HAND GESTURE DETECTION AND EYE MOTION TRACKER

A project synopsis submitted in fulfilment of the Academic requirementsfor the award of the Degree of

BACHELOR OF ENGINEERING
INFORMATION TECHNOLOGY

By

G.SATYA SWAROOP	**(2451-19-737-014)**
T.RITHIKA	**(2451-19-737-019)**
M.SAI PRIYANKA	**(2451-19-737-016)**

Under the guidance of

Mrs.Devaki Kuthadi

Assistant Professor,
Dept. of I.T

Maturi Venkata Subba Rao (MVSR)
Engineering College Nadergul, Hyderabad.

DEPARTMENT OF INFORMATION TECHNOLOGY
MATURI VENKATA SUBBA RAO (MVSR) ENGINEERING COLLEGE
(An Autonomous Institution)
(Affiliated to Osmania University, Hyderabad. Recognized by AICTE)
Nadergul, Saroornagar Mandal, Hyderabad-501510 2022-23

MATURI VENKATA SUBBA RAO (MVSR) ENGINEERING COLLEGE

(An Autonomous Institution)
(Affiliated to Osmania University, Hyderabad. Recognized by AICTE)Nadergul, Saroornagar Mandal, Hyderabad-501510

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project work entitled "Hand Gesture Detection and Eye Motion Tracker" is a bona fide work carried out by **G.Satya Swaroop (2451-19-737-014), T.Rithika (2451-19-737-019), M.Sai Priyanka (2451-19-737-016)** in partial fulfilment of the requirements for the award of degree of **Bachelor of Engineering** in **Information Technology** from **Maturi Venkata Subba Rao Engineering College,** affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2022-23. under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Signature of the Coordinator Signature of Guide

Signature of Head, ITD Signature of External Examiner

DECLARATION

We hereby declare that the contents presented in the Project Thesis titled **“HAND GESTURE DETECTION AND EYE MOTION TRACKER”** submitted in partial fulfillment for the award of Degree of Bachelor of Engineering *in INFORMATION TECHNOLOGY (IT), MATURI VENKATA SUBBA RAO (MVSR) ENGINEERING COLLEGE* affiliated to *OSMANIA UNIVERSITY, Hyderabad*
is a record of the original work carried out by us under the supervision Of **Mrs. DEVAKI KUTHADI (Assistant Professor).** Further this is to state that the results embodied in this project report have not been submitted to any University or Institution for the award of any Degree or Diploma.

Signature of the Student	Signature of the Student	Signature of the Student
G.SATYA SWAROOP	**T.RITHIKA**	**M.SAI PRIYANKA**
2451-19-737-014	**2451-19-737-019**	**2451-19-737-016**

ACKNOWLEDGEMENT

We with extreme jubilance and deepest gratitude would like to thank **Mrs. Devaki Kuthadi, Assistant Professor**, Department of Information Technology, Maturi Venkata Subba Rao (MVSR) Engineering College, for her constant encouragement and facilities provided to us to complete our project in time.

We would like to extend our gratitude to **Dr. D. Shanthi, Project Co-Ordinator and Associate Professor and Mrs. Devaki Kuthadi, Project Co-Ordinator and Assistant Professor** Department of Information Technology, Maturi Venkata Subba Rao Engineering College, for his valuable suggestions and timely help during the course of the project.

With immense pleasure, we record our deep sense of gratitude to **Dr. K.Venu Gopal Rao Dean of academics and Head of the Department** of Information Technology, MVSR Engineering College, for permitting us to carry out this project.

We express, from the bottom of my heart, my deepest gratitude to my parents and family for the support, dedication, comprehension, and love.

Finally, we express our heartfelt thanks to each and everyone who directly and indirectly helped us in the successful completion of this project work.

G. SATYA SWAROOP (2451-19-737-014)
T. RITHIKA (2451-19-737-019)
M.SAI PRIYANKA (2451-19-737-016)

Vision & Mission

MVSR Engineering College Department of Information Technology

COURSE NAME: MINI PROJECT-I

COURSE CODE: PW653IT

VISION

To impart technical education to produce competent and socially responsible engineers in the fieldof Information Technology.

MISSION

M1. To make teaching learning process effective and stimulating.
M2. To provide adequate fundamental knowledge of sciences and Information Technology withpositive attitude.
M3. To create an environment that enhances skills and technologies required for industry.M4. To encourage creativity and innovation for solving real world problems.
M5. To cultivate professional ethics in students and inculcate a sense of responsibility towardssociety.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

The Bachelor’s program in Information Technology is aimed at preparing graduates who will:

I. Apply knowledge of mathematics and Information Technology to analyze, design and implement solutions for real world problems in core or in multidisciplinary areas.
II. Communicate effectively, work in a team, practice professional ethics and apply knowledgeof computing technologies for societal development.
III. Engage in Professional development or postgraduate education to be a life-long learner.

PROGRAM OUTCOMES (POs)

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate considerationfor the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clearinstructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leaderin a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOS):

(1) Hardware design: An ability to analyze, design, simulate and implement computer hardware / software and use basic analog/digital circuits, VLSI design for various computing and communicationsystem applications.

(2) Software design: An ability to analyze a problem, design algorithm, identify and define the computing requirements appropriate to its solution and implement the same.

COURSE OBJECTIVES:

1. To enhance practical & Professional skills.
2. To familiarize the tools and techniques of symmetric literature survey and documentation.
3. To expose students to industry practices and teamwork.
4. To encourage students to work with innovative and entrepreneurial ideas.

COURSE OUTCOMES:

On successful completion of this course students will be able to:

1. Define a problem of the recent advancements with applications towards society.
2. Outline requirements and perform requirement analysis for solving the problem.
3. Design and develop a software and/or hardware- based solution within the scope ofproject using contemporary technologies and tools.
4. Test and deploy the applications for use.
5. Develop the Project as a team and demonstrate the application, with effective writtenand oral communications.

ABSTRACT

The basic goal of Human Computer Interaction is to improve the interaction between users and computers by making the computer more receptive to user needs. Human Computer Interaction with a personal computer today is not just limited to keyboard and mouse interaction. Interaction between humans comes from different sensory modes like gesture, speech, facial and body expressions. Being able to interact with the system naturally is becoming ever more important in many fields of Human Computer Interaction. an individual human computer interface system using eye motion and hand gestures is introduced. Traditionally, human computer interface uses mouse, keyboards an input device. This paper presents interface between computer and human. This technology is intended to replace the conventional computer screen pointing devices for the use of disabled. The paper presents a novel idea to control computer mouse cursor movement with human eyes and hand gestures. Hand gesture is used as a mechanism for interaction with the computers.

The use of a physical controller like mouse, keyboard for human computer interaction hinders natural interface as there is a strong barrier between the user and computer. In this paper, we have designed a robust marker- less hand gesture recognition system which can efficiently track both static and dynamic hand gestures. Our system translates the detected gesture into actions such as opening websites and launching applications like VLC Player and PowerPoint. The dynamic gesture is used to shuffle through the slides in presentation. Our results show that an intuitive HCI can be achieved with minimum hardware requirements.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

1.1. PROBLEM STATEMENT

- Deaf and Dumb people in our society face a huge Communication Barrier.
- Our primary goal is to implement the required set of signs in sign language for understanding the sign and displaying the actual meaning of the sign.

1.2. OBJECTIVES

Objective of this project is to create a complete system to detect, recognize and interpret the hand gestures through computer vision. Humans use gestures to interact with the environment. Gestures such as touching, grabbing, and dragging are used in the real world. Using this to control computers could make for a more natural interaction than what is currently used.
Gestures and eye tracking have been used in many projects to enable a new sort of interaction, although it is often as separate inputs.
Creating a multimodal interaction by combining eye tracking and gestures is an area which remains mostly unexplored.

1.3. MOTIVATION

Biometric technologies make use of various physical and behavioral characteristics of human such as fingerprints, expression, face, hand gestures and movement. These feature is then processed using sophisticated machines for detection and recognition and hence used for security purposes. Unlike common security measures such as passwords, security cards that can easily be lost, copied or stolen; these biometric features are unique to individuals and there is little possibility that these pictures can be replaced or altered. Among the biometric sector hand gesture recognition are gaining more and more. Attention because of their demand regarding security for law enforcement agency as well as in private sectors such as surveillance systems

Hand gestures are important to intelligent human and computer interaction to build fully automated systems that analyze information contained in images, fast and efficient hand gesture recognition algorithms are required.

1.4. EXISTING SYSTEM

In the existing system, the project would not recognize multiple hands or hand gestures. If there are multiple hands visible to the computer, then it fails to recognize the hands and classify the gestures. The Existing system doesn’t have the feature of recognizing ‘Eye motion’.

Drawbacks:

Based on the drawbacks of the project we would like to incorporate a few changes to this existing project making it efficient and add a new feature to it. Since Facial expressions are also included in the Non-verbal communication, we would like to add the feature of Eye motion tracking which would make our project more efficient in understanding the expressions.

ADVANTAGES:

The most important advantage of the usage of hand gesture-based input modes is that using this method the user can interact with the application from a distance without any physical interaction with the keyboard or mouse. The gesture vocabulary designed can be further extended for controlling different applications like game control etc. As the system provides the flexibility to the users and specifically physically challenged users to define the gesture according to their feasibility and ease of use. Gestures allow the user to handle multiple points of input and even define several parameters at once. They are, therefore, a more natural form of communication. Hand gestures can help you describe what you’re talking about. Eye tracking allows researchers to study the movements of a participant's eyes during a range of activities. It records actual eye movements.

1.5. SCOPE AND PURPOSE

The scope of the project is to construct a synchronous gesture classifying system that can recognize gestures in lighting circumstances spontaneously. To achieve this goal, a synchronous gesture which based on real time is generated to recognize gestures. An intention of this project is to generate a complete system which can identify, spot and explain the hand motioning through computer sight. This structure will work as one of the envisioning of computer sight and AI with user interaction. It creates function to identify hand motion based on various arguments.

2.2. LITERATURE SURVEY

S. NO	AUTHOR NAME	PROBLEMS IDENTIFIED	TECHNIQUES USED	ACCURACY	DRAWBACKS
1	Mahmoud E and Bernd M	To Recognize the Isolated and Meaningful Hand Gesture.	Hidden Markov Model	93.84%	It is having high Computational consuming.
2	H.R. Chennamm	To track eyes directly based on the photometric appearance.	Appearance based techniques	90.2%	Appearance-based methods typically do not require calibration of cameras and geometry data since the mapping is made directly on the image contents.
3	Robust	Gesture recognition for robotic control	Dynamic gesture recognition	93.2%	Using lowest pixels cameras
4	J.H. Goldberg et al, Int. J. Indust. Ergon	Computer interface evaluation using eye movements: methods and constructs	Eye movements HCI Computer interface design Software evaluation Fixation algorithms	80.5%	Used derivative measures of only a single oculomotor event instead of combinations of multiple events
5	Zaman Khan, Noor Adnan Ibraheem.	To recognize the hand gestures	Neural Network, HMM, fuzzy c-means clustering	92.2%	Problems in orientation histogram method.

SYSTEM REQUIREMENT SPECIFICATIONS

3.1 SOFTWARE REQUIREMENTS:

• Operating system	:	Windows xp/10
• Front End	:	Java AWT, Swing(JFC)
• Software	:	Python , Open CV

3.2 HARDWARE REQUIREMENTS:

• Processor	:	I3.
• Hard Disk	:	80 GB.
• Input Devices	:	Keyboard, Mouse ,Webcam
• Ram	:	1 GB

4.TECHNOLOGIES USED

Python:

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as wellas its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0.Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle-detecting garbage collection system (in addition to reference counting). Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020. Python consistently ranks as one of the most popular programming languages.

Fig:7.Python

VSCode:

Visual Studio Code is a distribution of the `Code - OSS` repository with Microsoft-specific customizations released under a traditional Microsoft product license. Visual Studio Code combines the simplicity of a code editor with what developers need for their core edit-build-debug cycle. It provides comprehensive code editing, navigation, and understanding support along with lightweight debugging, a rich extensibility model, and lightweight integration with existing tools.

Visual Studio Code is updated monthly with new features and bug fixes. You can download it for Windows, macOS, and Linux on Visual Studio Code's website. To get the latest releases every day, install the Insiders build.

Visual Studio Code, also commonly referred to as VS Code,[9] is a source-code editor made by Microsoft for Windows, Linux and macOS.[10] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality

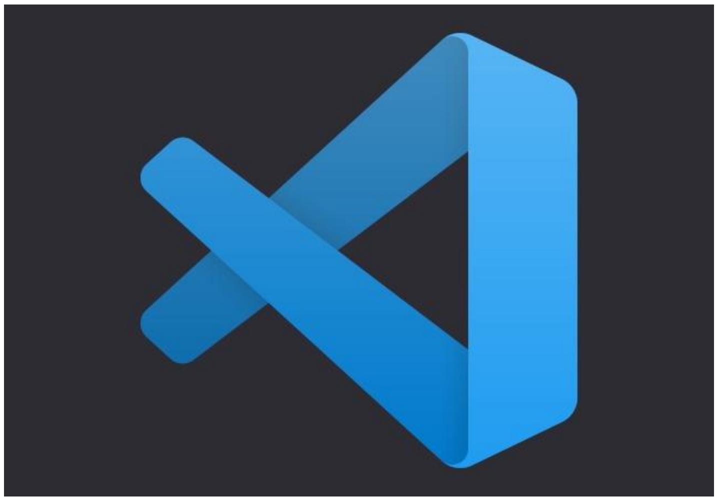


Fig:8.VSCode

OPENCV:

OpenCV (*Open Source Computer Vision Library*) is a library of programming functions mainly aimed at real-time computer vision. 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 Apache 2 License. Starting in 2011, OpenCV features GPU acceleration for real-time operations.

OpenCV is a cross-platform library using which we can develop real-time computer visionapplications. It mainly focuses on image processing, video capture and analysis including featureslike face detection and object detection.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

Fig:9.OpenCV

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3Dpoint clouds from stereo cameras, stitch images together to produce a high resolution image of anentire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony,Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms andabout 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

MEDIA PIPE:

Media Pipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform Framework works in Desktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano. The Media Pipe perception pipe line is called a graph. The ability to perceive the shape and motion of hands can be a vital component in improving the user experience across a variety of technological domains and platforms. For example, it can form the basis for sign language understanding and hand gesture control, and can also enable the overlay of digital content and information on top of the physical world in augmented reality. While coming naturally to people, robust real-time hand perception is a decidedly challenging computer vision task, as hands often occlude themselves or each other (e.g. finger/palm occlusions and handshakes) and lack high contrast patterns.
Media Pipe Hands is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame. Whereas current state-of-the-art approaches rely primarily on powerful desktop environments for inference, our method achieves real-time performance on a mobile phone, and even scales to multiple hands.

Fig: 10.Media Pipe

TENSORFLOW

TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015.Google releasedthe updated version of TensorFlow, named TensorFlow 2.0, in September 2019.

TensorFlow can be used in a wide variety of programming languages, most notably Python, as well as JavaScript, C++, and Java. This flexibility lends itself to a range of applications in many different sectors.

TensorFlow serves as the core platform and library for machine learning. TensorFlow's APIs useKeras to allow users to make their own machine learning models. In addition to building and training their model, TensorFlow can also help load the data to train the model, and deploy it using TensorFlow Serving.

TensorFlow provides a stable Python API, as well as APIs without backwards compatibility guarantee for JavaScript, C++, and Java. Third-party language binding packages are also available for C#, Haskell, Julia, MATLAB, R, Scala, Rust, OCaml, and Crystal. Bindings that are now archived and unsupported include swift and go.

TensorFlow is well-documented and includes plenty of machine learning libraries. It offers a fewimportant functionalities and methods for the same.

TensorFlow is also called a "Google" product. It includes a variety of machine learning and deeplearning algorithms. TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding and creation of various sequence models.

- It includes a feature that defines, optimizes and calculates mathematical expressions easilywith the help of multi-dimensional arrays called tensors.
- It includes programming support of deep neural networks and machine learningtechniques.

- It includes a highly scalable feature of computation with various data sets.

- TensorFlow uses GPU computing, automating management. It also includes a uniquefeature of optimization of the same memory and the data used.

FEATURES OF TENSORFLOW:

- Auto Differentiation
- Eager execution.
- Distribute
- Los

5.SYSTEM DESIGN

5.1. SYSTEM ARCHITECTURE/BLOCK DIAGRAM

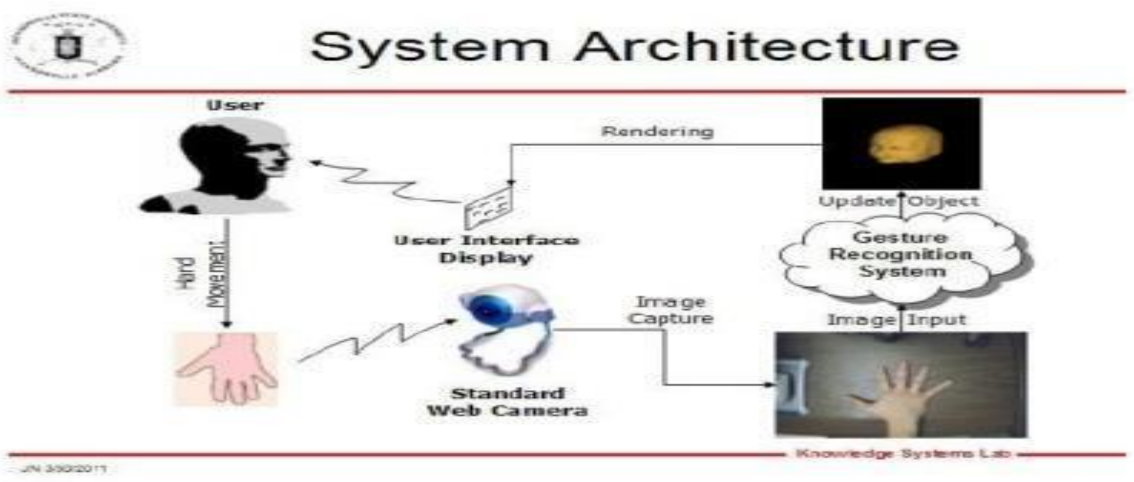


Fig:1 Hand Gesture detection

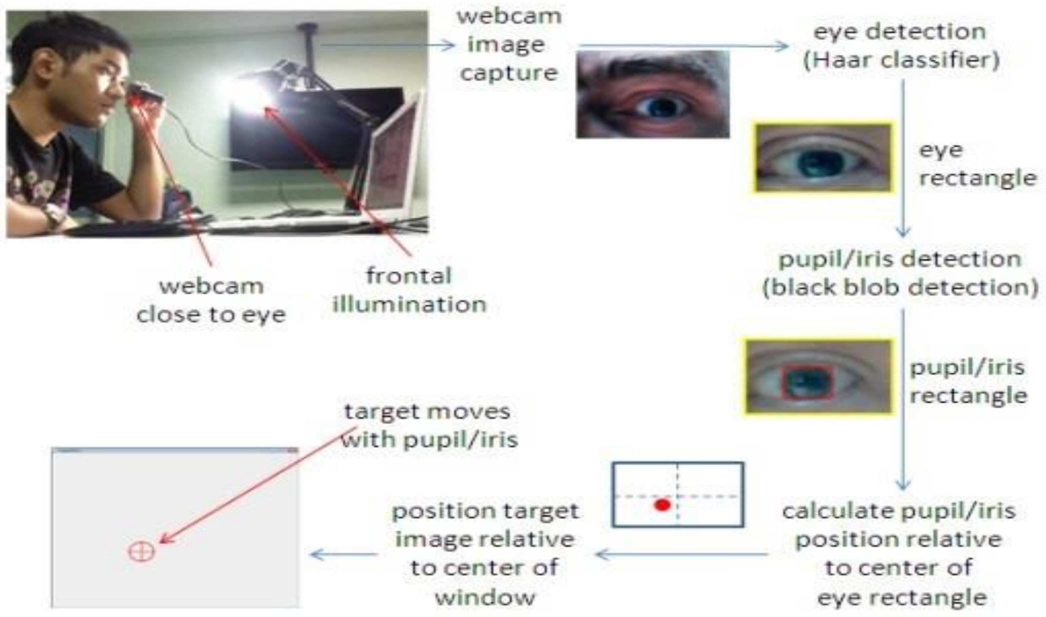

webcam image capture
eye detection (Haar classifier)
eye rectangle
pupil/iris detection (black blob detection)
pupil/iris rectangle
webcam close to eye
frontal illumination
target moves with pupil/iris
position target image relative to center of window
calculate pupil/iris position relative to center of eye rectangle

Fig:2 Eye motion tracker

5.2. DIAGRAMS APPLICABLE

UML DIAGRAMS:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed,and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML comprises two major components: a Meta-model anda notation. In the future, some form of method or process may also be added to; or associated with,UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructingand documenting the artifacts of software systems, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successfulin the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the softwaredevelopment process. The UML uses mostly graphical notations to express the design of softwareprojects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they candevelop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of the OO tools market.
6. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

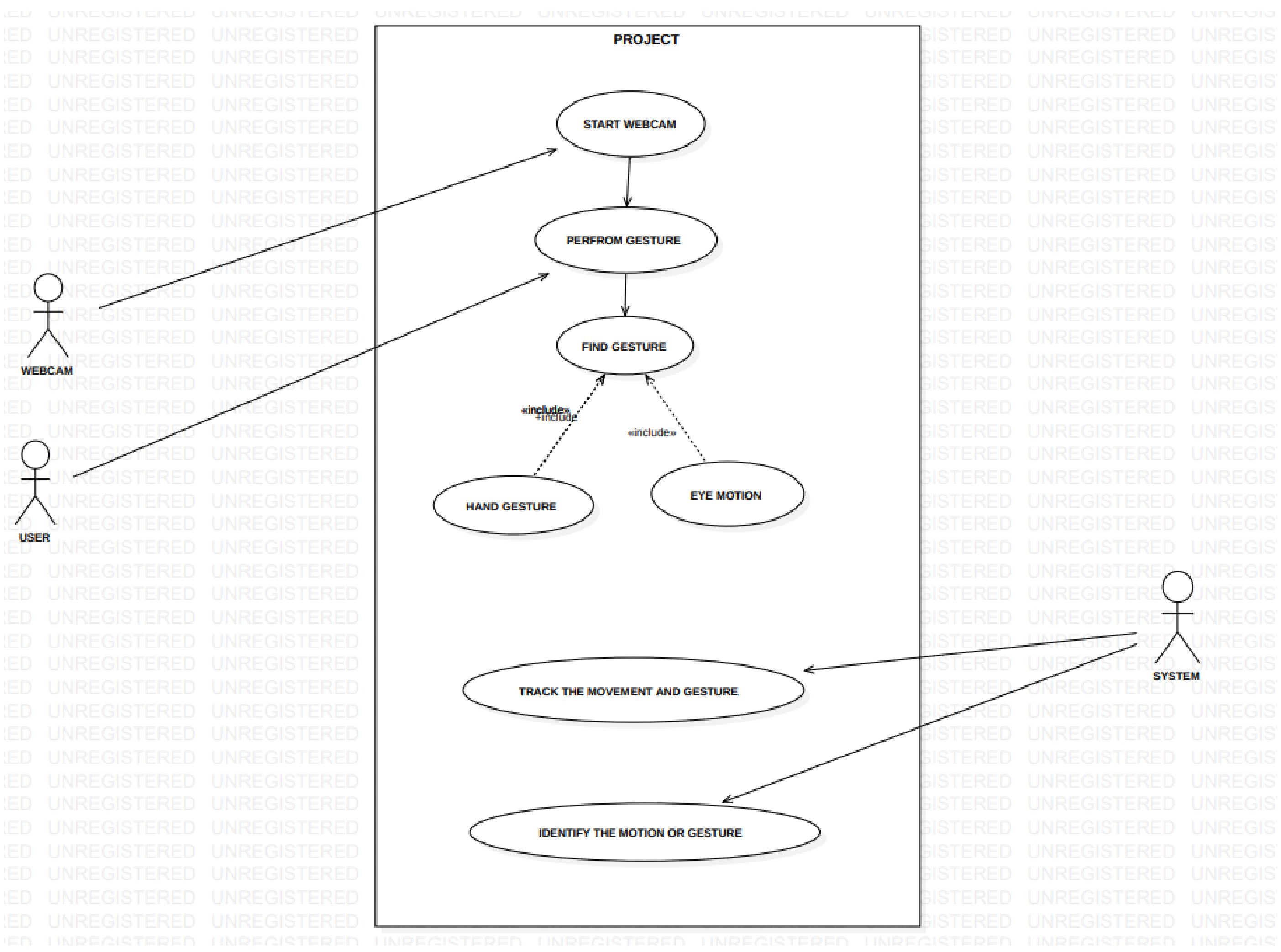


Fig.1.Use Case Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains whichclass contains information.

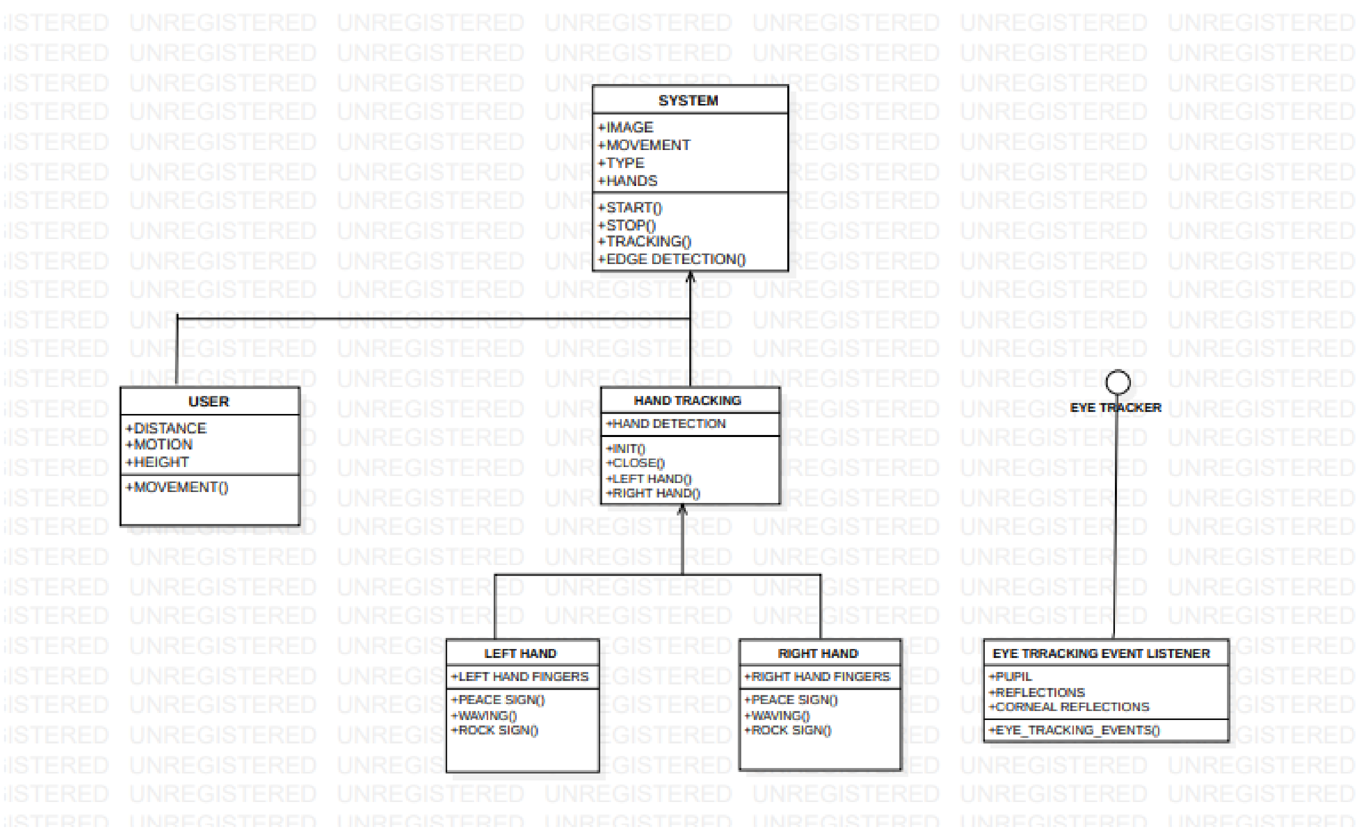


Fig 2: Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram thatshows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

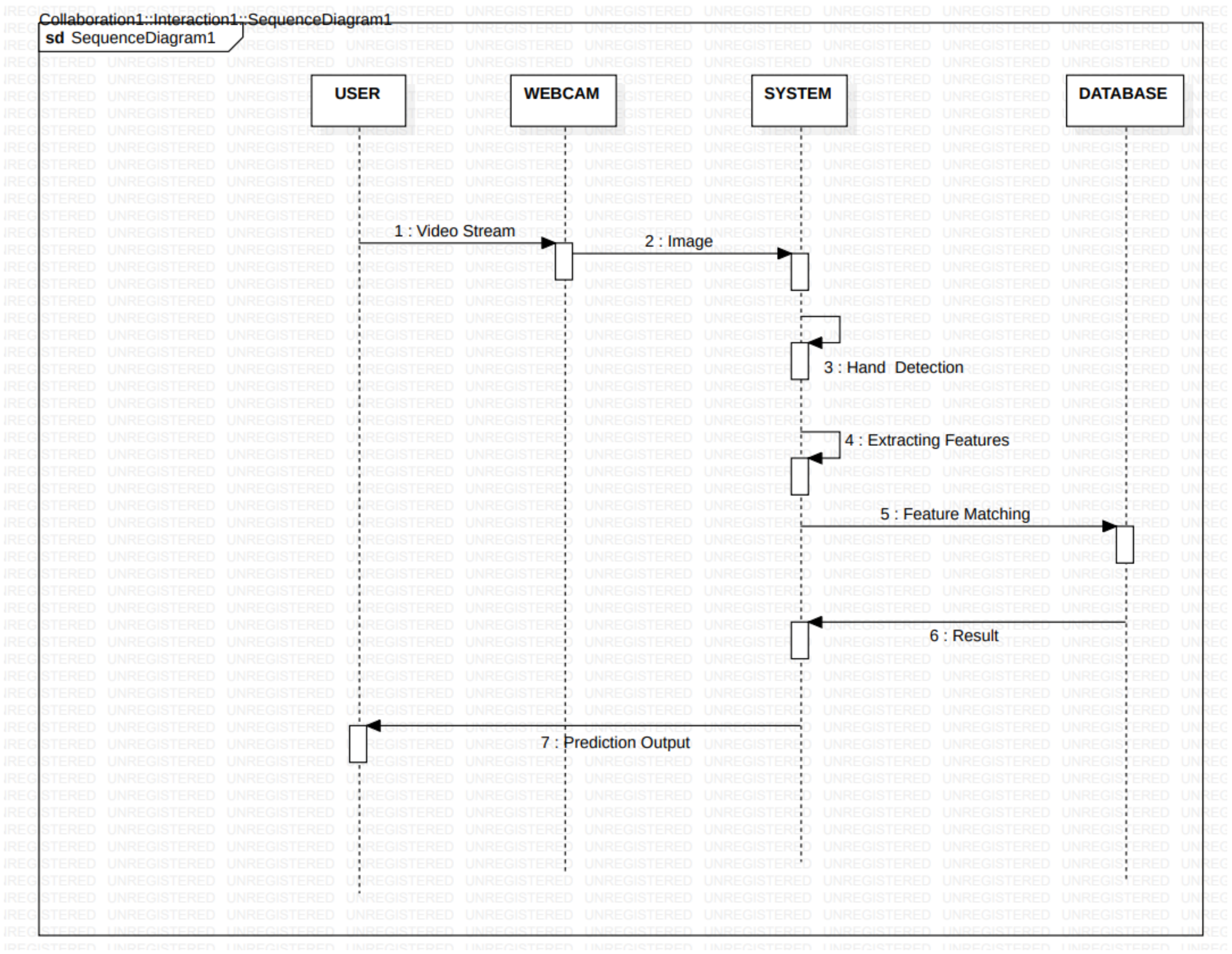


Fig 3: Sequence Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

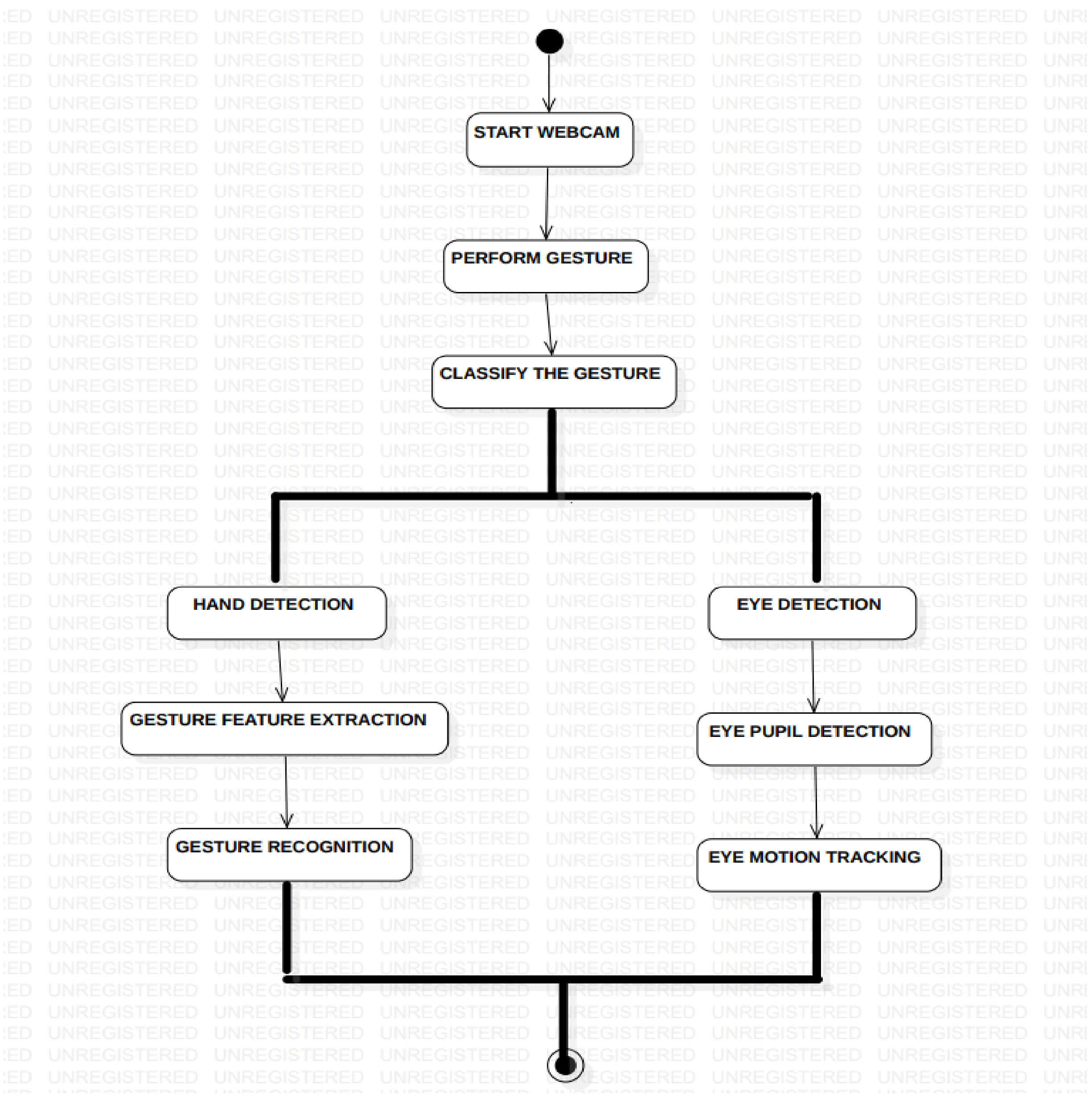


Fig 4: Activity Diagram

6.IMPLEMENTATION

6.1. ENVIRONMENTAL SETUP

1) Install Python for windows 64it or Mac using the below link:
    - https://www.python.org/downloads/

2) Install VS Code using the below link:
https://code.visualstudio.com/download

3) Install OPENCV for Python using the pip command:
```
pip install opencv-python
```

4) Install Media Pipe for Python using the pip command:
```
pip install mediapipe
```

5) Install Tensor Flow for Python using the pip command:
```
pip install tensorflow
```

6) Install tf-nightly 2.5.0 dev or later for Python using the pip command:

7) Install scikit-learn 0.23.2 or later for Python using the pip command:
```
pip install -U scikit-learn
```

8) Install matplotlib 3.3.2 or later for Python using the pip command:
```
pip install matplotlib
```

9) Install Cmake for Python using the pip command:
```
pip install cmake
```

10) Install Dlib for Python using the pip command:
```
pip install dlib
```

STEPS:

- Install Python in your system and set up the path.
- Using the IDE set up an virtual environment (VENV) . Create Python folder in C directory and put 'HAND_GESTURE_RECOGNITION and EYE_MOTION TRACKER' folder in it.

- Run the app.py File to check for any errors, if any errors then resolve those errors and re-run the file again.
- Execute the app.py file for the Hand_Gesture_recognition.

6.2 MODULE DESCRIPTION:

Here we have 2 modules in this project,

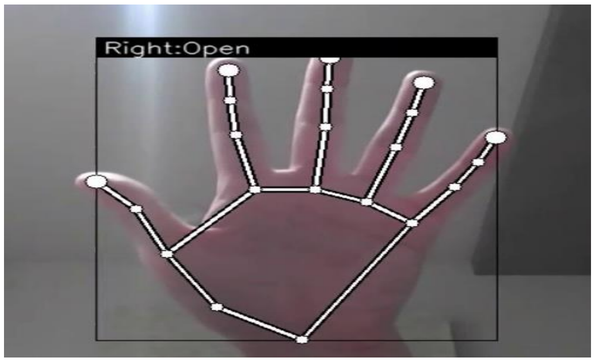
1. USER

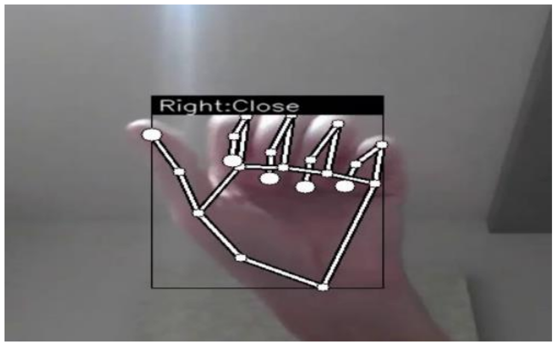
Here users give the hand gestures as the input for detection and recognition. Not only hand gesture but also the eye motion tracking can be done.
Based on the hand gesture sign given, the Mediapipe draws out the landmark classification and displays the name of the sign.
Based upon the motion of the pupil we detect the direction at which the pupil looks at, like {LEFT, CENTER, RIGHT} and it also counts the number of blinks.

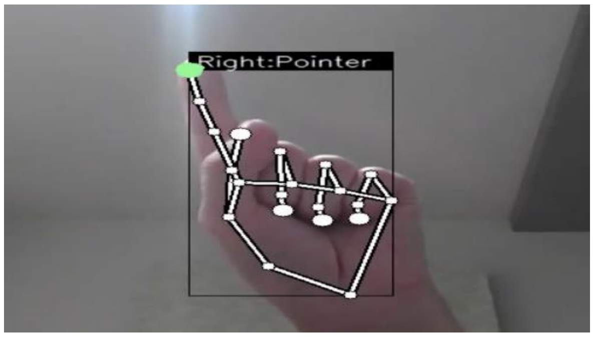
2.SYSTEM

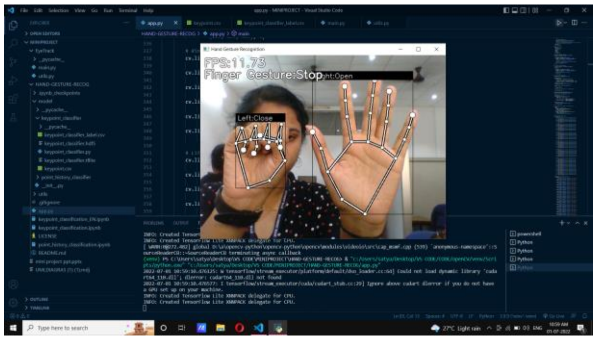
System does all the background work of pre-processing the data, Training the data and testing the data.

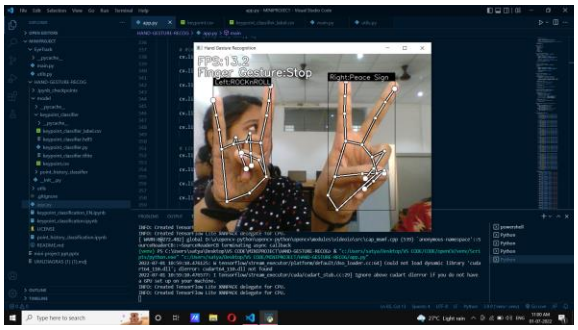
7. RESULTS

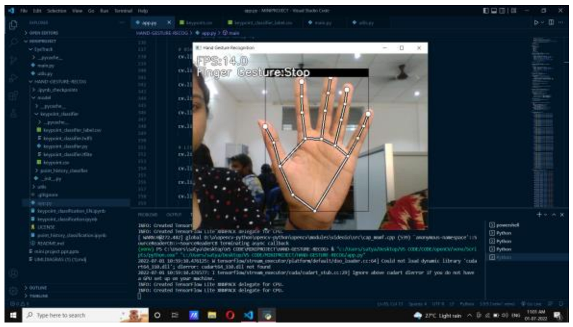

Right:Open

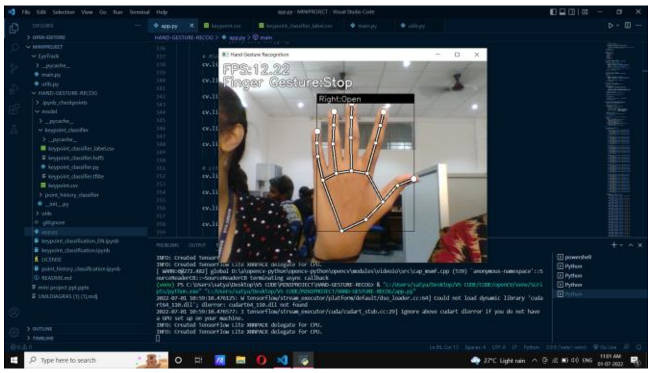

Right:Close

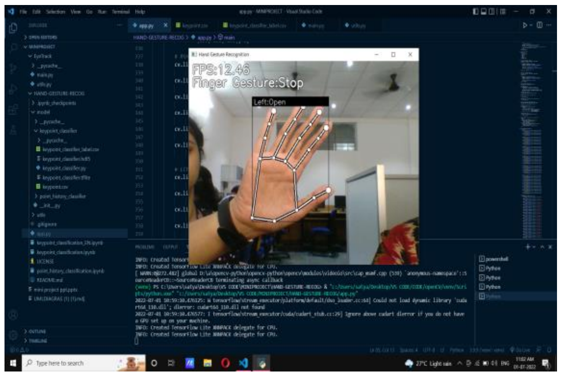

Right:Pointer

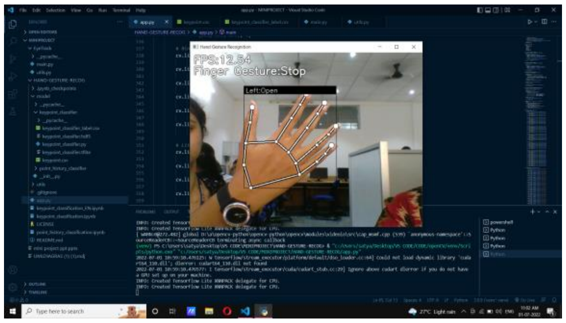

FPS:11.73
Finger Gesture:Stop
Right:Open
Left:Close

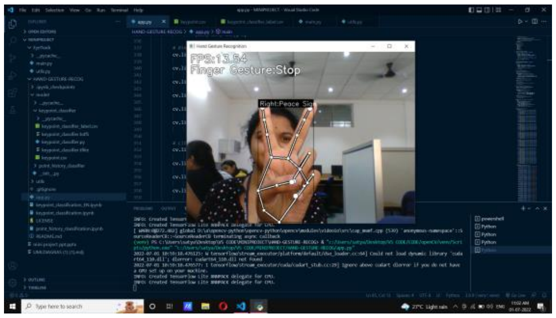

FPS:13.2
Finger Gesture:Stop
Left:ROCKnROLL
Right:Peace Sign

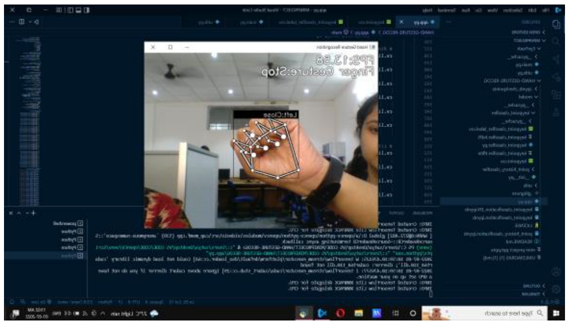

FPS:14.0
Finger Gesture:Stop

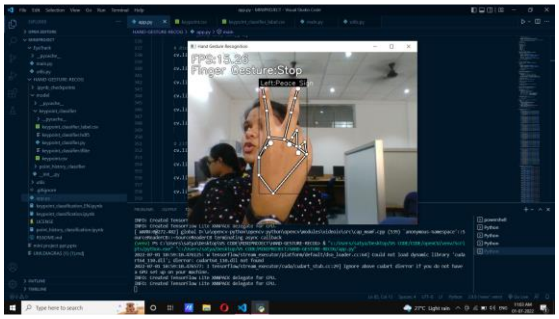

FPS:12.22
Finger Gesture:Stop
Right:Open

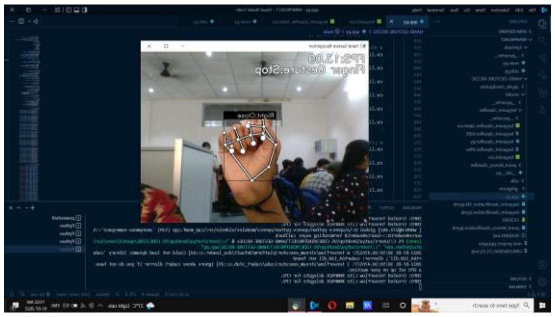

FPS:12.46
Finger Gesture:Stop
Left:Open

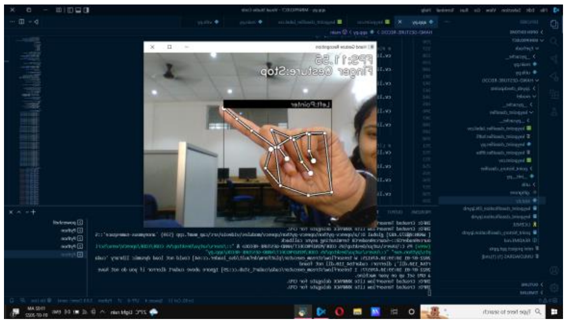

FPS:12.54
Finger Gesture:Stop
Left:Open

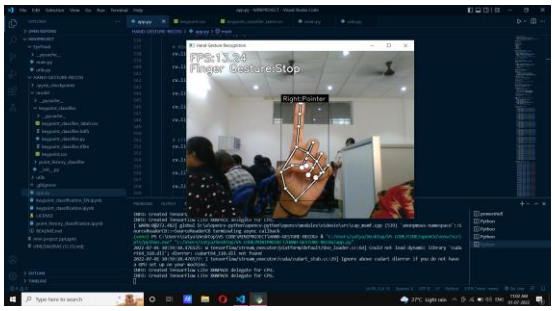

FPS:13.54
Finger Gesture:Stop
Right:Peace Si

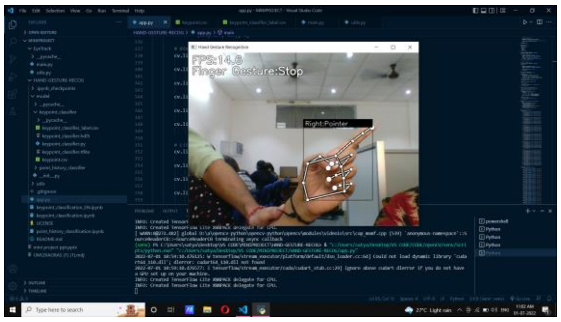

FPS:13.28
Finger Gesture:Stop
Left:Close

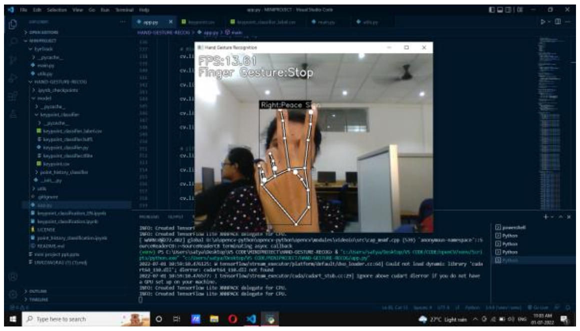

FPS:15.28
Finger Gesture:Stop
Left:Peace Si

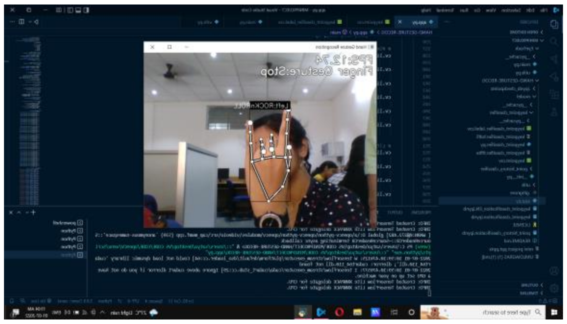

FPS:13.86
Finger Gesture:Stop
Right:Close

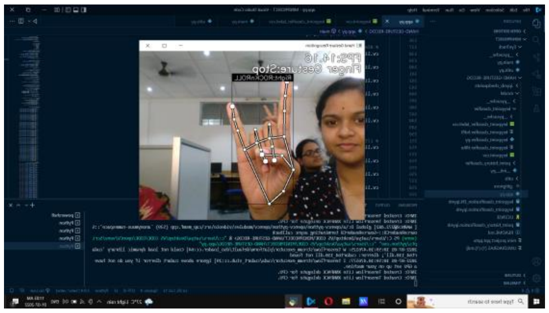

FPS:11.65
Finger Gesture:Stop
Left:Pointer

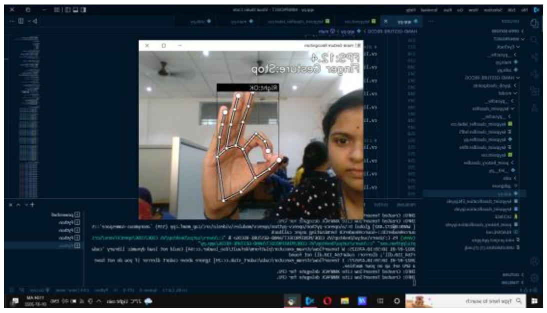

FPS:13.24
Finger Gesture:Stop
Right:Pointer

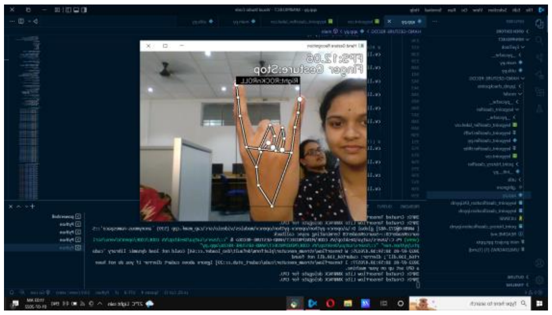

FPS:14.8
Finger Gesture:Stop
Right:Pointer

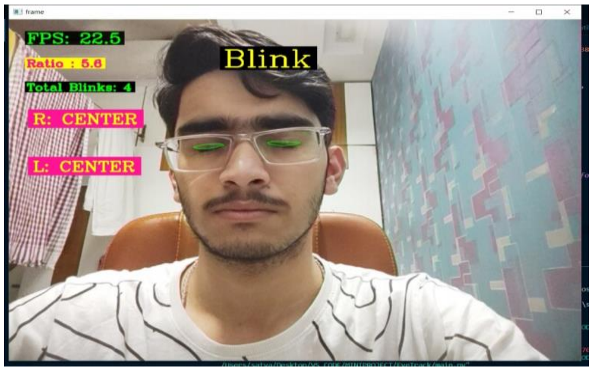

Hand Gesture Recognition
FPS:13.01
Finger Gesture:Stop

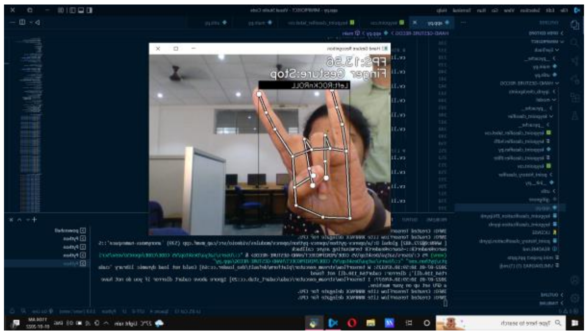

FPS:12.74
Finger Gesture:Stop

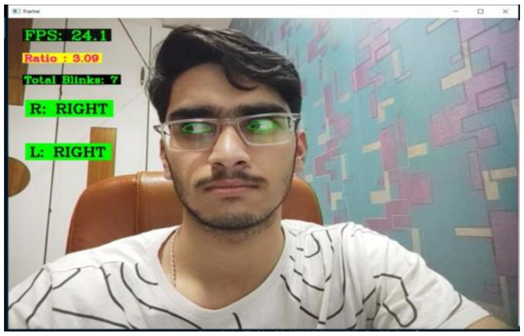

FPS:14.16
Finger Gesture:Stop
Right:ROCKnROLL

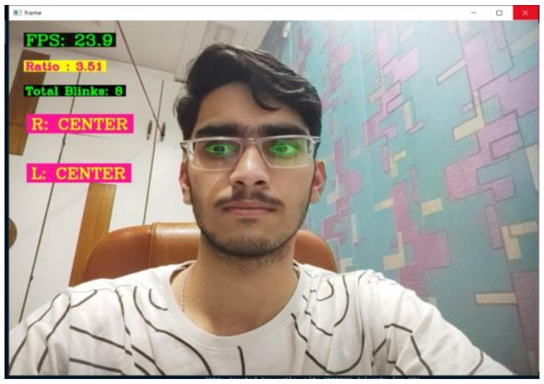

FPS:12.4
Finger Gesture:Stop
Right:OK

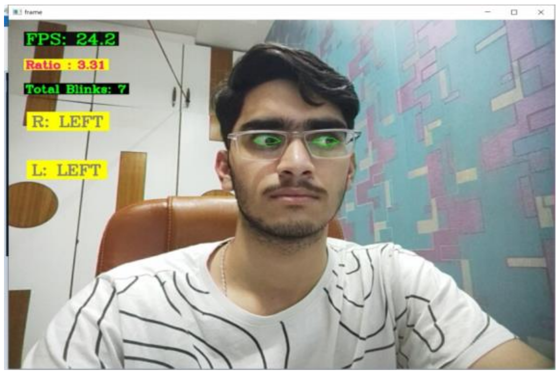

FPS:12.06
Finger Gesture:Stop
Right:ROCKnROLL


frame
FPS: 22.5
Blink
Ratio : 5.6
Total Blinks: 4
R: CENTER
L: CENTER


FPS:13.56
Finger Gesture:Stop
Left:ROCKnROLL


frame
FPS: 24.1
Ratio : 3.09
Total Blinks: 7
R: RIGHT
L: RIGHT



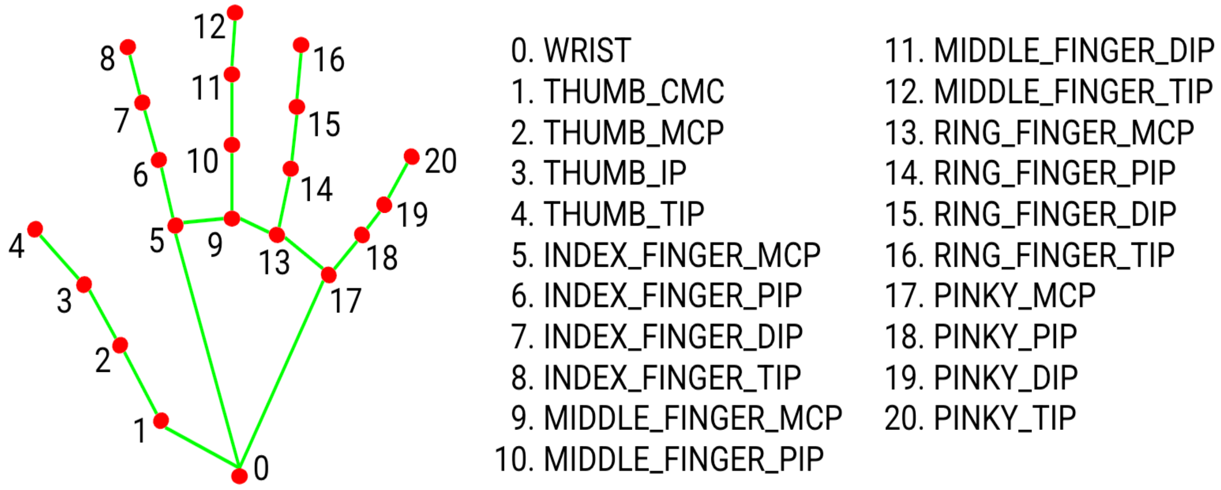


Hand sign recognition training

1.Learning data collection

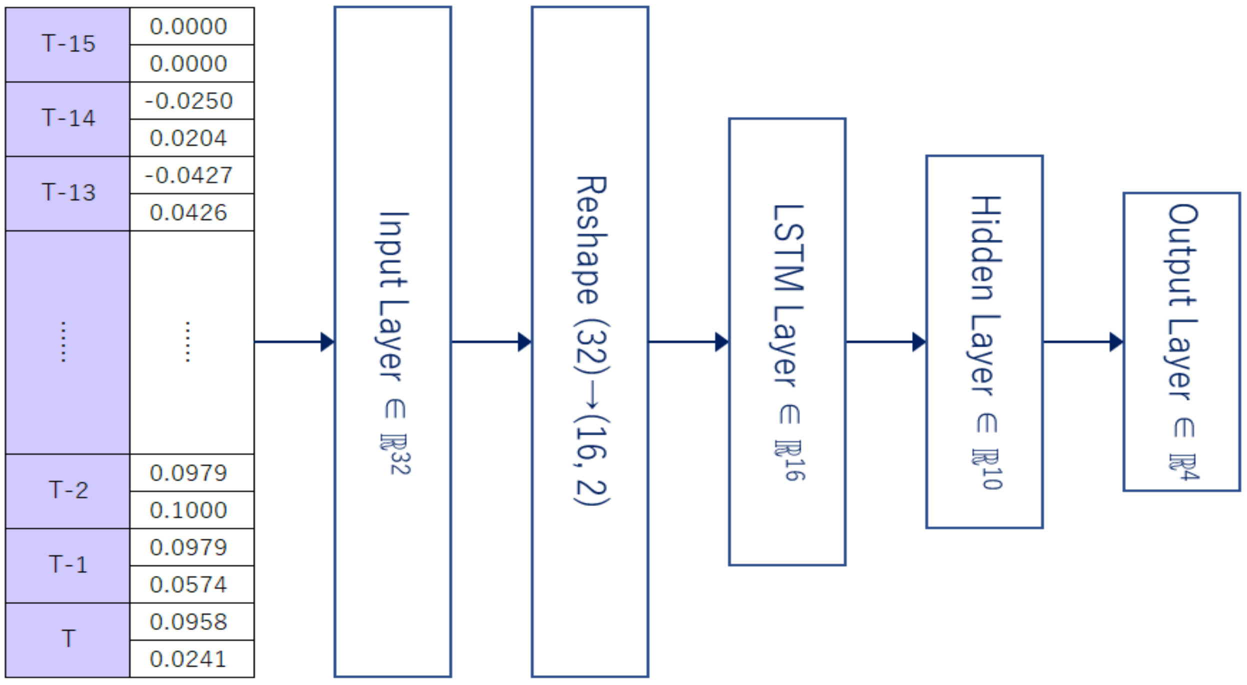
Press "k" to enter the mode to save key points displayed as [MODE: Logging Key Point] ）.

If you press "0" to "9", the key points will be added to "***model/keypoint_classifier/keypoint.csv***" as shown below.
1st column: Pressed number (used as class ID), 2nd and subsequent columns: Key point coordinates.



2.Model training

Open "keypoint_classification.ipynb" in Jupyter Notebook and execute from top to bottom. To change the number of training data classes, change the value of "NUM_CLASSES = 3" and modify the label of "model/keypoint_classifier/keypoint_classifier_label.csv" as appropriate.



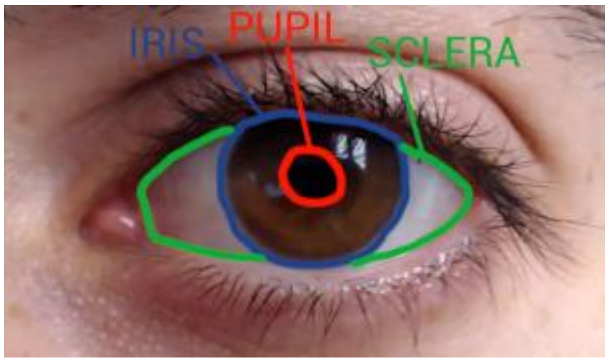
EYE MOTION Training:

Before getting into details about image processing, let’s study a bit the eye and let’s think what are the possible solutions to do this.

In the picture below we see an eye. The eye is composed of three main parts:

- Pupil – the black circle in the middle
- Iris – the bigger circle that can have different color for different people

•Sclera – it's always white



10. TESTING

10.1 UNIT TESTING

The unit test focused on the internal processing logic. All statements in a module have been exercised at least once. The interface module was tested to ensure that information properly flowed into and out of the program unit under test.

10.2 INTEGRATION TESTING

Integration testing is a technique for constructing the software architecture and conducting tests to uncover errors with interface. The objective of testing was to crosscheck for components fully functional or not according to design. Thus, I integrated all my unit components and saw if the system worked as a whole properly or not. The information flows between the components were checked once again.

10.3 RECOVERY TESTING

System fails in many ways but recovery must be properly performed. For example when a person trains the system with his gestures and the system fails to do so, it gives an error message indicating that the system was not properly trained and it keeps on doing so until it gets a valid gesture (that's acceptable to the system for differentiation). When the system is properly trained, only then we can expect that it will give us accurate results.

10.4 SENSITIVITY TESTING

Invalid input classes that may cause instability or improper processing. It was found During sensitivity testing that if the system was once fully trained for all the gesture types, it gave accurate results, otherwise if it were just trained for a single or two gestures and then tested, it performs erroneous processing

Test Results: All the test cases mentioned above passed successfully. No defects were encountered.

9.2 Test Cases

Comparative table

Histogram of Oriented Gradients	**94**
Sift features	**98.75**
CNN	**99.6**

10. **CONCLUSION**

Various types of hand gesture recognition methods and its application in automatic sign language recognition is highlighted. Various blocks of hand gesture recognition system are discussed with related research works. As the hand gesture recognition is suffered from background conditions, a few glimpses about relevant works have been considered. A few serious problems like movement epenthesis and coarticulation which prevent from proper continuous sign language recognition are also covered. It is seen that although glove-based hand gesture recognition techniques are accurate, vision-based techniques are preferable because of the naturalness of the hand movement and ease of operation.

FUTURE ENHANCEMENTS

In this project we have added signs and gestures which help the disabled people to communicate easier and the training model has been accurate and it can train new signs .

REFERENCES

- https://www.leadingindia.ai/downloads/projects/VP/vp_9.pdf
- https://github.com/kinivi/hand-gesture-recognition-mediapipe
- https://github.com/dabhisharad13/Hand-gesture-recognition-using-OpenCv-and-Cnn

APPENDIX

CODE:

app.py
This is a sample program for inference.
In addition, learning data (key points) for hand sign recognition,
You can also collect training data (index finger coordinate history) for finger gesture recognition.

keypoint_classification.ipynb

This is a model training script for hand sign recognition.

point_history_classification.ipynb

This is a model training script for finger gesture recognition.

model/keypoint_classifier

This directory stores files related to hand sign recognition.
The following files are stored.

- Training data(keypoint.csv)
- Trained model(keypoint_classifier.tflite)
- Label data(keypoint_classifier_label.csv)
- Inference module(keypoint_classifier.py)

model/point_history_classifier

This directory stores files related to finger gesture recognition.
The following files are stored.

- Training data(point_history.csv)
- Trained model(point_history_classifier.tflite)
- Label data(point_history_classifier_label.csv)
- Inference module(point_history_classifier.py)

utils/cvfpscalc.py

This is a module for FPS measurement.

HAND GESTURE DETECTION AND RECOGNITION:

PreProcessing of landmark classification list:

```
pre_process_landmark(landmark_list):
    temp_landmark_list = copy.deepcopy(landmark_list)
    # Convert to relative coordinates
    base_x, base_y = 0, 0
    for index, landmark_point in enumerate(temp_landmark_list):
        if index == 0:
            base_x, base_y = landmark_point[0], landmark_point[1]
        temp_landmark_list[index][0] = temp_landmark_list[index][0] - base_x
        temp_landmark_list[index][1] = temp_landmark_list[index][1] - base_y

    # Convert to a one-dimensional list
    temp_landmark_list = list(
        itertools.chain.from_iterable(temp_landmark_list))

    # Normalization
    max_value = max(list(map(abs, temp_landmark_list)))

    def normalize_(n):
        return n / max_value
    temp_landmark_list = list(map(normalize_, temp_landmark_list))
    return temp_landmark_list
```

Key Point Classifier:

```
KeyPointClassifier(object):
    def __init__(
        self,
        model_path='model/keypoint_classifier/keypoint_classifier.tflite',
        num_threads=1,
    ):
        self.interpreter = tf.lite.Interpreter(model_path=model_path,
                                               num_threads=num_threads)

        self.interpreter.allocate_tensors()
        self.input_details = self.interpreter.get_input_details()
        self.output_details = self.interpreter.get_output_details()
    def __call__(
        self,
        landmark_list, ):
        input_details_tensor_index = self.input_details[0]['index']
        self.interpreter.set_tensor(
            input_details_tensor_index,
            np.array([landmark_list], dtype=np.float32))
        self.interpreter.invoke()
        output_details_tensor_index = self.output_details[0]['index']
        result = self.interpreter.get_tensor(output_details_tensor_index)
        result_index = np.argmax(np.squeeze(result))
        return result_index
```

EYE MOTION TRACKER :

Landmark Detection and Classification:

```
    def landmarksDetection(img, results, draw=False):
    img_height, img_width= img.shape[:2]
    # list[(x,y), (x,y)....]
    mesh_coord = [(int(point.x * img_width), int(point.y * img_height)) for point in
results.multi_face_landmarks[0].landmark]
    if draw :
    [cv.circle(img, p, 2, (0,255,0), -1) for p in mesh_coord]

    # returning the list of tuples for each landmarks
    return mesh_coord
```

Getting the Position of the Eye:

```
    def positionEstimator(cropped_eye):
        # getting height and width of eye
        h, w =cropped_eye.shape

        # remove the noise from images
        gaussain_blur = cv.GaussianBlur(cropped_eye, (9,9),0)
        median_blur = cv.medianBlur(gaussain_blur, 3)

        # applying thrsholding to convert binary_image
        ret, threshed_eye = cv.threshold(median_blur, 130, 255,
cv.THRESH_BINARY)

        # create fixd part for eye with
        piece = int(w/3)

        # slicing the eyes into three parts
        right_piece = threshed_eye[0:h, 0:piece]
        center_piece = threshed_eye[0:h, piece: piece+piece]
        left_piece = threshed_eye[0:h, piece +piece:w]

        # calling pixel counter function
        eye_position, color = pixelCounter(right_piece, center_piece, left_piece)

        return eye_position, color

    # creating pixel counter function
    def pixelCounter(first_piece, second_piece, third_piece):
        # counting black pixel in each part
        right_part = np.sum(first_piece==0)
        center_part = np.sum(second_piece==0)
        left_part = np.sum(third_piece==0)
        # creating list of these values
        eye_parts = [right_part, center_part, left_part]

        # getting the index of max values in the list
        max_index = eye_parts.index(max(eye_parts))
        pos_eye =''
        if max_index==0:
            pos_eye="RIGHT"
            color=[utils.BLACK, utils.GREEN]
```

```
elif max_index==1:
    pos_eye = 'CENTER'
    color = [utils.YELLOW, utils.PINK]
elif max_index ==2:
    pos_eye = 'LEFT'
    color = [utils.GRAY, utils.YELLOW]
else:
    pos_eye="Closed"
    color = [utils.GRAY, utils.YELLOW]
return pos_eye, color
```

CO-PO/PSO MAPPING

Course code	**Statement** Student will be able to	**Cognitive Level**	**PO / PSO addressed**
PW961.1	Define a problem of the recent advancements with applications towards society.	An	PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PSO1, PSO2
PW961.2	Outline requirements and perform requirement analysis for solving the problem.	An	PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PSO1, PSO2
PW961.3	Design and develop a software and/or hardware-based solution within the scope of project using contemporary technologies and tools.	AP, E, An	PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PSO1, PSO2
PW961.4	Test and deploy the applications for use.	AP, E, An	PO8, PO9, PO10, PO11, PO12, PSO1, PSO2
PW961.5	Develop the Project as a team and demonstrate the application, with effective written and oral communications.	C	PO8, PO9, PO10, PO11, PO12, PSO1, PSO2

Table 1: Course Outcomes - Cognitive levels
Cognitive Levels: R-Remember; U-Understand; Ap-Apply; An=Analyze; E-Evluate; C-Create

Table 2: Number of performance indicators addressed by course outcomes

Course Code	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
No. of PIs addressed by course for a given PO	**4**	4	**4**	**4**	**4**	**4**	**4**	**5**	**5**	**7**	**4**	**6**	**5**	**6**
CO1	2	2	2	1	3	1	1	1	3	3	1	2	1	3
CO2	3	2	3	2	3	1	1	1	3	3	3	2	2	3
CO3	3	3	3	2	3	1	1	3	3	3	3	2	2	3
CO4								3	3	3	3	2	2	2
CO5								1	3	3	3	2	2	3

Table 3: Calculation of CO-PO/PSO correlation levels

	PO1		**PO2**		**PO3**		**PO4**		**PO5**		**PO6**		**PO7**		**PO8**		**PO9**		**PO10**		**PO11**		**PO12**		**PSO1**		**PSO2**	
PW961IT	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve	%	Leve
CO1	50	3	50	2	50	2	25	1	50	3	25	1	25	1	20	1	50	3	50	3	25	1	30	1	20	1	50	3
CO2	75	3	50	2	75	3	50	2	50	3	25	1	25	1	20	1	50	3	50	3	75	3	30	1	50	3	50	3
CO3	75	3	75	3	75	3	50	2	50	3	25	1	25	1	50	3	50	3	75	3	75	3	30	1	50	3	50	3
CO4	50														50	3	50	3	75	3	75	3	30	1	20	1	50	3
CO5	50														20	1	50	3	50	3	75	3	30	1	40	2	50	3
No. Mapped	3	9	3	7	3	11	3	8	3	9	3	3	3	3	5	9	5	15	5	15	5	13	5	5	5	10	5	15
Average of Level	9/3=3		7/3=2.3		8/3=2.6		8/3=2.6		9/3=3		3/3=1		3/3=1		9/5=1.8		15/5=3		15/5=3		13/5=2.6		5/5=1		10/5=2		15/5=3	
Rounded average level	**3**		**2**		**3**		**3**		**3**		**1**		**1**		**2**		**3**		**3**		**3**		**1**		**2**		**3**	

Table 4: Course Articulation Matrix

PW961IT	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	2	2	1	3	1	1	1	3	3	1	1	1	3
CO2	3	2	3	2	3	1	1	1	3	3	3	1	3	3
CO3	3	3	3	2	3	1	1	3	3	3	3	1	3	3
CO4								3	3	3	3	1	1	3
CO5								1	3	3	3	1	2	3
PW961IT	**3**	**2**	**3**	**3**	**3**	**1**	**1**	**2**	**3**	**3**	**3**	**1**	**2**	**3**