

SIGN VOICE BIDIRECTIONAL COMMUNICATION AID FOR HEARING AND VOCALLY IMPAIRED PEOPLE

Submitted in partial fulfillment of the requirements for the award of Bachelor
of Engineering Degree in Computer Science and Engineering

By

NAVEEN RAJ.R (Reg No. 38290056)

PRAVEEN.C (Reg No. 38290067)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING



SIGN VOICE BIDIRECTIONAL COMMUNICATION AID FOR HEARING AND VOCALLY IMPAIRED PEOPLE



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with Grade “A” by NAAC
(Established under section 3 of UGC Act, 1956)
JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI-600 119.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **NAVEEN RAJ.R (Reg No.38290056)** and **PRAVEEN.C (Reg No.38290067)** who carried out the project entitled **“SIGN VOICE BIDIRECTIONAL COMMUNICATION AID FOR HEARING AND VOCALLY IMPAIRED PEOPLE”** under our supervision from November 2020 to March 2021.

Internal Guide

Dr. J. ALBERT MAYAN ME., Ph.D.,

Head of the Department

Submitted for Viva Voice Examination held on

Internal Examiner

External Examiner

DECLARATION

Naveen raj.R (Reg No. 38290056) and Praveen.C (Reg No. 38290067) hereby declare that the Project Report entitled **“SIGN VOICE BIDIRECTIONAL COMMUNICATION AID FOR HEARING AND VOCALLY IMPAIRED PEOPLE”** done by us under the guidance of **Dr. J. ALBERT MAYAN ME., Ph.D.**, is submitted in partial of the fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE :

PLACE : CHENNAI

Signature of the Candid

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing and **Dr.S.VIGNESHWARI,M.E.,Ph.D.**, and **Dr.L. LAKSHMANAN, M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. J. ALBERT MAYAN M.E.,Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways For the Completion of the project.

ABSTRACT

Sign language interpretation is an importance task to facilitate communication among hearing and vocally impaired people and normal people. Hearing and Vocally impaired face problems in communicating with others. Hence, the goal of this paper is to design a desktop human computer interface application that is used to facilitate communication between normal, hearing and vocally impaired people. The system helps blind person to hear voice saying the word gestured by the hearing and vocally impaired people while the hearing and vocally impaired will receive a gesture representing the word said by the normal people. The system works in two directions, the first direction is processing from video to speech. The animated word gestures are mapped with language knowledge base into text. Then, the relevant audio is generated using Text-to-Speech (TTS) API. The KNN algorithm is being used for the classification and GLCM method for feature extraction. The second direction is processing from speech to video. The voice from normal people is converted into its corresponding text using Speech-to-Text (STT) API. Then, the text is mapped to relevant sign language form and displayed to hearing and vocally impaired people. The existing Glove based system is restricted by limitations such as high cost flex sensor, long time to adapt and it requires the user to be connected to the system .The proposed system does not

require any high technology and the implementation is low and the system does not require wearable devices.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
NO		NO
	ABSTRACT	iii
	TABLE OF CONTENTS	iv
	TABLE OF FIGURE	vii
1	INTRODUCTION	
	1.1 General	1
	1.2 Sign language	1
	1.3 Interpretation of the sign language and related work	3
	1.4 Objective	4
	1.5 Scope of the project	4
	1.6 Overview of the project	4
	1.7 Overview of the methodology	5
	1.7.1 Python	5
	1.7.2 Machine learning algorithm	5
	1.8 Advantage of the system	6
	1.9 Summary	6

2	LITERATURE SURVEY	
	2.1 General	7
	2.2 Literature review	7
	2.3 Literature survey on sign language interpretation	8
	2.4 Summary	13
3	SYSTEM ANALYSIS	
	3.1 General	14
	3.2 Existing system	14
	3.3 Proposed system	14
	3.4 System architecture	15
	3.4.1 Sign to voice	15
	3.4.2 Voice to sign	16
	3.5 Software requirements	16
	3.5.1 Spyder	16
	3.5.2 Package installed-anaconda	20
	3.6 Summary	21
4	SYSTEM DESIGN	
	4.1 General	22
	4.2 Module description	22
	4.2.1 Image to speech conversion	22
	4.2.2 Speech to image conversion	28
	4.2.2.1 Google API	29
	4.2.2.2 Speech recognition	30
	4.3 UML diagram	32

4.3.1	Use case diagram	33
4.3.2	Sequence diagram	34
4.3.3	Collaboration diagram	36
4.3.4	Activity diagram	37
4.4	Summary	39
5	IMPLEMETATION AND RESULTS	
5.1	General	40
5.2	Implementation	40
5.2.1	Sign to voice	40
5.2.2	Voice to sign	42
5.3	Results	43
5.4	Summary	43
6	CONCULSION AND FUTURE ENHANCEMENT	
6.1	Conclusion	44
6.2	Future enhancement	44
6.3	summary	
44		
	REFERENCES	45

LIST OF FIGURE

S.NO	TITLE	PAGE NO
1.1	Alphabets in Indian sign language	2
3.1	Sign to voice system architecture	15
3.2	Voice to sign system architecture	16
4.1	Dataset	22
4.2	Sign to voice use case diagram	33
4.3	Voice to sign use case diagram	34
4.4	Sign to voice sequence diagram	35
4.5	Voice to sign sequence diagram	36
4.6	Sign to voice collaboration diagram	37
4.7	Voice to sign collaboration diagram	37
4.8	Sign to voice activity diagram	38
4.9	Voice to sign activity diagram	39
5.1	Process of capturing image	40
5.2	Feature extraction process	41
5.3	Main program	41

5.4	Module 1 output	42
5.5	Recording audio and requesting google API	42
5.6	Module 2 output	43

CHAPTER 1 INTRODUCTION

1.1 GENERAL

According to the World Health Organization (WHO), there are around 466 million people worldwide have disabling hearing loss and 285 million vocally impaired. Hearing and Vocally impaired people face problems in communicating with others with difficulties in dealing with the communication technology. The goal of this project is to design a desktop human computer interface application that is used to facilitate communication between normal, hearing and vocally impaired people. The system helps Normal person to hear voice saying the word gestured by the “hearing and vocally impaired people” while the hearing impaired people will receive a gesture representing the word said by the Normal. It is important to give the people with hearing loss an individualized and appropriate communication system that supports different communication techniques, strategies and modes. These systems should reflect their assessed needs and respects their choice with facilitating their life by integrating them into the society.

1.2 SIGN LANGUAGE

A Sign language(also signed language) is a language which chiefly uses manual communication to convey meaning, as opposed to acoustically conveyed sound patterns this can involve simultaneously combining hand shapes, orientation and movement of the hand ,arms or body, and facial expressions to express a speaker's thoughts. Sign language many similarities with spoken languages (sometimes called "oral languages").which depend primarily on sound and linguists consider both to be type of natural language although there are some significant differences between signed and spoken languages, such as how they use space grammatically, sign languages show the same linguistic properties and use the same language faculty as do spoken languages. They should not be confused with body language, which is a kind of non-linguistic communication. Wherever communities of hearing impaired people exist, sign language have developed, and or at the cores of local hearing impaired people cultures. Although signing is used primarily by the hearing loss people, it is also used by others such as people who can hear but cannot physically speak or have trouble with spoken language due to some other disability (augmentative and alternative communication). It is not clearly how many sign language there are. A common misconception is that all sign languages are the same worldwide or that sign language is internationally. Indian sign language interpretation is an important task to facilitate communication among Indian hearing impaired people community and other people. Even now, hearing loss people are isolated from the society because of the communication gap with other people.

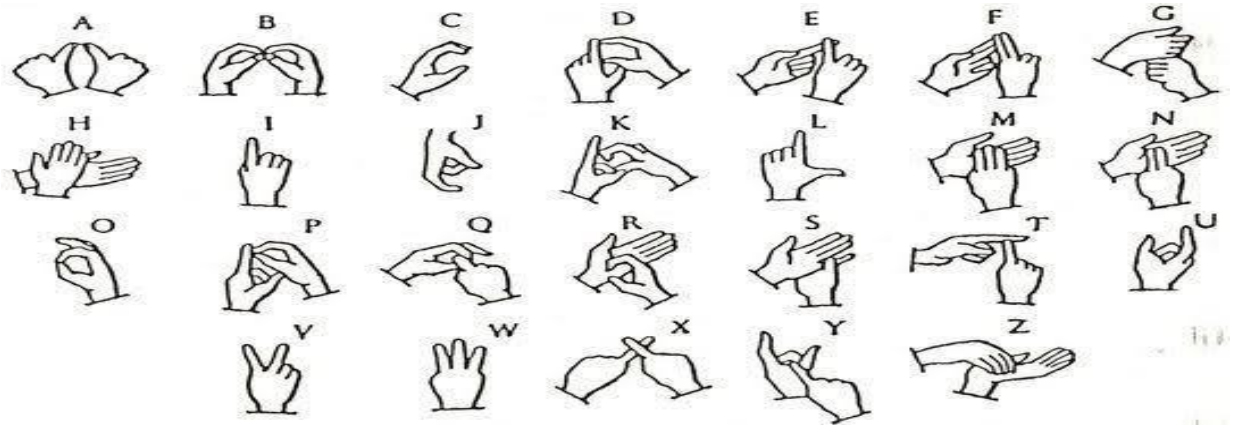


Figure 1.1 Alphabets in Indian Sign language

1.3 INTERPRETATION OF THE SIGN LANGUAGE AND RELATED WORK

There are two ways to bring hearing loss people into mainstream of society

1) Human interpreter 2) Computer interpreter. As compared to number of hearing loss people, human interpreters are much less in number and moreover they may not be available every time. Sign language recognition involves field of computer vision, pattern recognition as well as linguistic study. It is a very challenging problem for the researchers due to the multi-modality gesture interface and limitation of vision based technology. Sign language considers different modality gesture such as lip movements, facial expression, eyebrow movements and hand gestures.

Several algorithms are developed for the recognition of the hand gestures such as Dynamic Time wrapping which gave better accuracy for continuous word recognition, Inverted Indexing for convolutional continuous sentence recognition. Hardware devices such as a pair of data-gloves are used as the sensing devices for detecting motions of hands and fingers. For efficient recognition of gestures and postures, a technique of

efficient classification of motions is proposed and a fuzzy min-max neural networks is adopted for on-line pattern recognition.

A Novel feature vector is proposed which is suitable for recognizing dynamic hand gestures with a Leap Motion Controller. The feature vector with depth information is computed and fed into the Hidden Conditional Neural Field (HCNF) classifier to recognize dynamic hand gestures.

The static hand gestures were recognized by using hand contour as the hand gesture feature and SVM is used to classify the feature. A Multi-Scale weighted Histogram of Contour direction (MSWHCD) based direction normalization is proposed to ensure a good recognition performance.

1.4 OBJECTIVE

The system aims to develop a system for translating sign language (sequence of images) into speech as one direction and translating voice spoken words to sign language as a second direction. Machine Learning algorithms are used for recognition of gestures and voice and hence humans can communicate with the systems without any hardware devices, which makes the system easier to use.

1.5 SCOPE OF THE PROJECT

Hearing and vocally impaired people find a difficulty in communicating with normal people and therefore they cannot be fully part and effective personnel in their Societies. Hence it helps for fast and better communication between the normal people and hearing and vocally impaired people keeps a better understanding between them. In an

emergency situation and in need of any usable things, the vocally impaired person cannot communicate, for this purposes this system has been proposed. These systems should reflect their assessed needs and respects their choice with facilitating their life by integrating them into the society.

1.6 OVERVIEW OF THE PROJECT

The sign images are captured using the webcam and feature extraction is done to convert the raw pixel values into useful information then machine learning algorithm is used to recognize the captured sign language .In other direction, voice is taken as input from microphone and feature extraction is done on the recorded voice then recognition of the voice is done using the machine learning algorithm.

1.7 OVERVIEW OF THE METHODOLOGY

1.7.1 PYTHON

Python programming language is used for the implementation of the machine learning and feature extraction algorithms. Python is a general purpose and high level programming language, allows you to focus on core functionality of the application. Python is interpreted and is object oriented. Python is simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

1.7.2 MACHINE LEARNING ALGORITHM

Machine Learning (ML) is the scientific study of algorithms and standard models that computer systems use to perform specific task without using explicit instructions, relying on patterns and inference instead. Machine learning algorithms build a mathematical model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to perform the task. Machine Learning is used in project, since it is difficult to conventional algorithm to perform the task effectively.

K-NN (k-nearest neighbour) Algorithm is used for the pattern recognition and classification. The input consist of the k closest training examples in the feature space and the output is the class membership. The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consist of only of storing the feature vectors and class labels of the training samples. In the classification phase, k is a user-defined constant, and an unlabelled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point.

1.8 ADVANTAGES OF THE SYSTEM

The proposed vision-based hand tracking system does not require any special markers or gloves. It can operate with low-cost cameras. The proposed system puts no restrictions on the environment illumination or background simplicity.

1.9 SUMMARY

This chapter covers the framework for hand gesture recognition and voice recognition using the machine learning algorithms and feature extraction using the feature extraction algorithms.

CHAPTER 2

LITERATURE SURVEY

2.1 GENERAL

The chapter gives the overall description of the reference paper techniques and methodology used which we can identify the problems of existing methodology. The study helps us to identify the solutions to the problems prevailing in the existing methodology and to increase the overall performance of the system.

2.2 LITERATURE REVIEW

A literature review is a text of a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources, and do not report new or original experiment work. Most often associated with academic oriented literature, such reviews are founded in academic journals, and or not be confused with book reviews that may also appear in the same publication. Literature reviews are a basis for research in nearly every academic field. The study helps us to identify the solutions to the problems prevailing in the existing methodology and to increase the overall performance of the system. A narrow-scope literature review may be included as part of a peer-reviewed journal article presenting new research, serving to situate the current study within the body of the relevant literature and to provide content for the reader. In such a case, the review usually precedes the methodology and result sections of the work.

2.4 SUMMARY

This chapter explains some of the researches undergone which are used as reference for the development of this project. The drawback of existing technique are identified and listed.

CHAPTER-3

SYSTEM ANALYSIS

3.1 GENERAL

Identifying an existing system is essential to know about the drawback in order to rectify them. A new system is then proposed which is efficient when compare to existing system.

3.2 EXISTING SYSTEM

The existing work is the Glove based system, a hardware system working from signs to text direction. In this system speaker wear electronic hardware gloves to measure hands and face motion level up to six degrees of freedom using the flex sensor. Those systems are very efficient on its measurements and can exactly describe each moving object. The limitation of the existing work is high equipment cost and the user needs to be in a static place.

3.3 PROPOSED SYSTEM

The proposed system is a software system working in both directions (sign to voice & voice to sign). In this system the gesture of the hand is captured using webcam and the gesture is classified using a classification algorithm. In the other direction, the voice is given as which is converted to text and then the sign images are displayed for each alphabet in the text. The implementation of the system does not require any High-technology and the implementation cost is low. The system does not require any wearable devices.

3.4 SYSTEM ARCHITECTURE

The system mainly consists of two modules, first module is the translation of the sign language to text and then to speech as one direction. The second module is translation of the voice to text using google API.

3.4.1 SIGN TO VOICE

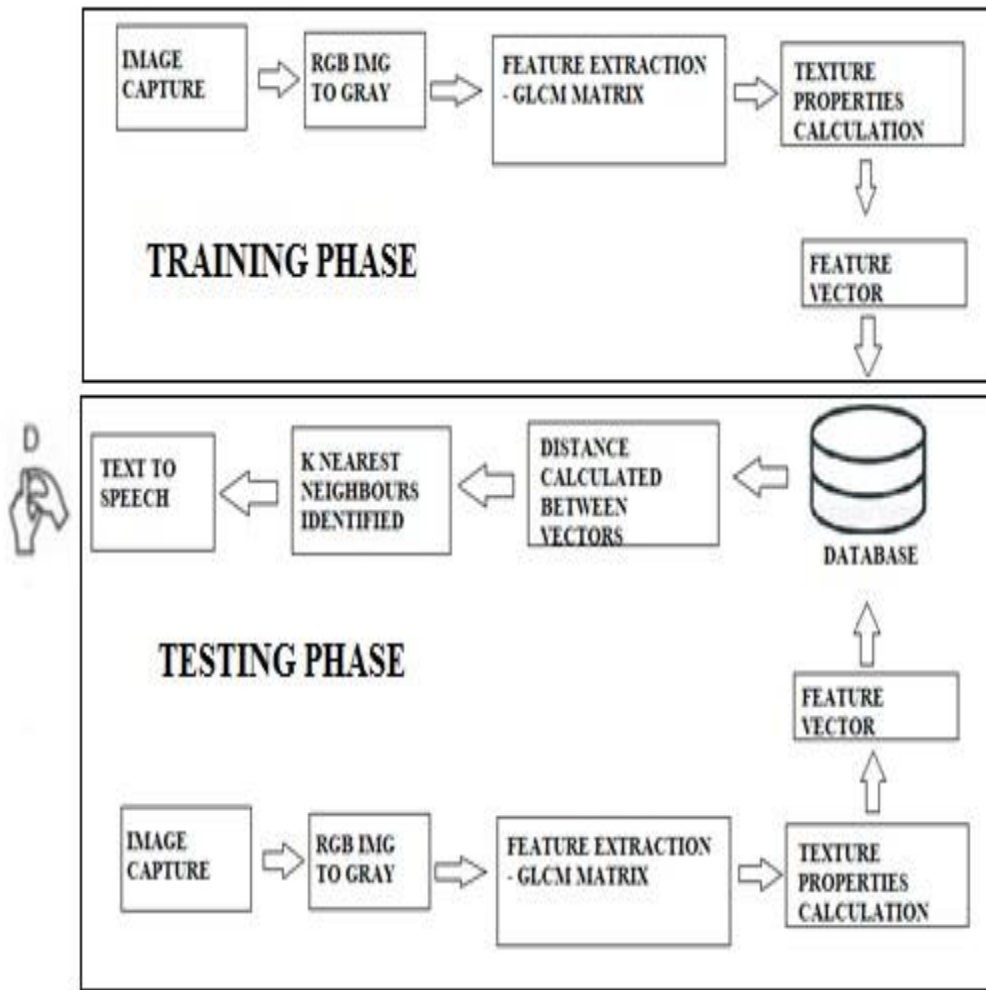


Figure 3.1 Sign to Voice System Architecture

Thus the figure 3.1 shows the architecture diagram of first module (sign to voice) of the project. It consist of two phases, one is training phase and the other is testing phase. Training phase deals with obtaining inbuilt input images from the system and it continues with pre-processing work and feature extraction and stores the feature vector formed in the repository. The same process is followed in testing phase with only difference i.e. it

deals with recognizing the live image which is given through the web camera.

3.4.2 VOICE TO SIGN

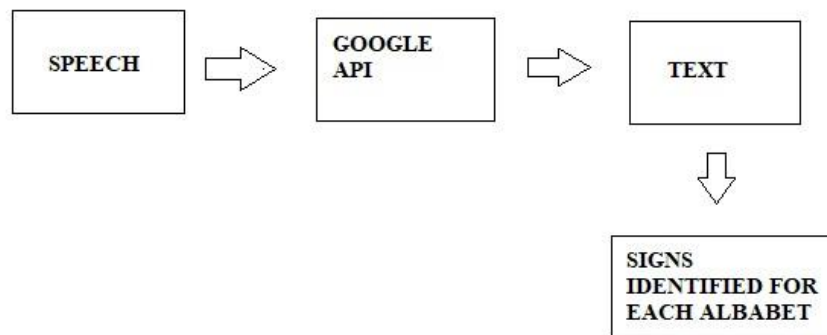


Figure 3.2 Voice to Sign System Architecture

Thus the figure 3.2 shows the architecture of the second module (voice to sign) of the project. It consist of only one phase i.e only testing voice .It deals with obtaining the input voice and converting into text and signs are displayed for each alphabet in the text.

3.5 SOFTWARE REQUIREMENTS

3.5.1 SPYDER

Spyder (Scientific Python Development Environment, previously known as Pydee) is an open source cross-platform integrated development environment (IDE) for scientific programming in the python language .It a

simple and lightweight, yet powerful interactive development environment. It can be installed on Windows using Python or Anaconda.

It integrates with number of prominent packages in the scientific python stack, including Numpy, Scipy, Matplotlib, pandas, Ipython, SymPy and Cython as well as other open source software. Spyder abilities can be extended even further via its plugin system and API.

FEATURES

➤ EDITOR

- Customizable syntax highlighting themes
- Debugging breakpoints and conditional breakpoints(through ipdb integration)
- Interactive execution: Run line/selection, run cell, run file, re-run and debug
- Run configuration settings:
- Working directory selection
- Command line options
- Outline Explorer: Navigate cells, functions, classes, blocks, and more □ Real-time code introspection features:
- On-demand (Tab) and “dot” automatic code completion
 - . Automatic popup call tips showing function signatures

➤ IPYTHON CONSOLE

Any number of individual consoles, each executed in a separate, isolated processes

- Each console uses the full IPython kernel as a backend with a light GUI frontend
- Supports all of the powerful IPython magic commands and functionality
- Automatic code completion and calltips, and automatic link to Help
- Range of code run options and interactivity
- Inline display of Matplotlib graphics (optional)
- The User Module Reloaded, automatically re-importing modified source files

➤ **VARIABLE EXPLORER**

- Can list all global variables, functions, classes, and other objects, or filter them by several criteria
- GUI-based editors for numerous data types (numeric, strings, collections, NumPy arrays, Pandas Data Frames, dates/times, images, and more)
- Import/export data or an entire session from/to many formats (text, csv, NumPy files, MATLAB files)
- Interactive data visualization options (plot, histogram, image...) using Matplotlib

➤ **HELP**

- Provides documentation or source code for any Python object (class, function, module...)

- Can be triggered manually, on demand (Ctrl-I by default) or automatically on typing a left parenthesis after a function name
(optional)
- **STATIC CODE ANALYSIS:** Detects an array of style issues, bad practices, potential bugs, and other problems with your code
- **PROFILER:** Measures the performance impact of every function in a file to identify bottlenecks and aid optimization
- **PROJECTS:** Allows for easy saving and restoring of settings, sessions and setup for working on multiple development efforts simultaneously
- **FILE EXPLORER:** Integrated filesystem viewing supporting many common operations
- **FIND IN FILES:** Find string occurrences in a file, directory, or entire project with full support for powerful regular expressions and excluded locations
- **ONLINE HELP:** Search and browse rich HTML documentation on installed Python modules, packages, functions, classes, builtins and more, including your own
- **HISTORY LOG:** Chronologically records every command entered into any Spyder console with timestamps, syntax highlighting and de-duplication
- **INTERNAL CONSOLE:** Provides access to viewing, exploring and controlling Spyder's own operation
- **GENERAL FEATURES**
 - User environment variables viewer/editor (Windows-only)

- Handy links to useful resources and documentation (Python, Matplotlib, NumPy, Scipy, Qt, IPython, etc.)
- Interactive tour, tutorial and shortcut cheat sheet for new users

3.5.2 PACKAGE INSTALLED – ANACONDA

Anaconda is a free and open-source distribution of the python and R programming language for machine learning applications. It's great for having short and simple setup. It has more data science packages and simplifies package management and deployment. It allow to build and train machine learning model with scikit-learning package. It is flexible for image processing through the skimage package. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. The Spyder application is available by default in Navigator.

Scikit-image is an open-source image processing library for the Python programming language. It includes algorithms for segmentation, geometric transformations, color space manipulation, analysis, filtering, morphology, feature detection, and more. It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests and DBSCAN.

3.6 SUMMARY

The proposed system along with the existing system is discussed. The basic requirement of software that are used to implement the project is also discussed in this section

CHAPTER-4

SYSTEM DESIGN

4.1 GENERAL

Based on the detailed analysis of the existing system, the new system must be designed. This is the phase of the system designing. It is the most crucial phase in development of a system.

4.2 MODULE DESCRIPTION

4.2.1 IMAGE TO SPEECH CONVERSION

It consists of training phase and test phase. Training Phase deals with obtaining input image and continues with pre-processing work and feature extraction. Test phase deals with recognizing the live image.

A. DATASET ACQUISITION

The images of the sign languages are collected and the data set is formed. For each alphabet 1000 images are captured.

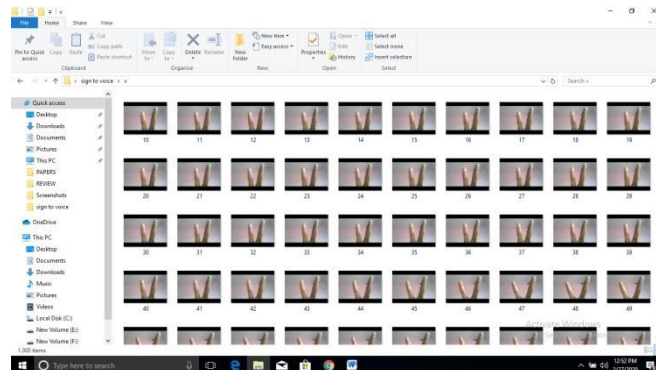


Figure 4.1 DATASET

B. DATA PREPROCESSING

Data pre-processing is a data mining technique that involves transforming raw data into an understandable form.

- The color scale of the collected image are transformed from RGB to **HSV color scale**. HSV color space is the most suitable color space for color based image segmentation. HSV color space consist of 3 matrices, 'hue', 'saturation', and 'value'. 'Hue' represents the color, 'saturation' represents the amount to which that respective color is mixed with white and 'value' represents the amount to which that respective color is mixed with black. HSV color spaces separate color from intensity.
- The skin detection take place using the cv2.inRange function, the HSV frame, the lower and upper boundaries are passed as arguments. The lower and upper boundaries are the pixel intensities to be considered as skin in the HSV frame. The output of the function is the mask, the mask is a single channel image. Pixels that are white (255) in the mask represent areas of the frame that are skin, pixels that are black (0) in the mask represent areas that are not skin.
- However, we may detect many small false-positive skin regions in the image. To remove these small regions, the structuring kernel will be created. The kernel (a matrix of size 3x3 pixels) work as a slider. When the slider slide around, and if all of the pixels are white, then we get white, otherwise black, the technique is termed as Erosion. The other version of this is Dilation, which basically does the opposite, if the entire area isn't black, then it is converted to white.

- Then, the mask is smoothened using a Gaussian blur. This smoothing step produces much cleaner mask. It is widely used to reduce image noise and enhance the image structure. The Gaussian Kernel size and the standard deviation is given as the input to the function. Gaussian function is a mathematical function that calculates the transformation to be applied to each pixel in the image.
- The skin mask is applied to the frame using the Bitwise AND operation. Bitwise operations helps in image masking and enhancing the properties of the input images.
- The frame is converted to GRAY scale, since GRAY scale reduces complexity and makes the data efficient for computations and makes the images suitable for any applications. For a grayscale images, the pixel value is a single number that represents the brightness of the pixel. The possible values ranges from 0 to 255.

C. FEATURE EXTRACTION

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.

- GLCM [Gray Level Co-Occurrences Matrix] is used to extract the features of the image. The basic idea of the GLCM is it considers of adjacency of the pixel values. The GLCM function characterize the texture of an image by calculating how often pairs of pixel with specific values and in a specific spatial relationship occur in an image, creating a GLCM, and then extracting the statistical measures from this matrix. Greycomatrix method is used to

calculate the grey-level co-occurrences matrix. The call to the greycomatix is of the form,

greycomatrix (Image,Distance, Angle,Level,Symmetry,Normed)

Image: integer type input image. The image will be cast to uint8, so the maximum value must be less than 256.

Distances: List of pixel pair distance offsets

Angles: List of pixel pair angles in distance

Levels: The input image should contain integers in [0, levels-1], where levels indicate the number of grey-levels counted.

Symmetric: If this is accomplished then the entry (i,j) and (j,i) will be same. The default is False.

Normed: If true, normalize each matrix by dividing by the total number of accumulated co-occurrences for the given offset. The elements of the resulting matrix sum to 1.

- The statistical properties such as Contrast, Dissimilarity, Correlation, Homogeneity, ASM, energy are calculated and the feature vector is formed.
 - Contrast property returns a measure of the intensity contrast between a pixel and its neighbour over the whole image.
 - Dissimilarity is similar to contrast, instead of weights increasing exponentially(0,1,4,9..etc) as one moves away from the diagonal as contrast did, the dissimilarity weights increase linearly(0,1,2,3,..).
 - Correlation returns a measure of how correlated a pixel is to its neighbour over the whole image. The range is [-1,1]. Correlation is 1 or -1 for a perfectly positively or negatively correlated image. Correlation is NaN for a constant image.

- Energy returns the sum of squared elements in the GLCM. The range is [0 1]. Energy is 1 for a constant image.
 - Homogeneity returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. The range is [0 1]. Homogeneity is 1 for a diagonal GLCM.
- Finally, the feature vector is constructed, which is of the form [Contrast, Dissimilarity, Correlation, Homogeneity, ASM, Energy, bm1, gm1, rm1]

Feature vectors are used to represent numeric characteristics called features, of an object in a mathematical, easily analyzable way. Feature vectors are used widely in machine learning because of the effectiveness and practicality of representing objects in a numerical way to help with many kinds of analyses. They are good for analysis because there are many techniques for comparing feature vectors.

D. CLASSIFICATION USING KNN ALGORITHM

K-nearest neighbors (KNN) algorithm is a type of supervised machine learning algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry.

KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification. KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

WORKING OF KNN ALGORITHM

k-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps

Step 1 – For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 – Next, we need to choose the value of K (user defined constant) i.e the nearest data points K can be any integer.

Step 3 – For each point in the test data do the following –

- (i) Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance, Minkowski. The default metric is minkowski.
- (ii) Now, based on the distance value, sort them in ascending order.
- (iii) Next, it will choose the top K rows from the sorted array.
- (iv) Now, it will assign a class to the test point based on most frequent class of these rows.

PARAMETER SELECTION

The best choice of k depends upon the data, generally larger values of k reduces effect of the noise on the classification, but make boundaries between classes less distinct. A very low value of k such as k=1, k=2 can be noisy and lead to the effect of outliers in the model A good k can be selected by various heuristic techniques.

MINKOWSKI DISTANCE

Minkowski distance is a distance/similarity measurement between two points in the normed vector space. It is a generalization of the Euclidean and Manhattan distances.

$X=(x_1, x_2, \dots, x_n)$ and $Y=(y_1, y_2, \dots, y_n)$ be the two points then minkowski distance is defined as,

$$D(x,y)=\sum_{i=1}^n (|x_i - y_i|^p)^{1/p}$$

BENEFITS OF USING KNN ALGORITHM

- It is very simple algorithm to understand and interpret
- It is very useful for nonlinear data because there is no assumption about data in this algorithm
- It is versatile algorithm as we can use it for classification as well as regression
- It has relatively high accuracy
- It is robust to the noisy training data and it can be more effective if the training data is large.

GENERATING KNN CLASSIFIER MODEL

Import the KNeighborsClassifier module from Sklearn Neighbors library and create KNN classifier object by passing argument number of neighbors in KNeighborsClassifier () function. Then, fit model on the train set using fit () and perform prediction on the test set using predict ().

E. TEXT TO SPEECH

The output of the classification step is the class which the image belongs to, which is in the text format. The text is converted to speech using the Gtts

(Google-text-to-speech) python library. Gtts supports many languages and sounds very natural. The speech can be delivered in any one of the two available audio speeds, fast or slow. However it is not possible to change the voice of the generated audio.

4.2.2 SPEECH TO IMAGE CONVERSION

- The speech is given as input through microphone.
- The speech is converted to Text using the Google API.
- The sign images are displayed for each character in the text.

4.2.2.1 GOOGLE API

- Google Speech-to-text is used to convert audio to text by applying powerful Neural Network models in an easy- to-use API. It can process real-time streaming or prerecorded audio, using Google's machine learning technology. The API recognizes more than 120 languages and variants to support global user base. It can be used to enable voice command-and-control, transcribe audio from call centres and more.
- Speech-to-Text can stream text results, immediately returning text as it's recognized from streaming audio. It's capable of analysing short-form and long-form audio. It works well with real-life speech and can accurately transcribe proper nouns and appropriate format language.
- Speech-to-Text comes with multiple pre-built enhanced machine learning models such as command and search, phone call, video so that we can optimize speech recognition for our use cases. Google has trained these speech recognition models for specific audio types and sources. Video

model is used for transcribing audio in audio clips, phone call method can be used for transcribing audio from a phone call, ASR Command and search can be used for transcribing shorter audio clips and ASR Default model can be used if your audio does not fit one of the previously described models.

- It has features such as speech adaptation, global vocabulary, real-time streaming or pre-recorded audio support, auto – detect language, inappropriate content filtering, automatic punctuation, model selection, speaker diarization, multichannel recognition. It provides Automatic speech recognition (ASR) powered by deep learning neural networking and it can handle noisy audio from many environments without requiring additional noise cancellation.
- Speech-to-text has three main methods to perform speech recognition they are Synchronous recognition, Asynchronous recognition, streaming recognition. The system uses the streaming recognition method, Streaming Recognition (GRPC) performs recognition on audio data provided within a GRPC bi-directional stream. Streaming requests are designed for real-time recognition purposes, such as capturing live audio from a microphone.

4.2.2.2 SPEECH RECOGNITION

The speech recognition process is divided into different components such as:

A. Speech

The block represents the acoustic environment plus the transduction equipment (microphone, preamplifier and AD-converter) can have a strong effect on the generated speech recognition.

B. Signal preprocessing

Signal pre-processing is the process of modifying the audio for the purpose of analysis or enhancement. It can be used for deriving acoustic representations that are good at separating classes of speech sounds and effective at suppressing irrelevant sources of variation.

C. Feature extraction

It involves extracting speech specific features of the pre-processed signal. This can be done with a variety of techniques like cepstrum analysis and the spectrogram. The MFCC (Mel Frequency Cepstral Coefficient) is the standard method for feature extraction in speech recognition. The steps involved in MFCC are pre-emphasis, Framing, Windowing, FFT, Mel filter bank, computing DCT. The most commonly used acoustic features are Mel-Scale Frequency Cepstral coefficients.

D. Speech classification

The commonly used techniques for speech classification include:

(i).Dynamic Time Warping (DTW)

This technique compares words with reference words. Every reference word has a set of spectra; but there is no distinction between separate sounds in the word. Because a word can be pronounced at different speeds, a time normalization will be necessary. Dynamic Time Warping is a programming technique where the time dimension of the unknown word is changed until there is a similarity with a reference word.

(ii).Hidden Markov Modelling (HMM)

This is the most successful and most used pattern recognition method for speech recognition. It's a mathematical model derived from a Markov Model. Speech recognition uses a slightly adapted Markov Model. Speech is split into the smallest audible entities (not only vowels and consonants but also conjugated sound. All these entities are represented as states in the Markov Model. According to transition probabilities there exist a transition from one state to another. A state can also have a transition to its own if the sound repeats itself. Markov Models seems to perform quite well in noisy environments because every sound entity is treated separately. If a sound entity is lost in the noise, the model might be able to guess that entity based on the probability of going from one sound entity to another.

(iii).Neural Networks (NN)

Neural networks have many similarities with Markov models. Both are statistical models which are represented as graphs. Where Markov models use probabilities for state transitions, neural networks use connection strengths and functions. A key difference is that neural networks are fundamentally parallel while Markov chains are serial. This means that both techniques are very powerful in a different context. As in the neural network, the challenge is to set the appropriate weights of the connection, the Markov model challenge is finding the appropriate transition and observation probabilities. In many speech recognition systems, both techniques are implemented together and work in a symbiotic relationship. Neural networks perform very well at learning phoneme probability from highly parallel audio input, while Markov

models can use the phoneme observation probabilities that neural networks provide to produce the likeliest phoneme sequence or word.

4.3 UML DIAGRAM

The Unified Modelling Language is a standard visual modelling language intended to be used for modelling business and similar processes, analysis, design and implementation of software-based system. It is a common language for business analysts, software architects and developers used to describe, specify, design and document existing or new business processes, structure and behavior of artifacts of software system. UML is a standard modelling language, not a software development process. It provides guidance as to the order of a team's activities, specific what artifacts should be developed, directs the task of individual developers and the team as a whole and offers criteria for monitoring and measuring a project's product and activities.

4.3.1 USECASE DIAGRAM

A use case diagram is a representation of a user's interaction with the system that shows the relationship between and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases. A use-case diagram can help provide a higher-level view of the system and used as a good communication tool for stakeholders. The use cases are represented by either circles or ellipses. The diagram captures the core functionalities of the system and visualize the interactions.

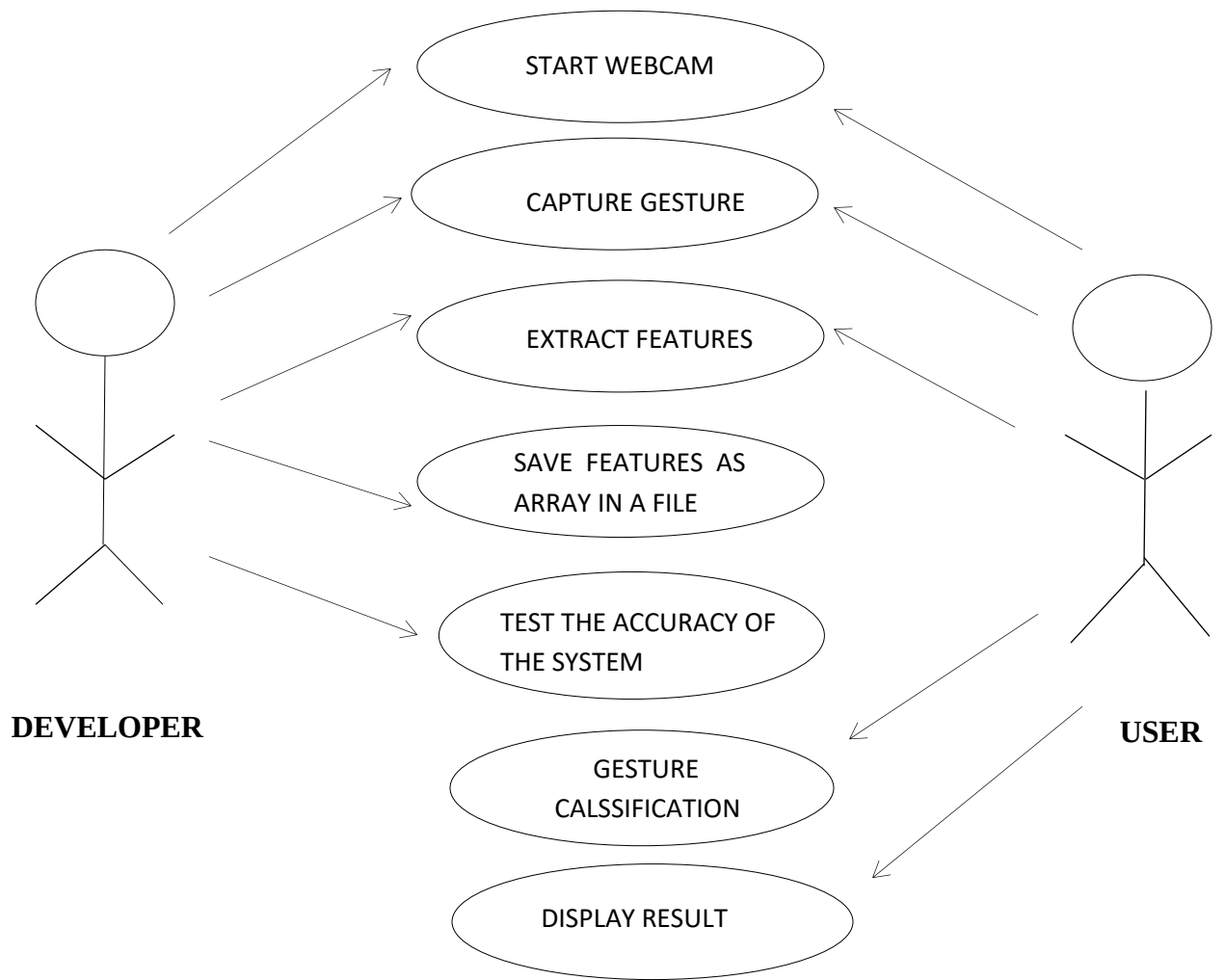


Figure 4.2 SIGN TO VOICE USE CASE DIAGRAM

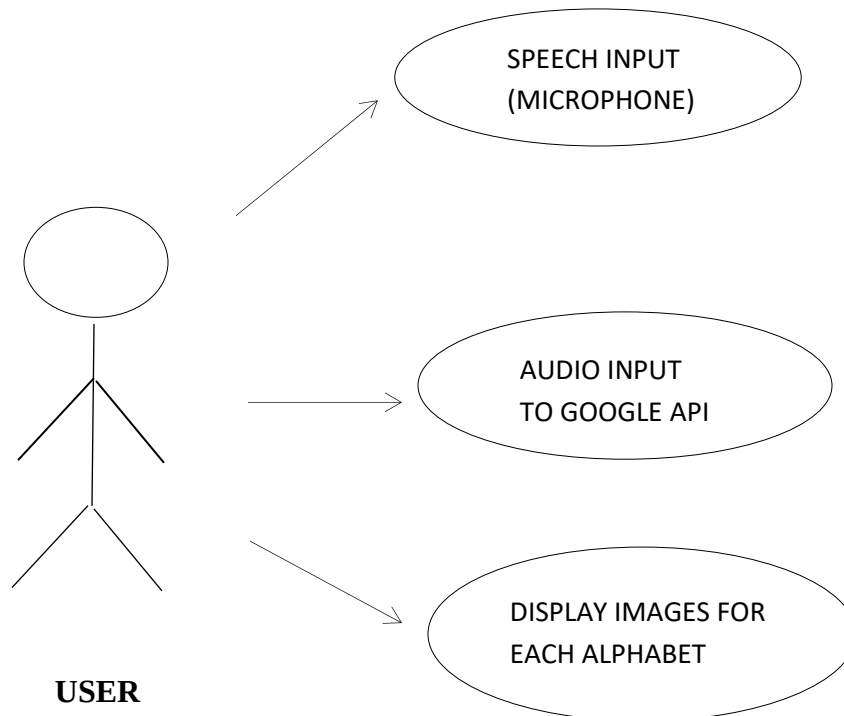


Figure 4.3 VOICE TO SIGN USE CASE DIAGRAM

4.3.2 SEQUENCE DIAGARM

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranges in time sequence. It depicts the object and classes involved in the scenario and the sequence of messages exchanged between the object needed to carry out the functionality of the scenario. Sequence diagram are typically associates with use case realization in the Logical View of the system under development. Sequence diagrams are sometimes called event diagram or events scenarios. Sequence diagrams are time focus and they show the order

of the interaction visually by using the vertical axis of the diagram to represent time. This allow the specification of simple runtime scenarios in a graphical manner shown in fig

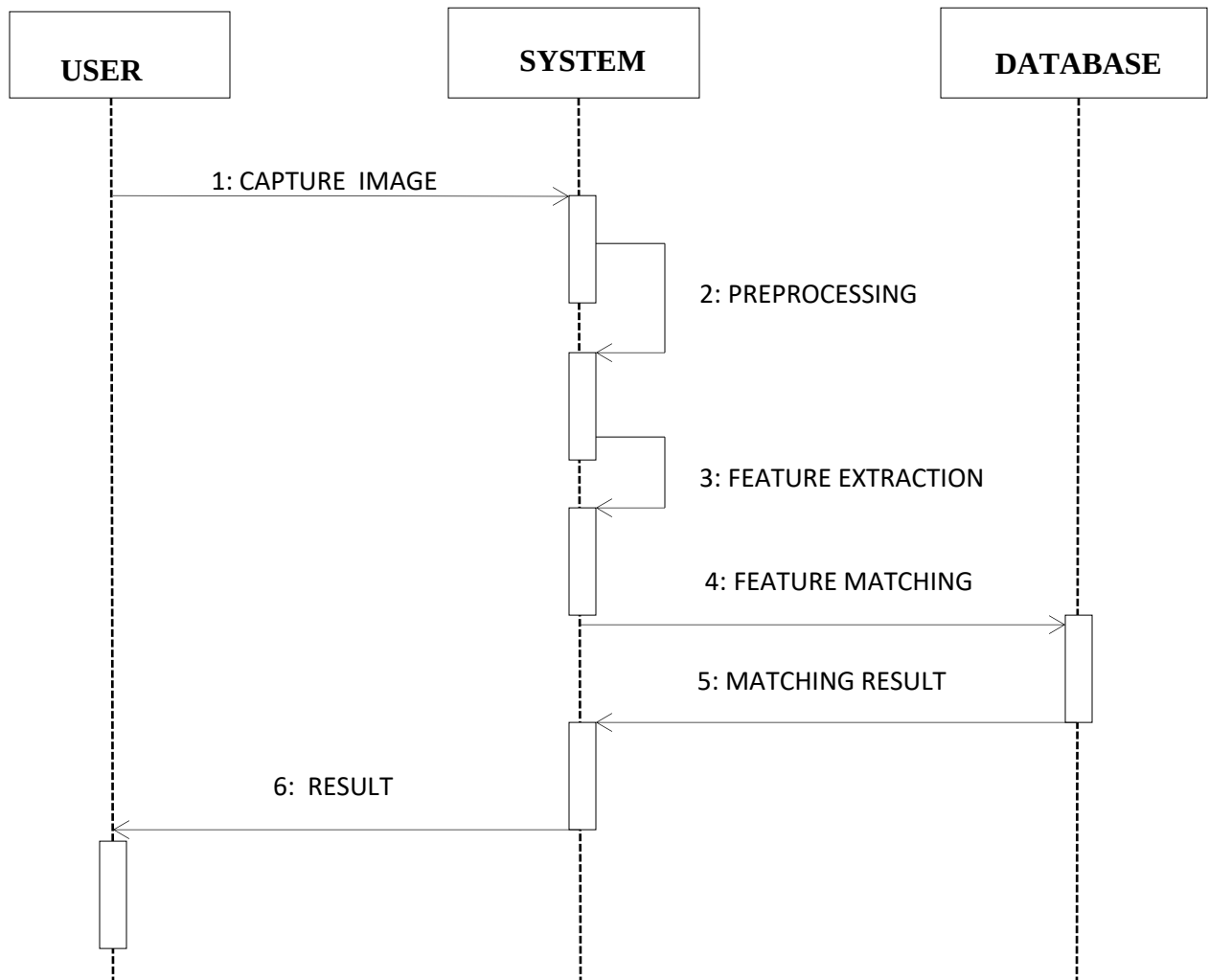


Figure 4.4 SIGN TO VOICE SEQUENCE DIAGRAM

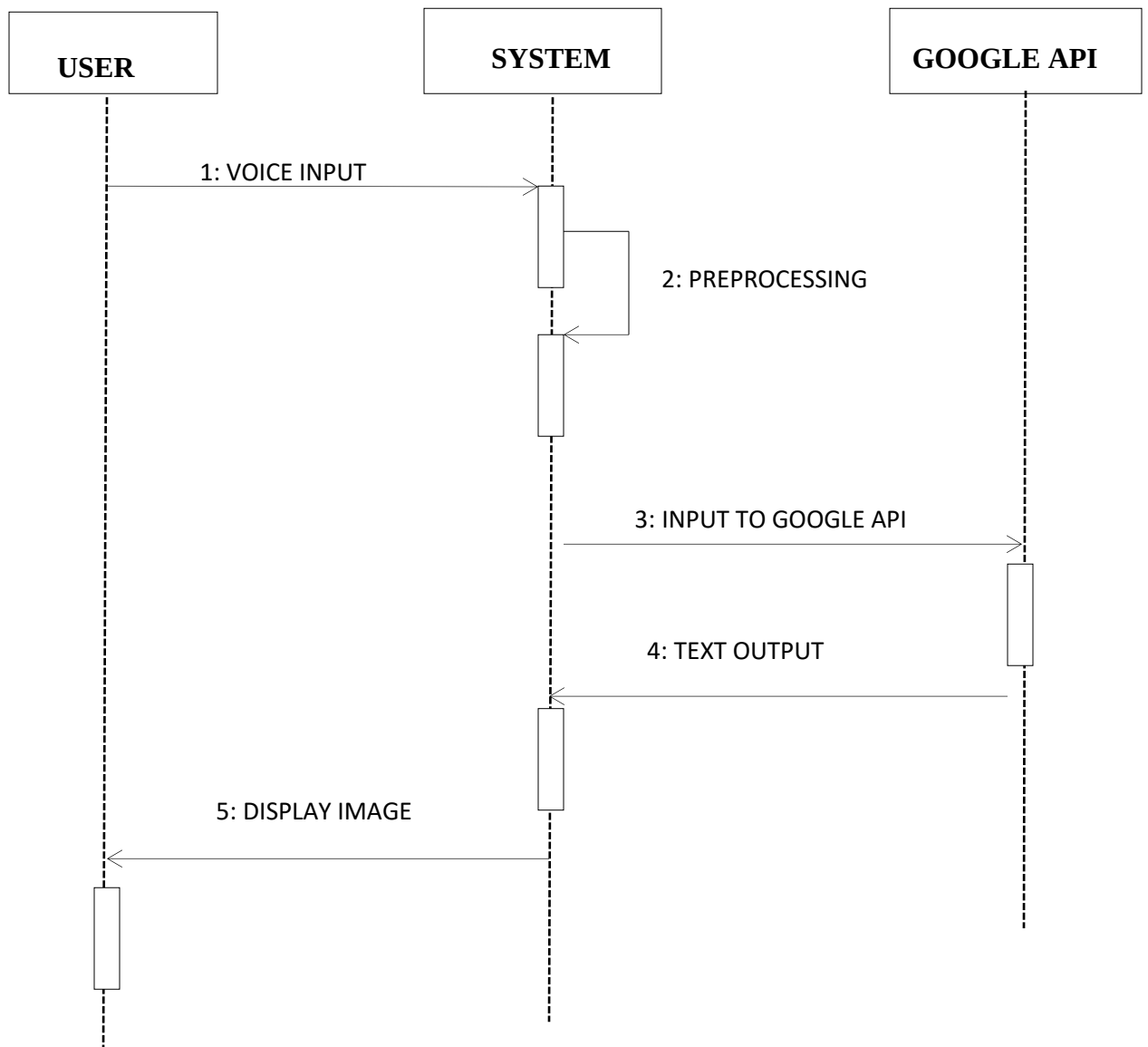


Figure 4.5 VOICE TO SIGN SEQUENCE DIAGRAM

4.3.3 COLLABORATION DIAGRAM

A Communication diagram models the interaction between objects or parts in terms of sequenced system. Communication diagram represent a combination

of information taken from Class, Sequence and Use case diagrams describing both the static structure and dynamic behavior of a system. It use the free-form arrangement of object and links as used in Object diagrams. It show a lot of the same information as sequence diagrams, but because of how the information is presented, some of it is easier to find in one diagram than the other. Communication diagrams show which elements each one interacts with better, but sequence diagrams show the order in which the interactions take place more clearly and shown in Fig

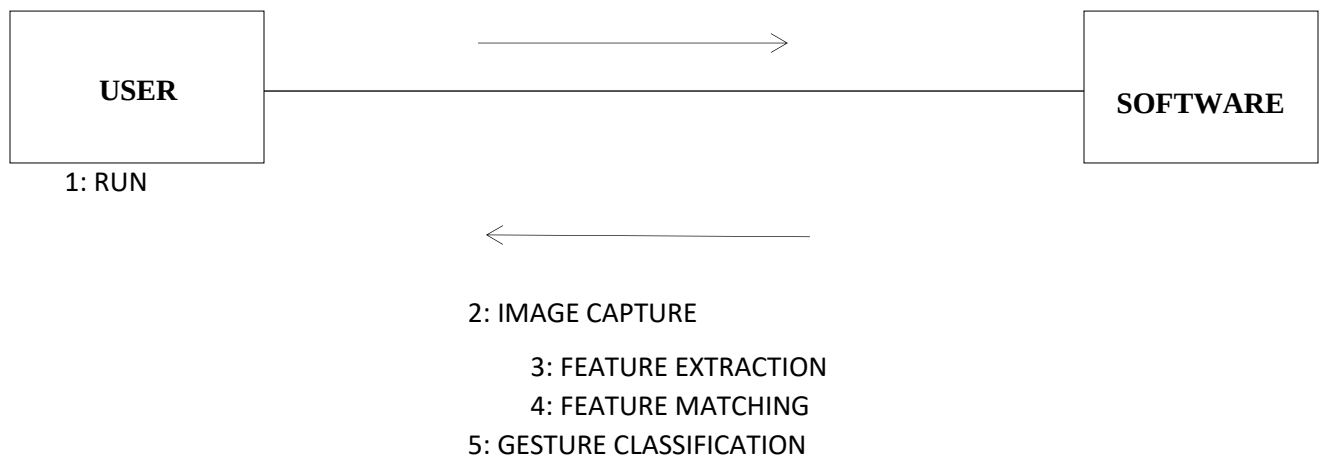


Figure 4.6 SIGN TO VOICE COLLABORATION DIAGRAM

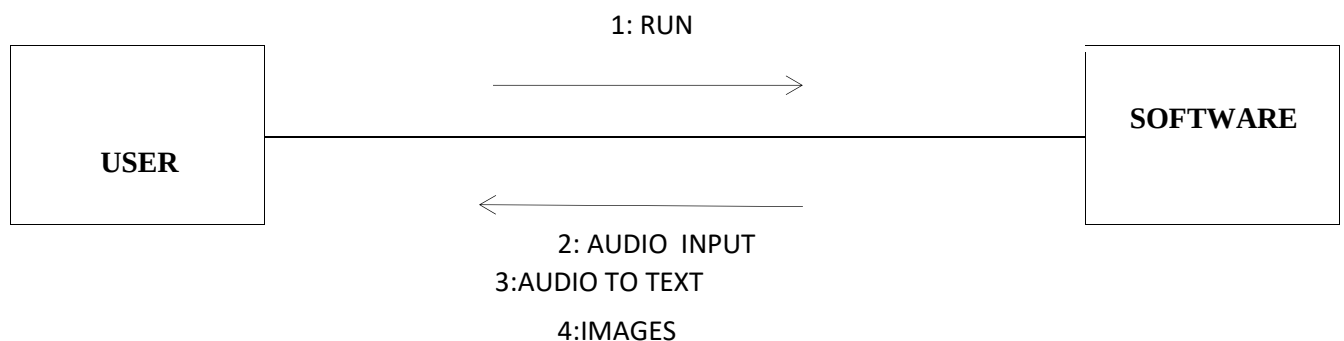


Figure 4.7 VOICE TO SIGN COLLABORATION DIAGRAM

4.3.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). It shows the overall flow of control. It may be regarded as a form of flowchart. Typical flowchart techniques lack constructs for expressing concurrency.

Activity diagrams are constructed from a number of shapes:

- Ellipses represent actions
- Diamond represent decisions
- Bars represent the a start or end of concurrent activities
- Black circle represent the start of the flow
- Encircled black circle represents the end
- Arrows run from the start towards the end and represent the order in which activities

happen.

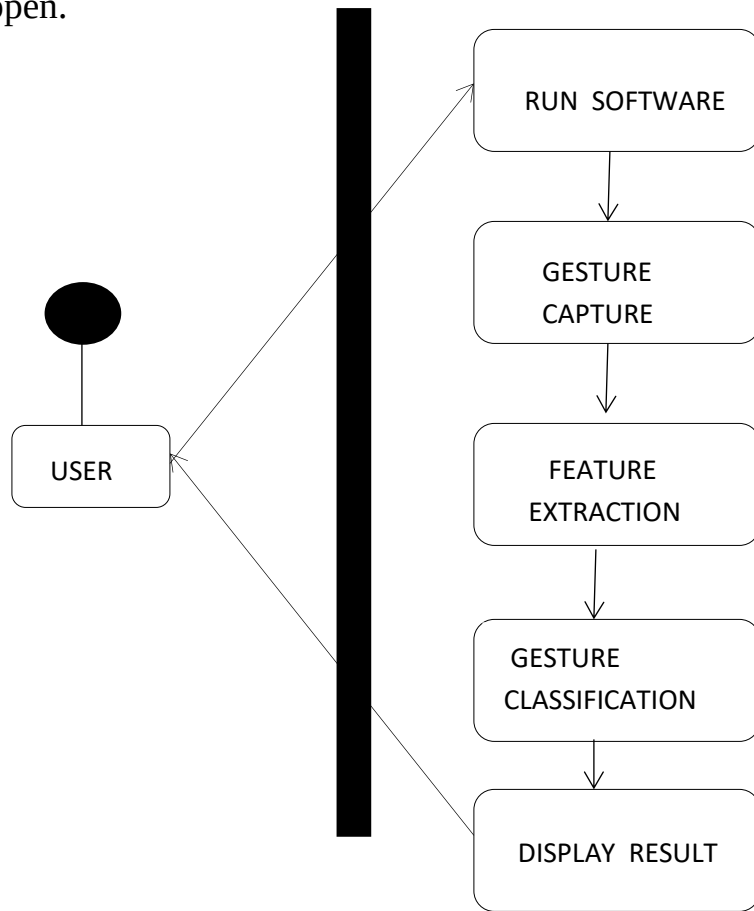


Figure 4.8 SIGN TO VOICE ACTIVITY DIAGRAM

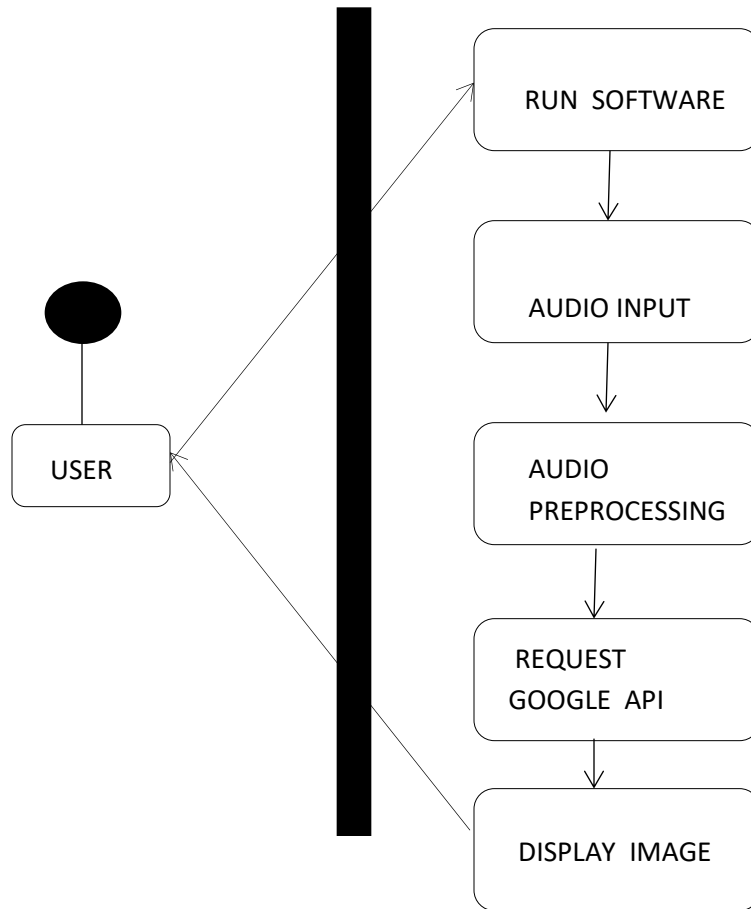


Figure 4.9 VOICE TO SIGN ACTIVITY DIAGRAM

4.4 SUMMARY

This chapter deals with system design in detail. The various UML diagrams gives a list of what take place in project.

CHAPTER 5

IMPLEMENTATION AND RESULT

5.1 GENERAL

Implementation is the most crucial stage in achieving a successful system and giving the user confidence that the new system is workable and effective. Implementation is the stage of the project when the theoretical design is turned out into a working system.

5.2 IMPLEMENTATION

5.2.1 SIGN TO VOICE

Thus the figure 5.1 shows the initial process of capturing the images of the sign languages and storing it in the respective alphabet folder.

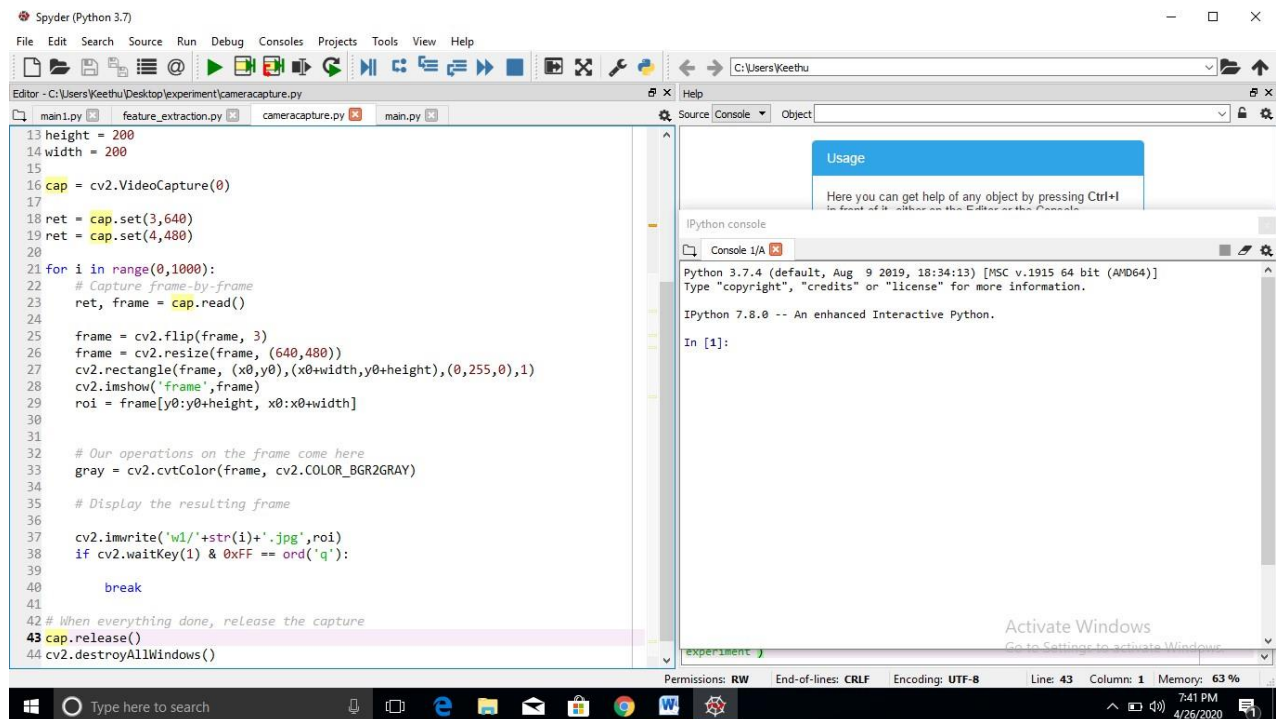


Figure 5.1 Process of Capturing Image

The figure 5.2 shows the implementation of feature extraction process. The images in the dataset is given as input to the feature extraction algorithm

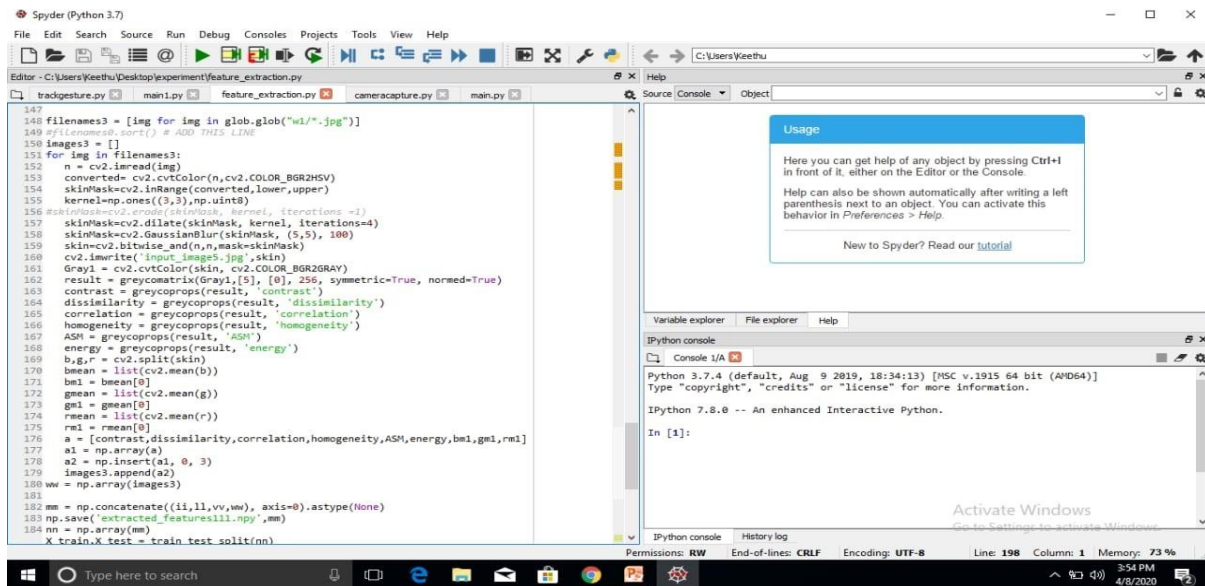


Figure 5.2 Feature Extraction Process

The figure 5.3 shows the implementation of the main program where the input image is captured and given to the classifier through predict () function.

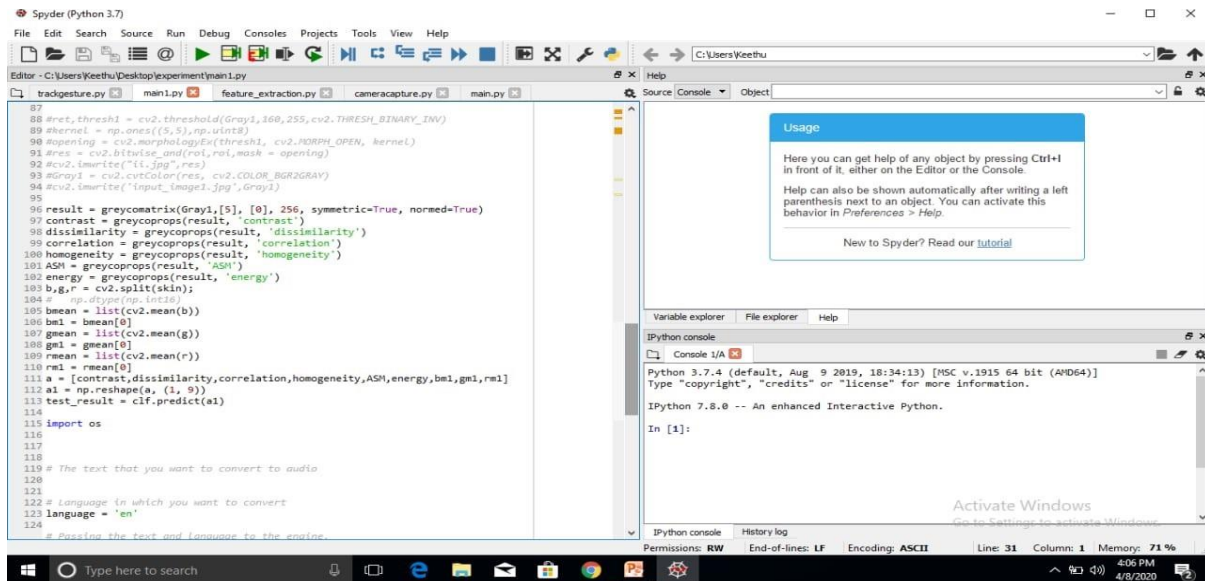


Figure 5.3 Main Program

Thus the figure 5.4 shows the implementation of the first module, where the input image is captured and the class to which the input image belongs to is displayed with the respective audio.

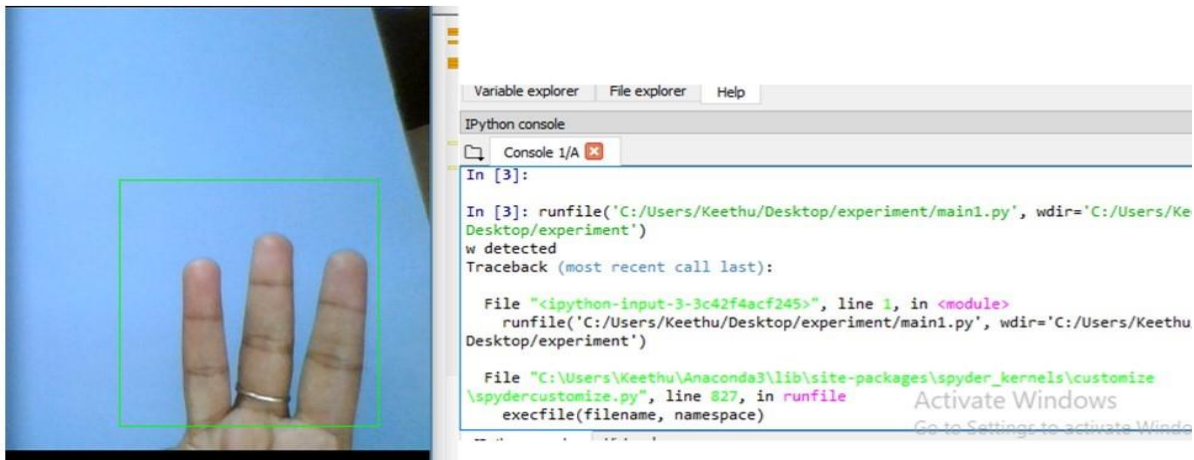


Figure 5.4 Module 1 Output

5.2.2 VOICE TO SIGN

The figure 5.5 shows the process of capturing the audio, followed by preprocessing and giving the processed audio as input to the Google API.

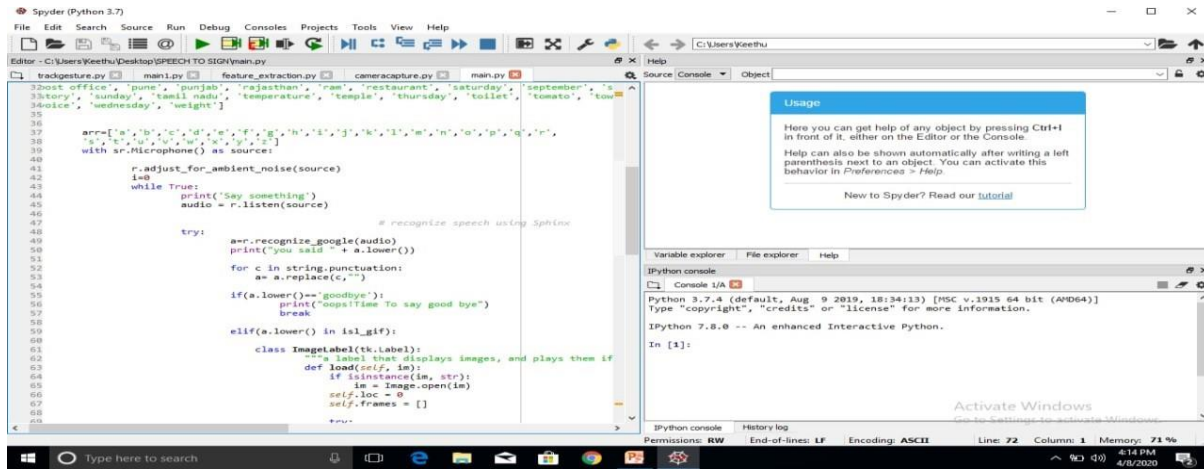


Figure 5.5 Recording Audio and Requesting Google API

The figure 5.6 shows the implementation of the second module, where the audio is displayed in the text format and sign images are displayed.



Figure 5.6 Module 2 Output

5.3 RESULTS

The sign images are classified using the KNN classification algorithm based on the feature extracted from the image using GLCM feature extraction algorithm and the audio is generated for the respective class. In the other direction, the audio is converted to text using the Google API and the sign images are displayed.

5.4 SUMMARY

In this section the step by step implementation of the project is illustrated. The screenshots shown the intermediate results generated by each module.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The major goal of the project is to convert the hand gesture language into audio format and audio to hand gesture for developing an easy interaction between a normal person and a disabled person. The main function of the project is implemented by recognizing the sign language of a disabled person through KNN classification algorithm. The main features of the system are high usability, easy communication, high stability, time saving. This project would be widely used by Government sectors, Banking, Teaching, public welfare etc.

6.2 FUTURE ENHANCEMENT

The system can be extended to incorporate the knowledge of facial expressions and body language. A mobile and web Based version of the application will increase the reach to move people.

6.3 SUMMARY

The chapter gives an overview of the conclusion of the system and the future works that will make the system a better one.

REFERENCES

1. B.Lakshmi, Rasheed Ahamed, Harshali Rane, and Ravi Kishore Kodali,“*Assistive Sign Language Converter for Deaf and Dumb*”,International Conference on Internet of Things(ICIoT-2019)
2. Rupesh Prajapati¹, Vedant Pandey², Nupur Jamindar³, Neeraj Yadav⁴, Prof. Neelam Phadnis⁵,”*Hand Gesture Recognition and Voice Conversion for Deaf and Dumb*”, International Research Journal of Engineering and Technology (IRJET2018)
3. Abey Abraham, Rohini V,”*Real time conversion of sign language to speech and prediction of hand gestures of using Artificial neural network*”, 8th International Conference on Advances in Computing and Communication (ICACC-2018)
4. Kusumika Krori Dutta, Sunny Arokia Swamy Bellary, ”*Machine Learning Techniques for Indian Sign Language Recognition*”, International Conference on Current Trends in Computer, Electrical, Electronics and Communication (ICCTCEEC-2017)

5. Mr. Sanket Kadam, Mr. Aakash Ghodke, Prof. Sumitra Sadhukhan, "*Hand Gesture Recognition Software Based on Indian Sign Language*" International Conference on IEEE Green Computing and Communications-2017
6. Taner Arsan and Oğuz Ülgen, "*Sign Language Converter*", International Journal of Computer Science & Engineering Survey (IJCSES-2015)
7. Rajaganapathy. S1, Aravind. B2, Keerthana. B3, Sivagami. M4, "*Conversation of Sign Language to Speech with Human Gestures*", 2nd International Symposium on Big Data and Cloud Computing (ISBCC-2015)
8. M.Suresh Anand, A.Kumaresan, "*An Integrated Two Way ISL (Indian Sign Language) Translation System – A New Approach*", International Journal of Advanced Research in Computer Science-2013