PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

Report on

Audio-Visual Speech Separation

Submitted by

Chandan Shankar M (PES1201801986)

Hemanth Nag (PES1201801332)

Sukruth C R (PES1201801476)

Abhijna V Maiya (PES1201801311)

Jan - May 2021

under the guidance of

Dr. Shikha Tripathi

Associate Professor

Department of ECE

PES University

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROGRAM B. TECH

CERTIFICATE

This is to certify that the Report entitled

'Audio-Visual Speech Separation'

is a bonafide work carried out by

Chandan Shankar M (PES1201801986)

Hemanth Nag (PES1201801332)

Sukruth C R (PES1201801476)

Abhijna V Maiya (PES1201801311)

In partial fulfillment for the completion of course work in the Program of Study B.Tech in Electronics and Communication Engineering, under rules and regulations of PES University, Bengaluru during the period Jan - Sep 2021. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The report has been approved as it satisfies the academic requirements in respect of Capstone project work.

Signature with date & Seal
(Dr. Shika Tripathi)
Internal Guide

Signature with date & Seal
(Dr. Anuradha M)
Chairperson

Signature with date & Seal
Dr. B. K. Keshavan
Dean - Faculty of Engg. &Technology

Name and signature of the examiners:

1.

2.

DECLARATION

We, (Chandan Shankar M, Hemanth Nag, Sukruth C R, Abhijna V Maiya), hereby declare that the report entitled, 'Audio-Visual Speech Separation', is an original work done by us under the guidance of Dr. Shikha Tripathi Assistant Professor, ECE Department and is being submitted in partial fulfillment of the requirements for completion course work in the Program of Study, B.Tech in Electronics and Communication Engineering.

Place: Bengaluru, Karnataka

Date: 25-09-2021

Name of the student	**Signature**
Chandan Shankar M	Chandan Shankar.M
Hemanth Nag	Hemanth Na
Sukruth C R	Sukruth C.R.
Abhijna V Maiya	

ABSTRACT

The main objective of speech separation is to differentiate between overlapping speeches and gain better insight about the information being conveyed. In this paper, we review previous work in the speech separation domain, analyze them and also propose our own approach. Our main objective was to find a fast and efficient algorithm, so we chose a time domain approach which used audio processing. To improve model accuracy, we took visual data into consideration thereby making our approach 'multi-modal'. We introduce a novel architecture which integrates visual component with the previously available audio-only 'Dual Path RNN' model. Adding visual component to speech separation enables wide array of use cases and applications. Future work on our architecture can potentially lead to real-time implementation of audio-visual speech separation.

ACKNOWLEDGEMENTS

We would like to dedicate the credits of our work to our guide Dr. Shikha Tripathi who has helped us in every step/ stage of our project and noting down points and dedicating their time in guiding us and using those points to useful provide feedback, give us insights on the necessary pre requisites that were highly crucial to our Capstone project.

A special thanks to the team members whose corporation and support was the most important component for the successful completion of this Capstone. Their dedication and contribution are immense and played a vital role. We Would like to thank our honorable Chairperson Dr. Anuradha. We would also like to heartfully thank the librarian who has given us the credentials to access the IEEE explorer and the digital library without which we couldn't have referred to any of the research and journal papers.

We humbly offer our gratitude to the panel members who have given us their valuable time in looking at our work and helping us, providing suggestions that were very helpful for the successful completion of our project. Without their mentoring, reviewing our work, guidance and feedback we would not have been able to complete or even move forward.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Speech is a medium that is used to express our feelings and what we think. Speech signal is made of both time and frequency components. Processing and manipulating these components together or separately would lead to various real-world applications.

Speech signals are commonly prone to environmental or high frequency noise. These noises are mixed with human voice which lie in the range of 100-120 KHz. There are a few techniques that can be applied on these mixtures, namely Speech Enhancement, Speech Separation, Speech Recognition, etc.

Speech enhancement refers to increasing the frequency of the required or desired speech to get an enhanced or clear speech.

Speech Recognition that’s used in Robotic applications and other fields, is applied to recognize the required speaker's voice.

Speech separation is an area with high future scope due to the increased noise pollution in our modern world. It helps to analyze and interpret the required or desired voice in a crowdy environment.

The main difference between Speech Separation and Enhancement would be target signal. In Enhancement, we focus to get enhanced speech from a mixture of single speaker and noise. But in separation, the signal is a mixture of two or more speakers with noise.

1.1 Need for Speech Separation

Separation of speech is mainly helpful in cases like "The cocktail party problem" where we have many people at a single location, making it difficult to recognize and distinguish the voice of an individual. Humans have the ability to extract the separated speech sources from the mixtures of speeches while it is burdensome for an automatic system when we have a single-channel recording of the noisy speech.

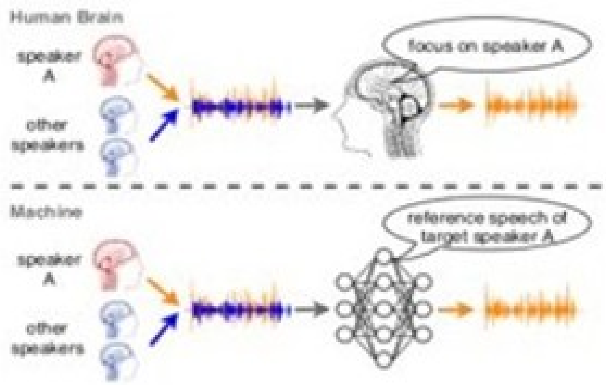


Figure 1.1: Speech processing by Humans vs Neural Nets

Speech Separation will be most helpful in AI assistants to reduce/discard the noise in background when a person speaks. To analyze and give results properly, AI assistants like SIRI, ALEXA uses this method.

Figure 1.2: Speech recognition devices

When it comes to Speech Separation, it has a greater advantage to mankind when applied to solve hearing issues. It can be used to enhance and separate voices in a noisy environment at real-time to give the impaired person a delightful and better life.

It is also helpful in closed environments which are prone to reverberations, background noise, etc. (as shown in the figure 1.3)

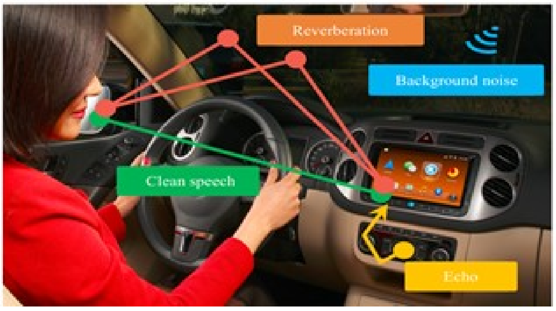


Figure 1.3: Noisy environment scenario

1.2 Why is it important to incorporate visual content?

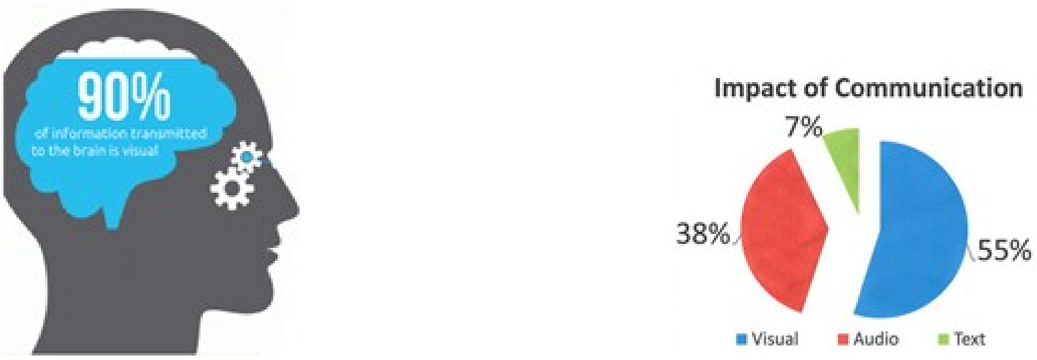


Figure 1.4: Importance of visual content

The eye is the most important sensory organ present. Our brain processes visual content 6000 times faster than any other form of content received.

Most of the information goes to the brain is visual (90%) and the people respond better to visual content than text. When we incorporate the visual data the efficiency and accuracy of the speech separation being done increases.

It's important so that the working of the model becomes efficient and interference errors are resolved when separating the individual voices.

CHAPTER 2

PROBLEM STATEMENT

To obtain enhanced and separated speech of an individual speaker from a mixed audio sample using visual perception.

2.1 Problem Formulation:

Basically, Speech Separation aims to separate the sources with no or little information about both mixing channels and sources.

Consider R independent sources excluding noise

$$X(t) = A * S(t) \text{ ------------------(1)}$$

or

$$y(n) = \sum (h_c(n) * s_c(n) + b(n)) \text{ ------------------(2)}$$

$S(t) = [S_1(t), S_2(t), ...S_R(t)]^T$

$X(t) = [X_1(t), X_2(t), ...X_P(t)]^T$

$[A]_{PXR}$ -> Mixing Matrix which is unknown.

S is clean speech which is also unknown.

X is overall mixed signal which is available.

P- Recorded Microphone Signals.

2.2 An approach to solve the problem

$$Y(n) = \sum_{c=1}^{C} h_c(n) * S_c(n) + b(n)$$ --------------------(3)

C= no of speakers n=time position of each sample

S_c(n)=speech signal of individual speaker h_c(n)=room impulse response

b(n)=background noise

(S_c(n)) * is close to S_c(n)

STFT

$$Y(t,f) = \sum_{n=0}^{N-1} y(n+tL)w(n)e^{-j2\Pi nf/N}$$ -------------(4)

W(n)=Hanning/Hamming window

$$Y(t,f) = \sum_{c=1}^{C} H_c(t,f)S_c(t,f) + B(t,f)$$ ----------------(5)

Then for the above equation get the complex spectrogram (Sc(n)) * i.e., the magnitude spectrum and phase spectrum will be obtained.

Since phase spectrum shows some temporal and spectral regularities, we cannot separate that easily.

$$\angle\widehat{S}_c = \angle Y$$ -------------(6)

Phase spectrogram of mixture signal will be considered as phase of each speaker

Frequency domain speech separation methods mainly focus on how to improve the quality of the estimated magnitude spectrogram |Sc(n)|*

The goal of the separation is to make the estimated magnitude |Sc(n)|* as close/near as the clean magnitude |Sc(n)| for each independent source.

When |(Sc(n)) |* i.e., magnitude spectrogram for C_{th} speaker is estimated the complex spectrogram (Sc(n)) * Could be obtained by combining the magnitude |Sc(n)|* and the phase Y of mixture signal.

Therefore,

$$\widehat{S}_c(n) = \sum_{t=0}^{T-1} V(n - Tl)\widehat{S_{c,t}}(n - tL)$$

-----------------(7)

$$\widehat{S_{c,t}}(n) = \frac{1}{N}\sum_{f=0}^{N-1} \widehat{S}_c(t,f)e^{\frac{j2\Pi f n}{N}}$$

--------------------(8)

CHAPTER 3

LITERATURE SURVEY

3.1 Basics of Speech Separation

There will be three scenarios used to distinguish by taking the number of sources(R) and microphones(P) taking into account

Overdetermined Case (R<P)

This case is achieved when there are more microphones compared to sources.

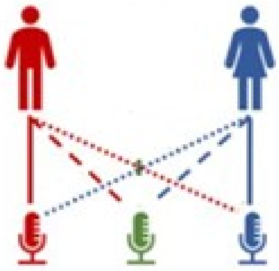


Figure 3.1: Overdetermined case

Determined Case (R=P)

This case deals when sources and microphones are equal.

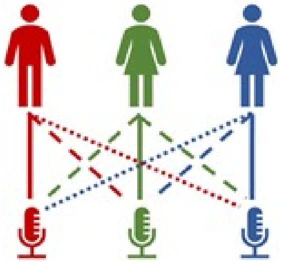


Figure 3.2: Determined case

Under-determined Case(R>P)

In this case the microphones are less when compared to speakers.

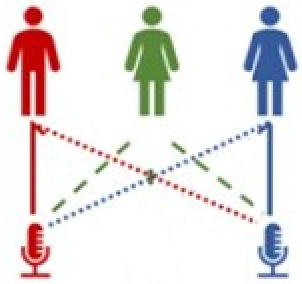


Figure 3.3: Under-determined case

SINGLE CHANNEL	MULTI CHANNEL
Single-channel systems have only the innate information carried by the speech signal itself when spatial cues aren’t present.	In multi-channel systems, spatial information is present and is used as extra information to separate given mixed speech.
Innate information incorporates the pitch of the voice, loudness of the voice and frequency of the speech signal itself.	Spatial information incorporates the distance of the speaker from the microphone, the additional environmental noise and the number of speakers present at a time.
This scenario is very much practical in real-world applications since speech activity is collected by a single microphone in most cases.	It is very impractical in real-world applications since in cocktail-party environments with many sound sources, if the target speaker isn’t predetermined, microphone arrays may not come in handy,

Table 3.1: Single vs Multi Channel

3.1.1 Types of Mixtures:

Instantaneous mixtures

Instantaneous mixture refers to the signal recorded by the microphone wherein it is the linear combination of both noises and the source signals.

Anechoic mixtures

As the name says the anechoic mixture means the signal with the echo sounds wherein mixture contains both direct sound and echoes of the source whereas if s(t) is the source then s (t- δ) refers to the echoed source where δ is the time required for the source to reach the microphone.

Convolutive mixtures

The convolutive mixture is commonly occurring in a closed environment where the room reverberation affects the source to get superposed with these reverberations and lead to the mixture.

3.1.2 Depending on the measurement Process

Linear

X=A*S

Non-Linear

X=exp(s)

3.1.3 Types of Approaches:

Learning-free:

In this learning free approach, it doesn’t depend on any of the training data all the parameters and everything will be analyzed and fixed manually by the user.

Unsupervised learning (Blind Source Separation**):**

In this Unsupervised learning, the user will be blind because he/she doesn't have any knowledge about both the source and the noise only he/she has an overall mixture of both from this mixture it tries to find out the desired output

Supervised learning:

In this Supervised learning, the model will be trained beforehand to get the desired result and ready with the model and at the time of new input feature we will be testing it on the already obtained pretrained model.

3.1.4 Types of Domains:

Time Domain

In time domain we have all the signals individually processed time domain and hence all the tools used in this are related to time.

Ex: -Time division multiplexing

Frequency Domain

The speech is converted from time domain to frequency domain for processing and after desired operation we'll comes back to time domain to get the result.

Ex: - Fourier transform

Time-Frequency Domain

In time frequency domain we will be using both temporal and spectral characteristics of the data hence need to process it in the midway.

Ex: -STFT (short time Fourier transform)

3.1.5 Types of Audio-Visual Integration

Early Integration

In early integration the feature integration is done of the obtained audio and video part.

Middle Integration

In middle integration the feature integration is done after the preprocessing of audio and video.

Late Integration

In late integration technique the decision integration is done after the modelling both audio and video process.

3.2 Classical methods

a) ICA (Independent component analysis): It is an unsupervised extensive machine learning form which is used to separate independent speakers from a mixed signal.

Disadvantage: - The independent sources formed from the ICA should have to be statistically independent and has non-Gaussian distribution.

b) CASA (computation auditory scene analysis): It is an Auditory model and is derived from the human perception of concentrating on any one particular source based on pitch.

Disadvantage: - CASA methods suffer from the main drawback of inaccurate trackers (i.e., pitch tracker)

c) NMF (Non negative matrix Factorization): NMF decomposition model assumes that the audio spectrogram is a low rank matrix. Individual sources are estimated by predicting a weight matrix for each of the basis matrices.

Disadvantage: - The system is unable to generalize to unknown signal that is unseen during training because there is no learned basis for this unknown signal in the basis dictionary.

3.3 Deep Learning Essentials

Basic terminologies:

a) **Perceptron**: Perceptron is a building block of artificial neural nets. It contains a single layer with inputs, weights and an activation function.

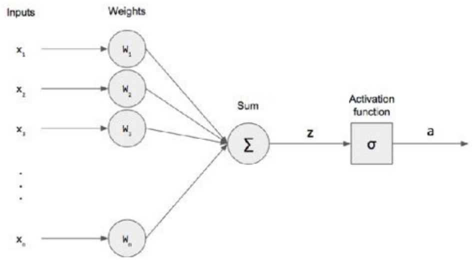


Figure 3.4: Perceptron

b) Multilayer perceptron: Multi-layer perceptron (MLP) is a network of perceptron's. It has three layers as shown below

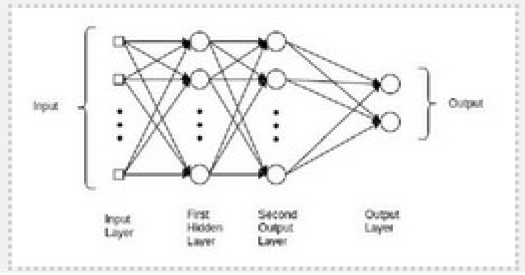


Figure 3.5: Multilayer perceptron

c) Forward pass: It is to determine values of the output layers from the input data. It is said to be completed once we have travelled from beginning to end.

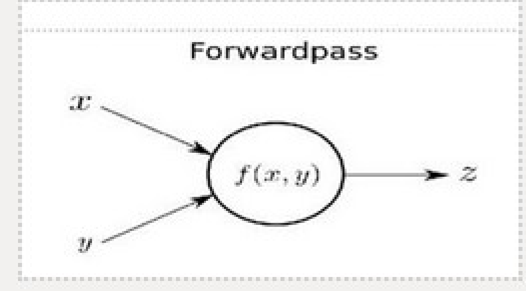


Figure 3.6: Forward Pass

d) Backward Pass (Backpropagation): It is the method of calculating changes in the layer weights, using grad algorithm.

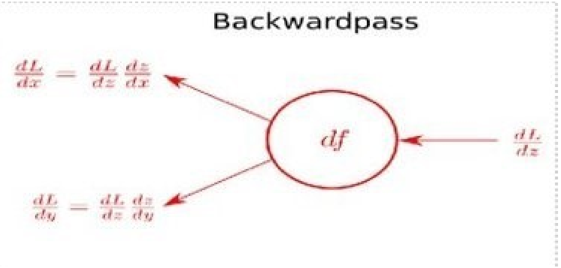

Backwardpass
$\frac{dL}{dx} = \frac{dL}{dz}\frac{dz}{dx}$
df
$\frac{dL}{dz}$
$\frac{dL}{dy} = \frac{dL}{dz}\frac{dz}{dy}$

Figure 3.7: Backward Pass

e) Convolutional Neural Network: It is a special type of neural network that has many convolutional layers. It is essentially sliding a filter over the input.

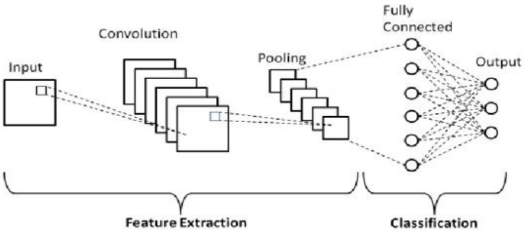


Figure 3.8: Convolutional Neural Network

f) Recurrent Neural Network: It is type of neural network where the output is taken from the previous iteration is fed as input to the current step. Helps in sequential inputs like speech.

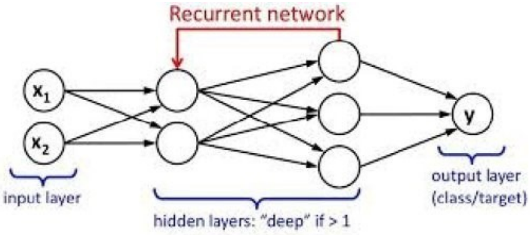


Figure 3.9: Recurrent Neural Networks

g) Long Short-Term Memory (LSTM): RNNs have short-term memory. To overcome this, LSTM, which have gates to decide what's passed on to next cell and what's forgotten are used.

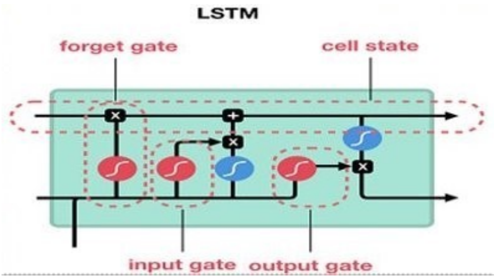


Figure 3.10: Long Short Term Memory

Forget gate: - The result of the forget gate instructs which information to be forgotten by multiplying 0 to the position.

Input gate: - It retains the information to be kept for subsequent stages.

Output gate: - The output gate produces the final output after several iterations and this can be further used for error correction.

h) Gated Recurrent Units (GRU): GRUs are similar to LSTMs, but they have just 2 types of gates. They are faster in training than LSTM because there are fewer tensor operations.

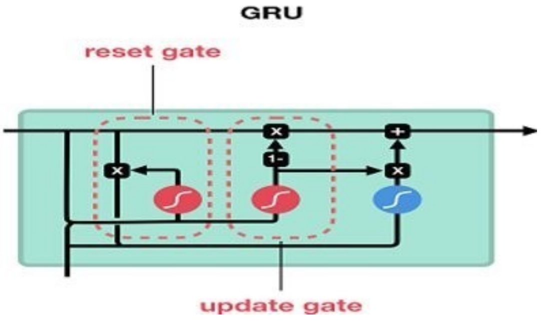


Figure 3.11: Gated Recurrent Unit

3.4 Frequency domain

a) Permutation Invariant Training (PIT):

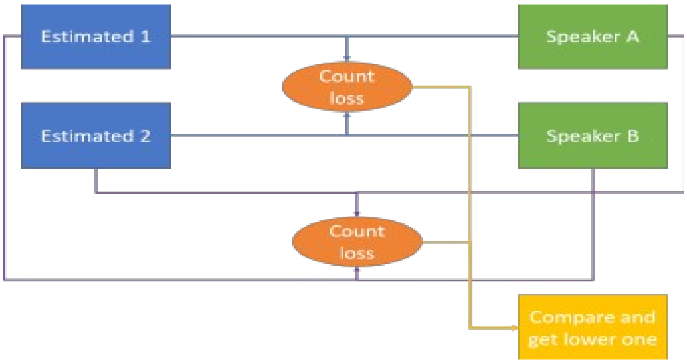


Figure 3.12: Block diagram of Permutation Invariant Training

In Permutation Invariant Training there are two straight output layers each having the output size of the source speech mixture at the rear of the Neural Network. With the help of these layers, the Neural Network evaluates segregates the signals of speaker A and speaker B, from the mixture given at the input end of the neural network. But Neural Network is not aware of the fact that which order of the speakers should appear at the output end layers. hence, it is not advisable to assume that speaker A will be separated with respect to the first linear layer and that speaker B to the second linear layer.

If the Neural Network decides to train with the above presumption, the objective function tends to give a bigger loss value result than expected. Considering the other case, when the Neural Network assumes that speaker A is associated with the second layer and speaker B with the first one, the objective function will give greater loss value results in different cases, but the end result is the same. Neural network will not be trained adequately and as a consequence it won't learn.

This complication can be solved by calculating the objective function results for all permutation of the speakers in the training procedure itself. This can be interpreted as to calculate the objective function in a combination, where the first linear layer output is compared side by side with the target of speaker A and the second linear layer output with speaker B. Then the system can calculate respective objective functions on the first linear layer output compared with the speaker B and the second linear layer output compared with the speaker A. Loss values are computed in each permutation has to be compared and lower one so as to find the lowest possible loss value among all the cases. So, the lower value is sent to Neural Network and is applied in the backpropagation algorithm to train the Neural Network.

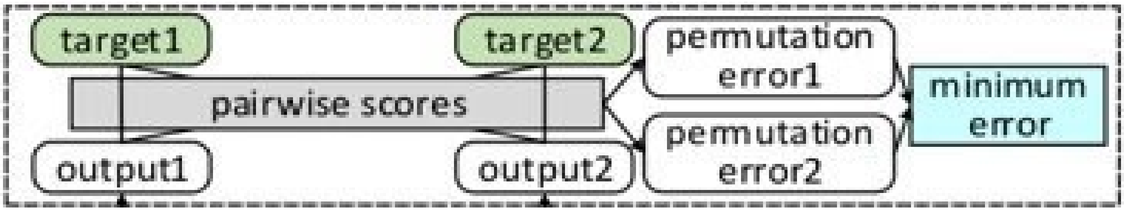


Figure 3.13: Error Analysis in PIT

Disadvantages of PIT:

1. We have to achieve speaker-independent training because most of the time we won't know much knowledge about the speaker/sources and hence we have to train the N/w with different speakers to get a good result in the real scenario.
2. As and when the number of speakers increases the permutation cases increases hence the complexity increases.

b) Deep Clustering (DC):

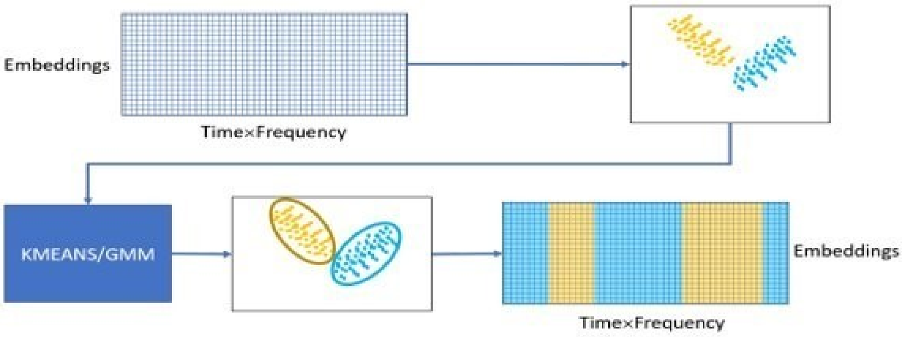


Figure 3.14: Block diagram of Deep Clustering

In Deep clustering (DC) we use the method of embedding vectors for each time-frequency grid estimated. Embedding is referred to the association of time-frequency and discrete values of the vector. These characteristics are very particular for each speaker in the mixture, to make sure that same embedding vectors should belong to that exact speaker.

The loss function in this method is calculated using matrices. For each speaker. the matrix contains zeros for bins, which do not belong to this speaker and ones for that which do. These matrices are estimated as a mask from known speakers and are rounded off to a particular value, to prevent computing loss for low value energy bins.

If the bin doesn’t belong to the same speaker Neural Network is forced to produce values that are situated further apart, where the first given matrix is a three-dimensional matrix of spectrogram associated with deep clustering embedding. The second one is a reshaped version of the first matrix to a two-dimensional space comprised of time and frequency in the same dimension.

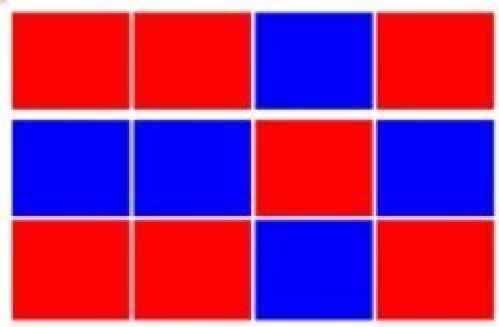


Figure 3.15: Showing Mixture of Two Speakers in Matrix Form

Let the colours red and blue denote two different speakers as depicted in the above diagram. According to ideal binary mask red takes 0 and blue takes 1.

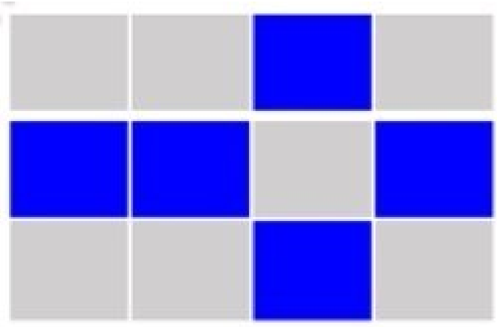
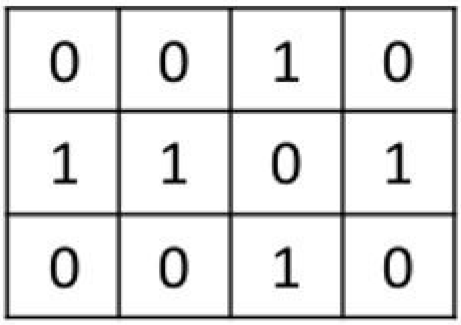


Figure 3.16: Segregated to get Single Speaker

Since the red value is zero, we isolate the blue squares which are assigned the value 1 to get the mask.



0	0	1	0
1	1	0	1
0	0	1	0

Figure 3.17: Mask Generation

Let the mixed signal be taken as a matrix with two speakers and then we have to generate the mask for the mixed signal matrix (ex: -Ideal binary mask) and then by identifying similar mask, we'll obtain the required signal for an individual speaker.

3.5 T-F Domain:

The explained paper is Looking to Listen at the cocktail party by Tali Dekel et.al published by the google AI research team. Here, they are working with the dataset of roughly 290000 YouTube Videos and used a specific API to find faces in each frame and took 3 seconds video for each frame and got 25FPS (Frames per second) hence and a total of 75 frames for an input.

Input features:

Given a video clip containing multiple speakers, a face detector is used to find faces. Then a pretrained face detection and recognition model is used to extract face embeddings for each of the recognized face. Then it learns about visual features using a dilated Convolutional Neural Network.

Coming to audio features, STFT of audio segments is computed and then the network learns an audio representation using a similar dilated convolutional NN.

Both audio and visual features are fused and then it undergoes processing to get a final output. After masking the final output, we get the spectrogram of the same. Taking ISTFT we get the desired output

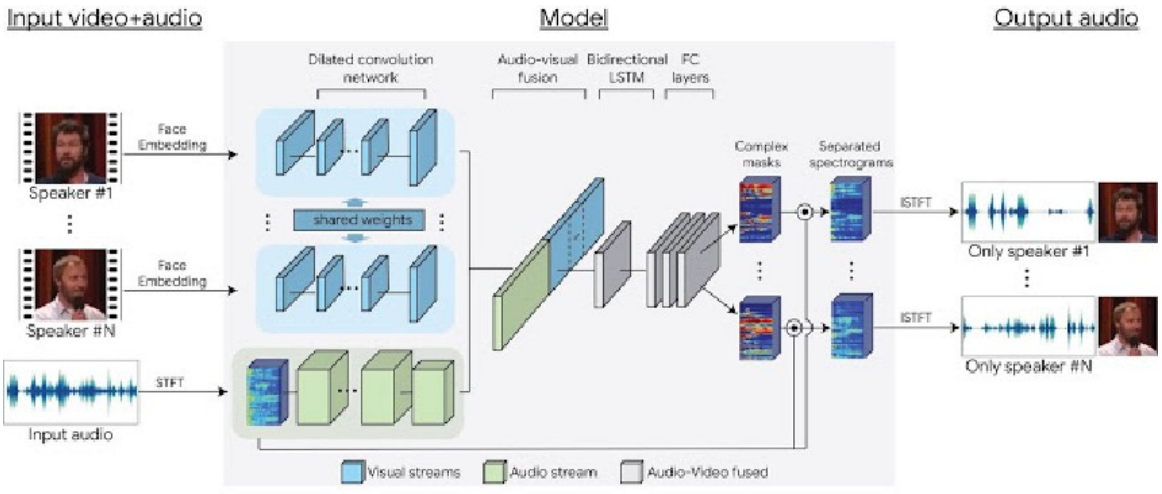


Figure 3.18: Block diagram of T-F Domain Model

Inference from the above model

From the above model we got to know that due to T-F domain modeling the training of the network becomes tougher because of we require more dataset to train and in most of the times it leads to oversampling and for less dataset won’t give minimum required accuracy and hence in telecommunication and real-life scenario it won’t work well hence our approach goes to Time domain.

Advantages and Disadvantages of T-F Domain

Merits:

- T-F domain captures both temporal and spectral characteristics of the input data.
- T-F domain works well for all combinations in gender and hence shows the gender robustness
- Speaker independent.

Demerits:

- Extremely large dataset to perform better.
- Huge time delay due to conversion from one domain to another. This makes real time implementation infeasible.
- This results in phase decoupling.

3.6 Time Domain

a) TASNET (Time-domain audio separation network)

TasNet is said to operate on waveforms smaller than 5ms, this system can be carried out in real-time with very low time latency. Furthermore, TasNet performs better than most STFT-based systems. In other applications that doesn’t require real-time processing, a module relying on separation can also be used to improve the performance.

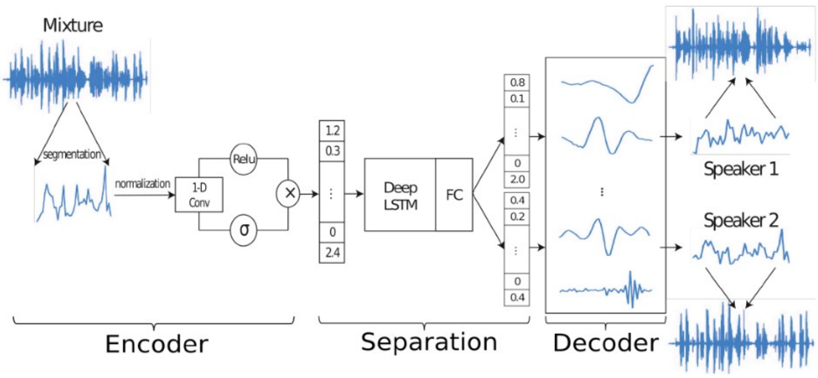


Figure 3.19: Block diagram of TASNET

Encoder:

The analysis and the prediction of the positive mixture having individual weights w_K for a segment k is carried out by a 1D con layer. Coming to the encoder block we use a sigmoidal activation block in which the entire mixture is generalized to have a unit L^2 norm to keep the uniformity and reduce the variations caused in the process.

Separation Network:

The evaluation of source masks can be carried out with the LSTM network to model the time subordinations across all K segments and subsequently followed by a connected layer with a softmax activation function used for the generation of mask. The generated mask is very useful in calculating the weights for the various inputs given to the network.

Decoder:

When the separation mask is obtained, the decoder takes the mask and does operations to get the individual source signals from each of the given inputs and hence the separation is obtained.

b) Conv-TasNet (Convolution time-domain audio separation network**)**

This method unlike other methods discussed before focusses on the time domain approach which overcomes the disadvantages of time-frequency approach.

Conv-Tasnet is a fully convolutional, time-domain only audio separation network, which involves a deep learning cycle for end-to-end time-domain separation. It uses a straight encoder to create a replica of the speech waveform very apt for separating individual speakers. Speaker separation is done by making use of a set of weighting functions which are considered to be masks, to the encoder output. The changed encoder visuals are then reverted back to waveforms using linear decoder.

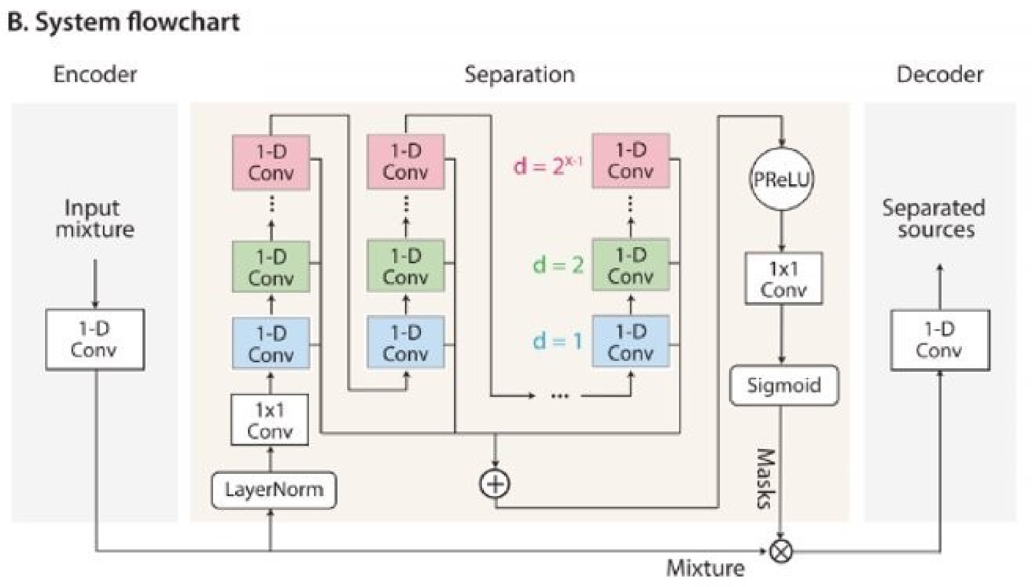


Figure 3.20: Block diagram of CONV-TASNET

A 1D conv encoder changes the waveforms and temporal convolutional network (TCN) separation module estimates the generated masks based only on the encoder output. Varying colors in the 1D conv blocks of TCN denote varying dilation factors which are treated to be constants for improvement in the detailed accuracy of the training model used.

c) Time domain Audio Visual Speech Separation

This approach includes:

- A new multi-modal approach for speech separation is introduced in this paper and a comparison is made with the typical Conv-Net, Conv-Net and PIT to illustrate its effectiveness.
- It has been verified using videos that the model stated performs much better than other baseline models.
- Visual features taken in previous works were usually face. But, here only the lip embeddings are considered since it has more speech data.

The proposed structure is as follows:

- Chunks of raw waveform is fed as input along with corresponding raw video frames.

- The ‘Lip Embedding Extractor’ as shown below is a pre-trained model to extract lip embedding.
- These lip embeddings are passed to the video encoder network. The video encoder contains a few Conv1D blocks. Each conv1D block contains ReLu activation, batch normalization and temporal convolution block. A residual connection is also given in the Conv1D block, but it isn’t of much importance.
- The output of Conv1D blocks is up-sampled and concatenated with output of audio encoder.
- The separation network and audio decoder predict separated speeches directly.
- Si-SNR is used for performance measure.

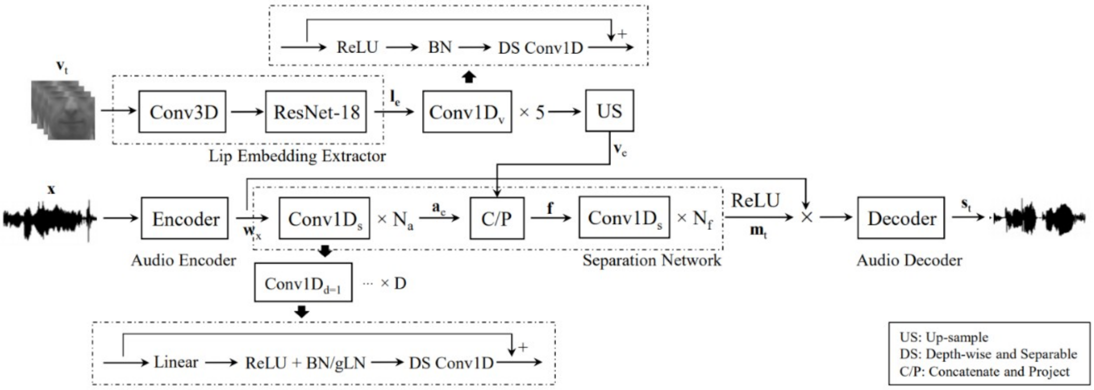


Figure 3.21: Block diagram of Time Domain AVSS

Conclusion:

We get the knowledge of the video encoder and concatenation layer with an audio-only model from this paper. The proposed audio-visual method improves Si-SNR by 3dB and 4dB on 2 and 3 speaker mixtures when compared to the audio-only approach.

Advantages of Time Domain

- Since the model computes in time domain, there are no interconversions.
- Minimal latency
- Faster computation.
- Speaker independent
- Best suitable for both real-time and offline.

- Real-time and low-latency leads to applications in wearable hearing aids and telecommunication devices.

3.7 Comparison Graph

A comparison graph of different models from 2016-2020 versus SI-SDR

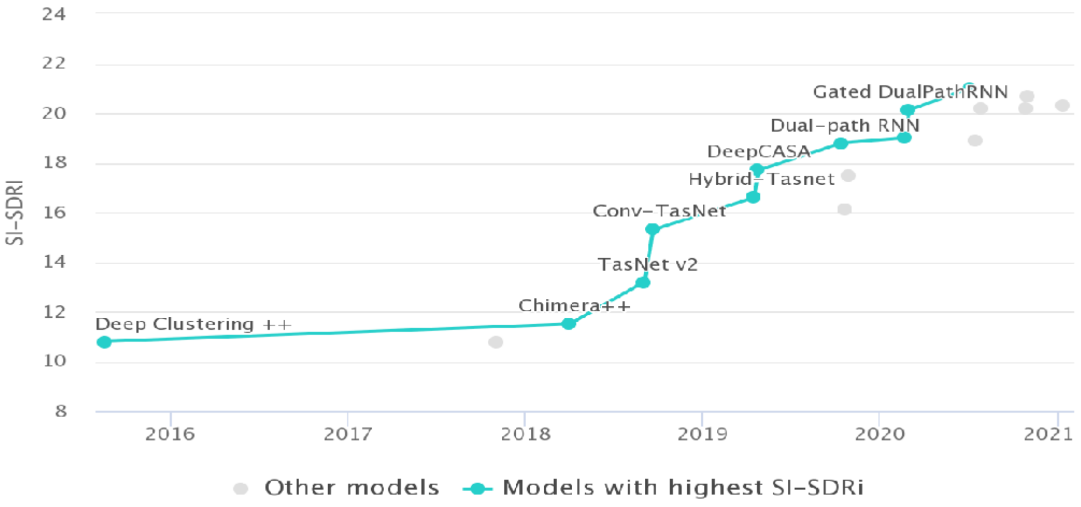


Figure 3.22: Comparison Graph of Different Models

From the above graph we got to know that the time domain models are efficient and most suitable for the real-world applications and we wish to select Dual-Path RNN as our model to implement further in our project.

CHAPTER 4

METHODOLOGIES

4.1 The Audio-Visual Matching Approach:

The audio-visual matching approach considers the visual optical flow data and the static image data of the lip area and integrates it with the spectrogram masks of the audio-only separation model. Since the matching model assigns predicted masks to correct speakers, the problem that exists in the beginning of time frames can be corrected.

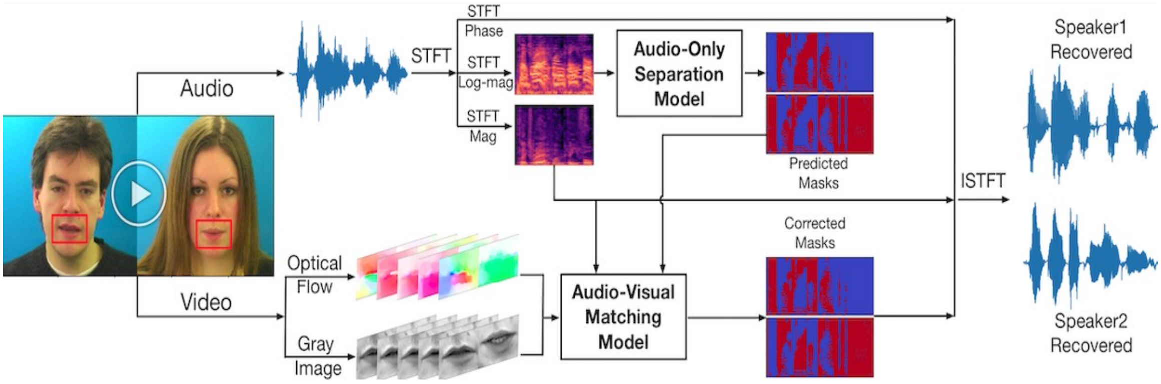


Figure 4.1: Audio-Visual Matching Framework

We get frame-wise embeddings of audio and visual streams. To get a similarity measure, the inner-product of temporally aligned AV embeddings is computed. Assigning separated sources to correct speakers is done using similarity computed, thereby, removing the permutation problem.

4.2 The Audio-Visual Fusion Approach:

The Audio-Visual Fusion model (Audio-Visual Deep Clustering) takes inputs similar to the matching model and predicts T-F masks of speakers directly. The speech mixture's magnitude and phase spectrograms are used to reconstruct the source signal using the predicted masks. Following are the contributions of this approach:

- The two-stage fusion approach outperforms single step fusion in the experiments.
- This model predicts T-F embeddings speaker-wise, thereby removing the permutation problem that happens in audio-only methods.
- This model can also be generalized to an arbitrary number of speaker mixtures. The model is robust with partially observed videos.

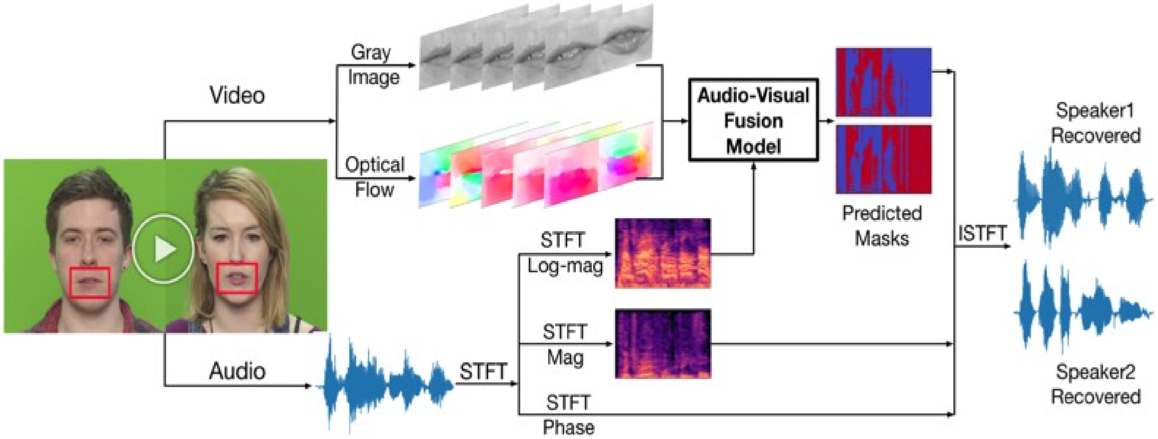


Figure 4.2: Audio-Visual Fusion Framework

We will be using Audio Visual Fusion method in our model.

CHAPTER 5

BLOCK DIAGRAM

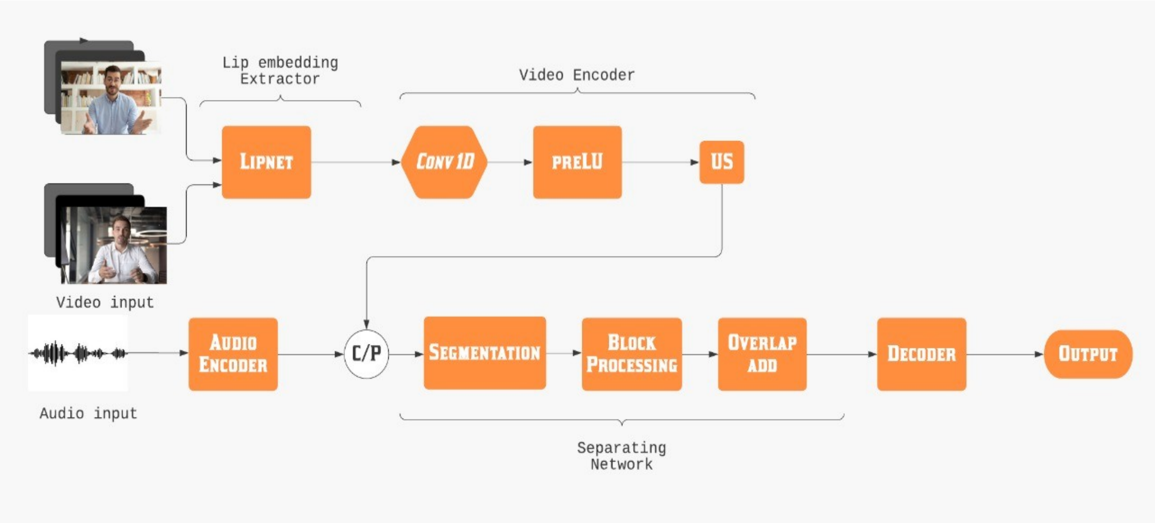


Figure 5.1: Proposed Model Block diagram

- Start with 2 video clips containing each speaker.
- Sample the video at 25 FPS to get frames.
- After getting the frames, use pretrained face recognition model to identify face in each frame.
- Extract the lip region from each face frame and pass through LipNet to get lip embeddings for the given lip.
- Pass lip embeddings to video encoder where we have convolution, PreLu and upsampling layers where in upsampling is used to balance the audio and video data.

- On the other side, mixed audio is given to the audio encoder and then we will be concatenating both audio and video features and propagating it in the separation model.
- The separating network contains three stages. The first stage is segmentation where the sequential input is processed to form a 3D tensor. The second stage is block processing, it processes the 3D tensor and gets the processed block in same shape. At last, in the third stage, perform overlap add on 3D tensor and convert into sequential output.
- Lastly, it is decoded to get the final result.

CHAPTER 6

PROPOSED METHODOLOGY

TasNet and Conv-TasNet explain two different methodologies for speech separation. TasNet consists a network of 1 convolution followed by deep LSTM and fully connected layers while Conv-TasNet contains series of convolutional layers. Since DPRNN has shown better efficiency, we are going for DPRNN.

6.1 Dual-Path Recurrent Neural Network

In this model, the mixed speech passes through three processes before getting the final separated speech.

The three processes are: -

- Segmentation
- Block processing
- Overlap and add

Segmentation:

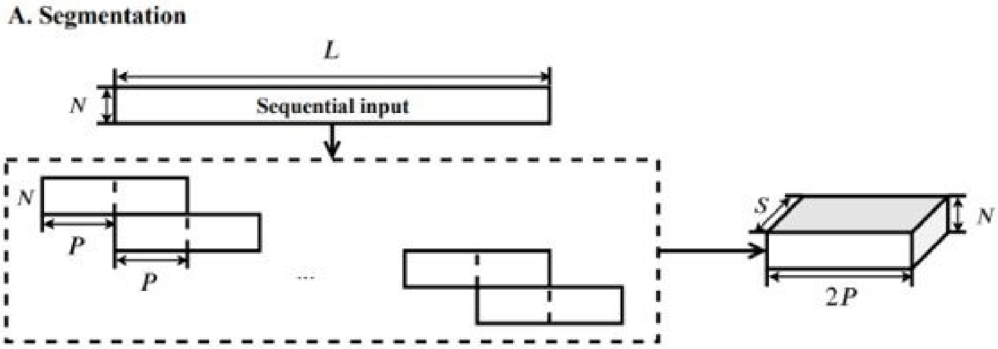


Figure 6.1: Segmentation Block

- In the segmentation stage, sequential input is split into overlapped chunks. The overlapped chunks are all concatenated into a 3-D tensor.
- Consider a sequential input: $V \in R^{N \times L}$, where N is the feature dimension and L is the number of time steps.
- W is split into chunk of length K with hop size of P.
- To make sure all chunks appear in K/P chunks, end chunks are zero padded.
- Finally, a 3-D tensor T is formed by concatenating all chunks.

Here, $T = [A_1, \cdots, A_S] \in R^{N \times K \times S}$

Block Processing:

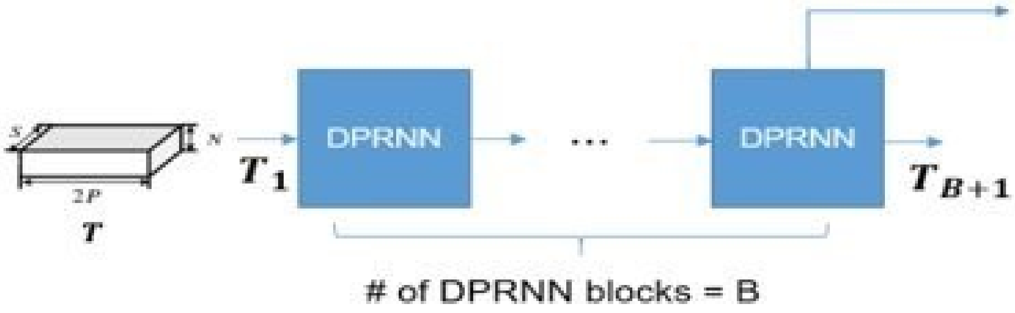


Figure 6.2: Block Processing

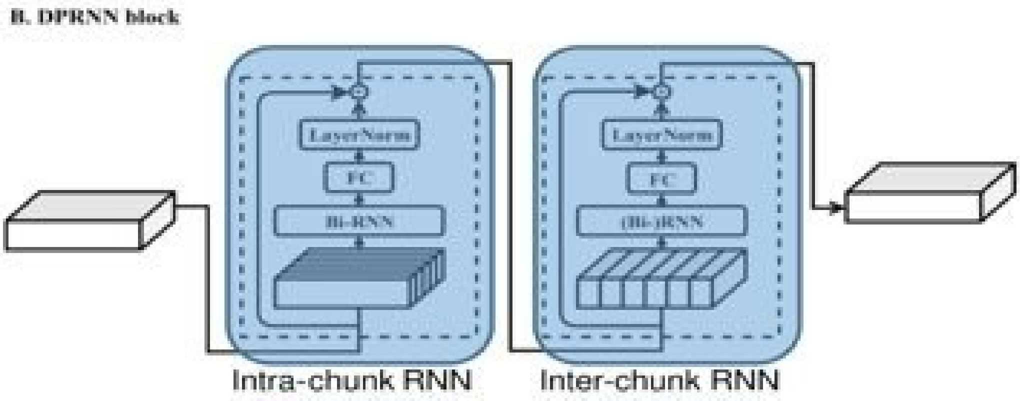


Figure 6.3: DPRNN Block

- The Bidirectional DPRNN block gets the output of segmentation as input.
- The input 3-D tensor is transformed into another 3-D tensor retaining the same shape by each DPRNN block.

- Each of the above blocks contains two sub-blocks named Intra and Inter hunk RNN.

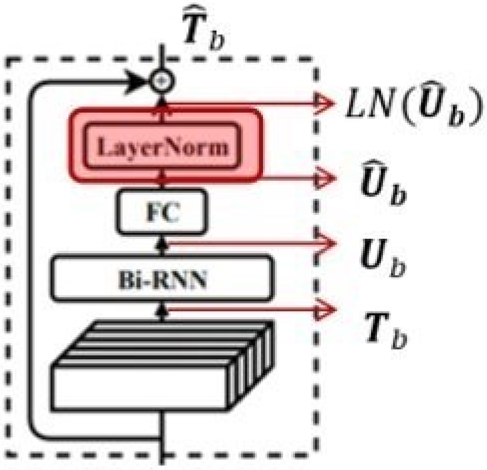


Figure 6.4: Sub Block (Inter or Intra)

Intra-chunk RNN-

- It is applied within each of the S blocks i.e. the second dimension of T_b. Intra-chunk RNN is always bidirectional.

$$U_b = f_b(T_b[; , ; , I]) \qquad i = 1, \cdots , S \quad \text{---------------(9)}$$

Here,

$U_b \in R^{H \times K \times S}$: output of the onsidered RNN

$f_b(\cdot)$: mapping function

$T_{b[}; , ; , I] \in R^{N \times K}$: sequence of chunk i.

Linear Fully connected layer

$$\hat{U}_b = G * U_b [:, :, i] + m \qquad i = 1, . .., S \text{ -----------------(10)}$$

$\hat{U}_b \in R^{N \times K \times S}$: It is the transformed feature

$G \in R^{N \times H}$: It is the weight of the FC layer

$m \in R^{N \times 1}$: Bias

$U_b[\,;\,,\,;\,,\,i] \in R^{H\times K}$: Chunk i.

Layer Normalization

$$LN(\widehat{Ub}) = \frac{\widehat{Ub} - \mu(\widehat{Ub})}{\sqrt{\sigma(\widehat{Ub}) + \epsilon}} \odot z + r$$

----------------(11)

z, r $\in R^{N\times 1}$: Rescaling factors.

ε is +ve

⊙ is the Hadamard product.

μ(·) and σ(·) : mean, varaince

$$\mu(\hat{\boldsymbol{U}}_{\boldsymbol{b}}) = \frac{1}{NKS}\sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{s=1}^{S}\hat{\boldsymbol{U}}_{\boldsymbol{b}}[i,j,s]$$

----------------(12)

$$\sigma(\hat{\boldsymbol{U}}_{\boldsymbol{b}})\frac{1}{NKS}\sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{s=1}^{S}(\hat{\boldsymbol{U}}_{\boldsymbol{b}}[i,j,s] - \mu(\hat{\boldsymbol{U}}_{\boldsymbol{b}}))^2$$

---------------(13)

Final Output from each block, $\widehat{T_b}$ = T_b + LN($\hat{U}_b$)

Inter-chunk RNN

- The input to the Inter-chunk RNN block is the output served by Intra-chunk block i.e. $\widehat{T_b}$. The RNN is applied on the aligned K timesteps of each of the S blocks i.e, the last dimension of $\widehat{T_b}$.

$$\boldsymbol{V}_b = \left[h_b\left(\hat{\boldsymbol{T}}_b[;,i,;]\right), i = 1,\cdots,K\right]$$

-----------------(14)

Here,

$V_b \in R^{H\times K\times S}$: Output

$h_b(\cdot)$: Is the mapping function.

$\widehat{T_b}$ $[;,i,;] \in R^{N\times S}$: Is the sequence given by the i-th timestep in all S chunks.

- Each timestep in $\widehat{T_b}$ contains the information for that entire chunk since Intra-chunk RNN is bidirectional.
- So, Inter-chunk block performs full sequence modeling.
- Layer normalization is applied on V_b after getting it as output from inter-chunk block.

$$T_{b+1} = \widehat{T_b} + LN(\widehat{V_b})$$

------------------(15)

- If (b < B) , the output is served as the input to the next block T_{b+1}

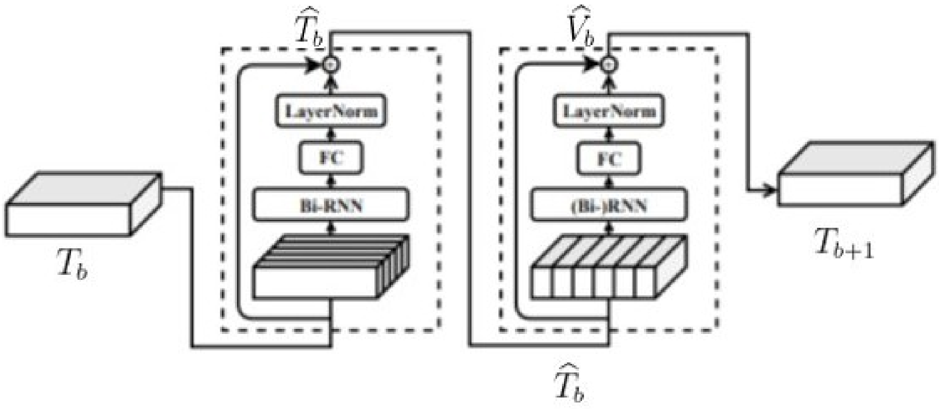


Figure 6.5: Processing in DPRNN Block

Overlap and add:

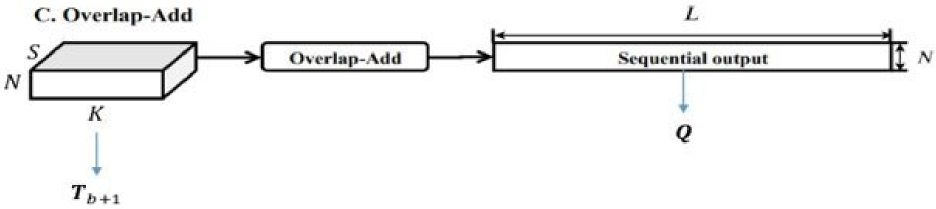


Figure 6.6: Overlap add Block

- Since the final output has to be sequential, overlap-add method is performed on the output of last layer.
- $T_{b+1} \in R^{N \times K \times S}$ will be the output.
- To obtain output sequence of the form $Q \in R^{N \times L}$, apply overlap-add to the chunks of S.

CHAPTER 7

EVALUATION METRICS

There are many Evaluation parameters like:

SAR- Signal to Artifacts Ratio: - It is ratio of sources such as noise, interference and target signal to that of the artifacts.

$$SAR = 10log_{10}\frac{||s_{target} + e_{interference} + e_{noise}||^2}{||e_{artifacts}||^2}$$

-------------------(16)

SDR- Signal to Distortion Ratio: - It is ratio of sources such as target signal to that of artifacts, interference and noise.

$$SDR = 10log_{10}\frac{||s_{target}||^2}{||e_{interference} + e_{noise} + e_{artifacts}||^2}$$

-----------------(17)

SIR- Signal to Interference Ratio: - It is the ratio of the sources such as target signal to that of the interference.

$$SIR = 10log_{10}\frac{||s_{target}||^2}{||e_{interference}||^2}$$

---------------------(18)

SNR- Signal to Noise Ratio: - It is ratio of sources such as interference and target signal to that of the noise.

$$SNR = 10log_{10}\frac{||s_{target} + e_{interference}||^2}{||e_{noise}||^2}$$

-------------------(19)

PESQ- Perceptual Evaluation of Speech Quality: -

It is used to find the quality of the speech signals and it the scores ranges from 0.5-4.5.

STOI- Short-time Objective Intelligibility: -

It is used to compute intelligibility, and the score ranges from 0 to 1.

We’ll be looking into SNR and SDR more deeply.

7.1 Signal to Noise Ratio (SNR) :

It is a measure used to compare the desired signal with noise

It is the ratio of the sources such as target and interference to the noise.

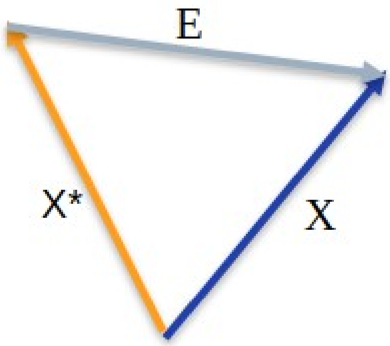


Figure 7.1: SNR Analysis in Vector Form

Let X be the input signal which is a mixture of both required speech and noise and consider E will be the error/noise we have to eliminate to get the desired signal X* so applying triangle law of vector addition we get E=X-X* since we have a formula to find SNR by using that formula, we get the required SNR value

$$SNR = 10log_{10}\frac{||X||^2}{||E||^2}$$

-----------------(20)

7.2 Signal to distortion ratio

It is a measure of the quality of a signal from a communications device.

It is the ratio of the source such as target with the types of noise/error such as interference, noise and artifacts noise

Since it is more realistic such that in speech, we encounter these types of noise and we have to get rid of it hence it become the better performance measurement for our model

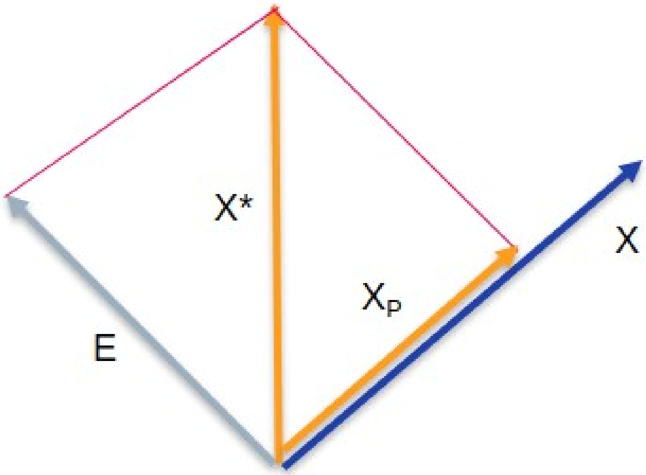


Figure 7.2: SDR Analysis in Vector Form

Let X be the input signal which constitutes both desired signal with the noise and X* be the desired signal since here we'll get the error of E which is opposite to the direction of the desired signal hence, we have to take the projection of the desired signal to that of an error on to the input signal hence then we get the projection vector on to the input signal denoted by X_p ,therefore error will be equal to $E=X-X_p$

$$X_p = \frac{(X^* \cdot X)X}{||X||^2}$$ --------------------(21)

Therefore, SDR will be $$SDR = 10log_{10}\frac{||X_p||^2}{||E||^2}$$ -----------------(22)

CHAPTER 8

APPLICATIONS

8.1 Audio Visual Speech Separation for Robots

Robot perception is an integral part of autonomous robot development. Nowadays, there are many efforts in building intelligent humanoid robots and adding advanced abilities such as speech recognition, object detection, etc.

We have seen voice command processing mini-robots in our day-to-day lives (e.g: Google mini, Amazon’s Alexa, etc.) These robots perform simple tasks and work in a single speaker environment.

But what if it’s a humanoid robot taking important commands through speech. Considering a scenario where two speakers are giving out user specific commands simultaneously, it would be difficult for the robot to decode the commands and to identify who is saying what.

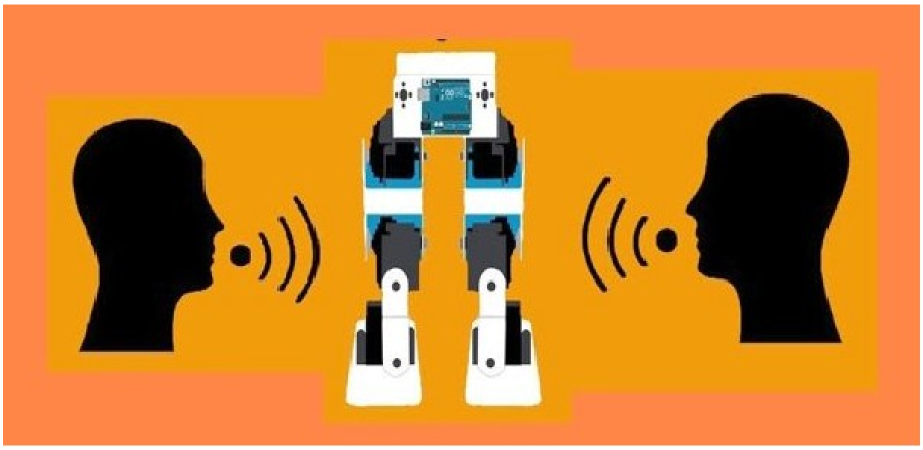


Figure 8.1: Application of Speech Separation in Robotics

This is the best use case of our Audio-Visual Speech Separation model. It would not only help in separating the voice commands with utmost accuracy but also identify which speaker said what.

8.2 Hearing Aid based on Real-Time AVSS

Hearing loss ranks 3rd among most common ailments and it is affecting 466 million individuals in the world, while its severity varies across all age groups.

Disadvantages of the present hearing aids in the market

- It doesn’t block background noise.
- It doesn't separate speech and noise in noisy environments
- It didn't allow people to hear sounds at a from far distance.

The small hearing aids have small batteries that drain quickly. Hence, we require an algorithm/device that would take less time and space to work efficiently over a larger duration of time for the benefit of the impaired person.

Due to the above-mentioned disadvantages of present hearing aids, hearing aids with AI are more advantageous and is a developing area now.

Nowadays, the artificial intelligence hearing aids are not too affordable. Hence, to reduce cost, we either reduce the working time or size. But reducing the size will affect performance. So, we’re looking at the algorithms that work well with less working time and reduced model size.

As mentioned earlier, visual cues play a greater role of interpretation in our brain. If you have observed, it is difficult to understand what your colleague is saying with their face mask on. The same is true with AI models.

So, AVSS hearing aids can help both the deaf and blind. Visual data can also be used for face recognition, so the visually impaired can recognize each individual in conversation with them.

The only constraint for real time implementation is the hardware requirements. There are no small enough processors to process audio visual speech separation in real time.

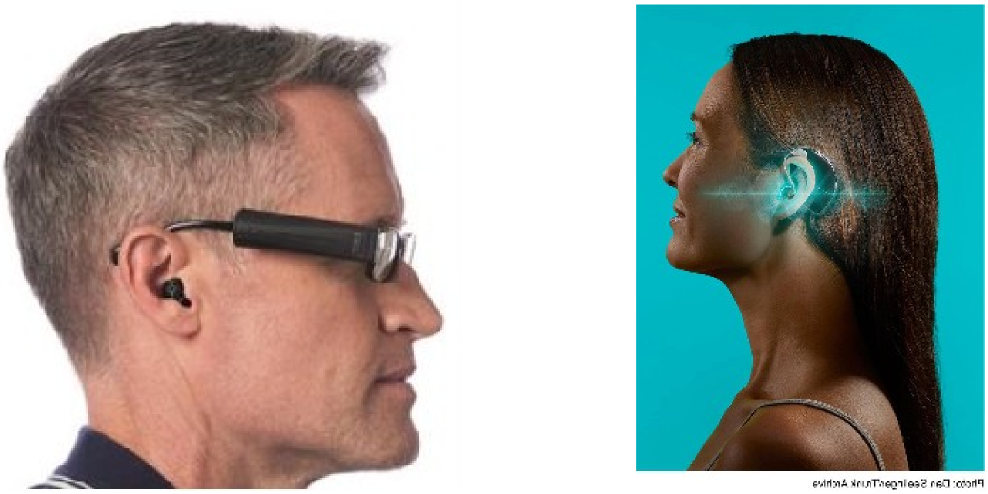


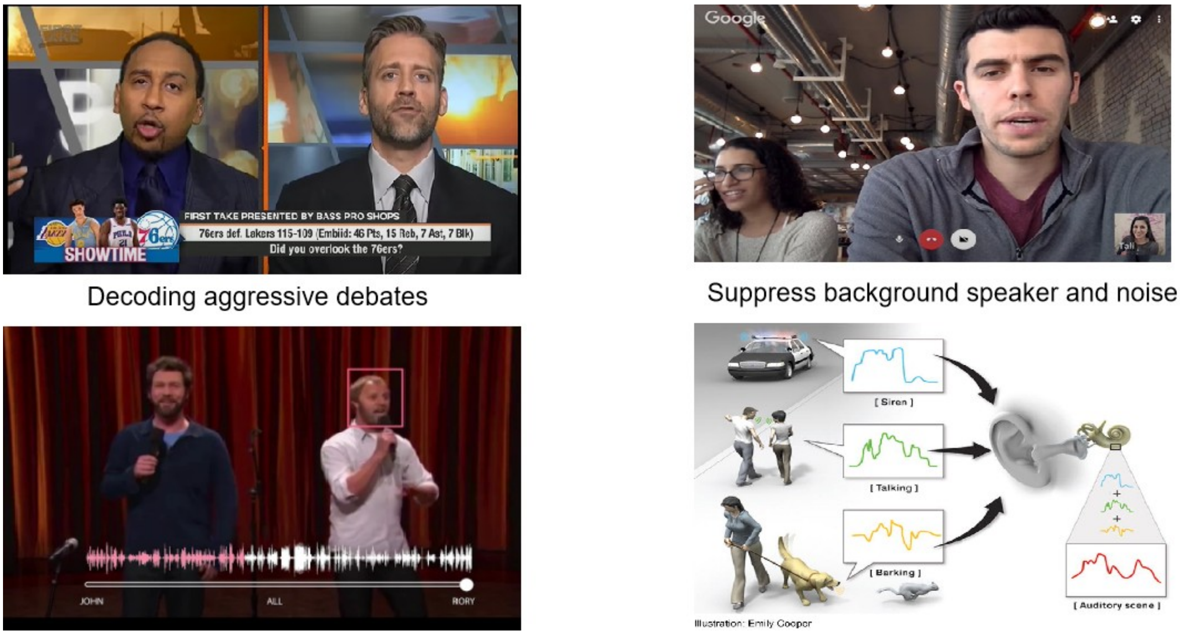
Figure 8.2: Application of Speech Separation in Hearing Aids

8.3 Transcript of a Meeting

If we observe, auto-generated subtitles in YouTube when there are simultaneous speakers, are all messed up. They consist the transcript of both speakers in a mixed manner. This is because the YT's model doesn’t differentiate individual voices. But if we use Audio-Visual Speech Separation on the video first and then generate subtitles on separated speaker audio, the subtitles obtained are clear of any discrepancies. The same can be used to generate transcript of multi-speaker meetings, debates, etc. to assist disabled people.

Figure 8.3: Application of Speech Separation in Transcript of a Meeting

Other Applications:


FIRST TAKE PRESENTED BY BASS PRO SHOPS
76ers def. Lakers 115-109 (Embiid: 46 Pts, 15 Reb, 7 Ast, 7 Blk)
Did you overlook the 76ers?
SHOWTIME
Google
JOHN
ALL
RORY
[Siren]
[Talking]
[Barking]
[Auditory scene]

Decoding aggressive debates

Suppress background speaker and noise

Illustration: Emily Cooper

Figure 8.4: Some more Application Domains

CHAPTER 9

IMPLEMENTATION

9.1 Video Pre-processing

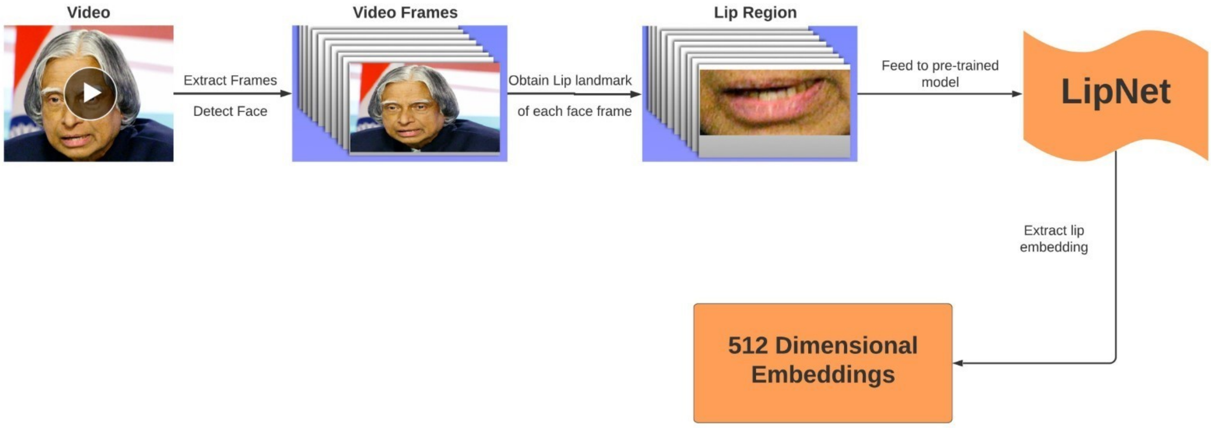


Figure 9.1: Video Preprocessing

- Before we pass visual data into our separator model, we have to pre-process it for better interpretability.
- Out dataset contains videos of individual speakers cropped to just their face.
- Since most of the phonetic (audio) data is concentrated in just the lip area, we focus on lip movement more than the whole face.
- Cropping out lips and passing it to our model isn't effective because it would require extensive training.
- So, we make use of the 'LipNet: End-to-End Sentence-level Lip-reading' pre-trained model.

- Before passing the video frames into the LipNet model, we have to perform certain tasks.

- Firstly, use the face_alignment module to generate facial landmarks of the face in the frame.
- And use affine transformation to align the face to a straight view as shown below.

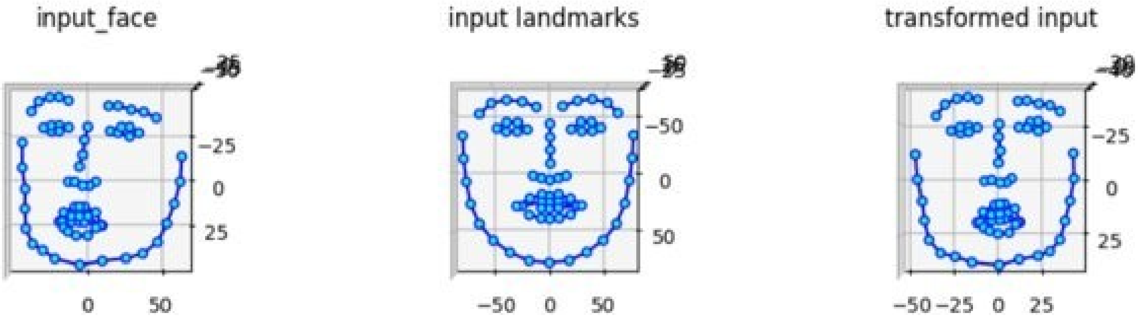


Figure 9.2: Showing working of Face Alignment

- Extract the image co-ordinates centered over the mouth and resize the image into 128x64 pixels.

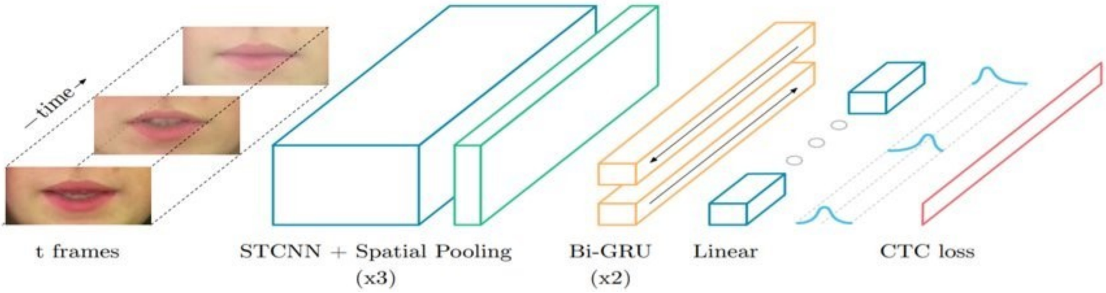


Figure 9.3: Architecture of LipNet

LipNet

- The LipNet model is trained for end-to-end sentence level lip-reading.

- LipNet Architecture: All the lip frames are passed as input. This is further processed by 3 STCNN layers followed by max-pooling. 2 Bi-GRUs process the output from pooling followed by linear layer and softmax.
- But we just want the lip embeddings for our implementation, so we obtain the output of the pre-trained Bi-GRU layer before passing it to the linear layer.
- This is a 512-dimensional vector and serves as lip embedding for the video encoder model.

9.2 Audio Pre-processing:

To train a speech separation model, we'll require both mixed speech and separated speech of each individual in the training dataset. Since it's difficult to obtain it, we just collect videos (with audio) of individual speakers and generate mixed audio from it. The steps to be followed are:

- Extract audio of speaker 1 from the mp4 file.
- Extract audio of speaker 2 from the mp4 file.
- Crop both files to desired length (3 seconds).
- Select or crop a random environmental noise to the same length (required to generate sample resembling real life)
- Overlay all 3 audios thereby generating a mixed audio of Speaker 1, Speaker 2 and noise.

Figure 9.4: Audio Preprocessing

9.3 Audio Encoder:

Block Diagram:

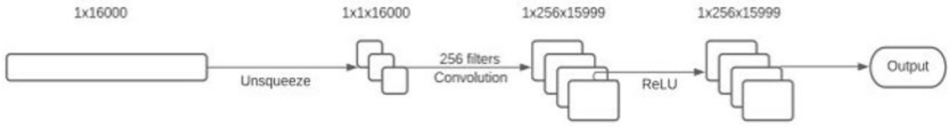
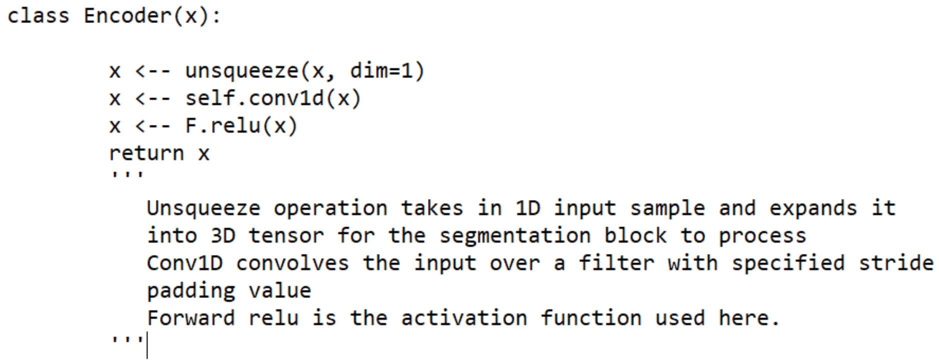


Figure 9.5: Implementation Block diagram of Audio Encoder

Pseudocode:



```
class Encoder(x):

        x <-- unsqueeze(x, dim=1)
        x <-- self.conv1d(x)
        x <-- F.relu(x)
        return x
        '''
           Unsqueeze operation takes in 1D input sample and expands it
           into 3D tensor for the segmentation block to process
           Conv1D convolves the input over a filter with specified stride
           padding value
           Forward relu is the activation function used here.
        '''
```

The audio encoder takes in the mixed audio sequential input and performs the above said operations to get the input in the form of tensor and then feed into the segmentation block.

9.4 Video Encoder:

Block Diagram:

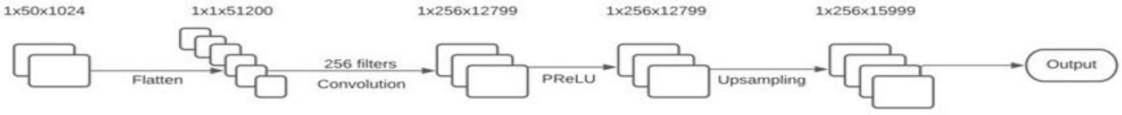


Figure 9.6: Implementation Block diagram of Video Encoder

Pseudocode:

```
class VideoEncoder(x):

        x <-- x.flatten(start_dim=1).unsqueeze(1)
        x <-- conv1d(x)
        x <-- prelu(x)
        x <-- upsample(x)
        return x
        '''
           Usually video samples have less dimensions compared to audio
           hence upsampling is done to match audio with video
           preLU is the activation function use in video decoder
        '''
```

Video encoder takes input of video streams and converts that into frames and then extract faces and then the lip embeddings are obtained from the above said functions.

9.5 Separator Network:

Block Diagram:

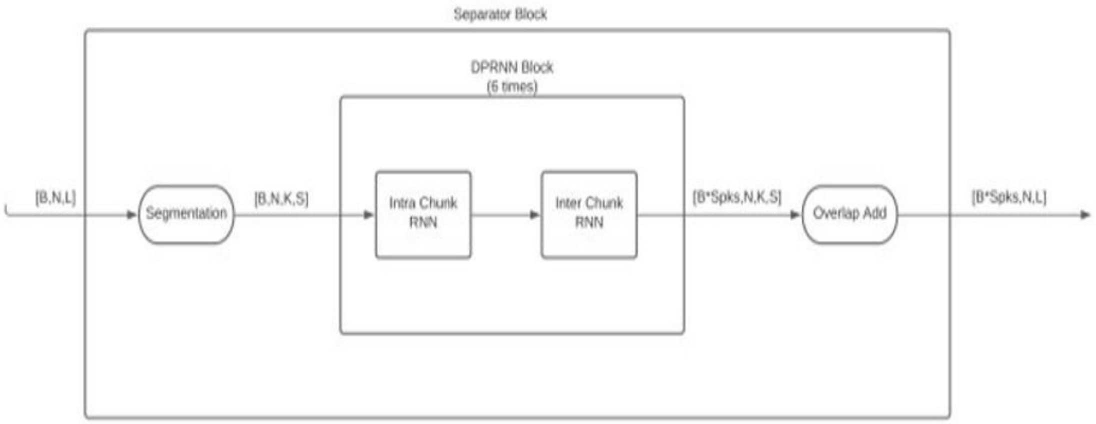
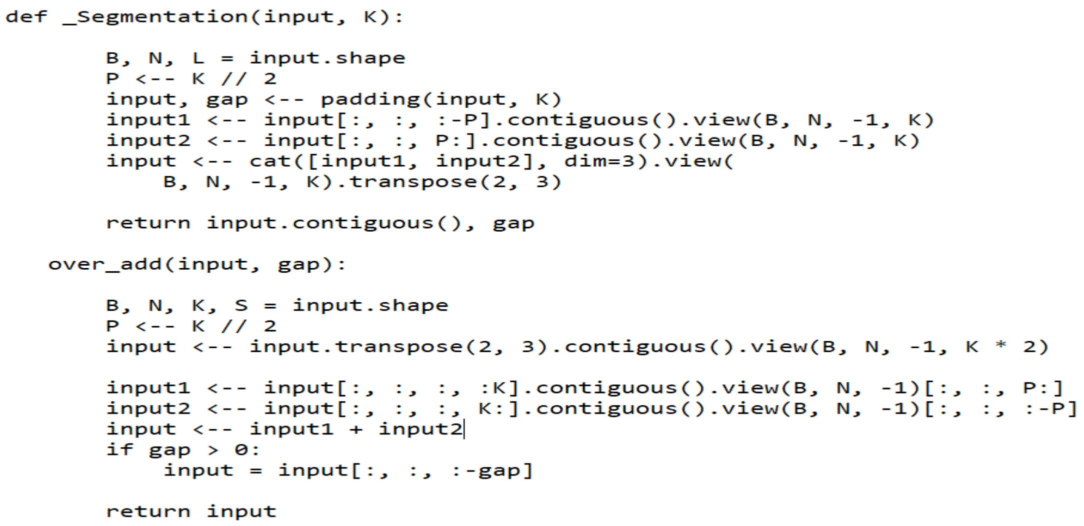


Figure 9.7: Implementation Block diagram of Separator Network

9.5.1 Segmentation Block and overlap add Block:

Pseudocode:

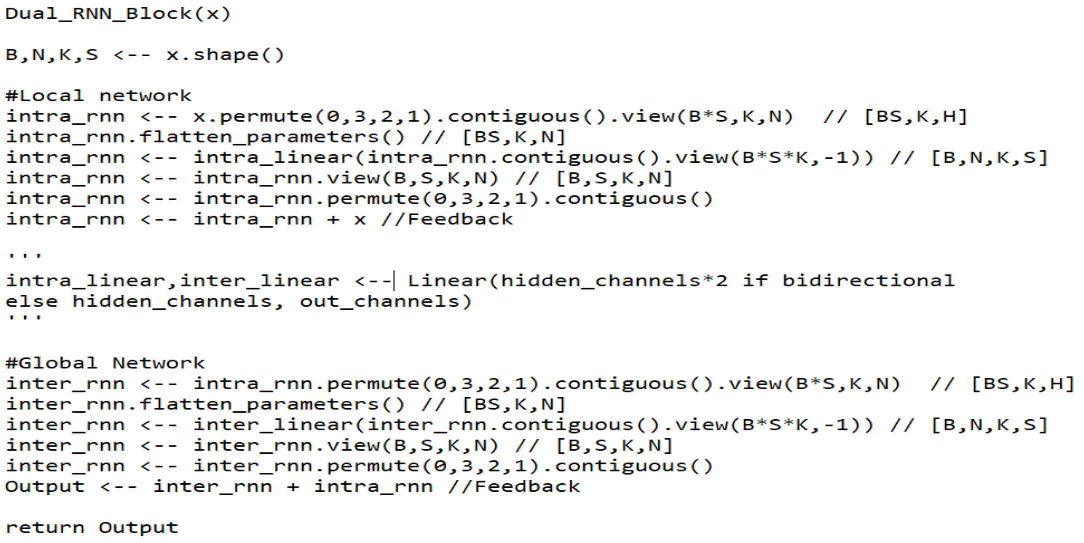


```
def _Segmentation(input, K):

       B, N, L = input.shape
       P <-- K // 2
       input, gap <-- padding(input, K)
       input1 <-- input[:, :, :-P].contiguous().view(B, N, -1, K)
       input2 <-- input[:, :, P:].contiguous().view(B, N, -1, K)
       input <-- cat([input1, input2], dim=3).view(
           B, N, -1, K).transpose(2, 3)

       return input.contiguous(), gap

   over_add(input, gap):

       B, N, K, S = input.shape
       P <-- K // 2
       input <-- input.transpose(2, 3).contiguous().view(B, N, -1, K * 2)

       input1 <-- input[:, :, :, :K].contiguous().view(B, N, -1)[:, :, P:]
       input2 <-- input[:, :, :, K:].contiguous().view(B, N, -1)[:, :, :-P]
       input <-- input1 + input2
       if gap > 0:
           input = input[:, :, :-gap]

       return input
```

This block splits the input in chunks and 1D input sample is converted to 3D tensor the padding/up sampling and concatenation are both done in this step itself. The overlap add function joins these chunks after processing basically works in the opposite way to the segmentation stage.

9.5.2 Dual path RNN Block:

Pseudocode:



```
Dual_RNN_Block(x)

B,N,K,S <-- x.shape()

#Local network
intra_rnn <-- x.permute(0,3,2,1).contiguous().view(B*S,K,N)  // [BS,K,H]
intra_rnn.flatten_parameters() // [BS,K,N]
intra_rnn <-- intra_linear(intra_rnn.contiguous().view(B*S*K,-1)) // [B,N,K,S]
intra_rnn <-- intra_rnn.view(B,S,K,N) // [B,S,K,N]
intra_rnn <-- intra_rnn.permute(0,3,2,1).contiguous()
intra_rnn <-- intra_rnn + x //Feedback
'''
intra_linear,inter_linear <-- Linear(hidden_channels*2 if bidirectional
else hidden_channels, out_channels)
'''

#Global Network
inter_rnn <-- intra_rnn.permute(0,3,2,1).contiguous().view(B*S,K,N)  // [BS,K,H]
inter_rnn.flatten_parameters() // [BS,K,N]
inter_rnn <-- inter_linear(inter_rnn.contiguous().view(B*S*K,-1)) // [B,N,K,S]
inter_rnn <-- inter_rnn.view(B,S,K,N) // [B,S,K,N]
inter_rnn <-- inter_rnn.permute(0,3,2,1).contiguous()
Output <-- inter_rnn + intra_rnn //Feedback

return Output
```

Intra chunk block processes the feature locally whereas inter chunk block processes these features globally as in it takes all the sub divisions and process it as a whole. It takes input from the segmentation block and gives the output to overlap add. In our model we have used 6 DPRNN blocks to get accurate results.

9.6 Decoder:

Block diagram:

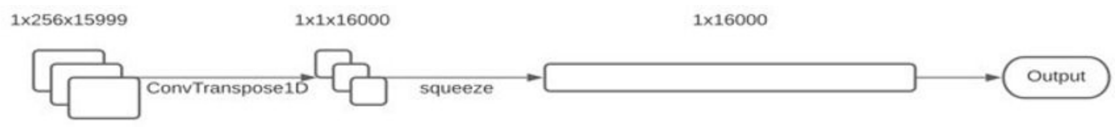
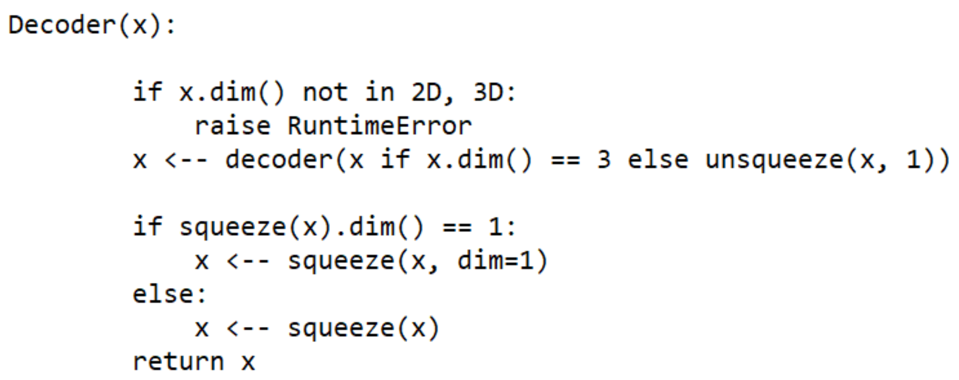


Figure 9.8: Implementation Block diagram of Decoder

Pseudocode:



```
Decoder(x):

        if x.dim() not in 2D, 3D:
            raise RuntimeError
        x <-- decoder(x if x.dim() == 3 else unsqueeze(x, 1))

        if squeeze(x).dim() == 1:
            x <-- squeeze(x, dim=1)
        else:
            x <-- squeeze(x)
        return x
```

Decoder Block takes in input similar to Conv1D. It works as a partial deconvolution in the sense it doesn't specifically deconvolute but acquires its principles.

9.7 Loss Function:

Pseudocode:

```
Loss(ests, egs, input_lengths=None):
      if input_lengths is not None:
          ests <-- [est[:input_lengths] for i, est in enumerate(ests)]
          egs <-- [eg[:input_lengths] for i, eg in enumerate(egs)]
     refs <-- egs
     num_spks <-- len(refs)

     sisnr_loss(permute):

         for s, t in enumerate(permute):
             for num, (est, ref) in enumerate(zip(ests[s], refs[t])):
                 return sisnr(est[:input_lengths[num]], ref[:input_lengths[num]])


     N = egs[0].size(0)
     sisnr_mat =stack(
         [sum(sisnr_loss(p)) / len(p) for p in permutations(range(num_spks))])
     max_perutt, _ = max(sisnr_mat, dim=0)
     return sum(max_perutt) / N        #returns the permutaion which yielded max value
```

Si-SNR function checks the si-snr loss for each of the permutations taken one at a time and Loss function returns the one with the maximum loss.

CHAPTER 10

RESULT AND COMPARISON ANALYSIS

10.1 Loss Plot:

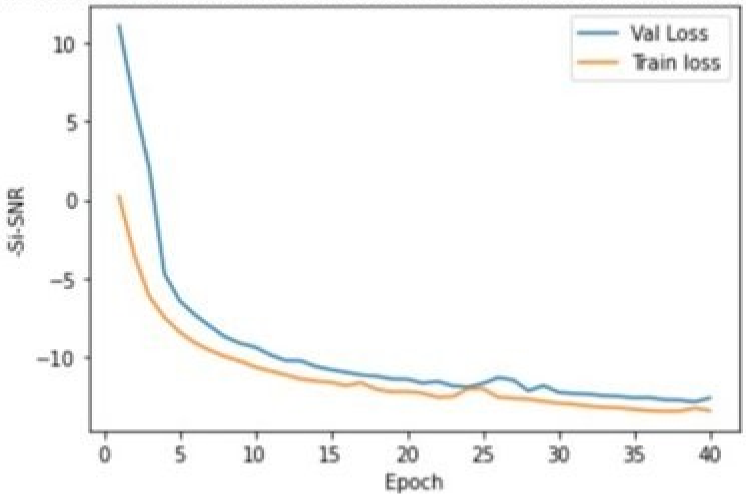


Figure 10.1: Loss Plot

This is a plot of Si-SNR vs epoch where we can see that the blue line represents the validation loss and the yellow line represents the train loss. An interesting thing to notice here is that the y axis takes negative values, this is due to the fact that we Si-SNR as a metric. As we know, a larger SNR value indicates better results, but we choose negative Si-SNR to conveniently depict reducing loss.

10.2 Comparison Analysis:

Model	Type	Dataset	Si-SNR (dB)
Tas-Net (2018)	Audio-Only	WSJ 0-2 Mix	10.8
Conv Tas-Net (2019)	Audio-Only	WSJ 0-2 Mix	12.7
DPRNN (By Yi Luo) (2019)	Audio-Only	WSJ 0-2 Mix	16
DPRNN(GitHub) (2020)	Audio-Only	LRS 3 (50K samples)	10.3
DPRNN (Proposed by Us)	Audio-Visual	LRS 3 (50K samples)	13.2
Time Domain Audio Visual (By Jian Wu) (2019)	Audio-Visual	LRS 2 (60K samples)	13.04

Table 10.1: Comparison Analysis with different models

The table given above indicates the performances of the different models and we have taken LRS 3 dataset. Compared to previous model, our model gives a good Si-SNR ratio. Since our model is based in time domain, we have experimented with limited dataset.

CHAPTER 11

HARDWARE AND SOFTWARE DETAILS

Hardware Specifications:

- **Platform:** Google Colab(Jupyter Notebook)
- **CPU:** Intel Xeon CPU @ 2.30GHz
- **GPU:** NVIDIA Tesla K80 24GB GDDR5
- **RAM:** 13GB
- **HDD:** 68GB
- **Storage Medium:** Google Drive

Software Specifications:

- Python 3.7.10
- PyTorch 1.8.1
- OpenCV 4.1.2
- HTML/CSS
- Flask

CHAPTER 12

CONCLUSION AND FUTURE WORK

First, we learnt the basics of speech separation and surveyed on all three domains referring different research papers, videos and lectures. Based on this survey, we chose the best suitable domain for real-time application which could be implemented with limited resources i.e., Time domain. In Time domain, we chose a fast and efficient algorithm DPRNN and went on about understanding and implementing the architecture. DPRNN splits the sequential speech input into overlapping chunks and performs inter-chunk (global) and intra-chunk (local) processing with two RNNs alternately and iteratively. Then we proceeded to train the model and obtained satisfying results. Through our model we stated that Audio-Visual processing works better compared to audio only. We began considering different test cases like two simultaneous speakers, single speaker with background noise and translation videos. All reported good results proving DPRNN's effectiveness in challenging acoustic conditions. We then built an application with simple UI to test our model and obtain live results.

For future work, the DPRNN can further be optimized by replacing LSTM with efficient networks like GRU, etc... to reduce model's time complexity and size. The project can be derived and improved to make interesting applications like real time hearing aids, individually separated YouTube subtitles, foreign language translation separation, etc.

CHAPTER 13

REFERENCES

- [1] D. Yu, M. Kolbæk, Z. Tan and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 2017, pp. 241-245, doi: 10.1109/ICASSP.2017.7952154.
- [2] J. R. Hershey, Z. Chen, J. Le Roux and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 2016, pp. 31-35, doi: 10.1109/ICASSP.2016.7471631.
- [3] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T. Freeman, and Michael Rubinstein. 2018. Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. ACM Trans. Graph. 37, 4, Article 112 (August 2018), 11 pages. DOI:https://doi.org/10.1145/3197517.3201357
- [4]Ruohan Gao, Kristen Grauman, VisualVoice: Audio-Visual Speech Separation with Cross-Modal Consistency(Jan 2021) arXiv:2101.03149 [cs.CV]
- [5] K. Tan, Y. Xu, S. Zhang, M. Yu and D. Yu, "Audio-Visual Speech Separation and Dereverberation With a Two-Stage Multimodal Network," in IEEEJournal of Selected Topics in Signal Processing, vol. 14, no. 3, pp. 542-553,March 2020.doi: 10.1109/JSTSP.2020.2987209
- [6]Y.Luo and N. Mesgarani, "TaSNet: Time-Domain Audio Separation Network for Real-Time, Single-Channel Speech Separation," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP), Calgary, AB, Canada, 2018, pp. 696-700.doi:10.1109/ICASSP.2018.8462116.

- [7] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 8, pp. 1256-1266, Aug. 2019.doi: 10.1109/TASLP.2019.2915167
- [8] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation,” arXiv preprint arXiv:1910.06379, 2019.
- [9] J. Wu et al., “Time domain audio visual speech separation,” in Proc. IEEE Autom. Speech Recognit. Understanding Workshop, 2019, pp. 667–673.
- [10] E. Vincent, R. Gribonval and C. Fevotte, "Performance measurement in blind audio source separation," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 4, pp. 1462-1469, July 2006, doi: 10.1109/TSA.2005.858005.
- [11] Pierre Comon,Independent componrnt analysis, A new concept?, Signal Processing, Volume 36, Issue 3,1994,pages 287-314,ISSN 0165-1684,https://doi.org/10.1016/0165-1684(94)90029-9.
- [12] Daniel P. W. Ellis,Prediction-driven computational auditory scene analysis PhD thesis Massachusetts Institute of Technology, 1996.
- [13] Schmidt, Mikkel N. / Olsson, Rasmus K. (2006): "Single-channel speech separation using sparse non-negative matrix factorization", In INTERSPEECH-2006, paper 1652-ThuFop.10.