LOOKING TO LISTEN AT THE COCKTAIL PARTY

Sudhir Kumar Suman
16D070027
Dept. of Electrical Engineering
IIT Bombay

Saarthak Kapse
16D070041
Dept. of Electrical Engineering
IIT Bombay

Abhinav Seth
16D070029
Dept. of Electrical Engineering
IIT Bombay

December 14, 2019

1 Task

The aim of the project is to separate speech signals from mix audio and assign it to corresponding speakers. There has been experiments to predict text using just lips movement of a person. So it is believed that instead of just separating audio signal alone, video input may help to separate audio better as what human beings do. And Audio-Visual Model solves one more underlying task, i.e without video input the task to assign which audio part belongs to which speaker was a really tough task and to do this it is required to store voice characteristics of speakers, and hence it makes the model speaker dependent. Whereas in Audio-Visual model it can assign which audio part belongs to whom with the help of lip movements/video stream features, and hence it is speaker independent model.

2 Prior work

There has been vast literature and experiments in Speech separation task. Our project is inspired by Google Research, Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation paper. Their research is among one of the first approach to make a speaker independent speech separation model. On the same line their is one more paper by university of oxford, The Conversation: Deep Audio-Visual Speech Enhancement. Both addressing the same task but with different approaches.

3 Method

Here we are using the same methodology as in Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation paper. The neural network model is mainly divided into 3 parts.
First is audio stream network whose input is Power Law compressed (To the power 0.3) STFT spectrogram of mixed audio signal. A dilated convolutional Neural Network architecture is used for audio stream network. Here input size is 298*257*2, where 2 channels comes after real and complex part of STFT is separated into 2 channels. Output of this network is of shape 298*257*8, which is reshaped as 298*(257*8)*1 i.e each channel is concatenated side by side for LSTM network.
Second is video stream network whose input is face embedding of faces detected in the video. In this same model is run through all detected faces with shared weights. The face embedding shape is 75*1*1792, where 75 comes as we are using 3 second video with fps 25 so total 75 frames in 3 seconds. For each frames face embedding is of length 1792. And output of this network is of shape 298*1*256 for each speaker. Which is then concatenated side by side and final shape of video stream is 298*1*(256*num_of_people).
Finally last model is a BiLSTM model followed by 3 FC layers. Its input shape is concatenation of audio stream output and video stream output and thus final shape of input is 298*1*(257*8 + 256*num_of_people). It is passed through BiLSTM model to extract features of sequential data, with a hidden size 400. Thus output after BiLSTM is 298*1*400. It is then passed through series of 3 fully connected layers. And the output shape after FC layers is 298*1*600. Which is then passed through one more FC layer to output a feature map with shape of 298*1*(257*2*num_of_people) followed by sigmoidal compression followed by reshaping it into 298*257*2*num_of_people, i.e predicting complex ratio mask for each speaker. This is then multiplied by input spectrogram of audio stream to give predicted complex spectrogram

for each speaker which is then decompressed with inverse of power of power law compression and then we take ISTFT of all the predicted spectrogram to output separated audio files for each speaker.
For training we are taking MSE loss between all the predicted spectrogram and their corresponding clean speech power law compressed STFT spectrogram. Its obvious as we basically wants to predict clean spectrogram of each speaker, and thus speech is separated.

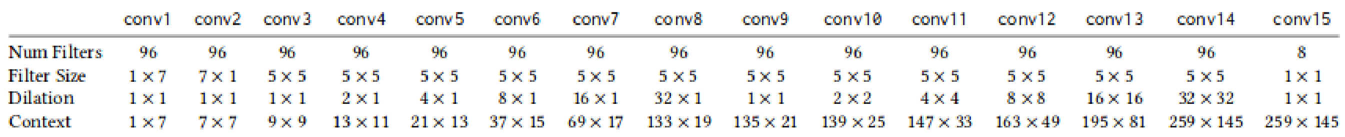
4 Implementation Details

We are using Batch_size of 5 and learning rate is set to 0.0001. Optimiser used is ADAM.

4.1 Network Architecture

We are using same network as used by Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation, just instead of using 3 FC layers we are using using only first FC layer as it increases model parameters too much.

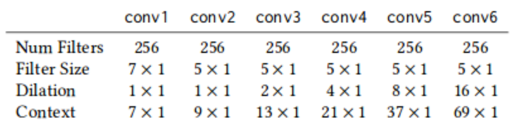
Audio Stream architecture details:-



	conv1	conv2	conv3	conv4	conv5	conv6	conv7	conv8	conv9	conv10	conv11	conv12	conv13	conv14	conv15
Num Filters	96	96	96	96	96	96	96	96	96	96	96	96	96	96	8
Filter Size	1×7	7×1	5×5	5×5	5×5	5×5	5×5	5×5	5×5	5×5	5×5	5×5	5×5	5×5	1×1
Dilation	1×1	1×1	1×1	2×1	4×1	8×1	16×1	32×1	1×1	2×2	4×4	8×8	16×16	32×32	1×1
Context	1×7	7×7	9×9	13×11	21×13	37×15	69×17	133×19	135×21	139×25	147×33	163×49	195×81	259×145	259×145

Figure 1: Audio Stream Model

Visual Stream architecture details :-



	conv1	conv2	conv3	conv4	conv5	conv6
Num Filters	256	256	256	256	256	256
Filter Size	7×1	5×1	5×1	5×1	5×1	5×1
Dilation	1×1	1×1	2×1	4×1	8×1	16×1
Context	7×1	9×1	13×1	21×1	37×1	69×1

Figure 2: Visual Stream Model

Audio-Visual Model :-

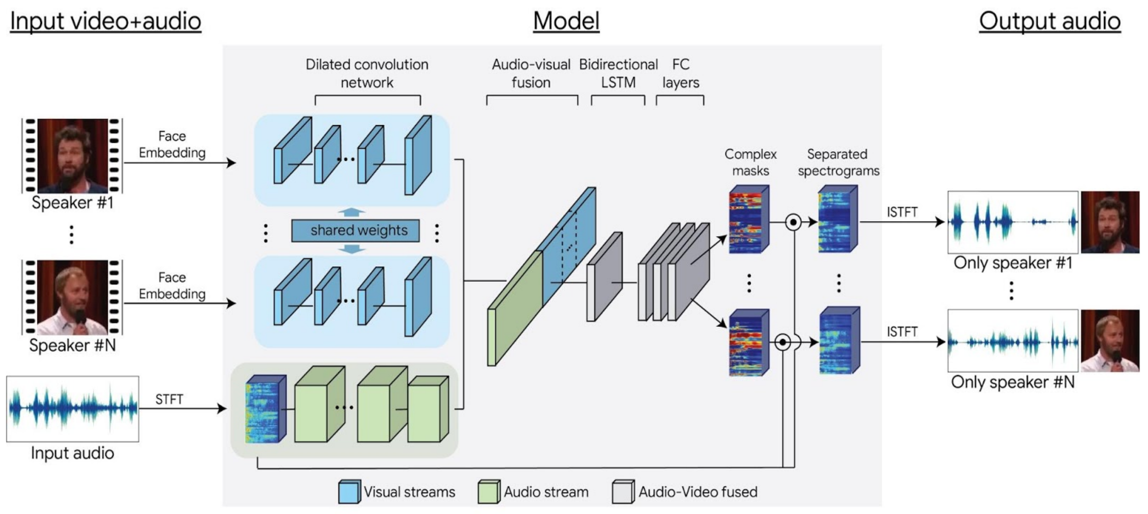


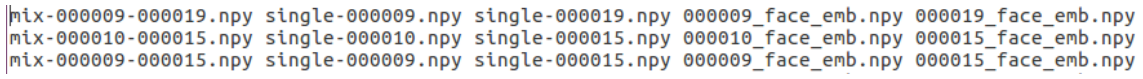
Figure 3: Model Architecture

4.2 Building Dataset

Here we are using AVSpeech data set, it is a CSV file which has 5 columns as follows :-

URL of youtube video	start_time	end_time	x coordinate	y coordinate
u5MPyrRJPmc	108.24	111.24	0.849219	0.305556
H1ulMfj5wRY	112.32	116.94	0.1125	0.345833
VvcwAGkSy2o	240.2	253.366667	0.491667	0.372222

Here x and y coordinates are given in ratio to length of x and y axis of that particular video. Minimum time of each video is 3 and maximum is around 10 seconds. Here we are only using first 3 seconds after start_time. We downloaded audio and video of first 10000 URLs from AVSpeech train csv file.
Then frames are extracted for each video which is total 75 frames for each video. Then all the 75 frames of all the videos are passed through MTCNN open source face detection model. The detected face is checked with the face centres given in AVSpeech CSV file and if detected face is shifted more than some fixed threshold we discard all the 75 frames of that video. So if face is detected correctly in all 75 frames then only we keep it. So after this process we are left with around 1800 video frames. Then we extract face embedding for each 75 frames of all the 1800 videos and store them each in shape 75*1*1792.
One point to be noted is AVSpeech dataset contains videos which have a one speaker only and have a clean audio. We are solving a task of separating 2 speakers, so we manually have to make a database of mix audio files and a log file so that while training their corresponding face embedding and clean speech power compressed STFT spectrogram can be loaded. For the kept 1800 video files we divide them into 2 separate list and combination of them are made, i.e mixing audio of 2 separate list with each other. So total combinations would be 900*900. We took only first 100000 pair samples. Then we calculate power compressed STFT spectrogram of both mix audio and that 1800 single speaker audio and is stored.
Log files looks like :-



```
mix-000009-000019.npy single-000009.npy single-000019.npy 000009_face_emb.npy 000019_face_emb.npy
mix-000010-000015.npy single-000010.npy single-000015.npy 000010_face_emb.npy 000015_face_emb.npy
mix-000009-000015.npy single-000009.npy single-000015.npy 000009_face_emb.npy 000015_face_emb.npy
```

Figure 4: Log file

5 Experimental Setup

We examine scenarios where we add 1 to 1 extra speaker on the clean signal, therefore we generate signals with 2 speakers in total. It should be noted that the task of separating the voice of multiple speakers with equal average "loudness" is more challenging than separating the speech signal from background babble noise.

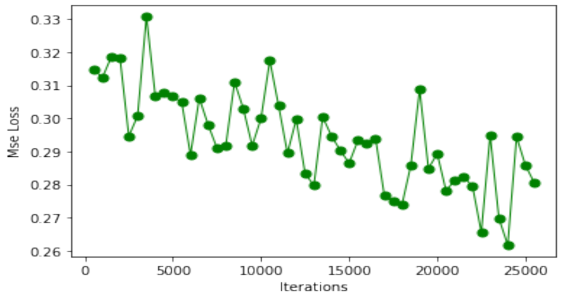
We made a manual Dataloader which takes Log file (see figure 4) as input and which while training randomly selects n (batch_size = n) lines and load mix spectrogram, single spectrograms and face embeddings and give it to audio stream model, target and visual stream model respectively. And while testing it just load mix spectrogram and face embedding.

6 Experiments and Discussion

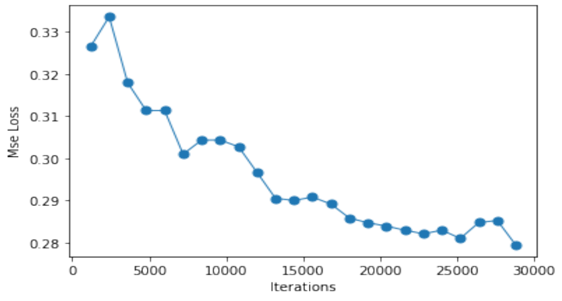
We initially were doing power law compression before STFT by mistake and it was distorting the Spectrogram so be careful that power law compression is applied after taking STFT of the audio signal. One more thing we observed is calculating mean and std of all the face embeddings and normalizing help model to learn more stably and smooth plot is generated as compared to directly using face embeddings.
Initially we are training our model on 70000 pair sample and was validating on 3000 pair sample. But we were facing the problem of not able to generalize well. hence later we took just one speaker from list 1 (see building dataset section) and combined it with around 800 rows of list 2. Out of which we used 580 pairs for training and rest 250 for validation. And we will explain further about its MSE plot on validation and training pairs in Section 7 Error Analysis.

7 Error Analysis

In the paper Separated speech quality is evaluated using signal-to-distortion ratio (SDR). We haven't calculated this value for our separated speech in validation dataset. We have calculated and plotted the MSE loss vs Iterations.



(a) Training Plot

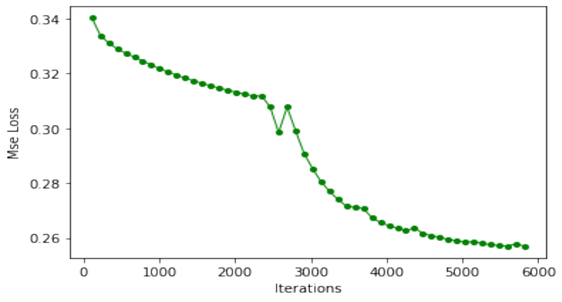


(b) Validation Plot

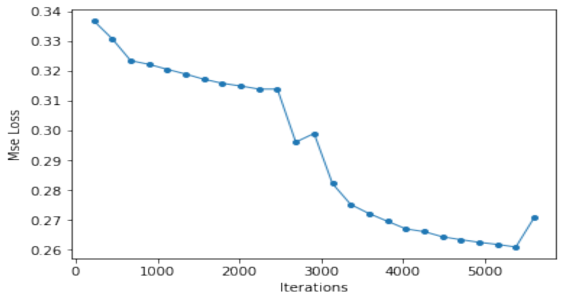
Figure 5: MSE Loss vs Iterations

Here 1 Iteration is 1 gradient step. And as batch size is 5 and we have plotted the mean loss of every 1000 iterations so it means we are plotting mean loss of every 5000 data in training plot. Whereas in validation plot we are using fixed 3000 data to validated and we validate our model after every 1200 Iterations. We can observe Training plot is little fluctuating as one main reason is as each point corresponds to 5000 data average loss, and training dataset size is 70000, so data keeps changing and hence fluctuation occurs but eventually decreasing loss. Whereas in validation plot as the 3000 data is fixed it is less prone to fluctuations, and here fluctuations occurs only when model itself oscillate around some local minima. Validation plot is really smooth as compared to Training plot because of above reasons.

As explained in above section the reason to train on smaller dataset, here we will show the MSE Loss Plot and the output of our model on one of the sample of validation dataset. Here we take average of loss 116 iteration (i.e total 580 data, so here each point in training plot basically corresponds to loss of one epoch) and we validated after every 2 epochs, i.e 232 iterations.

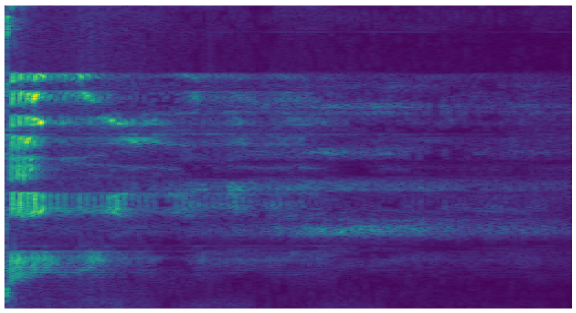


(a) Training Plot

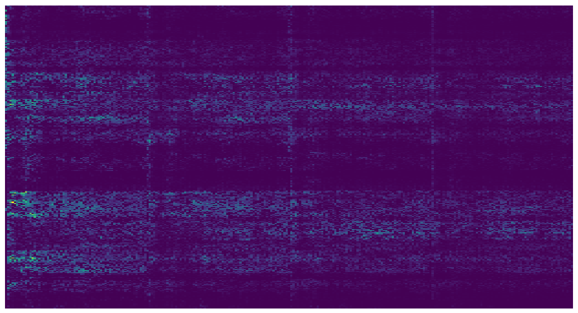


(b) Validation Plot

Figure 6: MSE Loss vs Iterations



(a) Ground Truth Spectrogram



(b) Predicted Spectrogram

Figure 7: Ground Truth vs Predicted comparison

We can see in figure 7 our model was able to learn the pattern to separate audio from mixed audio. While as predicted spectrogram was somewhat distorted so after converting it to audio signal it was not audible and was very noisy, but noise follows the pattern of the clean audio signal, and noise was high whenever speaker says in audio signal and rest of the time noise is very feeble.

8 Summary

We observe that while using small dataset after over fitting on training data model was able to separate spectrograms with some precision. Actually our model is very large, it was taking around 10 sec for each iterations so training on whole data would take 3-4 days. So a better way is to reduce size of audio stream model as there is 15 conv layer. And reducing the number of FC layer help reduce the number of parameters significantly and there is very little drop in SDR as stated in Google's paper.
Future work will be making a model which can separate even large number of speakers in the video with the help of better lips reading model combined with speech seperation model.

References

[1] ARIEL EPHRAT, INBAR MOSSERI, ORAN LANG, TALI DEKEL, KEVIN WILSON, AVINATAN HASSIDIM, WILLIAM T. FREEMAN, MICHAEL RUBINSTEIN Looking to Listen at the Cocktail Party:A Speaker-Independent Audio-Visual Model for Speech Separation *arXiv preprint arXiv:1804.03619*, 2018.

[2] Triantafyllos Afouras, Joon Son Chung, Andrew Zisserman The Conversation: Deep Audio-Visual Speech Enhancement *arXiv preprint arXiv:1804.04121*, 2018.

[3] Florian Schroff, Dmitry Kalenichenko, James Philbin FaceNet: A Unified Embedding for Face Recognition and Clustering *arXiv preprint arXiv:1503.03832*, 2015.

[4] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li and Yu Qiao Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks *arXiv preprint arXiv:1604.02878*, 2016.

[5] Bill Chang Speech Separation *Github repository Github:bill9800/speech*$_separation$, 2019.