.

PAKISTANI SIGN LANGUAGE RECOGNITION

Project ID: Winter-2021-33

Session: BSCS Winter 2018 to 2022

Project Advisor: Dr. Yasir Niaz

Submitted By

Ammar Sarwar	70069491
Abdullah	70067787
Haider Ali	70067920

Department of Computer Science & IT
The University of Lahore
Lahore, Pakistan

Declaration

We have read the project guidelines and we understand the meaning of academic dishonesty, in particular plagiarism and collusion. We hereby declare that the work we submitted for our final year project, entitled **PAKISTANI SIGN LANGUAGE RECOGNITION** is original work and has not been printed, published or submitted before as final year project, research work, publication or any other documentation.

Group Member 1 Name: Ammar Sarwar SAP No: 70069491

Signature:…………………………

Group Member 2 Name: Abdullah SAP No: 70067787

Signature:…………………………

Group Member 3 Name: Haider Ali SAP No: 70067920

Signature:…………………………

Statement of Submission

This is to certify that **Ammar Sarwar 70069491**. **Abdullah 70067787** and **Haider Ali 70067920** have successfully submitted the final project named as: **PAKISTANI SIGN LANGUAGE RECOGNITION**, at Computer Science & IT Department, The University of Lahore, Lahore Pakistan, to fulfill the partial requirement of the degree of **BS in Computer Science**.

Supervisor Name: ………………………

Signature: ………………………… Date: ………………………

Dedication

This project is dedicated to our supervisor, who taught us that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to our family, who taught us that even the largest task can be accomplished if it is done one step at a time.

.

Acknowledgement

We truly acknowledge the cooperation and help by **Dr Yasir Niaz Khan**, **Professor of University of Lahore**. He has been a constant source of guidance throughout the course of this project. We would also like to thank **Mr. Ali Ahmed, Lecturer of University of Lahore** for his help and guidance throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

Date:

December 24,2021

.

Abstract

Correspondence is an essential human need and language is the vehicle for this. A great many people can tune in and talk and they utilize various dialects to impart. Hearing impeded individuals use signs to speak with others. Pakistan Communication via gestures (PSL) is the favored language of the hard of hearing individuals in Pakistan. To make the correspondence among hard of hearing and ordinary individuals more straightforward and simpler, PC based PSL translators are required.

Area of the Project

Machine learning, image processing, web application, natural language processing, artificial intelligence

Technologies used

HTML, CSS, JavaScript, .NET, Php, Xampp, MySQL, python, Open-Cv, Open-Pose, ANN etc.

List of Figures

.

List of Tables

.

Table of Contents

.

Chapter 1: Introduction to the Problem

1.1 INTRODUCTION

Sign Language (SL) is not commonly learned by non-mute people, thus, mute people have problems communicating with people. Usually, people do not learn it if there is no mute person in their relation circles or if it is not required for their job. When they engage with a mute person the communication can be hard and tedious. As an example, a mute individual goes to an interview: if the interviewer does not know SL the common approach is to hire a translator. This action creates some problems, as hiring can be expensive and scheduling an appointment with three people, depending on the circumstances is difficult.

In this project we try to develop a system to convert PSL into Urdu text and speech for communication between signers and non-signers. Main objective is to facilitate a large population of hearing-impaired persons and making them an integral part of the society. The system is divided into 3 main modules, detection module, learning module and capture module. The detection module has two modes one for alphabet detection and other for word detection. The learning module is used for interactive PSL learning and the capture module is used for adding new data to dataset.

The detection system works by taking images of user making particular sign through webcam and uses skeletal tracking combined with machine learning to detect what sign they are making.

The mechanized dataset catching framework works by catching pictures of client for determined time, removing the central issues of pictures utilizing Open-Posture, erasing the pictures where Open-Posture shows less certainty, plotting the skeleton of outstanding pictures and showing them to the client, the client then, at that point, chooses the plotted pictures that doesn't match the sign he was making and erases them. The client is then approached to enter the mark for residual pictures and these pictures are put away by the framework in an envelope named as the worth of name entered by client. The learning framework works by showing the client pictures of various PSL signs and inciting them to make that sign, the framework then, at that point, checks whether or not they have made the right sign and shows a message appropriately.

1.2 PURPOSE

The purpose of this project is to develop a sign language recognition application which would address the issues of mute people and non-mute people in regards of their communication. A mute person would communicate with a non-mute person using signs and the system would convert those signs into words, which would end up facilitating both mute and non-mute speaker.

An observable measure of hard of hearing local area exists in Pakistan. In a nitty gritty review in light of factual information in regards to the hard of hearing individuals in Pakistan [1][2], It is expressed that around 3.3 million Pakistanis are experiencing various types of incapacity in which 0.24 million are weakened of hearing which approximates to 7.4% of by and large incapacitates. A vital point is that 55% of the absolute incapacitated lie in the age bunch from 5 - 29 years. At the point when hard of hearing individuals need to speak with healthy individuals in any circumstance, they either need to enlist an interpreter or solicitation the healthy individuals to convey through composition. The current interpreters are undeniably challenging to get constantly and they are pricey too. The collaboration of hard of hearing with typical individuals turns out to be exceptionally sluggish and humiliating. In such manner, hard of hearing individuals observe it challenging to would how they like to treat shopping centers, banks and mailing stations and so forth In addition, hard of hearing individuals in Pakistan for the most part wind up having companions that are additionally hard of hearing. There is most certainly a requirement for a robotized framework that can work with correspondence among hard of hearing and healthy individuals

1.3 OBJECTIVE

The aim of the project is to bridge the communication gap between signers and non-signers by developing an interpretation system which is cost effective and easy to use. The system is expected to have the ability to convert different PSL signs into text and speech.

The main objectives of this system are:

- Developing a system which will convert gestures into words and speech
- Conversion of PSL signs into letters and speech
- Conversion of PSL signs into words forming a stream of sentence

1.4 PROBLEM STATEMENT

Sign language is a nonverbal type of specialized strategy which is found among all hard of hearing people. Ordinary individuals don't get familiar with the communication via gestures. It causes hindrance in correspondence between hard of hearing moronic and ordinary individuals.

1.5 EXISTING SOLUTION

Research in the sign language system has two well-known approaches, image processing and data Glove. Worldwide efforts have been made to aid the deaf community in communicating with non-signers. Some of the known solutions are:

- A research-based development of a sign language recognition system using a data glove is proposed by Nicholas Born [4]. In this approach, detection of hand is done by the sensor glove that consists of flex sensor and accelerometer.
- Most recently Muhammad Wasim et al [5] used leap motion sensor to develop a system to communicate in sign language. They were able to detect some signs with an accuracy level of up to 100%.

- Another approach used by A. Muhammad at el [3] is to track skeleton using Kinect sensor. The Accuracy of system is not known.
- Another methodology utilizing information glove however this time the shaded one, is proposed by Sumaira et al [6]. They have utilized a fluffy classifier to perceive the signs/signals performed by the hard of hearing. Their calculation involves the point among fingertip and joint for arrangement of signal. Their dataset depended on Urdu letters in order of Pakistan gesture based communication. They have accomplished a general exactness of

95% as 35 out of 37 letter sets were perceived accurately. Once more, the

- cost of shading information glove, acknowledgment of single hand motions and static motions are the restrictions of their framework.

- ● One more work for decreasing correspondence hole between the hard of hearing and hearing is finished by Asif Ali [7]. He has proposed a framework which takes input in the two structures text and picture of sign and convert it into different structures. They have played out this for Urdu letter sets of PSL utilizing Haar classifier. They have involved a basic RGB camera for this reason. Be that as it may, they have not explicitly referenced the size and nature of the dataset utilized for trial and error. The exactness pace of the created framework is additionally absent.

The main drawback of most of the above-mentioned systems is the use of special equipment such as gloves, leap motion sensor or depth sensor (i.e. Microsoft Kinect), which are expensive and usually not available in real life scenarios. Another problem is that these sensors need a closed environment such as the Kinect sensor cannot work properly in sunlight

1.6 PROPOSED SOLUTION

We propose a PSL recognition system based on the human skeletal key points that are estimated by third party library Open-Pose. The Open-Pose is an open source toolkit for real-time multi-person key point detection it can estimate in total 130 key points where 18 key points are from body, 21 key points are from each hand, and 70 key points from a face.

Currently there is no publicly available dataset for PSL, especially for use with Open-Pose. In case of sign language dataset, it is more difficult to collect than the other dataset because, for the accuracy of the data, professional signers should be used for recording sign language videos of high quality, which are hard to find. So, we have decided to develop our own dataset for PSL words and alphabets. Different machine learning algorithms are used for recognizing signs and are analyzed for accuracy and performance. Machine Learning libraries like scikit-learn, TensorFlow and keras are also used.

Our recognition system is robust in different backgrounds as it only detects the human body. The system based on the human key point detection works well regardless of signer since the variance of extracted key points is negligible. Moreover, we normalize the feature set using the mean and standard deviation. Lastly, the use of high-level features is necessary when the scale of the dataset is not large enough.

Chapter 2: Software Requirement Specification

2.1 INTRODUCTION

2.1.1 PURPOSE

The reason for this record is to introduce an itemized depiction of Pakistan sign language communication. It will clarify the reason and highlights of the framework, the connection points of the framework, how the framework will treat the limitations under which it should work. This record is expected for both the partners and the engineers of the framework. Intended Audience can be:

- Developers: Developers should have idea about what they are going to develop because they want to develop the project according to the requirements
- Researchers: Researchers can use this document to research about the object recognition and tracking of different objects.
- Testers: Testers are the one who will use this document to test the requirements and to check the project has meet the requirements or not.

2.1.2 SCOPE

This product framework is a work area application for changing over Pakistan Communication through signing into text and discourse. This framework will be utilized by hearing weakened individuals to speak with typical hearing individuals, with next to no particular gear. By limiting the expense and accessibility issues and augmenting the result, the framework addresses clients issue while staying straightforward and use. Pakistani communication via gestures acknowledgment programming would record motions through webcam either in genuine time(live) or a video and would change over those signals into words and sound for non-quiet individuals. This application is intended to be utilized by confused individuals to facilitate their life. The advantage is that it will change over the motions done by a quiet individual into words so a non-quiet individual could comprehend.

2.1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

TABLE 2.1 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Term/Abbreviation	Description
OPEN-POSE	This is a convolutional neural network key point detector of human body. [8]
PYTHON	“Python is an interpreted high-level language”

.

TensorFlow	“TensorFlow [13] is a point of interaction for communicating AI algo, and an execution for executing such algo”.
Eel	“Eel is a product library that permits the advancement of straightforward work area GUI like applications utilizing front-end parts of web applications and back-end in Python/JavaScript”.
SVM	Support Vector Machine “officially characterized by an isolating hyperplane. As such, given marked preparation information (administered learning), the algorithm yields an ideal hyperplane which orders new models”.[9]
KNN	K-Nearest Neighbors “that stores generally accessible cases and classifies new cases in light of a similitude measure (e.g., distance capacities)”.[10][11]
ANN	Artificial Neural Networks “ANN or Neural Networks are computational algorithms. It expected to reproduce the conduct of natural frameworks made out of "neurons". ANNs are computational models motivated by a creature's focal sensory systems. It is fit for AI as well as example acknowledgment”.[12]
PSL	Pakistan Sign Language.

RNN	Recurrent Neural Network RNN is a class of counterfeit neural organization where associations between hubs structure a coordinated diagram along a grouping. This permits it to display worldly unique conduct for a period arrangement. Not at all like feedforward neural organizations, RNNs can utilize their interior state (memory) to handle successions of data sources.
SD	Standard Deviation SD is an amount communicating by how much the individuals from a gathering contrast from the mean incentive for the gathering.
JSON	JavaScript Object Notation JSON is a lightweight format for storing and transporting data.

2.2 OVERALL DESCRIPTION

The subsequent area, the General Depiction part of this report, gives an outline of the usefulness of the item and portrays the casual necessities of the item.

The third area, Prerequisites Detail segment of this archive depicts the functionalities of the item in specialized terms.

The two areas of the archive portray similar programming item completely, yet are planned for various crowds.

2.2.1 PRODUCT PERSPECTIVE

The proposed application is based on the human skeletal key points that are estimated by third party library Open-Pose. The Open-Pose is an open-source toolkit for real-time multi-person key point detection it can estimate in total 130 key points. The framework takes pictures of client making specific sign through webcam and utilizations skeletal following joined with AI to distinguish what sign they are making.

.

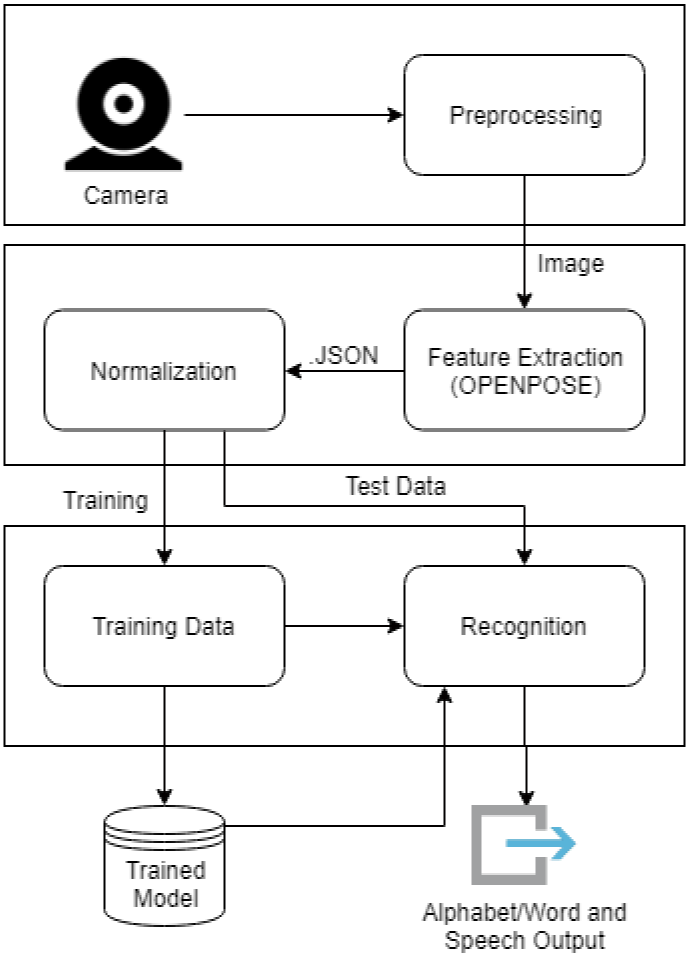
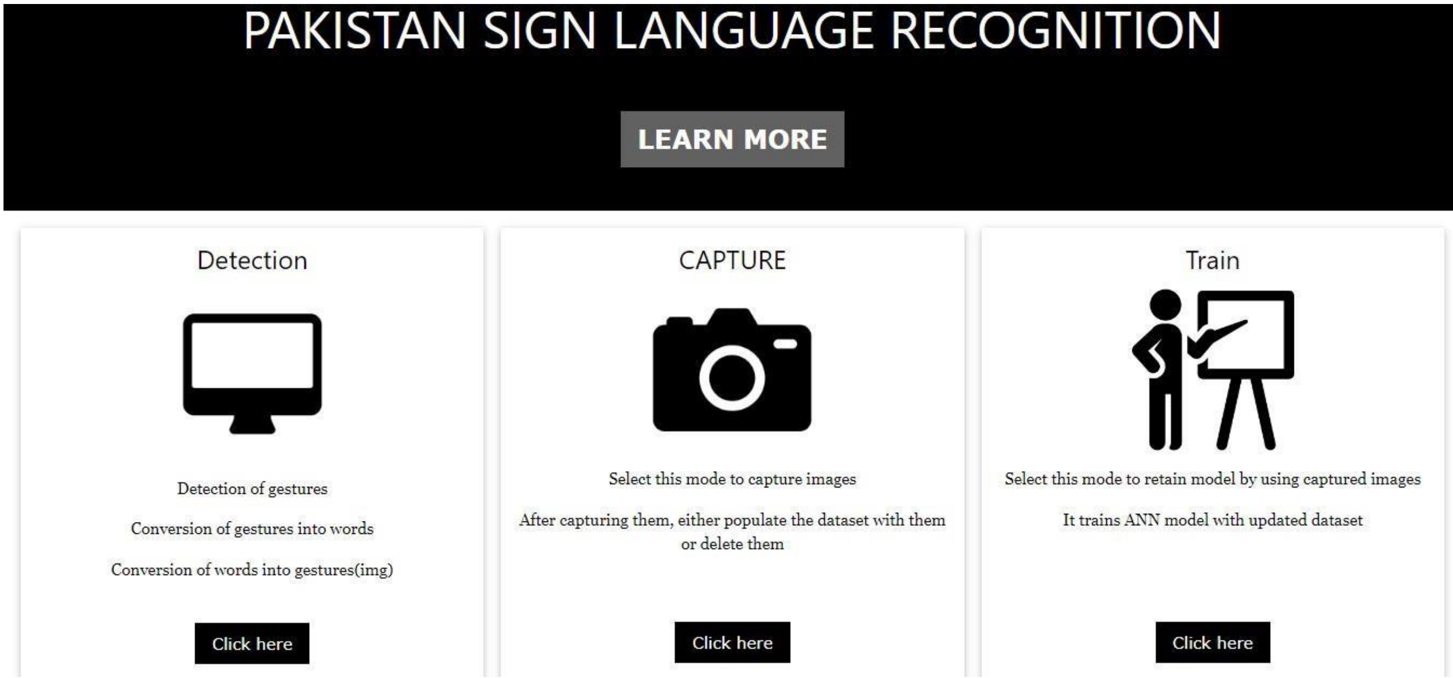


Figure 2.1. Product Perspective

This subsection is sub-divided into the following

- **System interfaces:** System interface includes gesture and their conversion into words and audio, and conversion of words into photo which would contain gestures.
- **User interfaces** This project can serve both speech-disordered people and the others. Yet it is designed especially for the speech-disordered people's convenient use. The system is divided into 3 main modules, detection module, learning module and capture module. The detection module has two modes one for alphabet detection and other for word detection. The learning module is used for interactive PSL learning and the capture module is used for adding new data to dataset.

MAIN SCREEN:-



PAKISTAN SIGN LANGUAGE RECOGNITION

LEARN MORE

Detection

Detection of gestures

Conversion of gestures into words

Conversion of words into gestures(img)

Click here

CAPTURE

Select this mode to capture images

After capturing them, either populate the dataset with them or delete them

Click here

Train

Select this mode to retain model by using captured images

It trains ANN model with updated dataset

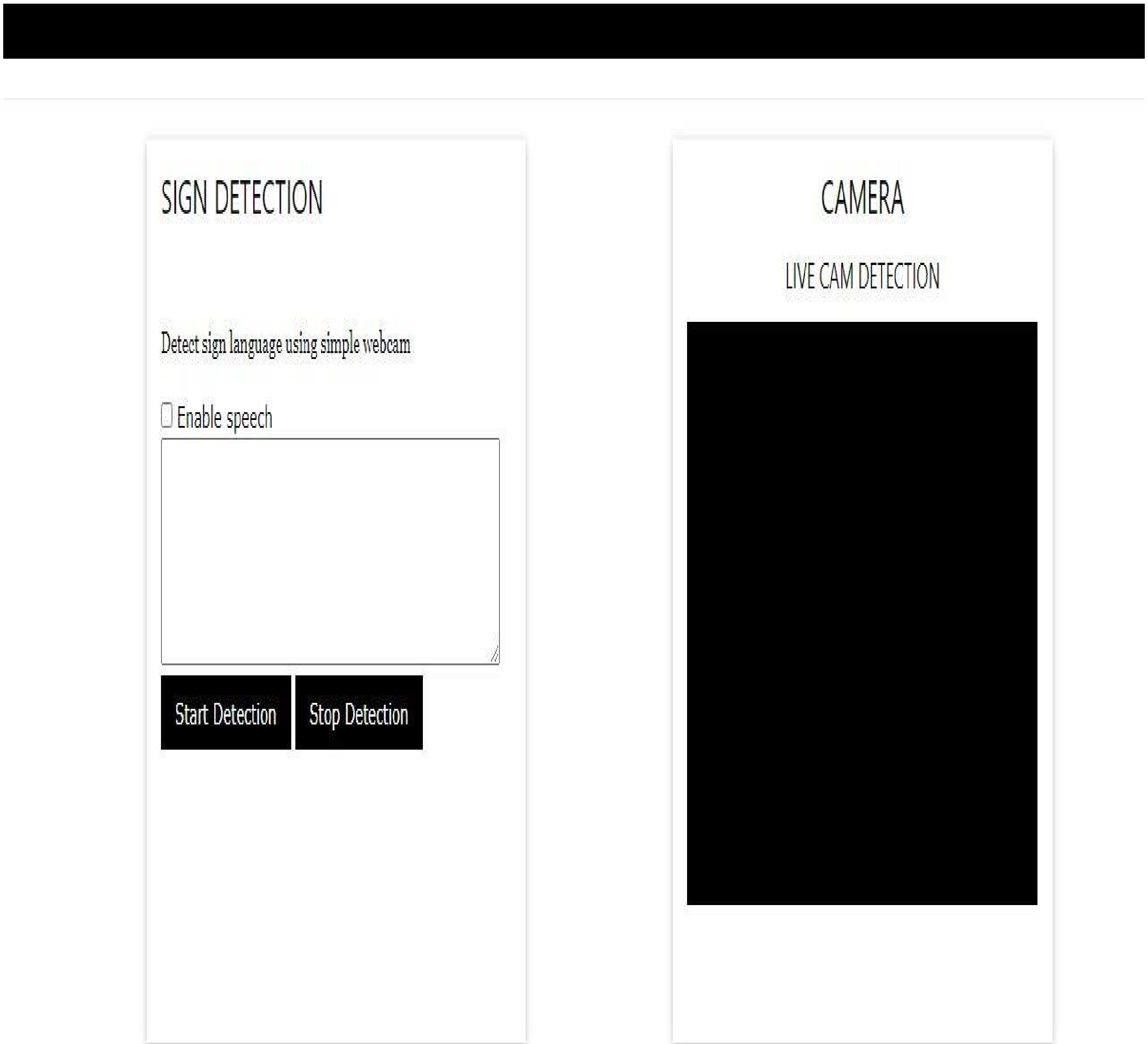
Click here

.

Figure 2.2. Main Screen

.

DETECTION SCREEN:-



SIGN DETECTION

Detect sign language using simple webcam

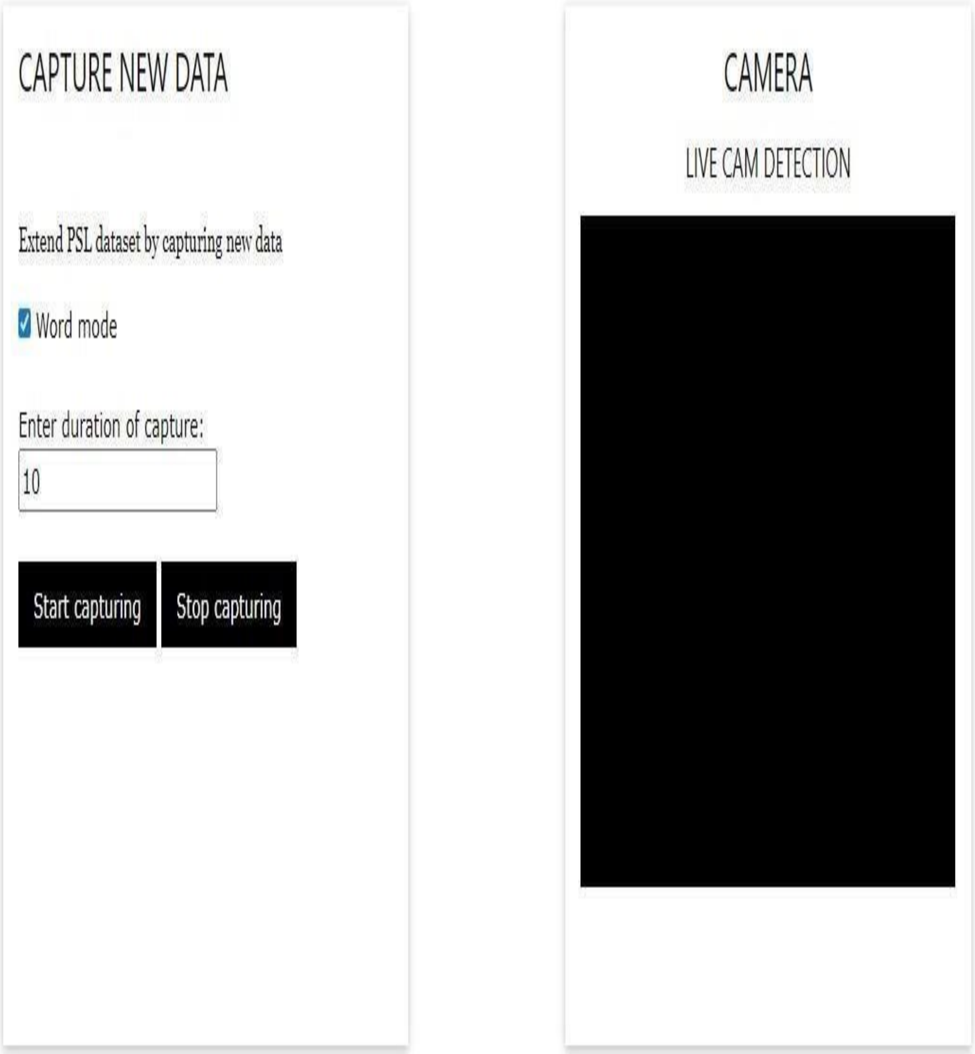
□ Enable speech

CAMERA

LIVE CAM DETECTION

Figure 2.3. Detection screen

CAPTURE SCREEN:-



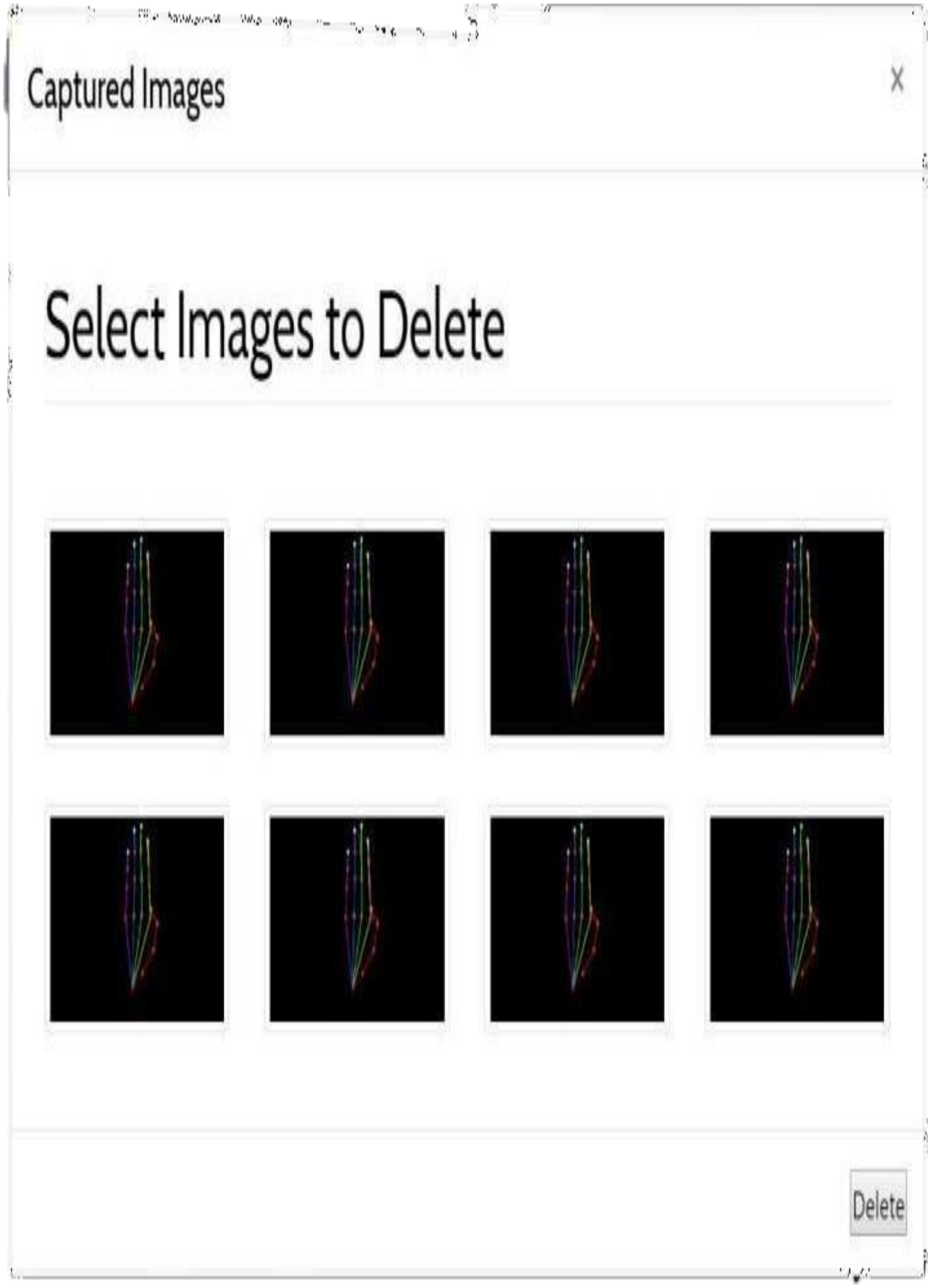
Extend PSL dataset by capturing new data

☑ Word mode

Enter duration of capture: 10

Figure 2.4. Capture Screen

.

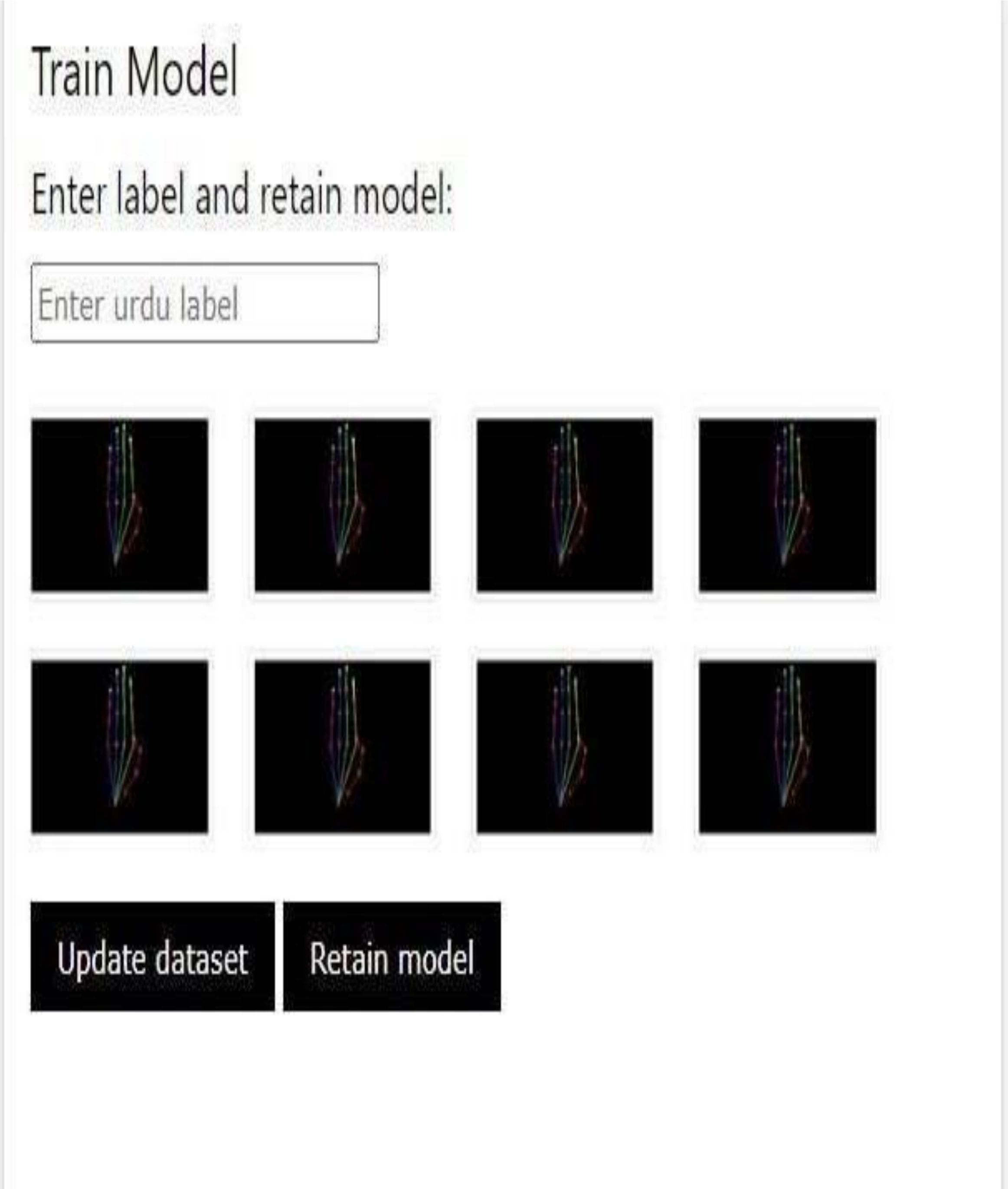


Captured Images

Select Images to Delete

Figure 2.5. Delete Image

TRAIN SCREEN:-



Train Model

Enter label and retain model:

Figure 2.6. Retrain model

.

- **Hardware interfaces:** The hardware components required for system to work properly are, a working web cam, a computer with good specs

- **Software interfaces:** The software will operate in Windows 10 environment. Our project uses ML python and various python libraries such as mediapipe, OpenCv, Open-Pose etc
- **Communications interfaces** There are no communications interfaces for the system.
- **Operations:** Following are the operations performed through our application,
- Dataset collection in the device
- Creating a training data in the device
- Using training data for the classification
- Conversion of gestures into words and speech
- Conversion of words into gestures

- **Site adaptation requirements:** No special site adaptions implemented neither required for this application.

- **User Placement:** User will sit accurately in front of the camera according to the range of camera.

2.2.2 PRODUCT FUNCTIONS

The main purpose of our application is to aid communication between signer and non-signer. The user will perform a sign in front of webcam, sign will be recognized by the system and system will output it in the form of text and speech

TABLE 2.2 START DETECTION

ID:	FR_01			
Name:	Start Detection			
Description	Input	Output	Requirements	Basic Workflow

Start recognizing signs	Click Start Detection button	Recognition Started	Recognition Stopped	Click Start Detection button to start recognizing signs

TABLE 2.3 RECOGNIZE SIGN

ID:	FR_02			
Name:	Recognize Sign			
Description	Input	Output	Requirements	Basic Workflow
Recognize specific sign made by user and convert it into text and speech	Valid sign in front of the webcam	Correct conversion into text and speech	Recognition has started, Valid sign made by user in front of webcam.	User Signs in front of camera, the system recognizes the sign and convert it into text and speech.

TABLE 2.4 STOP DETECTION

ID:	FR_03
Name:	Stop Detection

.

Description	Input	Output	Requirements	Basic Workflow
Stop recognizing signs	Click Stop Detection button	Recognition Stopped	Recognition Started	Click Stop Detection button to stop recognizing signs

TABLE 2.5 ENABLE SPEECH

ID:	FR_04			
Name:	Enable Speech			
Description	Input	Output	Requirements	Basic Workflow
Enable speech output	check the ‘enable speech’ checkbox	speech output enabled	‘enable speech’ checkbox unchecked	check the ‘enable speech’ checkbox to Enable speech output

TABLE 2.6 DISABLE SPEECH

ID:	FR_05
Name:	Disable Speech

Description	Input	Output	Requirements	Basic Workflow
Disable speech output	uncheck the enable speech checkbox	speech output disabled	enable speech checkbox checked	uncheck the enable speech checkbox to disable speech output

TABLE 2.7 RECOGNITION OUTPUT

ID:	FR_06			
Name:	Display Recognition Output			
Description	Input	Output	Requirements	Basic Workflow
display recognized sign output on screen	Recognition Output	Output Displayed	Detection started; some output received from recognition module	System gets input from recognition module and displays it on screen

TABLE 2.8 SPEECH OUTPUT

ID:	FR_07			
Name:	Play Speech Output			
Description	Input	Output	Requirements	Basic Workflow

play speech of recognized sign	Recognition Output	speech played	speech enabled; some output received from recognition module	System gets input from recognition module and play the speech

TABLE 2.9 SWITCH TO WORD MODE

ID:	FR_08			
Name:	Switch to Word Mode			
Description	Input	Output	Requirements	Basic Workflow
Switch to word mode from alphabet mode	check the ‘switch to word mode’ checkbox	Switched to word mode from alphabet mode	‘switch to word mode’ checkbox unchecked	check the ‘switch to word mode’ checkbox to Switch to word mode from alphabet mode

TABLE 2.10 SWITCH TO ALPHABET MODE

ID:	FR_09			
Name:	Switch to Alphabet Mode			
Description	Input	Output	Requirements	Basic Workflow

Switch to alphabet mode from word mode	uncheck the ‘switch to word mode’ checkbox	Switched to alphabet mode from word mode	‘switch to word mode’ checkbox checked	uncheck the ‘switch to word mode’ checkbox to Switch to alphabet mode from word mode

TABLE 2.11 VIEW HELP

ID:	FR_10			
Name:	View Help			
Description	Input	Output	Requirements	Basic Workflow
View Help Document	Click Help Button	Help document shown in new page	no requirements	user clicks help button, and system shows help document

TABLE 2.12 DISPLAY SIGN IMAGE

ID:	FR_11			
Name:	Display Sign Image			
Description	Input	Output	Requirements	Basic Workflow

display image of specific sign on screen and prompt user to try making that sign	Learning Module	Image displayed successfully	no requirements	user opens learning module, and system displays image of specific sign on screen

TABLE 2.13 VALIDATE SIGN

ID:	FR_12			
Name:	Validate Sign			
Description	Input	Output	Requirements	Basic Workflow
Validate sign made by user against an image shown to him	webcam	Display success message	Valid sign made by user in front of webcam.	User Signs in front of camera, the system validates the sign and displays a success message

TABLE 2.14 CHANGE SIGN IMAGE

ID:	FR_13			
Name:	Change Sign Image			
Description	Input	Output	Requirements	Basic Workflow

Show next sign image to user if previous sign was validated	Validation function	Image Changed successfully	previous sign was validated	The system shows next sign image to user if previous sign was validated

TABLE 2.15 SKIP SIGN

ID:	FR_14			
Name:	Skip Sign			
Description	Input	Output	Requirements	Basic Workflow
Skip to next sign if user clicks skip button	Click 'Skip' Button	Image Changed successfully, moved to next sign	sign has not been validated	The system shows next sign image to user if user clicks 'skip' button

TABLE 2.16 SELECT CAPTURE MODE

ID:	FR_15			
Name:	Select Capture Mode			
Description	Input	Output	Requirements	Basic Workflow

Select either word mode or alphabet mode from ‘select mode’ dropdown	‘select mode’ dropdown	mode changed to selected dropdown value	Capture has not been started	select mode from the ‘select mode’ dropdown to select word mode or alphabet mode

TABLE 2.17 SET CAPTURE TIME

ID:	FR_16			
Name:	Set Capture Time			
Description	Input	Output	Requirements	Basic Workflow
Enter capture time in seconds, in ‘enter time’ text field to set time for capture process	‘enter time’ text field	time set successfully	Capture has not been started	User enters capture time in ‘enter time’ text field and system set time for capture process

TABLE 2.18 DISPLAY CAPTURED IMAGES

ID:	FR_17			
Name:	Display Captured Images			
Description	Input	Output	Requirements	Basic Workflow

Display images captured in capture process	images captured in capture process	images displayed on screen	some images received from capture process	the system displays images captured in capture process

TABLE 2.19 DELETE INVALID IMAGES

ID:	FR_18			
Name:	Delete Invalid Images			
Description	Input	Output	Requirements	Basic Workflow
Select images of invalid signs and delete them	selected images	images deleted successfully	some images selected to delete	user selects the images to be deleted and the system deletes them

TABLE 2.20 SET LABEL OF SIGN

ID:	FR_19			
Name:	Set Label of Sign			
Description	Input	Output	Requirements	Basic Workflow

enter label in ‘set label’ text field corresponding to captured sign and save captured files with that label	‘set label’ text field	captured files saved successfully with entered label	some images captured in capture process	user enters label in ‘set label’ text field and the system save captured files with that label

TABLE 2.21 POPULATE DATASET

ID:	FR_20			
Name:	Populate Dataset			
Description	Input	Output	Requirements	Basic Workflow
Click ‘populate dataset’ button to update dataset with new captured files	Click ‘populate dataset’ button	dataset updated with new captured files	some images captured incapture process	user clicks ‘populate dataset’ button and the system updates dataset with new captured files

TABLE 2.22 RETAIN MODEL

ID:	FR_21
Name:	Retrain Model

Description	Input	Output	Requirements	Basic Workflow
Click ‘Train Model’ button to train ANN model with updated dataset	Click ‘Train Model’ button	ANN model trained with updated dataset	no requirements	user clicks ‘Retrain Model’ button and the system trains ANN model with updated dataset

2.2.3 USER CHARACTERISTICS

TABLE 2.23 USER CHARACTERISTICS

Name	**Description**
Real-Time User	A person who uses the system to convert PSL into text and speech in real time.
Learner User	A person who wants to learn PSL.

2.2.4 CONSTRAINTS

The current constraints on the project are related to hardware. At present, a working webcam, Core i3, 4 GB RAM and Cuda compatible Nvidia graphics card is required

- **Hardware limitations**: include camera quality and system hardware specifications.
- **Parallel operation:** It can perform parallel operations by recognizing gestures and converting them to words and audio and taking words as an input and converting them to gestures as an output
- **Control function:**
- Capture the gestures on user’s hand.

Project ID: Winter-2021-33 Session: Winter 2018 – 2022.

.

- Convert the gestures into words and audio
- Plain text is given as input
- Text is converted to gestures in the form of images

- **Higher-order language requirements:** Higher-order language will be python as all our ML code will be written in python.

- **Signal handshake protocols:** Our aim is to develop a system where the user can communicate with a non mute users

- **Reliability requirements:** Our aim is to achieve more than 90% accuracy for this and develop a reliable and trustworthy environment between the system and the user.

- **Criticality of the application:** The project depends on both hardware and software, there are a lot of things to explore in this area but right now we are going just for the basics of the detection/tracking of gestures and converting them to words/audio with the features like machine learning for the user and taking words as input and converting them to gestures in the form of photos.
- **Safety and security considerations:** Safety and security considerations includes accurately detecting/tracking gestures and converting them to relevant words/ audio against the gestures so that a mute user could have ease in communication with a nonmute person

2.2.5 ASSUMPTIONS AND DEPENDENCIES

A number of factors that may affect the requirements specified in the SRS include:

- If the person does not perform the correct gesture of required alphabet/word the systemwill not produce the desired output.
- The hand of the user should be clearly visible to the camera to maximize efficiency

2.2.6 APPORTIONING OF REQUIREMENTS

The main dependencies of this project are a working computer with Python and dependencies and enough storage to download and install our dataset and application

- The system will start detecting dynamic signs which are mostly used in words orsentences.
- The dataset to detect more words will be added gradually
- Android and iOS apps in future versions.

2.3 SPECIFIC REQUIREMENTS

This segment will depict the useful and non-useful necessities of Framework at an adequate degree of detail for the architects to plan a framework fulfilling the Client prerequisites and testicles to confirm that the framework fulfills the necessities.

2.3.1 FUNCTIONAL REQUIREMENT

- **SIGN LANGUAGE RECOGNITION:** The application must recognize signs of the user using thecamera's inputs.
- **LITERAL TRANSLATION:** The application must show on the screen the word or the idea thatrefers to the user movement detected.
- **INTERACTIVE LEARNING:** The application must provide an interface for interactive learningof PSL.
- **WORDS RECOGNITION:** The application must recognize words given as input.
- **WORDS CONVERSION:** The application convert the word into a gesture

2.3.2 NON-FUNCTIONAL REQUIREMENTS

This sub-section includes the following

- **Usability:**
  - This application has easily understandable design in order for users to use it o There will be two user types – the normal user and the hearing- impaired user – each of which will have its own corresponding interface.The minimal requirement what a normal user can do is learn gestures or type a word/alphabet which can be converted into gesture.
  - The minimal requirement what a hearing-impaired user can do is perform and learn gestures.
  - The minimal requirement what a normal user can do is learn to write words that would be converted into gestures

- **Reliability:**
  - The system must function under any given circumstances.
  - There can be a maximum of 1 bug/KLOC.
  - In the case of bad gesture recognition, the system will give an appropriateerror.

- **Performance:**
  - The recognition process will be done in real-time. (3 seconds maximum) The maximum response time to get the gesture animation will be 3seconds.
  - It will be more efficient and user friendly in terms of use. The trainedclassified model will be fast, reliable, secure and accurate

.

- **Design Constraints:** The quality of camera should be better in order for it to detect and track object and show desired results. The application is only supported by Desktop devices having Python installed and will run flawlessly on any device with Python installed and working. o The design of the system will be made to cater to both normal and people with hearing impaired. o Most of the coding will be done in Python.
- **Portability:** The system will be easily available and should be portable for both users. Theapplication is portable for any desktop device since the Python language is platform independent
- **Maintainability:**
  - The system will be regularly tested for any types of glitches and errors whichcan affect its efficiency.
  - It will be regularly updated to prevent any virus.
- **License Agreement:** It will be accessible under a GPL permit and will be opensource. All clients would need to stick to the approach of the permit understanding or the undertaking designers would have right to suspend their records for endless period. Client of the application can utilize it anyplace with next to no expense and with no kind of limitations be that as it may, since the application is a shut source and exclusive permit material, so client reserve no privileges to check, change or utilize the source code of the application

Chapter 3: Use Case Analysis

3.1 USE CASE DIAGRAM-(LEVEL-1)

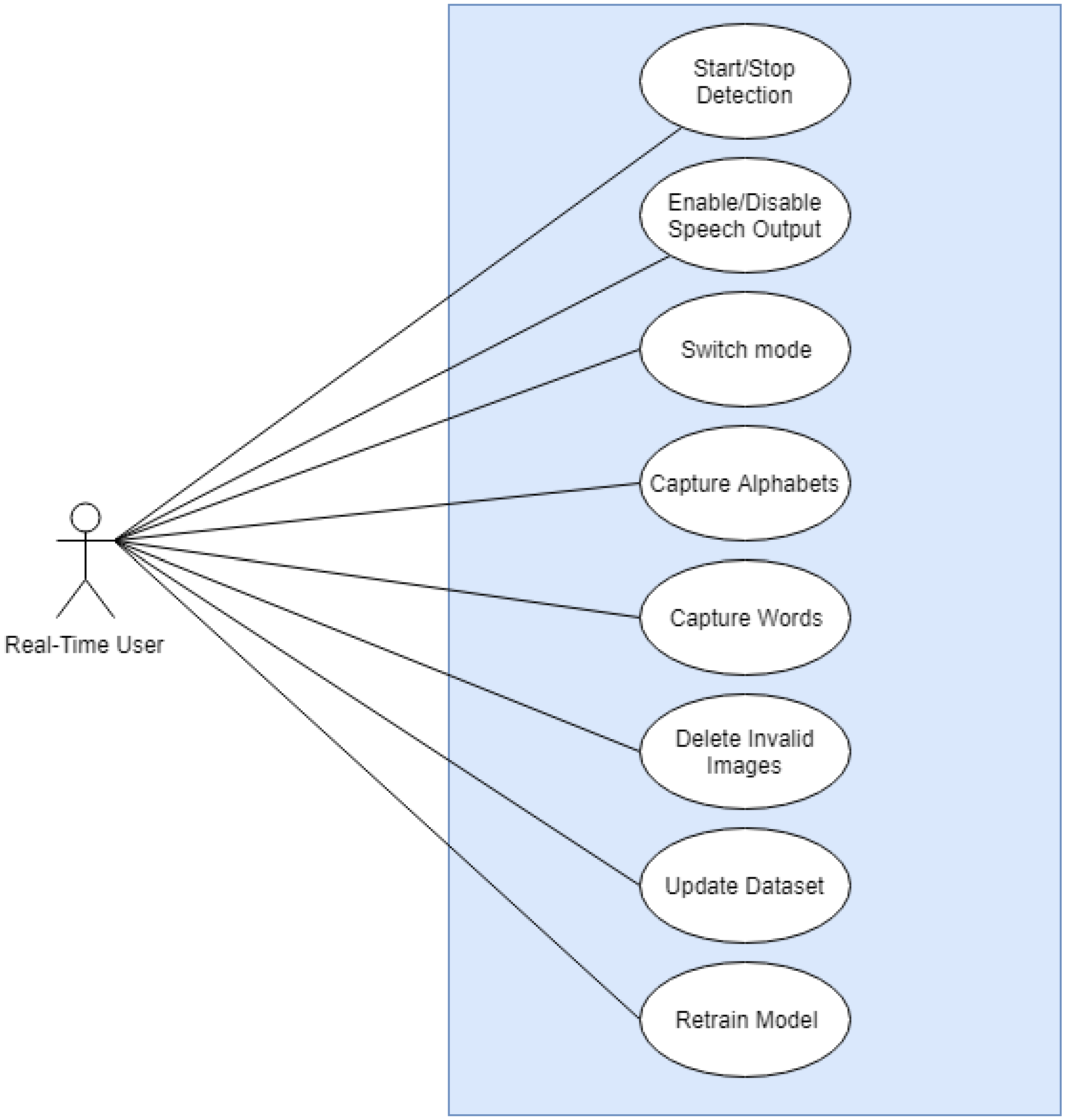


Figure 3.1. (a) Use Case Diagram (Level 1)

.

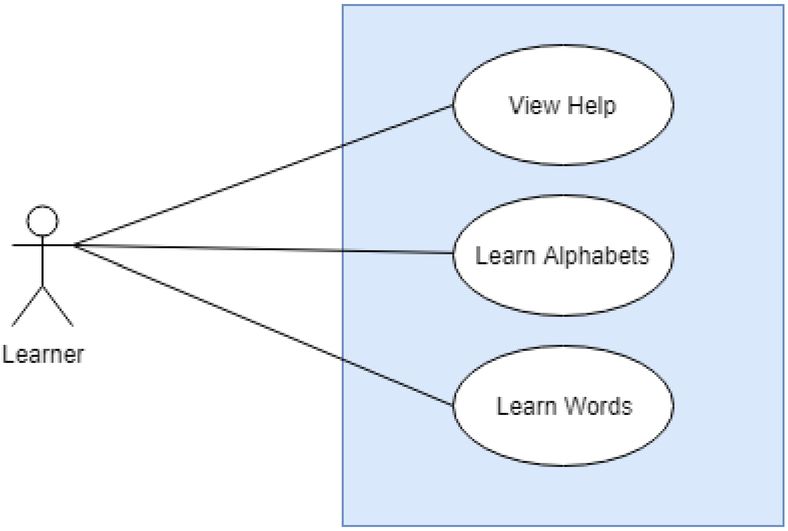


Figure 3.1. (b) Use Case Diagram (Level 1)

3.2 USE CASE DIAGRAM-(LEVEL-2)

UC_01(START/STOPDETECTION):

TABLE 3.1. UC_01 (START/STOP DETECTION)

Use Case ID	UC_01	
Use Case Name	Start/Stop Detection	
Description	Click the icon to Start or Stop Detection	
Primary Actor	Real-time User	
Secondary Actor	None	
Pre-Condition	Recognition Stopped / Recognition Started	
Post-Condition	Outputs Word/Alphabet / Outputs nothing	
Basic Flow	Actor Action	System Action
	Click Start/Stop Icon to Start/Stop Detection	System will Start/Stop Detecting gestures
Alternate Flow	If user performs invalid gesture system will not output anything.	

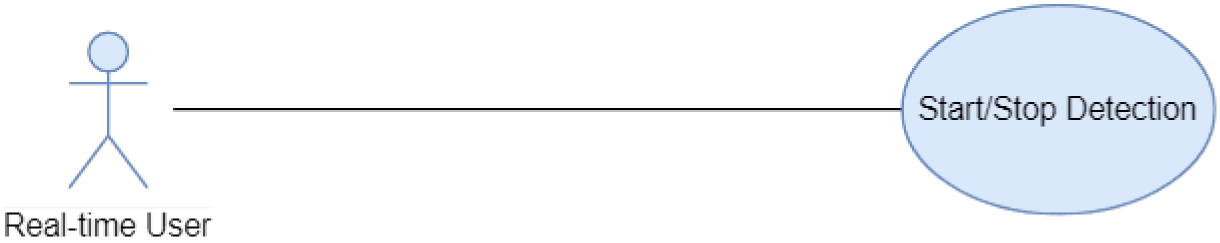


Figure 3.2. UC_01 (Start/ Stop Detection)

UC_02(LEARN ALPHABETS):

TABLE 3.2. UC_02 (LEARN ALPHABETS)

Use Case ID	UC_02	
Use Case Name	Learn Alphabets	
Description	Click Learn Icon to and then click Alphabets to Start Learning Process	
Primary Actor	Learner User	
Secondary Actor	None	
Pre-Condition	User should be at Main window	
Post-Condition	Click Start Learning then Click alphabets to Start process and skip for the next Alphabet	
Basic Flow	Actor Action	System Action
	Click Learn Icon, Click Alphabet then Perform Gestures, Skip for next Alphabet	Start Alphabet Learning Process, Output Good for matched gesture and Bad for unmatched gesture
Alternate Flow	None	

.

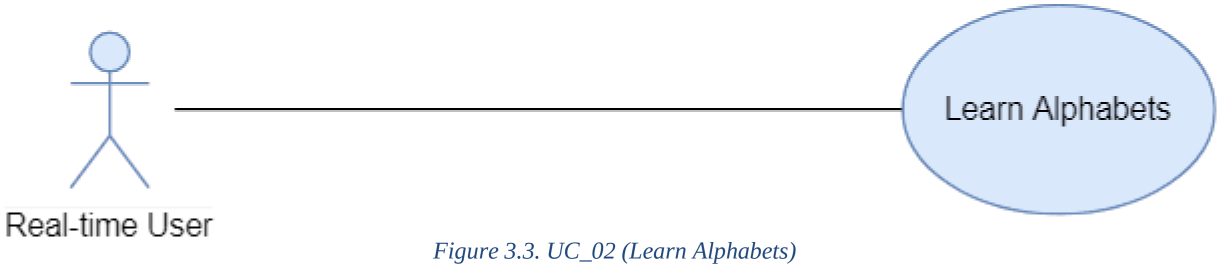


Figure 3.3. UC_02 (Learn Alphabets)

UC_03(ENABLE/DISABLE OUTPUT):

TABLE 3.3. UC_03 (ENABLE/DISABLE SPEECH OUTPUT)

Use Case ID	UC_03	
Use Case Name	Enable/Disable Speech Output	
Description	Enables and Disables Speech Output	
Primary Actor	Real-time User	
Secondary Actor	None	
Pre-Condition	Speech checkbox should be checked/unchecked	
Post-Condition	Outputs Speech / nothing	
Basic Flow	Actor Action	System Action
	User Checks/Unchecks the box to enable or disable speech.	System outputs Speech if enabled or nothing if disabled.
Alternate Flow	None	

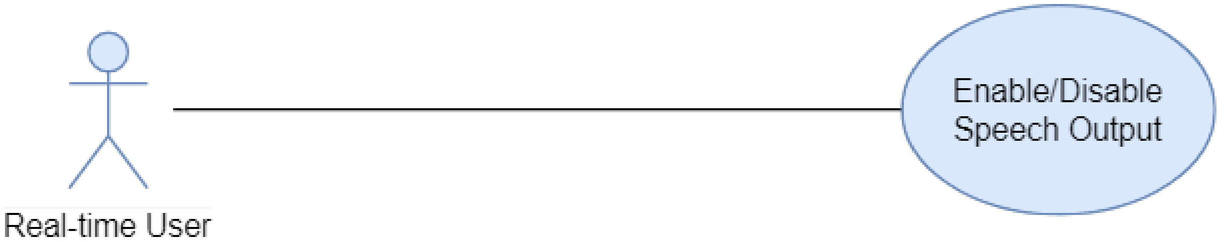


Figure 3.4. UC_03 (ENABLE/DISABLE SPEECH OUTPUT)

UC_04(SWITCH MODE):

TABLE 3.4. UC_04 (SWITCH MODE)

Use Case ID	UC_04	
Use Case Name	Switch Mode	
Description	Switches between different modes	
Primary Actor	Real-time User	
Secondary Actor	None	
Pre-Condition	Some mode should be selected	
Post-Condition	Mode should start its process	
Basic Flow	Actor Action	System Action
	User clicks on the Mode icon to start that Mode	System starts the process of selected Mode
Alternate Flow	User can go back from selected mode and select some other mode as well.	

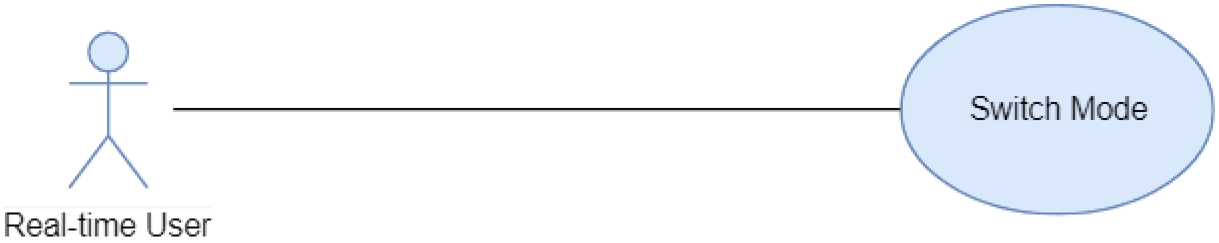


Figure 3.5. UC_04 (SWITCH MODE)

UC_05(VIEW HELP):

TABLE 3.5. UC_05 (VIEW HELP)

Use Case ID	UC_05	
Use Case Name	View Help	
Description	It is user manual which helps the user about the software	
Primary Actor	Real-Time User	
Secondary Actor	None	
Pre-Condition	Should be at the Main window	
Post-Condition	Should open User manual	
Basic Flow	Actor Action	System Action
	Click Help "?" Icon to show user manual window	System should show Help window
Alternate Flow	None	

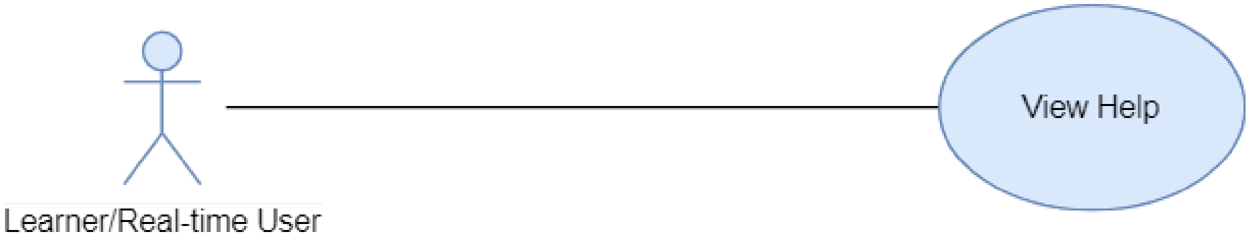


Figure 3.6. UC_05 (VIEW HELP)

UC_06(LEARN WORDS):

TABLE 3.6. UC_06 (LEARN WORDS)

Use Case ID	UC_06	
Use Case Name	Learn Words	
Description	Click Learn Icon to and then click Words to Start Learning Process	
Primary Actor	Learner User	
Secondary Actor	None	
Pre-Condition	User should be at Main window	
Post-Condition	Click Start Learning then Click alphabets to Start process and skip for the next Words	
Basic Flow	Actor Action	System Action
	Click Learn Icon, Click Words then Perform Gestures, Skip for next Word	Start Word Learning Process, Output Good for matched gesture and Bad for unmatched gesture
Alternate Flow	None	

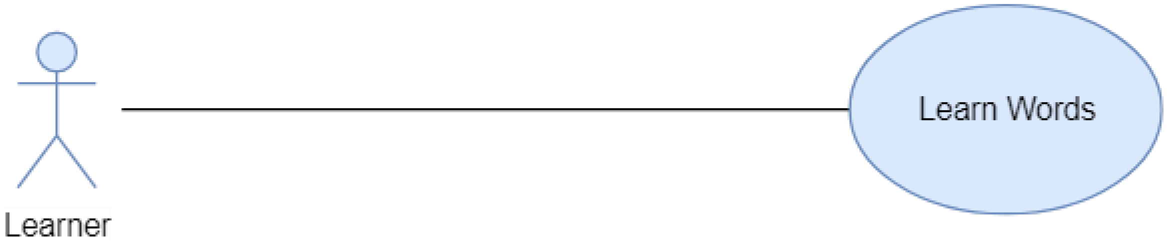


Figure 3.7. UC_06 (LEARN WORDS)

UC_07 (CAPTURE ALPHABETS):

TABLE 3.7 UC_07 (CAPTURE ALPHABETS)

Use Case ID	UC_07	
Use Case Name	Capture Alphabets	
Description	This Process will Capture Alphabets	
Primary Actor	Real-time User	
Secondary Actor	None	
Pre-Condition	Should be in Capture Dataset Window	
Post-Condition	It should show the images	
Basic Flow	Actor Action	System Action
	User should perform gestures of alphabets	System should capture the dataset
Alternate Flow	None	

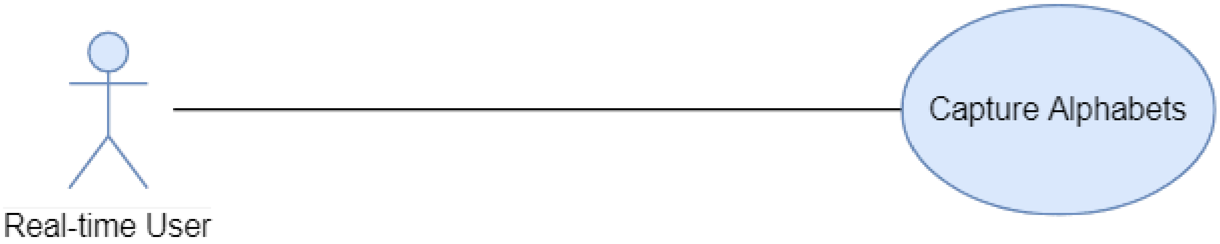


Figure 3.8. UC_07 (Capture Alphabets)

UC_08 (CAPTURE WORDS):

TABLE 3.8. UC_08 (CAPTURE WORDS)

Use Case ID	UC_08	
Use Case Name	Capture Words	
Description	This Process will Capture Words	
Primary Actor	Real-time User	
Secondary Actor	None	
Pre-Condition	Should be in Capture Dataset Window	
Post-Condition	It should show the images	
Basic Flow	Actor Action	System Action
	User should perform gestures of Words	System should capture the dataset
Alternate Flow	None	

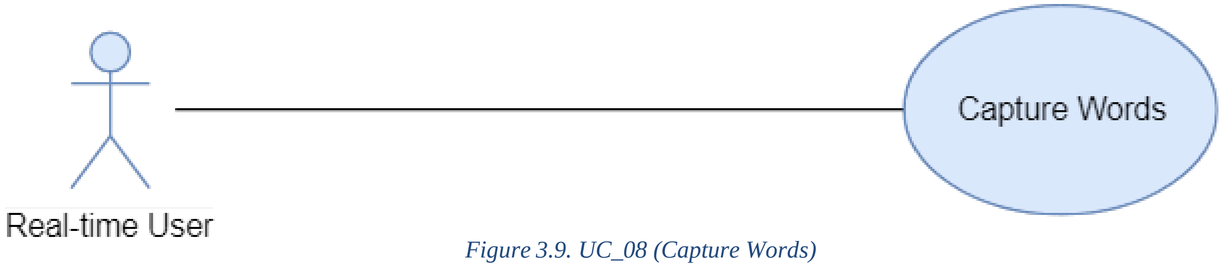


Figure 3.9. UC_08 (Capture Words)

UC_09 (DELETE INVALID IMAGES):

TABLE 3.9. UC_09 (DELETE INVALID IMAGES)

Use Case ID	UC_09	
Use Case Name	Delete Invalid Images	
Description	This process will Delete Invalid Images captured by the user	
Primary Actor	Real-time User	
Secondary Actor	None	
Pre-Condition	Should Capture the gestures	
Post-Condition	Images should be deleted from the folder	
Basic Flow	Actor Action	System Action
	Capture the gestures and select the one which are invalid and should be deleted	System should delete the invalid images from the folder
Alternate Flow	None	

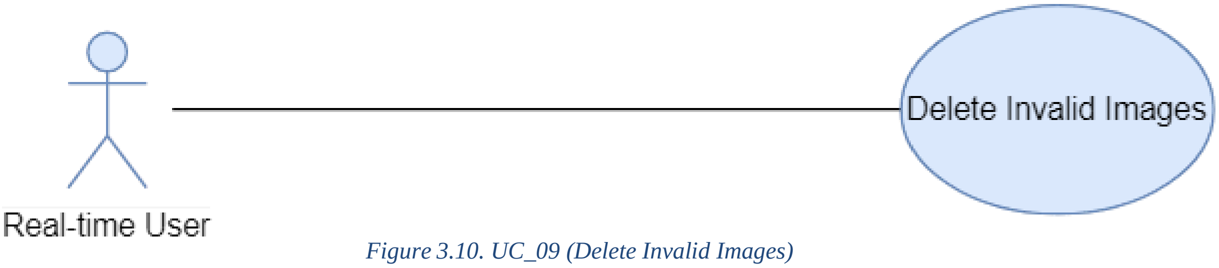


Figure 3.10. UC_09 (Delete Invalid Images)

UC_10 (UPDATE DATASET):

TABLE 3.10. UC_10 (UPDATE DATASET)

.

Use Case ID	UC_10	
Use Case Name	Update Dataset	
Description	This process should Update the Dataset	
Primary Actor	Real-time User	
Secondary Actor	None	
Pre-Condition	User should perform Gesture	
Post-Condition	It should be added in the database/server for training	
Basic Flow	Actor Action	System Action
	User should perform the gesture. Delete the invalid images and store the rest in database/server	System should update its dataset with the new images
Alternate Flow	None	

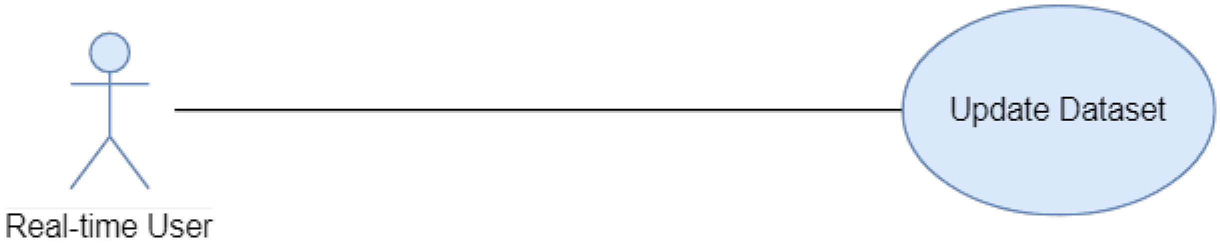


Figure 3.11. UC_10 (Update Dataset)

UC_11 (RETRAIN MODEL):

TABLE 3.11. UC_11 (RETRAIN MODEL)

Use Case ID	UC_11	
Use Case Name	Retrain Model	
Description	This feature Retrains Model after adding images in dataset	
Primary Actor	Real-time User	
Secondary Actor	None	
Pre-Condition	User should capture images	
Post-Condition	System should be able to detect the gesture	
Basic Flow	Actor Action	System Action
	Save the images in the database/server and then retrain the model.	System should be able to detect the gesture.
Alternate Flow	None	

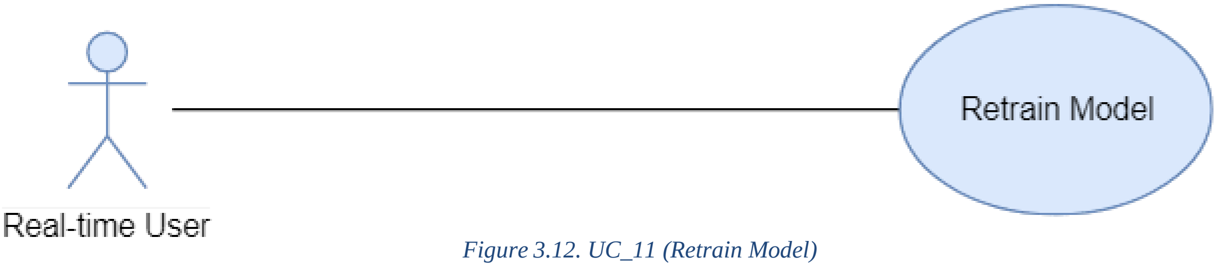


Figure 3.12. UC_11 (Retrain Model)

Chapter 4: Design

.

4.1 ARCHITECTURE DIAGRAM

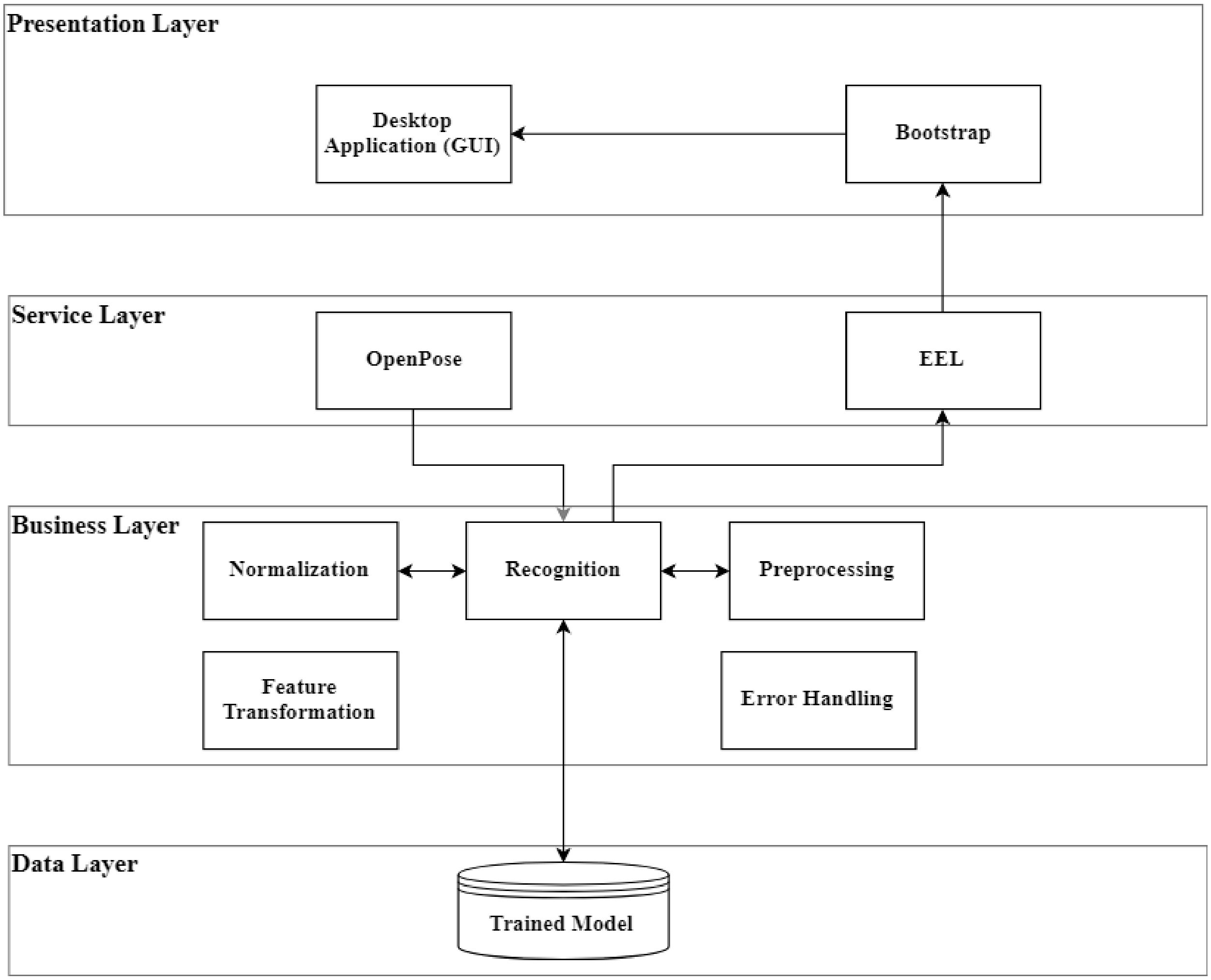


Figure 4.1. ARCHITECTURE DIAGRAM

4.1 ERD WITH DATA DICTIONARY

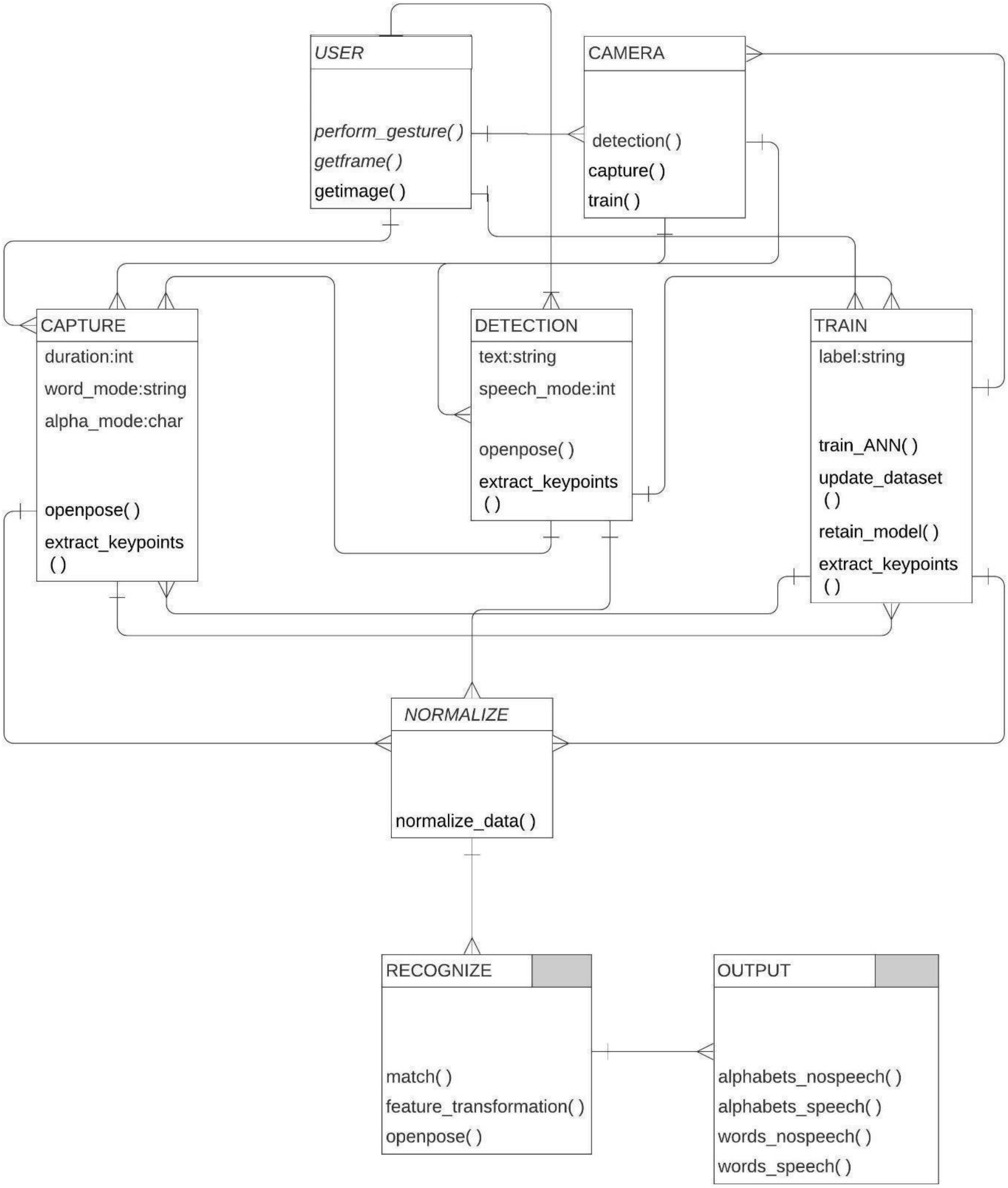


Figure 4.2.ERD DATA DICTIONARY

4.2 DATA FLOW DIAGRAM

4.2.1 THE LEVEL 0

.

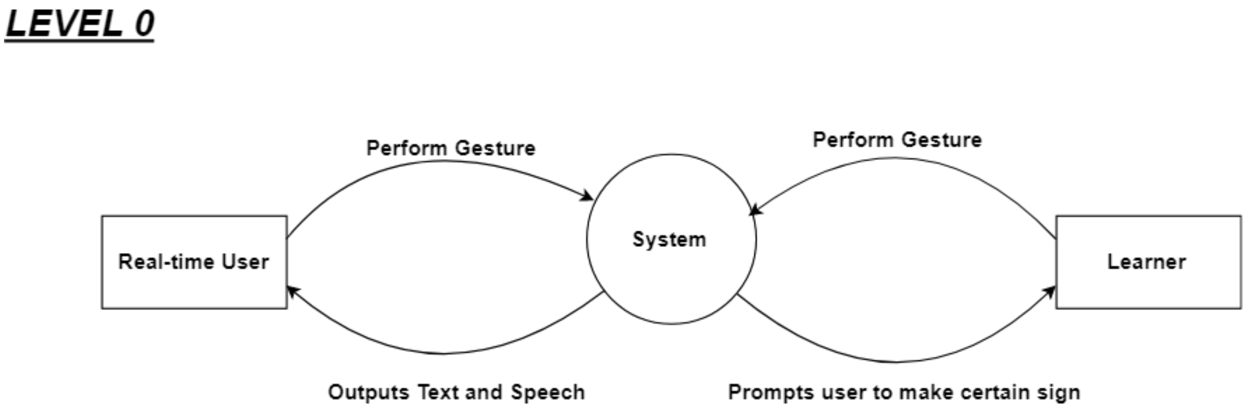


Figure 4.3. DATA FLOW DIAGRAM (LEVEL 0)

4.2.2 THE LEVEL 1

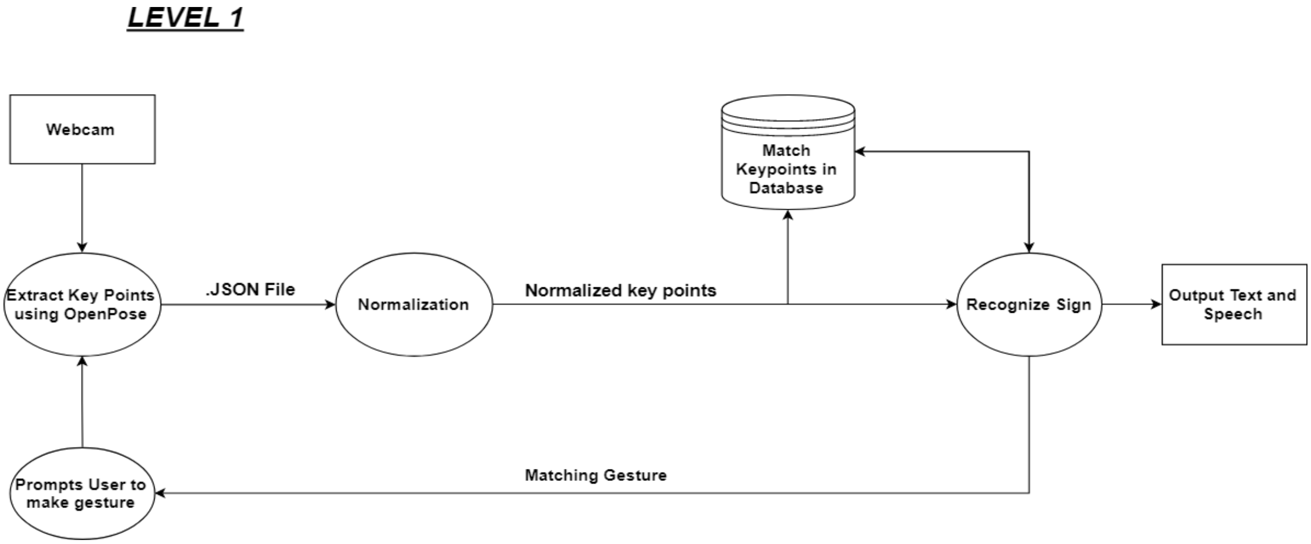


Figure 4.4. DATA FLOW DIAGRAM (LEVEL 1)

4.3 CLASS DIAGRAM

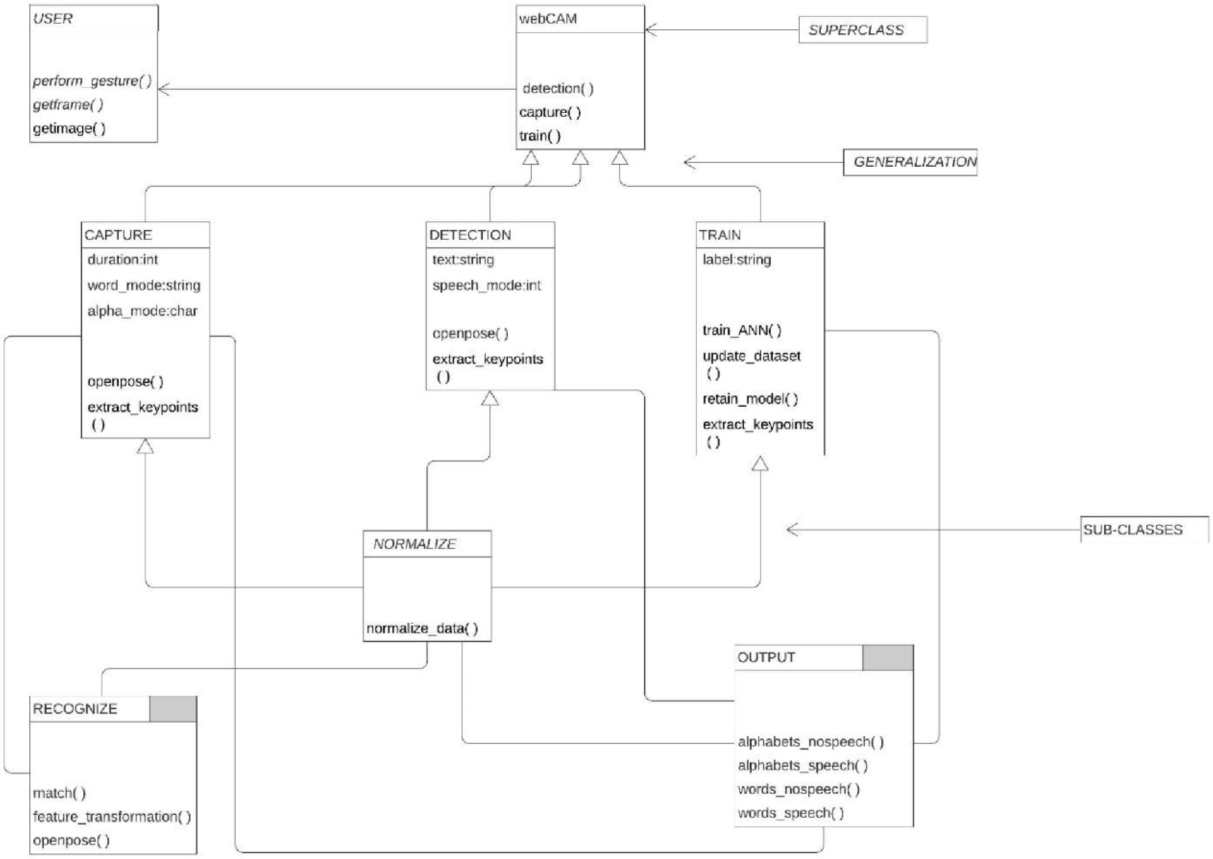


Figure 4.5. CLASS DIAGRAM

4.4 ACTIVITY DIAGRAM

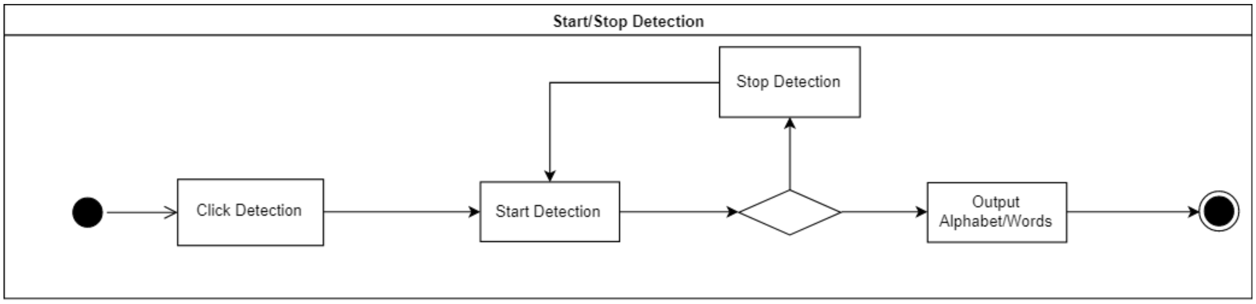


Figure 4.6 Activity Diagram (Start/Stop detection)

.

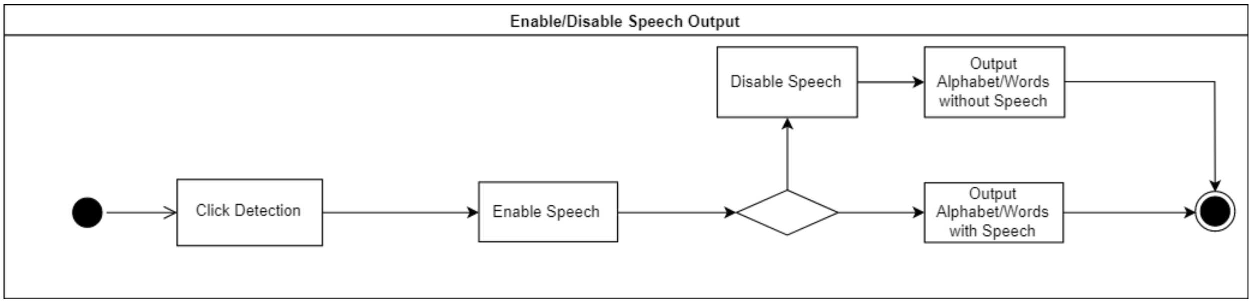

Enable/Disable Speech Output
Disable Speech
Output Alphabet/Words without Speech
Click Detection
Enable Speech
Output Alphabet/Words with Speech

Figure 4.7 Activity Diagram (Enable/Disable Speech)

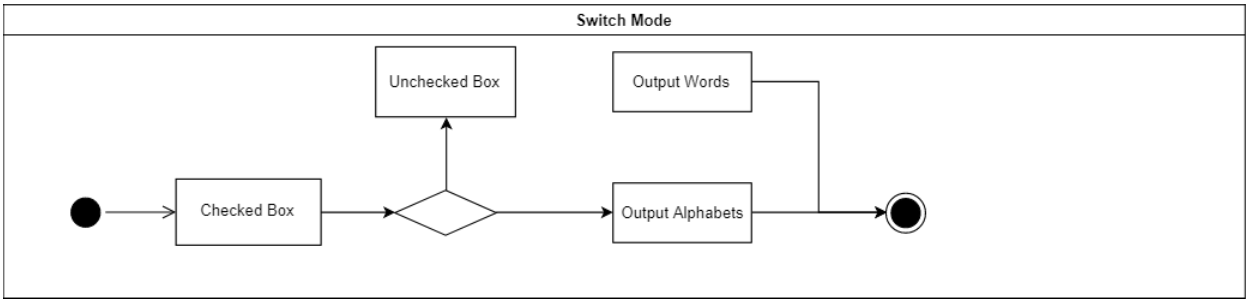


Figure 4.8. Activity Diagram (Switch Mode)

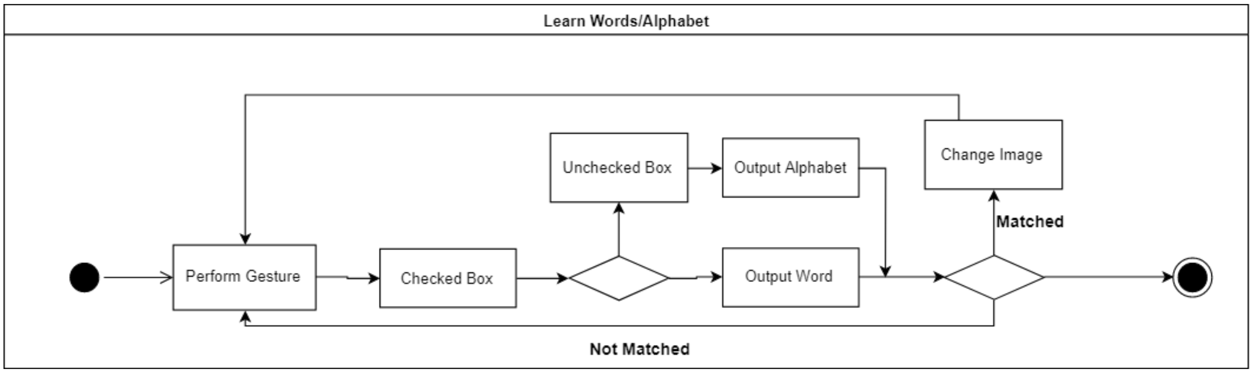


Figure 4.9. Activity Diagram (Learn Word/Alphabet)

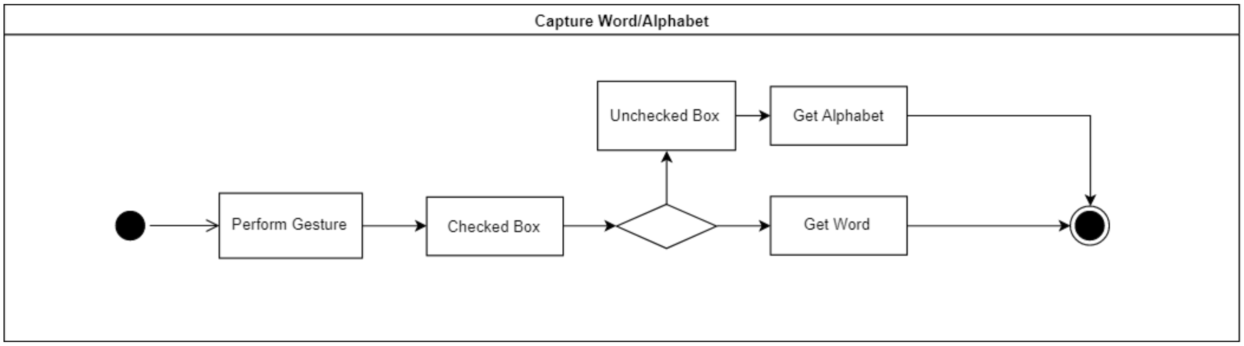


Figure 4.10. Activity Diagram (Capture Word/Alphabet)

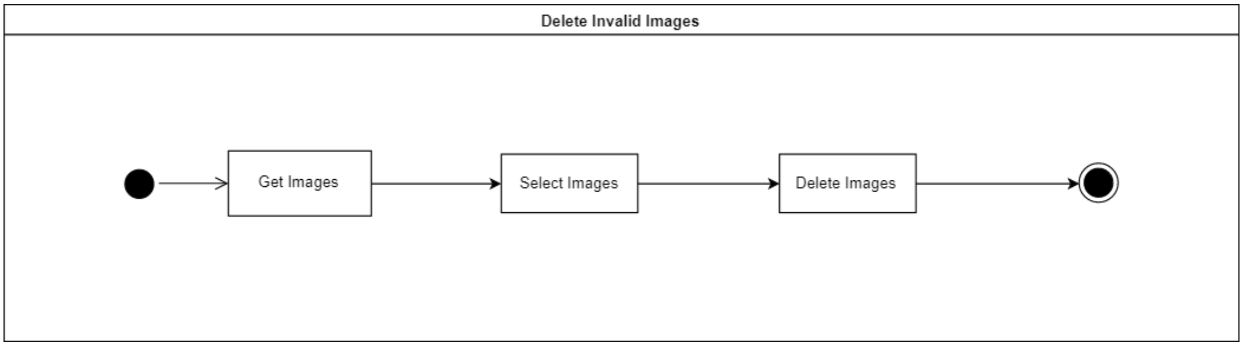


Figure 4.11. Activity Diagram (Delete Invalid Images)

4.5 SEQUENCE DIAGRAM

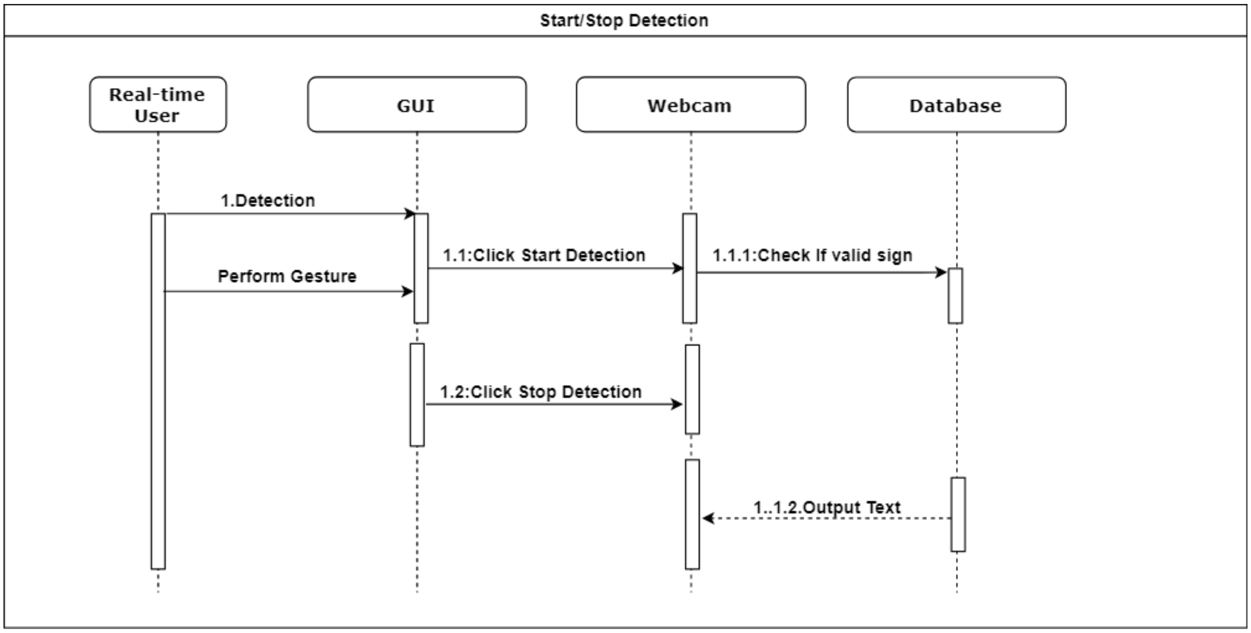


Figure 4.12 Sequence Diagram (Start/Stop Detection)

.

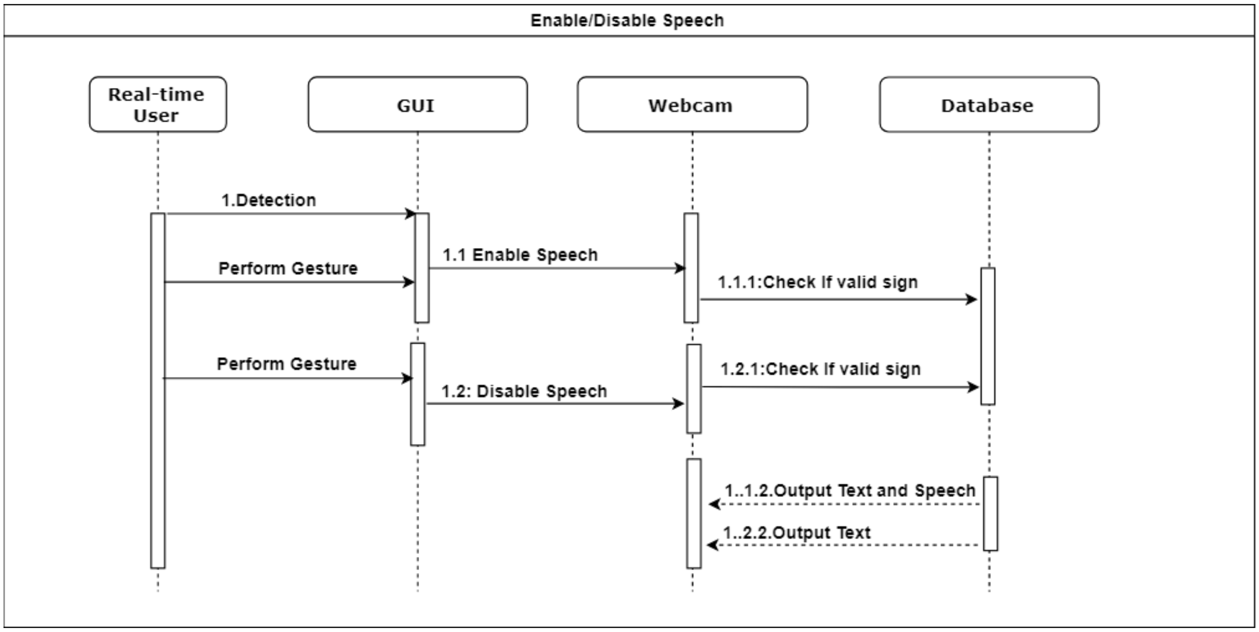


Figure 4.13. Sequence Diagram (Enable/disable Speech)

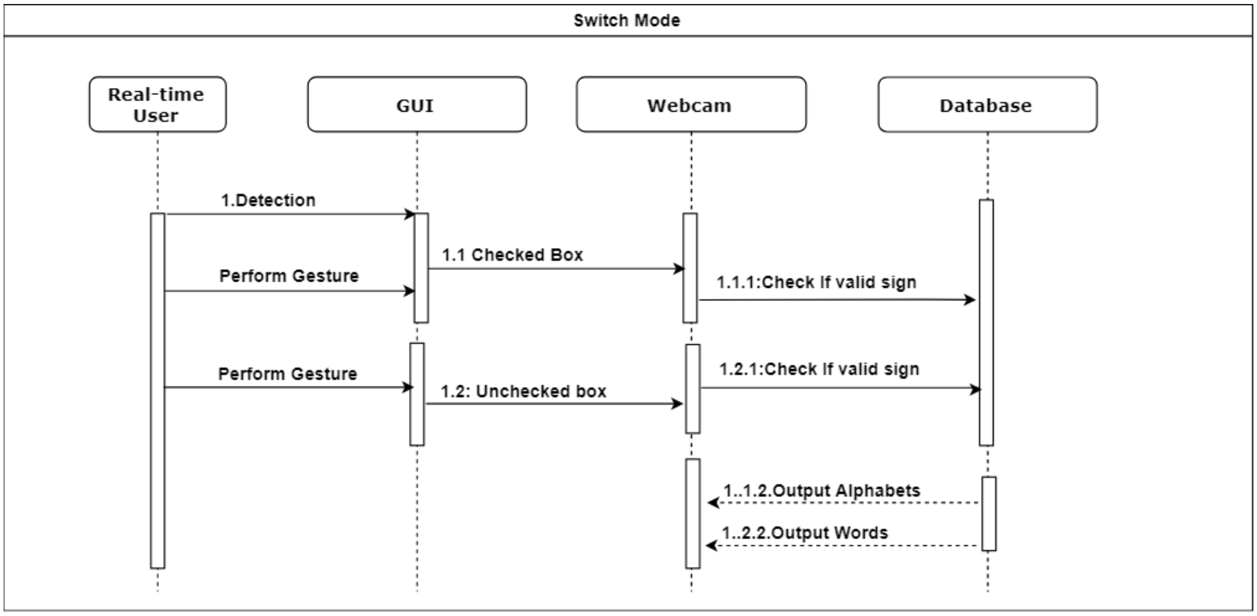


Figure 4.14 Sequence Diagram (Switch Mode)

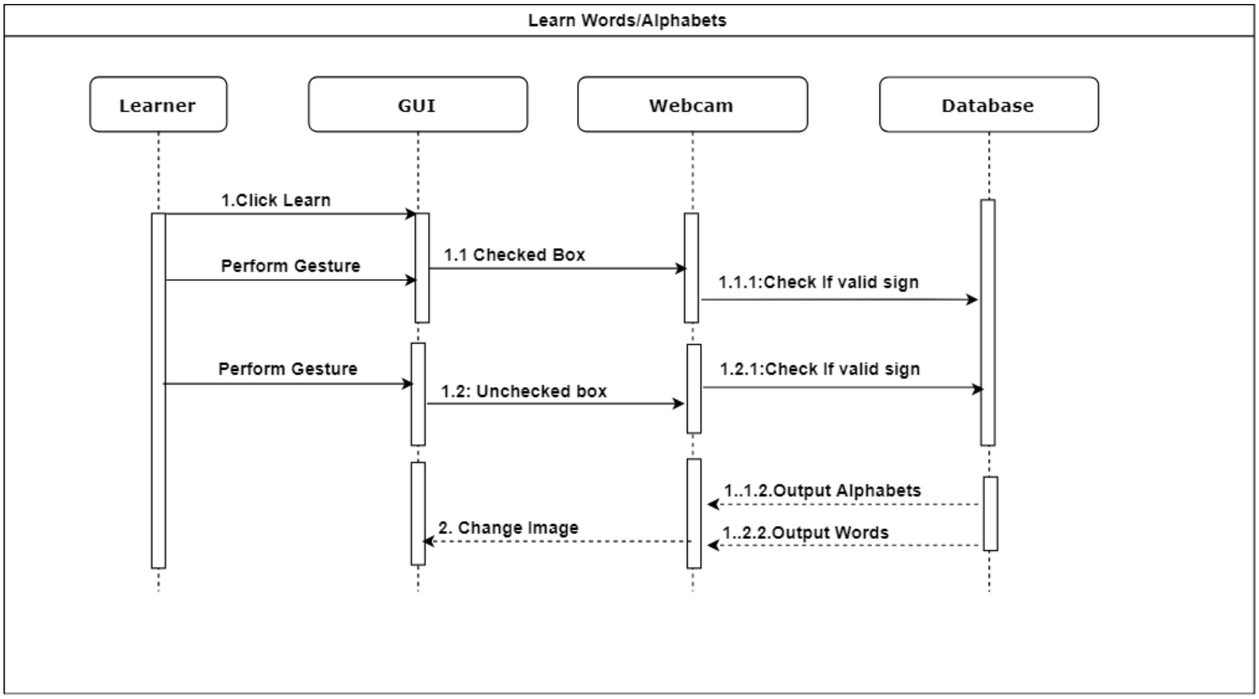


Figure 4.15 Sequence Diagram (Learn Words/Alphabets)

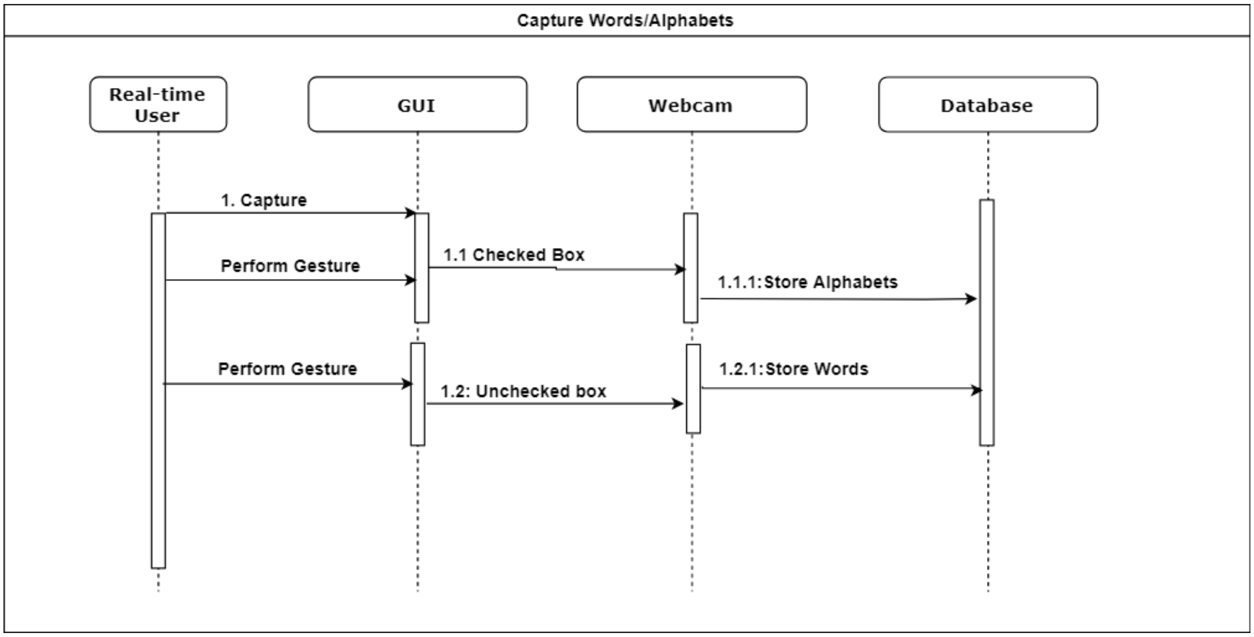


Figure 4.16 Sequence Diagram (Capture Words/Alphabets)

.

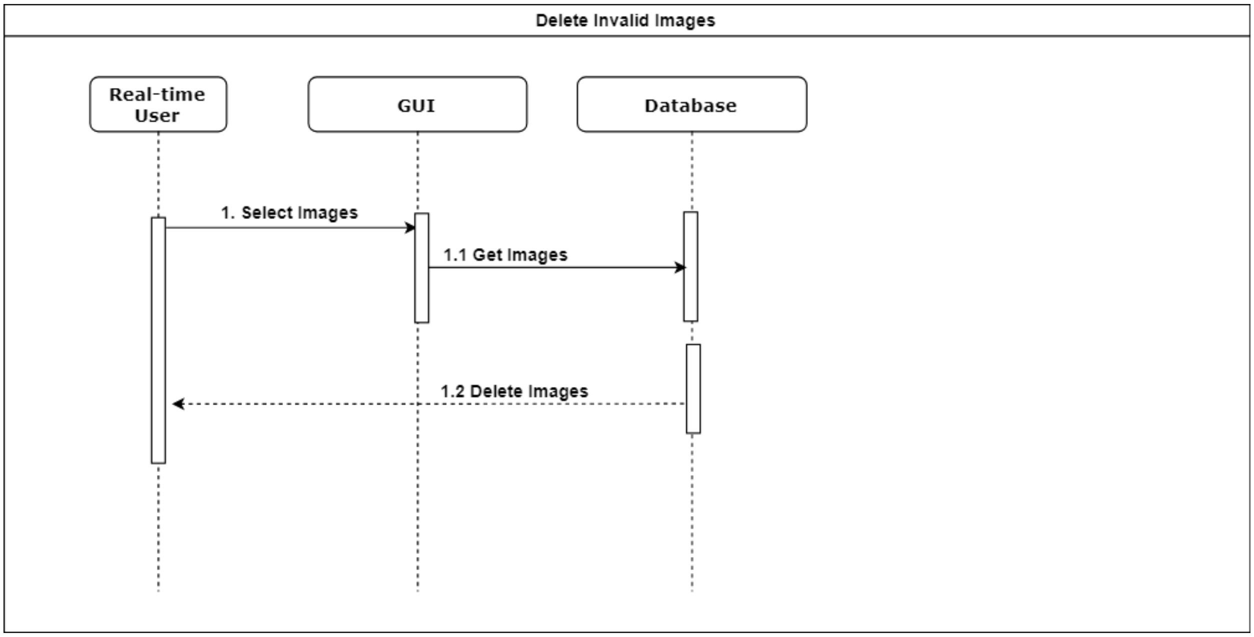


Figure 4.17. Sequence Diagram (Delete Invalid Images)

4.6 COLLABORATION DIAGRAM

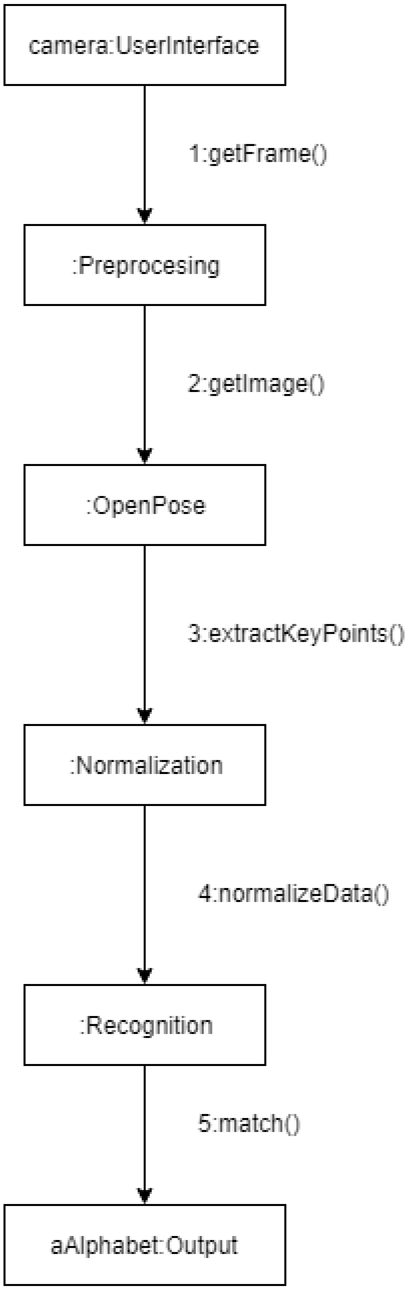


Figure 4.18. Collaboration Diagram

4.7 STATE TRANSITION DIAGRAM

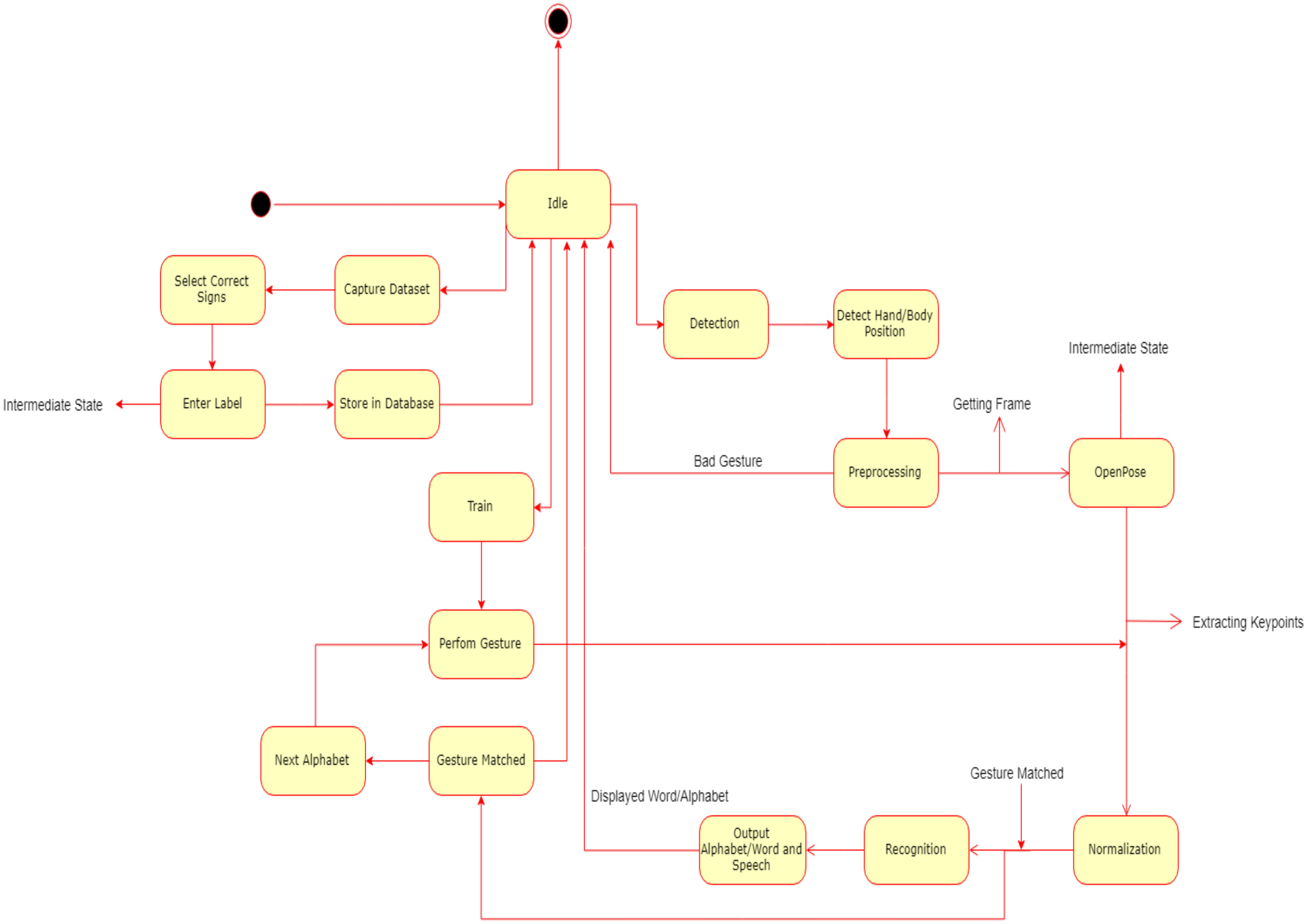


Figure 4.19. State Transition Diagram

.

4.8 Component Diagram

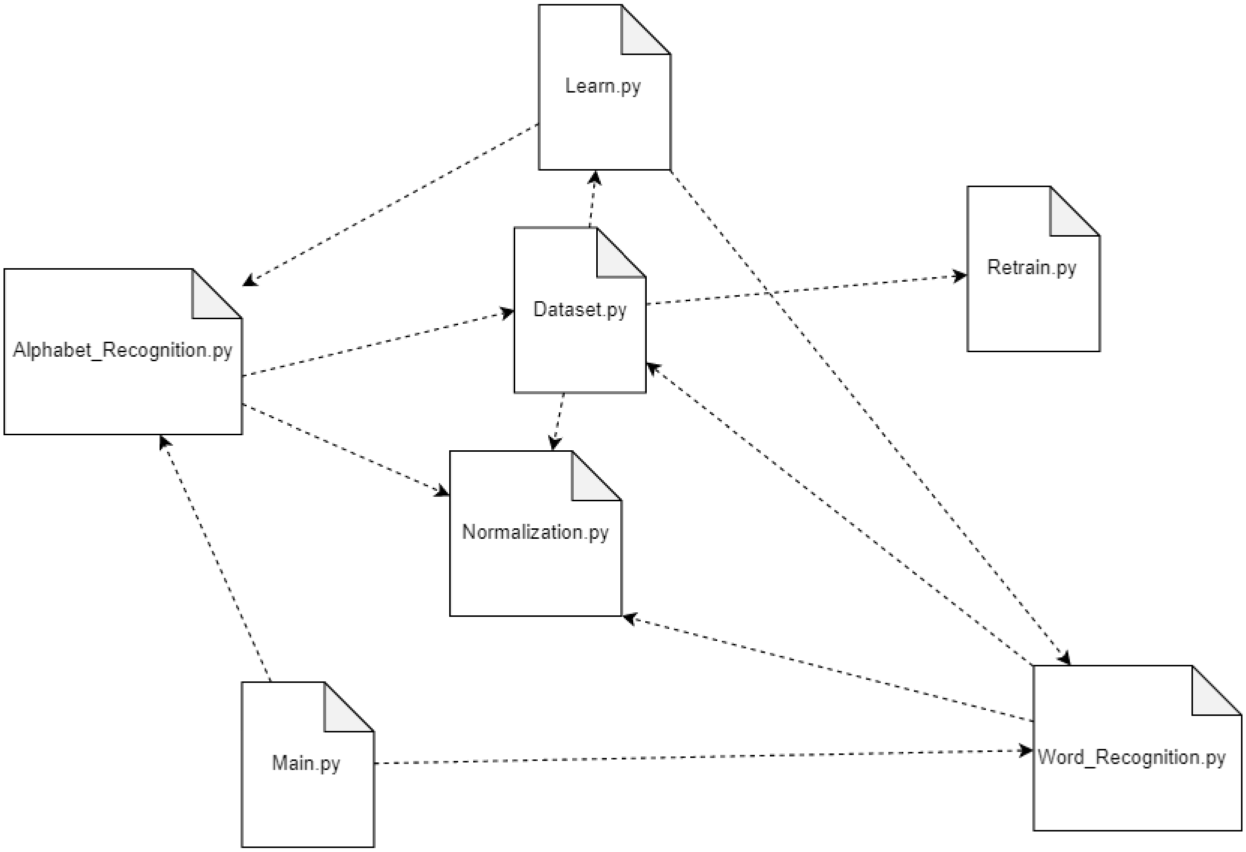


Figure 4.20. Component Diagram

4.9 Deployment Diagram

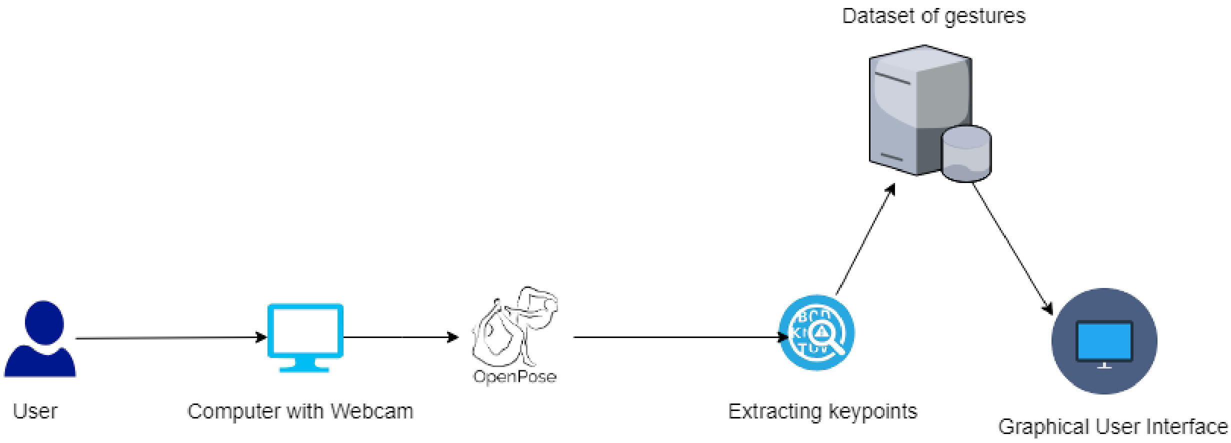


Figure 4.21. Deployment Diagram

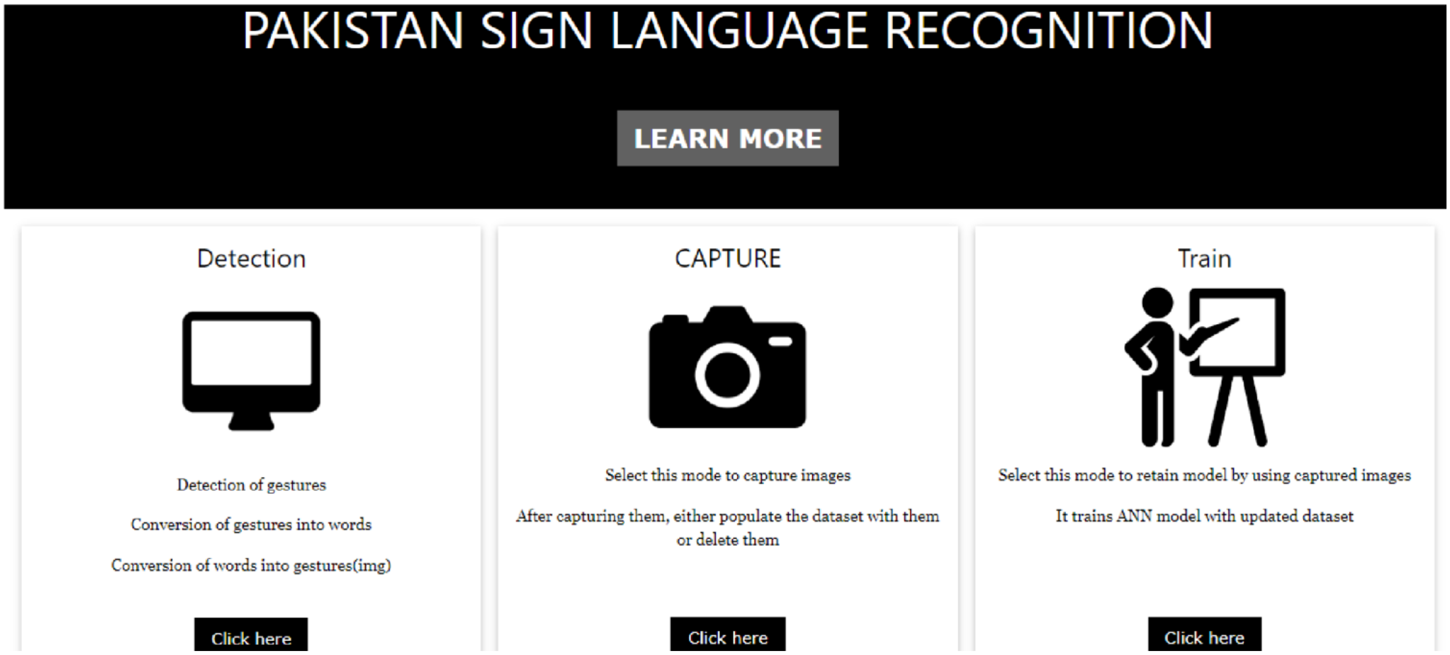
.

Chapter 5: User Manual

5.1 Homepage:

To see the application, go to our website PSLconverter.com

- Go to Detection Page to start Pakistan Sign Language conversion to text and speech. Or to convert text to gesture



Detection of gestures

Conversion of gestures into words

Conversion of words into gestures(img)

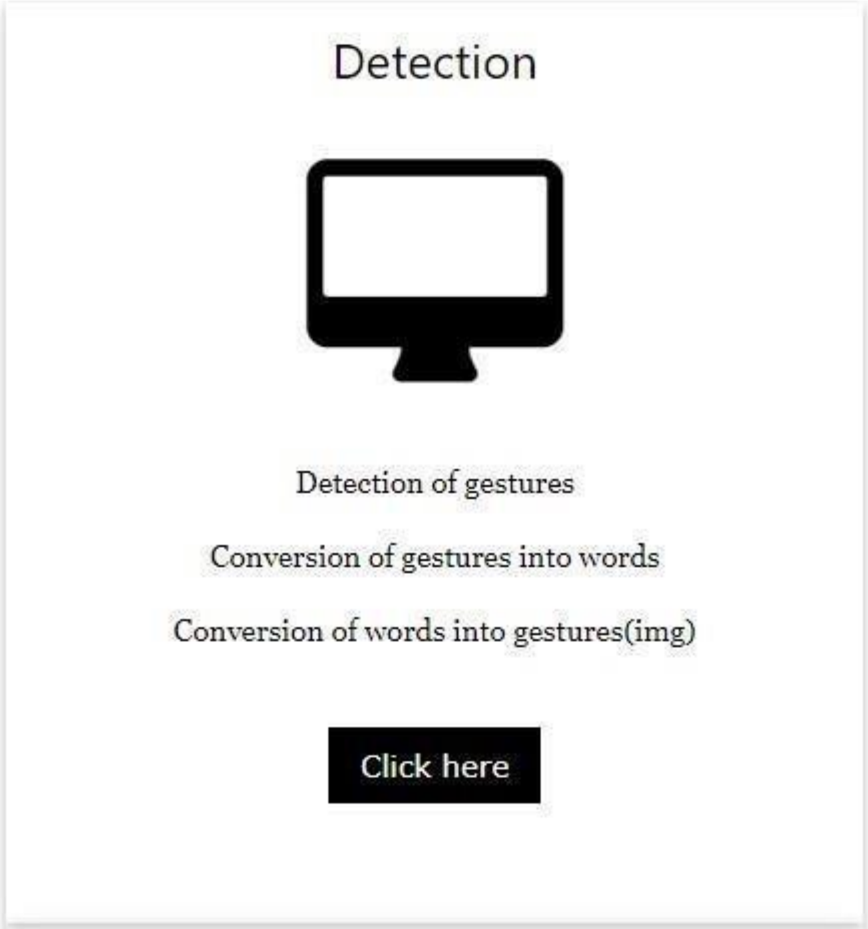
Select this mode to capture images

After capturing them, either populate the dataset with them or delete them

Select this mode to retain model by using captured images

It trains ANN model with updated dataset

Figure 5.1. MAIN SCREEN 1



Detection

Detection of gestures

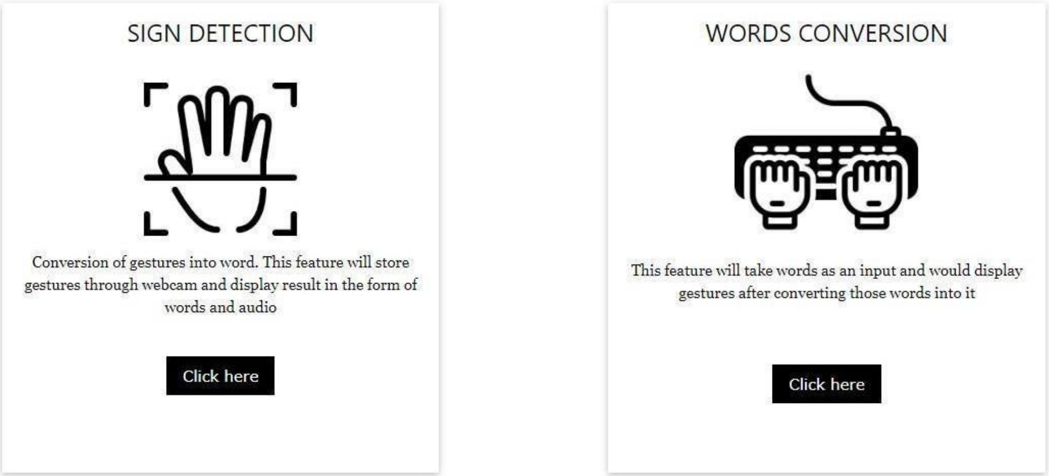
Conversion of gestures into words

Conversion of words into gestures(img)

Figure 5.2 DETECTION

- Click on click here button to go to detection page
- On detection page,

5.2 Detection Page:



SIGN DETECTION

Conversion of gestures into word. This feature will store gestures through webcam and display result in the form of words and audio

WORDS CONVERSION

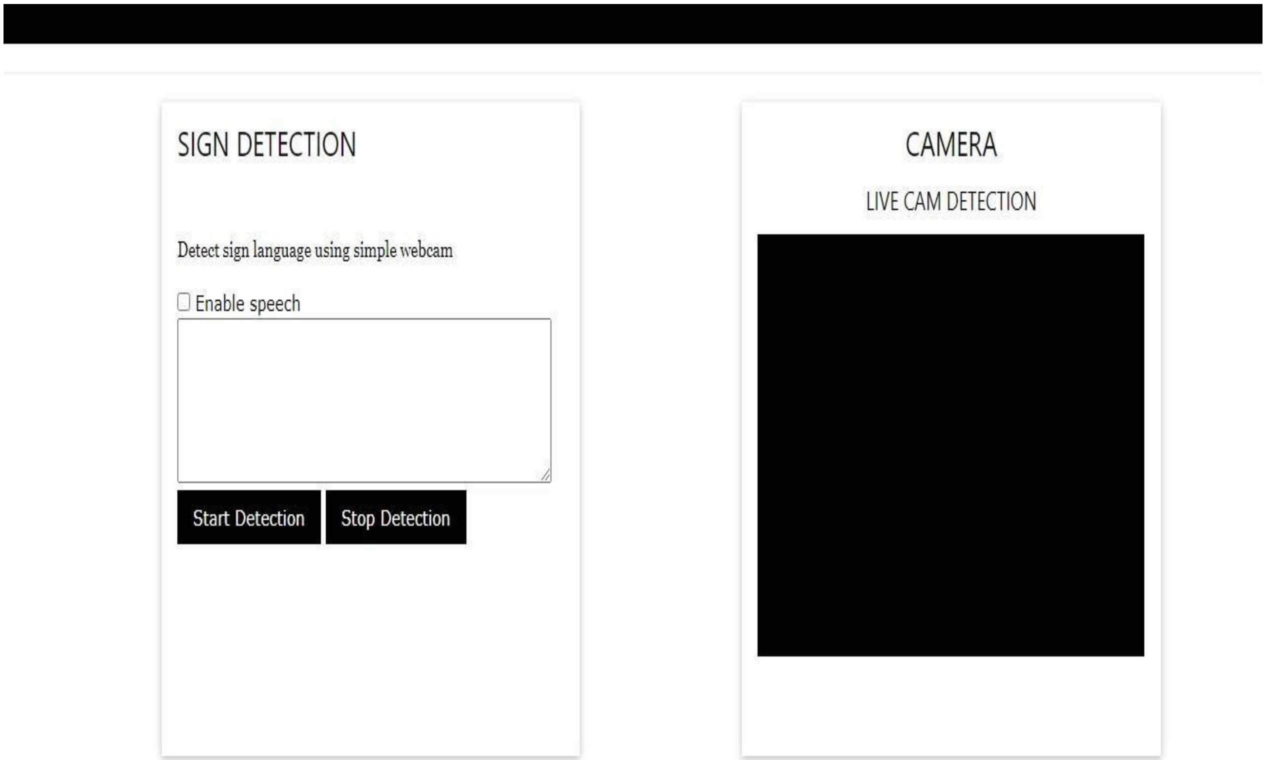
This feature will take words as an input and would display gestures after converting those words into it

Figure 5.2.1 DETECTION OPTIONS

- Click on click here to go to sign detection page or words conversion.

5.2.1 Sign detection

- Sign detection page interface is,



SIGN DETECTION

Detect sign language using simple webcam

☐ Enable speech

CAMERA

LIVE CAM DETECTION

Figure 5.2.2 SIGN DETECTION

5.2.3 Words Conversion

Converts signs into words

5.3 Learn page:

- Start learning Pakistan Sign Language interactively. Go to learn page

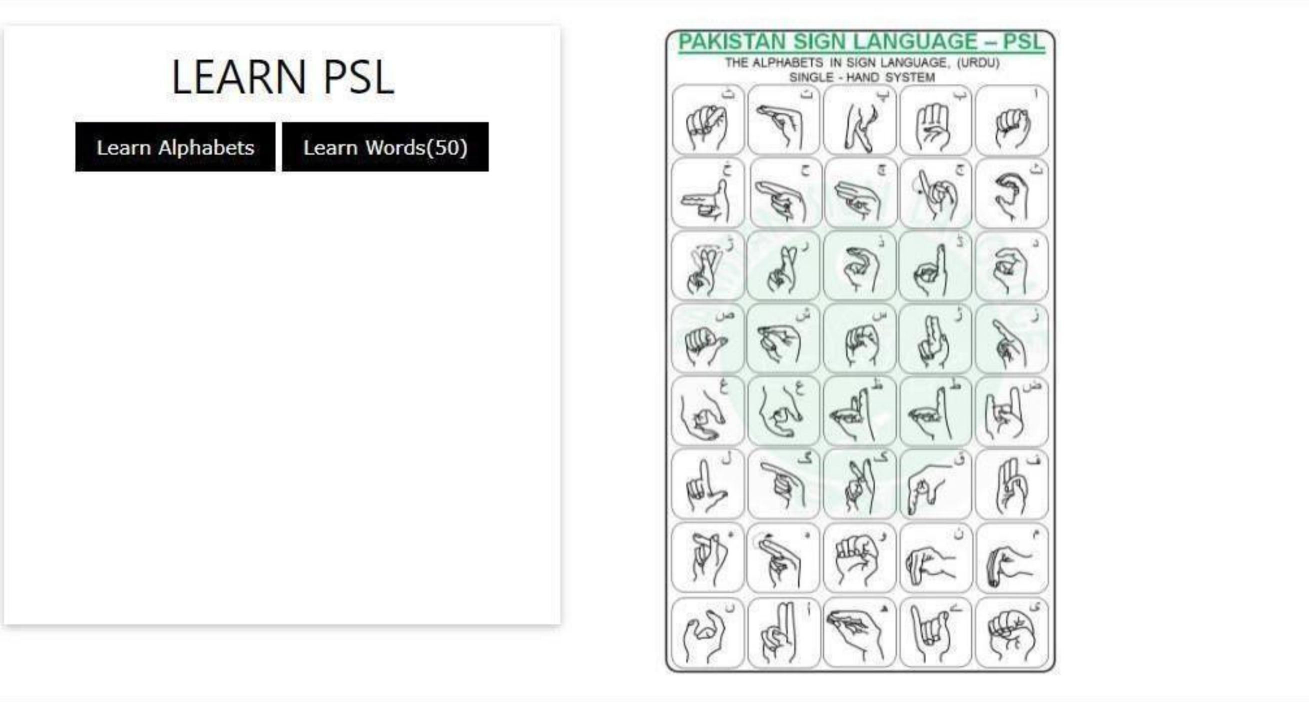

LEARN PSL
Learn Alphabets
Learn Words(50)
PAKISTAN SIGN LANGUAGE – PSL
THE ALPHABETS IN SIGN LANGUAGE, (URDU)
SINGLE - HAND SYSTEM

Figure 5.3 LEARN SCREEN

5.3.1 Learn Alphabets

Learn PSL alphabets

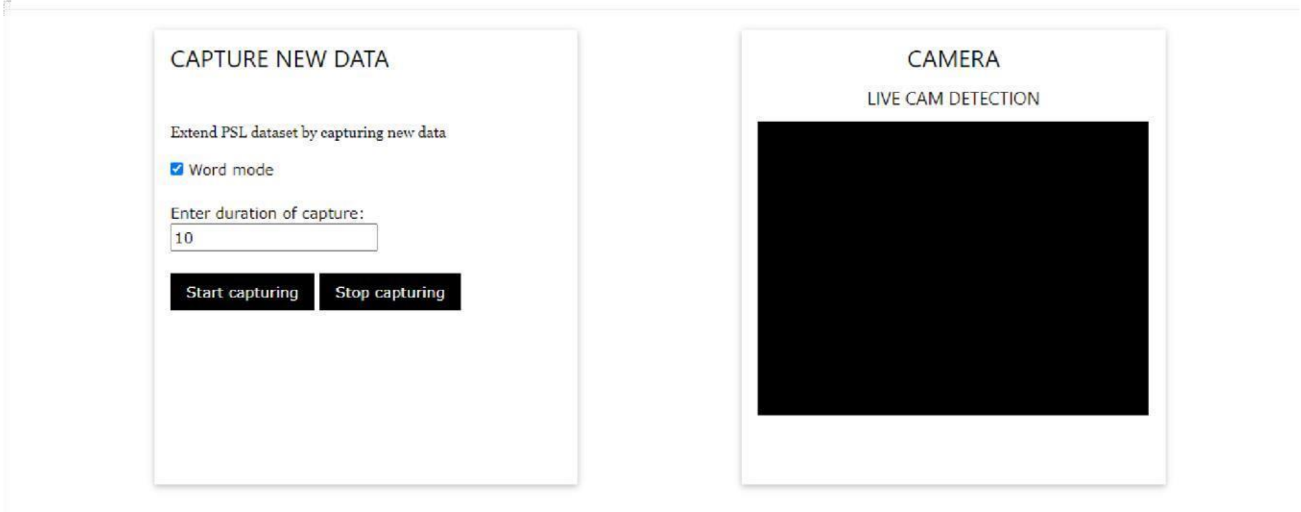
5.3.2 Learn Words

Learn PSL words

5.4 Capture Page:

- Go to capture page to capture new data for enhancing the conversion process

.



CAPTURE NEW DATA

Extend PSL dataset by capturing new data

☑ Word mode

Enter duration of capture: 10

Figure 5.4 screen capture

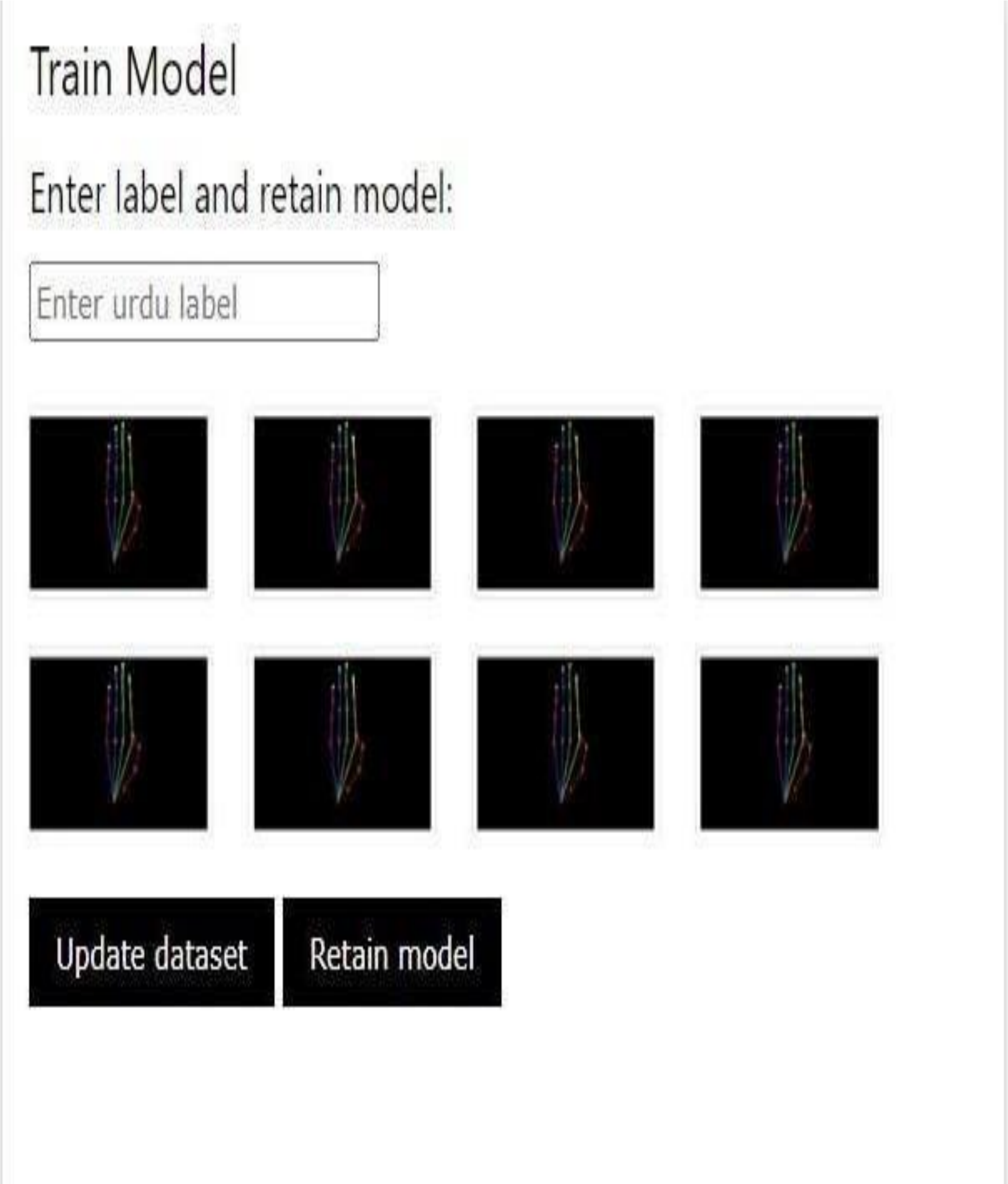
- Enter the duration in seconds, for how long you want to capture signs to be stored into the dataset.
- Select either word mode or alphabet mode
- Enter time capture process should run for in seconds.
- click ‘Start’ button.
- perform a specific PSL sign

5.5 Help Link:

Get help from the help document

5.6 Train Model

- Train the ANN model by going to train page and using the captured images.



Train Model

Enter label and retain model:

Figure 5.6 TRAIN SCREEN

To update the dataset and retrain ANN model:

- check if images left after deletion are all valid.
- click ‘populate’ button to update the dataset.
- click ‘retrain’ button to train the ANN model.
- if images are not valid click ‘discard’ button

Chapter 6: Testing

6.1. TEST CASE SPECIFICATION

6.1.1 START DETECTION

Positive Test Case	
ID	TC_START_DETECTION_SUCCESS
Priority	High
Description	To start detecting signs made by user
Reference	FR_01
Users	Real time user
Pre-requisites	● Recognition is not already started
Steps	● Open the application ● Go to detection page ● Click 'Start Detection' button
Input	Button Click
Expected result	recognition started successfully
Status	Tested, passed.

Negative Test Case	
ID	TC_START_DETECTION_FAILURE
Priority	High
Description	To start detecting signs made by user
Reference	FR_01
Users	Real time user
Pre-requisites	● Recognition is not already started
Steps	● Click ‘Start Detection’ button
Input	Button Click
Expected result	does not allow recognition to be restarted
Status	Tested, passed.

6.1.2 RECOGNIZE SIGN

Positive Test Case	
ID	TC_RECOGNIZE _SUCCESS
Priority	High
Description	Recognize the sign made by user and convert it into text and speech
Reference	FR_02
Users	Real time user
Pre-requisites	● Recognition has started
Steps	● Open the application ● Go to detection page ● Click ‘Start Detection’ button ● Make a valid sign in front of the webcam
Input	Valid sign in front of the webcam

Expected result	Correct conversion into text and speech
Status	Tested, passed.

Negative Test Case	
ID	TC_RECOGNIZE _FAILURE
Priority	High
Description	To start detecting signs made by user
Reference	FR_02
Users	Real time user
Pre-requisites	● Recognition has started
Steps	● Open the application ● Go to detection page ● Click ‘Start Detection’ button ● Make an invalid sign in front of the webcam
Input	Invalid sign in front of the webcam
Expected result	Gives no confidence error
Status	Tested, passed.

6.1.3. STOP DETECTION

Positive Test Case	
ID	TC_STOP_DETECTION_SUCCESS
Priority	High
Description	To stopped detection
Reference	FR_03
Users	Real time user
Pre-requisites	● Recognition has been started
Steps	● Open the application ● Go to detection page ● Click ‘Stop Detection’ button

Input	Button Click
Expected result	recognition stopped successfully
Status	Tested, passed.

Negative Test Case	
ID	TC_STOP_DETECTION_FAILURE
Priority	High
Description	To stop detection
Reference	FR_03
Users	Real time user
Pre-requisites	● Recognition has been started
Steps	● Click 'Stop Detection' button
Input	Button Click
Expected result	do nothing
Status	Tested, passed.

6.1.4.ENABLE SPEECH

Positive Test Case	
ID	TC_ENABLE_SPEECH_SUCCESS
Priority	High
Description	Enable speech output
Reference	FR_04
Users	Real time user
Pre-requisites	● 'enable speech' checkbox unchecked

Steps	● Open the application ● Go to detection page ● check the ‘enable speech’ checkbox
Input	check the ‘enable speech’ checkbox
Expected result	speech output enabled successfully
Status	Tested, passed.

Positive Test Case	
ID	TC_DISABLE_SPEECH_SUCCESS
Priority	High
Description	Disable speech output
Reference	FR_05
Users	Real time user
Pre-requisites	● ‘enable speech’ checkbox checked
Steps	● Open the application ● Go to detection page ● uncheck the ‘enable speech’ checkbox
Input	uncheck the ‘enable speech’ checkbox
Expected result	speech output disabled successfully
Status	Tested, passed.

6.1.5. DISPLAY OUTPUT SUCCESS

Positive Test Case	
ID	TC_DISPLAY_OUTPUT_SUCCESS
Priority	High
Description	display recognized sign output on screen
Reference	FR_06
Users	Real time user
Pre-requisites	● Recognition has been started ● output received from recognition module

Steps	● Open the application ● Go to detection page ● Click ‘Start Detection’ button ● Make a valid sign in front of the webcam
Input	Output from recognition module
Expected result	Output displayed successfully
Status	Tested, passed.

6.1.6. PLAY SPEECH SUCCESS

Positive Test Case	
ID	TC_PLAY_SPEECH_SUCCESS
Priority	High
Description	play speech of recognized sign
Reference	FR_07
Users	Real time user
Pre-requisites	● Recognition has been started ● output received from recognition module
Steps	● Open the application ● Go to detection page ● Click ‘Start Detection’ button ● Make a valid sign in front of the webcam
Input	Output from recognition module
Expected result	speech played successfully
Status	Tested, passed.

6.1.7. PLAY SPEECH OUTPUT

Positive Test Case	
ID	TC_SWITCH_WORD_MODE_SUCCESS
Priority	High
Description	Switch to word mode from alphabet mode

Reference	FR_08
Users	Real time user
Pre-requisites	● 'switch to word mode' checkbox unchecked
Steps	● Open the application ● Go to detection page ● check the 'switch to word mode' checkbox
Input	check the 'switch to word mode' checkbox
Expected result	Switched to word mode from alphabet mode successfully
Status	Tested, passed.

6.1.8. SWITCH ALPHABET MODE

Positive Test Case	
ID	TC_SWITCH_ALPHABET_MODE_SUCCESS
Priority	High
Description	Switch to alphabet mode from word mode
Reference	FR_09
Users	Real time user
Pre-requisites	● 'switch to word mode' checkbox checked
Steps	● Open the application ● Go to detection page ● uncheck the 'switch to word mode' checkbox
Input	uncheck the 'switch to word mode' checkbox
Expected result	Switched to alphabet mode from word mode successfully
Status	Tested, passed.

6.1.9. VIEW HELP

Positive Test Case	
ID	TC_VIEW_HELP_SUCCESS
Priority	Medium
Description	View Help Document

Reference	FR_10
Users	Real time user, Learner
Pre-requisites	● nothing
Steps	● Go to main page
Input	Click Help Button
Expected result	Help document shown in new page successfully
Status	Tested, passed.

6.1.10. DISPLAY IMAGE

Positive Test Case	
ID	TC_DISPLAY_IMAGE_SUCCESS
Priority	HIGH
Description	display image of specific sign on screen and prompt user to try making that sign
Reference	FR_11
Users	Learner
Pre-requisites	● nothing
Steps	● Open the application ● Go to learning page
Input	Learning Module
Expected result	Image displayed successfully
Status	Tested, passed.

6.1.11. VALIDATE SIGN

Positive Test Case	
ID	TC_VALIDATE_SIGN_SUCCESS
Priority	HIGH

Description	Validate sign made by user against an image shown to him
Reference	FR_12
Users	Learner
Pre-requisites	● Webcam working.
Steps	● Open the application ● Go to learning page ● Make sign shown on screen in front of webcam.
Input	Valid sign made by user in front of webcam.
Expected result	Success message displayed
Status	Tested, passed.

Negative Test Case	
ID	TC_VALIDATE_SIGN_FALIURE
Priority	HIGH
Description	Validate sign made by user against an image shown to him
Reference	FR_12
Users	Learner
Pre-requisites	● Webcam working.
Steps	● Open the application ● Go to learning page ● Make sign shown on screen in front of webcam.
Input	Invalid sign made by user in front of webcam.
Expected result	Try again message displayed
Status	Tested, passed.

6.1.12. CHANGE SIGN IMAGE

Positive Test Case	
ID	TC_CHANGE_IMAGE_SUCCESS
Priority	HIGH
Description	Show next sign image to user if previous sign was validated
Reference	FR_12
Users	Learner
Pre-requisites	● previous sign was validated
Steps	● Open the application ● Go to learning page ● Make a valid sign
Input	Validation function
Expected result	Image Changed successfully
Status	Tested, passed.

Negative Test Case	
ID	TC_CHANGE_IMAGE_FALIURE
Priority	HIGH
Description	Show next sign image to user if previous sign was validated
Reference	FR_13
Users	Learner
Pre-requisites	● previous sign was validated
Steps	● Open the application ● Go to learning page ● Make a valid sign
Input	Validation function

Expected result	Image Changed successfully
Status	Tested, passed.

6.1.13. SKIP SIGN

Positive Test Case	
ID	TC_SKIP_SIGN_SUCCESS
Priority	HIGH
Description	Skip to next sign if user clicks skip button
Reference	FR_14
Users	Learner
Pre-requisites	● sign has not been validated
Steps	● Open the application ● Go to learning page ● Click ‘Skip’ Button
Input	Click ‘Skip’ Button
Expected result	● Image Changed successfully ● moved to next sign
Status	Tested, passed.

6.1.16. SWITCH CAPTURE

Positive Test Case	
ID	TC_SWITCH_CAPTURE_MODE_SUCCESS
Priority	High
Description	Select either word mode or alphabet mode from ‘select mode’ dropdown
Reference	FR_15
Users	Real time user
Pre-requisites	● Capture has not been started

Steps	● Open the application ● Go to Capture page ● 'select mode' dropdown
Input	'select mode' dropdown
Expected result	mode changed to selected dropdown value
Status	Tested, passed.

6.1.17. SET TIME

Positive Test Case	
ID	TC_SET_TIME_SUCCESS
Priority	High
Description	Enter capture time in seconds, in 'enter time' text field to set time for capture process
Reference	FR_16
Users	user
Pre-requisites	● Capture has not been started
Steps	● Open the application ● Go to Capture page ● 'enter time' text field
Input	'enter time' text field
Expected result	time set successfully
Status	Tested, passed.

6.1.18. DISPLAY CAPTURE IMAGE

Positive Test Case	
ID	TC_DISPLAY_IMAGES_SUCCESS
Priority	High
Description	Display images captured in capture process

Reference	FR_17
Users	Real time user
Pre-requisites	● some images received from capture process
Steps	● Go to Capture page ● Strat Capture
Input	images captured in capture process
Expected result	images displayed on screen
Status	Tested, passed.

6.1.19. DISPLAY DELETED IMAGE

Positive Test Case	
ID	TC_DELETE_IMAGES_SUCCESS
Priority	High
Description	Select images of invalid signs and delete them
Reference	FR_18
Users	Real time user
Pre-requisites	● some images selected to delete
Steps	● Go to Capture page ● Strat Capture ● select images
Input	select images
Expected result	images deleted successfully
Status	Tested, passed.

6.1.20. DATABASE

Positive Test Case	
ID	TC_POPULATE_DATASET_SUCCESS
Priority	High

Description	Click 'populate dataset' button to update the dataset.
Reference	FR_20
Users	Real time user
Pre-requisites	● some images captured in capture process
Steps	● Go to Capture page ● Strat Capture ● Click 'populate dataset' button
Input	Click 'populate dataset' button
Expected result	dataset updated with new captured files
Status	Tested, passed.

6.1.21. TRAIN MODEL

Positive Test Case	
ID	TC_TRAIN_MODEL_SUCCESS
Priority	High
Description	Click 'Train Model' button to train ANN model with updated dataset
Reference	FR_21
Users	Real time user
Pre-requisites	● no requirements
Steps	● Go to Capture page ● Strat Capture ● Click 'Train Model' button
Input	Click 'Train Model' button
Expected result	ANN model trained with updated dataset
Status	Tested, passed.

6.2. BLACK BOX TEST CASES

6.2.1. EQUIVALENCE PARTITIONS (EP)

START DETECTION

Description	Using this test case, the sign detection process can be checked		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**
It will start detecting signs and displaying alphabets.	1. Start Open-Pose. 2. Start Detection.	1. Start Open-Pose.	It will not detect signs and display alphabets.

SPEECH CHECKING

Description	Using this test case, the speech of sign can be checked		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**
It will speak the alphabet which you made a sign of.	Make a sign of any alphabet if it exists in the database.	Make any sign randomly.	It will speak of an alphabet which is close to that sign.

STOP DETECTION

Description	Using this test case, the sign detection process can be checked		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**
It will stop detecting signs and displaying alphabets.	1. Start Open-Pose. 2. Start Detection. 3. Press "Esc" to exit from Open-Pose.	1. Start Open-Pose. 2. Start Detection.	It will not stop detecting signs and displaying alphabets.

DETECTION

Description	Using this test case, the detection process can be checked		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**
It will perform Detection function.	1. Select Detection Tab.	1. Select Capture Tab. 2. Select Learn Tab.	It will perform the Capture/Training function.

LEARNING

Description	Using this test case, the Learning process can be checked		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**
It will perform the Leaning function.	1. Select Learning Tab.	1. Select Capture Tab. 2. Select Detection Tab.	It will perform the Capture/Detection function.

CAPTURE

Description	Using this test case, the Capture process can be checked		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**
It will perform the Capture function.	1. Select Capture Tab.	1. Select Detection Tab. 2. Select Learn Tab.	It will perform Detection/Learning function.

HELP

Description	Using this test case, the Help process can be checked		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**

It will guide you about the functions.	1. Select "?" Tab For help.	1. Select Detection Tab. 2. Select Learn Tab. 3. Select Capture Tab.	It will not guide you about the functions instead it will do the function.

CAPTURE DATASET

Description	Using this test case, the Capturing dataset process can be checked.		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**
It will start Capturing Dataset.	1. Insert the seconds you want to run the process.	1. Start without inserting seconds. 2. Insert the alphabet instead of numbers.	It will not start the Capturing Dataset process.
Description	Using this test case, the Label Assignment process in Capturing Dataset can be checked.		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**
It will store/delete the captured dataset with the label you assigned.	1. Starts Capturing process. 2. Press 'Y' for storing captured dataset 'N' for deleting captured dataset.	1. Starts Capturing process. 2. Press something other than 'Y' or 'N'	It will not store/delete the captured dataset with the label you assigned.

LEARNING

Description	Using this test case, the Learning process can be checked.		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**

It will start Showing you sign images for you to learn to make them correctly.	1. Select Learn Tab. 2. Click 'Start Learning'	1. Select Learn Tab.	It will not start the process of learning.

SKIPPING SIGN

Description	Using this test case, the Skipping Sign in Learning Process can be checked.		
Valid Output	**Valid Input**	**Invalid Input**	**Invalid Output**
It will Skip the current sign and will show you the next sign.	1. Select Learn Tab. 2. Click 'Start Learning' 3. Skip Sign.	1. Select Learn Tab. 2. Click 'Start Learning'	It will not Skip any sign until you make it correct.

6.2.2. USE CASE TESTING

We have tested the system functionalities against each use case

6.3. WHITE BOX TEST CASES

We have tried inside code of each part and it was made sure that the code composed is effective in using different assets of the framework like memory or the using of information and result. The tests were mechanized utilizing py Test

6.3.1 CYCLOMATIC COMPLEXITY

Cyclomatic complexity is a estimation to decide the security and level of trust in a program. It estimates the quantity of straightly autonomous ways through a program module. Programs with lower Cyclomatic intricacy are more clear and safer to change.

6.4. PERFORMANCE TESTING

Performance testing is a non-functional software testing technique that determines how the stability,speed, scalability, and responsiveness of an application holds up under a given workload.

6.5. SYSTEM TESTING

System testing is testing led on a total coordinated framework to assess the framework's consistence with its predefined necessities. Framework testing takes, as its feedback, each of the incorporated parts that have passed joining testing.

6.6. REGRESSION TESTING

Regression Testing is **defined** as a type of software testing to confirm that a recent programor code change has not adversely affected existing features.

Chapter 7: Tools and Techniques

7.1. LANGUAGES

1. For Back-End, we have used Python 2.

For Front-End we have used:

a. HTML
b. CSS
c. JavaScript

7.2. APPLICATIONS AND TOOLS

- We have used Spyder IDE for backend development due to its highly interactive environment.
- For Front-End we have used Dreamweaver

7.3. LIBRARIES AND EXTENSIONS

- Open-Pose Library for extracting skeletal key points
- Eel
- Bootstrap
- jQuery

Chapter 8: Summary and Conclusion

8.1. SUMMARY

A recognizable measure of hard of hearing local area exists in Pakistan. This framework is created to help these individuals by changing over Pakistan Communication via gestures (PSL) into Urdu text and discourse. The principle objective is to work with an enormous populace of hearing-disabled people and making them a fundamental piece of society. Overall endeavors have been made to help the hard of hearing local area in speaking with non-underwriters yet the greater part of the current frameworks either utilize particular sensors or has low execution. The proposed framework works by taking pictures of a client making a specific sign through webcam and utilizations skeletal following joined with AI to recognize what sign is made by the individual. As of now, there is no openly accessible dataset for Pakistan Communication through signing. So we chose to make our own dataset for PSL; for that reason we made a mechanized dataset catching framework that catches pictures of client for indicated time, separate the central issues from pictures utilizing Open-Posture, erases the pictures where Open-Posture accurately distinguishes under 10 central issues from hand, plot the skeleton of outstanding pictures and show them to the client, the client then, at that point, chooses the plotted pictures that doesn't match the sign he was making and erases them. The client is then approached to enter the name for residual pictures then these pictures are put away by the framework in an envelope named as the worth of mark entered by client.

We have gathered the information of 37 Urdu letter sets from 9 unique individuals and 12 Urdu words from 4 distinct individuals. We have removed skeletal central issues of every main informative element as JSON documents by utilizing Open-Posture. On account of letters in order, just the 21 central issues of the right hand were chosen and standardized by scaling and moving them to the middle while keeping the first state of the hand. For Urdu language words, 13 central issues from chest area were chosen alongside 42 central issues of both ways hand, propositions central issues were standardized by scaling the body and each hand, and by moving the body to focus, passed close by to left wrist point and right hand to right wrist point, while keeping the first state of the body and the hands. In communication via gestures, a similar sign can vary minutely starting with one case then onto the next in any event, when it is a similar individual making the sign. For limit pivot and positional difference of signs, we chose to orchestrate new information from the first information. Along these lines, if there should be an occurrence of letter sets, we turned the right hand to +20 and - 20 degrees and store them as 2 distinct places of same sign henceforth limiting revolution fluctuation that existed in the letter set signs. On account of words the hands were moved 80 pixels up, down, left and right while keeping the normal stance of arms utilizing Opposite

Kinematics(IK) and pivoting the hands to +20 and - 20 degrees and putting away them as 12 distinct places of similar signs henceforth limiting revolution and positional change. We prepared two Fake Neural Organization (ANN) models one for letter set acknowledgment and one for word acknowledgment. To stay away from overfitting in view of the restricted information we considered a less complex ANN model, that performed shockingly well for the two words and letter sets. In Urdu letter set acknowledgment model we took care of 42 (21x and 21y) input exhibit into the main layer of ANN, following this we feed into stowed away layer with 64 units, at long last we send everything through the result layer with sigmoid enactment,

bringing about a 36-component vector. For Urdu word acknowledgment, a similar model was utilized however the info layer was taken care of with 110 (42+42+26) input cluster. The letter set acknowledgment model accomplished 99.4% precision on test information and ~93% exactness on continuous contributions through webcam, while the word acknowledgment model has accomplished 99.2% precision on test information and ~88% exactness on ongoing contributions through webcam. Because of the positional fluctuation of signs made by various clients the exactness of continuous information is not exactly the test information.

We have likewise presented an intuitive PSL learning framework. This framework works by showing the client various pictures of PSL signs and provoking him/her to make that sign, the framework then, at that point, checks whether or not the client have made the right sign and shows a message appropriately.

8.2. Conclusion

The generally speaking PSL making an interpretation of framework is parted into 3 fundamental modules, identification module, learning module and catch module. The recognition module has two modes one for letter set discovery and other for word location. The learning module is utilized for intelligent PSL learning and the catch module is utilized for adding new information to dataset.

Because of absence of huge previous PSL dataset we have presented another PSL dataset that comprises of human body central issues of 9 distinct individuals making indications of 37 Urdu letter sets and 4 unique individuals making indications of 12 Urdu words. Therefore, the dataset contains 4500 Urdu letters in order tests and 2000 Urdu word tests.

Different AI models were thought about yet we viewed ANN as generally reasonable for this errand. We were astonished to find that the ideal ANN configuration came from a basic engineering rather than a convoluted one. The last letters in order model plan comprises of an info layer with 42 units, a secret layer with 64 units and a result layer with sigmoid initiation bringing about 36-component vector. The word acknowledgment model contains an info layer with 110 units a secret layer with 64 units and a result layer bringing about 12-component vector. The letter set acknowledgment model accomplished 99.4% exactness on test information and ~93% precision on constant contributions through webcam, while the word acknowledgment model has accomplished 99.2% exactness on test information and ~88% precision on continuous data sources.

Chapter 9: LESSONS LEARNT AND FUTURE ENHANCEMENTS

As Pakistan Gesture based communication contains great many words it is absurd to expect to make an exhaustive framework that covers this multitude of signs at this level. We have given a framework that can change over all Urdu letters in order and a few Urdu words into text and discourse and have additionally given a structure to adding more words to the framework. The proposed framework has given extremely high precision rate for PSL acknowledgment. But there is still room for improvement. Following guidelines are suggested for the future enhancements:

- The system can be further extended to allow recognition of dynamic PSL signs.
- The dataset for PSL can be extended further for more vocabulary.
- Natural Language Processing (NLP) techniques can be used to formulate meaningful sentences from PSL words.
- The text and speech conversion to PSL animation can be added to allow two way communication between signer and non-signer.

9.1 Appendix A

snatching - spreading of the fingers from one another expectant arrangement
characterization of a worldly info arrangement before the finishing of that succession
Auslan - the gesture based communication of the Australian Hard of hearing local area backpropagation - a broadly utilized regulated preparation calculation which changes the conduct of a neural organization by changing its loads on
the premise of preparing models (see Segment 5.2.4)
Backpropagation Through Time - an augmentation of the backpropagation calculation which permits preparing of intermittent neural organizations (see Segment 6.4.2) co-explanation

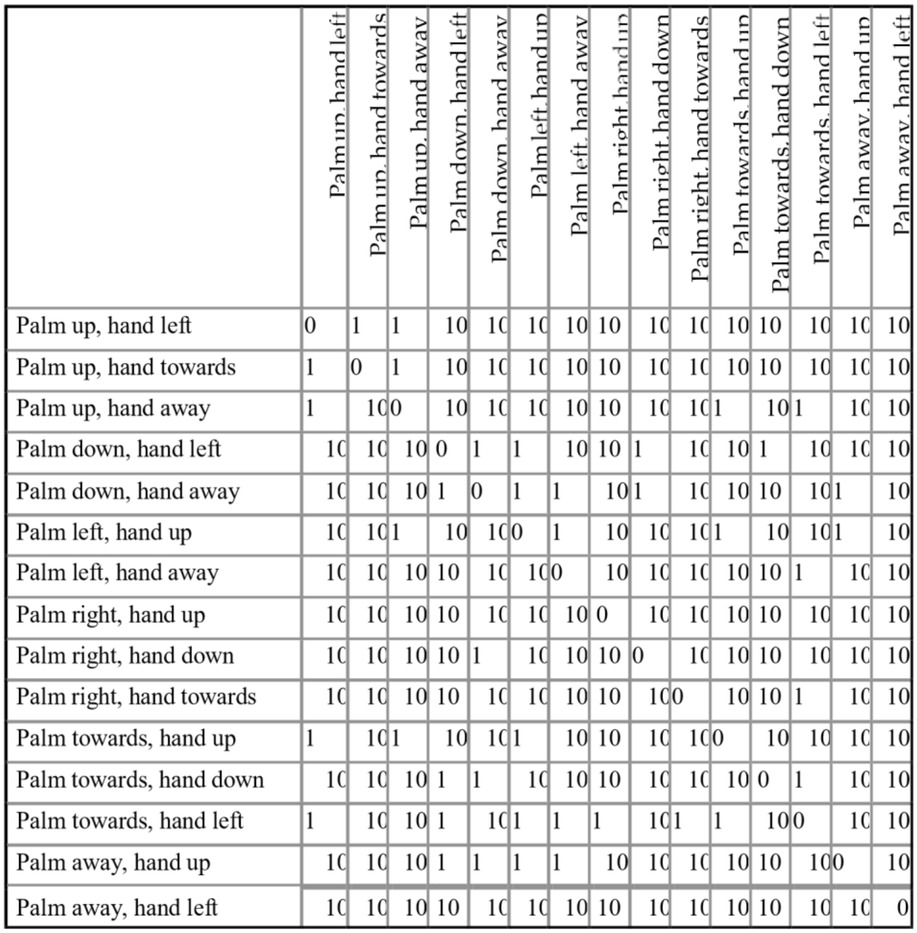
- the way where the development of a word or sign is adjusted by the former and ensuing words or signs co-verbal motion - a motion performed all the while with discourse to explain or change the importance of that discourse
council framework - a strategy for consolidating the results of various neural organizations to further develop speculation execution association crossing - a unit of estimation of the time taken to prepare an organization, comparing to the utilization of a load on an association in an augmentation activity
Cyber G love - an instrumented glove produced by Virtual
Innovations which estimates the level of flex of the joints of the fingers and hand repetitive information -information in which the requesting of values is repetitive in nature, like the hour of day, or compass course
hard of hearing/Hard of hearing - in uncapitalised structure, the actual condition hearing disabled; in promoted structure, the social parts of being hard of hearing distal interphalangeal - the peripheral joint of each finger fingerspelling - a technique for spelling words letter-by-letter utilizing hand motions

speculation - the capacity of an example acknowledgment framework to perform well on information which it has not been prepared on sparkle the English word or expression relating to a sign handshape

an unmistakable design of the joints of the hand

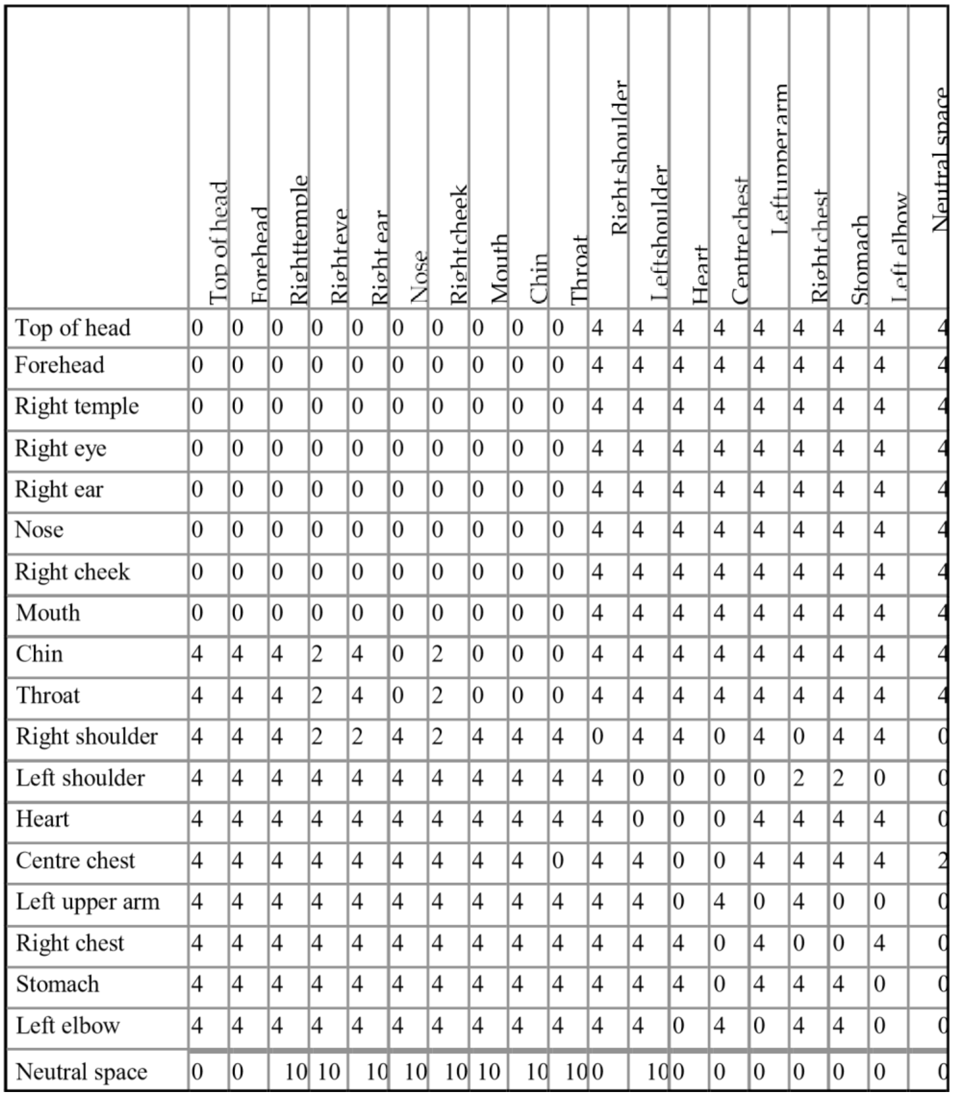
9.2 Appendix B

Table Heuristic distance measure for the orientation feature.



	Palm up, hand left	Palm up, hand towards	Palm up, hand away	Palm down, hand left	Palm down, hand away	Palm left, hand up	Palm left, hand away	Palm right, hand up	Palm right, hand down	Palm right, hand towards	Palm towards, hand up	Palm towards, hand down	Palm towards, hand left	Palm away, hand up	Palm away, hand left
Palm up, hand left	0	1	1	10	10	10	10	10	10	10	10	10	10	10	10
Palm up, hand towards	1	0	1	10	10	10	10	10	10	10	10	10	10	10	10
Palm up, hand away	1	10	0	10	10	10	10	10	10	10	1	10	1	10	10
Palm down, hand left	10	10	10	0	1	1	10	10	1	10	10	1	10	10	10
Palm down, hand away	10	10	10	1	0	1	1	10	1	10	10	10	10	1	10
Palm left, hand up	10	10	1	10	10	0	1	10	10	10	1	10	10	1	10
Palm left, hand away	10	10	10	10	10	10	0	10	10	10	10	10	1	10	10
Palm right, hand up	10	10	10	10	10	10	10	0	10	10	10	10	10	10	10
Palm right, hand down	10	10	10	10	1	10	10	10	0	10	10	10	10	10	10
Palm right, hand towards	10	10	10	10	10	10	10	10	10	0	10	10	1	10	10
Palm towards, hand up	1	10	1	10	10	1	10	10	10	10	0	10	10	10	10
Palm towards, hand down	10	10	10	1	1	10	10	10	10	10	10	0	1	10	10
Palm towards, hand left	1	10	10	1	10	1	1	1	10	1	1	10	0	10	10
Palm away, hand up	10	10	10	1	1	1	1	10	10	10	10	10	10	0	10
Palm away, hand left	10	10	10	10	10	10	10	10	10	10	10	10	10	10	0

Table Heuristic distance measure for the hand location feature.



	Top of head	Forehead	Righttemple	Righteye	Right ear	Nose	Rightcheek	Mouth	Chin	Throat	Right shoulder	Leftshoulder	Heart	Centre chest	Leftupper arm	Right chest	Stomach	Left elbow	Neutral space
Top of head	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4
Forehead	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4
Right temple	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4
Right eye	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4
Right ear	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4
Nose	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4
Right cheek	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4
Mouth	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4
Chin	4	4	4	2	4	0	2	0	0	0	4	4	4	4	4	4	4	4	4
Throat	4	4	4	2	4	0	2	0	0	0	4	4	4	4	4	4	4	4	4
Right shoulder	4	4	4	2	2	4	2	4	4	4	0	4	4	0	4	0	4	4	0
Left shoulder	4	4	4	4	4	4	4	4	4	4	4	0	0	0	0	2	2	0	0
Heart	4	4	4	4	4	4	4	4	4	4	4	0	0	0	4	4	4	4	0
Centre chest	4	4	4	4	4	4	4	4	4	0	4	4	0	0	4	4	4	4	2
Left upper arm	4	4	4	4	4	4	4	4	4	4	4	4	0	4	0	4	0	0	0
Right chest	4	4	4	4	4	4	4	4	4	4	4	4	4	0	4	0	0	4	0
Stomach	4	4	4	4	4	4	4	4	4	4	4	4	4	0	4	4	4	0	0
Left elbow	4	4	4	4	4	4	4	4	4	4	4	4	0	4	0	4	4	0	0
Neutral space	0	0	10	10	10	10	10	10	10	10	0	10	0	0	0	0	0	0	0

9.3 PROJECT REPORT

9.3.1Abdullah Report

The aim of the project how to convert the signs into language with cheapest and best result so I started from image segmentation

Introduction of Image Segmentation:

Image segmentation is normally used to find articles and limits (lines, bends, and so on) in pictures. All the more definitively, picture division is the most common way of allocating a name to each pixel in a picture to such an extent that pixels with a similar name share specific qualities.

How Image Segmentation works:

We can gap or parcel the picture into different parts called fragments. It's anything but smart to handle the whole picture simultaneously as there will be districts in the picture which don't contain any data. By separating the picture into fragments, we can utilize the significant portions for handling the picture. That, more or less, is the way picture division works.

A picture is an assortment or set of various pixels. We assemble the pixels that have comparative credits utilizing picture division. Take a moment to go through the below visual (it'll give you a practical idea of image segmentation):

Why do we need of Image Segmentation:

Malignant growth has for some time been a dangerous disease. Indeed, even in the present period of innovative headways, disease can be deadly on the off chance that we don't distinguish it at a beginning phase. Identifying malignant cell(s) as fast as conceivable might possibly save a great many lives.
The state of the destructive cells assumes a fundamental part in deciding the seriousness of the disease. You may have assembled the pieces - object location won't be exceptionally helpful here. We will just produce bouncing boxes which won't help us in distinguishing the state of the cells. Picture Division procedures have a Monstrous effect here. They assist us with moving toward this issue in a more granular way and come by more significant outcomes. A shared benefit for everybody in the medical care industry

Subsequent to realizing of what is picture division and how it functions I began further review on the kinds of Picture Division

Different types of image segmentation:

Region-based Segmentation
One straightforward method for portioning various items could be to utilize their pixel values. A significant highlight note - the pixel values will be different for the items and the picture's experience assuming there's a sharp difference between them.

For this situation, we can set a limit esteem. The pixel values falling underneath or over that edge can be grouped likewise (as an article or the foundation). This method is known as Limit Division. To partition the picture into two areas (item and foundation), we characterize a solitary limit esteem. This is known as the worldwide edge.
In the event that we have various items alongside the foundation, we should characterize different limits. These limits are on the whole known as the neighborhood edge.

Edge Detection Segmentation:

Which isolates two articles in a picture? There is consistently an edge between two neighboring areas with various grayscale values (pixel values). The edges can be considered as the intermittent neighborhood highlights of a picture.
We can utilize this brokenness to recognize edges and thus characterize a limit of the item. This assists us in recognizing the states of various articles with introducing in a given picture. Presently the inquiry is how might we distinguish these edges? This is the place where we can utilize channels and convolutions.

Image Segmentation based on Clustering:

This thought may have come to you while learning about picture division. Would we be able to utilize bunching procedures to partition pictures into sections? We unquestionably can!

In this part, we'll get an instinct of what grouping is (it's great all the time to change specific ideas!) and how we can utilization of it to fragment pictures.

Bunching is the errand of isolating the populace (informative elements) into various gatherings, to such an extent that elements in similar gatherings are more like different elements in that equivalent gathering than those in different gatherings. These gatherings are known as groups.

Quite possibly the most ordinarily utilized bunching calculation is k-implies. Here, the k addresses the quantity of groups (not to be mistaken for k-closest neighbor). We should see how k-implies functions:

1. First, haphazardly select k beginning bunches

2. Randomly appoint every information highlight any of the k bunches

3. Calculate the focuses of these bunches

4. Calculate the distance of the relative multitude of focuses from the focal point of each group

5. Depending on this distance, the focuses are reassigned to the closest bunch

6. Calculate the focal point of the recently framed bunches

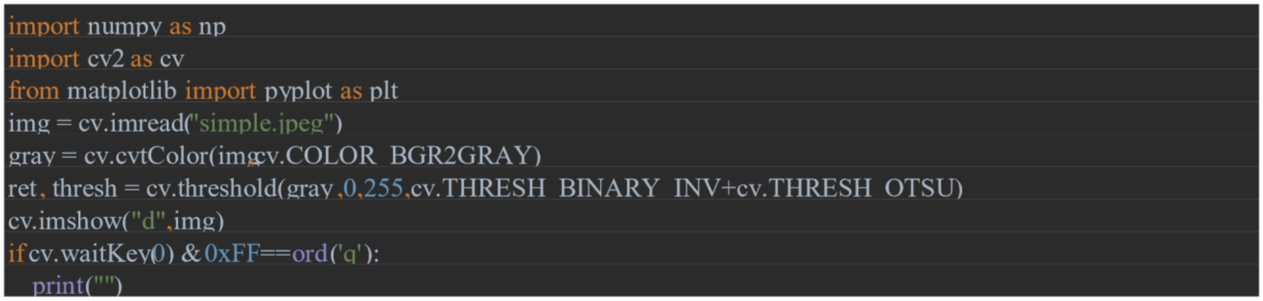
7. Finally, rehash steps (4), (5) and (6) until either the focal point of the groups doesn't change or we arrive at the set number of cycles

The vital benefit of utilizing k-implies calculation is that it is straightforward and straightforward. We are allotting the focuses to the bunches which are nearest to them

Mask R-CNN:

Information researchers and specialists at Facebook man-made intelligence Exploration (FAIR) spearheaded a profound learning engineering, called Veil RCNN, that can make a pixel-wise cover for each article in a picture. This is a truly cool idea so track with intently!
Veil R-CNN is an expansion of the well known Quicker R-CNN object identification engineering. Cover RCNN adds a branch to the generally existing Quicker R-CNN yields. The Quicker R-CNN technique creates two things for each article in the picture
Its class
The bouncing box organizes
Cover R-CNN adds a third branch to this which yields the article veil too. Investigate the underneath picture to get an instinct of how Cover R-CNN chips away at within:
We accept a picture as info and pass it to the ConvNet, which returns the element map for that picture
Area proposition organization (RPN) is applied on these element maps. This profits the article recommendations alongside their objectness score
A return on initial capital investment pooling layer is applied on these recommendations to cut down every one of the proposition to a similar size
At long last, the proposition are passed to a completely associated layer to group and result the jumping boxes for objects. It additionally returns the veil for every proposition
Veil R-CNN is the present status of-the-workmanship for picture division and runs at 5 fps.

Subsequent to learning of picture division began reasonable on the picture division with one kind of picture division and that type was Area based division
For picture division I introduce a translator named Pycharm people group release 2021.12
In this District based division by utilizing opencv and numpy as np and by bringing in the cv2 as cv handily applied the Area put together division with respect to a picture of an individual by the accompanying lines of code



```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt
img = cv.imread("simple.jpeg")
gray = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
ret, thresh = cv.threshold(gray,0,255,cv.THRESH_BINARY_INV+cv.THRESH_OTSU)
cv.imshow("d",img)
if cv.waitKey(0) & 0xFF==ord('q'):
   print("")
```

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt
img = cv.imread("simple.jpeg")
gray = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
ret, thresh = cv.threshold(gray,0,255,cv.THRESH_BINARY_INV+cv.THRESH_OTSU)
cv.imshow("d",img)
if cv.waitKey(0) & 0xFF==ord('q'):
   print("")
```

After the segmentation process moved on to the how a camera recognize a face in a image After installing I tried manycodes of image recognition with different algorithms and libraries and installing pip but I failed in image recognition process

But after trying with different techniques and libraries many time the image recognition process become successful withthe following lines of code

```
import cv2

# Load the cascade face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# To capture video from webcam. cap = cv2.VideoCapture(cv2.CAP_DSHOW+0)
# To use a video file as input
# cap = cv2.VideoCapture('filename.mp4')

while True:
   # Read the frame
```

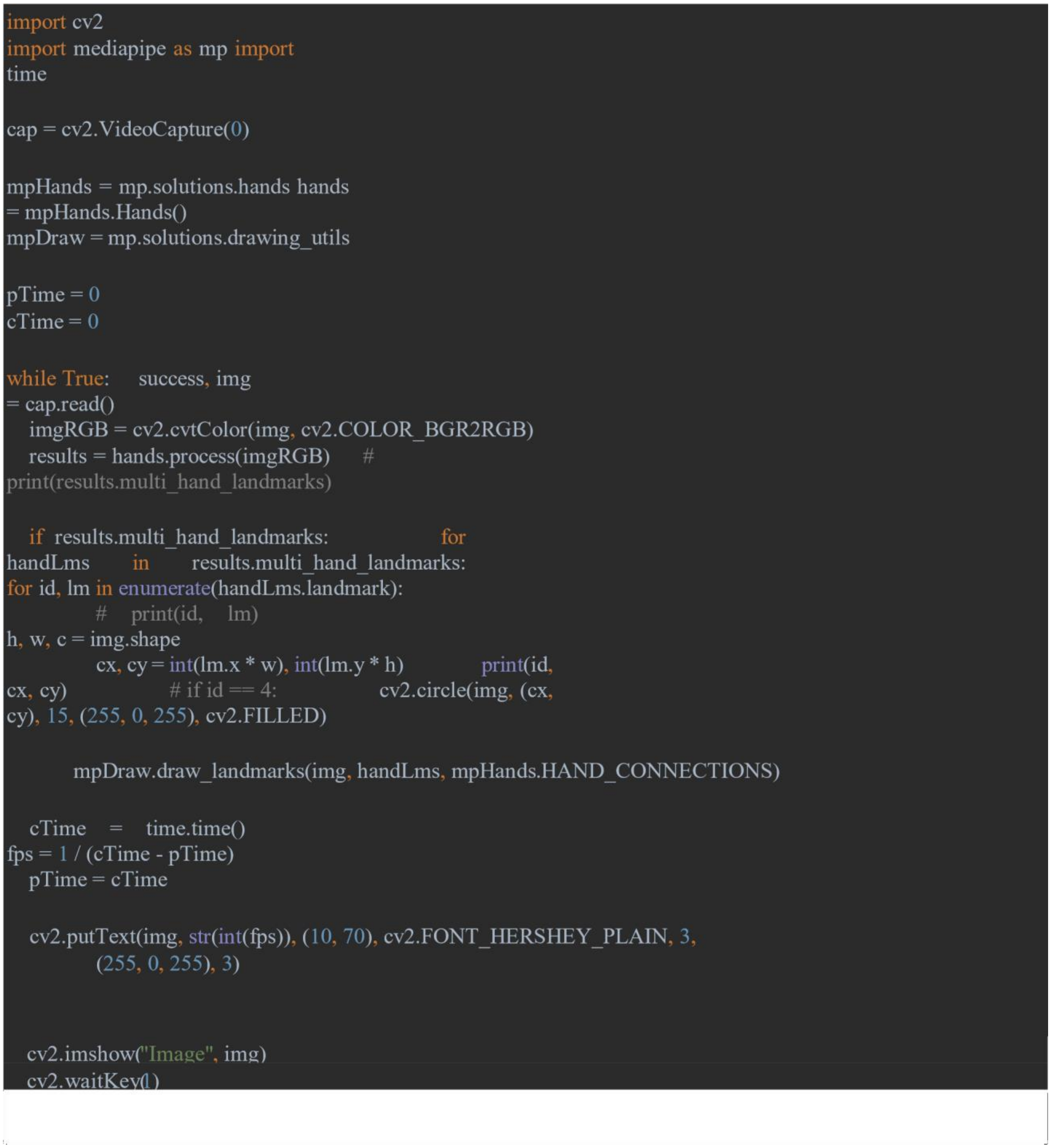
```
    ret, img = cap.read()

    # Convert to grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Detect the faces
    faces =             face_cascade.detectMultiScale(gray,             1.1,        4)
print(faces)

    # Draw the rectangle around each face      #for (x, y, w, h) in faces:
    #  cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

    # Display
    cv2.imshow('img', img)

    # Stop if escape key is pressed      k = cv2.waitKey(1) & 0xff     if k
== 27:         break

# Release the VideoCapture object cap.release()
```

After the face recognition in a image moved further towards the hand tracking process

For hand tracking I imported cv2 and also many other libraries like image.rar etc and before working on hand trackingI watched many videos of hand tracking on youtube that how it works

Because I face many errors and then I solve that errors by searching on the stackoverflow

And after that with opencv and other libraries it works with the following lines of code



```
import cv2
import mediapipe as mp import
time

cap = cv2.VideoCapture(0)

mpHands = mp.solutions.hands hands
= mpHands.Hands()
mpDraw = mp.solutions.drawing_utils

pTime = 0
cTime = 0

while True:     success, img
= cap.read()
   imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
   results = hands.process(imgRGB)     #
print(results.multi_hand_landmarks)

   if results.multi_hand_landmarks:                  for
handLms       in       results.multi_hand_landmarks:
for id, lm in enumerate(handLms.landmark):
            #    print(id,    lm)
h, w, c = img.shape
            cx, cy = int(lm.x * w), int(lm.y * h)            print(id,
cx, cy)               # if id == 4:                  cv2.circle(img, (cx,
cy), 15, (255, 0, 255), cv2.FILLED)

         mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS)

   cTime    =    time.time()
fps = 1 / (cTime - pTime)
   pTime = cTime

   cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT_HERSHEY_PLAIN, 3,
            (255, 0, 255), 3)



   cv2.imshow("Image", img)
   cv2.waitKey(1)
```

9.3.2 Ammar Report

TRACKING: -

Tracking is associating target objects in consecutive video frame. Sequential flow of object tracking, objectdetection, object identification and it's behavior completes the process framework of tracking.

KALMAN FILTER: -
Assessor which utilizes estimations noticed, additional time to deliver gauges. Forecast in object future area, numerous item following.
Kalman channels are utilized to ideally gauge the factors of interests when they can't be estimated straightforwardly, however an aberrant estimation is accessible. They are likewise used to track down the best gauge of states by joining estimations from different sensors within the sight of commotion. Kalman channels are great for frameworks which are ceaselessly evolving. They enjoy the benefit that they are light on memory (they don't have to keep any set of experiences other than the past state), and they are extremely quick, making them appropriate for continuous issues and installed frameworks.

MOTION SEGEMENTATION: -

Motion Segmentation is the task of distinguishing the autonomously moving objects (pixels) in the video and isolating them from the foundation motion. The region of the picture that do
not register well have a place with the moving articles.

BACKGROUN SUBTRACTION: -

Background subtraction is any strategy which permits a picture's frontal area to be extricated for additional handling (object acknowledgment etc.).Foreground recognition isolates closer view from foundation in light of

these progressions occurring in the forefront.

Closer view identification is one of the significant undertakings in the field of PC vision and picture handling whose point is to recognize changes in picture arrangements. Foundation deduction is any strategy which permits a picture's frontal area to be removed for additional handling.

TEMPORAL DIFFERENCE (TD): -

Temporal difference (TD) learning is a way to deal with figuring out how to anticipate an amount that relies upon future upsides of a given sign. The name TD gets from its utilization of changes, or contrasts, in expectations throughout progressive time steps to drive the learning system

OPTICAL FLOW: -

Optical flow, or motion estimation, is a fundamental method of working out the movement of picture forces, which might be attributed to the movement of items in the scene Optical-stream strategies depend on processing evaluations of the movement of the picture forces over the long run in a video.

OBJECT TRACKING: -

Object tracking is an application of deep learning where the program takes an underlying arrangement of article location and fosters an extraordinary recognizable proof for every one of the underlying discoveries and afterward tracks the identified items as they move around outlines in a video

POINT TRACKING: -

Detected objects are represented by points across the frame. Capable of tracking small objects only, Pointtracking is difficult due to errors, false detection of an object.

MULTIPLE HYPOTHESIS TRACKING: -

It is an iterative strategy, mix of a few edges together. Emphasis starts with a bunch of existing track speculation. An expectation of article movement in succeeding edge is produced using every speculation

KERNAL BASED TRACKING: -

Computes the moving object from one frame to another object motion is always in parametric motion

PARTICLE FILTER: -

Tracks non-linear, non-gaussian moving objects

SILHOUETTE APPROACH: -

It’s aim is to find the object region by means of an object model.

Object having shapes e.g hands, heads, shoulders are not defined by geometric shapes. Silhouette approach gives perfect description for them. The model verifies the object in each frame. Model can be represented inform of colour histogram, object edges, contour.

OPENCV: -

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV-Python utilizes Numpy, which is a profoundly improved library for mathematical activities with a MATLAB-style language structure. All the OpenCV cluster structures are changed over to and from Numpy exhibits. OpenCV (Open Source PC Vision Library) is an open source PC vision and AI programming library. OpenCV was worked to give a typical foundation to PC vision applications and to speed up the utilization of machine insight in the business items

OPEN-POSE: -

Open-Pose is a Real-time multiple-person detection library, which shows keypoints on body

IMAGE PROCESSING: -

Image processing is a method to perform certain procedure on a picture, to get an upgraded picture or to extricate some valuable data from it. It is a kind of sign handling wherein input is a picture and result might be picture or attributes/highlights related with that picture.

NATURAL LANGUAGE PROCESSING: -

Natural language processing is a subfield of etymology, software engineering, and man-made reasoning worried about the associations among PCs and human language, specifically how to program PCs to process and break down a lot of normal language information. Normal language handling (NLP) alludes to the part of software engineering and all the more explicitly, the part of computerized reasoning or artificial intelligence worried about enabling PCs to get text and verbally expressed words similarly people can

ARTIFICIAL INTELLIGENCE (AI): -

Artificial intelligence (AI) is the ability of a computer or a robot constrained by a PC to take care of assignments that are typically finished by people since they require human knowledge and wisdom.

MACHINE LEARNING: -

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

CNN ALGORITHM :-

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects inthe image and be able to differentiate one from the other.

R-CNN ALGORITHM: -

R-CNNs (Region-based Convolutional Neural Networks) are a family of machine learning models used in computer vision and image processing.

9.3.3 Haider Ali Report

TASK 1: Recognize Tone (RED, GREEN, BLUE) THROUGH WEBCAME IN PYTHON Utilizing OPEN CV.

What is Shading Location?

ANS: Shading identification is the most common way of distinguishing the name of any tone. Basic isn't it? All things considered, for people this is a very simple errand yet for PCs, it isn't clear. Natural eyes and cerebrums cooperate to make an interpretation of light into shading. Light receptors that are available in our eyes send the sign to the cerebrum. Our mind then, at that point, perceives the shading. Since adolescence, we have planned specific lights with their shading names. We will utilize the fairly same procedure to distinguish shading names.

Method:-

Import the cv2 and Num-Py modules

Catch the webcam video utilizing the cv2.VideoCapture (0) strategy. Show the current casing utilizing the cv2.imshow () technique.

Run some time circle and take the current casing utilizing the read () technique. Take the red, blue and green components and store them in a rundown.

Process the normal of each rundown.

which-at any point normal has the best worth, show that tone

TASK 2: Track down THECOLOR NAME AND R-G-B Worth OF Each Tone FROM THE Picture Data sources

The component "Shading Name Locater" gives the most widely recognized names of a shading. It observes shading names for 3 sorts of info:

Shading name from a picture or a photograph Shading name from a hex or a RGB code

Shading name from a shading picker Transferring Picture

To observe the name of a shading in a picture, the cloud symbol can be utilized to transfer or snap a picture.

When the picture is stacked, tapping on the picture regions will start the shading name distinguishing proof. You can utilize the left mouse snap to flip among focusing and delivering the objective.

Since lighting conditions unequivocally influence the tones in a picture, it is prescribed to take pictures in normal light to get the most agent shading names.

Shading Codes

The ArtyClick Shading Name Locater can be utilized to observe shading names from hex or RGB codes. The accompanying shading codes are upheld:

Hex (for example "#FF0000" or "#FFF")

RGB (for example "RGB(255,0,0)")

The upheld RGB codes compare to the 24-bit framework where every part goes somewhere in the range of 0 and 255 (8-cycle encoding)

Shading Tone

TASK 3: REMOVE (GREEN AND YELLOW) COLOR FROM THE IMAGE

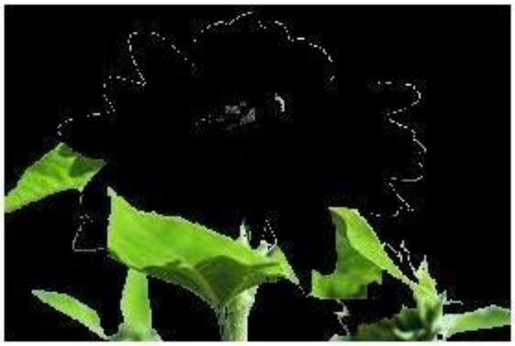
I simply need to identify just green items from a picture which caught in regular habitat.
How to characterize it? Since in here I need to pass the edge esteem suppose 'x', by utilizing this x I need to get just green shading objects in to one color(white) others are should show up in another colour(black).

Convert to HSV color-space,

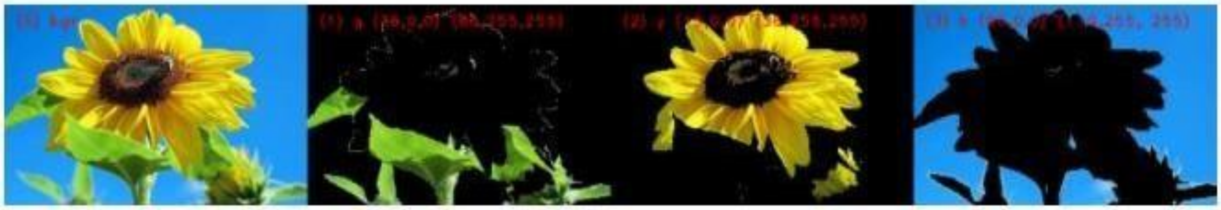
Use cv2.inRange(hsv, hsv_lower, hsv_higher) to get the green mask. We use the range (in hsv): (36,0,0) ~ (86,255,255) for this sunflower.

The source image:

The masked green regions:



More steps:



TASK 4: DETECT SKIN COLOR FROM IMAGE IN PYTHON

Skin Detection Using Open-CV Python What is Skin Detection?

Interaction of finding skin-shaded pixels and locales in a picture or a video.

Frequently utilized as a signal for recognizing, limiting and noticing targets containing skin(like faces and hands in a picture)

Assumes a significant part in human movement investigation and face identification.

(A skin classifier characterizes a choice limit of the skin shading class in the shading space in view of a preparation data set of skin-hued pixels)

Human Skin

The shade of human skin is made by a mix of crimson and melanin (yellow, brown).

Skin colors lie between these two outrageous tones and are to some degree soaked.

The human skin is a negligible part of the real shading shape, around 0.25 % of the absolute tones With the exception of incredibly shaggy subjects, which are uncommon, skin has just low-abundancy surface Skin recognition methods

Includes the definition of an effective numerical model to address the skin tone

conveyance.

Range based o change a given pixel into a suitable shading space o skin classifier to mark the pixel whether it is a skin or non skin pixel.

Histogram backprojection o histogram is a range of power repartition. A rundown that contains the quantity of pixels for every conceivable worth of pixel.

YCrCb Shading Space

Encoded nonlinear RGB signal, generally utilized in video coding and picture pressure work.

built as a weighted amount of the RGB values, and two shading contrast values Cr and Cb that are shaped by taking away luma from RGB red and blue parts16 to 235 for Y, 16 to 240 for Cb and Cr

Skin pixels form a compact cluster in the Cb-Cr plane.

\displaystyle Y = 0.299R + 0.587G + 0.114B$Y=0.299R+0.587G+0.114B$

\displaystyle C_r = R - Y$Cr=R-Y$ \displaystyle C_b = B- Y$Cb=B-Y$

Transformation simplicity and explicit separation of luminance and chrominance componentsmakes this colorspace attractive for skin color modelling

References

1. Statistics, D. (2015). Deaf Statistics. Pakistan Association of Deaf. Retrieved 24 October 2018, from http://padeaf.org/deaf-statistic/

2. Waqar, K. (2014). Disability: Situation in Pakistan (1st ed.). Karachi: Right to Education Pakistan. Retrieved from http://www.itacec.org/document/gaw/gaw2014/2.%20Disability%20Pages%202.pdf

3. Rini Akmeliawatil, Melanie PO-Leen Ooi, et al," RealTime Malaysian Sign Language Translation using Color Segmentation and Neural Network. Instrumentation and Measurement Technology Conference Warsaw, Poland.IEEE.1-6, 2007.

4. Nicholas Born, "Senior Project Sign Language Glove", ELECTRICAL ENGINEERING DEPARTMENT. California Polytechnic State University 1-49, 2010.

5. M. Wasim, Abdulbasit Shaikh, A.A Siddiqui. Communicator for Hearing-Impaired Persons using Pakistan Sign Language (PSL). The International Journal of Advanced Computer Science and Applications, Vol. 9, No. 5, 2018.

6. Kausar, S., Javed, M. Y., & Sohail, S. Recognition of gestures in Pakistani sign language using fuzzy classifier. In Proceedings of the 8th conference on Signal processing, computational geometry and artificial vision (pp. 101-105). World Scientific and Engineering Academy and Society (WSEAS), August, 2008.

7. Ali, S. A., "Detection of Urdu Sign Language using Harr Algorithms", International Journal of Inventive Engineering and Sciences (IJIES) ISSN: 2319–9598, Volume-1, Issue-6, May 2013.

8. Zhe Cao, Yaser Sheikh, et al, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", arXiv:1611.08050v2 [cs.CV], Apr 2017.

9. Avinash Navlani," Support Vector Machines with Scikit-learn", available at https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python, 12 July 2018.

10. Avinash Navlani," KNN Classification using Scikit-learn", availableat https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classificationscikitlearn, 2 Aug 2018.

11. Yun-lei Cai et al "A KNN Research Paper Classification Method Based on Shared Nearest Neighbor", Proceedings of NTCIR-8 Workshop Meeting, Tokyo, Japan, June 2010.

12. Ms. Sonali. B. Maind et al," Research Paper on Basic of Artificial Neural Network", International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 2 Issue: 1, 2014.

13. Mart́ın Abadi, Ashish Agarwal, et al," TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems", Preliminary White Paper, November 9, 2015.