this experiment analyze this behaviour of the model. We created a dataset which contained mixtures of various combinations, like different languages (one being English), different languages (none of them being English), and different accents of English. While we found that different languages does improve separation, in both cases, we did not find any significant improvement in the different accents for same language case. There are two possible explanations for this, either the network is learning some language based representation, or because the data is predominantly in English, the network treats all other languages as some sort of "anomalies".

## 5.4 Experiment no. 3

This was the experiment to perform sound localization on the same network architecture. After getting the data ready, we found out that unlike faces, irrelevant objects in a scene can be much more in number. Hence, if trained on all the objects in a scene, the network would be outputting a complex mask of all zeros for most objects which is not good for learning. To counter, we could either use a dataset which does not have many irrelevant background objects, or get a dataset which only has a specific set of object classes, say only musical instruments. This required us to find a dataset which matches the above criteria, pre-process it and train the network on it. Due to time constraints, we weren't able to achieve this.

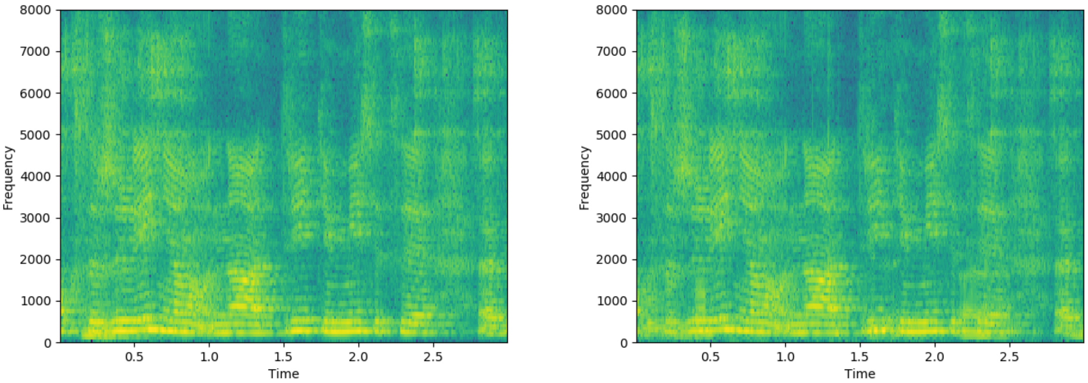
## 5.5 Evaluation

We used the SDR metric(Sound to Distortion Ratio) to evaluate our model. The mean SDR on the validation set(100 examples) was found to be 3.74 after training our model on 500 examples for 10 epochs. The highest SDR was found to be 7.03 and the lowest SDR was 0.50. We manually listened to a few random predictions as well as the predictions corresponding to the highest and lowest SDR values and found that indeed for most examples the model prediction weren't very good. Mostly, either the model would produce very similar predictions for both the speakers, or it would be able to perform some separation but the audio generated would be quite distorted.

To see whether this was an artifact of the particular data sampled from the training data or not, we tried to train the model on different random samples of the data-set for different number of epochs and found that the reason for low mean SDR was the result of less number of training epochs and not an artifact of the particular data-set.

Cherry Picked Result:
The following is the ground truth spectrogram (left), and the spectrogram generated by multiplying the complex ratio mask outputted by our model, , for an audio mixture (right).

g



# References

[1] Ariel Ephrat et al. "Looking to listen at the cocktail party". In: *ACM Transactions on Graphics* 37.4 (July 2018), pp. 1–11. ISSN: 0730-0301. DOI: 10.1145/3197517.3201357. URL: http://dx.doi.org/10.1145/3197517.3201357.

[2] Po-Sen Huang et al. "Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural Source Separation". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.12 (Dec. 2015), pp. 2136–2147. ISSN: 2329-9304. DOI: 10.1109/taslp.2015.2468583. URL: http://dx.doi.org/10.1109/TASLP.2015.2468583.

[3] F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A unified embedding for face recognition and clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.