Luxor University
Faculty of Computers and Information
CS Department

Text of silence

Title:

Text of silence is for generating text from lip movement for aphonia patient, Hearing-impaired people and in noisy environments

Under Supervisor

Dr. Mohammed Abdel Hameed

Students

1. Mamdoh Marof Abdulrazak
2. Mohammed Awadallah Bakir
3. Abdelrahman Mohammed Abdullah
4. Mahmoud Mohammed Fouad
5. Amr Ftouh Abbas
6. Abdelrahman Mustafa Mahmoud

ABSTRACT

Lip reading is a visual way of “listening” to someone. This is done by looking at the speaker's face to follow their speech patterns in order to recognize what is being said. it is used to understand or interpret speech without hearing it, a technique especially mastered by people with hearing difficulties. The ability to lip-read enables a person with a hearing impairment to communicate with others and to engage in social activities, which otherwise would be difficult. Recent advances in the fields of computer vision, pattern recognition, and signal processing have led to a growing interest in automating this challenging task of lip reading. Indeed, automating the human ability to lip-read, a process referred to as visual speech recognition, could open the door for other novel applications.

it is also known as audio-visual recognition, has been considered as a solution for speech recognition tasks, especially when the audio is corrupted or when the conversation happened in noisy environments. It can also be an extremely helpful tool for people who are hearing-impaired to communicate through video calls. This task, however, is challenging, due to factors such as the variances in the inputs (facial features, skin colors, speaking speeds, etc.) and the one-to-many relationships between viseme and phoneme.

Lip-reading technology mainly includes face detection, lip localization, feature extraction, training the classifier through the corpus, and finally recognition of the word/sentence through lip movement.

Contents List

List of Figures

Chapter 1

Introduction

1.1 Abbreviations

Keyword	**Meaning**
AI	Artificial intelligence
ML	Machine learning
DL	Deep learning
ANN	Artificial neural network
RNN	Recurrent neural network
CNN	Convolutional neural network
ADAM	adaptive moment estimation

Table 1. Abbreviation

1.2 Introduction

To know how our project works we must first discuss what Artificial intelligence (AI) is and its types.

Artificial intelligence (AI):

AI is a field of computer science that studies how machines can imitate the intelligence of their human counterparts. Over the last decade, definitions of the term have become quite loose and refer to just about any computerized or automated function. However, the difference between an AI system and traditional software packages is the ability to make informed judgments and decisions by responding to patterns in data.

Applications of Artificial intelligence:

1. Smart homes, cities and infrastructure
2. Artificial intelligence against Covid-19
3. Machine translations
4. Transport
5. Health

Machine learning (ML):

Machine learning (ML) is a subset of artificial intelligence, which build a mathematical model based on sample data, known as “training data,” in order to make predictions or decisions without being explicitly programmed to perform the task. In machine learning, neural networks, support vector machines, and evolutionary computation, we are usually given a training set and a test set. By the training set, it will mean the union of the labeled set and the unlabeled set of examples available to machine learners. In comparison, test set consists of examples never seen before.

Deep learning (DL):

Deep Learning is an emerging field of Machine learning; that is, it is a subset of Machine Learning where learning happens from past examples or experiences with the help of ‘Artificial Neural Networks’. Deep Learning uses deep neural networks, where the word ‘deep’ signifies the presence of more than 1 or 2 hidden layers apart from the input and output layer.

Deep learning algorithms:

1. Artificial neural network (ANN):

A series of algorithms that are trying to mimic the human brain and find the relationship between the sets of data.

2. Recurrent neural network (RNN):

Recurrent neural networks are designed to interpret temporal or sequential information. These networks use other data points in a sequence to make better predictions. They do this by taking in input and reusing the activations of previous nodes or later nodes in the sequence to influence the output. RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer.

3. Convolutional neural network (CNN):

A class of deep neural networks that extracts features from images, given as input, to perform specific tasks such as image classification, face recognition and semantic image system CNN has one or more convolution layers for simple feature extraction, which execute convolution operation (i.e. multiplication of a set of weights with input) while retaining the critical features (spatial and temporal information) without human supervision.

4. Adam optimizer:

The Adam optimization algorithm is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing.

And is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

What is lip reading?

The use of lip-reading has been documented since the 16th century and hearing-impaired people often use lip-reading as an adjunct to understanding fluent speech. When it comes to automating the process, there are many challenges compared to conventional audio recognition.[1]

Automated lip-reading system refers to the systems which utilizes the visual information of the movement of the speech articulators such as the lips, teeth and somehow tongue of the speaker. The advantages are that such a system is not sensitive to ambient noise and change in acoustic conditions, does not require the user to make a sound, and provides the user with a natural feel of speech and dexterity of the mouth.

lip-reading plays a crucial role in human communication and speech understanding, as highlighted by the McGurk effect (McGurk & MacDonald, 1976)[2], where one phoneme's audio dubbed on top of a video of someone speaking a different phoneme results in a third phoneme being perceived. Lip-reading is a notoriously difficult task for humans, especially in the absence of context1. Most lip-reading actuations, besides the lips and sometimes tongue and teeth, are latent and difficult to disambiguate without context (Fisher, 1968; Woodward & Barber, 1960)[3]. For example, Fisher (1968) gives 5 categories of visual phonemes (called visemes), out of a list of 23 initial consonant phonemes, that are commonly confused by people when viewing a speaker's mouth. Many of these were asymmetrically confused, and observations were similar for final consonant phonemes. Consequently, human lip-reading performance is poor. Hearing-impaired people achieve an accuracy of only 17±12% even for a limited subset of 30 monosyllabic words and 21±11% for 30 compound words (Easton & Basala, 1982). An important goal, therefore, is to automate lip-reading. Machine lip readers have enormous practical potential, with applications in improved hearing aids, silent dictation in public spaces, security, and speech recognition in noisy environments, biometric identification, and silent-movie processing. Machine lip-reading is difficult because it requires extracting spatiotemporal features from the video (since both position and motion are important). Recent deep learning approaches attempt to extract those features end-to-end. [4]

Lip reading the ability to recognize what is being said from visual information alone, is an impressive skill, and very challenging for a novice. It is inherently ambiguous at the word level due to homophones – different characters that produce exactly the same lip sequence (e.g. 'p' and 'b'). However, such ambiguities can be resolved to an extent using the context of neighboring words in a sentence, and/or a language model. A machine that can lip read opens up a host of applications: '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 the performance of automated speech recognition in general. That such automation is now possible is due to two developments that are well known across computer vision tasks: the use of deep neural network models; and, the availability of a large-scale dataset for training. In this case the model is based on the recent sequence-to sequence (encoder-decoder with attention) translator architectures that have been developed for speech recognition and machine translation.[5]

1.3 The Essential Question:

The essential question with relevance to the Vision and Mission of the Faculty of Computers and information at Luxor University would be: How can we, using our education, help the hearing-impaired and aphonia people nationally, regionally, and internationally to better serve individuals, society, and the environment.

1.4 Motivation and Justification:

We are encouraged to work on this project as it has the potential to help over ten million English-speaking deaf and hearing-impaired people all over the world 6

Based on our research, currently there is no product that could help solve the problem of lip reading at a reasonable cost, there is currently a solution being developed and is planned for release within the next year by a company called Liopa in the UK.

Seeing the lack of research in the field motivated us to explore how we can utilize our knowledge to implement a solution that could contribute to aiding the hearing-impaired.

The project is also relevant to our interests as it mainly involves two fields of computer engineering like Computer Vision, Neural Networks and Machine

Learning in general and Neural Networks in specific

1.5 Related work

Research on lip reading (a.k.a. visual speech recognition) has a long history Many of the existing works in this field have followed similar pipelines which first extract spatiotemporal features around the lips (either motion-based, geometric-feature based or both), and then align these features with respect to a canonical template.

Title	Author	Year	abstract
Improved speaker independent lip-reading using speaker adaptive training and deep neural networks	- Ibrahim Almajai	2016	Furthermore, we show that error rates can be even further reduced by the additional use of Deep Neural Networks (DNN). We also find that there is no need to map phonemes to visemes for context-dependent visual speech transcription.[7]
Out of Time: Automated Lip Sync in the Wild	- Joon Son Chung - Andrew Zisserman	2017	They apply the network to two further tasks: active speaker detection and lip-reading. On both tasks, we set a new state-of-the-art on standard benchmark datasets.8
Lip Reading Sentences in the Wild	- Joon Son Chung - Andrew Senior - Oriol Vinyals - Andrew Zisserman	2017	The goal of this work is to recognize phrases and sentences being spoken by a talking face, with or without the audio. Unlike previous works that have focused on recognizing a limited number of words or phrases, we tackle lip reading as an open-world problem - unconstrained natural language sentences, and in the wild videos. Our key contributions are:

			1.a Watch, Listen, Attend and Spell' (WLAS) network that learns to transcribe videos of mouth motion to characters; 2.a curriculum learning strategy to accelerate training and to reduce overfitting; 3.a `Lip Reading Sentences' (LRS) dataset for visual speech recognition9
Morse code application	- Morse Samsung	2019	where blind people could use their mobile phones with Morse code taping instead of ordinary texting.10
Combining Residual Networks with LSTMs for Lipreading	- Themos Stafylakis, - Georgios Tzimiropoulos	2017	We propose an end-to-end deep learning architecture for word-level visual speech recognition. The system is a combination of spatiotemporal convolutional, residual and bidirectional Long Short-Term Memory networks. We train and evaluate it on the Lipreading In-The-Wild benchmark.11
Learning to lip read words by watching videos	- Joon Son Chung - Andrew Zisserman.	2018	Our aim is to recognize the words being spoken by a talking face, given only the video but not the audio. Existing works in this area have focused on trying to recognize a small number of utterances in controlled environments (e.g. digits and alphabets), partially due to the shortage of suitable datasets.12
Deep Learning of Mouth Shapes for Sign Language	- Oscar Koller - Hermann Ney - Richard Bowden	2015	This paper deals with robust modelling of mouth shapes in the context of sign language recognition using deep convolutional neural networks. Sign language mouth shapes are difficult to annotate and thus hardly any

			publicly available annotations exist. As such, this work exploits related information sources as weak supervision. Humans mainly look at the face during sign language communication, where mouth shapes play an important role and constitute natural patterns with large variability.

Chapter 2

Domain Analysis and Techniques

2.1 Domain Analysis

Seeing the lack of research in the field motivated us to explore how we can utilize our knowledge to implement a solution that could contribute to aiding the hearing-impaired.

The project is also relevant to our interests as it mainly involves two fields of computer engineering like Computer Vision, Neural Networks and Machine Learning in general and Neural Networks in specific.

Text of Silence is a lip-reading mobile application that aims to help the hearing-impaired communicate and interact better with their surroundings.

It consists of a lip-extracting module-using image processing, a learning module using machine learning.

Description of Products and Services: The chief goal is to build a system that captures human face, detects the mouth position and traces its movements regarding lips positions and movements in order to predict the words said or willing to be said by the user.

Mobile camera captures the user’s video, passing it to a back-end where all the lips tracing will be processed by an already trained module to detect the desired word or group of words.

The actual text then propagates to the user’s application interface.

Many new initiatives and users can rely on this project.

Hearing-impaired users can rely on this application to communicate with those who cannot understand sign language.

Moreover, some organizations can use this functionality in order to detect words out of videos with no sound like football games.

Technology Consideration: Considering the technological issues that might happen during the live-phase of the prototype testing and subsequently commercial operation, problems may appear with the mobile camera quality regarding resolution, night sight mode, and similar environmental conditions.

Therefore, it might be prudent to operate on devices with medium to high camera capabilities.

Another issue that might arise is the internet speed as well as back-end server speed; since the internet speed in some countries is limited below, certain boundaries thus the application version operated in low-speed internet countries will have the most restricted video time.

this project will facilitate the communication between people as well as other field’s requirements like detecting any unethical words used in a sports event.

2.2 Techniques

1. Convolutional Neural Networks (CNN)

A CNN is one of the variants of neural networks used heavily in the field of Computer Vision. It derives its name from the type of hidden layers it consists of. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers, and normalization layers. Here it simply means that instead of using the normal activation functions, convolution and pooling functions are used as activation functions [13].Figure 1 illustrates a Convolutional Neural Networks.

We use it to extract features from images that we infer from video

“used to transform images of the lips region to its vector representation.”

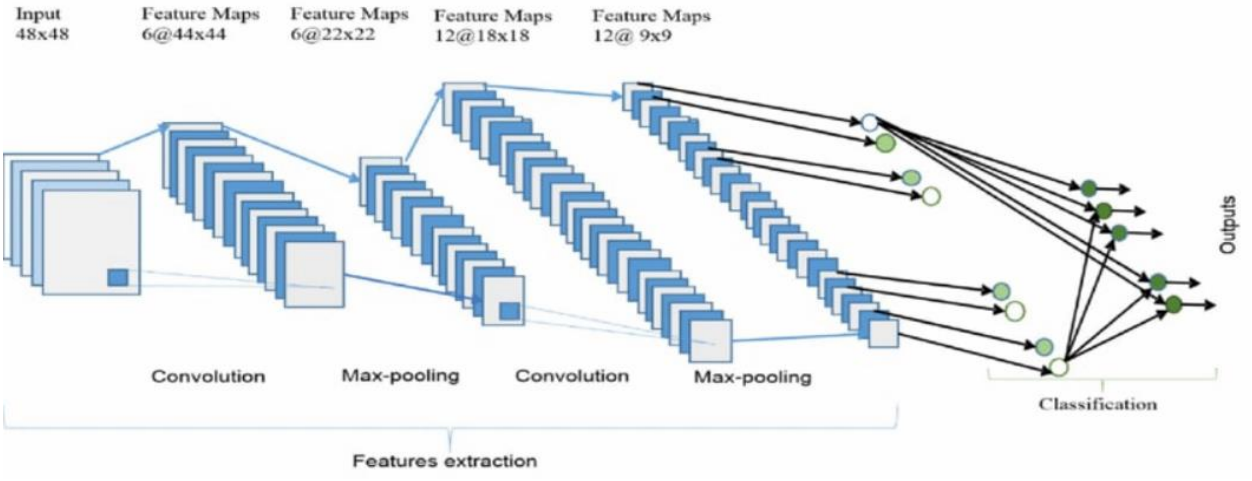


Figure 1. Convolutional Neural Networks

2. Recurrent Neural Network (RNN)

Humans do not start their thinking from scratch every second. As you read this report, you understand each word based on your understanding of previous words. You do not throw everything away and start thinking from scratch again. Your thoughts have persistence. Traditional artificial neural networks cannot do this, and it seems like a major shortcoming. For example, imagine you want to classify what kind of event is happening at every point in a movie. It is unclear how a traditional neural network could use its reasoning about previous events in the film to inform later ones.

Recurrent neural networks address this issue by allowing for a feedback between layers. They are networks that allow information to persist, where the output from previous step are fed as input to the current step[8] . Figure 2 shows a Recurrent Neural Network and how it's 'Recurrent' nature.

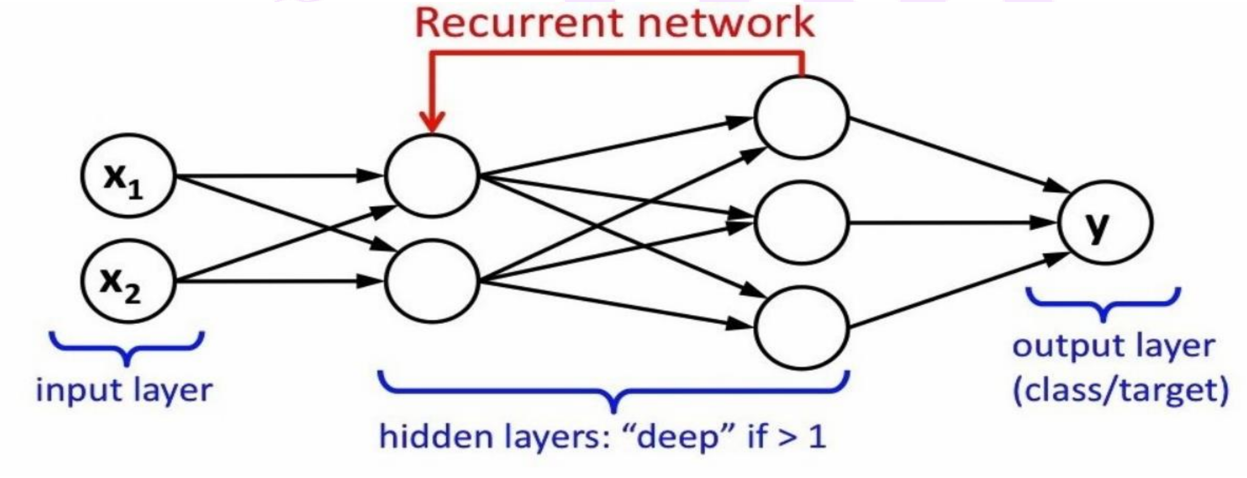


Figure 2. Recurrent Neural Network

3. Artificial Neural Networks (ANN)

They are one of the main tools used in machine learning. As the "neural" part of their name suggests, they are brain-inspired systems, which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as hidden layers consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize [14].

While neural networks (also called “perceptron”) have been around since the 1940s, it is only in the last several decades where they have become a major part of artificial intelligence. This is due to the arrival of a technique called “backpropagation,” which allows networks to adjust their hidden layers of neurons in situations where the outcome does not match what the creator is hoping for — like a network designed to recognize dogs, which misidentifies a cat, for example. Figure 3 shows an example of an Artificial Neural Network showing its input, hidden, and output

layers where the input is selected based on the number of input features while the output is based on the number of required classes to classify

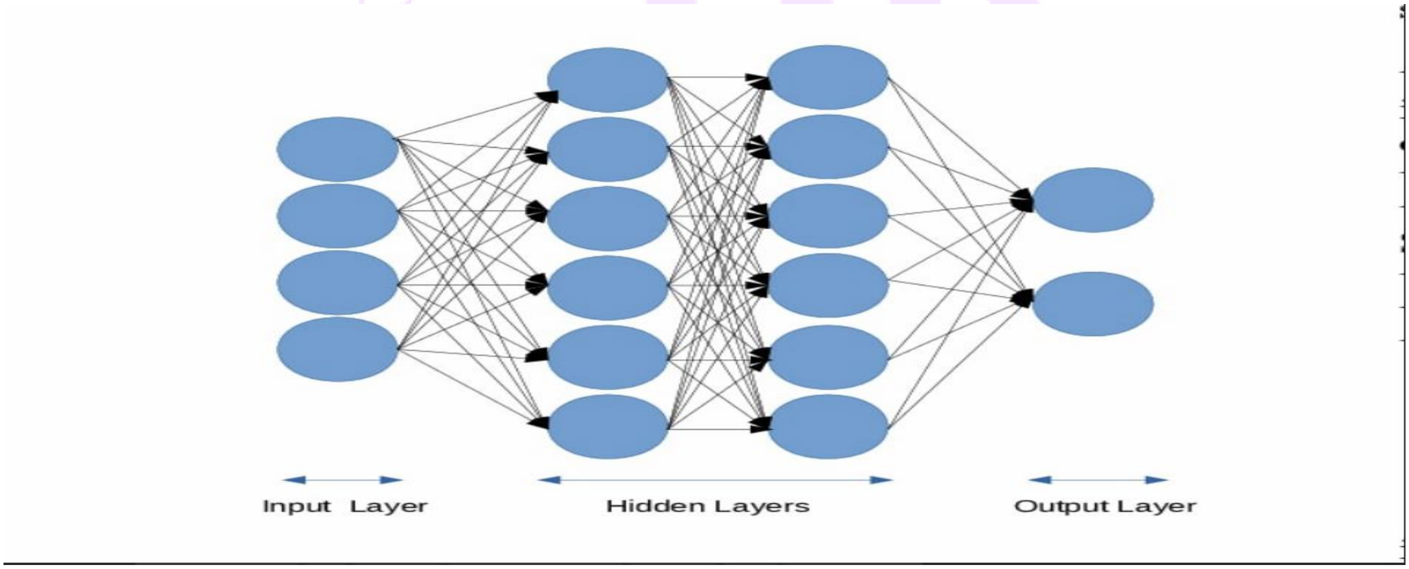


Figure 3. Artificial Neural Network

4. Adam Optimizer

Adaptive Moment Estimation (ADAM) is a method that calculates the individual

adaptive learning rate for each parameter from estimates of first and second moments of the gradients. It can be viewed as a combination of Adagrad, which works well on sparse gradients and RMSprop, which works well in online and nonstationary settings. Adam implements the exponential moving average of the gradients to scale the learning rate instead of a simple average as in Adagrad. It keeps an exponentially decaying average of past gradients while staying computationally efficient and having very little memory requirement. Adam optimizer is one of the most popular gradient descent optimization algorithms.

2.3 Hardware Specification

CPU	RAM	GPU	Time
Intel Core i7 6gh Generation HQ 2.7 GHz (6 Cores)	16 GB	AMD Radeon 2GB	45 : 60 Seconds
Intel Core i5-8250u 1.6 GHz (4 Cores)	8 GB	AMD Radeon TM 250 2GB	60 : 90 Seconds
intel core i5-3230M 2.6 GH (4 Cores)	4 GB	Nvidia NVS 5200M 3 GB	90 : 120 Seconds
intel core i7 8th generation 2.2GH (12 Cores)	16 GB	Nvidia GeForce GTX 1050	20 : 40 Seconds
intel core i7 7500u 7th generation 2.7GH (4 Cores)	8 GB	Nvidia GeForce 940MX	40 : 50 Seconds

Chapter 3

Proposed System and Methodology

1.1 System Use-Cases

Application

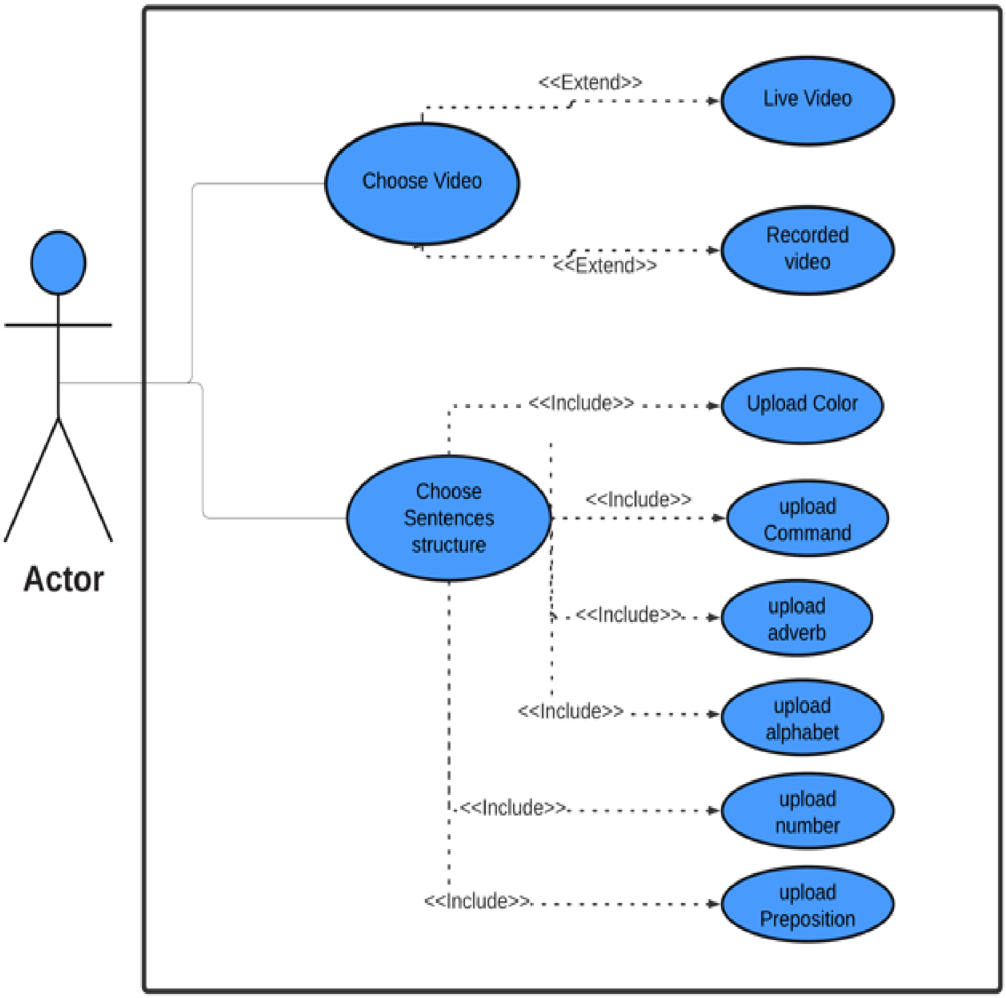


Figure 4. Application Use case

Server

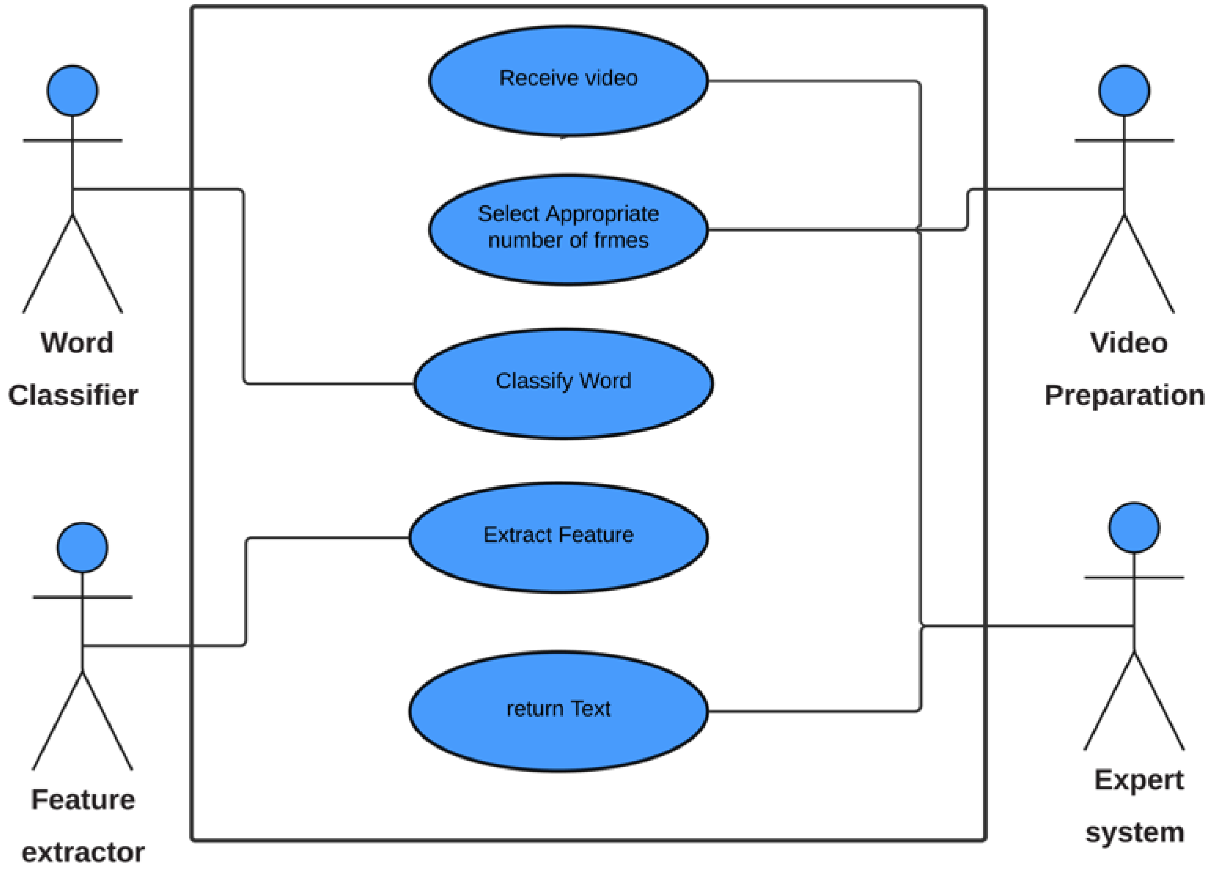
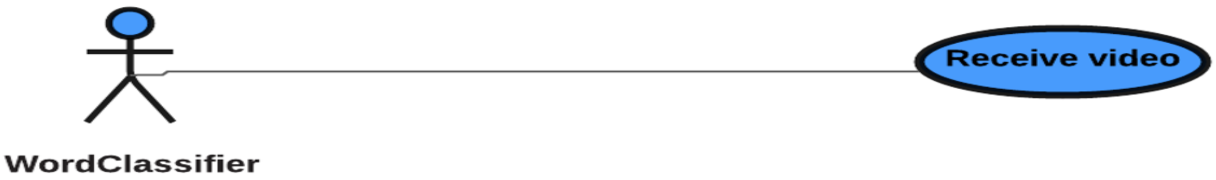
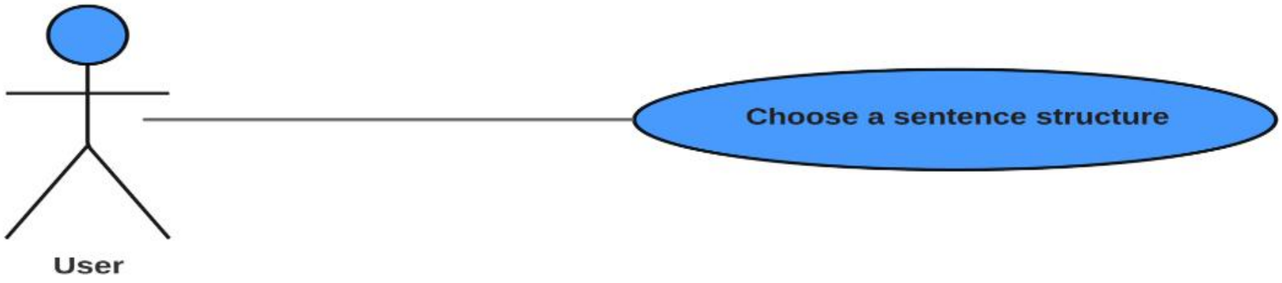


Figure 5. Server Use case

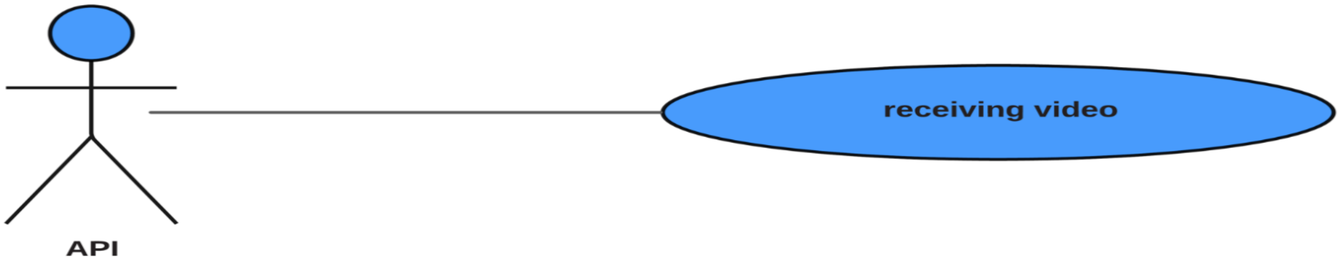
1.2 Use Case Description (Use case scenario)



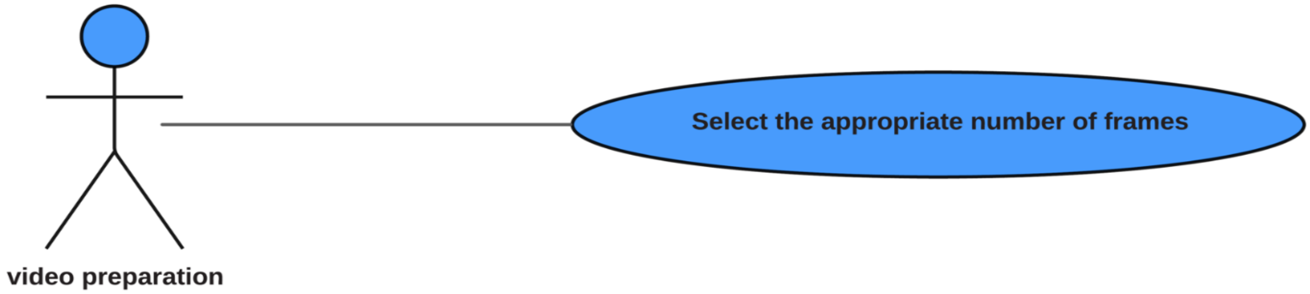
Use case name	**Choose a video**
Unique ID	user **001**
Area	Application
Actor(s)	user
Description	User adds a video to translate lip movement into words
Triggering Event	The user clicks the video selection button
Preconditions	The user needs to download the app The user needs access to the Internet Choose the sentence structure to represent in a video
Postconditions	The user uploaded the video to the server successfully Waiting for the expected word
Assumptions	The user predicted the words he made
Steps Performed	Information for Steps
Open the application and the Internet Choose the sentence structure you want to translate Click Add Video Choose whether you want to record a live video or a video stored in your phone	Step 3: Choose what you want to translate (alphabet - color - command - number -…..)
Extensions (Alternative Flows)	If you don't choose anything and he can't add videos, you will get a warning message



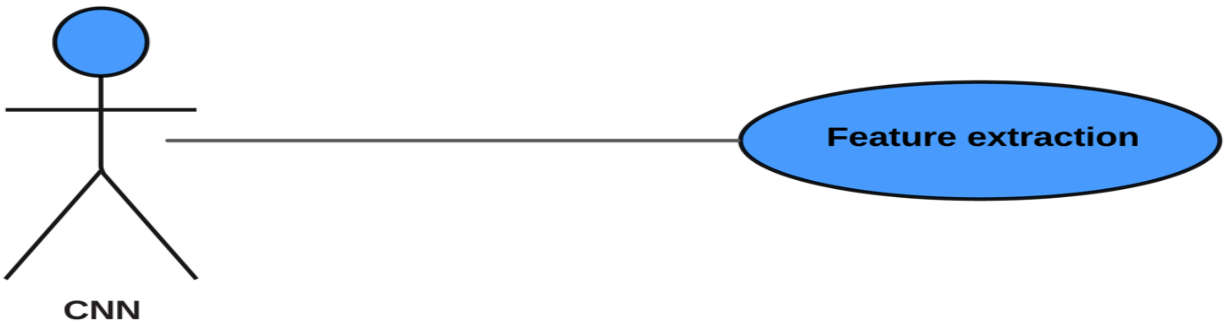
Use case name	Choose a sentence structure
Unique ID	User 002
Area	Application
Actor(s)	user
Description	Choose the appropriate types of words you want to translate
Triggering Event	Choose sentence structure by clicking on the checkboxes
Preconditions	The user needs to download the application The user needs access to the Internet
Postconditions	User can add videos
Assumptions	User can record videos for uploading
Steps Performed	Information for Steps
Open the app Choosing the right types of words Add videos	**Step 3: (alphabet – color -command – number - …..)**
Extensions (Alternative Flows)	If he doesn't choose any things and can't adding videos, he should get a warning message



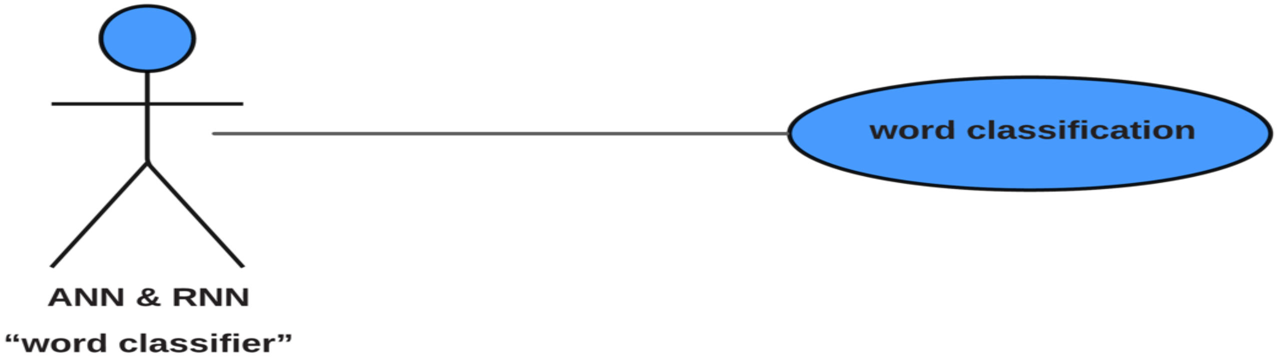
Use case name	receiving video
Unique ID	server 001
Area	server
Actor(s)	API
Description	Receiving video from the user after recording it
Triggering Event	Receive video from API
Preconditions	No preconditions
Postconditions	successful receiving
Assumptions	User can upload videos to server
Steps Performed	Information for Steps
Recording videos Get videos from API Send videos to prepare the video	**Step 2: From the user**
Extensions (Alternative Flows)	If he doesn't choose any things and can't adding videos, he should get a warning message



Use case name	Select the appropriate number of frames
Unique ID	server 002
Area	server
Actor(s)	video preparation
Description	Prepare the appropriate number of frames from the video
Triggering Event	Converts the video preparation to the appropriate number of frames and sends it to the feature extraction process
Preconditions	The server should succeed in receiving the videos
Postconditions	Returns the appropriate number of frames from the received video
Assumptions	The video can enter the feature extraction process
Steps Performed	Information for Steps
Receive video from API Select the appropriate number of frames from the videos Send the frames to the second step	**Step 3: Feature extraction**
Extensions (Alternative Flows)	If he doesn't choose any things and can't adding videos, he should get a warning message



Use case name	**Feature extraction**
Unique ID	Server 003
Area	Server
Actor(s)	CNN
Description	Getting the vector of features from the frames
Triggering Event	Receiving frames from video preparation process and returning feature vector
Preconditions	The video must be converted to the appropriate number of frames
Postconditions	Return Vector of Features
Assumptions	The vector can enter the following process "word classification"
Steps Performed	Information for Steps
Receive the appropriate number of frames Create a vector of feature Return Vector of Features	**Step 1: Video Preparation**
Extensions (Alternative Flows)	If he doesn't choose any things and can't adding videos, he should get a warning message



Use case name	word classification
Unique ID	server 004
Area	server
Actor(s)	ANN & RNN “word classifier”
Description	Get the correct word that fits the feature vector
Triggering Event	ANN & RNN classify the word
Preconditions	The feature vector must be valid
Postconditions	classified the word
Assumptions	The predicted word can return to the API
Steps Performed	Information for Steps
Receiving Vector of Features classified the word return classified the word	**Step 1: CNN** **Step 2: ANN & RNN**
Extensions (Alternative Flows)	If he doesn't choose any things and can't adding videos, he should get a warning message

1.3 Analysis Class

4.3.1 State Diagram

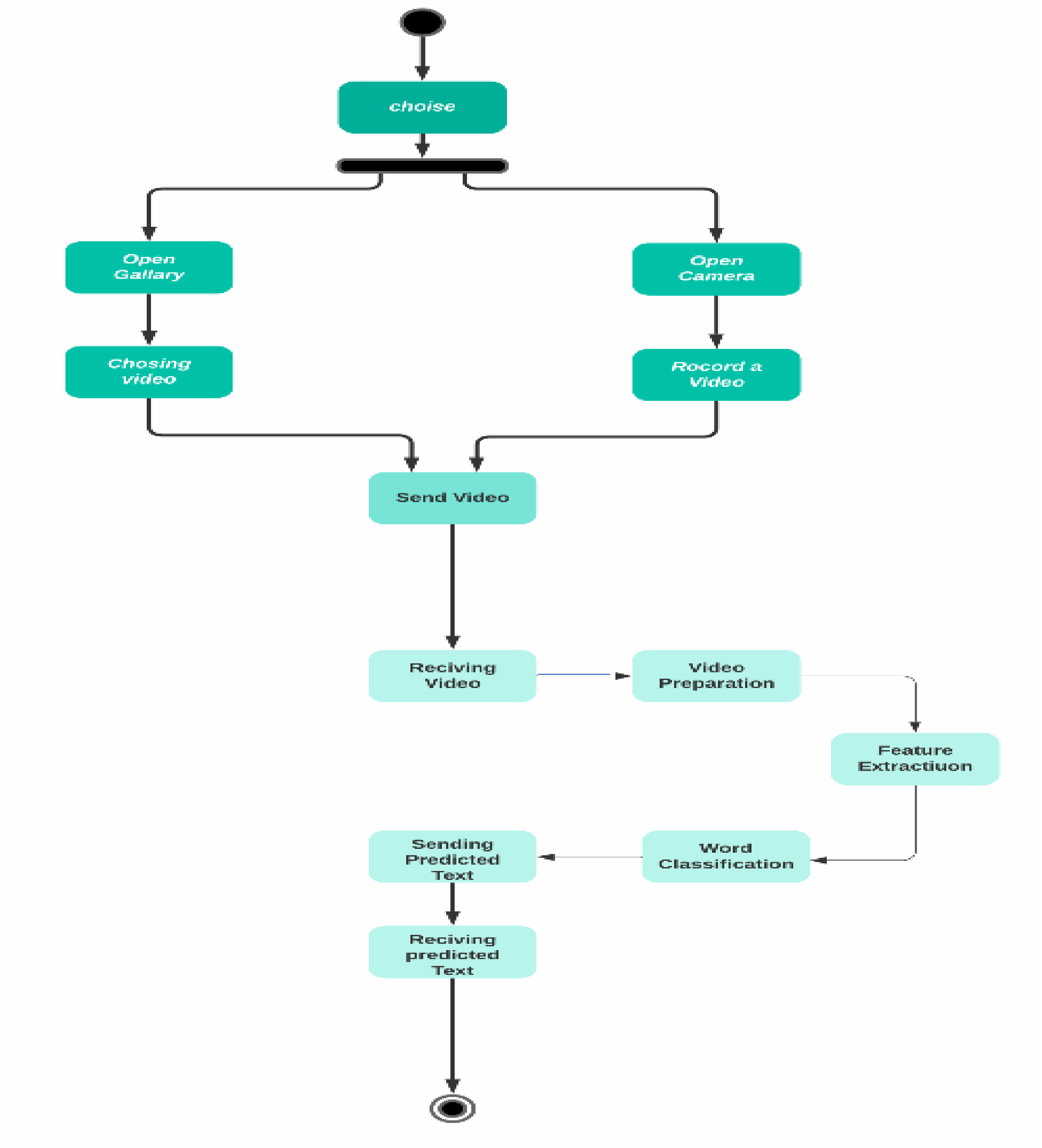


Figure 6. User State diagram

1.4 Interaction Diagram (Sequence Diagram)

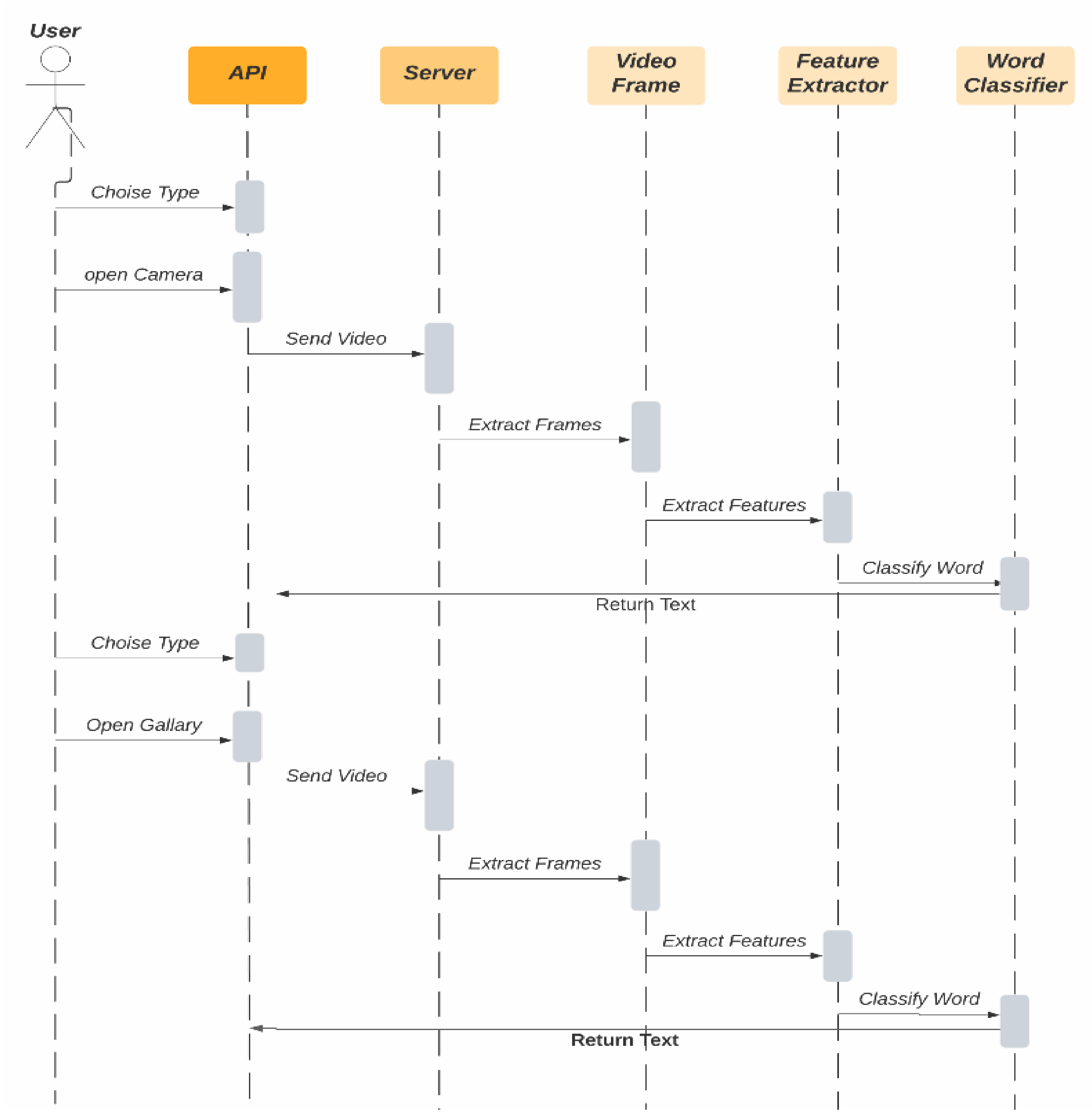


Figure 7. User Sequence diagram

1.4.1 System Architecture

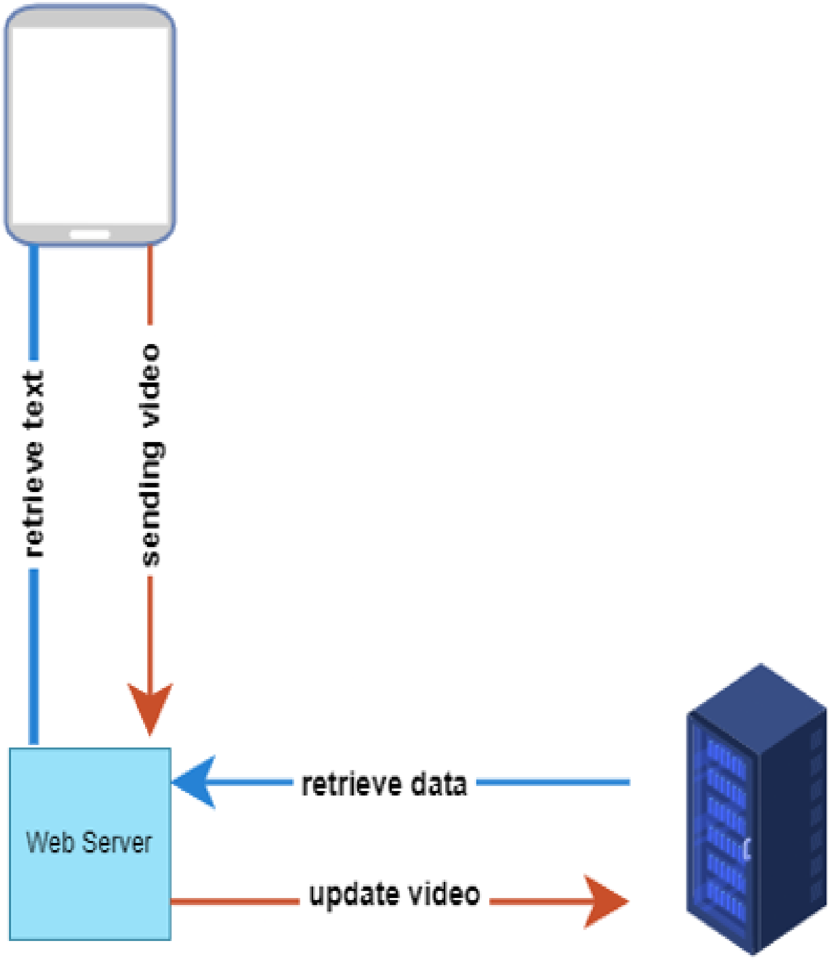


Figure 8. System Architecture

System Architecture Block Diagram.

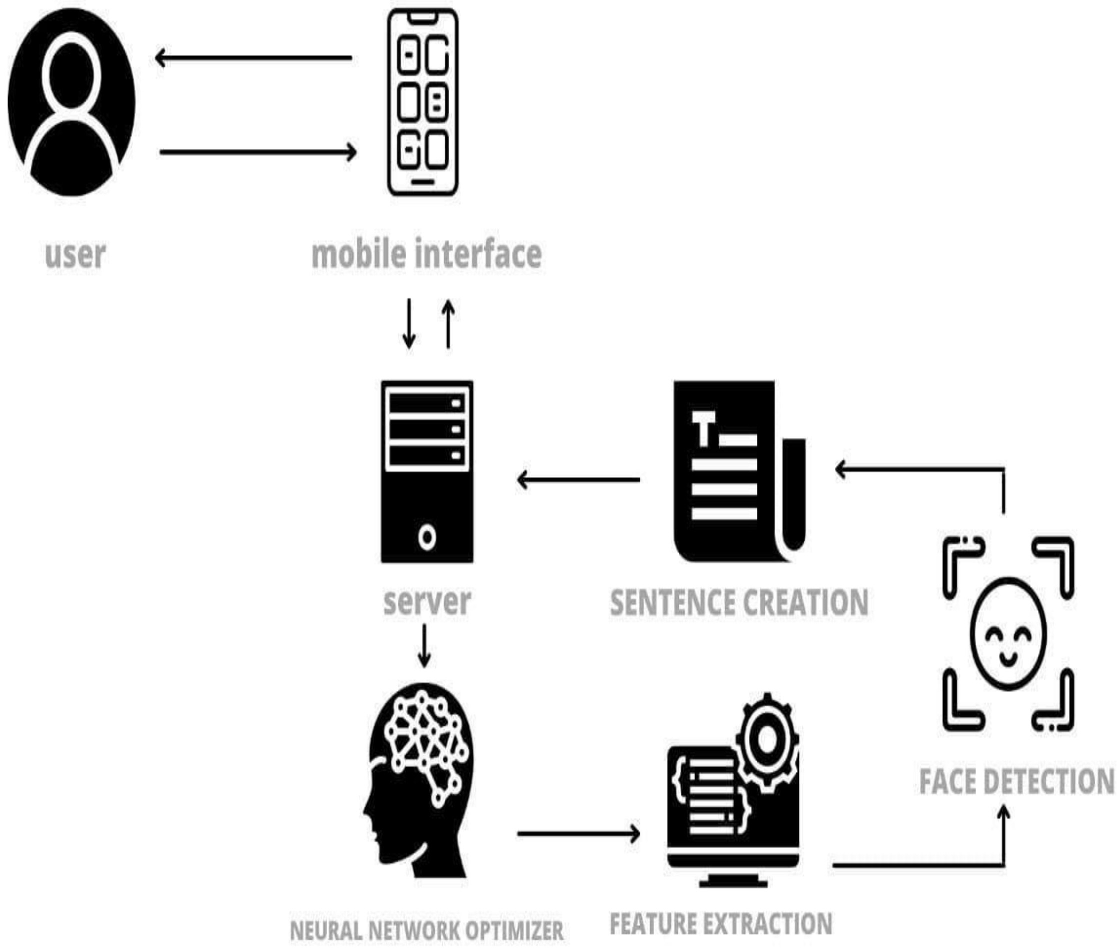


Figure 9. System Architecture Block Diagram.

Chapter 4

Risk and Functional and Non-Functional Requirement

1.1 Problem Statement /Constraint

3.1.1 Problem Statement

1. Different countries have their own sign language standards, and even the differences between British Sign Language and American Sign Language are significant.
2. Speech recognition in noisy environments (e.g. cars):

In noisy environments, it's difficult to hear the sound of video hence you can generate speak from lip movements by our project.

3. Silent dictation in public spaces:

for example, in reading places like libraries or any public place you can't talk with sound so you can use our project.

 Therefore, our aim to build a smart and intelligent mobile application that is able to see the user’s lips and transform their words into text.

3.1.2 Constraint

Constraint classification	Constraint	Effects
Event classification	1-light may be low and brightness may be very high	1-2-3: May leads to inaccurate predicted text. 4- If internet is low so it will take more time and May leads to incomplete process.
	2-Video quality may be very low	
	4-position of the speaker	
	3-Innternet connection speed may be slow	
Ambiguity constrains	1-similar letters	1-It may leads to inaccurate predicted text.
	2-more than one person in video	2-It may leads to unknown where speaker is?
Implementation constrains	1-Lack of data	1- Lack of enough data leads to cannot find all the words we want it to expect.

3.2 Project Plan

3.2.1 Project Plan

phase	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Gathering Information	🟥	🟥							
Define Requirements		🟦	🟦						
analysis			🟩	🟩					
design				🟨	🟨				
implementation					🟥	🟥	🟥		
Develop AI System							🟪	🟪	
Testing and Final Discussion									🟦

Figure 10. Project Plan

3.3 Quality Assurance Plan:

- **Black Box:**

  In this stage, we will use test inputs in our dataset in our system to ensure the accuracy of output.

- **White box:**

  - **Unity Testing:**

  In this stage of testing, we will take every component of our AI TEXT OF SILENCE system such as web service, CNN machine learning model, mobile application to test them separately.

  - **Validation Testing:**

  Validation testing is the process of ensuring if the tested and developed application satisfies its functionality requirements. The business requirement logic or scenarios have to be tested in detail. All the critical functionalities of an application must be tested here.

  - **Alpha:**

  In this part, a group of testers in our team test the product in a laboratory environment to ensure the efficiency of the product and fix errors.

  - **Beta:**

  In this stage of testing, the application has been sent to some users that can't speak or have a hearing impairment to test the system efficiency and its outputs are correct or not and retrieve feedback to our team

3.4 Requirements

3.4.1 Functional Requirements

1. A functional mobile app that controls workflow with servers, APIs, users, in organized, accurate, and quick methods.
2. A Server contains data and Machine Learning Models.
3. API to guarantee communication between server and application.
4. A functional and tested Machine Learning Model that analyzes lip movement to predict output text.
5. A system that specifies output text and generates an audio clip that helps user to communicate with other users.
6. The Application must forward the recorded video to the model for analysis and prediction text through an API
7. Web service to be used as a mobile app.

3.4.2 Non-Functional Requirements

1. Fast APIs and retrieve data from servers within seconds.
2. The speed of video analysis and the speed of word prediction from the movement of the lips within seconds
3. A clear, attractive, responsive, less cluttered and efficient user interface that ensures a good user experience and handles a variety of tasks.
4. The application works on most versions of Android and IOS
5. A well-organized and managed system architecture that facilitates maintenance, restoration and scalability.
6. The system correctly predicts most words from videos

3.5 System Request

This kind of problem is a very good challenge for AI/machine learning because it expands capabilities a machine to describe a human's innate and highly innate ability. In addition, there is a fair amount of data that can be used to train a machine to do this.

This project focuses on lip reading, lip reading is not a new science. Lip reading has been around in the local community as long as the language has been around. However, learning takes many years and can be difficult especially for people who have lost their hearing later in life or who have difficulty making sounds. Lip reading has deep roots in the intelligence community. There are entire audio and video recordings where the voice is too muted or too noisy to understand speech. Including multiple camera face directions[15].

1. **Project Sponsor**

- All people who suffer from hearing problems that happened to them in later life, as well as people who have speech problems, that is, they cannot get the sound out of their mouth.
- And also, people interested in improving the quality of video calls.

2. **Business need**

- It will facilitate communication between people who have lost their hearing in later life.
- Also, people who have trouble getting the sound out of the mouth.
- And assistance in communicating in video calls in noisy environments and in cases of sound interruption.

3. Business Requirements

- The application must be located on a mobile phone or device that can access the web pages
- It must be connected to the internet
- And that the phone has a camera to capture videos in which the movement of the lips will be translated
- And shoot the video in a place with good lighting so that the filming is done properly.

4. Business value

- Facilitating the process of communication between people who have a hearing problem or people who have problems removing sound from the mouth due to surgery or psychological problems.
- Improving future video calls, which currently depend on audio output only, but with the lip-reading technique, audio output will depend on audio and video outputs together, which leads to a significant improvement in the quality of video calls.

5. Special issue

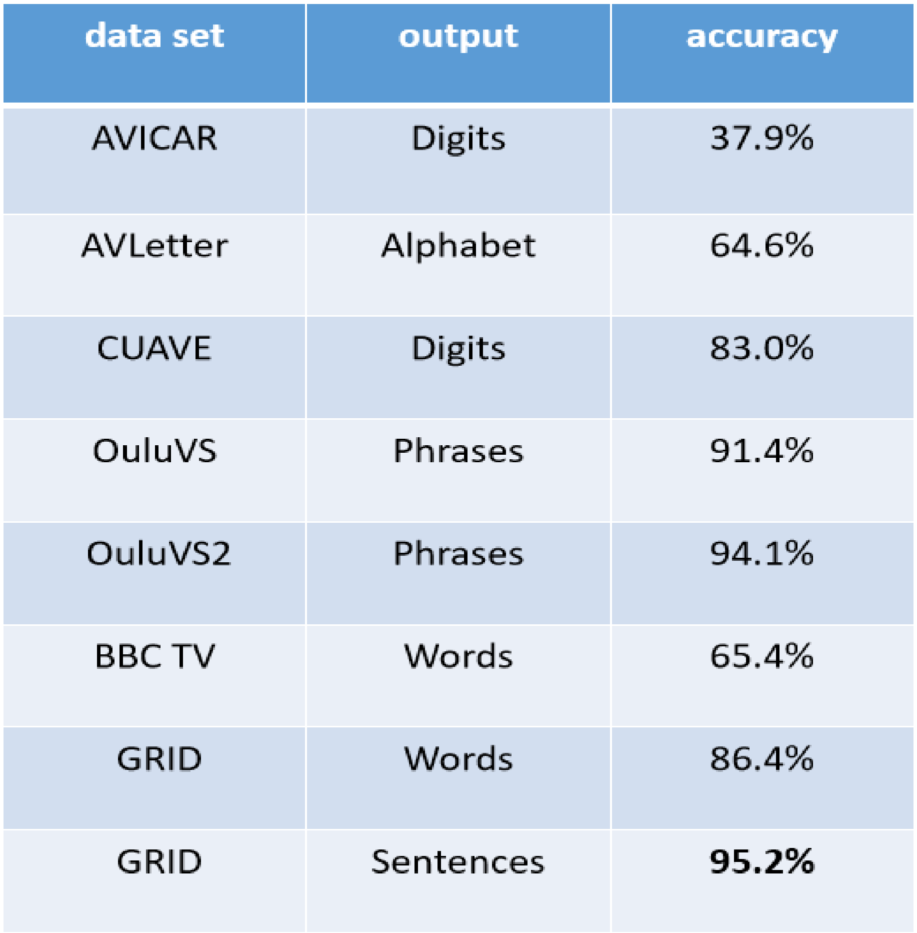
- The main problem that we will face is the quality of video capture and the surrounding lighting in the shooting environment.
- And also the confusion that occurs when similar letters are pronounced in the movement of the mouth
- There is also a major problem in implementing the project, which is the lack of enough data to be able to train the model so that it can find all the words we want it to expect.

3.6 Project Key Objectives

1. To take advantage of the current technological advance to make the lives of the hearing-impaired and who have problems in speak better.
2. It also aims at facilitating the everyday activities these people perform with their loved ones.
3. One of the major problems facing the hearing-impaired is communication with people who do not know sign language and may misunderstand them and to enhance voice when the voice is absent or corrupted by external noise. Or who suffers from a loss of ability to speak as a result of exposure to any surgical or other reason. Sign language is far from universal and leads to lots of misunderstandings; it is not a universal dialect.

3.7 Datasets:

Lip-reading datasets (AVICar, AVLetters, AVLetters2, BBC TV, CUAVE, OuluVS1, OuluVS2), but most only contain single words or are too small. One exception is the GRID corpus, which has audio and video recordings of 34 speakers who produced 1000 sentences each, for a total of 28 hours across 34000 sentences.



data set	output	accuracy
AVICAR	Digits	37.9%
AVLetter	Alphabet	64.6%
CUAVE	Digits	83.0%
OuluVS	Phrases	91.4%
OuluVS2	Phrases	94.1%
BBC TV	Words	65.4%
GRID	Words	86.4%
GRID	Sentences	**95.2%**

Figure 11. Datasets and Accuracy

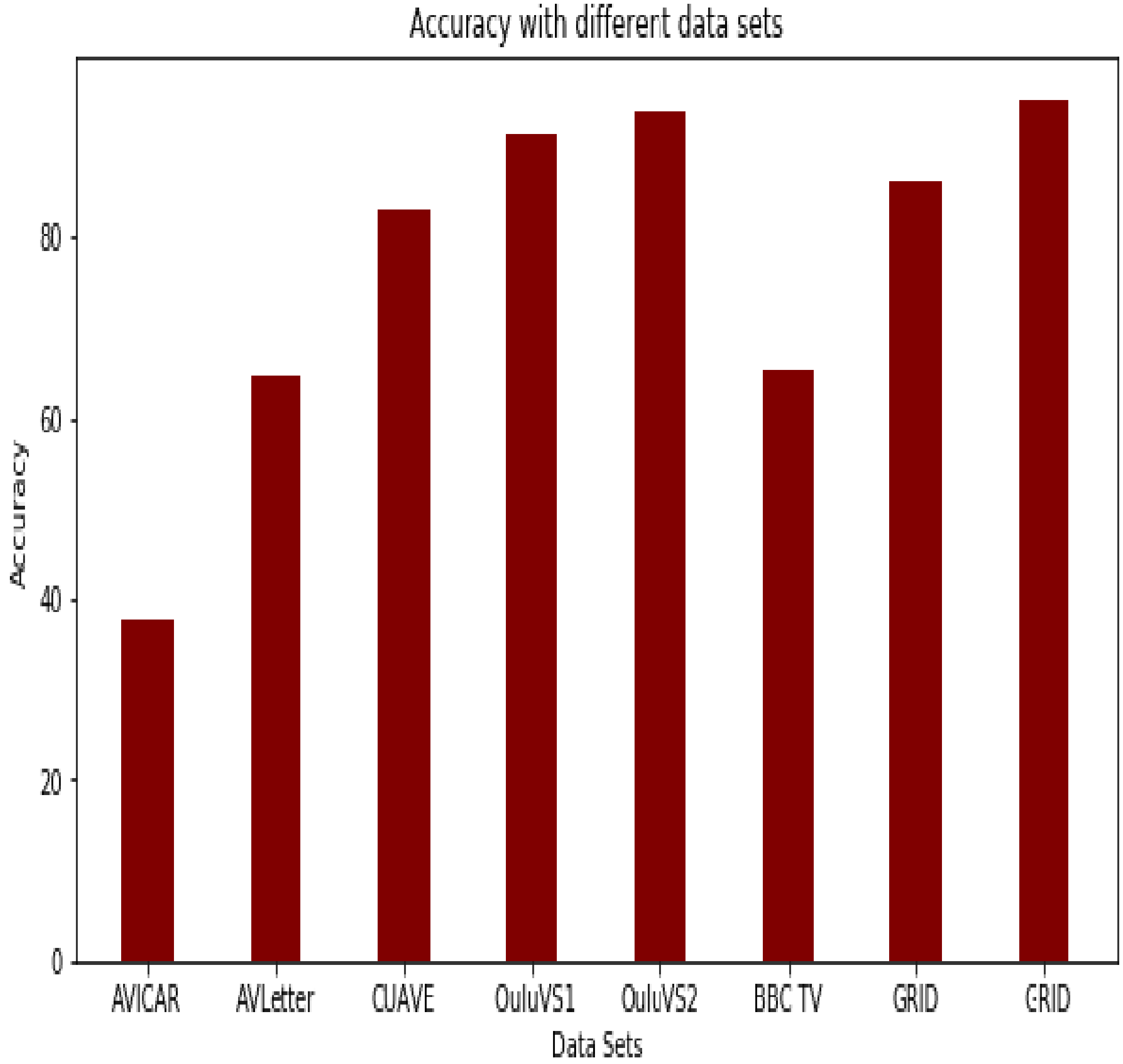


Figure 12. Accuracy with Different Datasets

We use the GRID corpus to evaluate our project because it is sentence-level and has the most data. The sentences are drawn from the following simple grammar: command + color + preposition + letter+ digit + adverb, where the number denotes how many word choices there are for each of the 6 word categories. The categories consist of, respectively, {bin, lay, place, set}, {blue, green, red, white}, {at, by, in, with}, {A, . . . , Z}\{W}, {zero, . . . , nine}, and {again, now, please, soon}, yielding

64000 possible sentences. For example, two sentences in the data are “set blue by a four please” and “place red at C zero again”.

Examples of Dataset:

Chapter 5

System Design

5.1 User Interface

TOS’s user guide on how to install, run, and use the application. The user manual is designed to be as easy and simple and possible to enable all types of users to easily access the application. After downloading and installing the application, the application will show in your app drawer as shown in Figure 13.

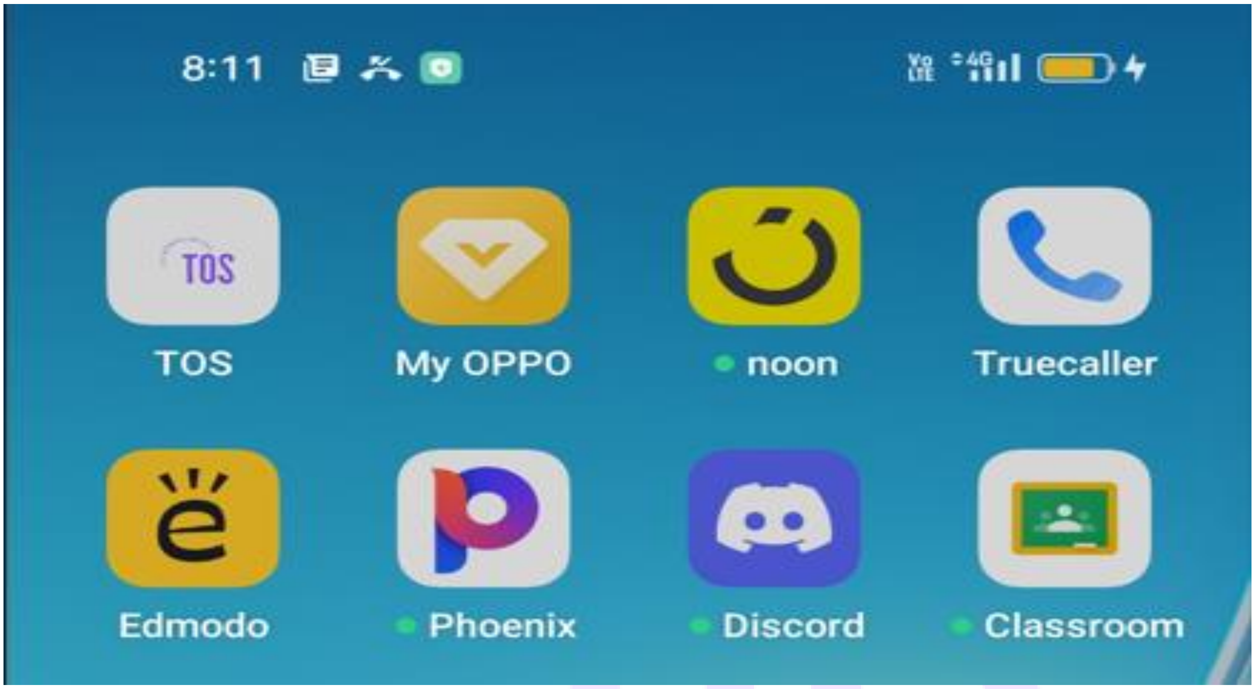
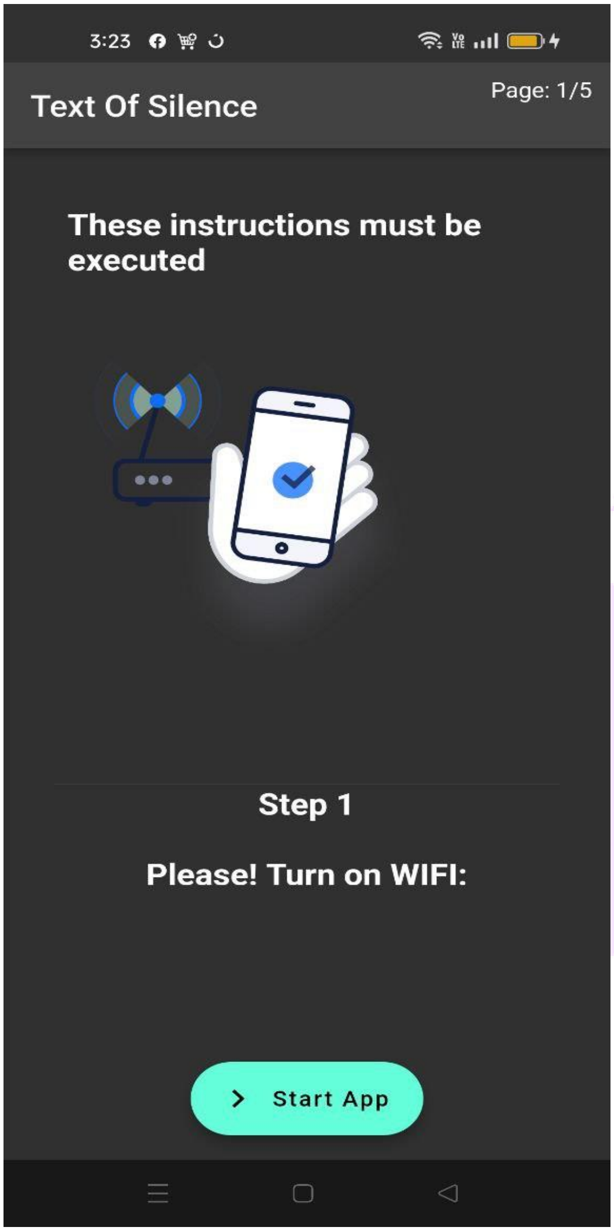


Figure 13. TOS in application drawer

Opening the app takes you to the instructions screen which consists of series of steps to make sure everything is working properly and is on track.

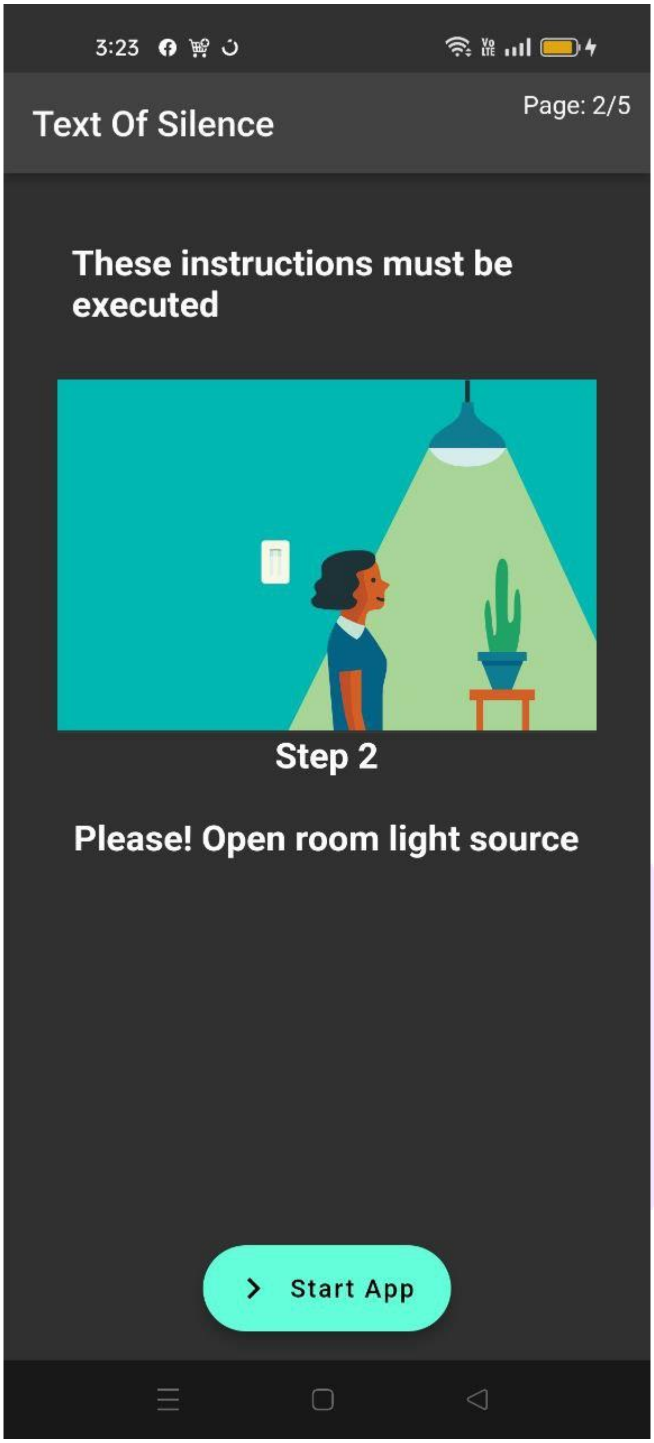


These instructions must be executed

Step 1

Please! Turn on WIFI:

Figure 14. Instructions Screen (Step 1)



Text Of Silence

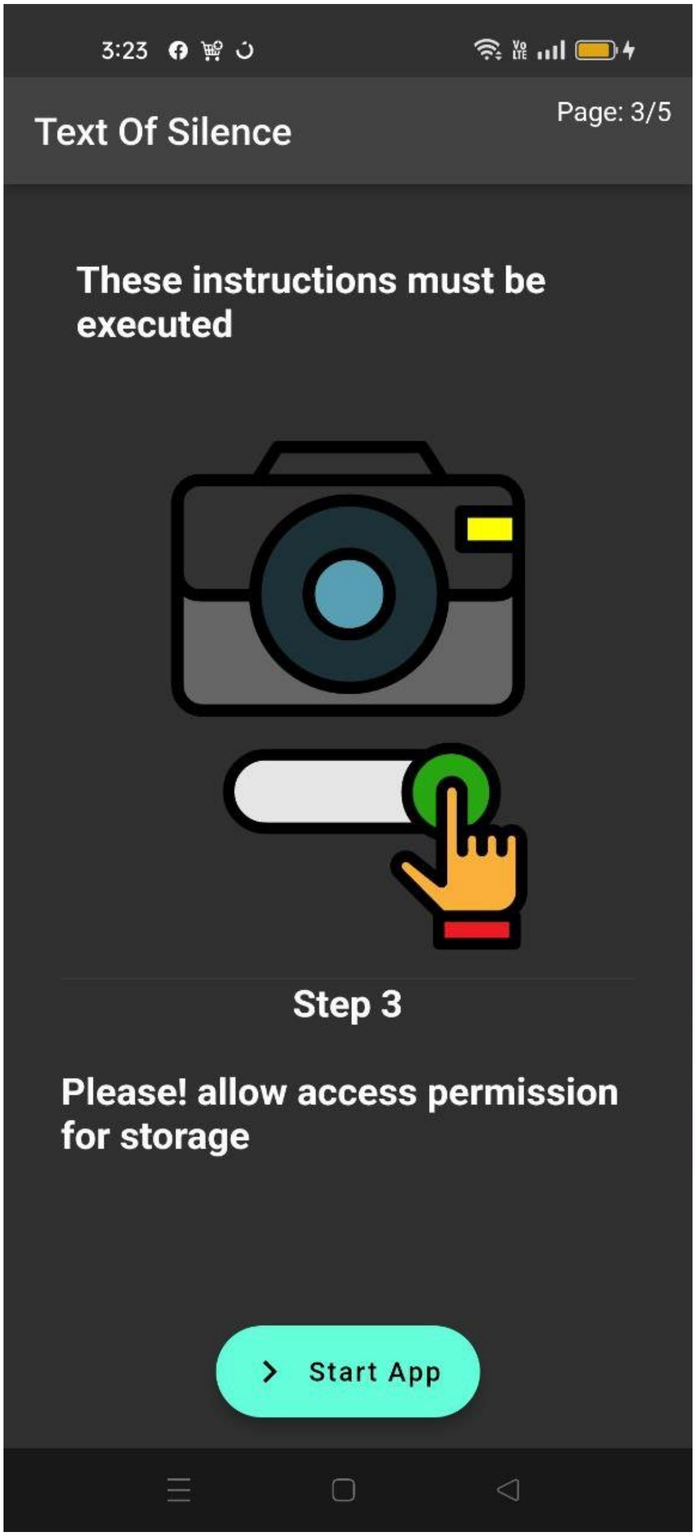
Page: 2/5

These instructions must be executed

Step 2

Please! Open room light source

Figure 15. Instructions Screen (Step 2)


3:23

Text Of Silence

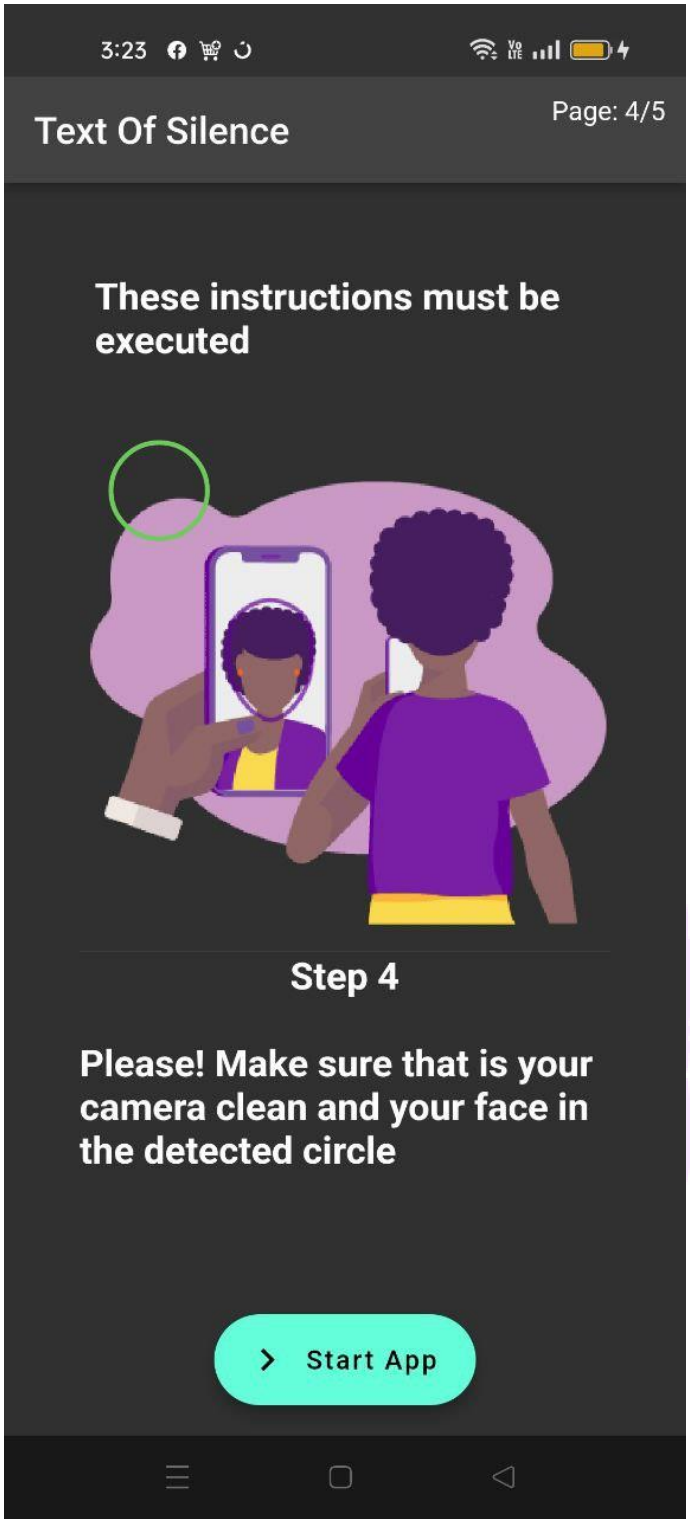
Page: 3/5

These instructions must be executed

Step 3

Please! allow access permission for storage

Figure 16. Instructions Screen (Step 3)

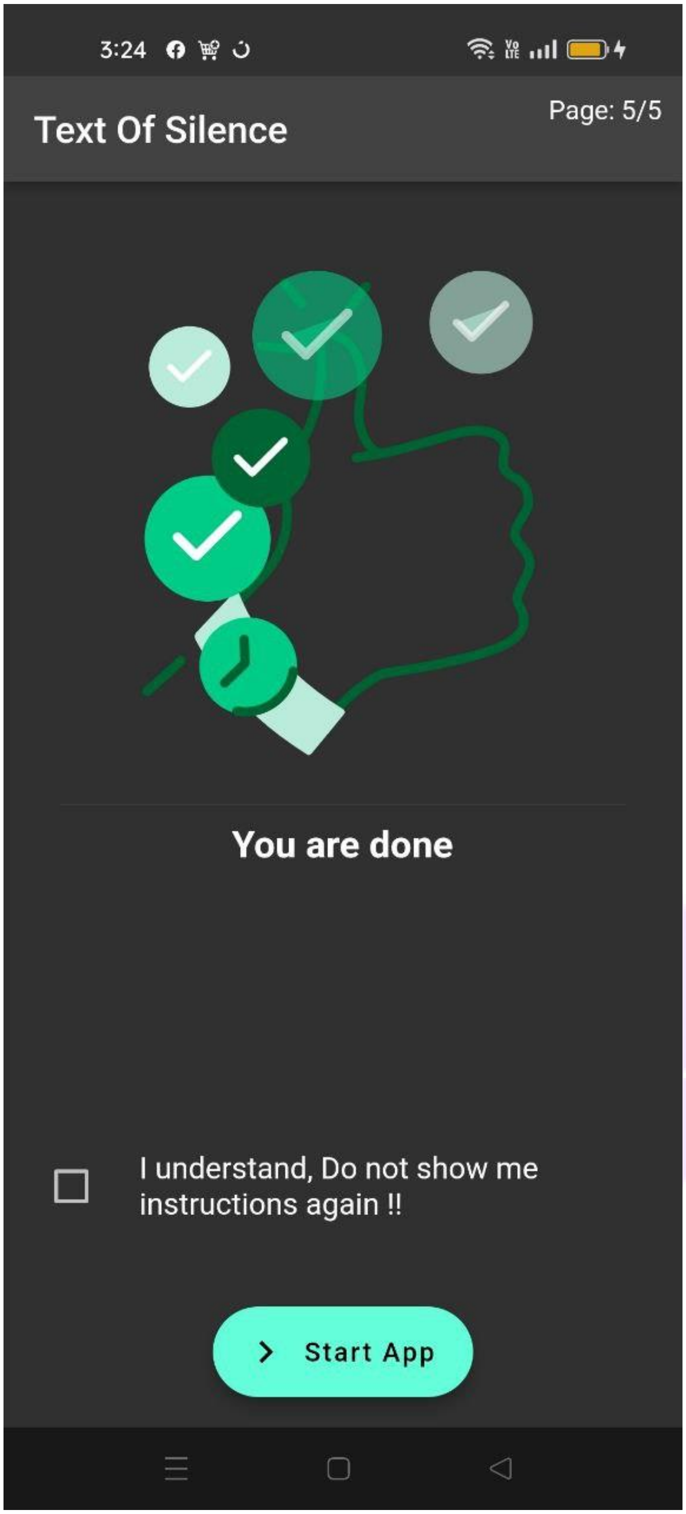


These instructions must be executed

Step 4

Please! Make sure that is your camera clean and your face in the detected circle

Figure 17. Instructions Screen (Step 4)



You are done

□ I understand, Do not show me instructions again !!

Figure 18. Instructions Screen (Step 5)

Then you will be navigated to the Home Screen shown in Figure 19. This is just a welcome screen that you just need to click on the “Upload video” button.

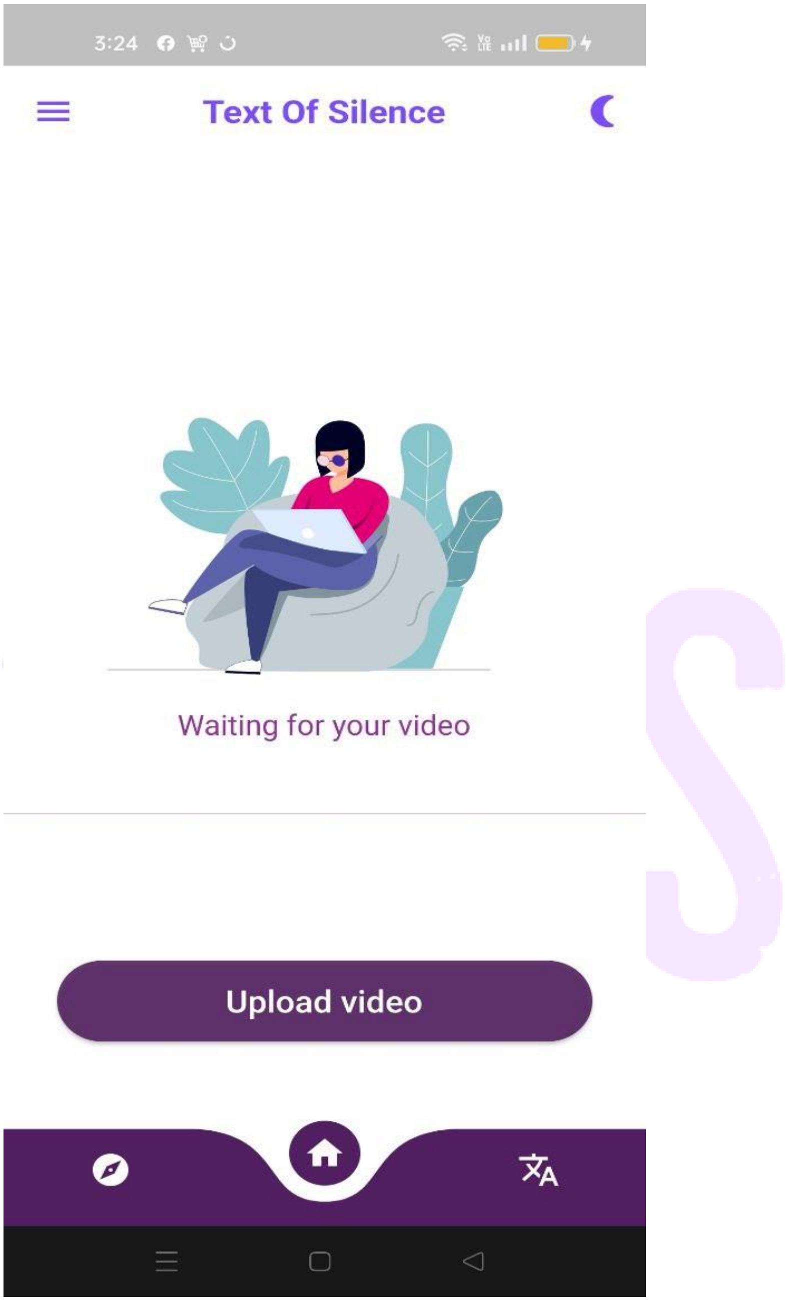


Figure 19. Home Screen

After selecting desired video as shown in Figure 20. It will be uploaded to the server to be processed

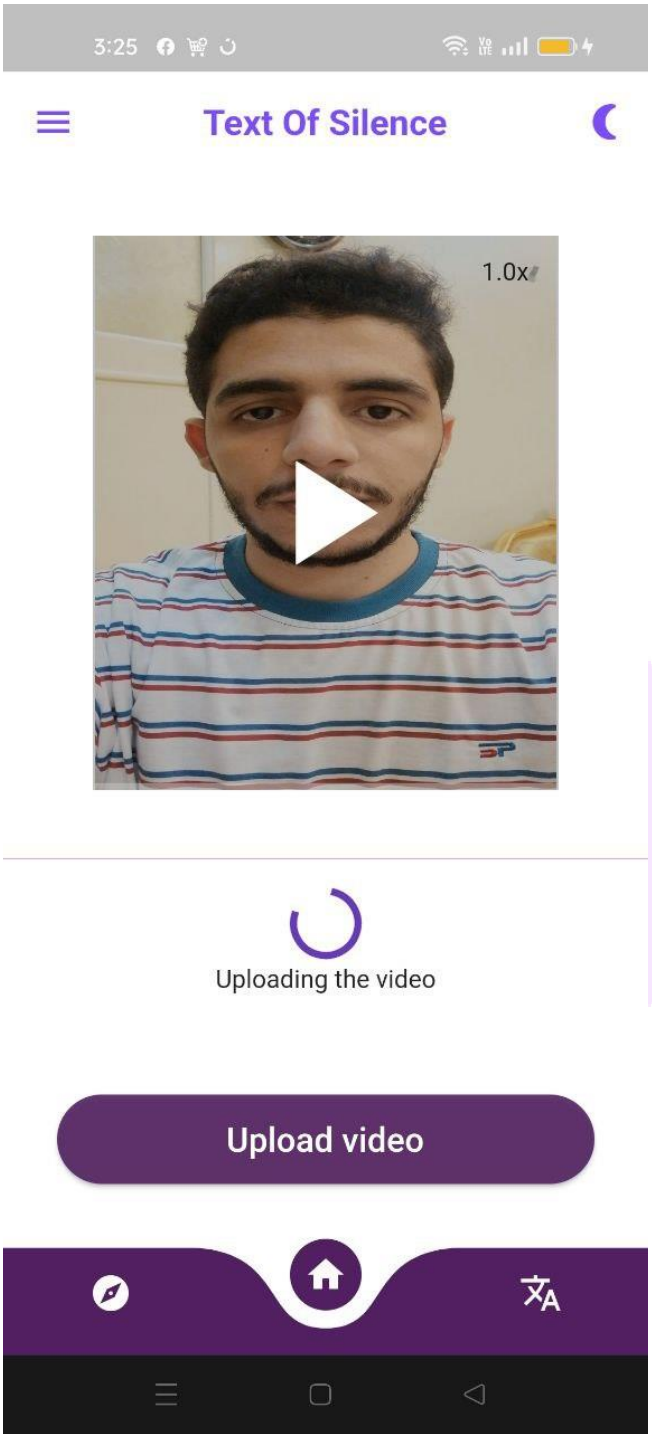
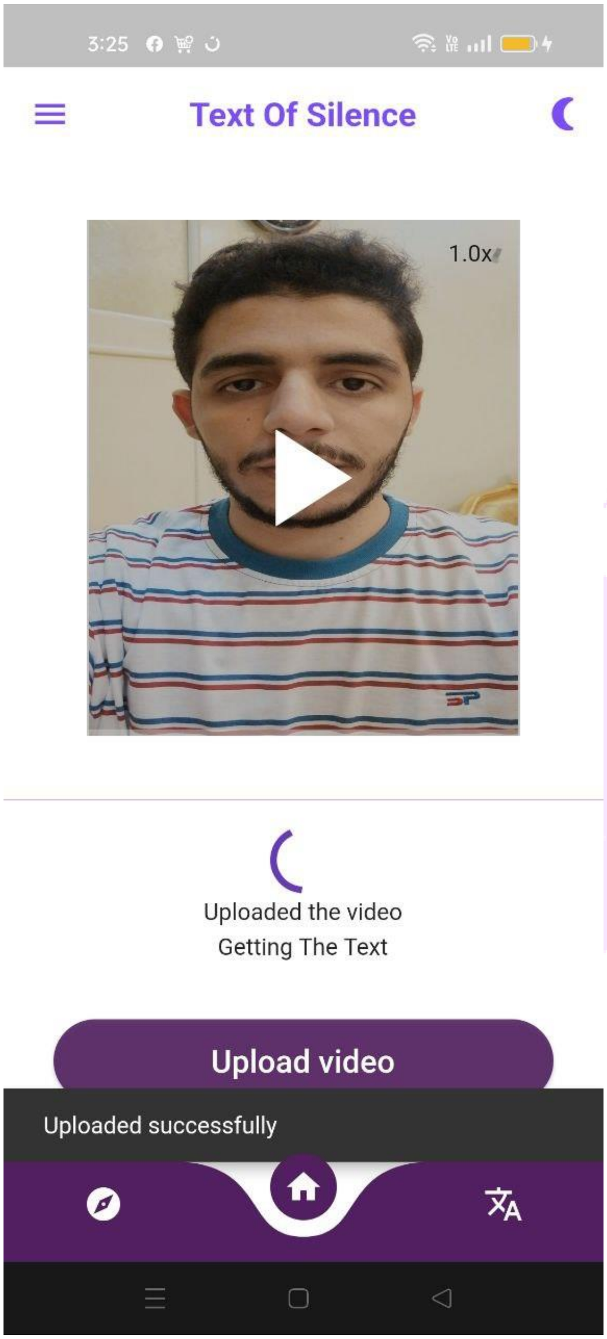


Figure 20. Uploading video

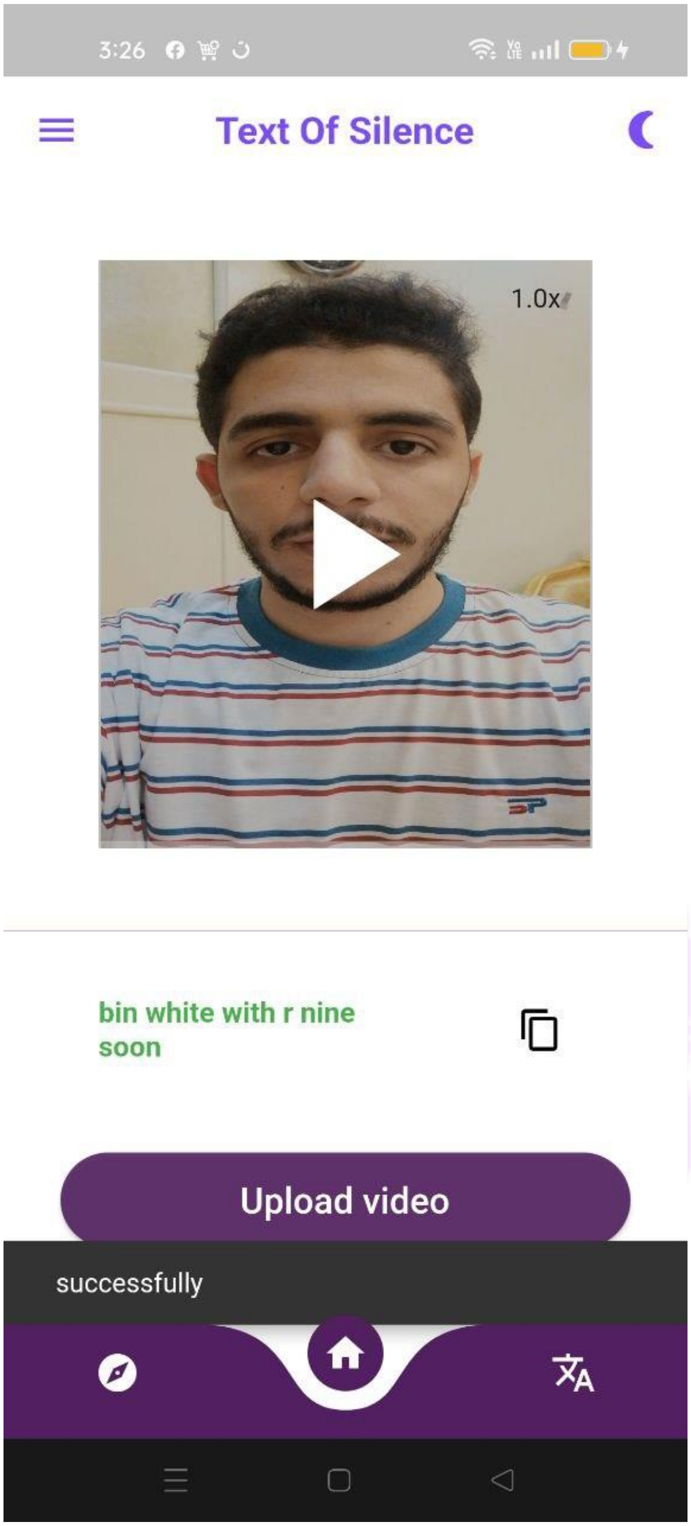
After uploading process has finished, the model extracts the words being said in the video as shown in Figure 21.



Uploaded the video
Getting The Text

Uploaded successfully

Figure 21. Getting the text

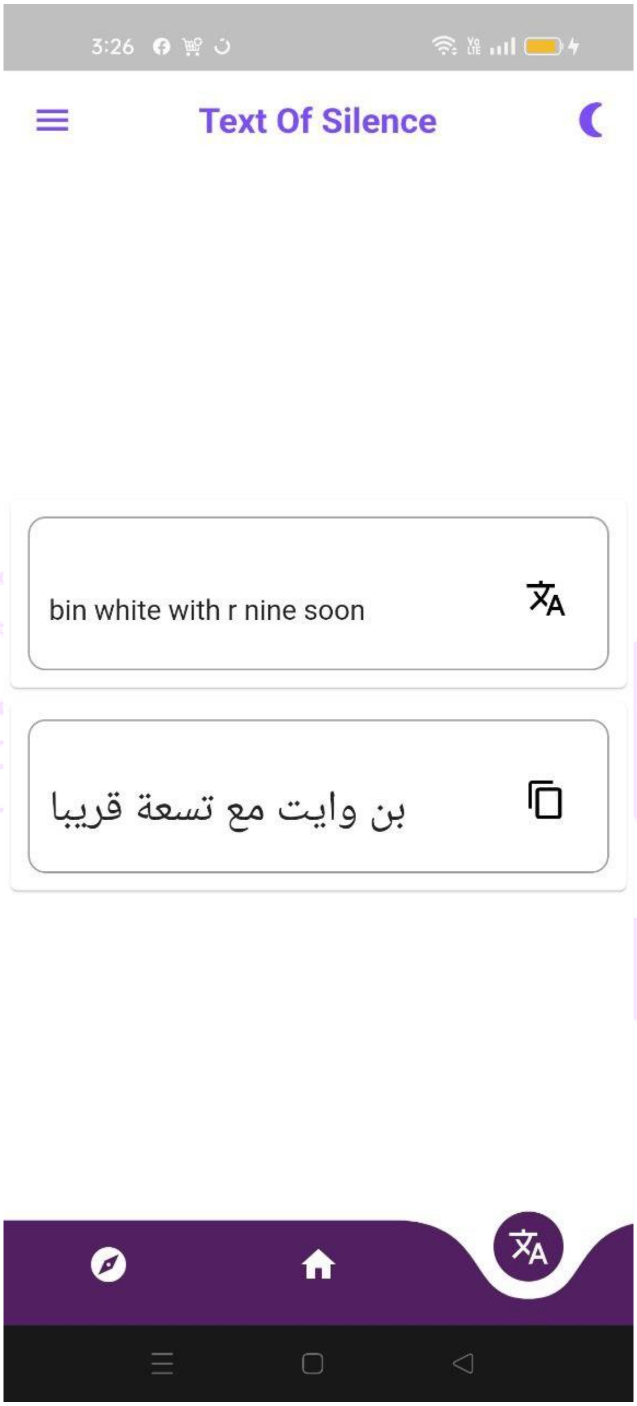


bin white with r nine soon

successfully

Figure 22. Getting the result

If desired, users can translate the output by switching to the right tab as shown in figure 23.



bin white with r nine soon

بن وايت مع تسعة قريبا

Figure 23. Translation

5.2 Samples:

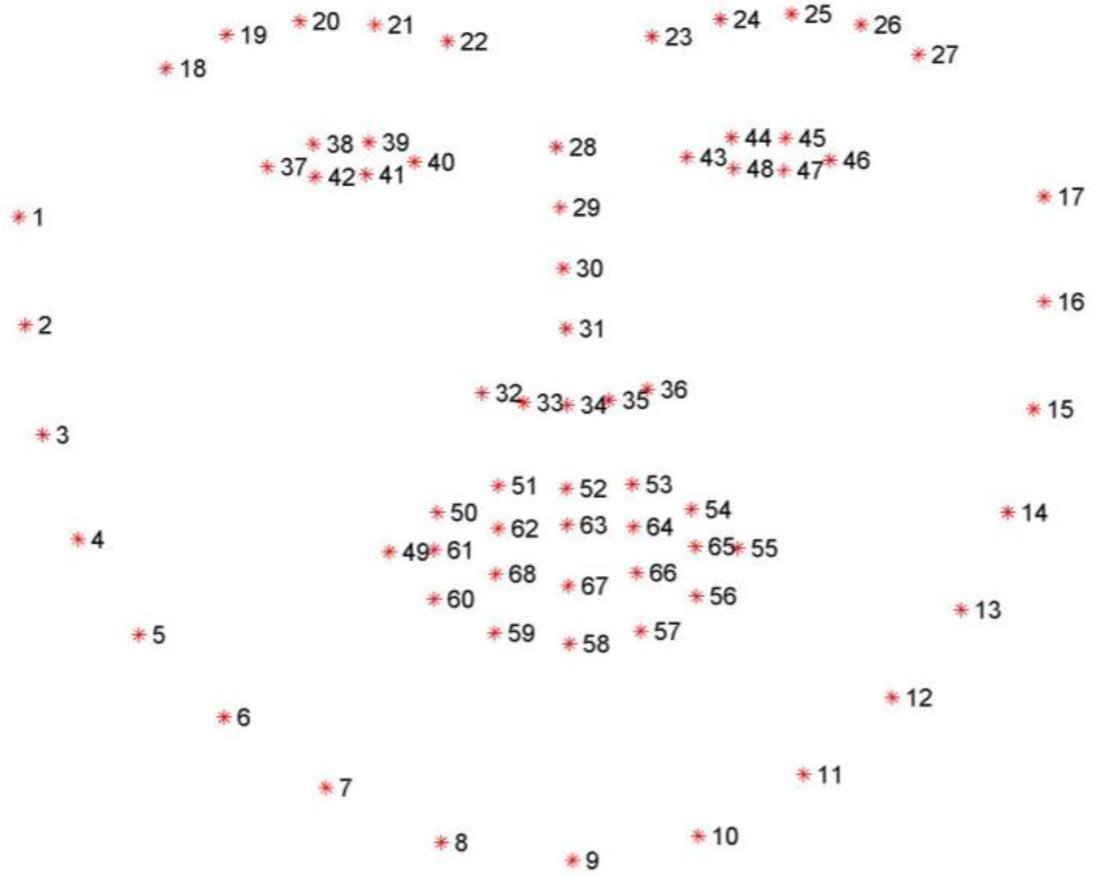
Data Preparation

We have to make some transformations and approximations in order to make it usable.

Capturing Lip

several approaches to use in order to capture the lips from our data set that will be used to train and test the network model

- Skin Segmentation Algorithms to detect skin region of speaker then applying contours on segmented parts to detect face followed by using human face aspect ratio to detect lips region.
- Applying Haar cascade on the face to extract face interest points which includes the mouth interest points that is used in feature extraction phase.
- Apply Facial landmarks on dlib, the mouth can be accessed through points [48, 68]



After pre-processing our dataset and normalizing the videos to 30 frames each, we started extracting the mouth regions from the videos to prepare our data for CNN training. We started by extracting video frames and then extracting the mouth region, in this part we compared the performance of various extraction techniques

	DLIB	Haar Cascade
Time to extract per video	72 Seconds	2.5 Seconds
Accuracy in extraction	~100%	~85%

We found that using DLIB produced an accuracy of almost 100% but made the segmentation process extremely slow and not viable for a large dataset.

That is why we decided to go with using Haar Cascades in the segmentation process as it produces acceptable results in the shortest time

Sample video and output:

Figure. Sample video

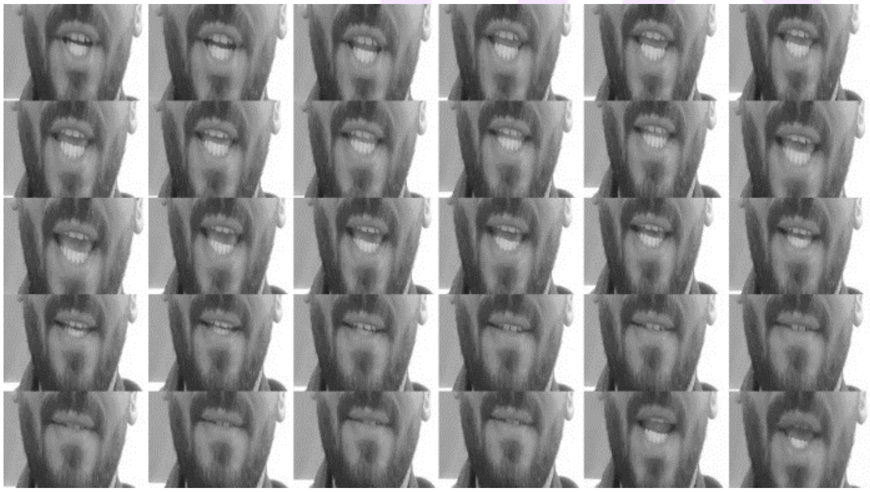


Figure 24. Lip region after segmentation.

Chapter 6

Experimental Result and Discussion

6.1 Result and Outputs

6.1.1 Confusion matrix of the model

According to "DeLand (1931) and Fisher (1968), Alexander Graham Bell" first hypothesised that multiple phonemes may be visually identical on a given speaker. This was later verified, giving rise to the concept of a viseme, a visual equivalent of a phoneme (Woodward & Barber, 1960; Fisher , 1968).

For our analysis, we use the phoneme-to-viseme mapping of Neti et al. (2000), clustering the phonemes into the following categories:

Lip-rounding based vowels (V),

Alveolar-semivowels (A),

Alveolar-fricatives (B),

Alveolar (C),

Palato-alveolar (D),

Bilabial (E),

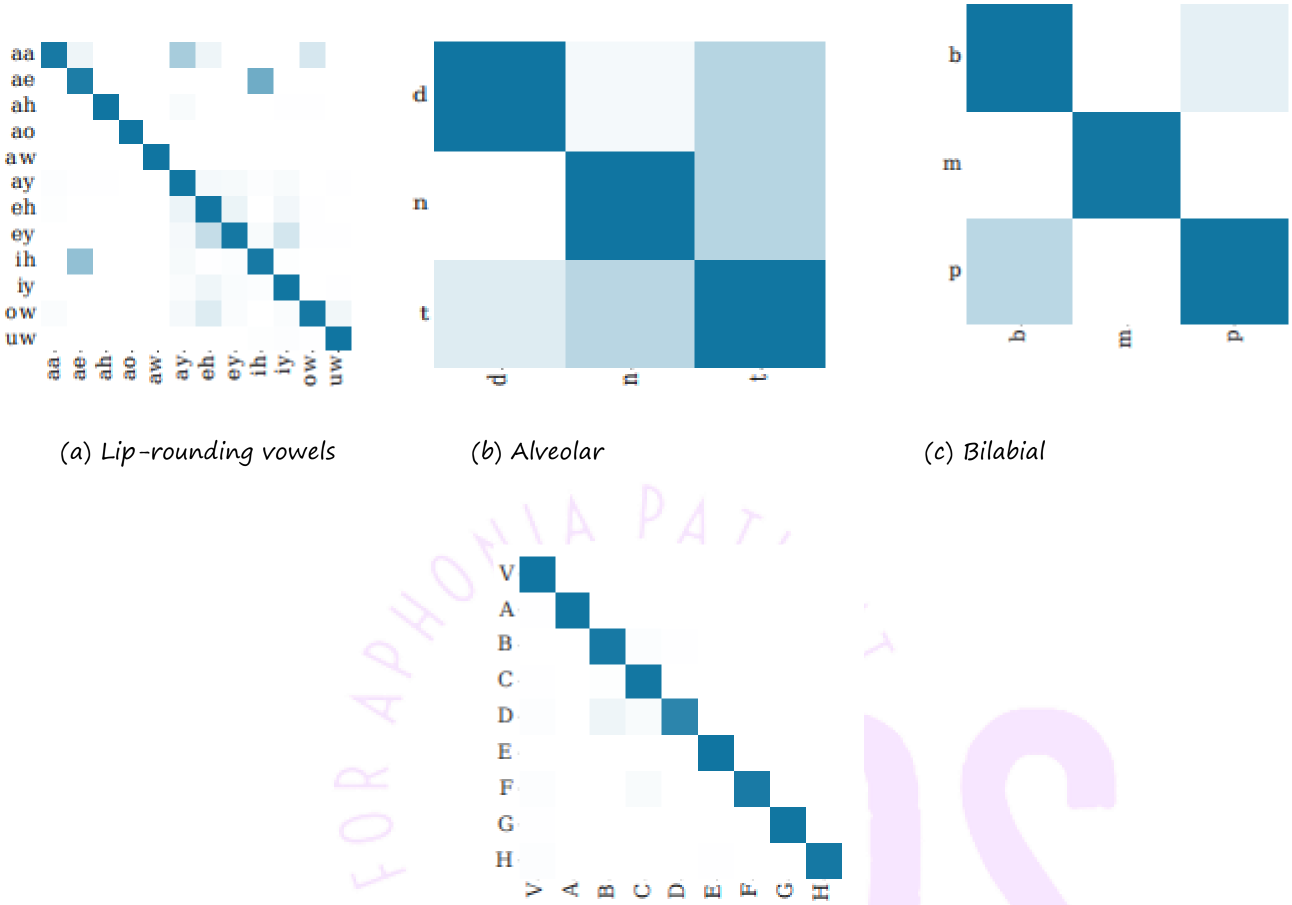
Dental (F), L

abio-dental (G),

and Velar (H).

the GRID corpus contain 31 out of the 39 phonemes in ARPAbet.

We compute confusion matrices between phonemes and then group phonemes into viseme clusters, following Neti et al. (2000). Figure 3 shows the confusion matrices of the 3 most confused viseme categories, as well as the confusions between the viseme categories.



(a) Lip-rounding vowels

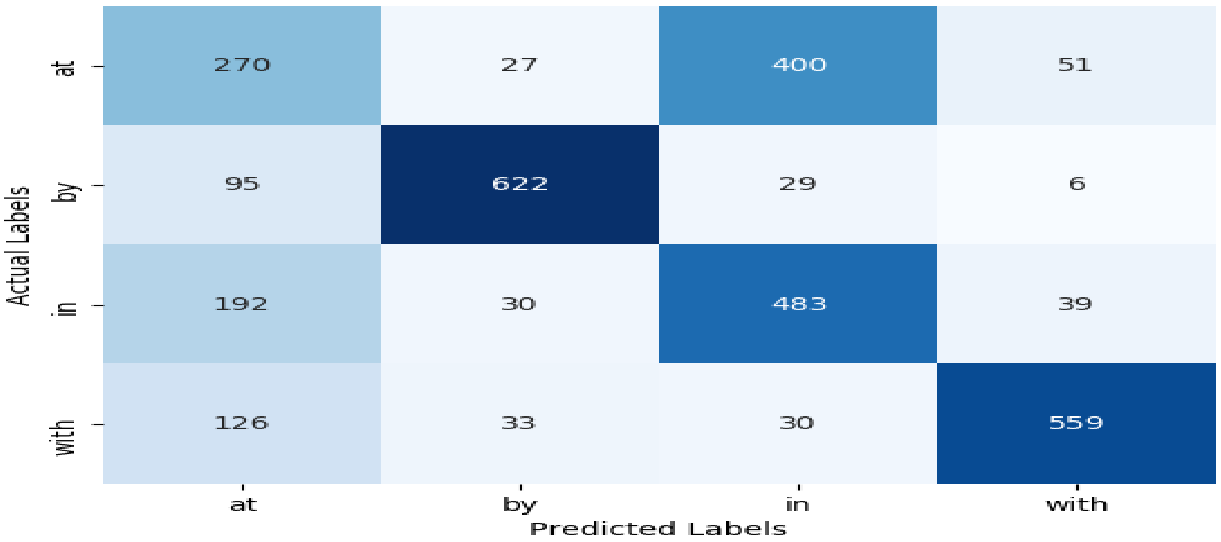
(b) Alveolar

(c) Bilabial

(d) Viseme Categories

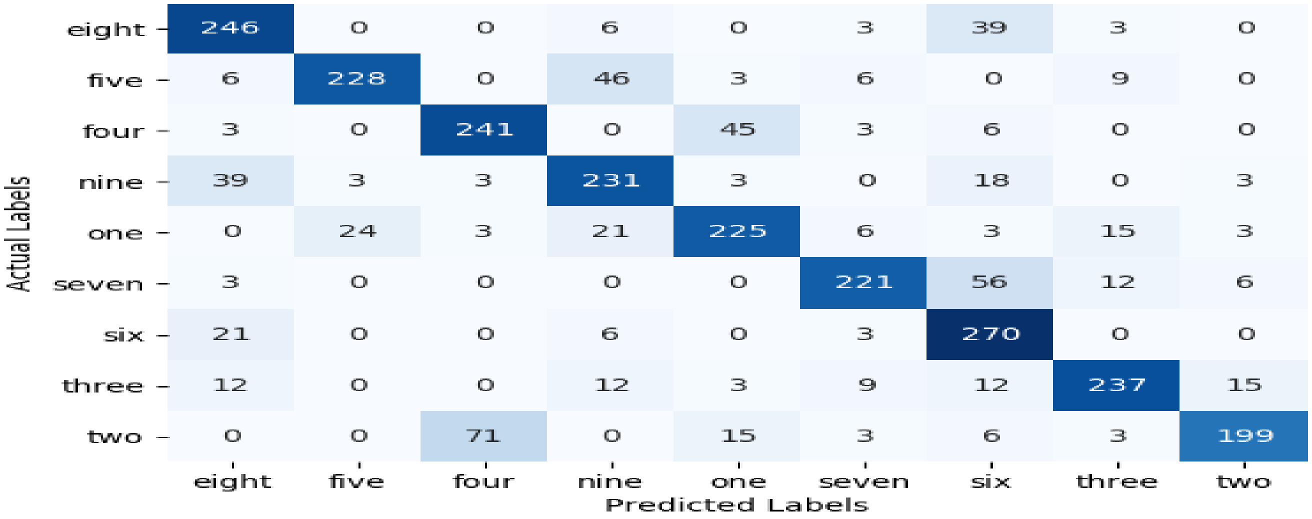
Figure : Confusion matrix of the model

6.1.2 : Confusion matrix of every part of sentences



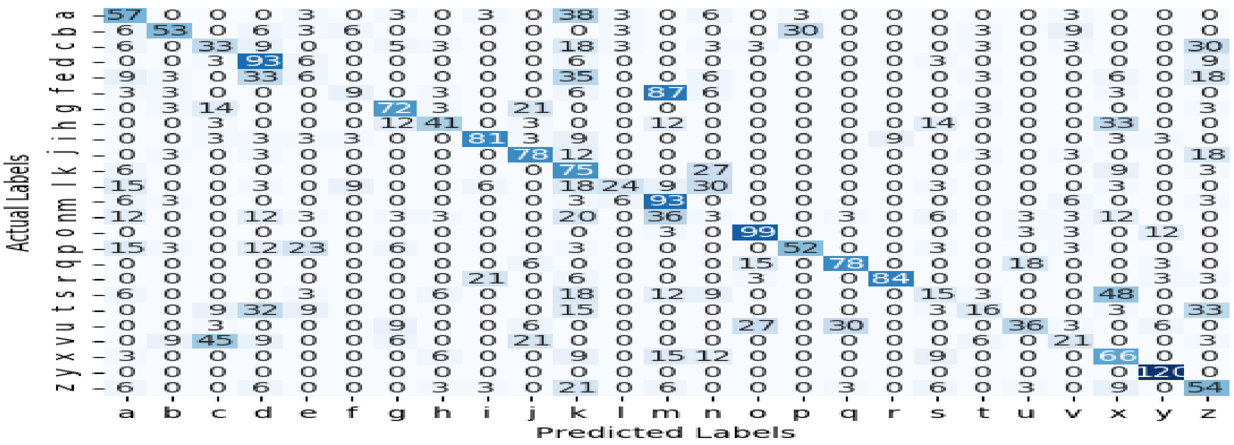
Actual Labels \ Predicted Labels	at	by	in	with
at	270	27	400	51
by	95	622	29	6
in	192	30	483	39
with	126	33	30	559

Prepositions CNN Confusion Matrix

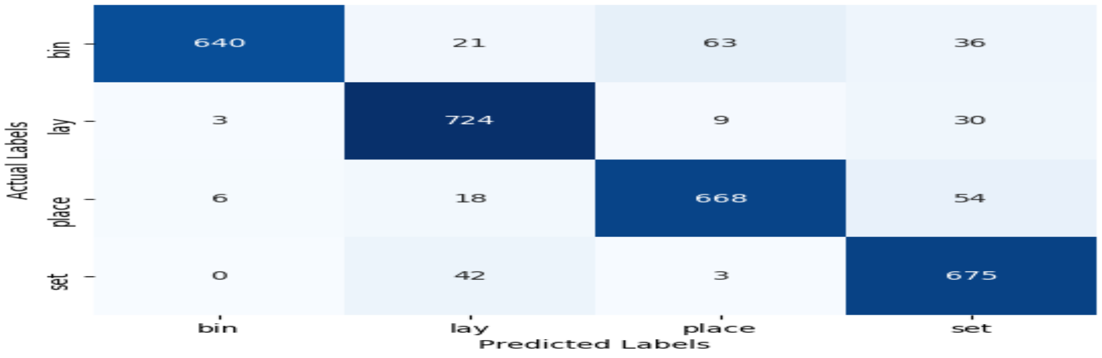


Actual Labels \ Predicted Labels	eight	five	four	nine	one	seven	six	three	two
eight	246	0	0	6	0	3	39	3	0
five	6	228	0	46	3	6	0	9	0
four	3	0	241	0	45	3	6	0	0
nine	39	3	3	231	3	0	18	0	3
one	0	24	3	21	225	6	3	15	3
seven	3	0	0	0	0	221	56	12	6
six	21	0	0	6	0	3	270	0	0
three	12	0	0	12	3	9	12	237	15
two	0	0	71	0	15	3	6	3	199

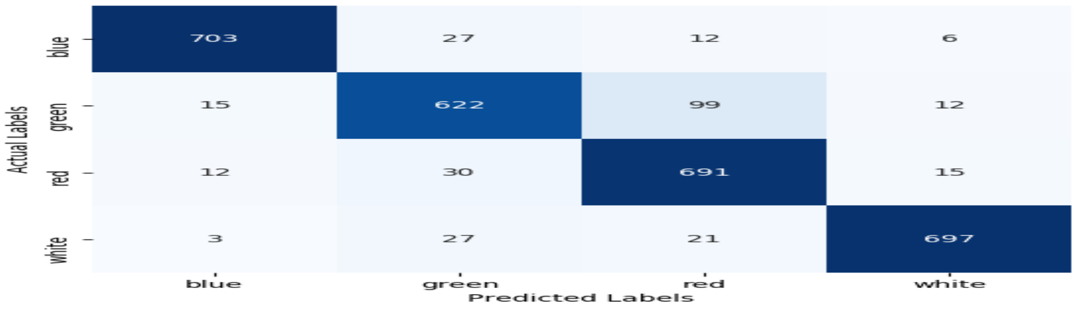
Numbers CNN Confusion Matrix



Alphabet CNN Confusion Matrix



Commands CNN Confusion Matrix

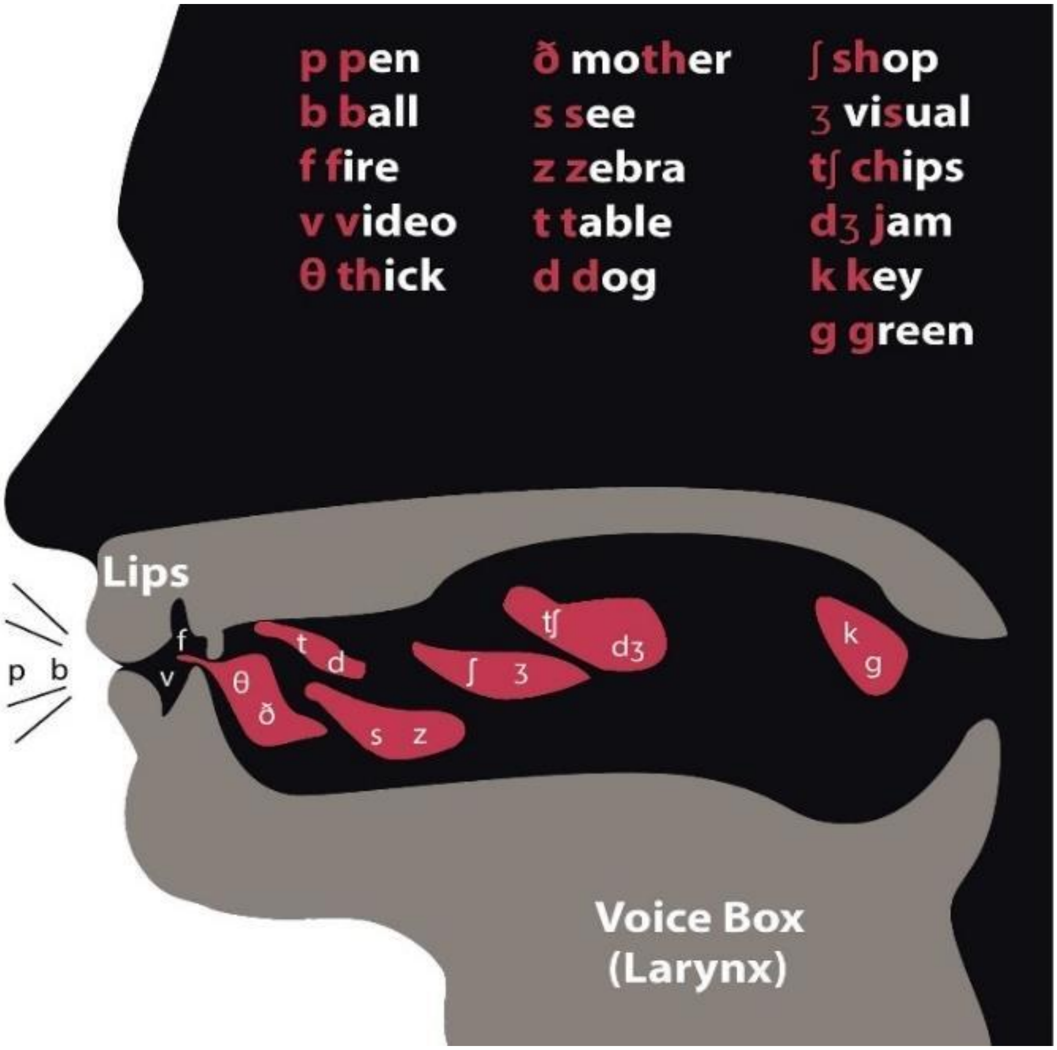


4 Color CNN Confusion Matrix

Figure : Confusion matrix of every part of sentences

The reason for such confusion is that the English alphabet letters don't just depend

on the lips' movement, they rely mainly on the tongueand airmovement .



6.2 Algorithms and Preprocessing

6.2.1 Algorithms

1. spatiotemporal convolutional neural network

we used Spatiotemporal Convolutional neural network it process video data by convolving across time

$$[\mathrm{stconv}(\mathbf{x}, \mathbf{w})]_{c'tij} = \sum_{c=1}^{C} \sum_{t'=1}^{k_t} \sum_{i'=1}^{k_w} \sum_{j'=1}^{k_h} w_{c'ct'i'j'} x_{c,t+t',i+i',j+j'} .$$

2. recurrent neural network

we used Gated Recurrent Unit (GRU) , a type of recurrent neural network (RNN) that improves upon earlier RNNs by adding cells and gates for propagating information over more timesteps and learning to control this information flow. It is similar to the Long Short-Term Memory (LSTM)

$$\begin{aligned}
[\mathbf{u}_t, \mathbf{r}_t]^T &= \mathrm{sigm}(\mathbf{W}_z \mathbf{z}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{b}_g) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{U}_z \mathbf{z}_t + \mathbf{U}_h (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \\
\mathbf{h}_t &= (\mathbf{1} - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \tilde{\mathbf{h}}_t
\end{aligned}$$

3. connectionist temporal classification loss

s designed for tasks where we need alignment between sequences, but where that alignment is difficult - e.g. aligning each character to its location in an audio file. It calculates a loss between a continuous (unsegmented) time series and a target sequence. It does this by summing over the probability of possible alignments of input to target, producing a loss value which is differentiable with respect to each input node. The alignment of input to target is assumed to be “many-to-one”, which limits the length of the target sequence such that it must be ≤ the input length.

4. detect and predict face lip region with dlib

The GRID corpus consists of 34 subjects, each narrating 1000 sentences We also use a sentence-level variant of the split (overlapped speakers) similar to Wand et al. The videos were processed with the DLib face detector, and the iBug face landmark predictor with 68 landmarks coupled with an online Kalman Filter. Using these landmarks, we apply an affine transformation to extract a mouth-centred crop of size 100 × 50 pixels per frame. We standardise the RGB channels over the whole training set to have zero mean and unit variance

5. saliency visualisation techniques

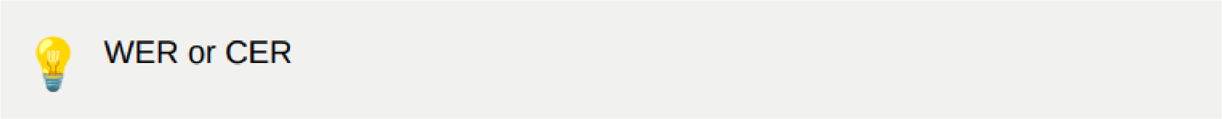
showing that the model attends to phonologically important regions in the video. Furthermore, by computing intra-viseme and interviseme confusion matrices at the phoneme level, we show that almost all of LipNet’s few erroneous predictions occur within visemes, since context is sometimes insufficient for disambiguation. using it showing that the model attends to phonologically important regions in the video. saliency Map for word Please

6.2.2 Preprocessing

The GRID corpus consists of 34 subjects, each narrating 1000 sentences We also use a sentence-level variant of the split (overlapped speakers) similar to Wand et al. The videos were processed with the DLib face detector, and the iBug face landmark predictor with 68 landmarks coupled with an online Kalman Filter. Using these landmarks, we apply an affine transformation to extract a mouth-centred crop of size 100 × 50 pixels per frame. We standardise the RGB channels over the whole training set to have zero mean and unit variance

Performance

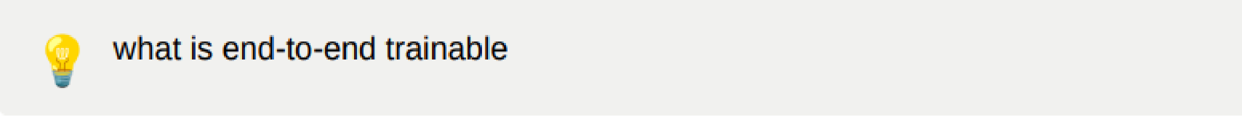
Compute Word Error Rate (WER) , Character Error Rate(CER)



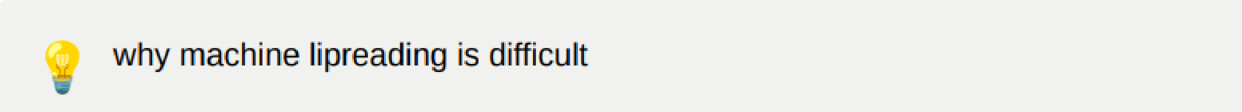
WER or CER

the minimum number of word (or character) insertions, substitutions, and deletions required to transform the prediction into the ground truth **divide by** the number of words (or characters) in the ground truth.

Model Architecture



what is end-to-end trainable

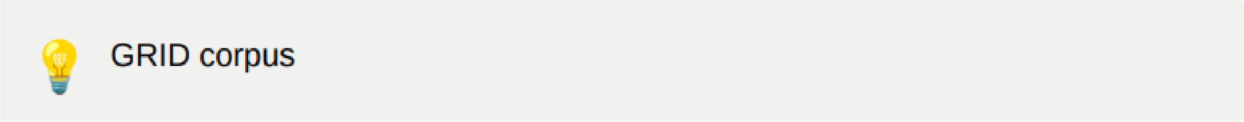


why machine lipreading is difficult

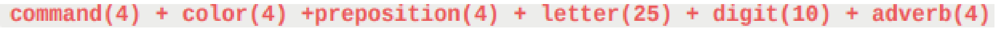
it requires extracting spatiotemporal features from the video (since both position and motion are important). Recent deep learning approaches attempt to extract those features end-to-end. Most existing work, however, performs only word classification, not sentence-level sequence prediction it based on automatic speed recognition our model is operating by

- spatiotemporal convolutional neural network
- recurrent neural network
- connectionist temporal classification loss

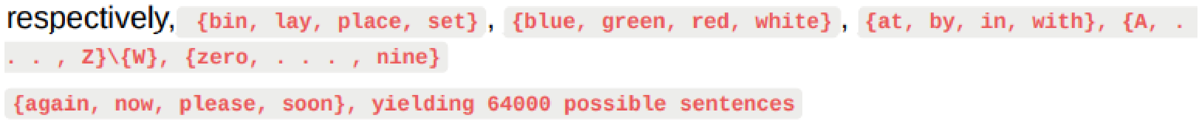
applying saliency visualisation techniques showing that the model attends to phonologically important regions in the video. Furthermore, by computing intra-viseme and interviseme confusion matrices at the phoneme level, we show that almost all of LipNet's few erroneous predictions occur within visemes, since context is sometimes insufficient for disambiguation.


💡 GRID corpus

which has audio and video recordings of 34 speakers who produced 1000 sentences each, for a total of 28 hours across 34000 sentence We use the GRID corpus to evaluate LipNet because it is sentence-level and has the most data. The sentences are drawn from the following simple grammar:


`command(4) + color(4) +preposition(4) + letter(25) + digit(10) + adverb(4)`

, where the number denotes how many word choices there are for each of the 6 word categories. The categories consist of,


respectively, `{bin, lay, place, set}`, `{blue, green, red, white}`, `{at, by, in, with}, {A, . . . , Z}\{W}, {zero, . . . , nine}` `{again, now, please, soon}, yielding 64000 possible sentences`

in CNN

we used Spatiotemporal Convolutional neural network it process video data by convolving across time

$$[\text{stconv}(\mathbf{x},\mathbf{w})]_{c'tij} = \sum_{c=1}^{C}\sum_{t'=1}^{k_t}\sum_{i'=1}^{k_w}\sum_{j'=1}^{k_h} w_{c'ct'i'j'}x_{c,t+t',i+i',j+j'}.$$

in RNN

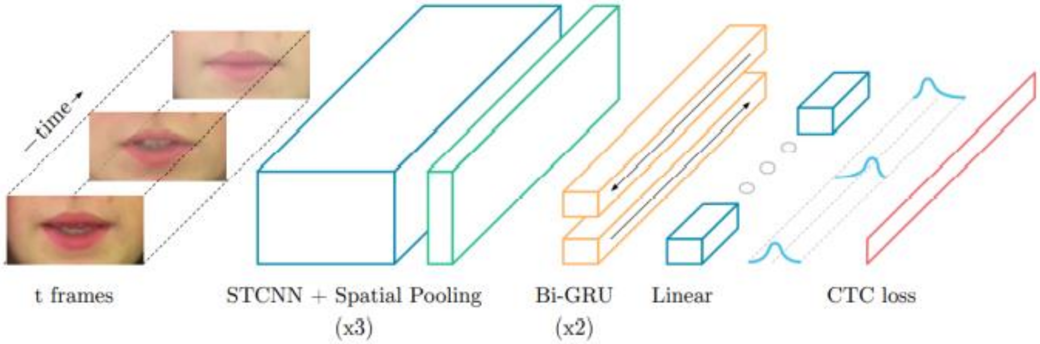
we used Gated Recurrent Unit (GRU) , a type of recurrent neural network (RNN) that improves upon earlier RNNs by adding cells and gates for propagating information over more timesteps and learning to control this information flow. It is similar to the Long Short-Term Memory (LSTM)

$$[\mathbf{u}_t, \mathbf{r}_t]^T = \mathrm{sigm}(\mathbf{W}_z \mathbf{z}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{b}_g)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{U}_z \mathbf{z}_t + \mathbf{U}_h (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h)$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \tilde{\mathbf{h}}_t$$

Model Architecture



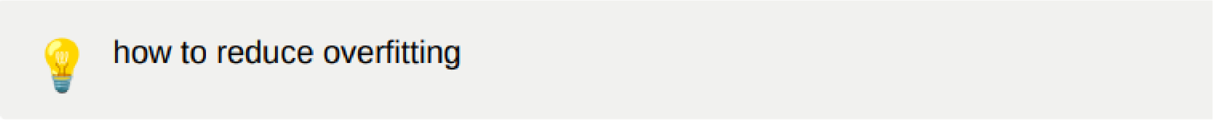
which starts with 3×(spatiotemporal convolutions, channel-wise dropout, spatial maxpooling). Subsequently, the features extracted are followed by two Bi-GRUs. The BiGRUs are crucial for efficient further aggregation of the STCNN output. Finally, a linear transformation is applied at each time-step, followed by a softmax over the vocabulary augmented with the CTC blank, and then the CTC loss. All layers use rectified linear unit (ReLU) activation functions.

Preprocessing :

we employ a split unseen speaker holding out the data of 1 , 2 , 20 , 22 for evaluation and reminder for training

the video were processed with Dlib face detector and IBug face landmark predictor with 68 landmarks coupled with an online kalman filter

we extract a mouth-centred crop of size 100 *50 pixel per frames then standardise RGB channels over training set to have zero mean and unit variance

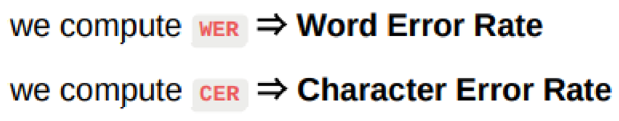


💡 how to reduce overfitting

we use Augmentation

- training on both regular and horizontally mirrored image sequence
- dataset provides word start and word end timing for each sentence video

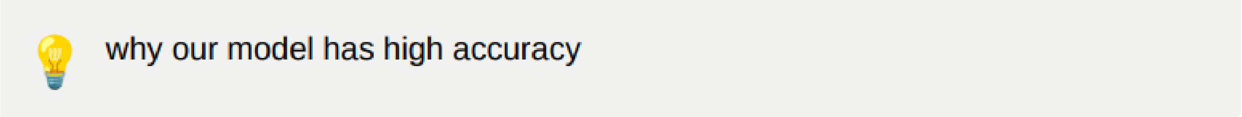
performance



we compute `WER` ⇒ **Word Error Rate**

we compute `CER` ⇒ **Character Error Rate**

it is minimum number of word or character insertion , substitution , deletions required to transform the prediction into ground truth **divide by** the number of word or character in the ground truth



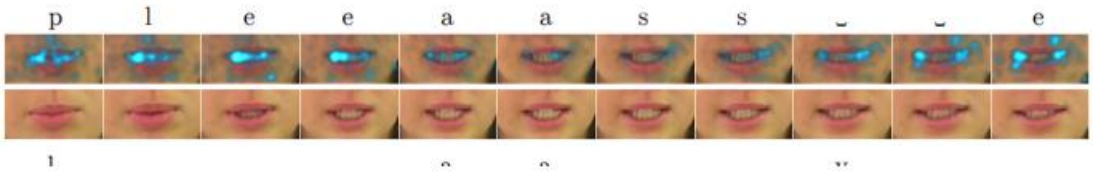
💡 why our model has high accuracy

because of extracting spatiotemporal features using STCNN is better than spatial-only features and using of CTC and RNN allow processing both variable-length input and variablelength output sequence.

Saliency Maps

using it showing that the model attends to phonologically important regions in the video**.**

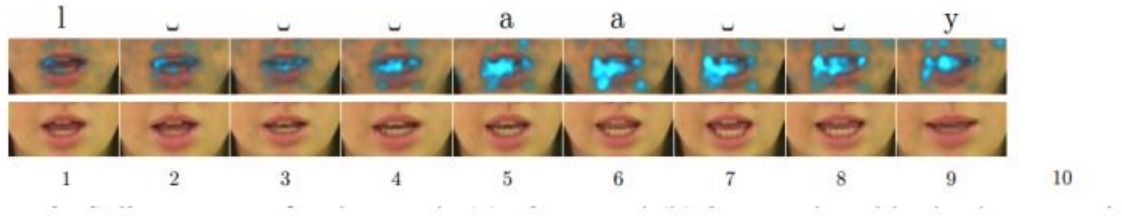
saliency Map for word Please



- the lips are pressed firmly together for the bilabial plosive /p/ (frame 1)

- allowing the compressed air to escape between the lips
- The jaw and lips then open further, seen in the distance between the midpoints of the upper and lower lips, and the lips spread (increasing the distance between the corners of the mouth), for the close vowel

saliency Map for word lay



6.3 Sample and Test Scenario

6.3.1 Sample Scenario

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-toend system specifications.

Date	Module	Action	Input	Expected output	Actual output	Test result	Test comments
2022-may - 15	Open model from pc and running the model on it	Passing the video to the model from the local pc	Video	The sentence from the video	The sentence from the video depending on our architecture but was wrong	Fail	Model Works but get wrong results because of the length of the video was large
2022 may - 20	Open model from pc and running the model on it	Passing the video to the model from the local pc	Video	The sentence from the video	The sentence from the video depending on our architecture but was wrong	Fail	The model crashed in the preprocessing phase because of the dimentional of the video was large
2022 – may -28	Open model from pc and running the model on it	Passing the video to the model from the local pc	Video	The sentence from the video	The sentence from the video depending on our architecture	pass	Model works
2022 – jun -4	Flutter application running local from the pc directly	Open the application and choice the video from the gallery	Video	The sentence from the video	The sentence from the video depending on our architecture	pass	Model works

2022 – jun - 10	Flutter application , fastapi	Open the application and choice the video from the gallery	Video	The sentence from the video	The sentence from the video depending on our architecture	pass	Model works

6.3.2 Test Scenario

Test scenario for word prediction:
Test scenario 1: test WI-FI connection.
test scenario 2: test video
input: video.
output: patient's data.
test scenario 3: test if found noisy enviroment.
Output: API will call the caregiver or the notification that will be sent to the caregiver's application version.

6.4 Main Work

we can devide our main work into some categories such as

1- learning android developing basics

- before we developing our application we need to read about andeoid specifically flutter developing

- because our knoledge about about android development was not sufficient enough to start developing the project

2- learning machine learning and deep learning techniques

- before developing our machine learning model we need to read and know about machine learning different techniques

because also out knoledge about machine learning techniques was not sufficient and not enough then applaying the machine learning to build our model with "face-detector , CNN , RNN"

3- learning about api to make it easy to send the video using internet from any where

6.5 Application and Language Used

6.5.1 The program we are using

1- Android studio

2- Visual Studio

3- Anaconda

4- jupyter notebook

5- spyder

6- google colab

7- flutter framework

6.5.2 languages we use

1- dart

2- python 3.7.0

6.7 What We Seek

- we seek to help all people who suffer from lossing their voices or impired people who lost there listining ability

Researchers estimate approximately 3% to 9% of people in the World deal with aphonia.

But some healthcare providers think the actual number is higher,

as many people do not seek medical help when they lose their voice.

- we seek to make our model helping in video quality by getting the said words without need to listining to the speaker voice or if voice has crashed for any reason like be in a noise environments

- generalize our application to use in the public places to make it easy to communicate withut producing sound only by the lip movement.

Chapter 7

Conclusion and Future work

7.1 Conclusion

Over the past twenty years, almost every article was dedicated to illustrating the benefit of using visual speech information from a speaker’s mouth in addition to the audio signal for the task of speech recognition. Even though all these works have shown that including the visual channel to the speech recognition system greatly improves its performance, no serious attempts were made into creating a sound independent system

In conclusion, the problems facing people with hearing disabilities are numerous and with the current waves of public awareness, we utilized our education to provide them with much needed help by reducing the communication gaps. Of course, the gap is still present as our vocabulary doesn’t contain all needed nouns/verbs but It is a monumental stride in a promising direction. Users can currently utter over than 20 Million different possible sentences using our vocabulary

Our proposed System TOS (Text of silence) is a mobile application to apply deep learning to end-to-end learning of a model that maps sequences of image frames of a speaker’s mouth to entire sentences. Thus, eliminates the need to segment videos into words before predicting a sentence. Furthermore, TOS greatly outperforms a human lipreading baseline

Our System allowed us to effectively use visual information for speach recognition and can be used for lip-reading (when audio is not available) or when audio is noisy.

7.2 Future work

While TOS is already an empirical success, the deep speech recognition literature suggests that performance will only improve with more data. In future work, we hope to demonstrate this by applying TOS to larger datasets, such as a LRS and LRW datasets provided by Oxford University.

A serious limitation in current audio-visual speech recognition videos is the lack of human pose/viewpoint estimation, if a human's head is tilted in any direction or captured from a side view then it won't be recognized as valid and consequently cannot be identified. In order to overcome these limitations a heavy deal of pre-processing for the videos is needed to accurately detect and predict mouth regions if they are obscured in any way.

In addition to Improving the Mobile application so it has the capability to Capture and Process videos in realtime

References

[1]Yuxuan Lan1, Richard Harvey1, Barry-John Theobald1, Eng-Jon Ong2 and Richard Bowden2 http://www2.cmp.uea.ac.uk/~bjt/avsp2009/proc/papers/paper-35.pdf

[2] McGurk and J. MacDonald. Hearing lips and seeing voices. Nature, 264:746–748, 1976. http://wexler.free.fr/library/files/mcgurk%20%281976%29%20hearing%20lips%20and%20seeing%20voices.pdf

[3] G. Fisher. Confusions among visually perceived consonants. Journal of Speech, Language, and Hearing Research, 11(4):796–804, 1968.

[4] Yannis M. Assael1,† , Brendan Shillingford1,† , Shimon Whiteson1 & Nando de Freitas1,2,3 Department of Computer Science, University of Oxford, Oxford, UK 1 Google DeepMind, London, UK 2 CIFAR, Canada 3 https://arxiv.org/pdf/1611.01599.pdf

[5]IEEE : Chung, J. S., Senior, A., Vinyals, O., & Zisserman, A.(2017). Lip Reading Sentences in the Wild. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.367 https://ieeexplore.ieee.org/abstract/document/8099850

[6] British Deaf Association (2015, Sept 7) Fast facts about the Deaf community. Last Accessed 20 March 2020:

https://bda.org.uk/fast-facts-about-the-deaf-community/

[7] IMPROVED SPEAKER INDEPENDENT LIP READING USING SPEAKER ADAPTIVE TRAINING AND DEEP NEURAL NETWORKS for Ibrahim Almajai https://sci-hub.se/10.1109/ICASSP.2016.7472172

[8] Out of Time: Automated Lip Sync in the Wild for Joon Son Chung and Andrew Zisserman https://link.springer.com/chapter/10.1007%2F978-3-319-54427-4_19

[9] 1.Lip Reading Sentences in the Wild https://sci-hub.se/10.1109/CVPR.2017.367

[10] 2. Combining Residual Networks with LSTMs for Lipreading Themos Stafylakis, Georgios Tzimiropoulos https://www.isca-speech.org/archive/pdfs/interspeech_2017/stafylakis17_interspeech.pdf

[11] 3. Learning to lip read words by watching videos Joon Son Chung∗ , Andrew Zisserman https://sci-hub.se/https://doi.org/10.1016/j.cviu.2018.02.001

[12] Deep Learning of Mouth Shapes for Sign Language https://sci-hub.se/10.1109/ICCVW.2015.69

[15] Computer Vision Lip Reading Grace Tilton, http://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26646023.pdf