Audio Visual Authentication

By

Talha Yousuf

Spring-2022-MS-RIME 05860 SMME

Supervisor

Dr. Hasan Sajid

Department of Robotics and intelligent Machine Engineering

A thesis submitted in partial fulfillment of the requirements for the degree of

Masters in Robotics and Intelligent Machine Engineering (MS RIME)

In

School of Mechanical and Manufacturing Engineering (SMME) ,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(June 2022)

Thesis Acceptance Certificate

Certified that final copy of MS/MPhil thesis entitled "**Audio Visual Authentication**" written by **Talha Yousuf**, (Registration No **Spring-2022-MS-RIME 05860 SMME**), of School of Mechanical and Manufacturing Engineering (SMME) has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Advisor: **Dr. Hasan Sajid**

Date: _____

Signature (HoD): _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

Approval

It is certified that the contents and form of the thesis entitled "**Audio Visual Authentication**" submitted by **Talha Yousuf** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Hasan Sajid**

Signature: _____

Date: _____

Committee Member 1: **Member 1**

Signature: _____

Date: _____

Committee Member 2: **Member 2**

Signature: _____

Date: _____

Committee Member 3: **Member 3**

Signature: _____

Date: _____

Certificate Of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at Department of Robotics and intelligent Machine Engineeringat School of Mechanical and Manufacturing Engineering (SMME) or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at School of Mechanical and Manufacturing Engineering (SMME) or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Declaration

I certify that this research work titled "**Audio Visual Authentication**" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged / referred.

Talha Yousuf

Reg. No: 276422

Copyright Statement

Dedication

This thesis is dedicated to all the deserving children who do not have access to quality education especially young girls.

Acknowledgments

Glory be to Allah (S.W.A), the Creator, the Sustainer of the Universe. Who only has the power to honour whom He please, and to abase whom He please. Verily no one can do anything without His will. From the day, I came to NUST till the day of my departure, He was the only one Who blessed me and opened ways for me, and showed me the path of success. Their is nothing which can payback for His bounties throughout my research period to complete it successfully.

Talha Yousuf

Contents

List of Figures

List of Tables

Abstract

In bio-related applications privacy is an essential element. While most of the techniques in Deep Learning rely on single modality, spoofing attacks can be minimized by employing multi-modal approaches. Purpose of this research is to develop a technique in which a person will be given some sentences to speak, audio-visual features will be merged and using this amalgam of both modalities, language model will validate if the text read actually validates against the passage given to read. This can be used as an authentication method to check if the user is actually live and hence can prevent the print attacks in case of mobile applications.

Chapter 1

Introduction

The ability to identify what is being said only based on visual information, is a remarkable skill that is extremely difficult for a novice to master. Homophones — separate characters that create the exact same lip sequence (e.g. 'p' and 'b') – make it intrinsically confusing at the word level. However, given the context of nearby words in a sentence and/or a language model, such ambiguities can be resolved to some extent.

Lip reading technology can be used for a variety of tasks, including 'dictating' instructions or messages to a phone in a noisy environment, transcribing and re-dubbing archival silent films, resolving multi-talker simultaneous speech, and improving automated speech recognition performance in general.

The usage of deep neural network models [30, 44, 47] and the availability of a large scale dataset for training [41] are two developments that are well recognised throughout computer vision jobs that have enabled such automation. The lip reading models in this context are based on recently published encoder-decoder architectures for voice recognition and machine translation [5, 7, 22, 23, 46].

Goal of this research is to develop a miulti-modal based authentication system which, if used alongside other recognition/verifications system, can allow for added information security. Automatic biometric identification systems face a significant obstacle in the interactive recognition of individuals. Solution approaches include speaker verification

systems based on image sequences. But as the field of adversarial learning is advancing, several state of the art models based on single modalities can be jsut easily fooled by adversarial attacks.

Apart from that with the growing consumer market there has been considerable increase in the no. of mobile using perople. Each mobile has got few sensors including camera and microphone which can be used not only to collect the datasets for such models but also to make the authentication process more smooth and allowing for use of multiple technologies.

Chapter 2

Background Information

In this chapter some terminologies related to the domain of biometrics and authentic ation will be defined.

2.1 Recognition

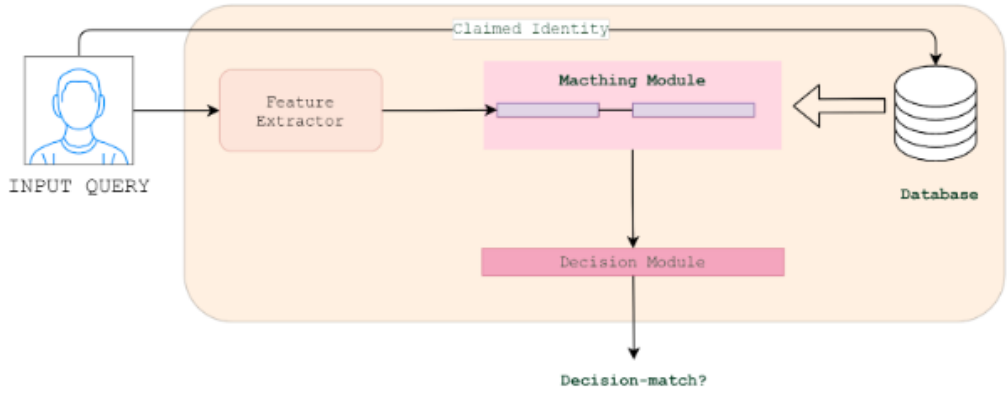


Figure 2.1: Person recognition

Case of recognition refers to a 1xN matching case where the actual question at hand is: "What is the id of person?" So essentially this is how such a system proceeds:

- person presents his identity
- system has a database associated with it
- database has templates of all the registered users
- all templates get matched against the query template

- the template with highest matching score refers to the person ID

2.2 Verification

Case of verification refers to a 1x1 matching case where the actual question at hand is: "Is the claim made by person right or wrong?" So essentially this is how such a system proceeds:

- person presents his identity
- person claims that i am "A"
- system has a database associated with it
- database has templates of all the registered users
- template corresponding to the person's claimed id gets matched
- decision is made if the claim of user is right or wrong

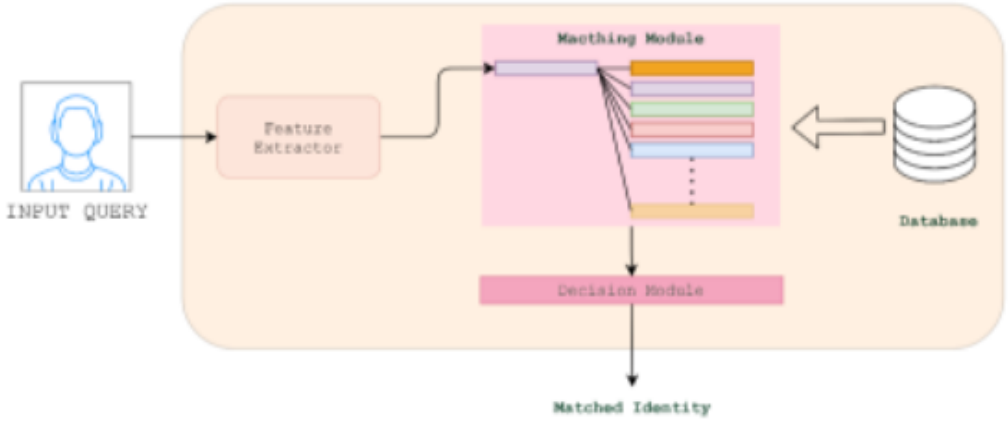


Figure 2.2: Person verification

2.3 Authentication

Authentication assumes that there exists a secret phrase only known by a specific user and which, if validated will allow the user to access the system. Think of it like a

password of a key-word phrase which is just known to a user so the system doesnot know who the speaker / typer is but relying on the authenticity of information provided, it will authorize the user. Here is how a typical authentication system will look like:

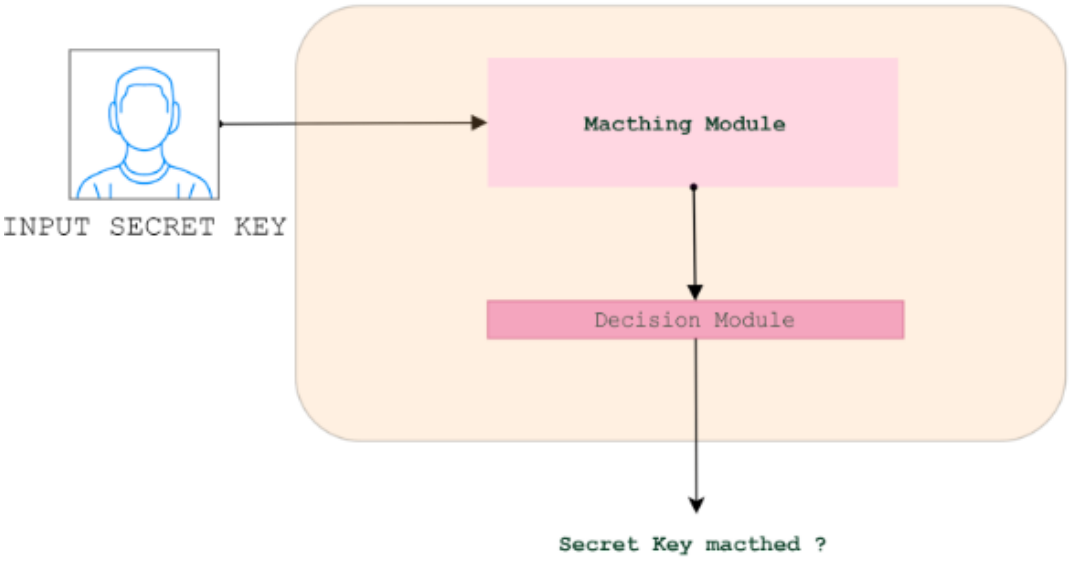


Figure 2.3: Person Authentication

- person types / speaks a secret phrase only known to user and the system
- database matches it against the credentials
- if the credentials are correct, user is authenticated

This can also be seen as a use case where a user is asked to enter the password while the mobile restarts, although the device already has a record of user's fingerprints. This is in a sense two factor authentication for ensuring the security of information systems.

2.4 Motivation

With the ever increasing use of mobile phones and their connectivity with almost every application from email to buying grocery items, information security is a major concern. One of the recent studies[10] performs adversarial attacks on one of the most accurate public face recognition model and finds it to be spoof-able at 98% of the attempts. 98%

is a large number. Advances in deep adversarial learning have made single modality basead systems un-reliable.

There is another dimension to it. Most of the people today use mobile phones and almost all of them have atleast a frontal camera and microphone. So audio-video based multiple modality authentication solutions are totally viable. Apart from that for the disabled people or the people who have got no limbs to type in a password or a pin code will be at more ease using speech based authentication method.

2.5 Scope

Here formal scope of the research will be defined. This researh aims for the development of a multi-modal authentication system. So given that the user has been presented with a secret key phrase and asked to utter it, the system will transform the audio visual features as well as the actual secret key phrase into an arbitrary N-dimensional space where metrics like L2 norm and cosine similarity will be applicable to them.

$$\cos similarity = \frac{V_a - V_s}{\mid V_a \| \| V_s \|} \tag{2.5.1}$$

Model will transform the V_s and V_a both into a projected space where similarity or dissimilarity can be measured on the basis of metrics like:

$$\begin{aligned} &\overline{V_s} \in \mathbb{R}^{384} \\ &\overline{V_a} \in \mathbb{R}^{384} \\ &V_s = spokenphraseembedding \\ &V_a = actualsecretkeywordphraseembedding \end{aligned} \tag{2.5.2}$$

Chapter 3

Literature Review

Foundation of early works in audio-visual speech-based person authentication was laid by [1]. They proposed a method by representing apparent lip movement by change in brightness intensity, then fusing it with speech-audio features and then passing it on to a GMM based feature vector representation. They tested their model on the XM2VTS dataset which contained around 300 identities. This dataset was the state-of-the art of that era and they reported very high and optimistic recognition rates on the said dataset. Apart from being the verification task, "liveness" detection as a byproduct of their method was also proposed [1].

Das et al. proposed a technique based on neural network classifiers. They used nearest-neighbor classification technique in conjunction with neural networks. They captured multiple face views from their in-house generated dataset and proposed a technique to convert voice data into "VSF" spectrograms. They also introduced a method to remove un-voiced portions from the VSF features. These features where fed-in to a neural network and nearest-neighbor classification conjuncture. They reported an EER of 0% on their in-house dataset by testing their method with imposter and real attempts of speaking the spoken password [2]. Later, Das et al. proposed a method to compress the audio spectrograms by to a very light 1D representation by using the FGRAM method. The same methodology was used to convert images of multiple faces to represent them in lower dimension. An improvement to the fusion and training techniques along with additive improvements to the verification pipeline was also proposed [3].

As improvements in the field of person verification grew, Li et al. proposed a detailed overview of an audio-visual biometric system. They discussed in detail the pipeline of

authentication, as well as audio-visual fusion technique. Moreover, they discussed effects of early, mid and late fusion and their effect on overall performance on the system [4].

As deep neural network based techniques grew in popularity, Noda et al. laid grounds for deep-learning based audio-visual speech representation for speech recognition purposes. They used a DNN based autoencoder for denoising and also representing the voice features into a 1D vector representation. And for every time instance they extracted, a 1D feature vector through using a CNN feature vector extractor. These features were fed into a multi-stream implementation of HMM for audio-visual feature integration. They conducted tests on closed-speaker open-vocabulary environment and also included experimentation of the effects of varying SNR and by changing different hyperparameters of their proposed system [5].

[6] has used contrastive loss with novel coupling technique to obtain EER of 13.5%. [8] Used LSTM and proposed and novel Audio-Visual Fusion Strategy to improve CER by 30% (<20%). [9] proposed a novel transformer-based audio-visual fusion technique and CTC loss usage for WER reduction and obtained WER of 48.3% on LRS2-BBC, this was the beginning of use of transformers for fusing multiple modalities. Since their self attention power has made them excel LSTM and traditional temporal models, self-attention based encoder building blocks have found their place in multi-modality based lip reading and AVSR tasks.

[11] has employed temporal convolutions to benchmark his approach against LRW dataset which contains specific words spoken by a number of different speakers. Major contribution was to rectify the limitatations of previous SOTA model for the said dataset which comprised of Resnet+GRU and replace the temporal model with temporal convolutions. Thus instead of training in parts, they simplified the training method in single end2end stage.

Employing the use of temporal convolutions, [12] devised a two-stage network for AVSR purpose in high noise environments. The ist stage separates noise from target voice by employing the use of mouth region movements along with stft based features, thus enhancing the voice. In second stage the enhanced voice is fused with the mouth crop regions to perform speech recognition in extreme noisy environments.

[13] has researched for a number of different features that are good for audio and video modalities.

Chapter 4

Design and Methodology

4.1 Dataset

According to literature review following are the datasets found to be useful for the purpose of this research. LRS2 dataset contains less variety than LRS3 dataset. While the former contains BBC news dataset, latter one has video clips from TED talks. In terms of variety and vocabulary, the LRS3 dataset is the most recent and bulky dataset available for research purposes.

Table 4.1 and 4.2 show statistics and details of these datasets. LRS3 has almost 3.9 million pretraining vocabulary as compared to the LRS2 which has almost 41k unique words.

As for the training data samples, LRS2 has a vocabulary of almost 17k words while LRS3 has 17k.

4.2 Preprocessing

Preprocessing of dataset is necessary with reference to two aspects:

Table 4.1

Set	Utterances	Words	Vocabulary
Pre-train	96,318	2,064,118	41,427
Train	45,839	329,180	17,660
Val.	1,082	7,866	1,984
Test	1,243	6,663	1,698

Table 4.2

Set	Utterances	Words	Vocabulary
Pre-train	96,318	2,064,118	41,427
Train	45,839	329,180	17,660
Val.	1,082	7,866	1,984
Test	1,243	6,663	1,698

	LRS 2 Dataset	LRS 3 Dataset
Type	Sentence Level	Sentence Level
Info.	Thousands of spoken sentences from BBC television. Each sentences is up to 100 characters in length. Disjoint train and test sets	Spoken sentences from TED and TEDX talks with 151,819 utterances in 13 different languages, different DL models are used to synchronize the audio and video at word level. Disjoint train and test sets

Figure 4.1: Datasets Available

- This is multi-modality based fusion scheme and hence both need to be processed in some way to feed to the model
- Without pre-processing the training of transformer based models lags efficiency and stability.

There are a number of steps related to pre-processing but it all depends on the type of downstream task that the model has to perform. Generally spectrograms best suited for audio related inference tasks whereas ML models based feature extractors are used for transforming the video into vectors to be used by some downstream task model.

Following sections will explain these pre-processing steps further.

4.2.1 Text Preprocessing

The corpus of LRS2 and LRS3 datasets includes a number of characters which are not suitable for direct processing. The generic operations performed for pre-processing include :

- text should not contain small letters
- text should not contain special characters

- as for the alphabets, only english ones are allowed
- for numerics, only mathematical characters are allowed.

Common text prep-processing operations

In any machine learning task, data cleansing and preprocessing is at least as important as model creation. This process is even more crucial when it comes to unstructured data, such as text.

Common text preprocessing and cleaning steps include:

1. Lower casing
2. Removal of Punctuations
3. Removal of Stopwords
4. Removal of Frequent words
5. Removal of Rare words
6. Stemming
7. Lemmatization
8. Removal of emojis
9. Removal of emoticons
10. Conversion of emoticons to words
11. Conversion of emojis to words
12. Removal of URLs
13. Removal of HTML tags
14. Chat words conversion
15. Spelling correction

There are two distinct methods for sequence-decoders:

Table 4.3: Vocabulary for the CTC decoder

Character	Index	Character	Index
A	0	S	20
B	1	T	21
C	2	U	22
D	3	V	23
E	4	W	24
F	5	X	25
G	6	Y	26
H	7	Z	27
I	8	0	28
J	9	1	29
K	10	2	30
L	11	3	31
M	12	4	32
N	13	5	33
O	14	6	34
P	15	7	35
Q	16	8	36
R	17	9	37
space	18	<EOS>	38
’	19		

- sequence2sequence
- ctc

For the sequence to sequence models, vocabulary size is large but the model correctly predicts the words since the actual vocabulary is predicted over the whole vocabulary corpus. This makes them very good for language translation tasks for which the vocabulary might increase upto $30k$ tokens. But to generate such a large vector against each temporal unit is compute expensive so character level decoder has been chosen for the said purpose. Following table represents the character to vocabulary index.

4.2.2 Audio Preprocessing

Each of the input videos is processed before starting the pre-training and training process. Since the model depends on multiple modalities so length of both the modalities needs to be somehow fixed in order to synchronize them. More specifically audio information needs to be converted to a matrix in which each column represents information

against a specific time step. Model doesnot directly operate on the raw video and audio data and latent features from both modalities need to be saved to disk prior to pre-training and training to lessen the amount of training time.

Figure 4.2 shows all the steps involved audio processing.

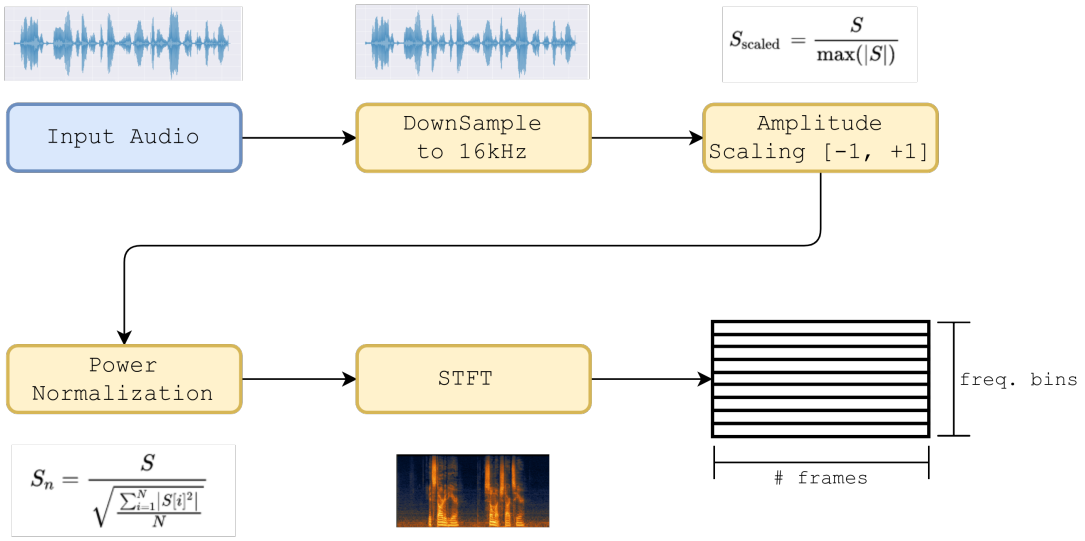


Figure 4.2: Audio pre-processing pipeline

A pipeline has been developed for processing each of the audio files. Since dataset size is large and it is time taking to perform audio processing at runtime or during training hence audio from all the videos has been extracted even before pretraining and training and saved to disk in the form of .wav file.

Briefly these audio processing steps have been summarized below:

1. Perform downsampling to $16kHz$
2. Ensure that length of audio is atleast equal to 4 STFT window sweeps
3. $[-1, +1]$ normalization
4. Power normalization
5. Perform STFT according to parameters defined in the Table 4.4
6. Perform zero-padding on both sides of signal such that audio samples are $4\times$ video frames

In the later sections, all steps will be explained along with their importance.

Downsampling

After all the videos have been processed, another iteration of processing is applied which iterates over the .wav files. Since audio information is generally recorded at 44kHz which is fairly large sampling rate, there is a need to down-sample audio and according to most of the AVSR systems 16kHz is enough, all wav files are resampled at the said value of 16kHz. Figure 4.3 illustrates the sample signals. Top image represents orignally recorded image at $44kHz$ while the bottom one represents $16kHz$ sampled image.

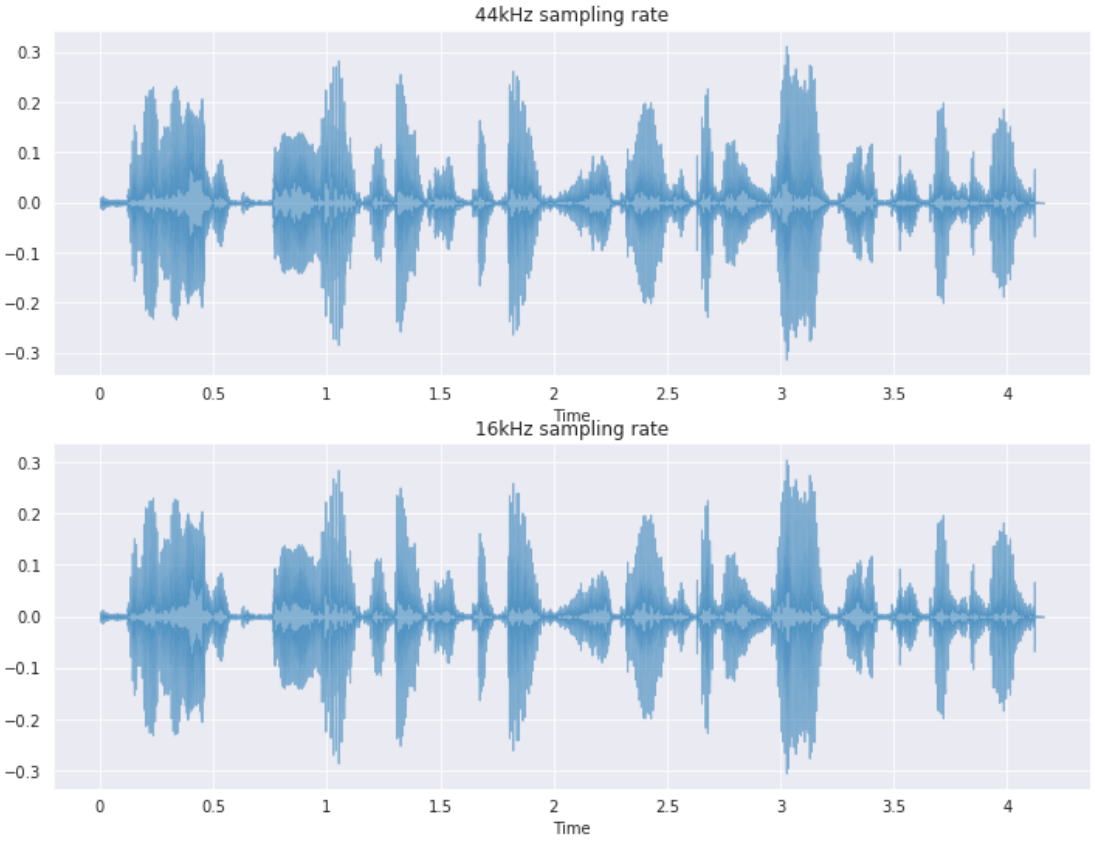


Figure 4.3: 44kHz and 16kHz sampling rates

Top image represents orignally recorded image at $44kHz$ while the bottom one represents $16kHz$ sampled image.

Scaling

After downsampling, next step is to squash all the varying values to a common scale of $[-1, +1]$ to ensure all samples have nearly equal weightage. Most supervised and unsupervised learning methods make decisions based on the data sets supplied to them, and algorithms frequently compute the distance metrics between data points in order to

Table 4.4: Audio Preprocessing Parameters

Parameter	**Value**
stftWindow	hamming
Window Size	0.04 seconds * $16kHz$
Hop Length	0.01 seconds * $16kHz$

draw more accurate conclusions from the data. Equation 4.2.1 shows scaling operation.

$$\begin{aligned} S_{\text{scaled}} &= \frac{S}{\max(|S|)} \\ S_{\text{scaled}} &= \text{scaled signal} \\ S &= \text{original audio signal sampled at} 16kHz \end{aligned} \tag{4.2.1}$$

Power normalization

Next signal processing operation is governed by Equation 4.2.2. The denominator is the RMS Value of given signal. Thus, can be deemed as a straightforward RMS normalisation.

It is in a sense, levelling signal's average but still permitting some peaks to be clipped (instead of being set to 1). In other words, standard division by the greatest absolute value of your signal will always ensure that sample values fall inside the interval $[-1, +1]$, whereas RMS normalisation does not. This technique is commonly employed for audio and voice processing.

$$\begin{aligned} S_n &= \frac{S}{\sqrt{\frac{\sum_{i=1}^{N} |S[i]^2|}{N}}} \\ S &= \text{original signal samples at} 16kHz \\ S[i] &= \text{ith sample} \end{aligned} \tag{4.2.2}$$

STFT

Figure 4.4 shows a sample signal. It is ensured that at least 4 stft windows sweeps are possible with the parameters defined in Table 4.4.

Output from STFT is a 2D matrix of shape $[freq.bins, frames]$, Equations 4.2.3, 4.2.4 show this calculation. As the audio corresponding to a video is loaded, *total samples* is already known while the f_{size} is the same as the $\frac{1}{FPS}$ seconds of video.

$$\begin{aligned} n\text{-freq} &= \frac{f_{\text{size}}}{2} + 1 \\ f_{\text{size}} &= \text{generally is same as no. of samples in a window length} \end{aligned} \tag{4.2.3}$$

In the later section when synchronization will be discussed, equations 4.2.3 and 4.2.4 will be used to synchronize the audio vectors with the video.

$$\begin{aligned} n\text{-frames} &= \frac{T_s - f_{\text{size}}}{H_s} + 1 \\ T_s &= \text{total samples in the audio at 16kHz} \\ H_s &= \text{number of samples between each FFT window} \end{aligned} \tag{4.2.4}$$

$$H_s = \frac{T_s - f_{size}}{n\text{-frames} - 1} \tag{4.2.5}$$

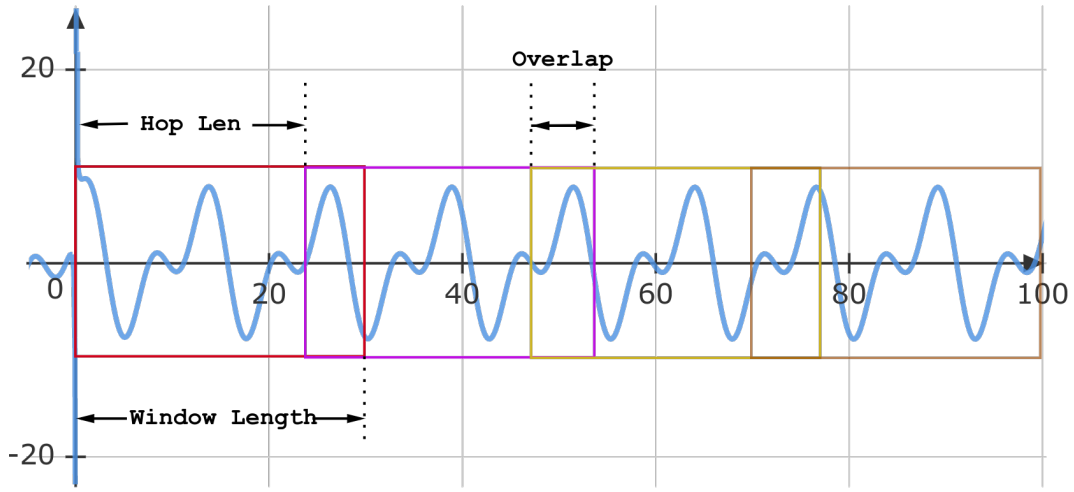


Figure 4.4: STFT strategy for audio features

FT vs STFT

Equations 4.2.7 and 4.2.6 define the STFT and DFT equations respectively. By applying the Fourier transform, we move into the frequency domain with *freq* on the x-axis and the magnitude itself is a function of frequency. However, by doing so, we lose information about time. Hence, Fourier transform gives frequency information averaging throughout the entire time interval of a signal.

When the frequency components of a signal change over time, the STFT offers time-localized frequency information. In STFT, there's a trade-off between time and frequency resolution. In other words, while a narrow-width window produces higher time-domain resolution, it produces poor frequency-domain resolution, and vice versa. It can be utilised to make representations that capture the signal's relative temporal and frequency information. Like Fourier transform, it also uses fixed basis functions, but it does so using fixed-size time-shifted window functions.

$$\widehat{x}(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-i2\pi n \frac{k}{N}} \tag{4.2.6}$$

$$\begin{aligned} S(m,k) &= \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n \frac{k}{N}} \\ m &= \text{freq. index} \\ k &= \text{time index} \\ x(n+mH) &= N_{ft}\text{points of x} \\ w(n) &= \text{window} \\ e^{-i2\pi n \frac{k}{N}} &= \text{DFT kernel} \end{aligned} \tag{4.2.7}$$

$$\begin{aligned} x(n) &= \text{ input signal at time } n \\ w(n) &= \text{ length } M \text{ window function (e.g., Hamming)} \\ X_m(\omega) &= \text{ DTFT of windowed data centered about time } mR \\ R &= \text{ hop size, in samples, between successive DTFTs.} \end{aligned}$$

4.3 Video preprocessing

With reference to Table 4.5, following are the video pre-processing steps involved. These steps are performed on each video:

As a result of pre-processing, corresponding to each video, 4 files are made:

- *mp4* video only file
- *wav* audio only file
- *png* ROI only file
- *npy* features only file

Pre-processing steps applied to each video are shown in Fig. 4.5. Hence against each frame in the dataset, we are extracting a latent meaningful information V_i. So we get a pre-processed matrix V of vectors against each video.

$$\begin{aligned} V_K &= \{V_1, V_2, V_3 \dots \dots V_N\} \\ K &= \{1, 2, 3 \dots \dots, B\} \\ B &= \text{total no. of videos} \\ N &= \text{ no. of frames in Kth video} \\ V_i &\in \mathbb{R}^{512} \end{aligned} \tag{4.3.1}$$

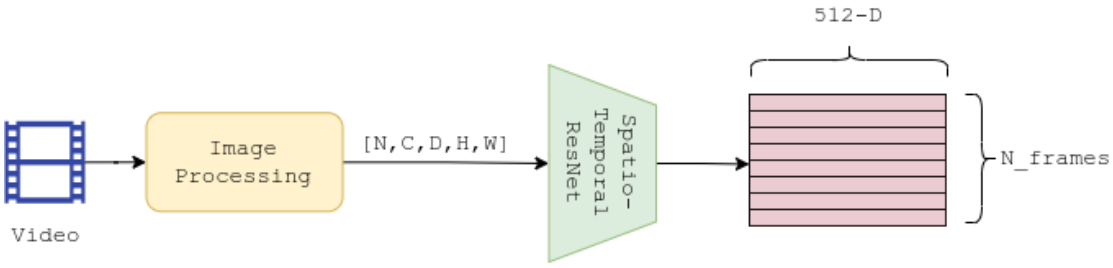


Figure 4.5: Preprocessing pipeline overview

As per mathematical formulation in equation 4.3.1 we get a $512D$ feature vector against each frame of the video. This represents the condensed representation of all the salient features.

Table 4.6 shows the model used for pre-processing each video. Before training, all the videos have been transformed into feature vectors using the said model. One salient

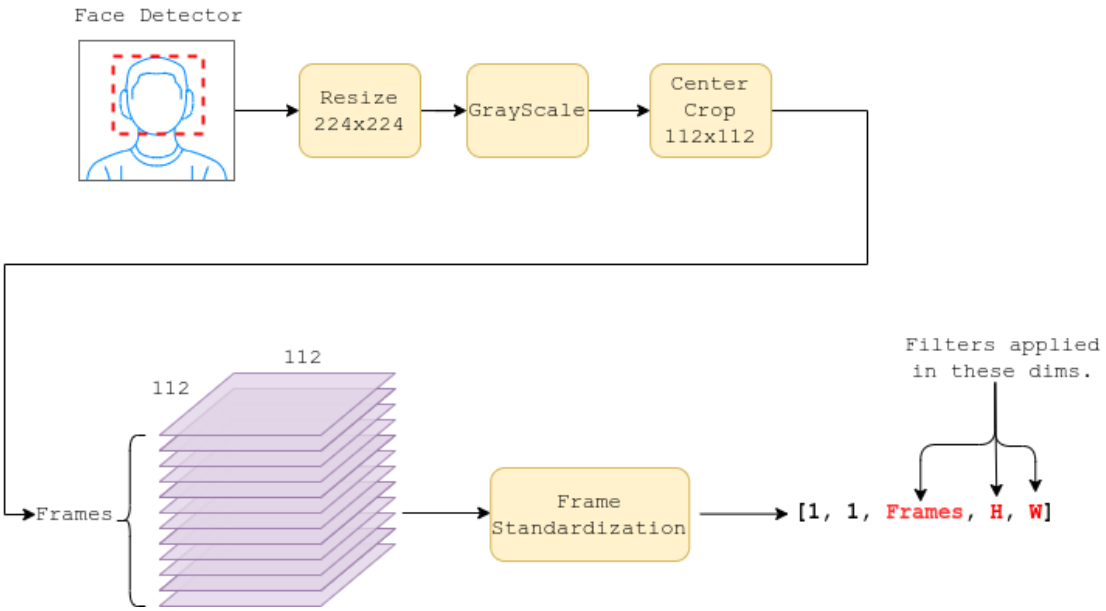


Figure 4.6: Preprocessing steps per video

Table 4.5: Video pre-processing parameters

Parameter	Value
ROI size	112
μ	0.4161
σ	0.1688

feature of this model is the $3D - convolution$ layer at the beginning. This layer takes as input a $5D$ tensor. The I/O from a $3D - convolution$ layer are defined in Equation 4.3.2.

$$\begin{aligned} \text{Input} &: (N, C_{in}, D_{in}, H_{in}, W_{in}) \; or \; (C_{in}, D_{in}, H_{in}, W_{in}) \\ \text{Output} &: (N, C_{\text{out}}, D_{\text{out}}, H_{\text{out}}, W_{\text{out}}) \; or \; (C_{\text{out}}, D_{\text{out}}, H_{\text{out}}, W_{\text{out}}) \end{aligned} \tag{4.3.2}$$

Table 4.6: Model Used for extracting features for each video

Type of Layer	**Kernel**	**Stride**	**Zero-Pad**	**Output Shape**
Conv3D	64 x (5 x 7 x 7)	(1, 2, 2)	(2, 3, 3)	[1, 64, NF, 56, 56]
BatchNorm3D	-	-	-	[1, 64, NF, 56, 56]
ReLU	-	-	-	[1, 64, NF, 56, 56]
MaxPool3D	(1 x 3 x 3)	(1, 2, 2)	(0, 1, 1)	[1, 64, NF, 28, 28]
Transpose	-	-	-	[1, NF, 64, 28, 28]
Reshape (needed by resnet)	-	-	-	[NF, 64, 28, 28]
Conv2D	64 x (3 x 3)	(1, 1)	(1, 1)	[NF, 64, 28, 28]
BatchNorm2D	-	-	-	[NF, 64, 28, 28]
Conv2D	64 x (3 x 3)	(1, 1)	(1, 1)	[NF, 64, 28, 28]
Conv2D	64 x (1 x 1)	(1, 1)	(0, 0)	[NF, 64, 28, 28]
BatchNorm2D	-	-	-	[NF, 64, 28, 28]
Conv2D	64 x (3 x 3)	(1, 1)	(1, 1)	[NF, 64, 28, 28]
BatchNorm2D	-	-	-	[NF, 64, 28, 28]
Conv2D	64 x (3 x 3)	(1, 1)	(1, 1)	[NF, 64, 28, 28]
BatchNorm2D	-	-	-	[NF, 64, 28, 28]
Conv2D	128 x (3 x 3)	(2, 2)	(1, 1)	[NF, 128, 14, 14]
BatchNorm2D	-	-	-	[NF, 128, 14, 14]
Conv2D	128 x (3 x 3)	(1, 1)	(1, 1)	[NF, 128, 14, 14]
Conv2D	128 x (1 x 1)	(2, 2)	(0, 0)	[NF, 128, 14, 14]
BatchNorm2D	-	-	-	[NF, 128, 14, 14]
Conv2D	128 x (3 x 3)	(1, 1)	(1, 1)	[NF, 128, 14, 14]
BatchNorm2D	-	-	-	[NF, 128, 14, 14]
Conv2D	128 x (3 x 3)	(1, 1)	(1, 1)	[NF, 128, 14, 14]

Table 4.6 continued from previous page

Type of Layer	Kernel	Stride	Zero-Pad	Output Shape
BatchNorm2D	-	-	-	[NF, 128, 14, 14]
Conv2D	256 x (3 x 3)	(2, 2)	(1, 1)	[NF, 256, 7, 7]
BatchNorm2D	-	-	-	[NF, 256, 7, 7]
Conv2D	256 x (3 x 3)	(1, 1)	(1, 1)	[NF, 256, 7, 7]
Conv2D	256 x (1 x 1)	(2, 2)	(0, 0)	[NF, 256, 7, 7]
BatchNorm2D	-	-	-	[NF, 256, 7, 7]
Conv2D	256 x (3 x 3)	(1, 1)	(1, 1)	[NF, 256, 7, 7]
BatchNorm2D	-	-	-	[NF, 256, 7, 7]
Conv2D	256 x (3 x 3)	(1, 1)	(1, 1)	[NF, 256, 7, 7]
BatchNorm2D	-	-	-	[NF, 256, 7, 7]
Conv2D	512 x (3 x 3)	(2, 2)	(1, 1)	[NF, 512, 4, 4]
BatchNorm2D	-	-	-	[NF, 512, 4, 4]
Conv2D	512 x (3 x 3)	(1, 1)	(1, 1)	[NF, 512, 4, 4]
Conv2D	512 x (1 x 1)	(2, 2)	(0, 0)	[NF, 512, 4, 4]
BatchNorm2D	-	-	-	[NF, 512, 4, 4]
Conv2D	512 x (3 x 3)	(1, 1)	(1, 1)	[NF, 512, 4, 4]
BatchNorm2D	-	-	-	[NF, 512, 4, 4]
Conv2D	512 x (3 x 3)	(1, 1)	(1, 1)	[NF, 512, 4, 4]
BatchNorm2D	-	-	-	[NF, 512, 4, 4]
AvgPool2D	(4 x 4)	(1, 1)	(0, 0)	[NF, 512, 1, 1]
Reshape	-	-	-	[NF, 512]

Table 4.6 shows model architecture. It is spatio-temporal-ResNet. With a filter width of 5 frames, the network applies 3D convolutions to the input image sequence, followed by a 2D ResNet that shrinks the spatial size with depth. After passing each video through the preprocessing pipeline with reference to Fig. 4.5, each video gets saved to a *.npy* file containing the matrix of shape $[NF, 512]$[1].

[1] no. of frames

Table 4.7: Hyper parameters of transformer encoder

Hiden Dim	Max. length for positional encoding	attention heads	no. of layers	Feed Forward Dim.	Dropout
512	2500	8	6	2048	0.1

4.4 Fusion Model details

As for the fusion of modalities standard implementation of the transformer encoder module as presented by [7]. None of the hyper-parameters have been changed. In the Figure 4.7 all of the three blocks i.e. video model, audio model and the fusion model implement the exact same transformer encoder implementation.

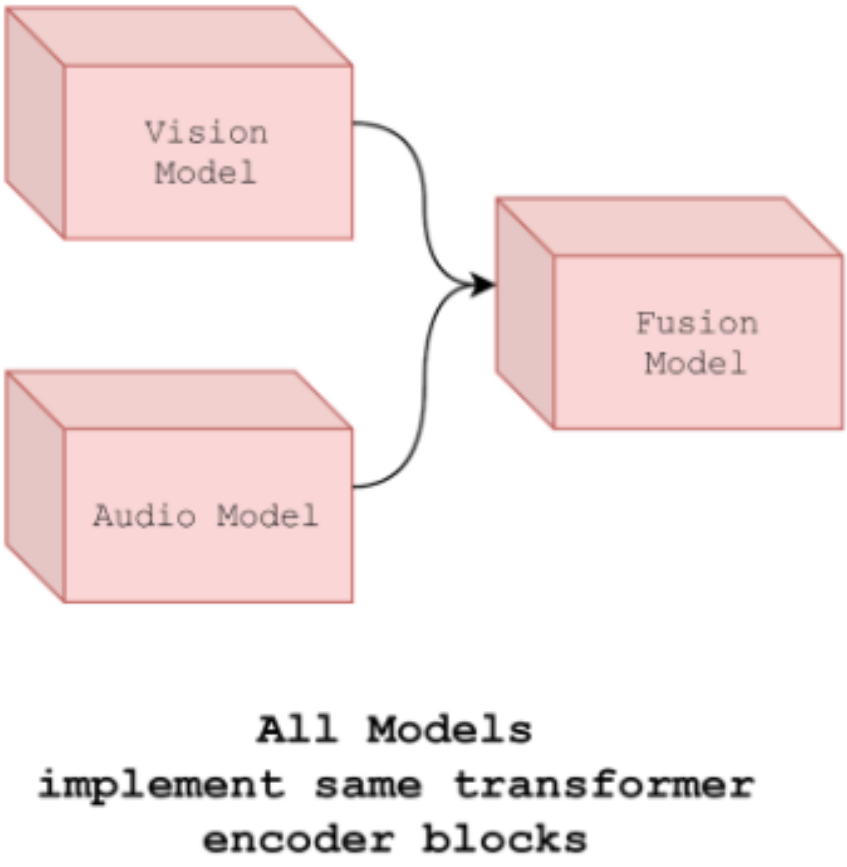


Figure 4.7: Fusion Model

The brief overview of transformer encoder is shown in Figure 4.8. Sine and Cosine embeddings have been used respectively for even index and odd index along the hidden dimension.

Table 4.7 shows all the hyper parameters of the transformer model used. It is evident that there is no change in the standard implementation and is same as proposed by [7].

Multiheaded self attention layers are like weightage learning layers where each component learns weightage w.r.t each other token in the sequence. These weightages are then summed up. So it acts like a self aligning layer where tokens learn to align with themseleves whereas the postional encodings induce absolute positional information to them so as to make them learn temporal dependencies. All the sequences are 0 padded to the max length and during training, the zero padding masks are provided to ignore results aat those locaitons.

4.5 Training pipeline

Figure 4.9 shows the training pipeline of the system. It starts from pre-processing of the audio and visual features. For each video there will be a 2D matrix for audio features and a 2D matrix for video features. The frame dimension has been used as the sequence dimension. So essentially teh input to transformer and output will be a 3D tensor of shape $[N, T, D]$ wher N is the batch size, T denote frames in that video and D represents hidden dimension.

Training details have been shown in Figure 4.10. A learning rate of $1e-4$ has been used for training using GTX-1080 Ti with Adam optimizer. The whole training process took almost 7 days while the model has been trained for 40 epochs with batch size of 8. Such large training times are not uncommon for multi-modality based models. This doesnot include the pre-processing time which alone takes almost 3 days to transform all the samples on disk into useful features which then need to be loaded while training.

4.6 Inference Pipeline

Figure 4.14 shows the typical inference pipeline, here are the steps involved at inference time:

- User will be asked to speak a provided secret phrase
- User records the video while uttering, from which video and audio information have been extracted.

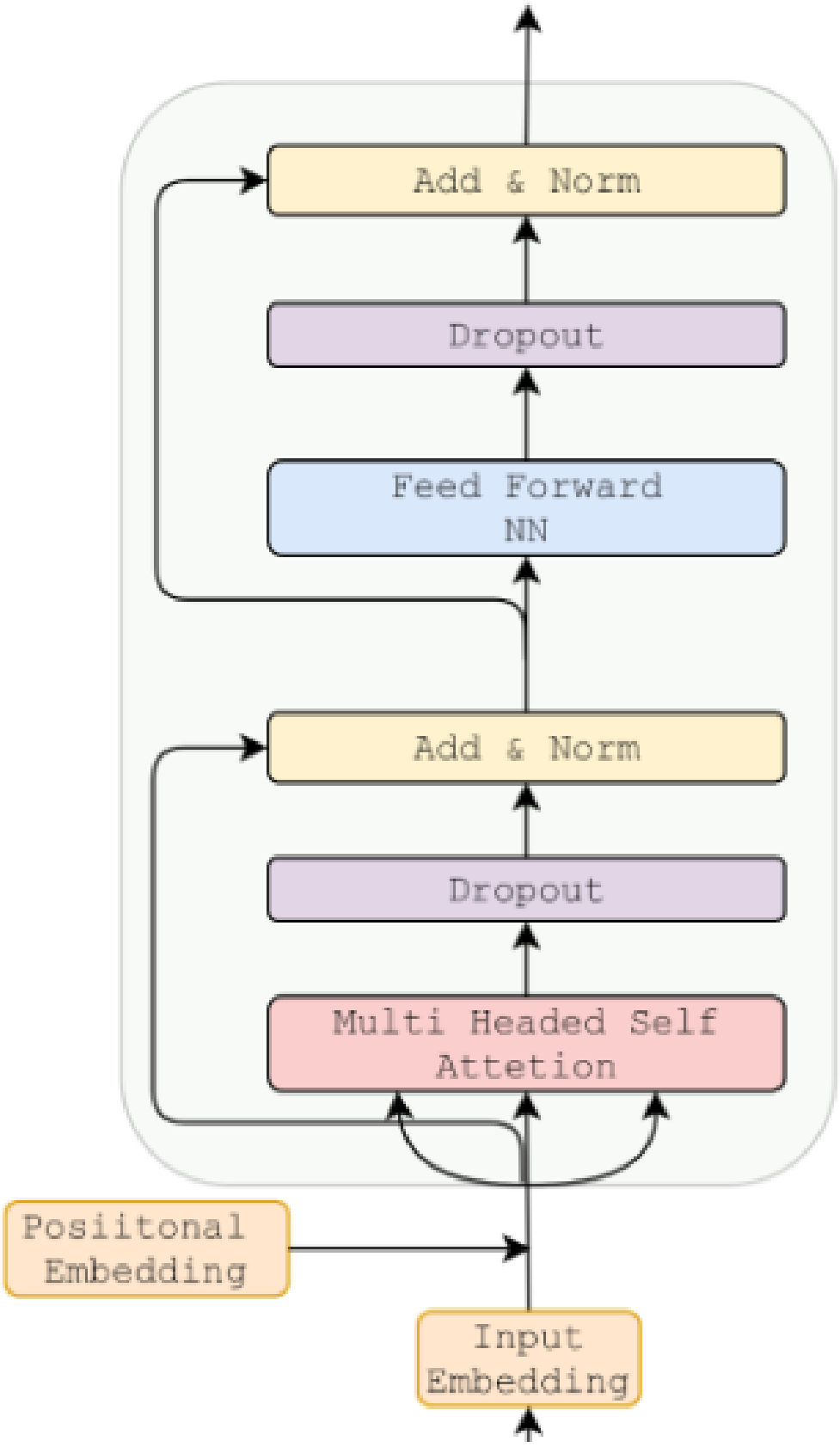


Figure 4.8: Transformer Encoder

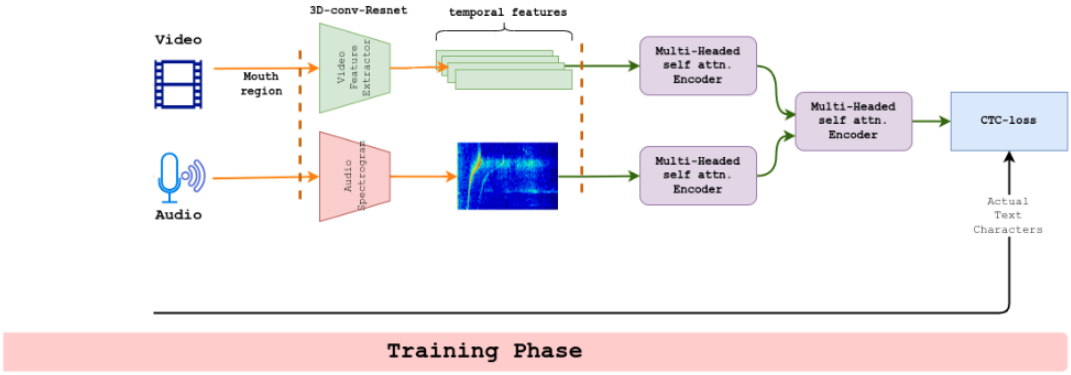


Figure 4.9: Training pipeline for the system

- The audio-visual model alongwith decoder will transform the utterances into a sequence of characters predicted against each temporal step
- Both the utterance and the actual secret key word phrase are transformed in arbitrary $384D$ space as unique unit vectors.
- In that arbitrary space, cosine-similarity metric is computed which if greater than a threshold.

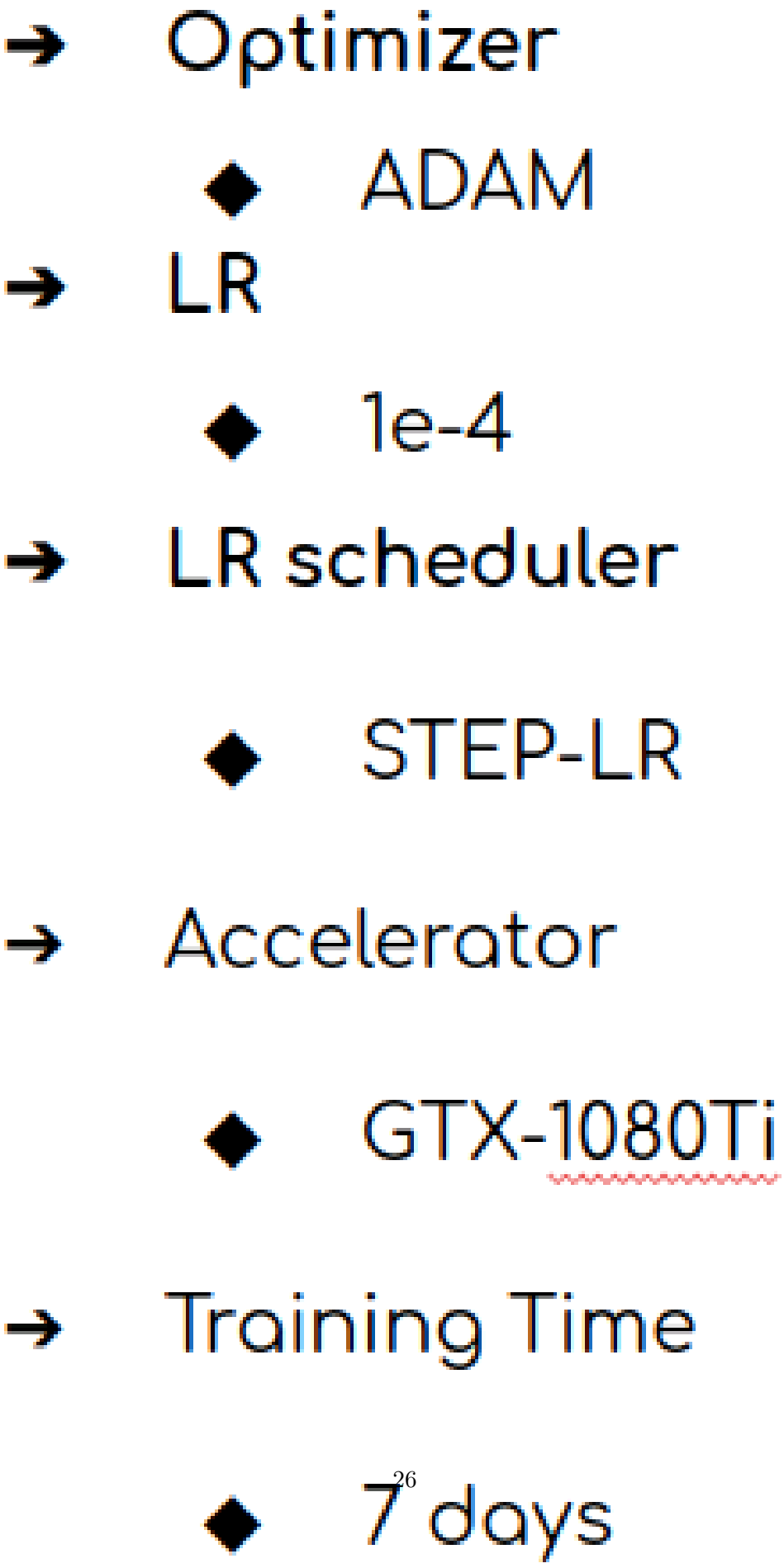
4.7 Loss function

Figure 4.12 shows ctc loss computation process.

- all the paths which can condense to the target characters are listed
- individual losses are calculated by multiplying all values at specific nodes
- in the end all get summed up
- loss is then negative log of summed value

4.8 Decoding

At decoding time, eachn temporal segment is passed via softmax layer and the max. value of discrete probability is chosen, which can be then mapped to hash of characters.


Optimizer
ADAM
LR
1e-4
LR scheduler
STEP-LR
Accelerator
GTX-1080Ti
Training Time
7 days

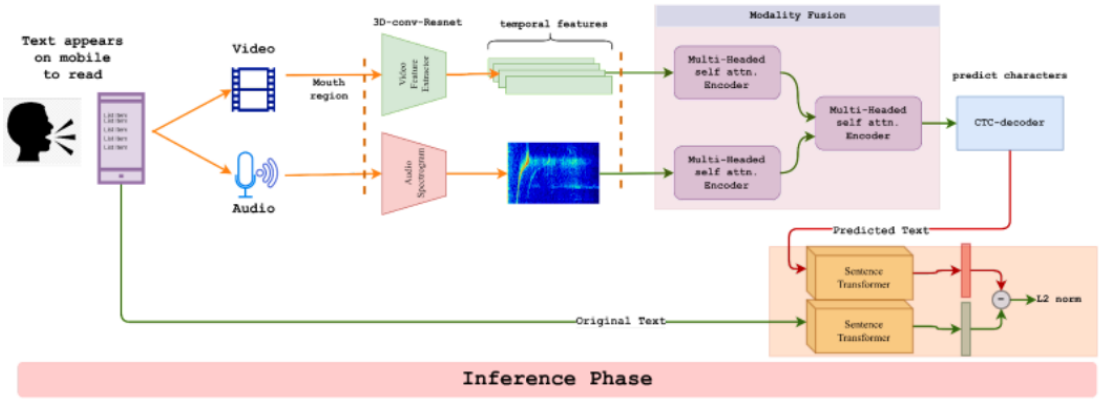


Figure 4.11: Inference pipeline

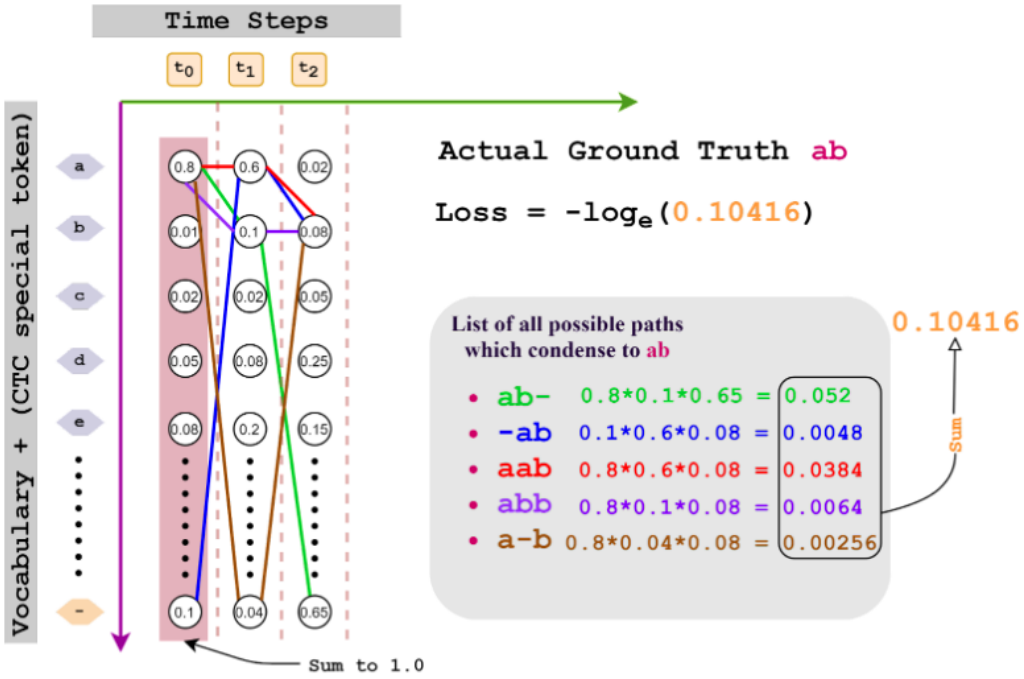


Figure 4.12: CTC loss computation

After all temporal steps have been processed, ctc-specific condensation operation is performed to condense the temporal predictions into reduced characters.

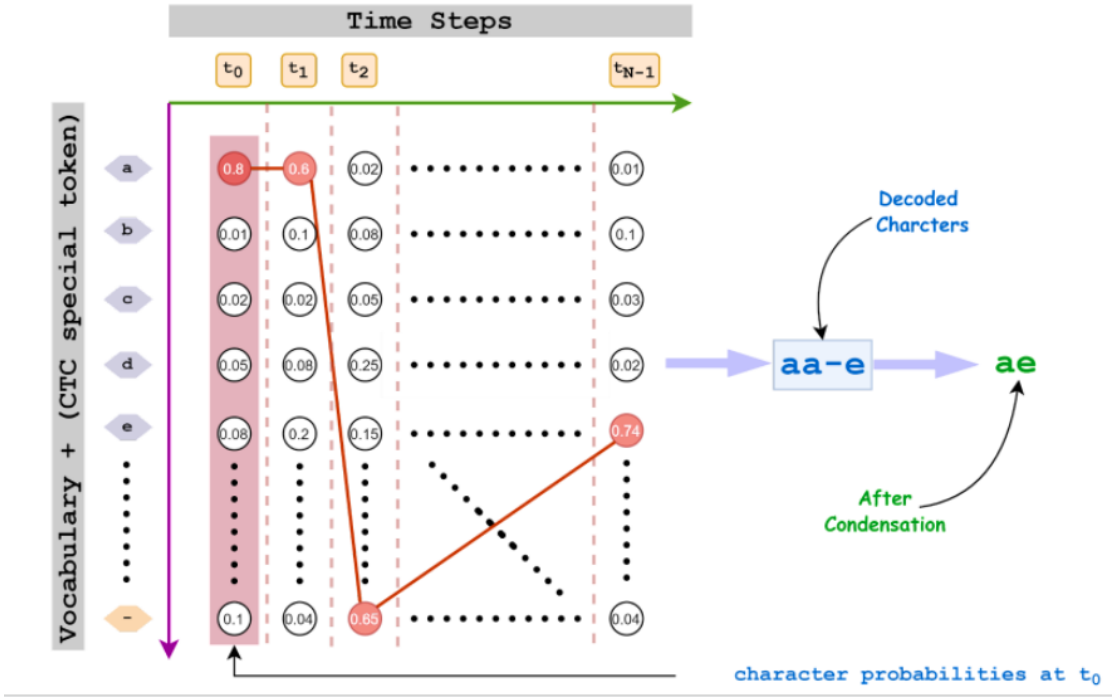


Figure 4.13: Decoding using CTC

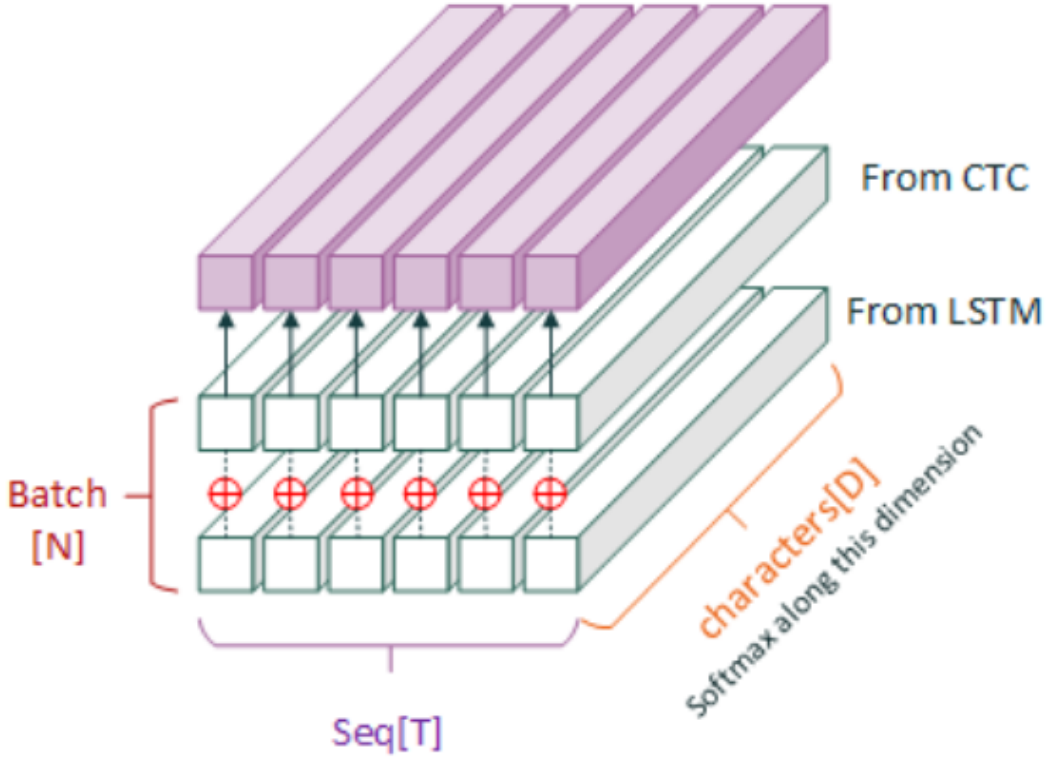


Figure 4.14: Weighted decoding with language model and CTC decoder

Chapter 5

Results

5.1 Evaluation metrics

To evaluate the accuracy of a biometric system, i.e., to quantify its biometric performance, numerous authentic and fraudulent efforts are made with the system, and all similarity scores are recorded. By applying a variable score threshold to the cosine similarity, it is possible to calculate combinations of FRR and FAR (or FNMR and FMR).

5.1.1 FNMR or FRR

False non match refers to a case where the user has uttered correct statement but the algorithm rejects user. It can also be defined more precisely in terms of biometrics as ratio of impostor attempts falsely claiming to match the blueprint of another object.

5.1.2 FMR or FAR

False match refers to a case where the user has uttered incorrect statement but the algorithm accepts user. It can also be defined more precisely in terms of biometrics as ratio of impostor attempts falsely claiming to match the blueprint of another object.

5.1.3 EER

Equal Error Rate refers to the point where both FNMR and FMR are equal. It is defined as a sweet spot for a system but

5.2 Scores Distributions

Let x_i and $\widehat{x_i}$ be real and inferred text respectively, a non-linear transformation $g(x_i, \widehat{x_i})$ has been applied to transform each into vectors V_i and $\widehat{V_i}$ where;

$$V_i \in \mathbb{R} \tag{5.2.1}$$

Fig. 5.1 shows that contrastive learning model's performance is great on test dataset of LRS2 as it pulls apart both of the histograms depicting the representation power of the algorithm. It shows very less area of mutual overlap which is depictive of its good representation power.

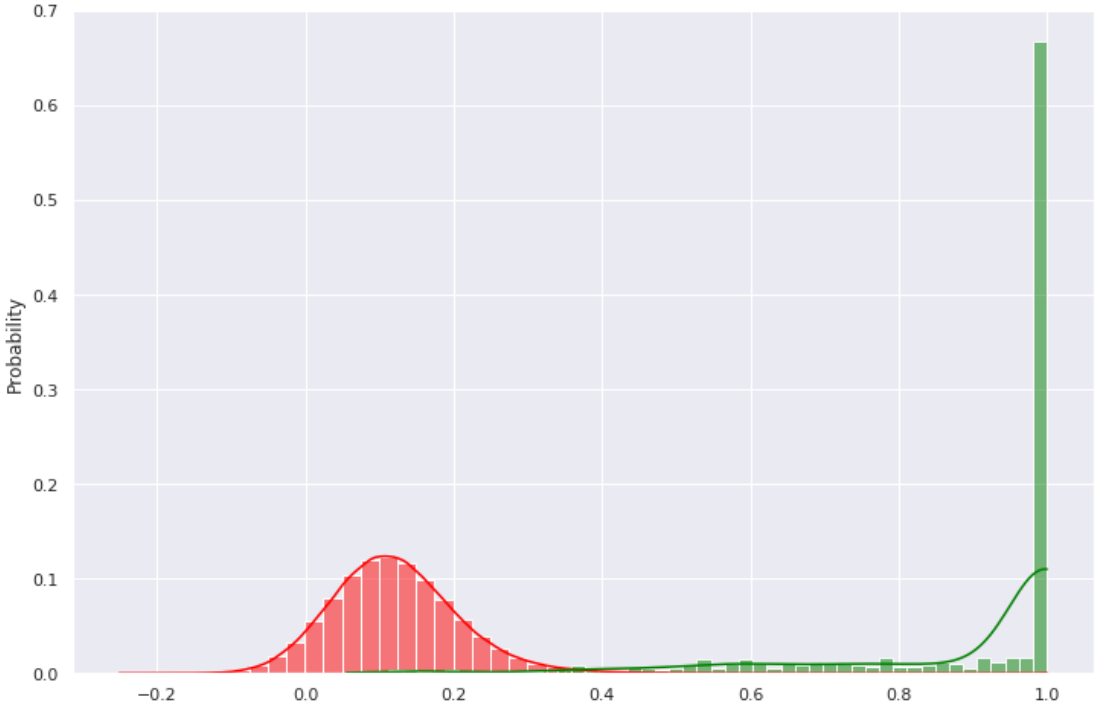


Figure 5.1: Histograms of the real and impostor scores

Training curves for the training subset of the LRS2 dataset have been presented here in Figure 5.4. Training details are explained already in the training section.

Figure **??** shows a contour plot of the real (similar) and impostor (different) scores. Similar scores means system matches the uttered secret phrase correctly. Impostor means false attempt which means that all negative pairs of the inferred phrase vector

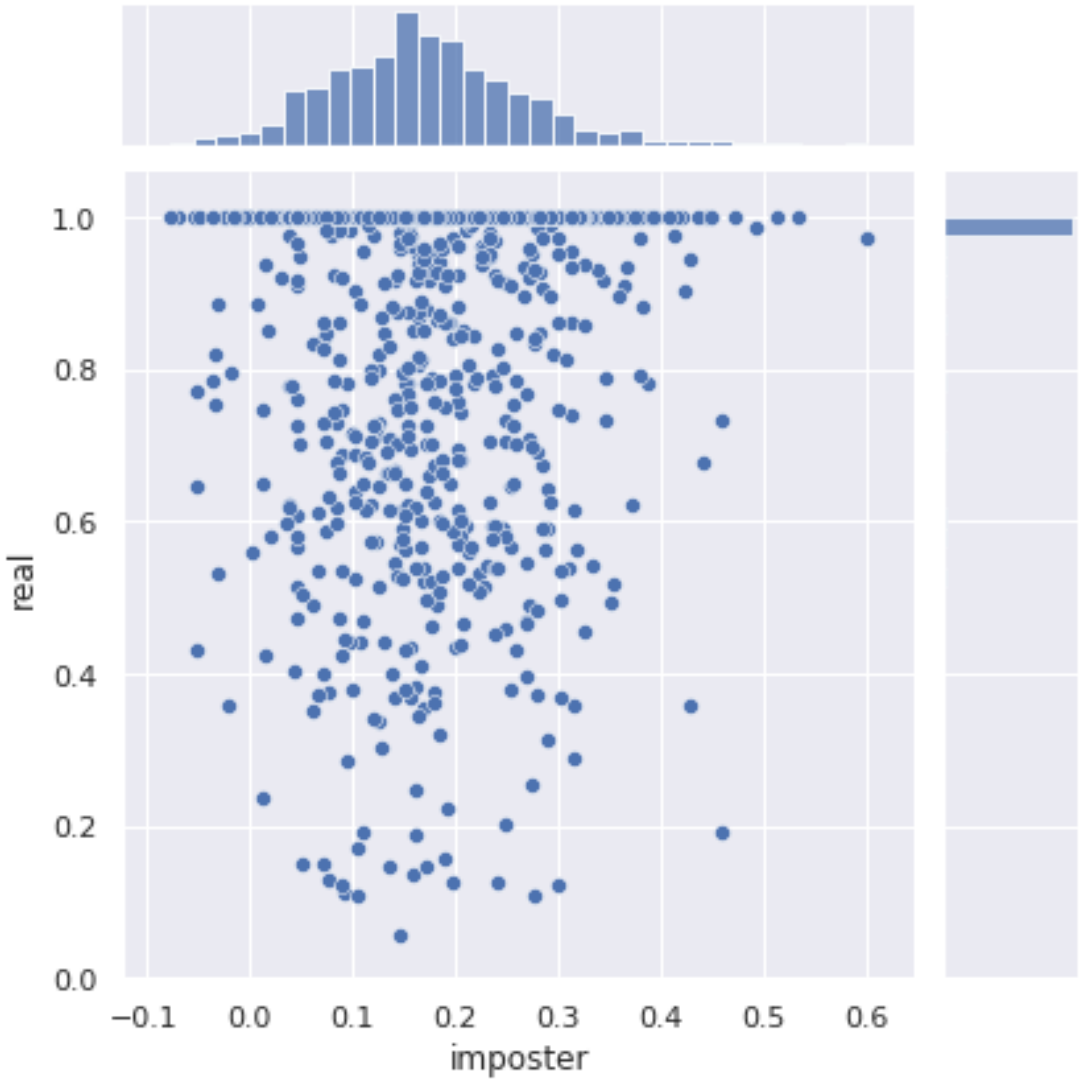


Figure 5.2: Relative Histograms and Scatter plots of scores

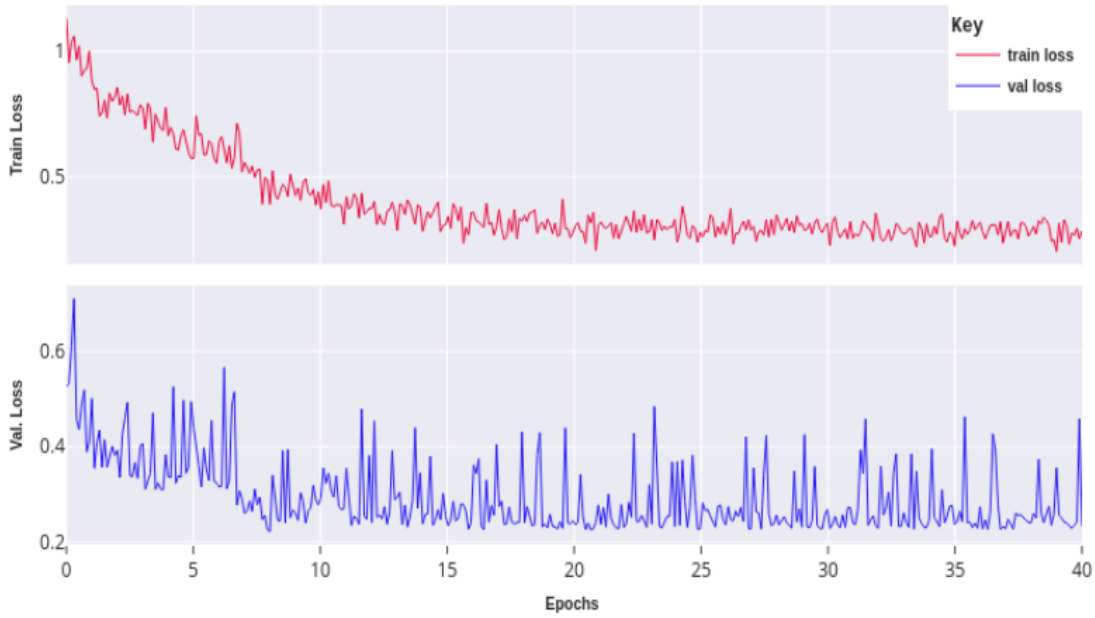


Figure 5.3: training and validation loss curves

against all other phrases in the dataset.

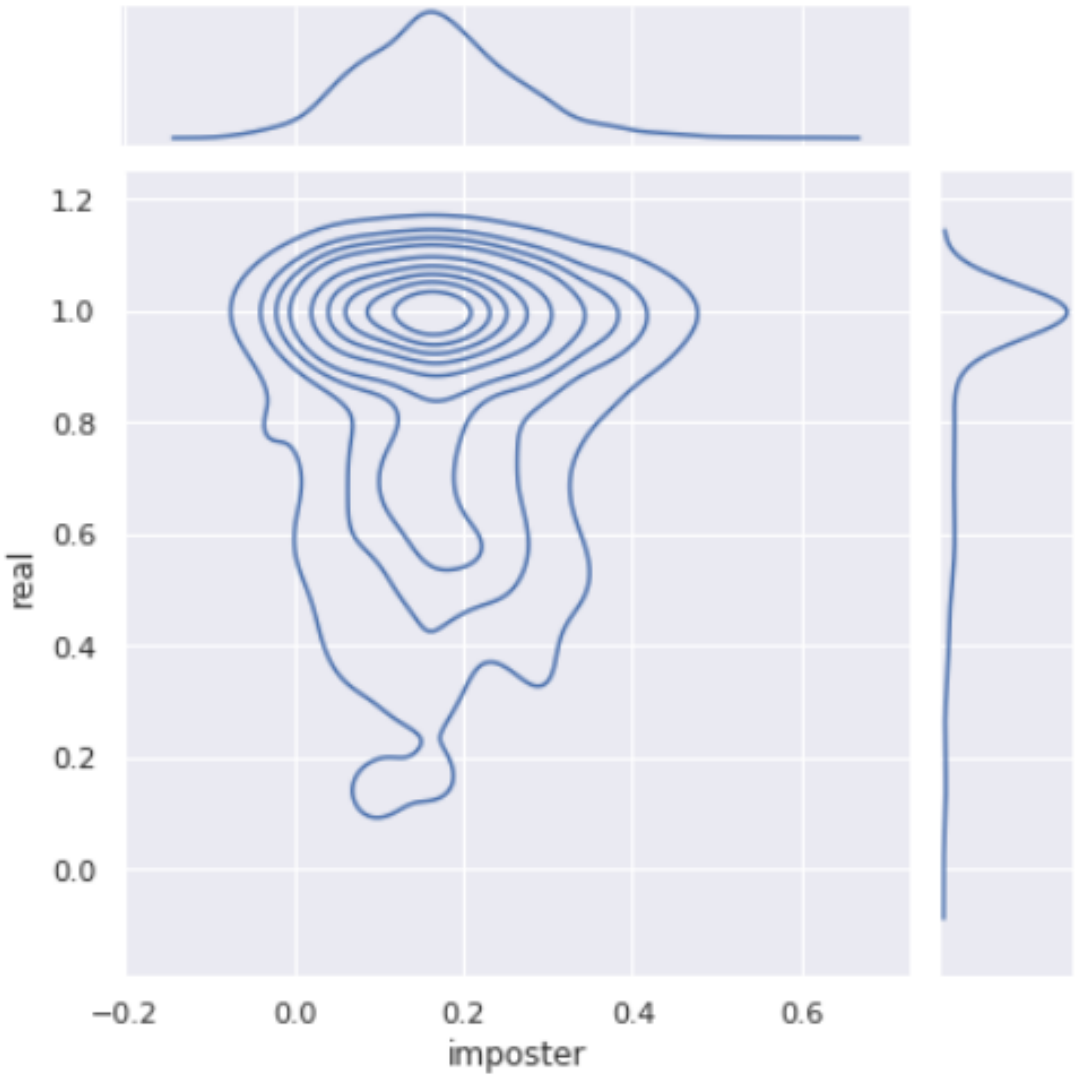


Figure 5.4: cnto

Chapter 6

Conclusion

A mid-fusion model was trained using CTC loss and afterwards uttererd phrase was transformed into vector embedding using sentence-transformer which has been pre-trained for text similarity task on wikipedia dataset. Actual secret key phrase has also been transformed and in that arbitrary space, cosine similarity metric defines the authentication status.

After experimentation it is evident that phrases with greater sequence length the system is more accurate.

Chapter 7

Future Work

Although the first of their kind, LRS2 and LRS3 datasets lack the multilingual information about the data. These datasets consist solely of English language utterances. To examine the impact of multilingual queries on the system, a multilingual dataset analysis must be conducted. In addition, the effect of accent on the authentication system must be determined.

The training dataset solely contains british accent speech so it might have a hard time to authenticate any other accent despite language being english.

Such authentication systems have a great deal of room for improvement if more data from the real world are collected to examine the effects of differences in accent and lightning conditions.

References

[1] Maycel-Isaac Faraj and Josef Bigun. "Audio–visual person authentication using lip-motion from orientation maps". en. In: *Pattern Recognition Letters*. Advances on Pattern recognition for speech and audio processing 28.11 (2007), 1368–1382. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2007.02.017.

[2] Amitava Das, Ohil K. Manyam, and Makarand Tapaswi. "Audio-Visual Person Authentication with Multiple Visualized-Speech Features and Multiple Face Profiles". In: *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*. 2008, 39–46. DOI: 10.1109/ICVGIP.2008.106.

[3] Amitava Das and Vaibhav Bedia. "Audio-visual person authentication with multiple face-profiles and compressed-feature-dynamics signatures of spoken passwords". In: *2009 IEEE International Workshop on Multimedia Signal Processing*. 2009, 1–6. DOI: 10.1109/MMSP.2009.5293273.

[4] Kai Li. *Identity Authentication based on Audio Visual Biometrics : A Survey*. en. 2013. URL: https://www.semanticscholar.org/paper/Identity-Authentication-based-on-Audio-Visual-\%3A-A-Li/19c64faa7f9d8e007a1d6aa187987d6b71df615f.

[5] Kuniaki Noda et al. "Audio-visual speech recognition using deep learning". en. In: *Applied Intelligence* 42.4 (2015), 722–737. ISSN: 1573-7497. DOI: 10.1007/s10489-014-0629-7.

[6] Amirsina Torfi et al. "3D Convolutional Neural Networks for Cross Audio-Visual Matching Recognition". In: *IEEE Access* 5 (2017), 22081–22091. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2761539.

[7] Ashish Vaswani et al. "Attention Is All You Need". en. In: (June 2017). DOI: 10.48550/arXiv.1706.03762. URL: https://arxiv.org/abs/1706.03762v5 (visited on 06/28/2022).

[8] George Sterpu, Christian Saam, and Naomi Harte. "Attention-based Audio-Visual Fusion for Robust Automatic Speech Recognition". In: *Proceedings of the 20th ACM International Conference on Multimodal Interaction* (2018), 111–115. DOI: 10.1145/3242969.3243014.

[9] Triantafyllos Afouras et al. "Deep Audio-Visual Speech Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), 1–1. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2018.2889052.

[10] Mikhail Pautov et al. "On adversarial patches: real-world attack on ArcFace-100 face recognition system". en. In: (Oct. 2019). DOI: 10.1109/SIBIRCON48586.2019.8958134. URL: https://arxiv.org/abs/1910.07067v3 (visited on 06/28/2022).

[11] Brais Martinez et al. "Lipreading using Temporal Convolutional Networks". In: *arXiv:2001.08702 [cs, eess]* (2020). URL: http://arxiv.org/abs/2001.08702.

[12] Bo Xu et al. "Discriminative Multi-modality Speech Recognition". In: *arXiv:2005.05592 [cs, eess]* (2020). URL: http://arxiv.org/abs/2005.05592.

[13] Wentao Yu, Steffen Zeiler, and Dorothea Kolossa. "Multimodal Integration for Large-Vocabulary Audio-Visual Speech Recognition". In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021, 341–345. DOI: 10.23919/Eusipco47968.2020.9287841.