SIGN LANGUAGE RECOGNITION SYSTEM USING IMAGE PROCESSING

Project Report submitted to

Chhattisgarh Swami Vivekanand Technical University Bhilai (India)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

Electrical & Electronics Engineering

by

ANKIT KUMAR

MOHIT KUMAR GUPTA

ANISHA PAUL

YANENDRA KUMAR

Under the Guidance of

MISS. POORVA SHARMA

Department of Electrical & Electronics Engineering
Government Engineering College Raipur Chhattisgarh India-492015

Session: 2018-2022

DECLARATION BY THE STUDENTS

We, the undersigned, solemnly declare that the project report titled **"SIGN LANGUAGE RECOGNITION USING IMAGE PROCESSING"** is based on our own work carried out during the course of our study under the supervision of **MISS. POORVA SHARMA.**

We assert that the statements made and conclusions drawn are an outcome of our work. We further certify that

i. The work contained in the report is original and has been done by us under the general supervision of our supervisor(s).

ii. The work has not been submitted to any other Institute for any other degree/diploma/certificate in this University or the any other University of India or abroad.

iii. We have followed the guidelines provided by the University in writing the report.

iv. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references.

S.No.	Name of Student	Roll Number	Enrollment Number	Signature
1.	**ANISHA PAUL**	**301602518041**	**BF3230**	
2.	**ANKIT KUMAR**	**301602518011**	**BF3200**	
3.	**MOHIT KUMAR GUPTA**	**301602518034**	**BF3223**	
4.	**YANENDRA KUMAR**	**301602518049**	**BF3238**	

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the work incorporated in the project report entitled **"SIGN LANGUAGE RECOGNITION USING IMAGE PROCESSING"** is a record of work carried out by: -

S.No.	Name of Student	Roll Number	Enrollment Number	Signature
1.	**ANISHA PAUL**	**301602518041**	**BF3230**	
2.	**ANKIT KUMAR**	**301602518011**	**BF3200**	
3.	**MOHIT KUMAR GUPTA**	**301602518034**	**BF3223**	
4.	**YANENDRA KUMAR**	**301602518049**	**BF3238**	

under my/our guidance and supervision for the award of Degree of Bachelor of Engineering in the faculty of Department of Electrical& Electronics Engineering of Chhattisgarh Swami Vivekanand Technical University, Bhilai, Chhattisgarh, India.

To the best of my/our knowledge and belief the project report

i) Embodies the work of the candidates themselves,
ii) Has duly been completed,
iii) Fulfils the requirement BE degree of the University and
iv) Is up to the desired standard both in respect of contents and language for being referred to the examiners.

……. …….

(Signature of the Head of The Department) (Signature of the Supervisor)

Dr. R.S.PARIHAR **Ms. POORVA SHARMA**

HOD Assistant Professor

Forwarded to Chhattisgarh Swami Vivekanand Technical University, Bhilai

……..……..…….

(Seal of the Department Electrical Engineering)

CERTIFICATE BY THE EXAMINERS

This is to certify that the project report entitled **"SIGN LANGUAGE RECOGNITION USING IMAGE PROCESSING"** which is submitted by

S.No.	Name of Student	Roll Number	Enrollment Number	Signature
1.	**ANISHA PAUL**	**301602518041**	**BF3230**	
2.	**ANKIT KUMAR**	**301602518011**	**BF3200**	
3.	**MOHIT KUMAR GUPTA**	**301602518034**	**BF3223**	
4.	**YANENDRA KUMAR**	**301602518049**	**BF3238**	

has been examined by the undersigned as a part of the examination for the award of the degree of Bachelor of Engineering in Electrical Engineering from Chhattisgarh Swami Vivekanand Technical University, Bhilai.

_____ _____

(Signature of the External Examiner) (Signature of the Internal Examiner)

_____ _____

(Name of the External Examiner) (Name of the Internal Examiner)

Date: Date:

Designation: Designation:

Institute: Institute:

ACKNOWLEDGEMENT

We sincerely acknowledge with deep gratitude the valuable guidance from **MISS. POORVA SHARMA** of Government Engineering College, Raipur Chhattisgarh. She was constant source of imputation during our project and provided her expert and sagacious guidance and unrestrained co-operation. She has given us generous encouragement that we are presenting this project entitled **"SIGN LANGUAGE RECOGNITION USING IMAGE PROCESSING"** We are highly thankful to **Dr. R.S PARIHAR**, Head of Electrical & Electronics Engineering Department, for providing us necessary facilities and co-operation during the course of study. We will also like to avail this opportunity to express our sense of gratitude toward **Principal Dr. M.R. KHAN** of our institute for their inspiration and fruitful advice and active support. There are many people who have helped and supported us and we would like to take this opportunity to thank every-one.

S.No.	Name of Student	Roll Number	Enrollment Number	Signature
1.	**ANISHA PAUL**	**301602518041**	**BF3230**	
2.	**ANKIT KUMAR**	**301602518011**	**BF3200**	
3.	**MOHIT KUMAR GUPTA**	**301602518034**	**BF3223**	
4.	**YANENDRA KUMAR**	**301602518049**	**BF3238**	

ABSTRACT

Sign language is a visual non-verbal communication. Which use hand gestures and movements of head, eyes, eyebrows to express and communicate the feelings of one person to another. But this communication usually used by some special people who are deaf and dumb. These kinds of people can't express their feelings, messages and information by just simply telling to us, they need some special kind of actions and expressions to express, which is quite difficult to understand for the normal person who can express their thoughts through their vocabulary speeches. This creates a communication gap between these special people and the normal people. The number of using sign language is large but the popularity of sign language is less. According to WHO on 22 March 2022, Millions of people across the world live with disabling hearing loss. The vast majority live in low- and middle-income countries where they often do not have access to appropriate ear and hearing care services.[1] Each and every person having so many talents but due to some lack of education, knowledge and support those talents are not express by the person, those talents become hidden for them. Similarly, these deaf and dumb community is large but the level of education and providing knowledge and support is less.

In our minor project "SIGN LANGUAGE RECOGNITION SYSTEM USING ML." we try to overcome these problems by using a device known as smart gloves which is made up of flex sensors, Arduino Nano BLE 33 BLE, jumper wires, PCB and many electronics equipment. But we realised that these gloves become more expensive for the poor and middle-class people. This again cause a barrier with the benefits and facility for those needy people.

So, in our major project we try to overcome these problems by simply applying the technology and the knowledge we gain throughout the period of study, by making a software which work as a digital image recognition system which captures the image of hand gestures and convert it into text and voice. With the help of neural network of image processing, the future of the technology. This creates a friendly and handy system to achieve and fulfil our goal for these special people who can able to join the world without any hesitation and feeling shy due to their speciality. We use the PyCharm community edition 2021.3.3 software which use the python language.

Keywords: - American sign language (ASL), image processing, sign language, human computer interaction (HCI), real time conversion, Frames per second (FPS)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER- 1
INTRODUCTION

Body language is an important way of communication among humans. Normal people can communicate their thoughts and ideas to others through speech. The only means of communication method for the hearing-impaired community is the use of sign language. The hearing-impaired community has developed their own culture and methods to communicate among themselves and with ordinary person by using sign gestures. Instead of conveying their thoughts and ideas acoustically they convey it by means of sign patterns. Sign gestures are a non-verbal visual language, different from the spoken language, but serving the same function. It is often very difficult for the hearing-impaired community to communicate their ideas and creativity to the normal humans. This system was inspired by the special group of people who have difficulties communicate in verbal form. It is designed with the ease of use for the deaf or hearing-impaired people. The objective of this research is to develop a system prototype that automatically helps to recognize sign languages of the signer and translate them into voice in real time.[3]

1.1 HISTORY

One of the earliest written references to a sign language is from the fifth century BC, in Plato's Cratylus, where Socrates says: "If we hadn't a voice or a tongue, and wanted to express things to one another, wouldn't we try to make signs by moving our hands, head, and the rest of our body, just as dumb people do at present?"[4]

In the Middle Ages, monastic sign languages were used by a number of religious orders in Europe since at least the 10th century. These are not true "sign languages", however, but well-developed systems of gestural communication.

In Native American communities prior to 1492, it seems that Plains Indian Sign Language existed as an extensive lingua franca used for trade and possibly ceremonies, story-telling and also daily communication by deaf people.[5] Accounts of such signing indicate these languages were fairly complex, as ethnographers such as Cabeza de Vaca described detailed communications between them and Native Americans that were conducted in sign. In the 1500s, a Spanish expeditionary, Cabeza de Vaca, observed natives in the western part of modern-day Florida using signs,[6] and in the mid-16th century Coronado mentioned that communication with the Tonkawa using signs was possible without a translator.

Some sign languages are known to have developed spontaneously in small communities with a high number of deaf members. Martha's Vineyard, an island in Massachusetts, USA was settled by people carrying a gene causing deafness in the late 17th century. Limited outside contact and high inter-marriage on the island led to a high density of deaf individuals on the island, peaking around 1840.[7] This environment proved ideal for the development of what is today known as Martha's Vineyard Sign Language, which was used by hearing and deaf islanders alike until increased mixing with the outside world reduced the incidence of deafness on the island. They created a sign language that had specific signs relevant to that area, such as native types of fish and berries.[8] Almost all of the school-aged population became students at ASD, which led to mutual influence of American Sign Language and Martha's Vineyard Sign Language on each other.[9] Other examples include:

Al-Sayyid Bedouin Sign Language, Israel

Kata Kolok, Bali

Adamorobe Sign Language, Ghana

Yucatec Maya Sign Language, Mexico [10]

The events that occurred in the history of sign language are actually pretty shocking. How deaf people experience life today is directly related to how they were treated in the past. It wasn't long ago when the deaf were harshly oppressed and denied even their fundamental rights.

The are many famous deaf people who have made a name for the deaf throughout the history of sign language and proved that deaf people can, in fact, make history.

The examples are given in the below:

1. Juan Pablo de Bonet, a Spanish priest, studied Leon's successful methods and was inspired to teach deaf people using his own methods. Bonet used the methods of writing, reading, and speechreading as well as his manual alphabet to educate the deaf. His manual alphabet system was the first recognized in Deaf history. The handshapes in this alphabet corresponded to different sounds of speech.

Figure: 1.1 Juana Pablo de Bonet [11]

2. Abbe Charles Michel de L'Epee established the National Institute for Deaf-Mutes in 1771. This was the first public free deaf school. Deaf children came from all across France to attend the school and brought the signs they learned from signing at home with them to the school. L'Epee learned all of these different signs and utilized the signs he learned from his students to teach his students French.

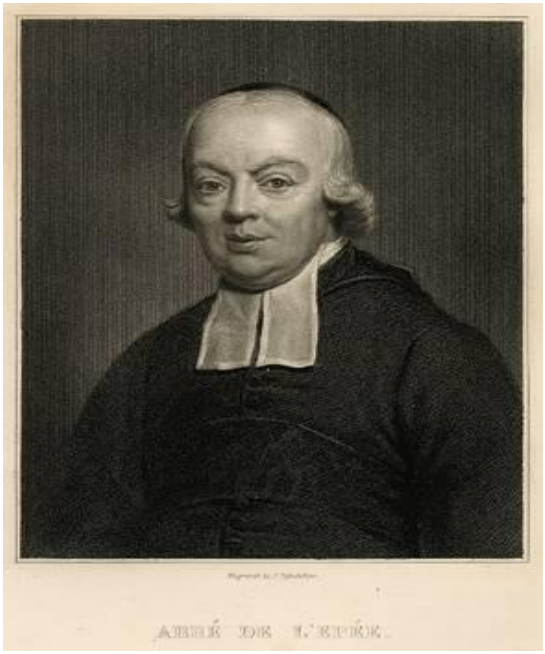

Figure: 1.2 Abbe Charles Michel De L'Epee [11]

3. In relation to the deaf-blind, the first deaf-blind person to be educated was Laura Bridgman. She was born 50 years before Helen Keller, but is usually not credited with being the first deaf-blind person to learn language.

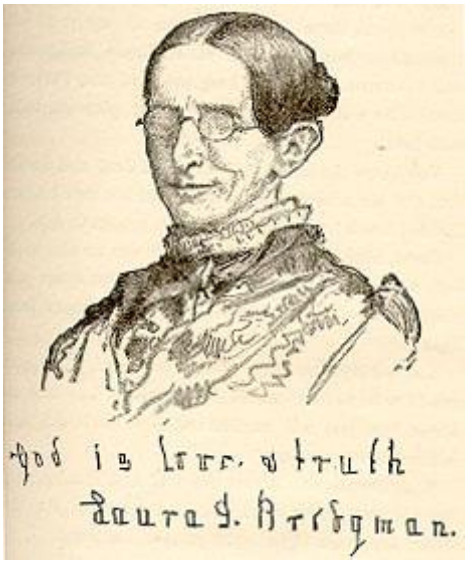

Figure: 1.3 Laura Bridgman [11]

4. Helen Keller is the most well-known deaf-blind person (she has taken the credit before Laura Bridgman). Her teacher was Anne Sullivan and while Helen Keller wasn't the first deaf-blind person to be educated, she was the first one to graduate from college, and she did it with honors.

Figure 1.4 Helen Keller [11]

5. Probably the most devoted supporter of the oralism method was Alexander Graham Bell (yes, the man who is credited with inventing the telephone). Bell started an institution in Boston in 1872 to train teachers of deaf people to use oral education. He was one person in the history of sign language who really tried to damage the lives of deaf people. In 1890, he founded an organization that is now known as the Alexander Graham Bell Association for the Deaf.

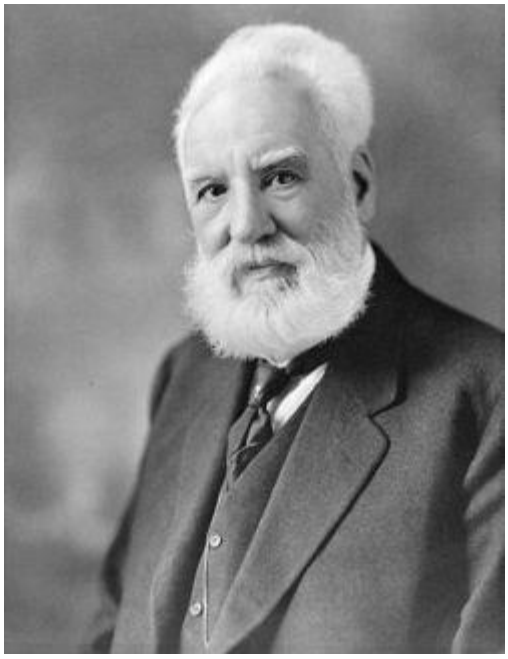

Figure: 1.5 Alexander Graham Bell [11]

1.2 IDEA BEHIND THIS PROJECT

The 2011 Indian census cites roughly 1.3 million people with "hearing impairment". In contrast to that numbers from India's National Association of the Deaf estimates that 18 million people –roughly 1 per cent of Indian population are deaf. These statistics formed the motivation for our project. As these speech impairment and deaf people need a proper channel to communicate with normal people there is a need for a system. Not all normal people can understand sign language of impaired people. Our project hence is aimed at converting the sign language gestures into text that is readable for normal people.[12]

1.3 PROBLEM STATEMENT

Case I - Consider two Especially Abled People. One person is Blind, and the other person is Dumb, so if they want to communicate with each other, they have a problem doing that. Also, when a Dumb person uses sign language to communicate with a blind person, a blind person cannot understand the sign language. Generally, deaf and dumb people use sign language for communication. Still, they find it challenging to communicate with others who don't understand sign language. Sign language is an expressive and natural way to communicate between ordinary and dumb people (mainly through hand gestures).

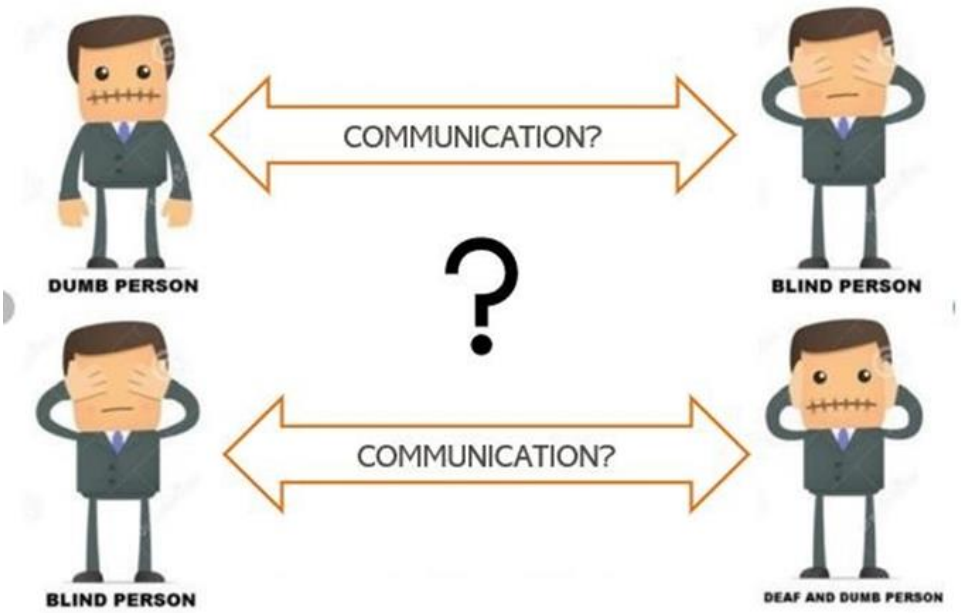

Figure:1.6 Communication Barrier (a)

Case II - Consider two people who want to comminate. One person is average (with no physical disability). Another one is DUMB, so now what happens is that Generally deaf and dumb people use sign language for communication. Still, they find it challenging to communicate with others who don't understand sign language. Sign language is an expressive and natural way to communicate between normal and dumb people (information is mostly conveyed through hand gestures).

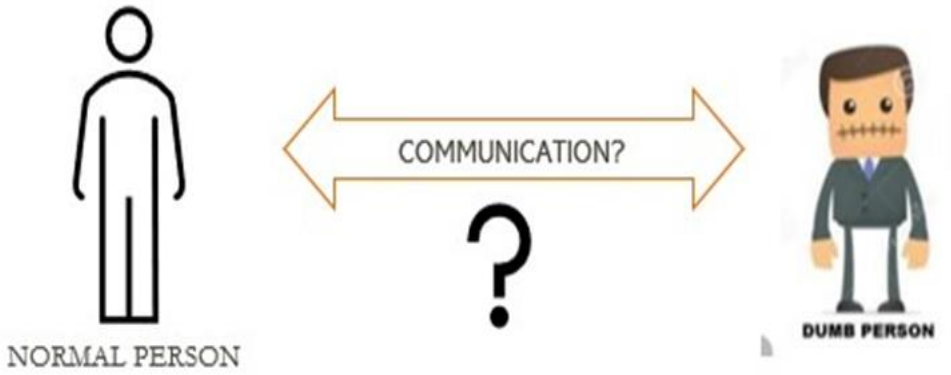

Figure: 1.7 Communication Barrier (b)

1.4 SOLUTIONS

As mentioned above, we have to find a way to overcome this problem. We researched this area and found two solutions for this problem.

a) The first solution is to make smart gloves that can measure the gesture by using a flex sensor and translating it to the digital data quantity. This can be further processed by using a Microcontroller and P.C. Which we have done in our minor project.

b) The second solution is to make a virtual environment so that the computer can convert sign/gesture language to the desired output. It is basically done by gesture detection using a Camera. and some machine learning process.

We currently research on this method which having much advantages then the gloves.

1.5 PREVIOUSLY PROPOSED WORK

We done our minor project by making a hardware implementation of smart gloves. A Wireless data glove is used, which is a normal cloth driving glove fitted with flex sensors along the length of each finger. Mute people can use gloves to perform hand gestures, and they will be converted into speech so that normal people can understand their expressions. A gesture in sign language is a particular movement of the hands with a specific shape made out of them.

A sign language glove is an electronic device that attempts to convert the motions of a sign language into written or spoken words. Some critics of such technologies have argued that the potential of sensor-enabled gloves to do this is commonly overstated or

misunderstood because many sign languages have a complex grammar that includes the use of the sign space and facial expressions (non-manual elements).

Figure: 1.8 Minor Project Model "Smart Gloves"

1.6 PROBLEMS WE FACED DURING COMPLETION OF THE PROJECT

Our previously proposed work done in our minor project by a hardware implementation of the hand gloves known as "SMART GLOVES".

The smart gloves are a wireless system in which microcontroller, PCB, flex sensors are used which may cause damage due to the repetition of use. This electronics equipment increases the cost for the average peoples who can't afford to buy the smart glove which does not make it a user-friendly system.

So, to overcome this problem we use PyCharm software for making this software implementation. We also use MATLAB software but we faced some implementation problems and subscription problems which take too much time to proceed so we go through with this PyCharm software which is an open-source software, which is easy to use.

1.7 OBJECTIVES

1. To develop an automatic sign language recognition system with the help of image processing and computer vision techniques.
2. To use natural image sequences, without the signer having to wear data gloves or colored gloves, and to be able to recognize hundreds of signs.
3. The motivation for this work is to provide real time interface so that signers can easily and quickly communicate with non-signers.

CHAPTER- 2

LITERATURE REVIEW AND THEORY

Human beings interact with each other to convey their ideas, thoughts, and experiences to the people around them. But this is not the case for deaf-mute people. Sign language paves the way for deaf-mute people to communicate. Through sign language, communication is possible for a deaf-mute person without the means of acoustic sounds. The aim of this work is to develop a system for recognizing sign language, which provides communication between people with speech impairment and normal people, thereby reducing the communication gap between them. Compared to other gestures (arm, face, head, and body), hand gesture plays an important role, as it expresses the user's views in less time. In the current work flex, a sensor-based gesture recognition module is developed. Generally, dumb people use sign language for communication, but they find difficulty in communicating with others who don't understand sign language. This project aims to lower this barrier in communication. It is based on the need to develop an electronic device that can translate sign language into speech in order to make the communication take place between the mute communities with the general public possible. software-based model sign language converter. we are making a software model which convert different signs of hand into text or audio file. We will build a more efficient and more accessible software-based model. we are doing this by using the application of PYTHON in digital image processing and machine learning model. Sign language is the language used by mute people, and it is a communication skill that uses gestures instead of sound to convey meaning, simultaneously combining hand shapes, orientations, and movement of the hand's arms or body and facial expressions to express a speaker's thoughts fluidly. Signs are used to communicate words and sentences to the audience. This system facilitates communication between silent, hearing-impaired, blind people and normal people. It also helps the mute, hearing-impaired, and blind to interact among themselves. It is not an easy task for normal people to perceive the intended meaning of these sign language used by the hearing-impaired and silent. Moreover, blind people cannot watch their gestures. Official sign language is used by the dumb and deaf but is not familiar with the normal world, and the people who are blind cannot follow sign language. This Application converts gestures into voice and vice-versa, which is suitable for both disabled and normal people. To help deaf people, the gestures are converted into text. This text gets displayed on a screen.

2.1 SIGN LANGUAGE

Sign languages (also known as signed languages) are languages that use the visual-manual modality to convey meaning. Sign languages are expressed through manual articulations in combination with non-manual elements. Sign languages are full-fledged natural languages with their own grammar and lexicon. Sign languages are not universal and are usually not mutually intelligible, although there are also similarities among different sign languages.[2]

Linguists consider both spoken and signed communication to be types of natural language, meaning that both emerged through an abstract, protracted aging process and evolved over time without meticulous planning. Sign language should not be confused with body language, a type of nonverbal communication.

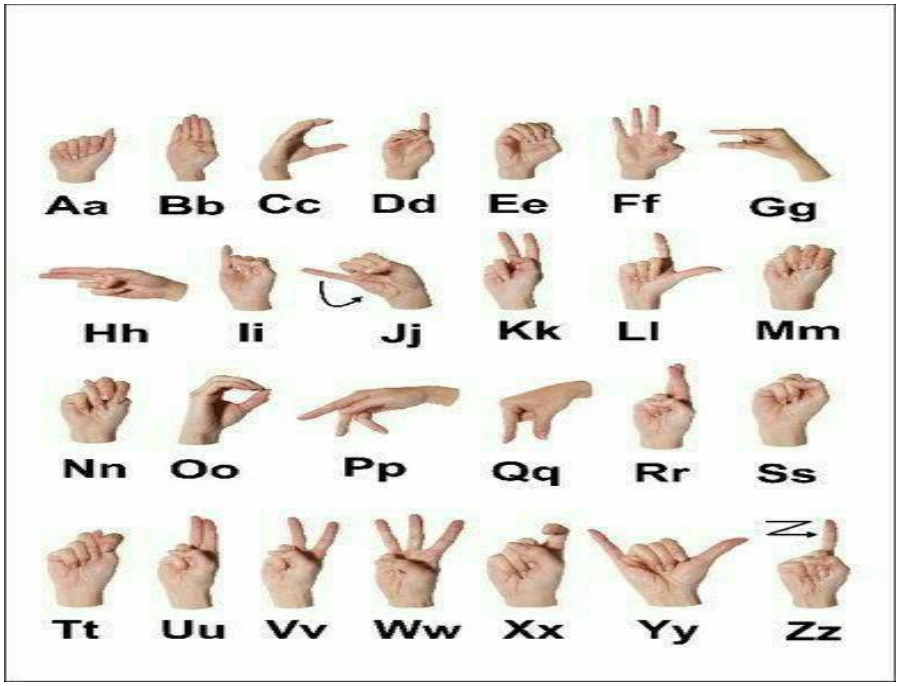

Figure: 2.1 Sign language Alphabets [13]

2.2 PYTHON

In technical terms, Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development. It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options.

Figure: 2.2 Logo of Python

Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability. Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers.

Additionally, Python supports the use of modules and packages, which means that programs can be designed in a modular style and code can be reused across a variety of projects. Once you've developed a module or package you need, it can be scaled for use in other projects, and it's easy to import or export these modules.[14]

We used libraries, modules to create a software.

2.3 PYTHON LIBRARIES

Normally, a library is a collection of books or is a room or place where many books are stored to be used later. Similarly, in the programming world, a library is a collection of precompiled codes that can be used later on in a program for some specific well-defined operations. Other than pre-compiled codes, a library may contain documentation, configuration data, message templates, classes, and values, etc.

A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer. As we don't need to write the same code again and again for different programs. Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc. [15]

Why is Python so popular?

According to the TIOBE index, which measures the popularity of programming languages, Python is the third most popular programming language in the world, behind only Java and C. There are many reasons for the ubiquity of Python, including:

Its ease of use- For those who are new to coding and programming, Python can be an excellent first step. It's relatively easy to learn, making it a great way to start building your programming knowledge.

Its simple syntax- Python is relatively easy to read and understand, as its syntax is more like English. Its straightforward layout means that you can work out what each line of code is doing.

Its thriving community- As it's an open-source language, anyone can use Python to code. What's more, there is a community that supports and develops the ecosystem, adding their own contributions and libraries. Its versatility. As we'll explore in more detail, there are many uses for Python. Whether you're interested in data visualisation, artificial intelligence or web development, you can find a use for the language.

2.3.1 TENSOR FLOW

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

Figure: 2.3 Tensor Flow Logo [16]

Features: TensorFlow provides stable Python (for version 3.9 across all platforms) and C APIs; and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift (early release). Third-party packages are available for C#, Haskell Julia, MATLAB, R, Scala, Rust, OCaml, and Crystal. "New language support should be built on top of the C API. However, not all functionality is available in C yet." Some more functionality is provided by the Python API.

Application: Among the applications for which TensorFlow is the foundation, are automated image-captioning software, such as Deep Dream.

2.3.2 OpenCV

OpenCV (Open-Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.[1] Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel. The library is cross-platform and free for use under the open-source BSD license. OpenCV's application areas include: 2D and 3D feature toolkits

- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM).
- Motion tracking
- Augmented reality

To support some of the above areas, OpenCV includes a statistical machine learning library that contains:

- Boosting
- Decision tree learning
- Gradient boosting trees
- Expectation-maximization algorithm
- k-nearest neighbor algorithm
- Naive Bayes classifier
- Artificial neural network
- Random forest
- Support vector machine (SVM)
- Deep neural networks (DNN)

Figure: 2.4 OpenCV Logo [17]

AForge.NET, a computer vision library for the Common Language Runtime (.NET Framework and Mono). ROS (Robot Operating System). OpenCV is used as the primary vision package in ROS. VXL, an alternative library written in C++.Integrating Vision Toolkit (IVT), a fast and easy-to-use C++ library with an optional interface to OpenCV. CVIPtools, a complete GUI-based computer-vision and image-processing software environment, with C function libraries, a COM-based DLL, along with two utility programs for algorithm development and batch processing. OpenNN, an open-source neural networks library written in C++.

List of free and open-source software packages

- OpenCV Functionality
- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching,videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, Flann)
- CUDA acceleration (gpu)

2.3.3 MEDIA PIPE

Media Pipe is Google's open-source framework, used for media processing. It is cross-platform or we can say it is platform friendly. It is run on Android, iOS, web, and YouTube servers that's what Cross-platform means, to run everywhere.

uses of media pipe

Every YouTube video we watch is processed with machine learning models using Media Pipe. Google has not hired thousands of employees to watch every video people upload, because thousands of people are not enough to look after and check each published video, the amount of data Google gets daily is not easy for humans to check. Machine Learning models are developed to make our life easier, so tasks that are hard for us to complete, machine learning and deep learning models help us to do in less amount of time, on the other hand, we can save money by not hiring employees.

Figure: 2.5 Media Pipe [18]

Yes, Google has machine learning/deep learning models to see if the videos match their policies and the content is not having copy-right issues.

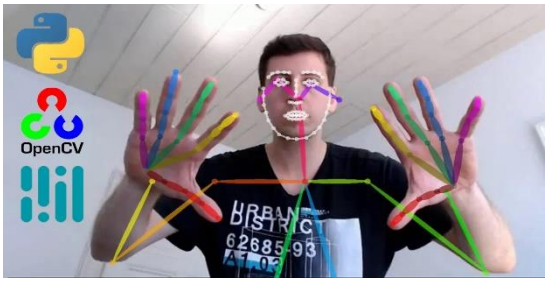

Figure: 2.6 How Media Pipe Work [19]

Basically, Media Pipe is a framework for Computer Vision and Deep Learning that builds perception pipelines. For now, you just need to know, perception pipelines are some sort of audio, video, or time-series data that catch the process in pipelining zone.

2.3.4 NumPy

The name "NumPy" stands for "Numerical Python". It is the commonly used library. It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations. Even libraries like TensorFlow use NumPy internally to perform several operations on tensors. Array Interface is one of the key features of this library.

Figure: 2.7 NumPy Logo [20]

2.4 IMAGE PROCESSING

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too. Image processing basically includes the following three steps:

1. Importing the image via image acquisition tools;
2. Analyzing and manipulating the image;
3. Output in which result can be altered image or report that is based on image analysis.

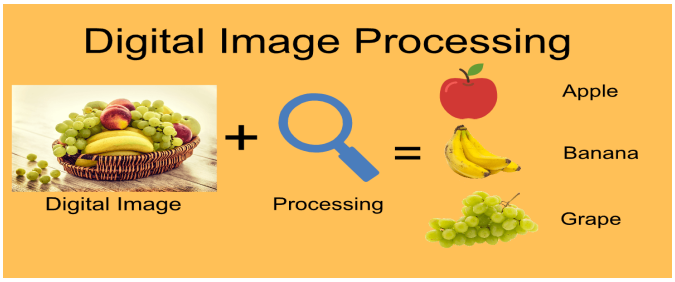

Figure: 2.8 Digital Image Processing [21]

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

2.5 GESTURE RECOGNITION IN IMAGE PROCESSING

Gesture recognition is the fast-growing field in image processing and artificial technology. The gesture recognition is a process in which the gestures or postures of human body parts are identified and are used to control computers and other electronic appliances.

There are different ways of gesture recognition such as Hand, Face and Body Gesture Recognition explained below: 1) Hand Gesture Recognition a) Glove-based Hand Gesture Recognition A glove-based system requires the user to be connected to the computer.

Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse and interact naturally.

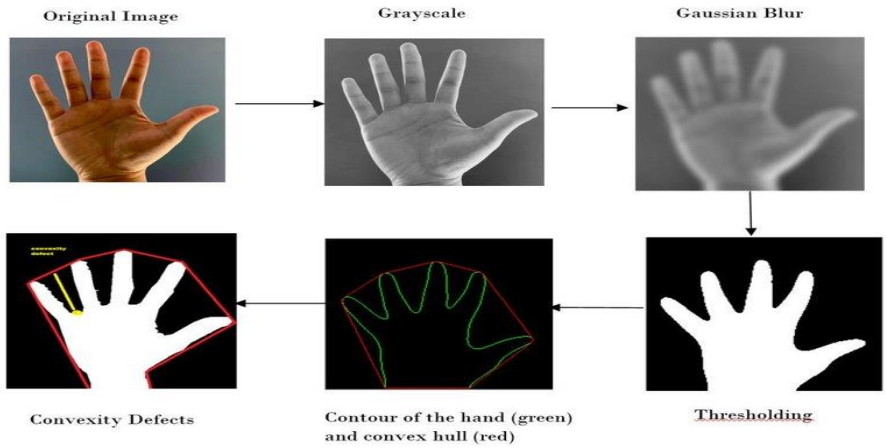

Figure: 2.9 Image Processing Steps for Hand Gesture Recognition [22]

2.6 ADVANTAGES OVER HARDWARE BASED MODEL

1. Instead of using High - end technology like gloves equipped with sensors, we aim to solve this problem using computer vision and Machine learning algorithms.
2. It makes the deaf and dumb people communicate faster and easier with outer world.
3. The CNN is more efficient and reliable than the other clustering algorithms in many applications.
4. This is a technology of the future that can be of great utility.

CHAPTER-3
METHODOLOGY

3.1 INTRODUCTION

This project introduce the implementation of image processing based fast and effiicient way for sign language detection. The basic objective of this project is to develop a computer based intelligent system using image processing, machine learinings, artificial intelligence concepts to take visual inputs of sign language's hand gestures and generate easily recognizable form of outputs. The methodology lies withing mini step by step processes which will describe below-

3.1.1 BLOCK-DIAGRAM

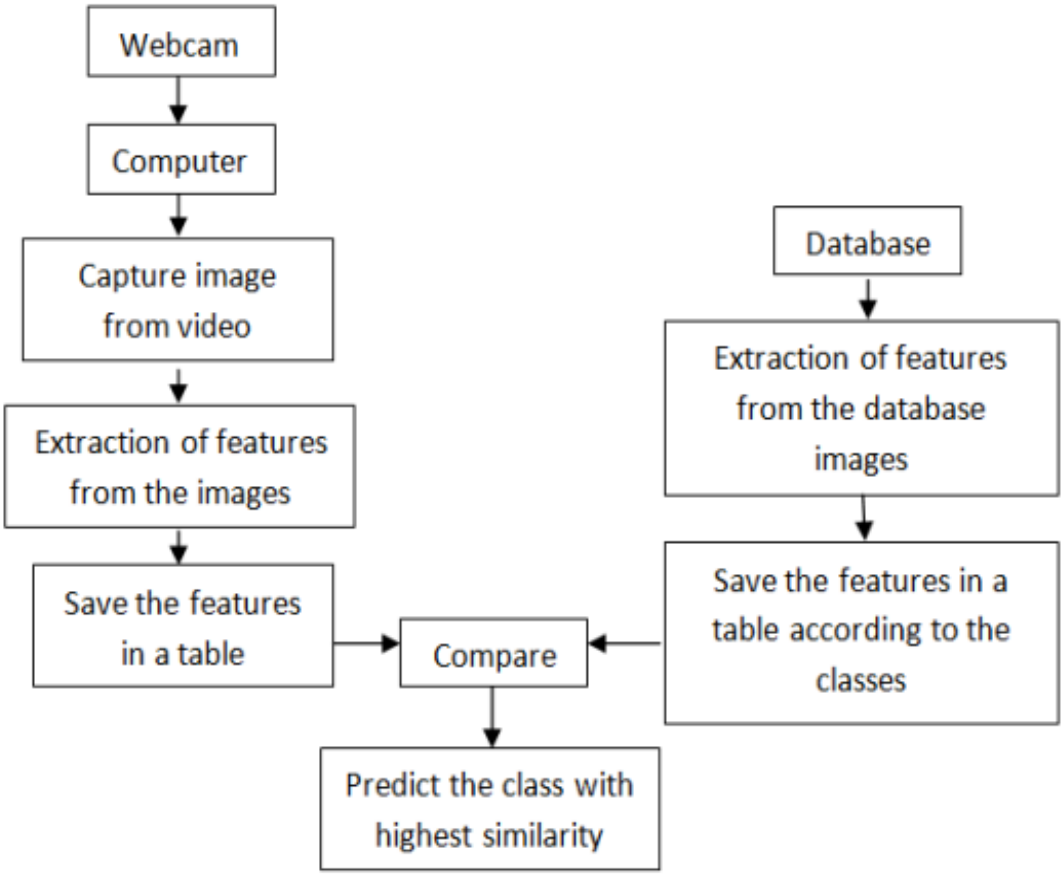

Figure: 3.1 Sytem Block Diagram

Figure Shoes the overall idea of proposed system. The system consist of 4 such modules. Image is captured through the webcam. The camera mounted on top of system facing towards user hand. Our processing systems captures all these images frame by frame and collects them into the form packets. Every image has it's own features, so all these features are classified in table.

On other hand sytems has it's own dataset unit or collection unit, it has large amount of trained datasets, these datasets are previously performed sign gestures. For our proposed system, the libraries we are using is fill with millions of training datasets, Also it has feature to collect more training datasets for each iteration/gesture we perform so that the system can accuretly Extract the output. Also it has feature to collect more training datasets for each iteration/gesture we perform

Comparison module consist of classifiaction and sensing unit , it is the processing unit of the whole system. For recognintion of each sign/gestures, coordinates of captured image is calculated with respect to X and Y coordinates. The calculated are compared and with existing one. If comparison leads to success then the same will be converted into audio and texted form.

The system works in two different mode i.e training mode and operaational mode. Training mode is the part of machine learning where we are training our system to accomplish the task for which it is implemented i.e Alaphabet Recognintion.

3.1.2 ALGORITHM

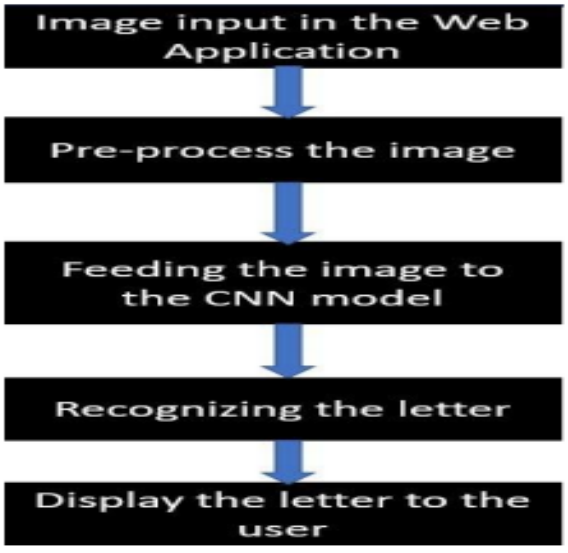

Figure: 3.2 Proposed Frame Workflow

3.1.3 SYSTEM ARCHITECTURE

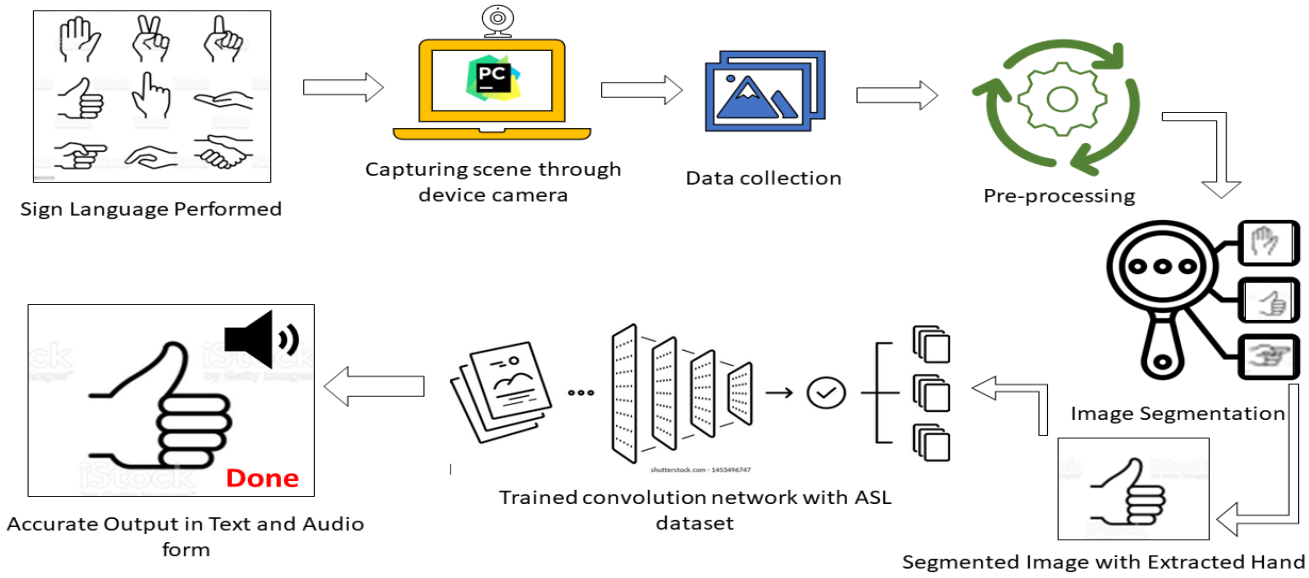

Figure: 3.3 Architecture diagram of sign Language recognition system

3.2 CONSTRUCTION

3.2.1 WORKPLAN

Our workplane is to practically implement this sytem and find the accuracy level for each gesture detection. To do so we need to perform all these mini processes step by step. we are doing this by using the application of PyCharm in digital image processing and machine learning model. The application of PyCharm in image processing and how we're going to implement our software-based system is described below-

Step by Step processes-

Step-1 Software installation

Step-2 Practical implementation

Step-3 Dataset collection

Step-4 Testing the outputs

3.3 SOFTWARE INSTALLATION

3.3.1 PYTHON INSTALLATION

- Downloading
- Click https://www.python.org/downloads/ The following page will appear in your browser.

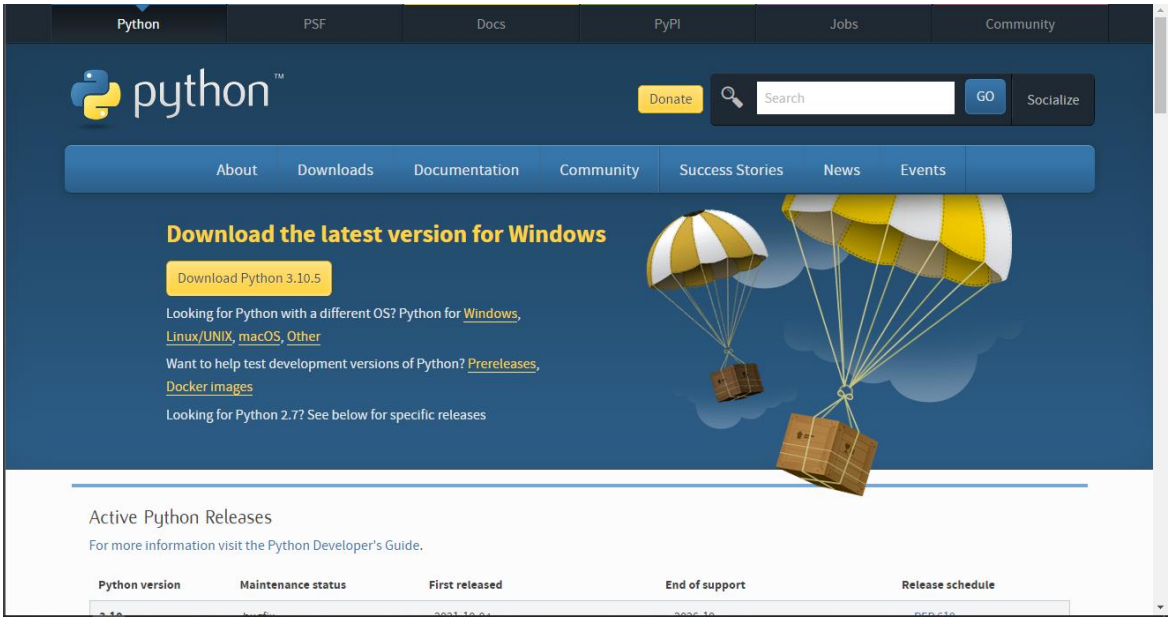

Figure: 3.4 Front Page of Python Installation

- Click the Download Python 3.10.5 button. The file named python 3.10.5.exe should start downloading into your standard download folder.
- Double-click the icon labelling the file python 3.10.5.exe.An Open File - Security Warning pop-up window will appear.
- Click Run. A Python 3.10.5 (64-bit) Setup pop-up window will appear. Ensure that the Install launcher for all users (recommended) and the Add Python 3.10.5 to PATH checkboxes at the bottom are checked. If the Python Installer finds an earlier version of Python installed on your computer, the Install Now message may instead appear as Upgrade Now(and the checkboxes will not appear).
- Highlight the Install Now (or Upgrade Now) message, and then click it. A User Account Control pop-up window will appear, posing the question Do you want the allow the following program to make changes to this computer?
- Click the Yes button. A new Python 3.10.5 (64-bit) Setup pop-up window will appear with a Setup Progress message and a progress bar. During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new Python 3.10.5 (64-bit) Setup pop-up window will appear with a Setup was successfully message.
- Click the Close button.
- Python should now be installed.

3.3.2 PYCHARM INSTALLATION

Here is a step-by-step process on how to download and install PyCharm IDE on Windows:

Step 1) To download PyCharm visit the website https://www.jetbrains.com/pycharm/download/ and click the "DOWNLOAD" link under the Community Section.

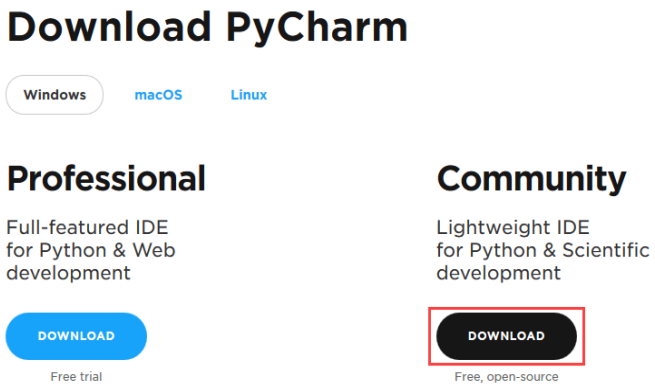

Figure:3.5 Download PyCharm

Step 2) Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click "Next"

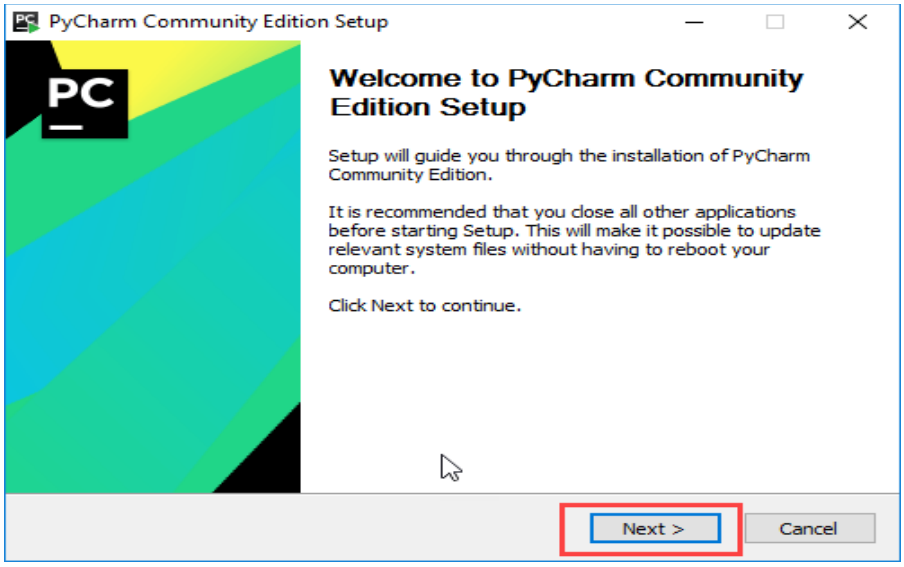

Figure: 3.6 Install PyCharm

Step 3) On the next screen, Change the installation path if required. Click "Next".

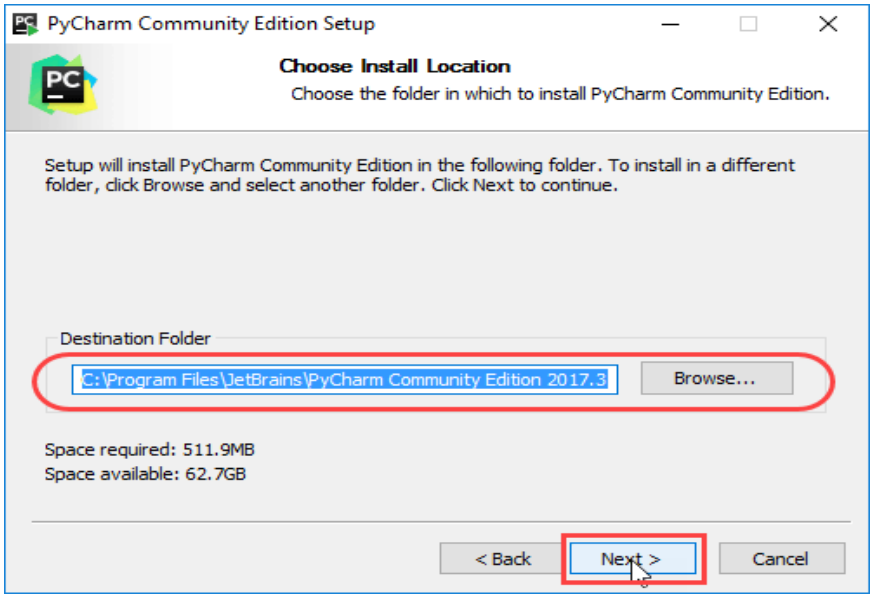

Figure: 3.7 Changing the Installation Path

Step 4) On the next screen, you can create a desktop shortcut if you want and click on "Next".

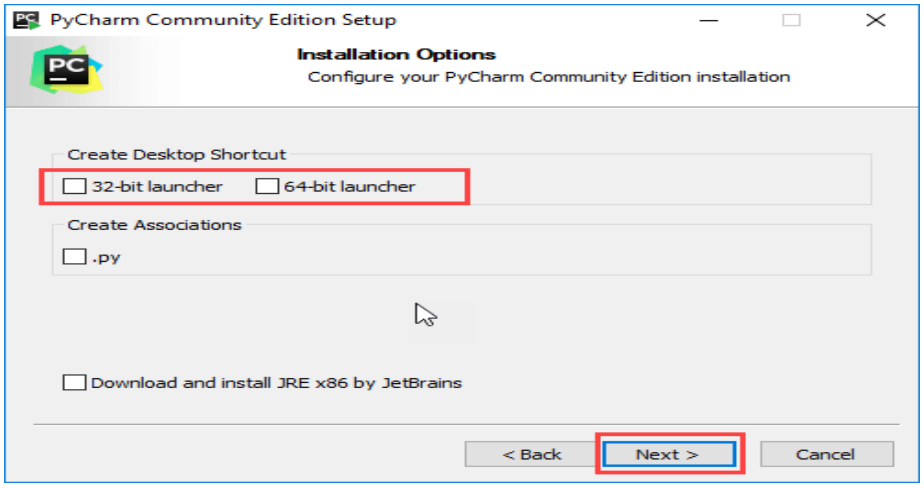

Figure: 3.8 Creating Shortcut

Step 5) Choose the start menu folder. Keep selected JetBrains and click on "Install".

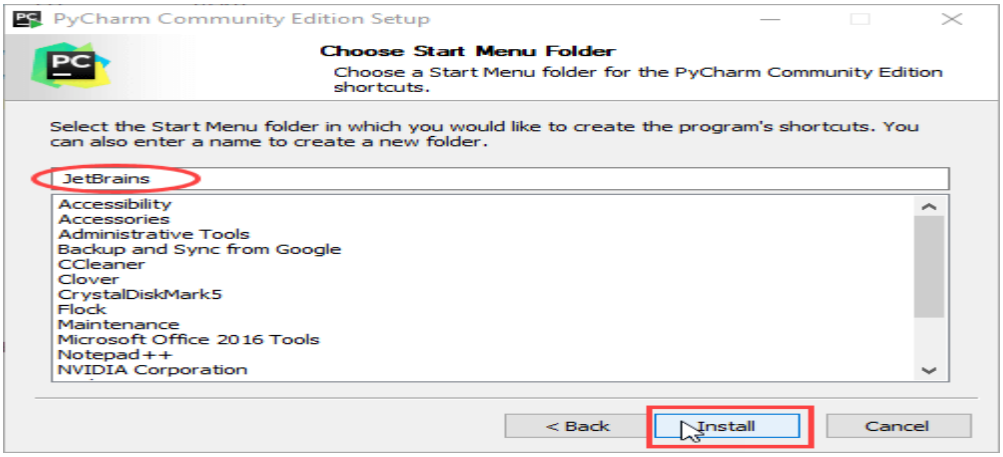

Figure: 3.9 Choose Start Menu

Step 6) Wait for the installation to finish.

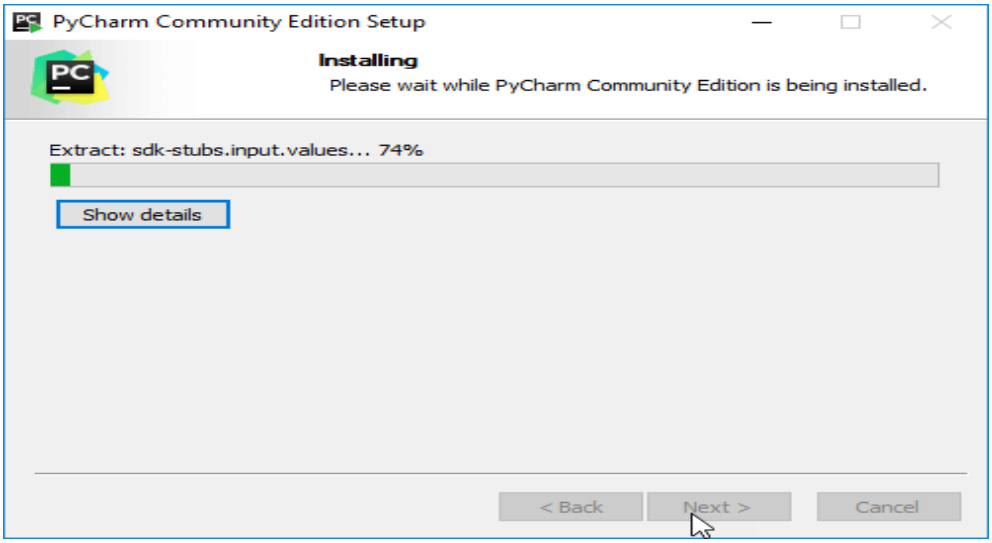

Figure: 3.10- Installing

Step 7) Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the "Run PyCharm Community Edition" box first and click "Finish".

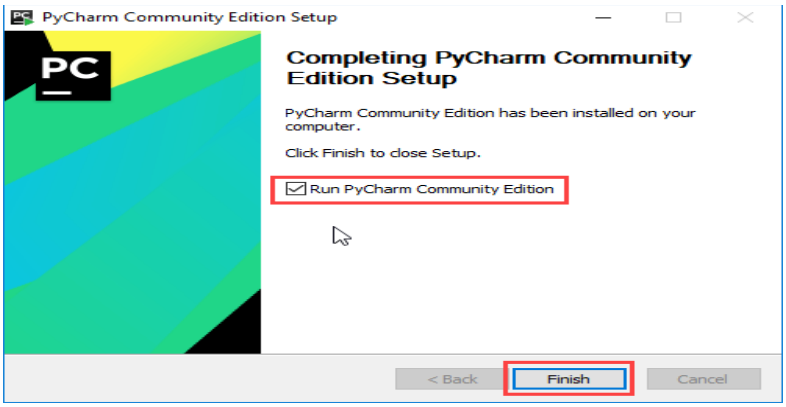

Figure: 3.11 Completing PyCharm

Step 8) After you click on “Finish,” the Following screen will appear.

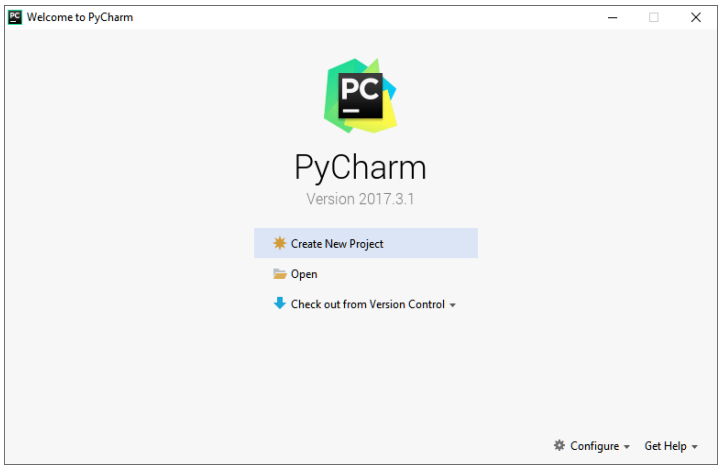

Figure:3.12 Final Page

3.4 PRACTICAL IMPLEMENTATION

3.4.1 MATLAB IMPLEMENTATION

We’ve tried to build our project by using the MATLAB software, but we faced the following problems and concluded that the MATLAB is not identical for this project.

Problems associated with MATLAB IMPLEMENTATION-

- Problems associated with MATLAB
- MATLAB is interpreted language and hence it takes more time to execute
- It requires fast computer with enough memory.
- It is expensive
- It is difficult to develop real time applications using MATLAB
- It is not free and hence users need to obtain licensed version from MathWorks

3.4.2 PYCHARM IMPLEMENTATION

We have faced a lot of problems while using the MATLAB software for this project design. So, moving forward we researched and found that the, Python Can be an excellent choice for these types of image processing tasks due to its growing popularity as a scientific programming language and the free availability of many state-of-the-art image processing tools in its ecosystem, also it has the vast number of libraries and packages available for Python. Libraries contain code for certain basic functions so that programmers don't have to write them from scratch.

It is an interface for open-source machine vision libraries in Python with a readable interface for cameras, image manipulation, format conversion, feature extraction, and many more. It helps in making computer vision tasks very easy and simple over-complex code.

So, moving forward, to implement our project model in python language, PyCharm is the one of the best IDE (integrated development Environment), With PyCharm, you can access the command line, connect to a database, create a virtual environment, and manage your version control system all in one place, saving time by avoiding constantly switching between windows.

3.4.3 ADVANTAGES OF USING PyCharm IDE

- A plethora of productive shortcuts.
- Ability to view the entire Python source code with a single click.
- Availability of an array of plugins.
- Easy-to-use.
- Excellent community support.
- Facilitates faster code development.
- More powerful, commercial version available.

3.4.4 LIBRARIES INSTALLATION

Firstly, we need to find what are libraries we can use for the sign language detection. For that we researched in various papers published before found that the Python is a very popular and trending programming language with a wide variety of Python libraries. Thus, it is essential to use Python libraries for image processing in completing all machine learning tasks. A lot of libraries are open-source libraries, an open-source library is any library with an open-source license, which denotes software that is free to reuse, modify, and/or publish without permission.

So, moving forward lets' start to setup our PyCharm before coding our sign language recognition models-

How to install external libraries in PyCharm-

By using Settings menu-

- Open **File > Settings > Project** from the PyCharm menu.
- Select your current project.
- Click the **Python Interpreter** tab within your project tab.
- Click the small **+** symbol to add a new library to the project.
- Now type in the library to be installed, for example Pandas, and click **Install Package**.
    - Wait for the installation to terminate and close all popup windows

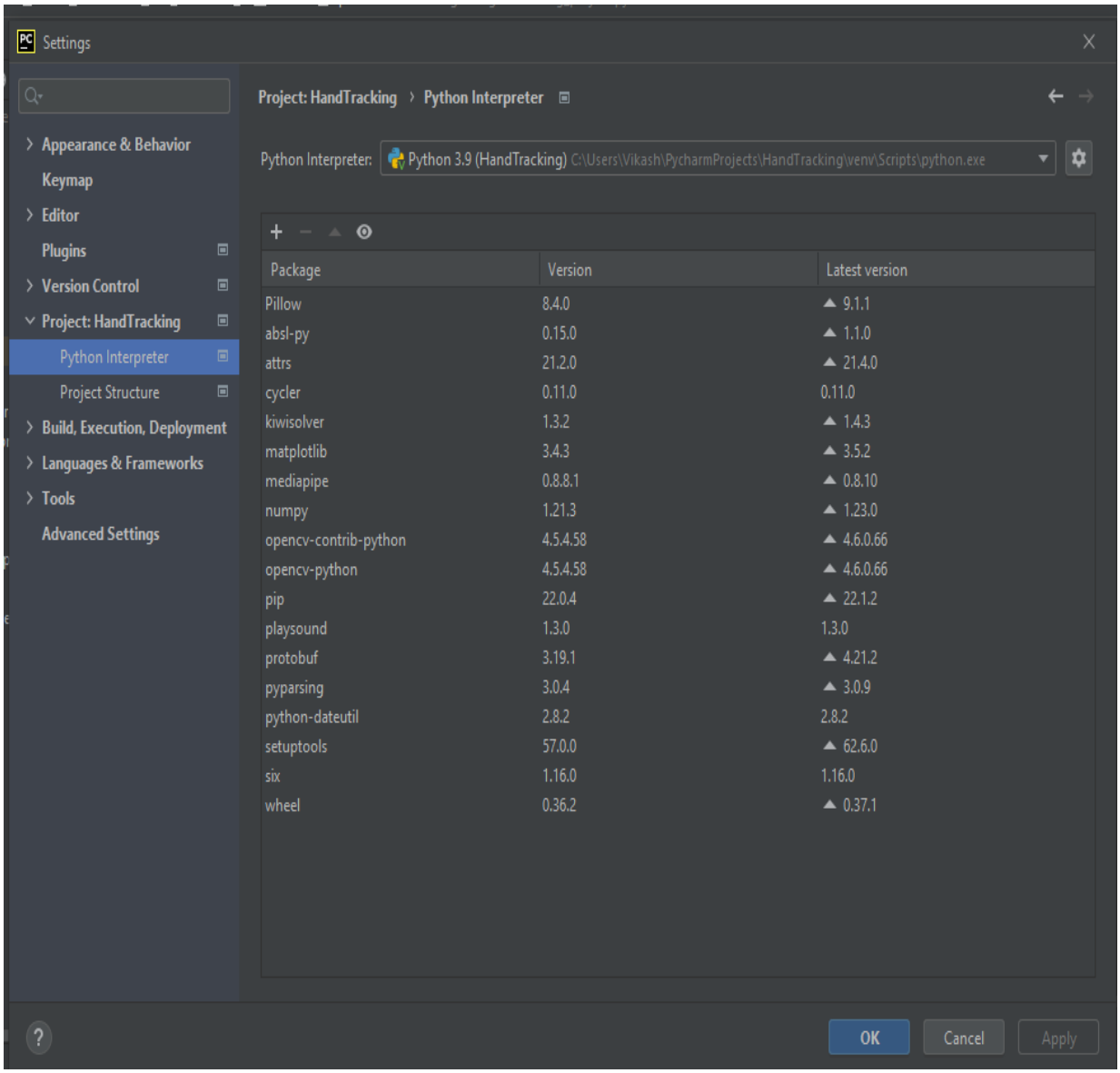

Figure: 3.13 Python Interpreter

By using Terminal/command prompt-

Pip is one of the best tools to install and manage Python packages. Pip has earned its fame by the number of applications using this tool. Used for its capabilities in handling binary packages over the easily installed package manager, Pip enables 3rd party package installations i.e.-

For our project we are using following Packages: -

TABLE.3.1- Python Packages

S.N	Package name	version
1	Pillow	8.4.0
2	absl-py	0.15.0
3	attrs	21.2.0
4	cycler	0.11.0
5	kiwisolver	1.3.2
6	matplotlib	3.4.3
7	mediapipe	0.8.8.1
8	numpy	1.21.3
9	opencv-contrib-python	4.5.4.58
10	opencv-python	4.5.4.58
11	pip	22.0.4
12	playsound	1.3.0
13	protobuf	3.19.1
14	pyparsing	3.0.4
15	python-dateutil	2.8.2
16	setuptools	57.0.0
17	six	1.16.0
18	wheel	0.36.2

- Write python code for the library you want to install-
- run the following command by creating a virtual environment using python 3 and run

  “pip3 install OpenCV-python”

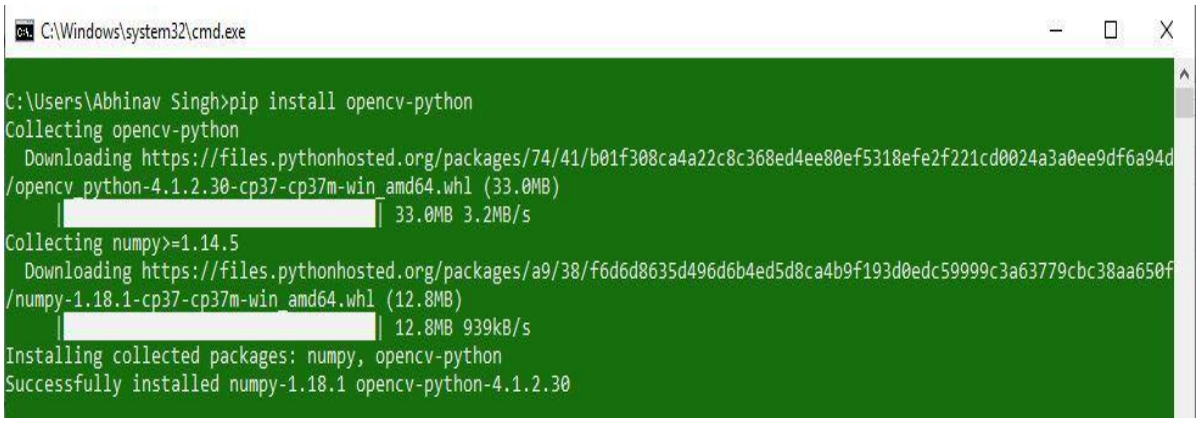

Figure: 3.14 Command Window

- to check it has installed correctly run

  “python3 -c "import cv2"

MediaPipe –

To install mediapipe we have to follow same process as mentioned above-

- Open Terminal/ command prompt in your PC
- Write the code- **“PIP install mediaPipe”**
- The library is installed successfully-

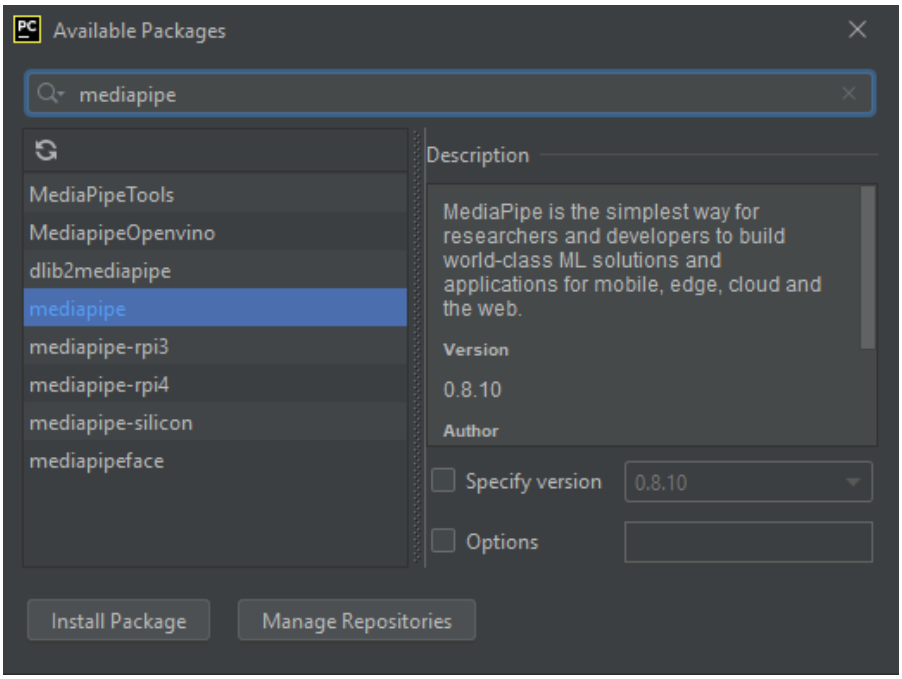

Figure:3.15 Available Package

Playsound- To install play sound we have to follow same process as mentioned above-

- Open Terminal/ command prompt in your PC
- Write the code- **“PIP install Playsound”**
- The library is installed successfully-

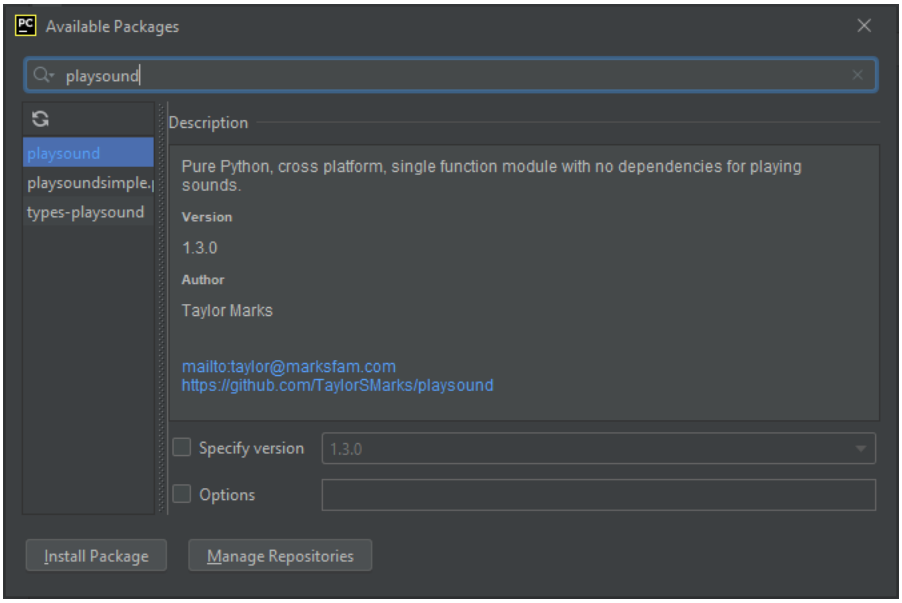

Figure:3.16 Play sound Package

Numpy - To install play sound we must follow same process as mentioned above-

- Open Terminal/ command prompt in your PC
- Write the code- **"PIP install NumPy"**
- The library is installed successfully-

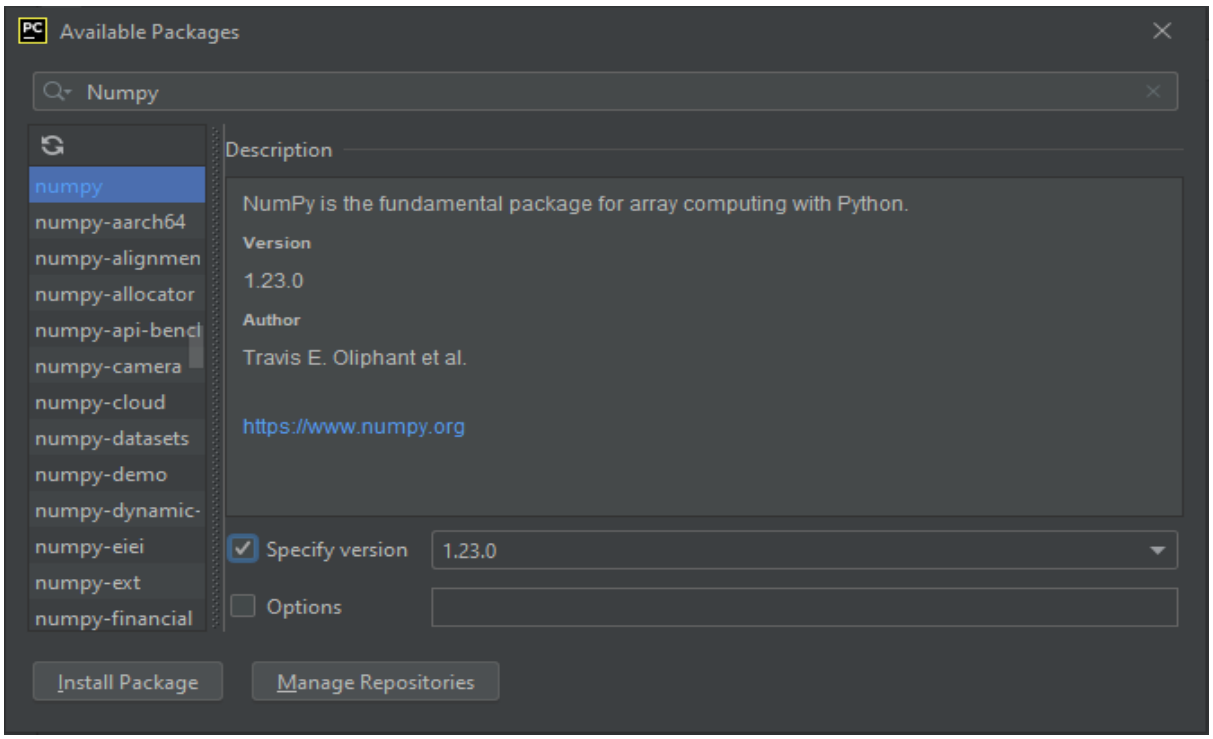

Figure:3.17 Numpy Package

3.5 WORKING

3.5.1 BASIC CODE IMPLEMENTATION

To design major project model, we must start the project at beginner level. In Basic code implementation we've created basic code model like importing libraries, setting- up the webcam and frame rates and other setup.

Output-

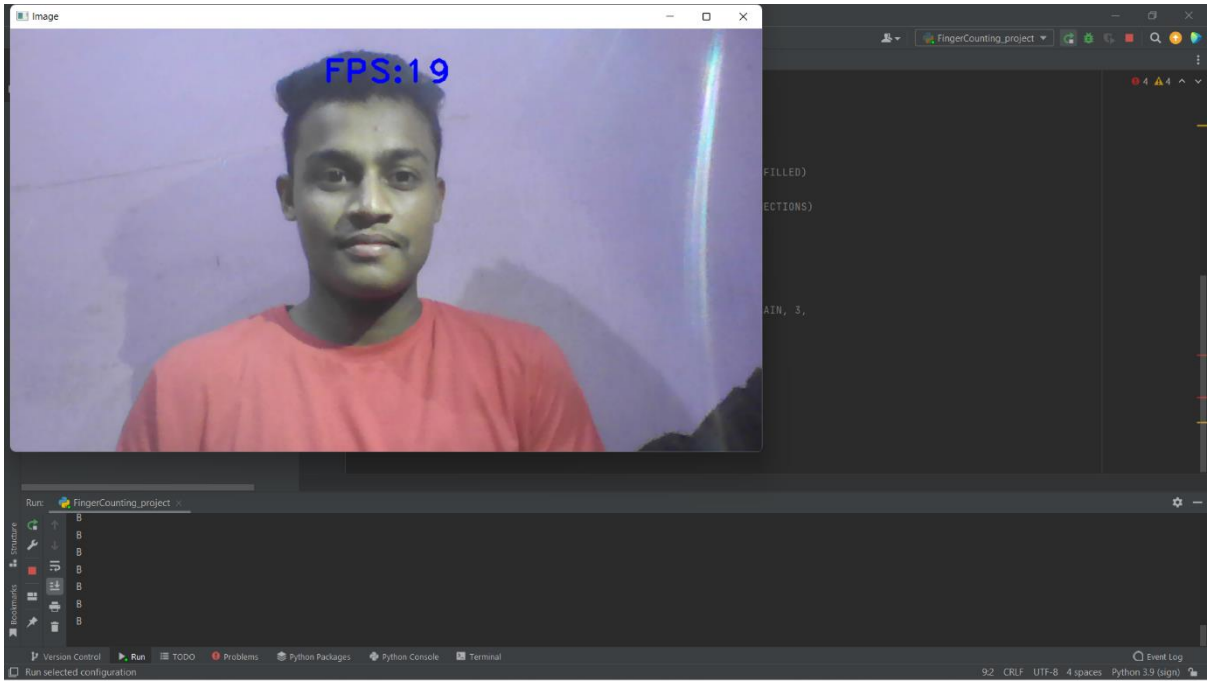

Figure:3.18 Setting the webcam

3.5.2 HAND TRACKING IMPLEMENTATION

Moving forward the next step is to detect the hand landmark points, the hand movements, according to different gestures. We've modified our python code and the code is working properly.

Output-

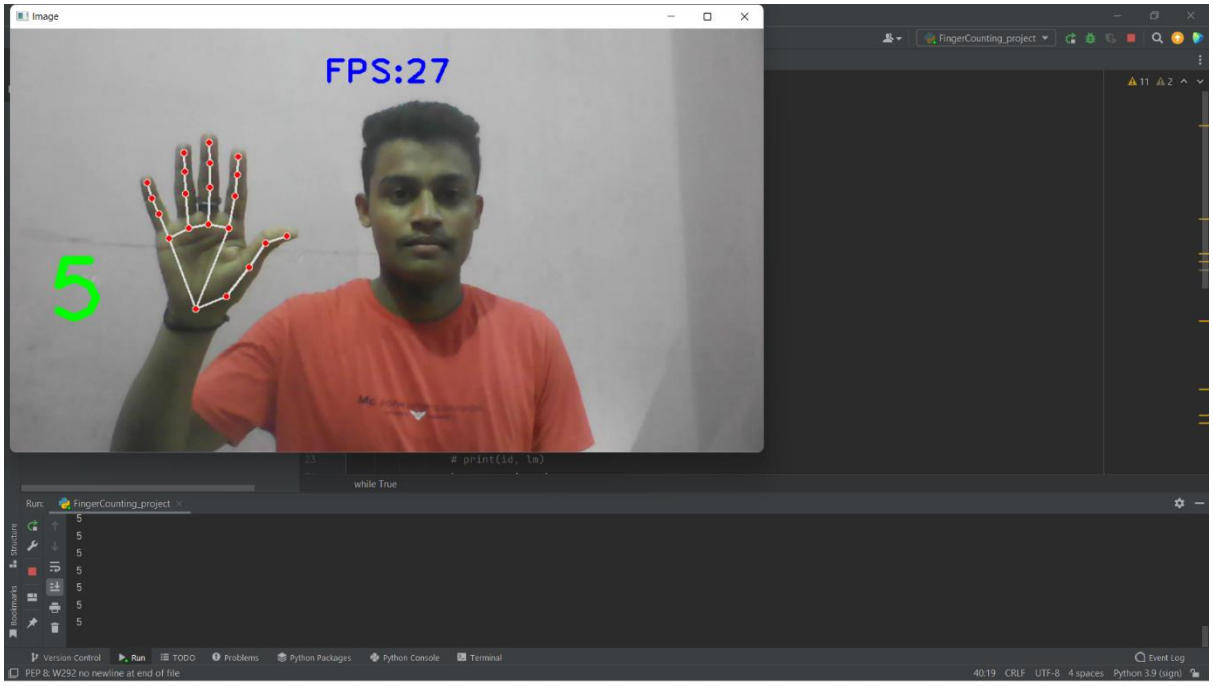

Figure:3.19 Hand tracking

3.5.3 DATA COLLECTION FOR DIFFERENT HAND GESTURES

Now the next step is to collect the datasets for each sign gestures. Every sign gesture has its own landmarks datasets, according to that we must program the model and modify the python code. The datasets can be collected by analysing each gesture at RUN window after running the python code.

Output-

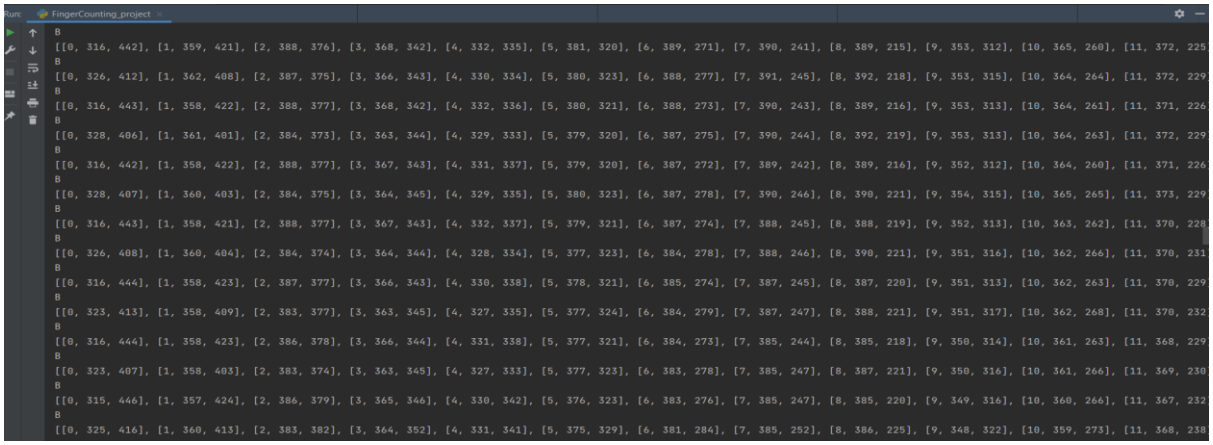

Figure:3.20 Data collection

3.5.4 AUDIO INCLUSION FOR VARIOUS HAND GESTURES

After dataset is collected, the next step is to add audio files for each sign gestures so that we can easily detect the gestures texted form into vocalize form so that the impaired(blind) people can also understand the sign languages.

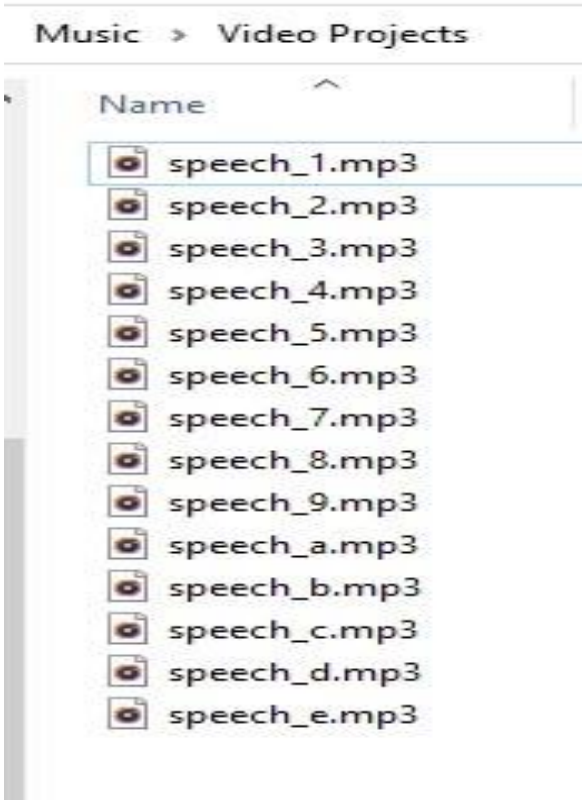

Figure:3.21 Audio Dataset

3.5.5 FINAL CODE IMPLEMENTATION (HAND GESTURES TRACKING WITH SOUND)

After audio file is included with python code, we implemented the final code by putting all modules together. We have tested this final python code and it is running successfully.

CHAPTER-4
RESULT ANALYSIS & DISCUSSION

4.1 FINAL TESTING

After completing the final module, we need to test our final code to check whether it is working correctly or not for this we are performing various type of sign gestures which are discussed below-

4.2 PERFORMING VARIOUS SIGNS

Figure: 4.1 Performing "A" sign.

Figure: 4.2 Performing "Number 2" sign.

Figure: 4.3 Performing "B" sign.

Figure: 4.4 Performing "D" sign.

Figure: 4.5 Performing "C" sign.

Figure: 4.6 Performing "Number 5" sign.

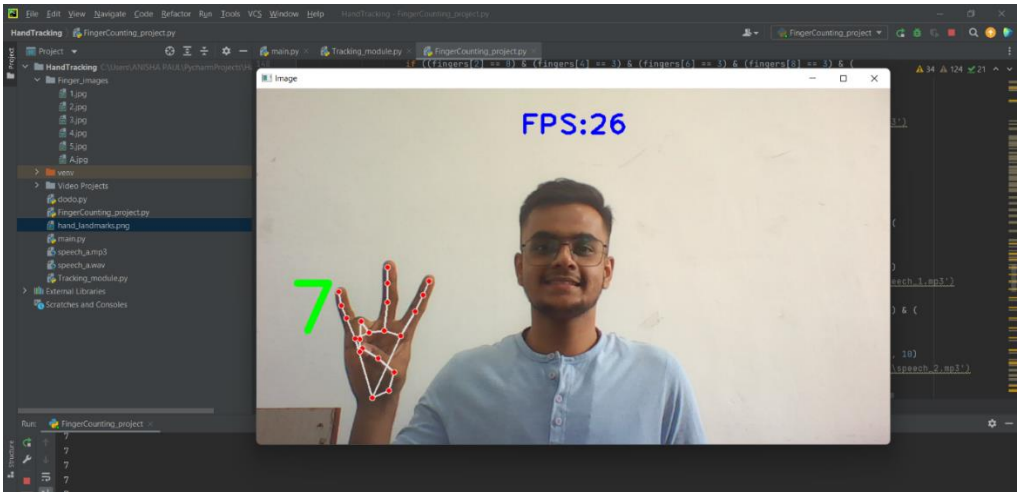

Figure: 4.7 Performing "Number 7" sign

Figure: 4.8 Performing "Number 3 " sign

CHAPTER -5
CONCLUSION & FUTURE SCOPE

5.1 APPLICATIONS

Applications of Computer Vision based sign gesture recognition-

Here we have listed down some of major domains where Computer Vision is heavily used.

- Robotics Application
- Localization − Determine robot location automatically
- Navigation
- Obstacles avoidance
- Assembly (peg-in-hole, welding, painting)
- Manipulation (e.g. PUMA robot manipulator)
- Human Robot Interaction (HRI) − Intelligent robotics to interact with and
- Serve peoples as home automation applications
- Classification and detection (e.g. lesion or cells classification and tumor
- detection)
- 2D/3D segmentation
- 3D human organ reconstruction (MRI or ultrasound)
- Vision-guided robotics surgery
- Industrial Automation Application
- Industrial inspection (defect detection)
- Assembly
- Barcode and package label reading
- Object sorting
- Document understanding (e.g. OCR)
- Security Application
- Biometrics (iris, finger print, face recognition)
- Surveillance − Detecting certain suspicious activities or behaviors
- Transportation Application
- Autonomous vehicle

- Safety, e.g., driver vigilance monitoring

5.1.1 OTHER APPLICATION

Image Zoom Gesture Recognition- one of applications of sign gesture recognition is the Zoom image by using different hand gesures. we've build this application in pycharm and tesed successfully.

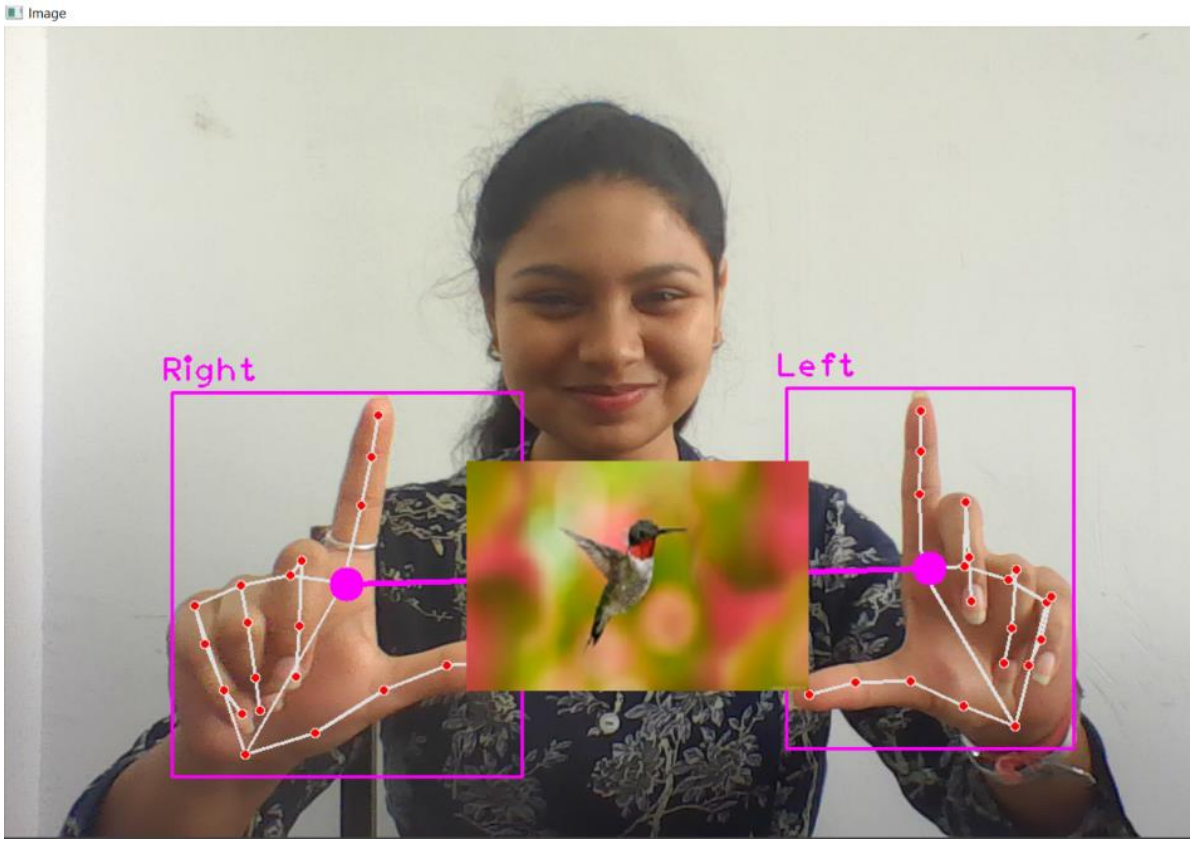

Figure: 5.1 Application on Virtual Zoom Image using Sign Gesture

5.2 LIMITATIONS

- Mobile application is difficult to execute and are difficult to be used commercially.
- Complex process more data sets are required
- Distance from webcam is limited for accuretely recognition of sign gestures.
- Sign language requires the use of hands to make gestures. This can be a problem for people who do not have full use of their hands.
- Requires Expertise in Python Coding, as this project is made up with vast number of codes.
- Requires enough knowledge in sign language.

- Less accuracy with blurry camera, as the program cannot be able to recognize the image captured.
- Misuse of copyright
- Quality reduced if it is enlarged certain size
- Processor should be faster
- Cost effective.

5.3 FUTURE SCOPE

The future scope of this project is can be vast as the development of a system for sign language is been evolving research. The model currently is limited to only the American sign language thus, it is expected to be enhanced enough to work on Indian sign language. The model requires enhancement on fields of capturing dynamic gestures as well because currently the model can only predict or interpret for static finger spellings. The datasets are also in need to be enhanced with better and accurate images. The images can be modified by adding pictures with different light density. Further training of the model to achieve efficient detection for two hand gestures. Sign language doesn't just consist of alphabets and digits, there are words also present in sign language. Android Application that can be used by anyone, anywhere to communicate with people who are deaf and dumb and understand sign language.The deployment of the model to an application would make the idea of bridging the gap for communication between normal people and the hearing impaired people.Hence the development of an efficient and easy operable app is underway. The App would contain a module to convert live captured images to text and vice versa. One can't communicate without understanding the words or just by the means of an alphabet. To overcome this the model needs continuous training to detect words and common expressions.

5.4 CONCLUSION

Hand gesture is one of the typical methods used in sign language for non-verbal communication. Sign gestures are a non-verbal visual language, different from the spoken language, but serving the same function. It is often very difficult for the hearing-impaired community to communicate their ideas and creativity to the normal humans.Our project objective is to bridge the gap by introducing an economical computer application in the communication path so that the sign language can be automatically captured, recognized and translated to text for the benefit of deaf people. The purpose of this application is to recognize hand gestures. The design is very simple, and the deaf and dumb people doesn't need to wear

any hand gloves. The system consists of camera attached to computer that will take images of hand gestures. Image segmentation & feature extraction algorithm is used to recognize the hand gestures of the impaired community. According to recognized hand gestures, corresponding data text is displayed. Although this sign language recognition application can be run on an ordinary computer having a web camera, ideally, it requires Android Smartphone having a frontal camera.The image obtained must be analyzed, processed and converted to either sign or textual display on the screen for the benefit of the hearing impaired. We have learned and demonstrated that Programso we can learn how to identify and predict the text. We have created a model that pre-process the image to the required nature for it to be fed into the model. The system is an approach to ease the difficulty in communicating with those having speech disabilities. The amount of training and validation loss observed with the proposed Sign gesture recognition architecture was less. We have tried different image processing techniques to find the best one we need for our use. During the live capture testing, the OPENCV python library with mediapipe demonstrated better results than Sensors based systems. Nowadays, applications need several kinds of images as sources of information for elucidation and analysis. Several features are to be extracted so as to perform various applications. When an image is transformed from one form to another such as digitizing, scanning, and communicating, storing, etc. degradation occurs. Therefore ,the output image has to undertake a process called image enhancement, which contains of a group of methods that seek to develop the visual presence of an image. Image enhancement is fundamentally enlightening the interpretability or awareness of information in images for human listeners and providing better input for other automatic image processing systems. Image then undergoes feature extraction using various methods to make the image more readable by the computer.Sign language recognition system is a powerful tool to preparae an expert knowledge, edge detect and the combination of inaccurate information from different sources. This can be used by anyone, anywhere, to communicate with people who are deaf and dumb and understand sign language. The deployment of the model to an application would bridge the gap for communication between normal people and hard-of-hearing people. Hence the development of an efficient and easy operable app is underway. The App would contain a module to convert live captured images to text. One can't communicate without understanding the words or just using an alphabet. To overcome this, the model needs continuous training to detect words and common expressions.

APPENDIX

Final Sign_language_Recognition_code.py-

```
import cv2
import time
import os
import Tracking_module as htm
from playsound import playsound

wCam, hCam = 640, 480
#cap1 = cv2.VideoCapyute()
cap = cv2.VideoCapture(0)
cap.set(3, wCam) #for width
cap.set(4, hCam) #for hight

folderPath = "Finger_images"
myList = os.listdir(folderPath)
print(myList)
overlayList = []

for imPath in myList:
image = cv2.imread(f'{folderPath}/{imPath}')
#print(f'{folderPath}/{imPath}')
overlayList.append(image)
print(len(overlayList))
```

```
pTime = 0
detector = htm.handDetector(detectionCon=0.5)

tipIds = [4, 8, 12, 16, 20]
while True:
success, img = cap.read()
img = detector.findHands(img)
lmList = detector.findPosition(img, draw=False)
#print(lmList)
if len(lmList) != 0:
fingers = []
#Thumb
if (lmList[tipIds[0]][1] < lmList[tipIds[0] - 1][1]):
fingers.append(1)
else:
fingers.append(0)
#index finger
#secondpoint
if (lmList[tipIds[1]][2] > lmList[tipIds[1] - 2][2]):
fingers.append(1)
else:
fingers.append(0)
#thirdpoint
if (lmList[tipIds[1]][2] > lmList[tipIds[1] - 3][2]):
fingers.append(3)
```

```
else:
fingers.append(0)

# MiddleFinger
if (lmList[tipIds[2]][2] > lmList[tipIds[2] - 2][2]):
fingers.append(1)
else:
fingers.append(0)
#thirdpoint
if (lmList[tipIds[2]][2] > lmList[tipIds[2] - 3][2]):
fingers.append(3)
else:
fingers.append(0)

# RingFinger
if (lmList[tipIds[3]][2] > lmList[tipIds[3] - 2][2]):
fingers.append(1)
else:
fingers.append(0)
#thirdpoint
if (lmList[tipIds[3]][2] > lmList[tipIds[3] - 3][2]):
fingers.append(3)
else:
fingers.append(0)
#pinkifinger
```

```
if (lmList[tipIds[4]][2] > lmList[tipIds[4] - 2][2]):
fingers.append(1)
else:
fingers.append(0)
#thirdpoint
if (lmList[tipIds[4]][2] > lmList[tipIds[4] - 3][2]):
fingers.append(3)
else:
fingers.append(0)
#thumb+middlefinger
if (lmList[tipIds[0]][1] <= lmList[tipIds[1]- 3][1]):
fingers.append(4)
else:
fingers.append(0)
#thumb+middlefinger
if (lmList[tipIds[0]][2] <= lmList[tipIds[2]-3][2]):
fingers.append(5)
else:
fingers.append(0)
#thumb+pinkifinger
if (lmList[tipIds[0]][2] <= lmList[tipIds[4]][2]):
fingers.append(6)
else:
fingers.append(0)
#thumb+ringfinger
```

```
if (lmList[tipIds[0]][2] <= lmList[tipIds[3]][2]):
fingers.append(7)
else:
fingers.append(0)
#thumb+middlefinger
if (lmList[tipIds[0]][2] <= lmList[tipIds[2]][2]):
fingers.append(8)
else:
fingers.append(0)
#thumb+indexfinger
if (lmList[tipIds[0]][2] <= lmList[tipIds[1]][2]):
fingers.append(9)
else:
fingers.append(0)
#4 fingers
#for id in range(1,5):
#    if lmList[tipIds[id]][2] < lmList[tipIds[id]-2][2]:
#        fingers.append(1)
#    else:
#        fingers.append(0)

#print(fingers)
if ((fingers[2]==3) & (fingers[4]==3)& (fingers[6]==3)& (fingers[8]==3)& (fingers[0]==1)):
print('A')
cv2.putText(img, str('A'), (45,375), cv2.FONT_HERSHEY_PLAIN, 8, (0, 255, 0), 10)
```

```
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_a.mp3')


#cv2.waitKey(3)
else:
if ((fingers[1]==0) & (fingers[3]==0)& (fingers[5]==0)& (fingers[7]==0)& (fingers[0]==1)):
print('B')
cv2.putText(img, str('B'), (45,375), cv2.FONT_HERSHEY_PLAIN, 8, (0, 255, 0), 10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_b.mp3')
else:
if ((fingers[1] == 1) & (fingers[3] == 1) & (fingers[5] == 1) & (fingers[7] == 1) & (fingers[0]
== 0) & (fingers[9]==0)):
print('C')
cv2.putText(img, str('C'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8, (0, 255, 0), 10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_c.mp3')
else:
if ((fingers[2] == 0) & (fingers[4] == 3) & (fingers[6] == 3) & (fingers[8] == 3) & (
fingers[9] == 4) ):
print('D')
cv2.putText(img, str('D'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8, (0, 255, 0), 10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_d.mp3')
else:
if ((fingers[1] == 1) & (fingers[4] == 3) & (fingers[6] == 3) & (fingers[8] == 3) & (
fingers[9] == 4)):
print('E')
cv2.putText(img, str('E'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8, (0, 255, 0), 10)
```

```
else:
if ((fingers[2] == 0) & (fingers[4] == 3) & (fingers[6] == 3) & (fingers[8] == 3) & (
fingers[0] == 1)):
print('1')
cv2.putText(img, str('1'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8, (0, 255, 0), 10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_1.mp3')
else:
if ((fingers[2] == 0) & (fingers[4] == 0) & (fingers[6] == 3) & (fingers[8] == 3) & (
fingers[0] == 1)):
print('2')
cv2.putText(img, str('2'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8, (0, 255, 0), 10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_2.mp3')
else:
if ((fingers[2] == 0) & (fingers[4] == 0) & (fingers[6] == 0) & (
fingers[8] == 3) & (
fingers[0] == 1)):
print('3')
cv2.putText(img, str('3'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8, (0, 255, 0),
10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_3.mp3')
else:
if ((fingers[2] == 0) & (fingers[4] == 0) & (fingers[6] == 0) & (
fingers[8] == 0) & (
fingers[9] == 4)):
print('4')
```

```
cv2.putText(img, str('4'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8,
(0, 255, 0), 10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_4.mp3')
else:
if ((fingers[2] == 0) & (fingers[4] == 0) & (fingers[6] == 0) & (
fingers[8] == 0) & (
fingers[0] == 0)):
print('5')
cv2.putText(img, str('5'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8,
(0, 255, 0), 10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_5.mp3')
else:
if ((fingers[2] == 0) & (fingers[4] == 0) & (fingers[6] == 0) & (
fingers[8] == 3) & (
fingers[11] == 6)):
print('6')
cv2.putText(img, str('6'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8,
(0, 255, 0), 10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_6.mp3')
else:
if ((fingers[2] == 0) & (fingers[4] == 0) & (fingers[6] == 3) & (
fingers[8] == 0) & (
fingers[12] == 7)):
print('7')
cv2.putText(img, str('7'), (45, 375), cv2.FONT_HERSHEY_PLAIN, 8,
```

```
(0, 255, 0), 10)
playsound(r'C:\Users\Vikash\Music\Video Projects\speech_7.mp3')
else:
if ((fingers[2] == 0) & (fingers[4] == 3) & (
fingers[6] == 0) & (
fingers[8] == 0) & (
fingers[13] == 8)):
print('8')
cv2.putText(img, str('8'), (45, 375),
cv2.FONT_HERSHEY_PLAIN, 8,
(0, 255, 0), 10)
playsound(
r'C:\Users\Vikash\Music\Video Projects\speech_8.mp3')
else:
if ((fingers[2] == 3) & (fingers[4] == 0) & (
fingers[6] == 0) & (
fingers[8] == 0) & (
fingers[14] ==9)):
print('9')
cv2.putText(img, str('9'), (45, 375),
cv2.FONT_HERSHEY_PLAIN, 8,
(0, 255, 0), 10)
playsound(
r'C:\Users\Vikash\Music\Video Projects\speech_9.mp3')
else:
```

```
print('RELAXED')

#totalFingers = fingers.count(1)

#print(totalFingers)

#h, w, c = overlayList[totalFingers-1].shape

#img[0:h, 0:w] = overlayList[totalFingers-1]


#cv2.rectangle(img, (20, 255), (170, 425), (0,255,0), cv2.FILLED)

#cv2.putText(img, str(totalFingers), (45,375), cv2.FONT_HERSHEY_PLAIN, 10, (255, 0, 0),
10)

cTime = time.time()

fps = 1/(cTime-pTime)

pTime=cTime

cv2.putText(img, f'FPS:{int(fps)}', (400,70), cv2.FONT_HERSHEY_PLAIN, 3, (255, 0, 0),
3)

cv2.imshow("Image", img)

cv2.waitKey(1)
```

REFERENCES

[1] Deafness (who.int)

[2] projects1920C6.pdf

[3] SIGN TO SPEECH IEEE.itkarkar2013.pdf

[4] Bauman, Dirksen (2008). *Open your eyes: Deaf studies talking*. University of Minnesota Press. ISBN 978-0-8166-4619-7.

[5] Nielsen, Kim (2012). *A Disability History of the United States*. Boston, Massachusetts: Beacon Press. ISBN 978-080702204-7.

[6] BONVILLIAN, JOHN D.; INGRAM, VICKY L.; McCLEARY, BRENDAN M. (2009). "Observations on the Use of Manual Signs and Gestures in the Communicative Interactions between Native Americans and Spanish Explorers of North America: The Accounts of Bernal Díaz del Castillo and Álvar Núñez Cabeza de Vaca". *Sign Language Studies*. **9** (2): 132–165. ISSN 0302-1475. JSTOR 26190668.

[7] Groce, Nora Ellen (1985). *Everyone Here Spoke Sign Language: Hereditary Deafness on Martha's Vineyard*. Harvard University Press. ISBN 9780674270404.

[8] "Deafness on Martha's Vineyard | Britannica"

[9] *Sign Languages of the World : A Comparative Handbook*, edited by Julie Bakken Jepsen, et al., De Gruyter, Inc., 2015. *ProQuest Ebook Central*, https://ebookcentral.proquest.com/lib/bu/detail.action?docID=4006782.

[10] History of sign language - Wikipedia

[11] History of Sign Language - Deaf History | Start ASL

[12] Home | Government of India (censusindia.gov.in)

[13] sign language - Bing images

[14] What is Python? - PythonForBeginners.com

[15] Libraries in Python - GeeksforGeeks

[16] tensor flow in python - Bing images

[17] opencv - Bing images

[18] Google Image Result

[19] Google Image Result

[20] Google Image Result

[21] image processing - Bing images

[22] Google Image Result