Maryam Ijaz
01-134162-068
Syeda Miral Kazmi
01-134162-047

Hand gesture recognition for sign language

Bachelors of Science in Computer Science

Supervisor: Dr Kashif Naseer
Department of Computer Science
Bahria University, Islamabad
February, 2020

Bahria University
Islamabad Campus
(Department of Computer Science)

CERTIFICATE

We accept the work contained in the report titled"Hand Gesture Recognition for Sign Language" as a confirmation to the required standard for the partial fulfillment the degree of BS(CS)

Sabina

Supervisor
Name:_____
Date:_____

Internal Examiner
Name: Dr. Sabina Akhtar
Date: 23 July 2020

External Examiner
Name: Dr. Azhar Mahmood
Date: 23rd July 2020

Project Coordinator
Name:_____
Date:_____

Head of Department
Name:_____
Date:_____

Abstract

We have come to know a very genuine issue of sign language recognition, that problem being the issue of two way communication i.e. between normal person and deaf/dumb. Current sign language recognition applications are lack of basic characteristics which are very necessary for the interaction with environment. Our project is focused on providing a portable and customizable solution for understanding sign language through an android app. The report provides a summary of the basic concepts and steps involved in creating this android application that uses gestures recognition to understand American sign language words. The project uses different image processing tools to separate the hand from the rest and then uses pattern recognition techniques for gesture recognition. A complete summary of the results obtained from the various tests performed is also provided to demonstrate the validity of the application.

Acknowledgement

We have successfully completed the project by the grace of Allah, and we thank our parents who have always supported us in each part of our lives and our teachers for their constant support and guidance. A major gratitude to our supervisor Dr. Kashif Naseer Qureshi who supported us in every phase of our project especially documentation, it was not possible as he was always there to guide us.

Maryam Ijaz, Syeda Miral Kazmi
Bahria University Islamabad, Pakistan.

April 2020

Contents

List of Tables

1 Introduction

1.1 Introduction

In daily life, human beings communicate with each other and interact with computers using gestures. As a kind of gesture, sign language (SL) is the primary communication media for deaf people. In other words, sign language is a visual language that is used by deaf individuals as their tongue. Not like acoustically sent sound patterns, language uses visual communication and manual communication to fluidly convey the thoughts of someone. It is achieved by combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions. It is used by a person who cannot speak or by a person who cannot hear or both and by normal people to communicate with hearing disabled people. Sign language is very important for deaf and dumb people for their social, emotional and linguistic growth [**1**].

Sign languages can be divided into two categories: Static signs and Dynamic signs. In static signs only hand shape and orientation are analyzed for recognition of signs. Whereas in case of dynamic or continuous signs, along with shape and orientation, direction of movement and sequence of gestures is also considered to completely understand sign.

Sign language translation is one among the foremost difficult tasks within the field of Computer vision. The thought is to tackle this challenge in real time. Given a video feed, the idea is to sight the hand gesture in real time then translate the corresponding sign into the corresponding linguistic communication. In order to achieve this task, we would be using American Sign language dataset [**2**].

The project would consist of a user interface of an android application that would capture hand gestures in real time

1.2 Objective

The main objective of this project is to design an application that can supports the deaf and dump people to communicate with normal people. This project also aims to meet the following objectives:

- To give services to the deaf community.
- To bridge the communication gap between normal person and dumb person.
- To solve the problem on real time

1.3 Problem Statement

Even though the deaf and dumb people can communicate with each other without having any issues, but there is a major challenge for the deaf and dump people to communicate with the normal people. This is because not every single typical person can understand their gesture-based communication. Most of normal people has not been educated about the sign language. Therefore, they need a middle person to translate for both sides to make them understand each other. As communication is imperative, this issue makes an obstacle for the disabled people to compare with the normal. Therefore, a sign language translator must be developed to tackle those issues.

1.4 Problem Description

Problem Description Sign languages are languages that convey messages using manual communication like concurrent hand gestures, movement, and orientation of fingers, arm or body movements, and facial expressions. These languages are used by deaf and dumb community. There are many varieties of sign languages, however the universal language is employed everywhere in the world [3].

These sign languages cannot be understood by everyone; this is the basic rationale behind the idea of this project. By this project users will be able to translate the language into text. This would result in the elimination of the middle person that usually acts as a medium of translation. This application would provide outcomes by providing text output for a signal gesture input.

How to capture gestures in real time:

- Detection of location of hand using object detection model
- Zoom in and center that image.
- This would be then used as an input for sign language translation model.

1.5 Methodology

This application focuses mainly on the iterative method for development. The project throughout its various stages undergo review from the project advisor, and other relevant sources. All following developments and the direction of the project is based upon the output from these review processes. We will use Agile methodology because It uses an incremental approach which has ability to adapt to changes.

Since the application works by projecting an image of hand which is showing a sign in real space, a good part of the project involves image processing. Workflow of application

1. **For translation of hand gestures**
    - Application Start.
    - Search for hand gesture.
    - Image Processing.
    - Identify Target gesture.

- Image tracking.
- Track/lock co-ordinates.
- Verify data from gesture.

2. **For translation of text**

- Open writing area
- Input the text
- Process the text
- Translate the text
- Image tracking
- Verify Gesture with text Show the output in real time to user

1.6 Scope

In this project we are focusing on these modules

- Translation of the gestures.
- Translation of alphabets to gestures.

1. **Translation of the gestures**
   In this part we translate the gestures in simple language that a person can easily understand. We provide the gestures in real time.
2. **Translation of alphabets to gestures**.
   In this part we translate the text in the gesture of the Sign Language that a deaf/dumb person can easily understand.

2 Literature Review

This section reviews the research on the important elements in developing the sign language recognition application. The first research study focuses on gesture recognition method for detecting the movements of the hand. The second research study discusses the hardware that will be used in this project.[**3**]

2.1 Image Processing and Their Techniques

The application implements image processing in the following steps:

- Acquire the source Image
- Process the Image
- Show Output

2.2 TECHNIQUES

1. **PREPROCESSING**

   I **RGB to Grayscale** The color to grayscale method can be applied using different algorithms such as luminance, value and intensity among many others. There are two methods to convert it. Both has their own merits and demerits. The methods are:

   - Average method
   - Weighted method or luminosity method

   II **Binarization** Image binarization is the process in which they use already converted grayscale images to produce a binary image, which can further make it easier to detect markers by reducing the color range further from grayscale to a direct black and white. This is sometimes known as image thresholding.

III **Filtering** Filtering is a technique for modifying or enhancing an image. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement

IV **Noise removal and smoothing**

Image smoothing is an operation aiming to improve image quality by manipulating image parameter and is a way to disappear noise. There are some methods of image smoothing. One of them is Gaussian filtering

V **Region Filling** Region filling is a Morphological algorithm in image processing, which deals with filling the region in the image with some colors. The image region can be selected in two ways: • Interior region • Boundary region

2. **Hand Detection** First, colored image is read which is captured in image achievement step. After we get the image it starts calculating the dimensions of the image. There are number of color bands in the image which converted in one. If the image that you captured is not a grayscale, convert it to grayscale by only taking green channel. Now find the biggest blobs. This technique results in giving two biggest blobs, ignore the first biggest blob, which is the largest one. The second biggest blob will be the hand. This result in drawing box around the blobs and second biggest blob is separated from the image. The limitation of this technique is that color of clothes and other objects in scene might affect it.

3. **Hand Cropping:** The hand is cropped out, after the portion of hand is separated from the Image, for doing this process certain threshold is used. In binarizing of the image, a threshold value is used, which only gives out the portion of image with hand and then we can crop out the hand. This image of hand is then stored and passed to the next phase for further process.

4. **Feature Extraction**: In features extractions there are various algorithms used like Zernike moments and Fourier descriptors. In general, descriptors there are some set of numbers that are produced to describe a given shape.

2.3 Basic Operation

1. Input for the gesture recognition system must be liable and decided. **preprocessing**: Separation and tracking are important in order to extract useful information from raw gesture images. Thus, it is necessary to be able to recognize the region of foreground and split it from the background in each gesture image.

2. **Feature vector extraction**: In the third stage, the features (geometric and non-geometric) must be removed; these all features will be used at the time of testing operation.

3. **Efficient classification algorithm**: Classifier is used to identify the current presented testing gesture belongs to which trained class.

2.4 Comparison table

Existing System	Proposed System
The existing system can only detect the gesture if the background is black or white.	In our project hand gesture recognition works on every background with average light.
The person should wear full selves' shirt for detecting the right image.	In our project there is no such condition the user can wear whatever he wants.
There is only one module in their system which is gesture to text.	In our project there are two-way communication because we are going to add both the module text to gesture and gesture to text.
The system is web based with machine learning.	Our project is android based in which we use the concepts of image processing.

Figure 2.1: Comparison table

3 Requirement Specifications

3.1 Purpose

Sign languages are languages that convey messages using manual communication like concurrent hand gestures, movement, and orientation of fingers, arm or body movements, and facial expressions. These languages are used by deaf and dumb community. There are many varieties of sign languages, however the universal language is employed everywhere in the world. These sign languages cannot be understood by everyone; this is the basic rationale behind the idea of this project.

3.2 Product Scope

By this project users will be able to translate the language into text. This would result in the elimination of the middle person that usually acts as a medium of translation. This application would provide outcomes by providing text output for a signal gesture input. The project would consist of a user interface of an android application that would capture hand gestures in real time.

3.3 Overall Description

3.3.1 Product Perspective

- The application must work on real time bases.
- The output of the sign gesture to normal language must be accurate, same for the normal language to the sign language
- The application will not hang during using the application.
- The design interface of the application must be innovative.

3.3.2 Application Functions

The application would provide two main function:
1. Translation of the gestures.
2. Translation of alphabets to gestures.
The first function is for deaf and dump people to convey their message to the normal person. In which they translate the gestures in simple language that a normal person can easily understand. We provide the gestures in real time. The second function is for normal person so that they can convey their message to the deaf and dump people. In which they translate the text in the gesture of the Sign Language that a deaf/dumb person can easily understand.

3.4 External Interface Requirements

3.4.1 User Interface

The user interface of the android application will be simple and easy to understand. The navigation of the application will be intuitive so that users will be able to immediately identify how to get around. The user interface will be optimized in way so that it maintains speed and does not lag if readings of the patient are constantly refreshing.

3.4.2 Hardware Interface

There is no hardware interface used for our application.

3.4.3 Software Interface

Following tools and technologies will be used for performing task:
Language: Open CV, VS code, C Scripting, Python
Tools: Android Studio, Unity
Unity:
A popular tool and Integrated Development Environment for building inter-

active environments such as games, simulations, animations and much more.
Android Studio:
Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJent IDEA software and designed specifically for Android development.

3.4.4 Communication Interface

The communication interface is the cellphone through which both the people can interact.

3.5 System Requirements

System Requirement
System specification are divided into following two categories:
a) Functional requirements
b) Non-functional requirements

3.5.1 Functional Requirements

- Scan Hand gesture: Application allows user to capture a real time image from the built-in camera. After capturing, the captured image will be used to project the output in the simple English language.
- Read the English language into sign language Application allows user to write the alphabets in a real time. After translating, the text will be used to project the output in the hand gesture images.

3.5.2 Non Functional Requirements

- **Reliability**: The application should reliably detect abandoned objects from given image.

Table 3.1: Image capture

Use Case	Image capturing
Use Case id	1
Actors	User.
Goal	To capture the image.
Pre-Condition	View Button for camera opening
Post-Condition	Marker Placement
Basic Flow	1. Open the application 2. Initiated when user click the camera button. 3. User can capture the real time image. 4. The image will be projected by the application in device's temporary memory.

- **Security**: The application does not have any specific security requirements.
- **Usability**: Application will be user friendly, easy to use, and understandable to every common user.
- **Performance**: The application shall perform well and very fast and it must be accurate.
- **Maintainability**: Developers will be responsible for the maintenance of the system.
- **Availability**: The system will be available once the application has been installed on a phone.
- **Re-usability**: Different modules of the system should be developed in such a way to be reusable in other similar systems.

3.6 Use case

A. Use Case for translate the gestures in simple language

B. Use Case for translate the alphabets in sign language.

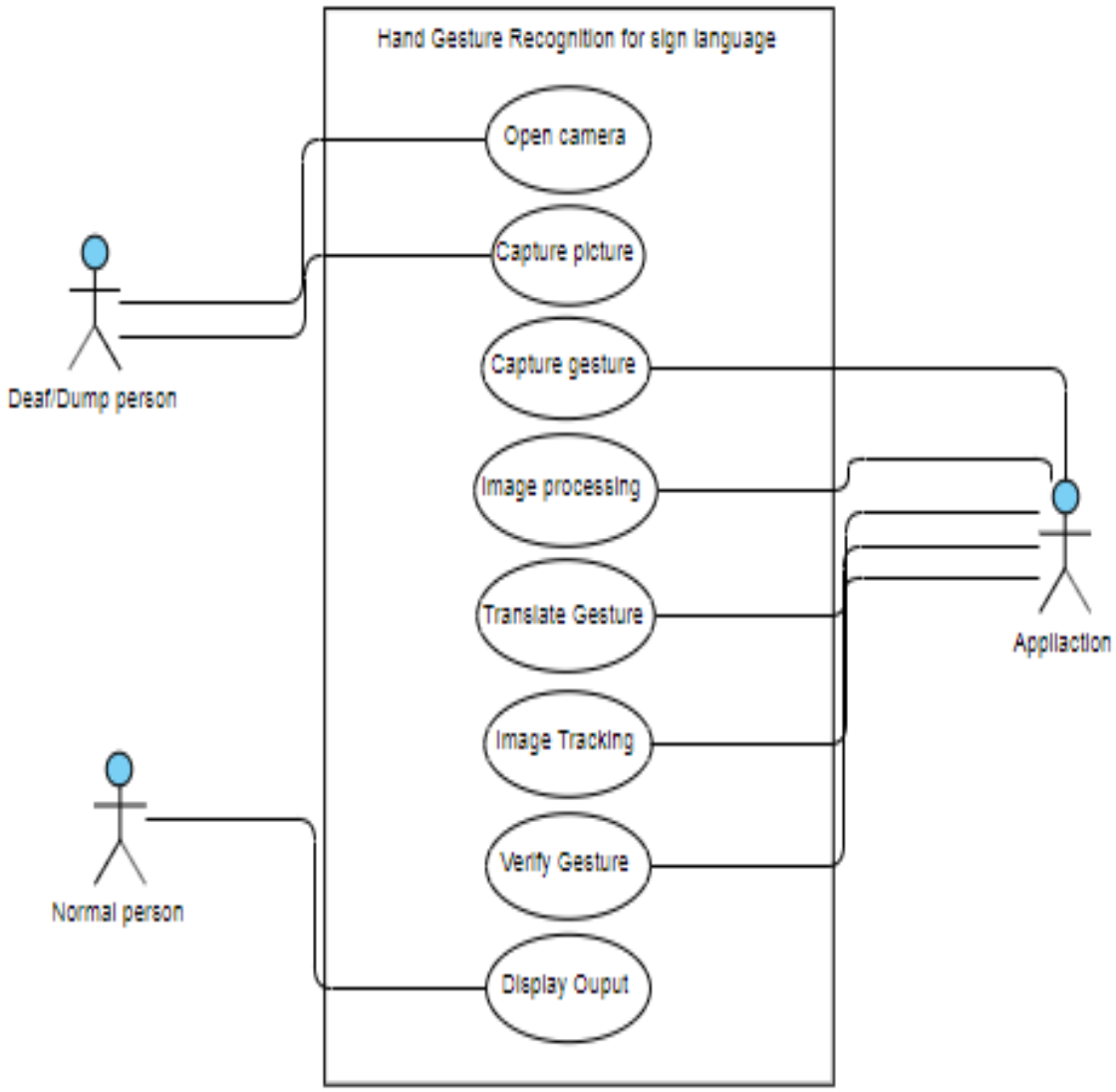


Figure 3.1: Gesture to Simple Language

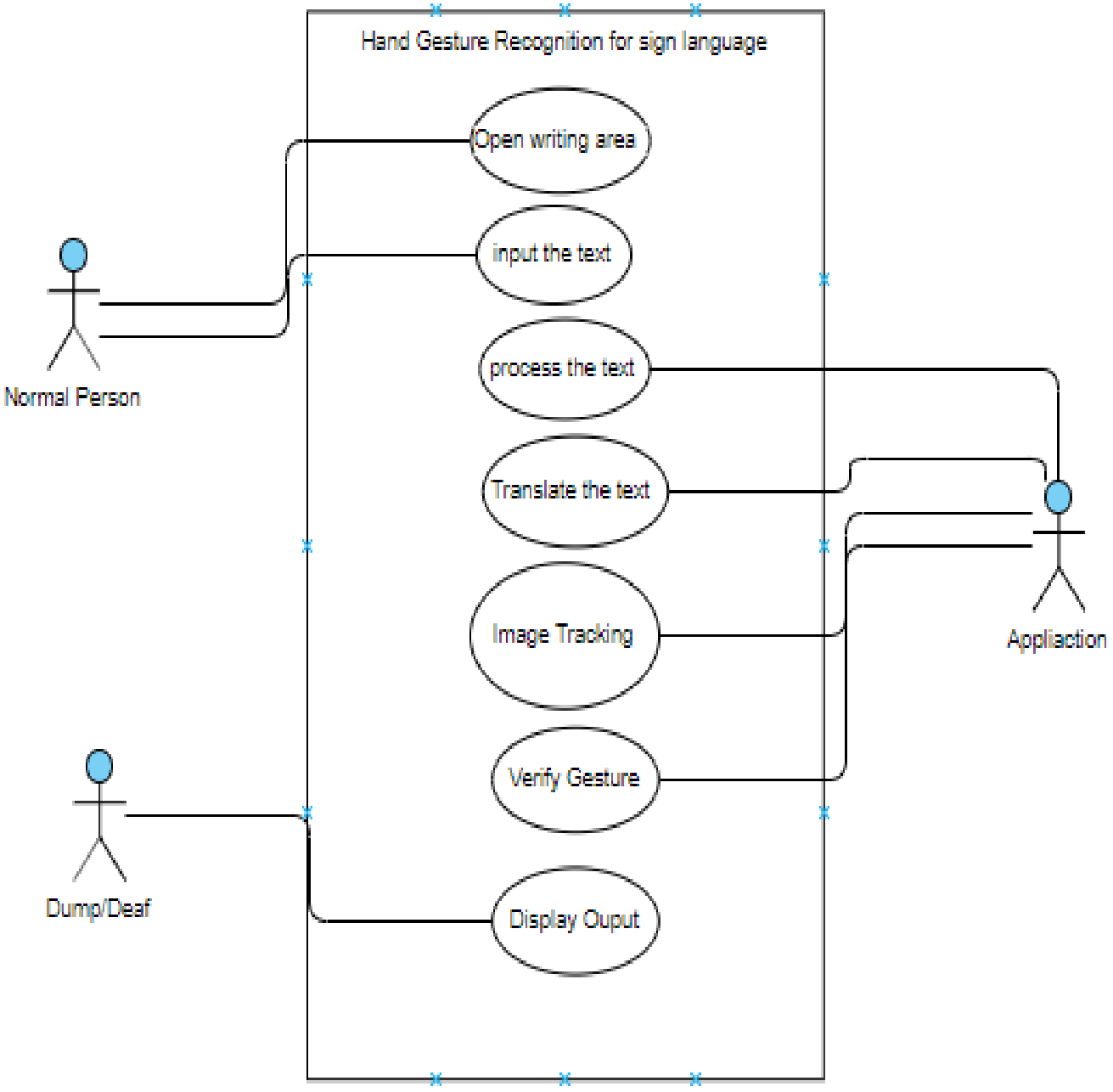


Figure 3.2: Simple language to gesture

Usecase Description

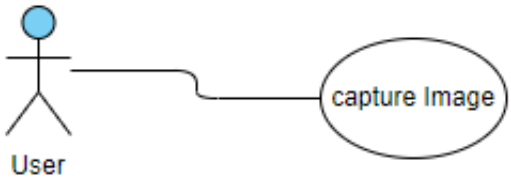


Figure 3.3: capture Image

Table 3.2: Image capture

Use Case	Image capturing
Use Case id	1
Actors	User.
Goal	To capture the image.
Pre-Condition	View Button for camera opening
Post-Condition	Marker Placement
Basic Flow	1. Open the application 2. Initiated when user click the camera button. 3. User can capture the real time image. 4. The image will be projected by the application in device's temporary memory.

Figure 3.4: Image processing

Table 3.3: Image processing

Use Case	Image processing
Use Case id	2
Actors	User.
Goal	Identify Target Gesture
Pre-Condition	Capture image
Post-Condition	Translate gesture
Basic Flow	1. 1. Acquire the source Image from the user 2. Process the Image 3. Show Output

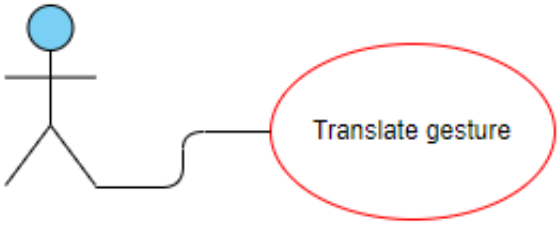


Figure 3.5: Translating gesture

Table 3.4: Translating gesture

Use Case	Translate gesture
Use Case id	3
Actors	application.
Goal	To Translate sign language to normal language.
Pre-Condition	Image tracking
Post-Condition	Verify gesture
Basic Flow	1. Application Start 2. Search for hand gesture 3. Image Processing 4. Identify Target gesture 5. Image tracking 6. Track/lock co-ordinates 7. Verify data from gesture

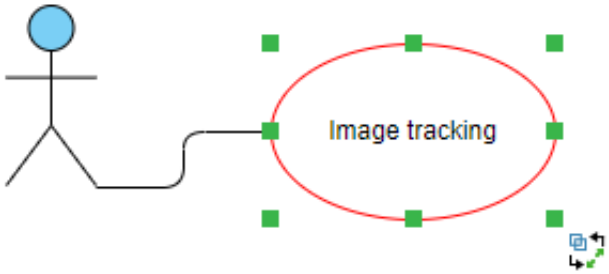


Figure 3.6: Translating gesture

Table 3.5: Image tracking

Use Case	Image tracking
Use Case id	3
Actors	User.
Goal	To lock coordinates .
Pre-Condition	Translate text .
Post-Condition	Marker Placement.
Basic Flow	1. Taking an initial set of object detection. 2. Creating a unique ID for each of the initial detections. 3. And then tracking each of the objects as they move around frames in a video, maintaining the assignment of unique IDs.

Figure 3.7: Image tracking

Table 3.6: Verifying gesture

Use Case	Verifying Gesture
Use Case id	4
Actors	application.
Goal	To fetch information.
Pre-Condition	Image tracking
Post-Condition	Display output
Basic Flow	1. Verify data from gesture 2. Compare the gesture to trained model. 3. If matches, then displays output.

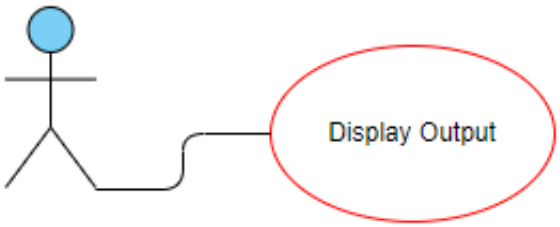


Figure 3.8: Verifying gesture

Table 3.7: Output

Use Case	Output
Use Case id	5
Actors	User.
Goal	Output in real time to the user.
Pre-Condition	Verify gesture
Post-Condition	Nil
Basic Flow	1.Shows the output in real time to user.

Figure 3.9: Verifying gesture

Table 3.8: Entering data

Use Case	Entering data
Use Case id	1
Actors	User.
Goal	Enter text .
Pre-Condition	View Button for camera opening.
Post-Condition	Translate input Text
Basic Flow	1. Open the application. 2. Write the text you want to translate.

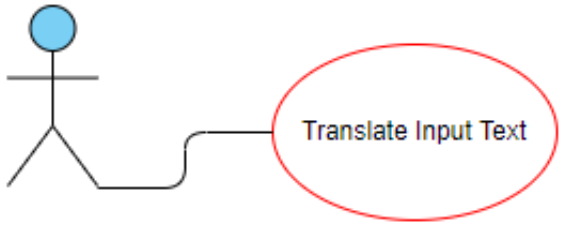


Figure 3.10: Verifying gesture

Table 3.9: Translate input Text

Use Case	Translate input Text
Use Case id	2
Actors	User.
Goal	To translate the text .
Pre-Condition	Enter input text.
Post-Condition	Image tracking.
Basic Flow	1. Input the text 2. Process the text. 3. Translate the text.

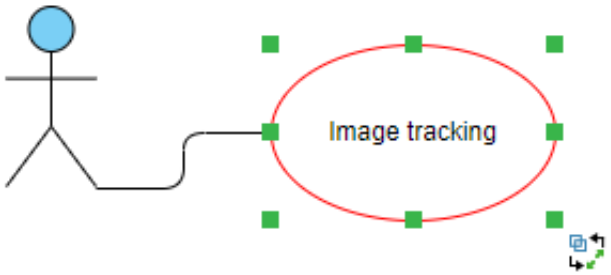


Figure 3.11: Image tracking

Table 3.10: Image tracking

Use Case	Image tracking
Use Case id	3
Actors	User.
Goal	To lock coordinates .
Pre-Condition	Translate text .
Post-Condition	Marker Placement.
Basic Flow	1. Taking an initial set of object detection. 2. Creating a unique ID for each of the initial detections. 3. And then tracking each of the objects as they move around frames in a video, maintaining the assignment of unique IDs.

Figure 3.12: Verifying gesture

Table 3.11: Verify gesture

Use Case	Verify gesture
Use Case id	4
Actors	Application.
Goal	To fetch information.
Pre-Condition	Image tracking .
Post-Condition	Display output.
Basic Flow	1. Verify Gesture with text. 2. Compare the gesture to trained model. 3. If matches, then displays output.

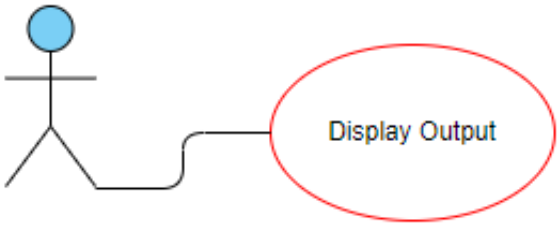


Figure 3.13: Verifying gesture

Table 3.12: Output

Use Case	Output
Use Case id	5
Actors	User.
Goal	Output in real time to the user.
Pre-Condition	View Button for camera opening.
Post-Condition	Nil
Basic Flow	1.Shows the output in real time to user.

4 System Design

The following chapter will discuss certain technical aspects of the project, specifically related to the project's design both visual and technical. explain the architecture, components, modules, interfaces, and data for a system to satisfy requirements.

4.1 Low Level Design

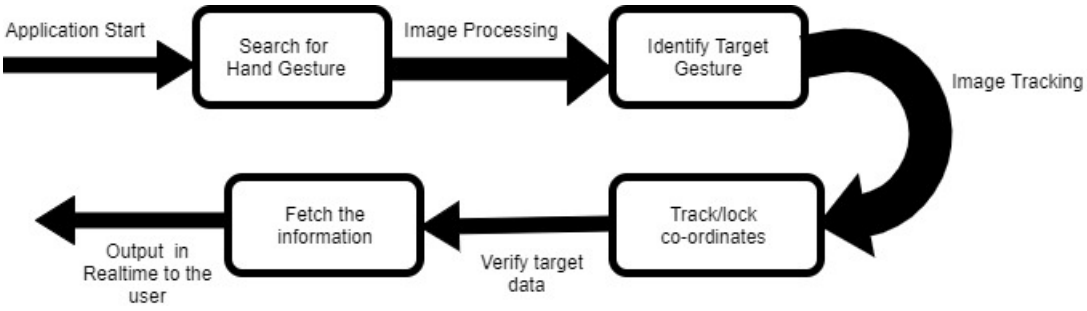


Figure 4.1: Low Level Design for translating of gesture

Figure 4.2: Low Level Design for translating normal language

4.2 System flow diagram

4.2.1 For gesture to text

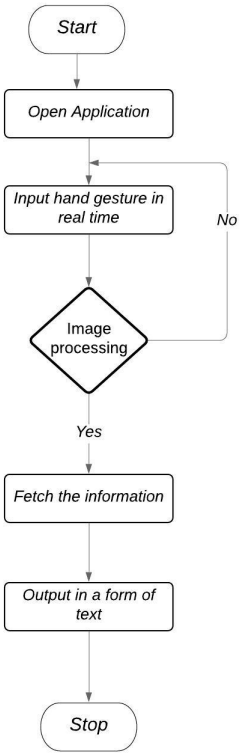


Figure 4.3: System Flow diagram for gesture to text system

4.2.2 system flow diagram for gesture to normal language

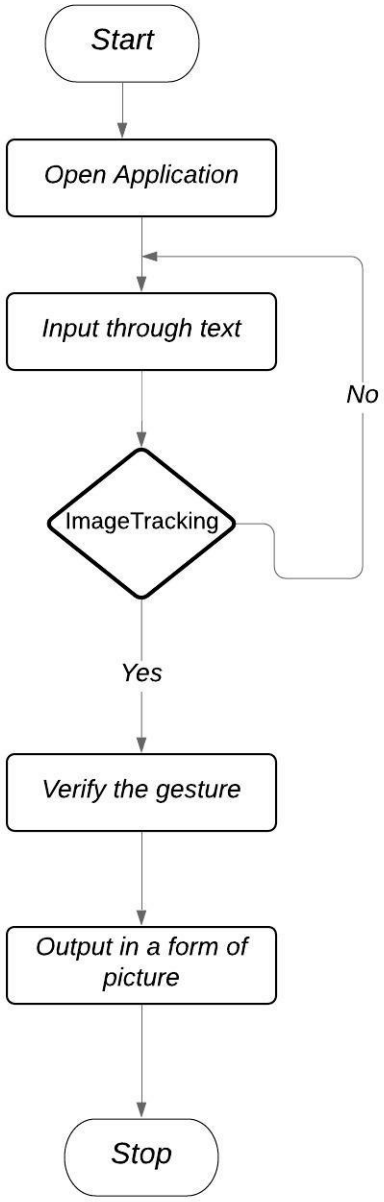


Figure 4.4: system flow diagram for text to gesture

4.3 High level Diagram

The main components for transiting the gesture to text are:

- User Interface
- Image Processing
- Output

The image processing components further process according to the flow diagram given below:

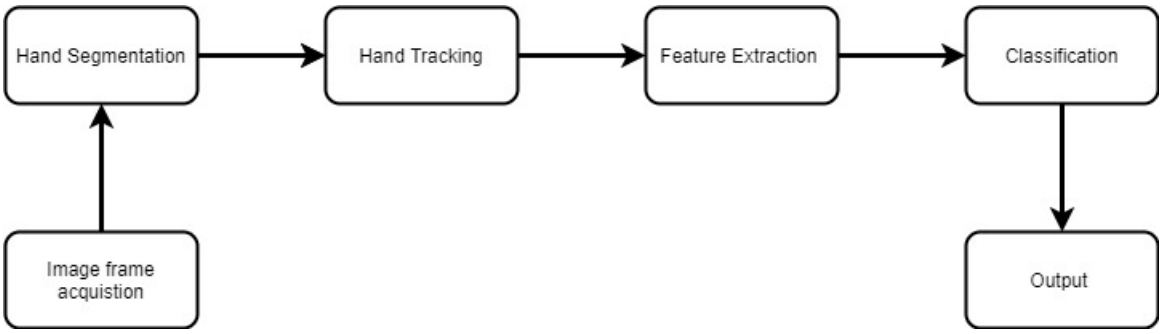


Figure 4.5: High Level Design for gesture to text

The main components for transiting the text to gesture are:

- User Interface
- Translating text
- Output

The Translating text components further process according to the flow diagram given below:

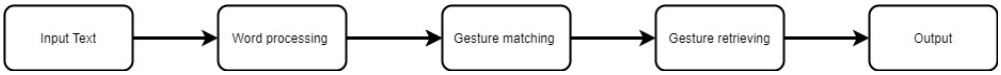


Figure 4.6: High Level Design for text to gesture

4.4 System Interaction Diagram

4.4.1 For gesture to text

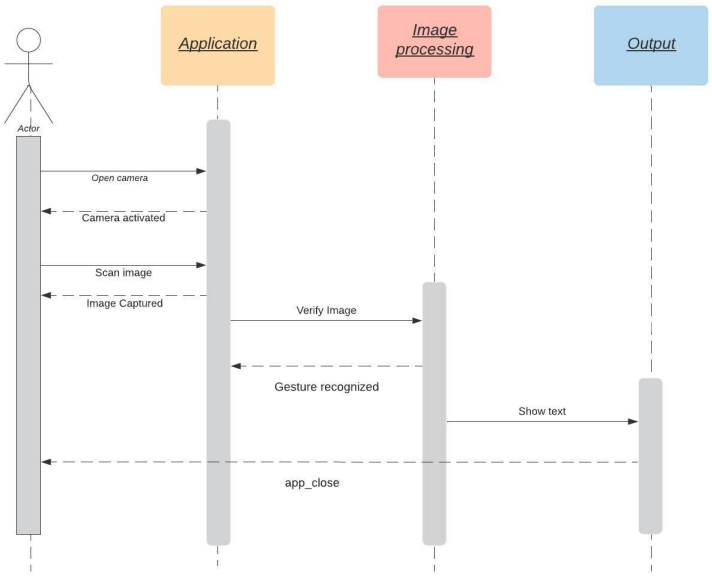


Figure 4.7: Sequence diagram for gesture to text system

4.4.2 For text to gesture

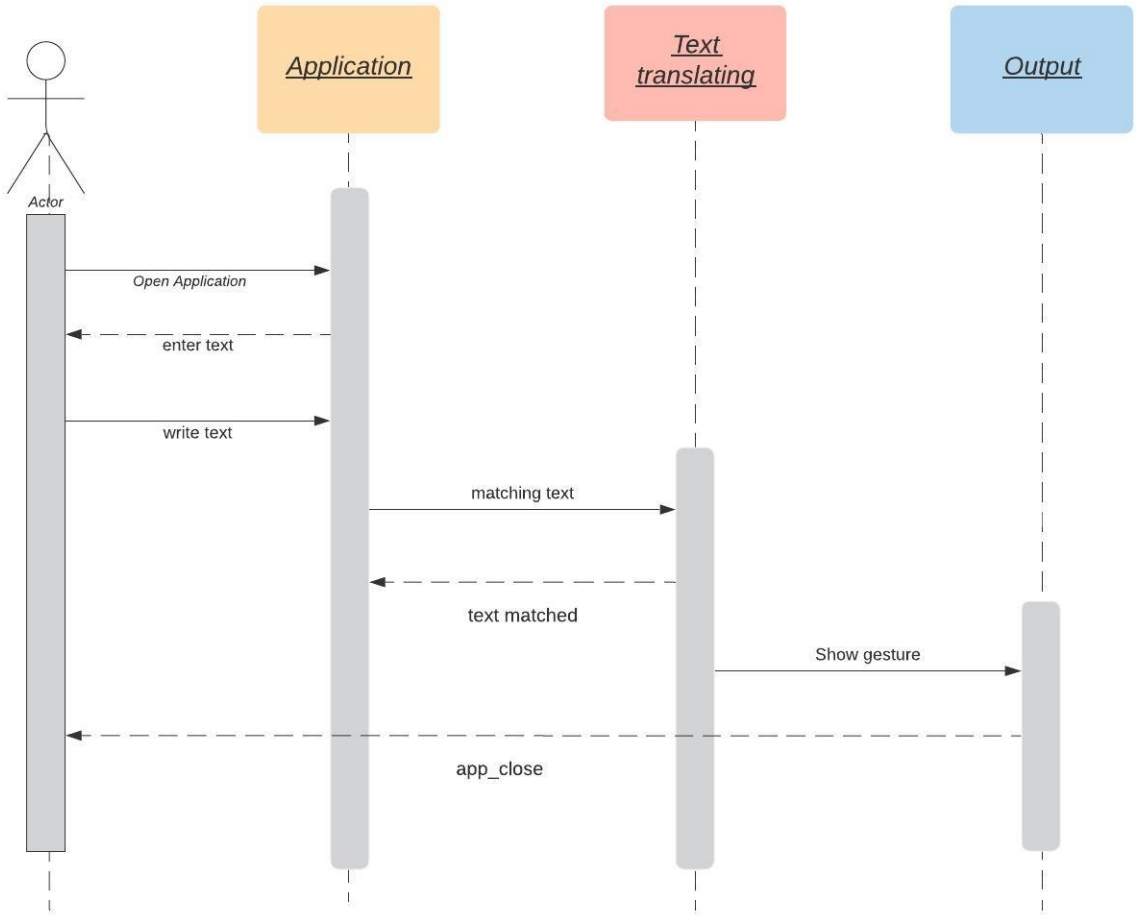


Figure 4.8: Sequence diagram for text to gesture

4.5 Design Constraints

The project and its deliverable are subject to the following constraints:

- The end user device must have an Android platform of version 4.4 or above to run the application.

- The end user device must have a camera of at least 5 mega pixel which the base requirement for any image processing operations. The quality of image may affect the performance of our application.

4.6 Assumptions and Dependencies

4.6.1 Assumption

- Detection of location of hand using object detection model, Zoom in and center that image. This would be then used as an input for sign language translation model.
- The Hand Gesture Recognition for sign language application will detect the correct gesture in the real time.
- The application will not hang when the user is using, because of ant technical fault in the application.

4.6.2 Dependencies

- The person who is making this application must know about the sign language it will help in the development of this application in the phase of translating sign language into normal language.
- The hardware needs for the user is that one must have a phone with camera to operate this application.

5 System Implementation

This chapter will include all the tools and techniques we used in the implementation of our project. Implementation is the process of moving an idea from concept to reality. The System implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment.[**4**]

5.1 Development Environment (IDE)

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. Android studio is used in this project to develop the android application for smart mobile phones. This application could be installed by the user and connected with the Wi-Fi.

5.2 Architecture and Components Integration

In this Section the integration between components of system is discussed

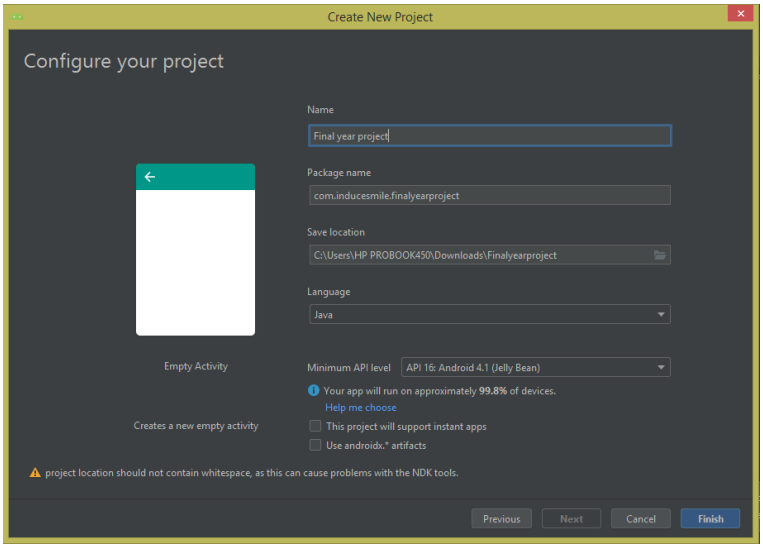


Figure 5.1: Creating new project in android

The figure explains creating of new project in android studio. The details required for the creation of project it involves the name of project, Language, and location to store

5.3 System Implementation

This section is about the integration of system in which Hand gesture recognition for sign language app is designed and developed. We are using android studio as a platform for our system. We have developed an android application that would capture hand gestures in real time. We also used OpenCV library which is used to train and train models for image processing. Furthermore, the chapter also describes the tools and techniques which are used in the implementation of Hand gesture recognition for sign language app. The steps that are used to follow up for setting the development environment are listed below:

- Download and Install android Studio (3.5 version)
- After installation of android studio, create a new project.

- After that start using it.

5.3.1 Interface

The android studio main window is made up of several logical areas identified as shown in the figure below.

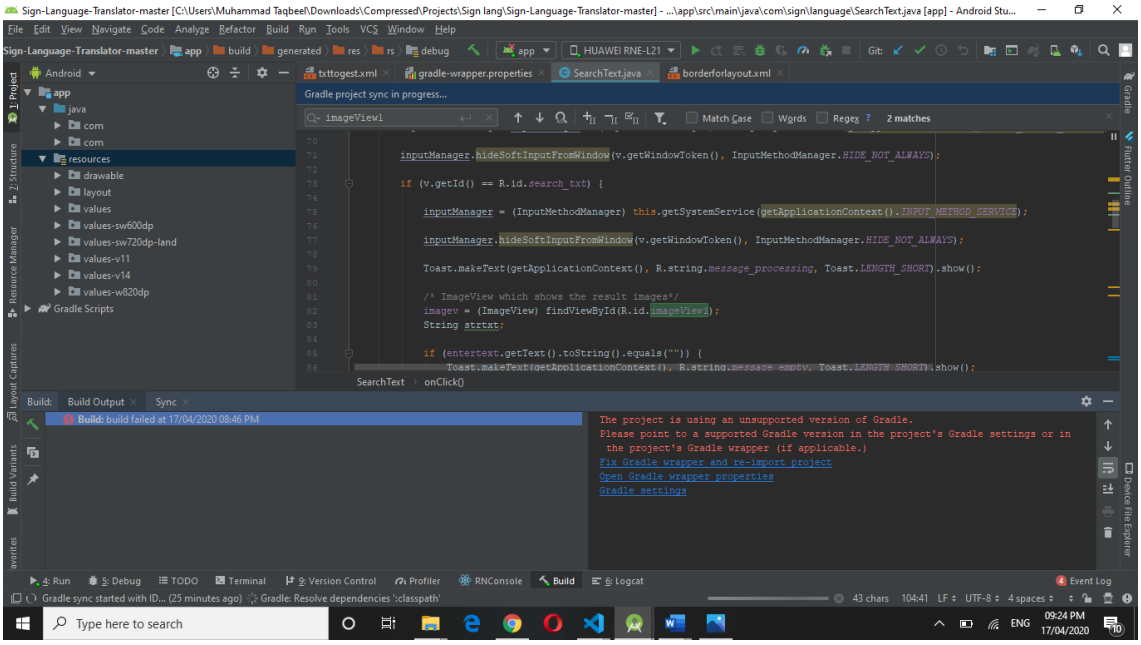


Figure 5.2: Interface of android studio

5.3.2 Application GUI

This illustrate the User Interface design of our project.The user interface of our application is displayed below.

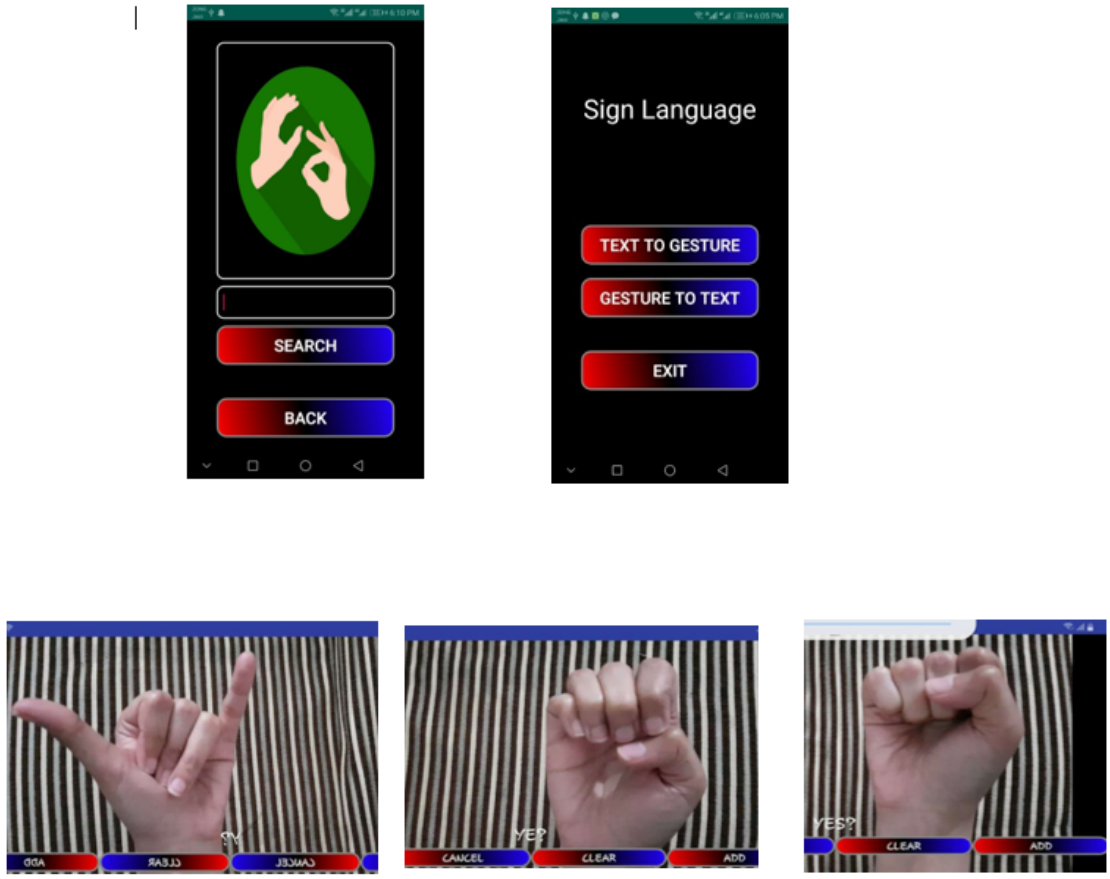


Figure 5.3: User Interface

5.4 Application Structure

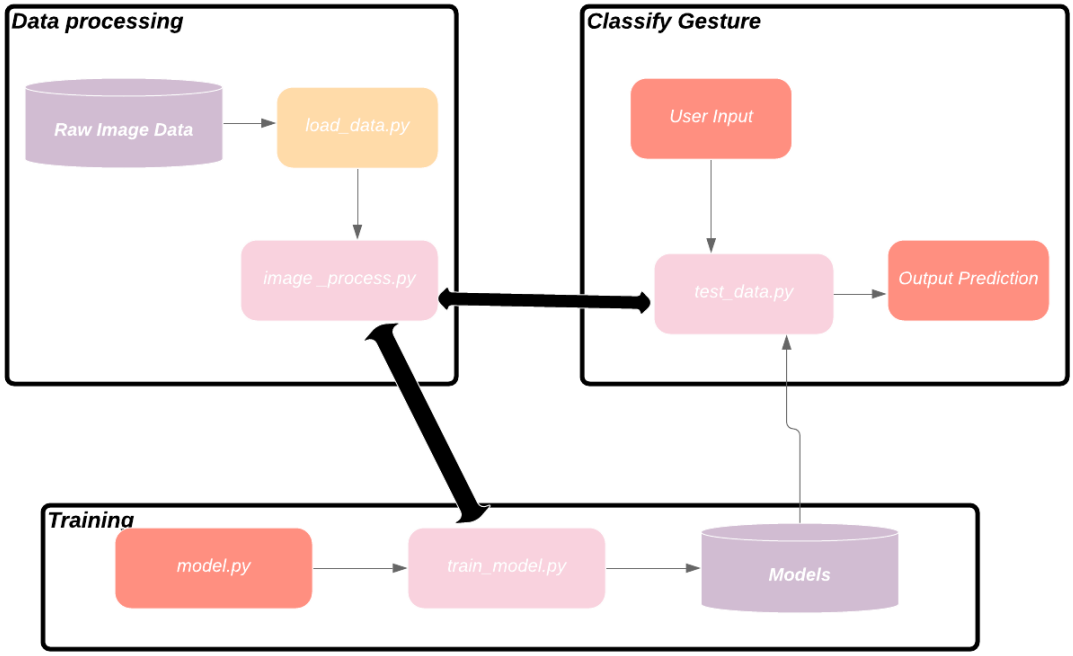


Figure 5.4: Application Structure

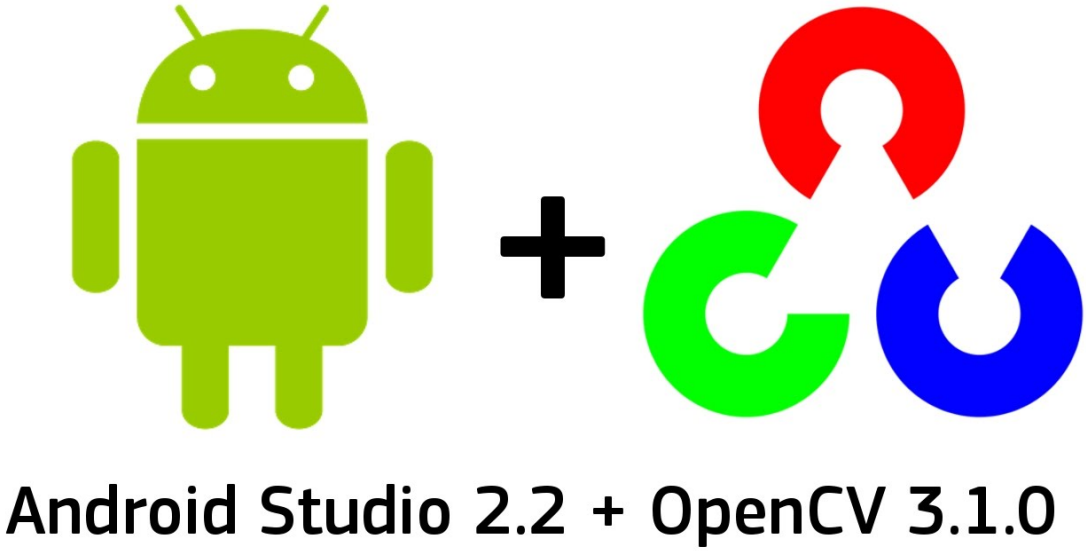
5.4.1 Tools and Techniques

The application is developed in Java and python using Anroid Studio , in the integrated development environment (IDE). For the processing of images from the webcam or mobile phone and detecting hand gestures we are using OpenCV for Anroid Studio. OpenCV for Anroid Studio is used for the image processing part i.e. taking input from webcam and conversion of camera image to mat for further processing. These are the tools and technologies used for our system.

- Computer (Hardware).
- Android Phone.
- Android Studio version 3.5

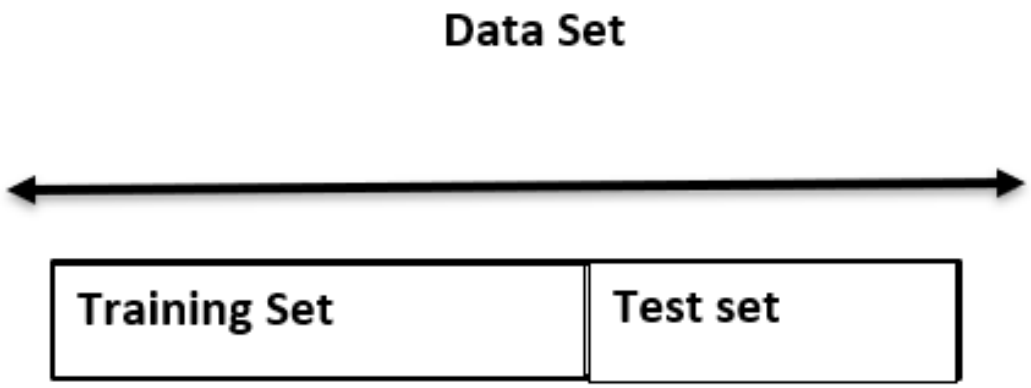
- It has as strong editor tool for developing creative UI and emulators for different.
- Versions to test and simulate sensors without having actual Android Devices.
- We are using Java and python language for our application.
- Excel for sign language dataset creation.

1. **OpenCv**



For our project we used a version i.e. 3.4.3 since that was the only free version available online. We can do image processing in real-time by using the camera with a help of open cv.

2. **Trained dataset**
Trained dataset is the model of gesture which is set by training gestures of same type in many poses.



3. **Support Vector Machine**
   SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. It is used in a variety of applications such as face detection, intrusion detection, classification of emails, news articles and web pages. SVM algorithm is implemented in practice using a kernel. A kernel transforms an input data space into the required form. SVM uses a technique called the kernel trick. Kernel trick helps you to build a more accurate classifier. SVM Classifiers offer good accuracy and perform faster prediction compared to Naïve Bayes algorithm. They also use less memory because they use a subset of training points in the decision phase. SVM works well with a clear margin of separation and with high dimensional space. SVM is not suitable for large datasets because of its high training time and it also takes more time in training compared to Naïve Bayes but we used the SVM classifier because we used a small dataset and there are no overlapping classes going to be used. SVM is effective in cases where number of dimensions is greater than the number of samples. SVM is relatively memory efficient.

5.4.2 Preprocessing and initialization

Preprocessing and segmentation

1. Start
2. Input BGR image.
3. Convert frame into RGBa
4. Down sampling the real image.
5. Convert image in to grayscale.
6. Convert an intensity image to a binary image.
7. Convert the grayscale threshold-based image into binary image.
8. Crop and extract the detected binary hand from the previous step.
9. Normalize size of the image vertically.
10. Identify the edge boundary of the extracted binary Palm using canny approximation to the derivative.
11. Display the boundary of the palm

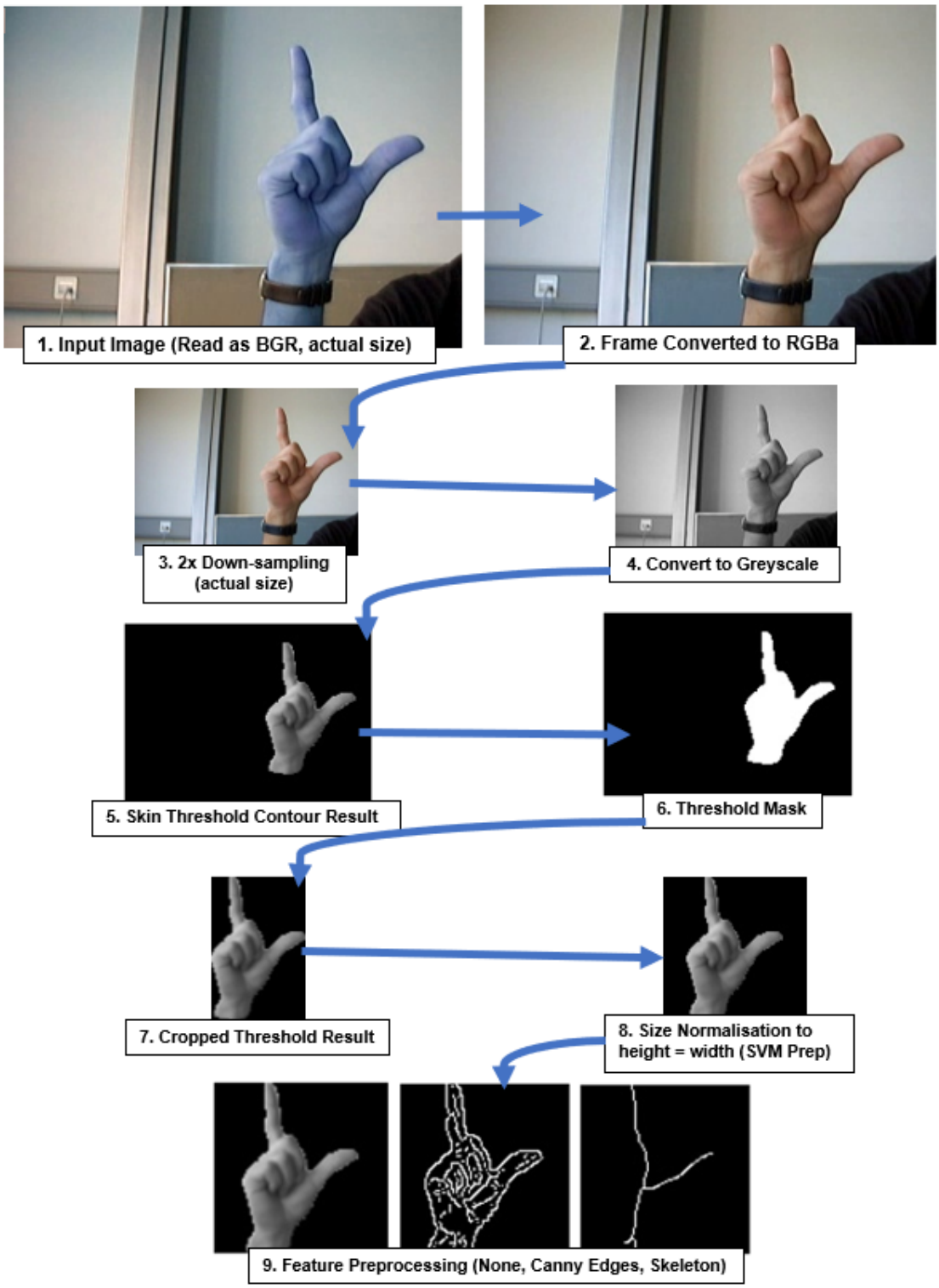


Figure 5.5: Processing and Initialization

5.5 Libraries

- OpenCv
- http://www.lifeprint.com/asl101/pages-signs (for text detection)

5.6 Processing Logic

1. For translation of hand gestures
    - Application Start
    - Search for hand gesture
    - Image Processing
    - Identify Target gesture
    - Image tracking
    - Track/lock co-ordinates
    - Verify data from gesture

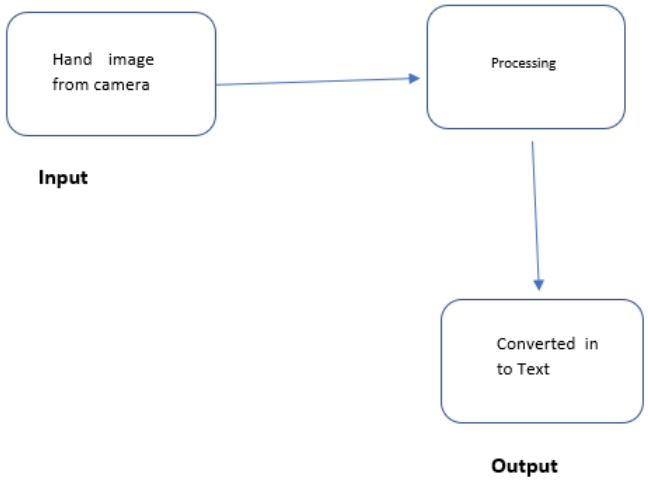


Figure 5.6: Processing logic of text to gesture

2. For translation of text

   - Open writing area
   - Input the text
   - Process the text
   - Translate the text
   - Image tracking
   - Verify Gesture with text

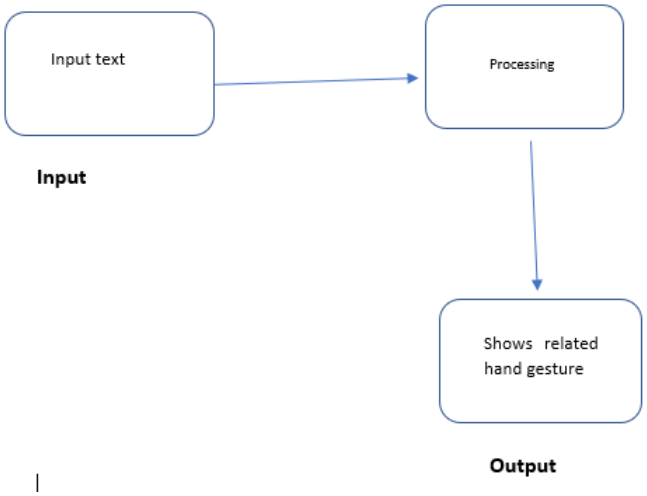


Figure 5.7: Processing logic of text to gesture

5.7 Trained data accuracy

We have a small dataset to work with; 100 samples for each of 26 classes, after processing the images into numpy arrays, we found our personal computers could load a maximum of 50-100 samples.We found the following relation between samples and model accuracy:

Sample size	Validation Accuracy
25	28.5 %
50	35.8%
75	48.2%
100	62.5%

Figure 5.8: Effect of Increasing Dataset Size on Validation Accuracy

Training our initial models on less data led to the models quickly overfitting as shown in Figure 5.7. This is likely due to the small amount of samples to train on leading to bad generalization and learning of the sample space. Increasing the size of our dataset to 100 samples led to better model results, with peak validation accuracy of 62.5

- The time complexity for training time = 20.73m.
- The time complexity for testing time = 2.53sec.

6 Testing and Evaluation

This chapter includes all the checking methodologies and test cases we have used to test our project. The most objective behind testing is to investigate whether or not the developed system meets the specified quality guidelines or not. System testing is significant for any system because it checks the capabilities of software and makes sure it is reliable.

6.1 Graphical user interface testing

Graphical User interface testing is a testing technique used to identify the presence of defects in an exceedingly software underneath test .This sort of test verifies that the application we tend to test will behaves as expected when a user performs a particular activity or enters a specific input in its activities and provide the proper program output .In our project the buttons in the android device were tested for actual response area size and by measuring the particular placement of the buttons on the Android device render.[**5**]

6.2 Usability testing

Usability testing is a testing technique which is utilized to check how simple and easy it is to use this system. Usability testing is the method of how a product can be used by users to reach specified goals. Usability testing is aimed to uncover how much the application is easy to use, understandable, is it able to satisfy the users' needs effectively. This testing method is performed by the target users, who are going to be a part of the system. We make sure that our application camera runs in lower frame rates due to load on device to capture and detect and giving output at the same time. Almost 6 frames average

6.3 Compatibility testing

Compatibility testing is that the methodology of evaluating compatibility of the system to examine whether the product is ready for running on various platforms like web-based browsers, operating systems, software and hardware. Various software is available in a number of different versions. In this way, it's common for applications to confront issues performance issues while operating more up to date or older versions of the software. With the help of compatibility testing we can evaluate the performance of a given application on various versions of the software. our application is built for android 6 and above and works for all android devices above android 6. This application cannot be used on web.

6.4 Performance Testing

Performance Testing is the testing to check the performance of the software we executed the application through different circumstances. We change the background while giving hang gesture as an input in real time and so concluded that the application will give the correct output.

6.5 Installation testing

Installation testing is a process of running the application on different platforms. During this testing we installed at the user's end and checked that user is not experiencing any difficulty while utilizing the application and all functionalities are completed. For our application you can installed it on different android devices with different OS it will worked as same.

6.6 Test Cases

Table 6.1: Working of screen

Test case ID	01
Test case title	Working of screen
Description:	Check all the screens are working.
TestSteps	1. Run the application on your phone. 2. Check every screen turn by turn.
Expected output:	All the screens are working
Status	Sucessfull

Table 6.2: Functionality of camera

Test case ID	02
Test case title	Functionality of camera
Description:	To check whether the camera is functional or not.
TestSteps	1. Open the application. 2. Choose to gesture to text option. 3. Show the hand-gesture Infront of the camera.
Expected output:	Camera is functional
Status	Sucessfull

Table 6.3: Gesture detection

Test case ID	03
Test case title	Gesture detection
Description:	Checking gestures which are trained in the application to show result.
TestSteps	1. Choose the gesture to text option. 2. Scan Realtime view of hand gesture of a person.
Expected output:	Gesture recognized
Status	Sucessfull

Table 6.4: Test to gesture

Test case ID	04
Test case title	Test to gesture
Description:	Checking If text input is scanning words for gesture.
TestSteps	1. Open the application. 2. Choose to text to gesture option. 3. Input text in textbox.
Expected output:	input text is scanning words for gesture is working
Status	Sucessfull

Table 6.5: Gesture recognition irrespective of background

Test case ID	05
Test case title	Gesture recognition irrespective of background
Description:	Check that the application recognized the real time hand gesture by changing different background.
TestSteps	1. Scan Realtime view of hand gesture of a person. 2. Change the background every time.
Expected output:	Gesture Recognized.
Status	Sucessfull

Table 6.6: Display Menu Screen

Test case ID	06
Test case title	Display Menu Screen
Description:	To observe that the display menu of application is user friendly.
TestSteps	1. User must install application on his android phone. 2. User can do all other functions that were described in the requirements specification
Expected output:	User is comfortable with the interface of the application.
Status	Sucessfull

7 Conclusion

An application is developed for deaf and mute people through which they can communicate with a normal person who does not understand their gesture. This application can translate words and can translate gestures. We have developed an application which can detect hand gesture in Realtime. and display the recognized gesture into text on screen. The device's back camera is used to capture real time alphabetical gesture through OpenCV java camera enabled which gets the gesture in low frame rate by detecting a single hand gesture and testing that captured gesture with the classified datasets of gesture which is consist of 80:20 train and test dataset. Classification of gesture is processed fast therefore the framerate is low so it can detect and test the gesture smoothly. The Realtime gesture recognition works on every background with average light. In very high or very low light it will be difficult to capture the gesture. The User can enter the letter, word or sentence in text area. The application will first try to recognize the word through the given website linked and will display the gesture, if it does not get the word then it will spell gesture of that word.

7.1 Future Enhancements

In the future we can enhance the feature of gesture to text. Right now, we are only converting gestures into alphabetic text but in future we can convert the words and sentences too. The other feature can also be enhanced by just not type the text to convert into gesture, but the user can speak the word, sentences so that the application show the related gesture as an output. usepackage[english]babel

References

[1] K. Hogan. . *K. Hogan, R. Stubbs," Can't get Through 8 Barriers to Communication",.* (2003) [*Pelican Publishing Company, Grenta, LA,.*].

[2] K. Assaleh *K. Assaleh and M. Al-Rousan, "Recognition of Arabic sign language alphabet using polynomial classifiers,".* (2005) [*EURASIP Journal on Applied Signal Processing*].

[3] M. A. Mohandes *M. A. Mohandes, "Recognition of two-handed Arabic signs using the cyber glove,"* (2013) [*Arabian Journal for Science and Engineering, vol. 38, no. 3, pp.669–677.*]

[4] R. Alzohairi *R. Alzohairi, R. Alghonaim, W. Alshehri, S. Aloqeely, M. Alzaidan, O.Bchir, "Image based arabic sign language recognition"*(2018) [*International Journal of Advanced Computer Science and Applications, Vol. 9, No. 3, 2018*]

[5] N. M. Kakoty *N. M. Kakoty and M. D. Sharma, "Recognition of Sign Language Alphabets and Numbers based on Hand Kinematics using A Data Glove,"* (2018) [*Procedia Comput. Sci., vol. 133, pp. 55–62, 2018*]