Looking to Listen: Audio-Visual Speech Separation Model

Yue Feng (yf2466), Tong Yu (ty2387) and Fan Li (fl2502)

***Abstract*— Our idea comes from Cocktail Party Effect in auditory and speech area. The human beings have an inherent ability to isolate clean speech from a multi-speaker environment. But it's difficult for machines to do it. So we trained a joint audio-visual model to separate each individual speech from a mixed speech. To train our model, we first need to build our video dataset on our own, since there isn't an off-the-shelf dataset. After data pre-processing, we apply face embedding to extract face features, using visual features to help to focus the audio on desired speakers. Mel-spectrogram of mixture is the audio input of the network. And we concatenate audio and visual features, input them into a Bidirectional LSTM architecture. This audio-visual model is speaker-indenpendent (trained once, appliable to any speaker that we meet in the future). The outputs are two multiplicative mask which can be multiplied by the mixture to acquire separated Mel-spectrogram. The separated speech is obtained by inverse Mel converting on Mel-spectrogram.**

I. INTRODUCTION

Imagine you are at a cocktail party, there are several people talking at the same time. For us, we can pay attention to one of them, listen and talk. But for machine, it's hard for it to separate mixed speech into several clear speeches. In order to show this ability on computer, using only audio as input to isolate the sound mixture is extremely challenging. In Aug 2018, Google research team put forward a method that use the combination of both audio and visual sources to implement speech separation. Compared to methods that only use audio as input, the separation quality is improved and the connection between separated speech tracks and speakers is also acquired. Also, permutation problem is better solved by this method. Our project is going to reproduce the idea of audio-visual separation in order to isolate the clean individual sounds from mixture sounds. In addition, we present connections between speaker facial features and the spectrogram of the speech.

II. AUDIO-VISUAL SPEECH DATASET

We collect 50,000 videos on Youtube (for example, TED talks and interviews). For each video, we process it to be a clip, between 3 and 10 seconds long, with no interfering background signals.

A. Data Processing

We download the CSV file from the original paper[1] which contains video addresses and duration from Google website. First, we use ffmpeg to download the videos in the form of Mp4 from Youtube. Second, we separate the intact videos into audios and videos. For video part, we use opencv to extract 75 frames from each 3 seconds' video. Next, we use Google Vision API to extract faces from these frames. Then we use facenet to apply face embeddings to these faces. For audio part, we first combine two audios into one mixture. Then we apply short-time Fourier transform on both mixture and separated audios. Then these signals are passed through Mel frequency filter to acquire Mel spectrogram for both mixture and separated audios.

B. Face Extraction

- We first use an off-the-shelf face detector (google-cloud-vision API) to extract face in each frame and output box coordinates of obtained faces.
- Then we crop face area of each frame using box coordinates and output thumbnails (cropped face images).
- After capturing faces in each video clip, we modify image quality (that is, downsample face images' resolution) to a fixed size: $64 * 64$.

C. Face Embedding

We apply an efficient face verification and recognition deep convolutional network, called FaceNet, to directly learn a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. FaceNet directly trains its output to be a compact 128-D embedding using a triplet-based loss function on Large Margin Nearest Neighbor (LMNN).

1) Triplet Loss: The embedding is represented by $f(x) \in R^d$. It embeds an image x into a d-dimensional Euclidean space. Here, we want to ensure an image x_i^a (anchor) of a specific person is closer to all other images x_i^p (positive) of the same person than it is to any image x_i^n (negative)of any other person. Additionally, we constrain this embedding to live on the d-dimensional hypersphere, $i.e. \|f(x)\|_2 = 1$. The loss is motivated in the context of nearest-neighbor classification. We want

$$\|f(x_i^a) - f(x_i^p\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n\|_2^2$$
$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in T.$$

where α is a margin that is enforced between positive and negative pairs. T is the set of all possible triplets in the training set and has cardinality N. The loss that is being minimized is then

$$L = \|f(x_i^a) - f(x_i^p\|_2^2 - \|f(x_i^a) - f(x_i^n\|_2^2 + \alpha]$$

2) Triplet Selection: Correct triplet selection is crucial for fast convergence. On the one hand, using small mini-batches tends to improve convergence during Stochastic Gradient Descent (SGD). On the other hand, implementation

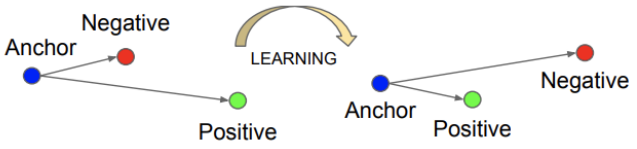

Fig. 1. The Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity.[2]

details make batches of tens to hundreds of exemplars more efficient. The main constraint with regards to the batch size, however, is the way we select hard relevant triplets from within the mini-batches.

D. Audio Processing

We combine two audios from different speakers into one mixture as audio input. The two separated audios are the targets of the train model. The short-time Fourier transform of 3-second audio segment is computed on both mixture and separated audios. We only extract the real part of each time-frequency bin. The sample rate for audio processing is 16000Hz. The FFT window size is 512. The number of Mel filters is 150.

E. Dataset Creation Pipline

We download 50000 samples in total. Among these samples, we pick 100 samples in a selecting pool. First, we separate the videos into audios and videos. For videos, we extract face embeddings for all the 100 samples. For audios, we first normalize the audio signals. Then we merge the first sample and the second sample. Then first sample and the third sample and etc. After combining process, we extract the Mel-spectrogram of the mixture and separated spectrograms. Finally, we store the mixture and face embedding as input and spectrograms as targets. We have 2000 samples for training set, 500 samples for validation set and 500 samples for testing set. Figure.2. shows the over all pre-processing procedure

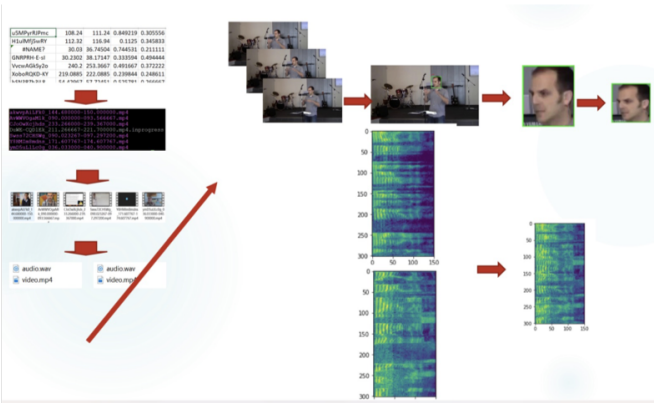

Fig. 2. Our data pre-processing procedures are as above.

III. Model Architecture

We apply face embedding, dilated convolution network and bidirectional LSTM to train our model.

A. Face Embedding

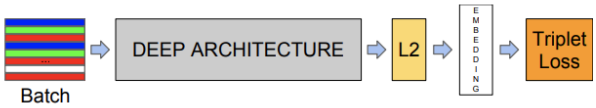

Fig. 3. FaceNet model structure consists of a batch input layer and a deep CNN followed by L2 normalization, which results in the face embedding. This is followed by the triplet loss during training.[3]

B. Multi-stream neural network Architecture

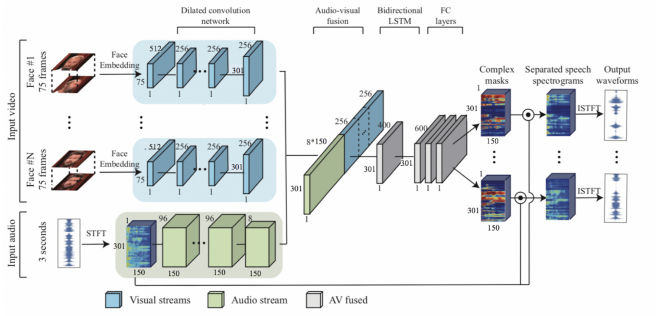

Fig. 4. Multi-stream neural network-based architecture: The visual streams take as input thumbnails of detected faces in each frame in the video, and the audio stream takes as input the videos soundtrack, containing a mixture of speech and background noise. The visual streams extract face embeddings for each thumbnail using a pretrained face recognition model, then learn a visual feature using a dilated convolutional NN. The audio stream first computes the STFT of the input signal to obtain a spectrogram, and then learns an audio representation using a similar dilated convolutional NN. A joint, audio-visual representation is then created by concatenating the learned visual and audio features, and is subsequently further processed using a bidirectional LSTM and three fully connected layers. The network outputs a complex spectrogram mask for each speaker, which is multiplied by the noisy input, and converted back to waveforms to obtain an isolated speech signal for each speaker.[1]

TABLE I
DILATED CONVOLUTION NETWORK PARAMETERS OF VISUAL INPUTS

	conv1	conv2	conv3	conv4	conv5	conv6
Filtered size	7	5	5	5	5	5
Dilation rate	1	1	2	4	8	16

- Network Input: When training the network, the inputs are mixed-audio spectrogram and two separated video frames (each frame contains a vector representing face in that frame).
- Network Output: After the last fully connected layers with sigmoid activation, it outputs a vector that can be separate to two masks. Here the mask is ideal

ratio mask ranging from 0 to 1. Multiplying the masks with mixed-audio spectrogram input gives the separated spectrogram of each input video.

- Network Architecture: Both video steam and audio stream followed by dilated convolution network with batch normalization after each layer. For video input, face vectors of 75 frames are fed into dilated convolution network. The parameters are shown in Table 1. The activation function is ReLU, and each layer has 256 filters.
- After dilated convolution network, it outputs 3 tensors. These 3 tensors are aligned in time domain and concatenated to 1 tensor which will go into a Bidirectional-LSTM (BLSTM) layer. The output of BLSTM layer is fed into 4 dense layers. And the output of the fully connected layers is separated into 2 masks. Separated spectrogram can be obtained by multiplying 2 masks with mixed-spectrogram. Inverse STFT can give the .wav form of separated audio.

TABLE II

DILATED CONVOLUTION NETWORK PARAMETERS OF AUDIO INPUTS

	conv1	conv2	conv3
Filtered size	1*7	7*1	5*5
Dilation rate	1*1	1*1	1*1
	conv4	conv5	conv6
Filtered size	5*5	5*5	5*5
Dilation rate	2*1	4*1	8*1
	conv7	conv8	conv9
Filtered size	5*5	5*5	5*5
Dilation rate	16*1	32*1	1*1
	conv10	conv11	conv12
Filtered size	5*5	5*5	5*5
Dilation rate	2*2	4*4	8*8
	conv13	conv14	conv15
Filtered size	5*5	5*5	1*1
Dilation rate	16*16	32*32	1*1

For audio input, the spectrogram is fed into more complicated dilated convolution network. Each layer has 96 filters except the last layers which has 8 filters.

IV. EXPERIENCES AND RESULTS

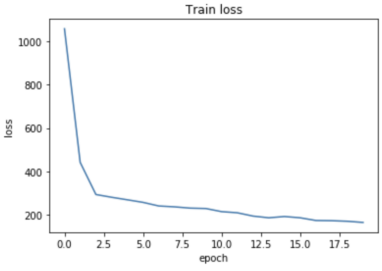

Fig. 5. Training loss.

- Why introducing video into the network?
  Using face embedding can represent face in a high-level context, thus capturing the movement from faces to help separating audio. A figure from original paper can explain this well. See Figure 8. The heat map from this figure represents which part in the face contribute more in the output. We can see that actually the movement of the face affect the output most, which make sense.

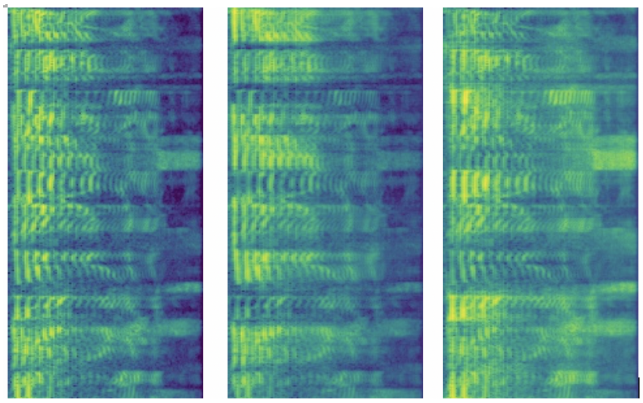

Fig. 6. Mixture Mel-spectrogram (left), predictive Mel-spectrogram for speaker 1 (middle), and predictive Mel-spectrogram for speaker 2 (right).

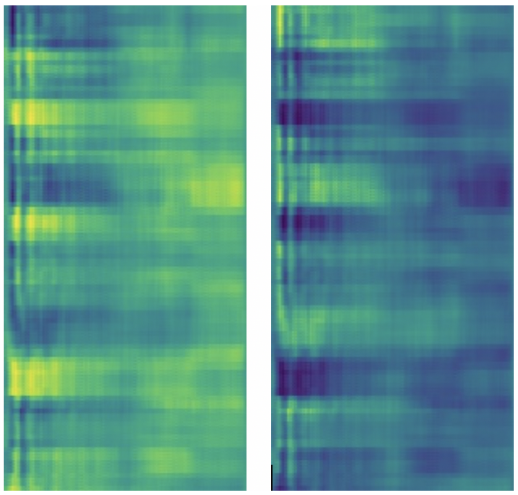

Fig. 7. Mask for speaker 1 (left) and mask for speaker 2 (right).

V. CONCLUSIONS

Audio-visual neural network is a novel model for speech separation. This model works well on speaker-independent speech separation. The performance on multi-speaker mixtures remains to be seen. To train the model, we create a 2000 samples training set and 10000 samples training set. We first train the model in the smaller set to over-fitting and then apply the model to a larger one. We show the output mask of the network and the separated speech. Also, we create one ground truth video demo and one test video demo for comparison. The architecture of the network and

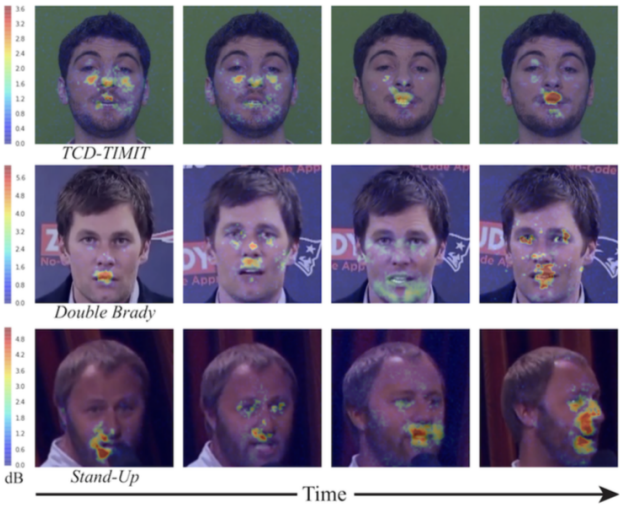

Fig. 8. Heat map represents contribution of different regions on faces.[1]

training performance are also shown.

The result we present at class was not so good is because we did not do speech normalization when mixing the separated video, this will make the network separating audio by determining loudness of audio. We add speech normalization before the network and the result is improved.

ACKNOWLEDGMENT

We wish to thank various people for their contribution to this paper; Google research team, for their great work on selecting the data source; Google Cloud, for credits on computation resources; Professor Andrew Laine and Professor Paul Sajda, for their value suggestions on this paper; Mr. Arunesh Mittal, for his mentoring on improving our skills.

CONTRIBUTIONS

Yue Feng mainly focuses on building the overall network (dilated convolutional network, BLSTM, fully connected layers) and operations about the model.

Tong Yu mainly focuses on platform setup and data processing, face extraction, face embedding (FaceNet architecture) and frame interpolation.

Fan Li mainly focuses on data collecting and preprocessing, audio signal processing, managing files and git.

REFERENCES

[1] Ephrat, Ariel, et al. "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation." arXiv preprint arXiv:1804.03619 (2018).

[2] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[3] Weinberger, Kilian Q., John Blitzer, and Lawrence K. Saul. "Distance metric learning for large margin nearest neighbor classification." Advances in neural information processing systems. 2006.