PROJECT REPORT

on

AI VIRTUAL KEYBOARD AND MOUSE USING MACHINE LEARNING

submitted in partial fulfilment for the award of the Degree of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

By

G JYOTHSNA

R170826

Under the Guidance of

Ms. P UdayaSree

Assistant Professor

Department Of Computer Science and Engineering,

IIIT RK Valley, RGUKT-AP.

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(Catering the Educational Needs of Gifted Rural Youth of A.P)

RK Valley Kadapa (Dist.)-516330

Andhra Pradesh

DECLARATION

I declare that this report of the B.Tech Major Project Work entitled with the "**AI Virtual Keyboard and Mouse Using Machine Learning"** which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by me. The material contained in this report has not been submitted to any university or institution for award of any degree.

Date: **G JYOTHSNA**

Place: RK VALLEY R170826

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P. Government Act 18 of 2008)
RGUKT-RK Valley, A.P.
RK Valley (Idupulapaya), Vempalli(M), Y.S.R Kadapa (Dist.) A.P–516330

CERTIFICATE

This is certified that this project entitled "**AI Virtual Keyboard and Mouse Using Machine Learning**" submitted by the **G Jyothsna (R170826)** under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the academic year 2022-2023 at RGUKT, RK VALLEY. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

Project Internal Guide

Ms. P UdayaSree
Assistant Professor
Computer Science and Engineering

Head of the Department

Mr. N Satyanandaram
Lecturer
Computer Science and Engineering

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director, Prof. **Mrs. K.SANDHYA RANI** for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department **Mr. N SATYANANDARAM** for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college **Ms. P.UDAYASREE** for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics. My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

ABSTRACT

Nowadays computer vision has reached its pinnacle, where a computer can identify its owner using a simple program of image processing. In this stage of development, people are using this vision in many aspects of day-to-day life, like Face Recognition, Colour detection, Automatic car, etc. In this project, Python programming language is used for developing the AI virtual mouse system, and also, OpenCV which is the library for computer vision is used in the AI virtual mouse system. In this virtual mouse system, the model makes use of the MediaPipe package for the tracking of the hands for both keyboard operation and mouse operations like keyboard button click and for mouse functions such as mouse movement, left click, right click, drag and drop, double click, scrolling up and down using mouse python package. A virtual keyboard is software that is used to emulate a standard keyboard. A picture of a keyboard is displayed on a computer screen and the user points and clicks on the pictures of keys to enter text. Switches activated in a wide variety of ways make use of the most appropriate muscles for the individual user. Virtual keyboards allow computer use by people with significant mobility impairments. Some virtual keyboards incorporate word prediction to increase entry speed. AI virtual mouse system that makes use of the hand gestures and hand tip detection for performing mouse functions in the computer using computer vision. The main objective of the proposed system is to perform computer mouse cursor functions and scroll function using a web camera or a built-in camera in the computer instead of using a traditional mouse device.

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Introduction to Virtual Keyboard and Mouse

Human-computer interaction (HCI) is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. The construction of human interfaces is both a matter of design and engineering. These topics are concerned with the methodology and practice of interface design. Other aspects of the development process include the relationship of interface development to the engineering (both software and hardware) of the rest of the system. One of the HCI for future development is to find a new way to interact with computer. Some of new interaction has been a dream for better and revolutionary way that even used in many futuristic movies. In this interaction, the input device as interaction way has been replace to a non-physical, unknown device.

Figure 1.1: Touch screen based virtual Keyboard

Figure 1.2: Optical sensor based virtual keyboard

Python programming language is used for developing the AI virtual mouse system, and also, OpenCV which is the library for computer vision is used in the AI virtual mouse system. In this virtual mouse system, the model makes use of the MediaPipe package for the tracking of the hands for both keyboard operation and mouse operations like keyboard button click and for mouse functions such as mouse movement, left click, right click, drag and drop, double click, scrolling up and down using mouse python package.

A virtual keyboard is software that is used to emulate a standard keyboard. A picture of a keyboard is displayed on a computer screen and the user points and clicks on the pictures of keys to enter text. Switches activated in a wide variety of ways make use of the most appropriate muscles for the individual user. Virtual keyboards allow computer use by people with significant mobility impairments. Some virtual keyboards incorporate word prediction to increase entry speed.

AI virtual mouse system that makes use of the hand gestures and hand tip detection for performing mouse functions in the computer using computer vision. The main objective of the proposed system is to perform computer mouse cursor functions and scroll function using a web camera or a built-in camera in the computer instead of using a traditional mouse device. With the use of the AI virtual mouse system, this model can track the fingertip of the hand gesture by using a built-in camera or web camera and perform the mouse cursor operations, scrolling function. In this system user uses his/her built-in camera or a webcam and uses his/her hand gestures to control the computer mouse operations. In the proposed system, the web camera captures and then processes the frames that have been captured and then recognizes the various hand gestures and hand tip gestures and then performs the particular mouse function.

Figure 1.3: Sensor based Virtual Mouse

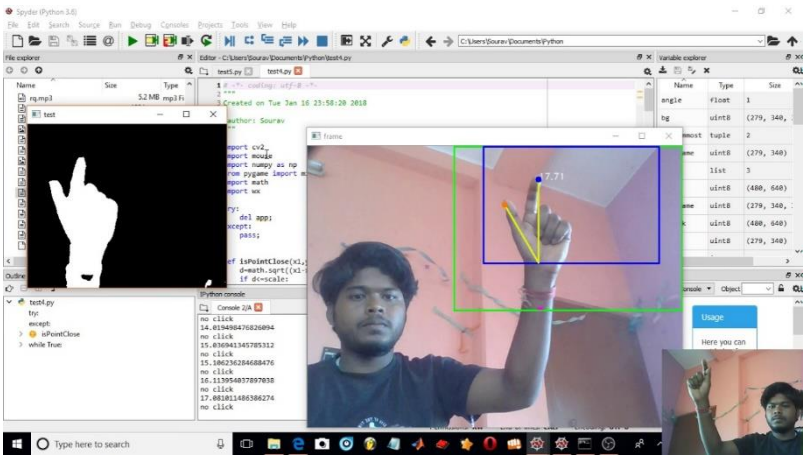


Figure 1.4: HGR based Virtual mouse

1.2 MACHINE LEARNING

Machine learning is a discipline that deals with programming the system so as to make them automatically learn and improve with experience. Here, learning implies recognizing and understanding the input data and taking informed decisions based on the supplied data. It is very difficult to consider all the decisions based on all possible inputs. To solve this problem, algorithms

are developed that build knowledge from a specific data and past experience by applying the principles of statistical science, probability, logic, mathematical optimization, reinforcement learning, and control theory.

1.2.1 TYPES OF MACHINE LEARNING

Machine learning is available all around the world and all are experiencing any one of the occurrences every day. A new method of data is arising from using machine learning. As there are few complexities seen in machine learning it is divided in to two main two main areas called supervised and unsupervised learning Each type has its own advantages and different working process.70% of machine learning is said to be supervised learning and can expect only 10-20% as unsupervised learning.

1.2.1.1 Supervised learning

Supervised learning is an approach to creating artificial intelligence (AI), where a computer algorithm is trained on input data that has been labeled for a particular output. The model is trained until it can detect the underlying patterns and relationships between the input data and the output labels, enabling it to yield accurate labeling results when presented with never-before-seen data**.** Supervised learning is good at classification and regression problems, such as determining what category a news article belongs to or predicting the volume of sales for a given future date. In supervised learning, the aim is to make sense of data within the context of a specific question.

1.2.1.2 Unsupervised learning

Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision. Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, they have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

1.2.1.3 Reinforcement learning

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer

but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

1.2.2 PHASES OF MACHINE LEARNING

The task of imparting intelligence to machines seems daunting and impossible. But it is actually really easy. It can be broken down into 7 major steps:

1.2.2.1 Collecting Data

As you know, machines initially learn from the data that you give them. It is of the utmost importance to collect reliable data so that your machine learning model can find the correct patterns. The quality of the data that you feed to the machine will determine how accurate your model is. If you have incorrect or outdated data, you will have wrong outcomes or predictions which are not relevant. Make sure you use data from a reliable source, as it will directly affect the outcome of your model. Good data is relevant, contains very few missing and repeated values, and has a good representation of the various subcategories/classes present.

1.2.2.2 Preparing the Data

After you have your data, you have to prepare it. You can do this by :

- Putting together all the data you have and randomizing it. This helps make sure that data is evenly distributed, and the ordering does not affect the learning process.
- Cleaning the data to remove unwanted data, missing values, rows, and columns, duplicate values, data type conversion, etc. You might even have to restructure the dataset and change the rows and columns or index of rows and columns.
- Visualize the data to understand how it is structured and understand the relationship between various variables and classes present.

1.2.2.3 Choosing a Model

A machine learning model determines the output you get after running a machine learning algorithm on the collected data. It is important to choose a model which is relevant to the task at hand. Over the years, scientists and engineers developed various models suited for different tasks like speech recognition, image recognition, prediction, etc. Apart from this, you also have to see if your model is suited for numerical or categorical data and choose accordingly.

1.2.2.4 Training the Model

Training is the most important step in machine learning. In training, you pass the prepared data to your machine learning model to find patterns and make predictions. It results in the model learning from the data so that it can accomplish the task set. Over time, with training, the model gets better at predicting.

1.2.2.5 Evaluating the Model

After training your model, you have to check to see how it's performing. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that you split our data into earlier. If testing was done on the same data which is used for training, you will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give you disproportionately high accuracy.

1.2.2.6 Parameter Tuning

Once you have created and evaluated your model, see if its accuracy can be improved in any way. This is done by tuning the parameters present in your model. Parameters are the variables in the model that the programmer generally decides. At a particular value of your parameter, the accuracy will be the maximum. Parameter tuning refers to finding these values.

1.2.2.7 Making Predictions

In the end, you can use your model on unseen data to make predictions accurately.

1.2.3 APPLICATIONS OF MACHINE LEARNING ALGORITHMS

- Image Recognition:
- Speech Recognition
- Traffic prediction:
- Product recommendations:
- Self-driving cars:
- Email Spam and Malware Filtering:
- Virtual Personal Assistant:
- Medical Diagnosis:
- Automatic Language Translation:

1.3 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's 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. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

1.3.1 INTRODUCTION TO PYTHON

Python is a popular programming language. It was created by Guido van Rossum and released in 1991. It is used for:

- ✓ Web development
- ✓ Software development
- ✓ Mathematics
- ✓ System scripting

Python was imagined in the late 1980s and its usage started in December 30, 1989 by Guido van Rossum at Centrum Wiskunde and Informatica (CWI) in the Netherlands as a successor to the ABC dialect (itself roused by SETL) capable of exemption dealing with and interfacing with the Amoeba working system. Van Rossum remains Python’s chief creator. His proceeding with focal part in Python’s advancement is reflected in the title given him by the Python people group.

1.3.2 WHAT CAN PYTHON DO

- ✓ Python can be used on a server to create web applications.
- ✓ Python can be used alongside software to create workflows.
- ✓ Python can connect to database system. It can also read and modify files.
- ✓ Python can be used to handle big data and perform complex mathematics.
- ✓ Python can be used for rapid prototyping, or for rapid prototyping, or for production-ready software development.

1.3.3 WHY PYTHON

- ✓ Python works on different platforms.
- ✓ Python has a simple syntax that allows developers to write programs with fewer lines than some other programming languages.
- ✓ Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

1.3.4 PYTHON FEATURE

- ✓ Simple and easy to learn Python as only 33 keywords but JAVA as 83 keywords.
- ✓ High level programming language. Python is platform independent.
- ✓ Both object oriented and procedure oriented language.
- ✓ Interpreted language (It means not going to compile)
- ✓ Extensible**,** Portability, moving from one platform to another without any change.
- ✓ Dynamically typed Programming Language. In python it is not required to declare type in python.

1.3.5 LIMITATIONS OF PYTHON

- Performance wise it is not up to the mark. Because its an interpreted language. Interpreter able to see only one line (JAVA is better performance compare to python in java JIT (just in compiler)).
- Mobile applications it is not up to the mark python is not suitable large scale enterprise applications.

1.3.6 FLAVORS OF PYTHON

- Cpython: It can be standard, it ca be used to c language python.
- Jpython: It is for JAVA application.
- Iron python: to work with Microsoft .net platform.
- Py: Internally JIT (just in time complex) compiler is there so performance wise too good.
- Ruby python: To handle Big data happily go for Anaconda python.
- Stackless (python for concurrency)
- Parallely you execute (like multithreaded) go for stackless.

1.3.7 APPLICATIONS OF PYTHON

Python is known for its general-purpose nature that makes it applicable in almost every domain of software development. Python makes its presence in every emerging field. It is the fastest-growing programming language and can develop any application.

- Web Applications
- Desktop GUI Applications
- Console-based Application
- Software Development
- Scientific and NumericSciPy
- Business Applications
- Image Processing Application
- 3D CAD Applications

1.3.8 PYTHON IN MACHINE LEARNING

Python is a popular platform used for research and development of production systems. It is a vast language with number of modules, packages and libraries that provides multiple ways of achieving a task.Python and its libraries like NumPy, SciPy, Scikit-Learn, Matplotlib are used in data science and data analysis. They are also extensively used for creating scalable machine learning algorithms. Python implements popular machine learning techniques such as Classification, Regression, Recommendation and Clustering.

Libraries and Packages

To understand machine learning, there is need to have basic knowledge of python programming. In addition, there are a number of libraries and packages generally used in performing various machine learning tasks as listed below:

- **numpy** – is used for its n-dimensional array objects.
- **pandas** – is a data analysis library that includes data frames.
- **matplotlib** – is 2D plotting library for creating graphs and plots.
- **scikit-learn** – the algorithms used for data analysis and data mining tasks.
- **seaborn** – a data visualization library based on matplotlib.

- **Installation**

Software for machine learning can be installed in any of the two methods as discussed below:

Method

Download and install python separately from python.org on various operating systems as explained below:

To install python after downloading, double click the .exe (for Windows) or .pkg (for Mac) file and follow the instructions on the screen.

For Linux OS, check if python is already installed by using the following command at the prompt

$ python –version. …

If python 2.7 or later is not installed, install python with the distribution's package manager. Note that the command and package name varies.

On Debian derivatives such as Ubuntu, apt can be used:

$ sudo apt-get install python3

Now, open the command prompt and run the following command to verify that python is installed correctly:

$python3 – version Python 3.6.2

Python Machine Learning – Environment Setup

Similarly, numpy, matplotlib and necessary libraries can be downloaded and installed, individually using installers like pip. For this purpose, the commands can be used as shown below:

$pip install numpy

$pip install matplotlib

$pip install pandas

$pip install seaborn

1.4 APPLICATIONS OF AI VIRTUAL KEYBOARD AND MOUSE

- ✓ Amidst the COVID-19 situation, it is not safe to use the devices by touching them because it may result in a possible situation of spread of the virus by touching the devices, so the proposed AI virtual keyboard and mouse can be used to control the PC without using the physical input devices.
- ✓ This model has a greater accuracy of 95% which is far greater than the that of other previously proposed models for virtual mouse and keyboard, and it has many applications
- ✓ Physically challenged persons can use this system to control basic functions in the computer.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 SYSTEM ANALYSIS

AI virtual keyboard and Mouse will not only used by physically challenged people it can also used by everyone who needs a keyboard and mouse in any situation. This is a software program it can run in any type of operating system. This system can be used in simple steps.

AI virtual Keyboard and Mouse can be used in crucial conditions like physical keyboard and mouse broken situation. This system mainly used for Human Computer Interaction (HCI) and it can also used for any other applications like typing in applications like notepad, MS world and many other, the mouse system is used for mouse operations and other operations like drawing, editing and others.

2.2 FUNCTIONAL REQUIREMENTS

The functions of AI virtual Keyboard and Mouse as follows:

- This system should take video as input in the form of frames.
- Form the input video system will able to detect hands and track them till end of video.
- From detected hand it will able to differentiate between Left hand and Right hand.
- The keyboard system will able to type basic keyboard keys.
- The mouse system will able to operate curser movement, left click, right click, drag and drop and scroll up and down operations.

2.3 NON-FUNCTIONAL REQUIREMENTS

- Accuracy: In this system should be able to have more accuracy compared to existing system.
- Cost: this system should be less cost.
- Portable: this software should be portable to any ware at any time.
- This system should be able to operate all basic HCI operation.

2.4 HARDWARE REQUIREMENTS

2.4.1 Intel Pentium (I5) processor and above

Intel Core is a line of streamlined midrange consumer, workstation and enthusiast computer central processing units (CPUs) marketed by Intel Corporation. These processors displaced the existing mid- to high-end Pentium processors at the time of their introduction, moving the Pentium to the entry level. Identical or more capable versions of Core processors are also sold as Xeon processors for the server and workstation markets. The lineup of Core processors includes the Intel Core i3, Intel Core i5, Intel Core i7, and Intel Core i9, along with the X-series of Intel Core CPUs.

Figure 2.1: Intel i5 processor

2.4.2 Hard disk 10GB and above

hard disk is a magnetic storage medium for a computer. Hard disks are flat circular plates made of aluminum or glass and coated with a magnetic material. Hard disks for personal computers can store terabytes (trillions of bytes) of information. Data are stored on their surfaces in concentric tracks. A small electromagnet, called a magnetic head, writes a binary digit (1 or 0) by magnetizing tiny spots on the spinning disk in different directions and reads digits by detecting the magnetization direction of the spots

Figure 2.2: Hard Disk

2.4.3 Minimum of 4GB of RAM

RAM (Random Access Memory) is the hardware in a computing device where the operating system (OS), application programs and data in current use are kept so they can be quickly reached by the device's processor. RAM is the main memory in a computer. It is much faster to read from and write to than other kinds of storage, such as a hard disk drive (HDD), solid-state drive (SSD) or optical drive. Random Access Memory is volatile. That means data is retained in RAM as long as the computer is on, but it is lost when the computer is turned off. When the computer is rebooted, the OS and other files are reloaded into RAM, usually from an HDD or SSD.

Figure 2.3: 4GB RAM

2.4.4 System type 32/64 bit

A 32-bit system can access 2^{32} memory addresses, i.e., 4 GB of RAM or physical memory; ideally, it can also access more than 4 GB of RAM. A 64-bit system can access 2^{64} memory addresses, i.e., actually 18-Quintillion bytes of RAM. In short, any amount of memory greater than 4 GB can be easily handled by it.

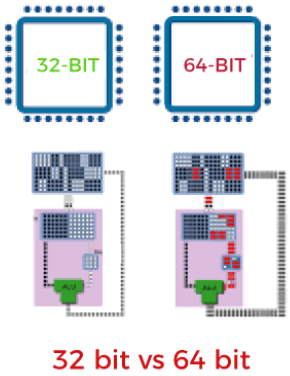


Figure 2.4: 32/64 bit system

2.4.5 web-camera

Security cameras, also known as CCTV cameras, are used to convey signals from one particular place to a monitor situated at a distance, whereas IP cameras normally work on IP networks which link the camera from the remote area to the assigned security location.

Figure 2.5: Web camera

2.5 SOFTWARE REQUIREMENTS

- IDLE: python 2.7 or higher
- IDE: Pycharm
- openCV package

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. 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 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire 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.

- PyautoGUI package

PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be simple. PyAutoGUI works on Windows, macOS, and Linux, and runs on Python 2 and 3.

PyAutoGUI has several features:

- ✓ Moving the mouse and clicking in the windows of other applications.
- ✓ Sending keystrokes to applications (for example, to fill out forms).
- ✓ Take screenshots, and given an image (for example, of a button or checkbox), and find it on the screen.
- ✓ Locate an application's window, and move, resize, maximize, minimize, or close it (Windows-only, currently).
- ✓ Display alert and message boxes.

CHAPTER 3

SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of system analysis, systems architecture and system engineering. If the broader topic of product development "blends the perspective of marketing, design and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. System design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

3.1 SYSTEM ARCHITECTURE

AI virtual keyboard and mouse system consists of four modules which are :

3.1.1 Hand Gesture Recognition

Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms .It is a subdiscipline of computer vision. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current[when?] focuses in the field include emotion recognition from face and hand gesture recognition. Users can use simple gestures to control or interact with devices without physically touching them. Many approaches have been made using cameras and computer vision algorithms to interpret sign language. However, the identification and recognition of posture, gait, proxemics, and human behaviors is also the subject of gesture recognition techniques. Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a better bridge between machines and humans than older text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse and interact naturally without any mechanical devices.

Gesture recognition features:

- Higher accuracy
- Higher stability
- Quicker time to unlock a device

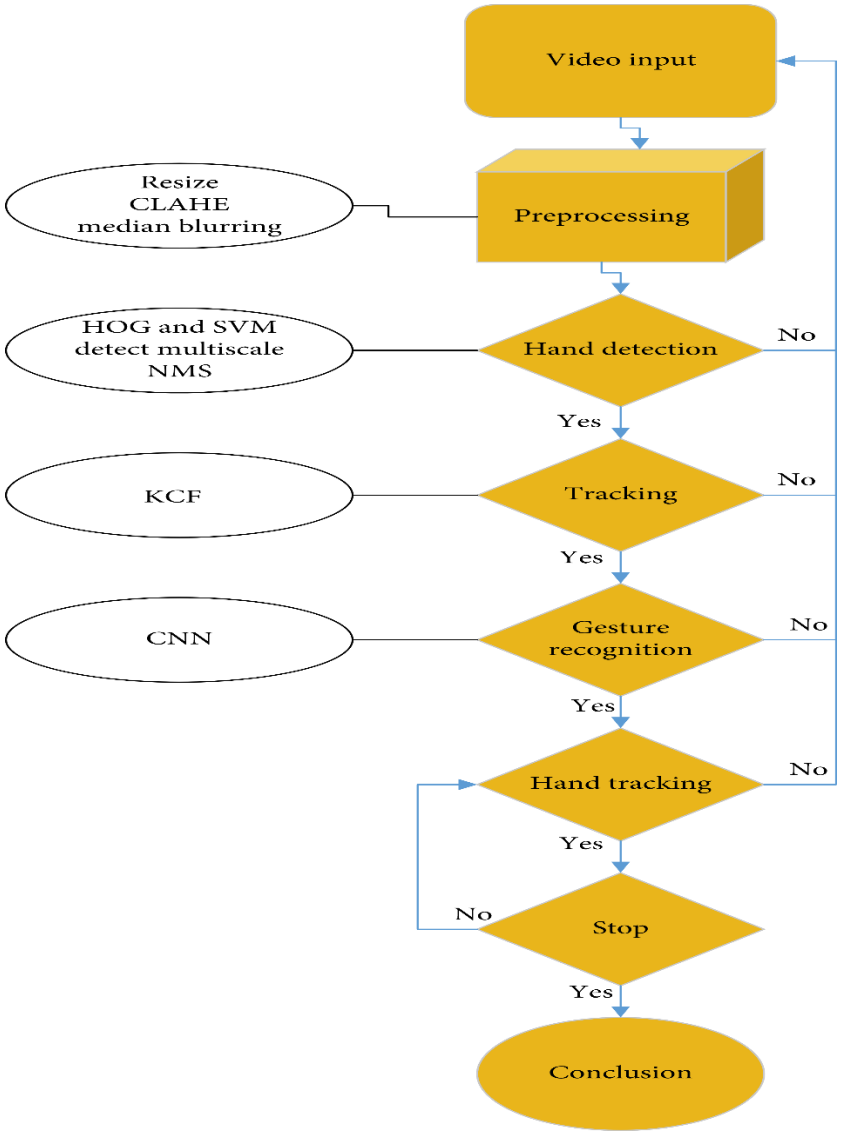


Figure 3.1: Hand gesture Recognition system

In hand gesture recognition we used four algorithms which are :

3.1.1.1. HOG Algorithm

The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

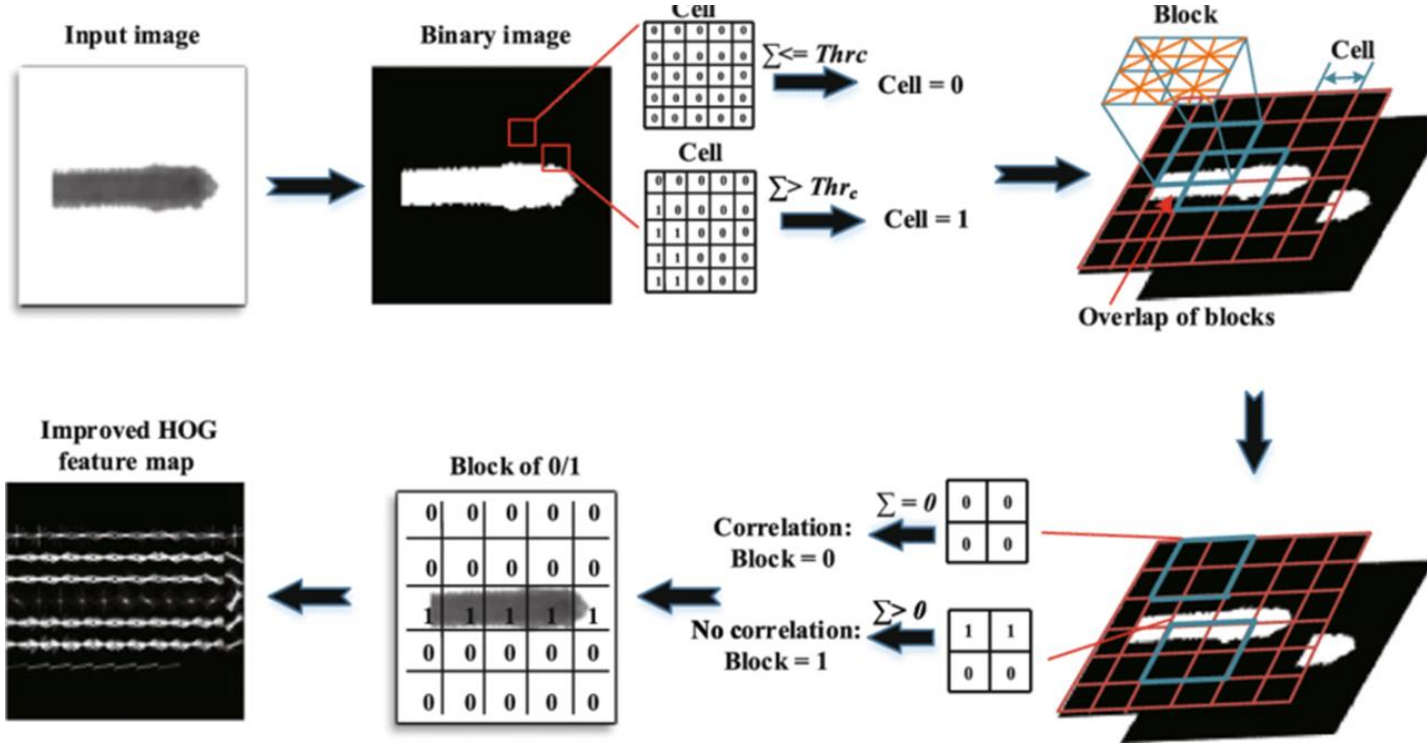


Figure 3.2: Hand gesture Recognition system

3.1.1.2. SVM Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

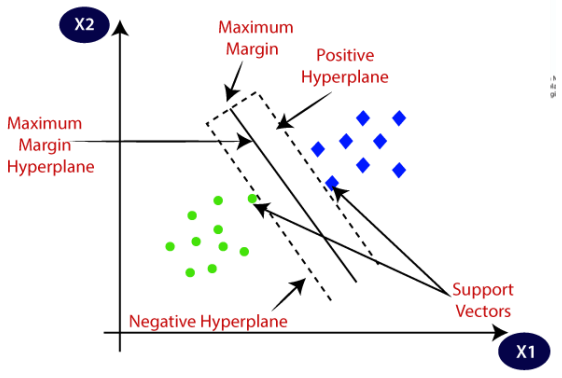


Figure 3.3: SVM Hyperplane

3.1.1.3. KCF Algorithm

An object to be tracked is usually selected by a rectangular bounding box. The task of a tracker is to follow the object in the video by updating the bounding box parameters (the position at the simplest case).The basic idea of the correlation filter tracking is estimating an optimal image filter such that the filtration with the input image produces a desired response. The desired response is typically of a Gaussian shape centered at the target location, so the score decreases with the distance.The filter is trained from translated (shifted) instances of the target patch. When testing, the response of the filter is evaluated and the maximum gives the new position of the target. The filter is trained on-line and updated successively with every frame in order the tracker adapts to moderate target changes.

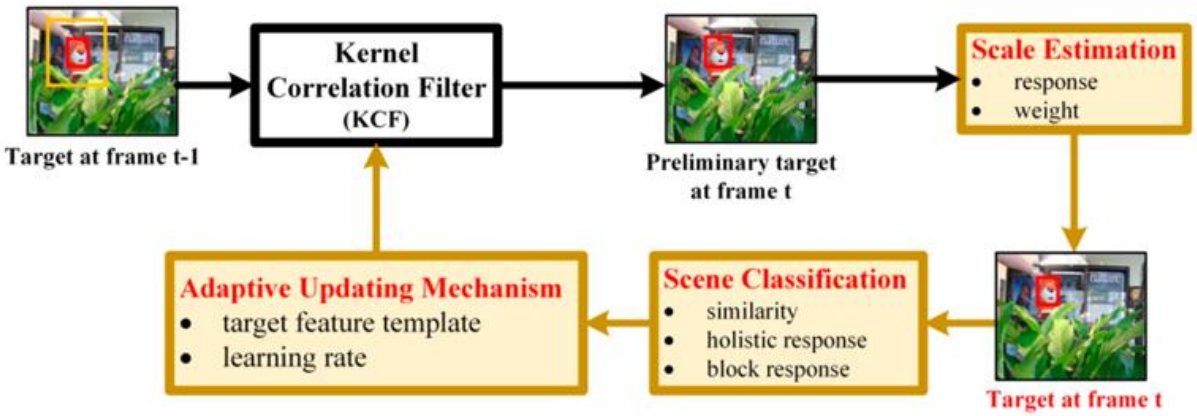


Figure 3.7: Kcf Filter Usage

3.1.1.4. CNN Algorithm

A convolutional neural network is a feed-forward neural network that is generally used to analyze visual images by processing data with grid-like topology. It's also known as a ConvNet. A convolutional neural network is used to detect and classify objects in an image.

Below is a neural network that identifies two types of flowers: Orchid and Rose.

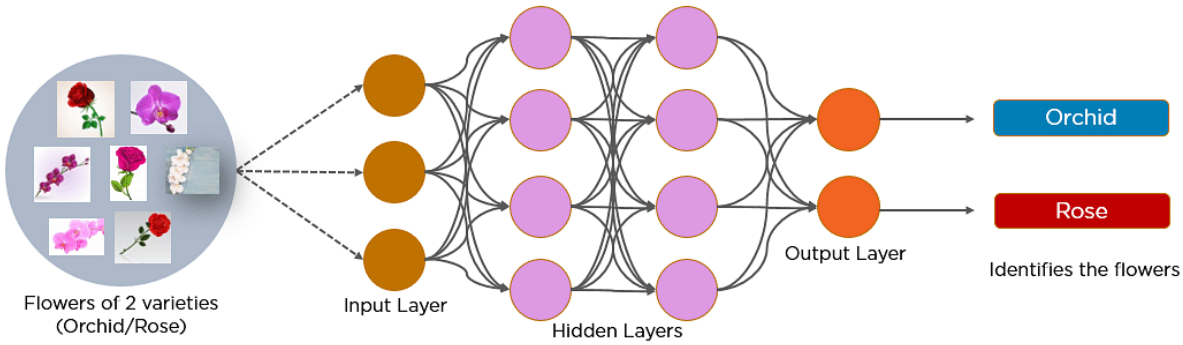


Figure 3.8: Neural network that identifies Orchid and Rose

In CNN, every image is represented in the form of an array of pixel values.

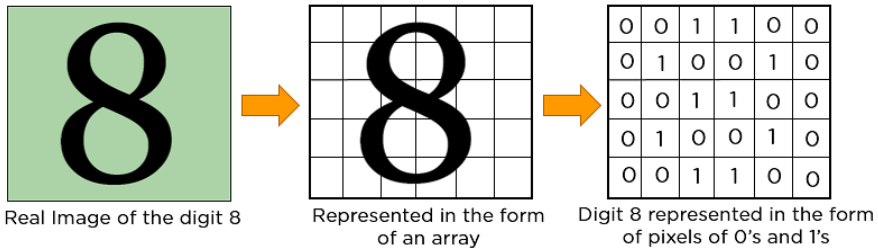


Figure 3.9: Image is represented in the form of an array of pixel values

Hand Type Classification

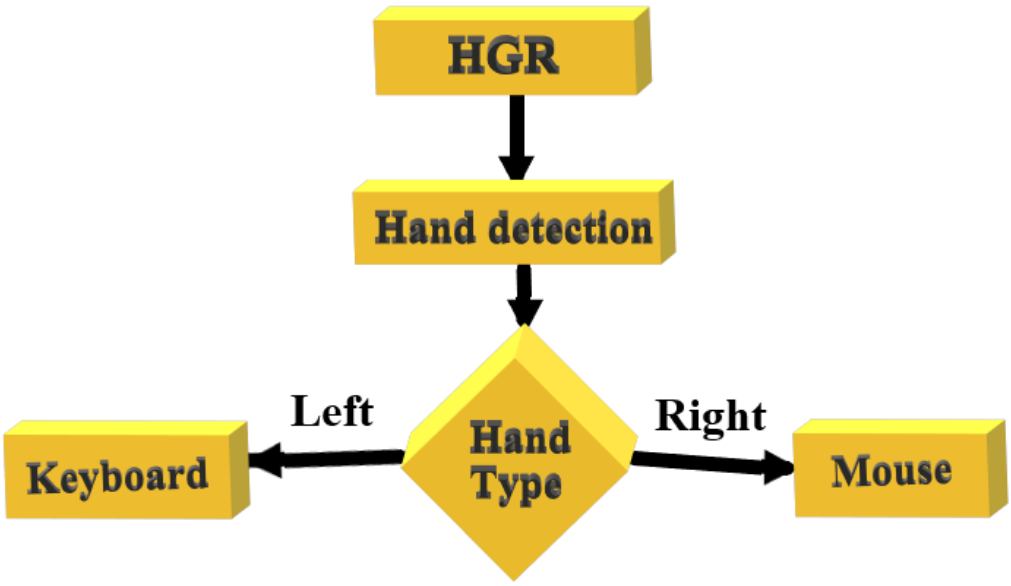


Figure 3.19: Hand Type classification system

Hand type classification is used to classify between left hand and right hand. This system will use mediapipe and OpenCV libraries in python to detect the Right Hand and Left Hand. This system will be using the Hands model from mediapipe solutions to detect hands, it is a palm detection model that operates on the full image and returns an oriented hand bounding box. After returning detected hand image it classifies as left hand or right hand. If the detected hand is left then the system will shift to virtual keyboard system or detected hand is right then the system will shift to virtual mouse system.

3.1.2 Virtual Keyboard System

Virtual keyboard system will be activated when there is a left hand in the web-cam video. In Virtual keyboard system the detected hand will be tracked through end of video. For displaying keyboard we used image displayed in video method from opencv. Opencv will help to display the keyboard button in the form of images, where this images are created in 40X40 pixel size. We draw all images one by one to form a full keyboard. After that the HGR system will send land mark locations in pixels for the detected left hand. By using those landmarks locations we will able to use it for indexing key-butttons of keyboard. When the detected hands index finger placed between the image start pixel location and end pixel location the button will highlighted in green colored corner rectangle.

Then the system will calculate the pixel distance between tip of index finger and tip of middle finger, if the distance is less than the 40 pixel then respective keyboard button will be clicked using pyautogui method, then the process will be continued until the system will stopped by user.

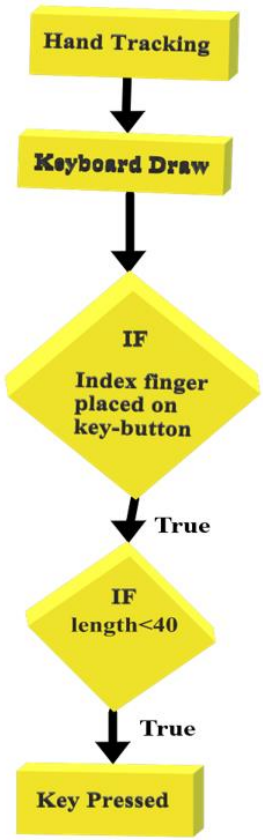


Figure 3.20: Virtual Keyboard System

3.1.3 Virtual Mouse System

Virtual mouse system is based on the frames that have been captured by the webcam in a laptop or PC. By using the Python computer vision library OpenCV, the video capture object is created and the web camera will start capturing video. The web camera captures and passes the frames to the AI virtual system. The AI virtual mouse system uses the webcam where each frame is captured till the termination of the program. The video frames are processed from BGR to RGB color space to find the hands in the video frame by frame. Then the HGR system will send the locations of all 21 landmarks of hand. By using those land marks the system will calculate which finger is up and which finger is down, based on those fingers the system will do fallowing operations.

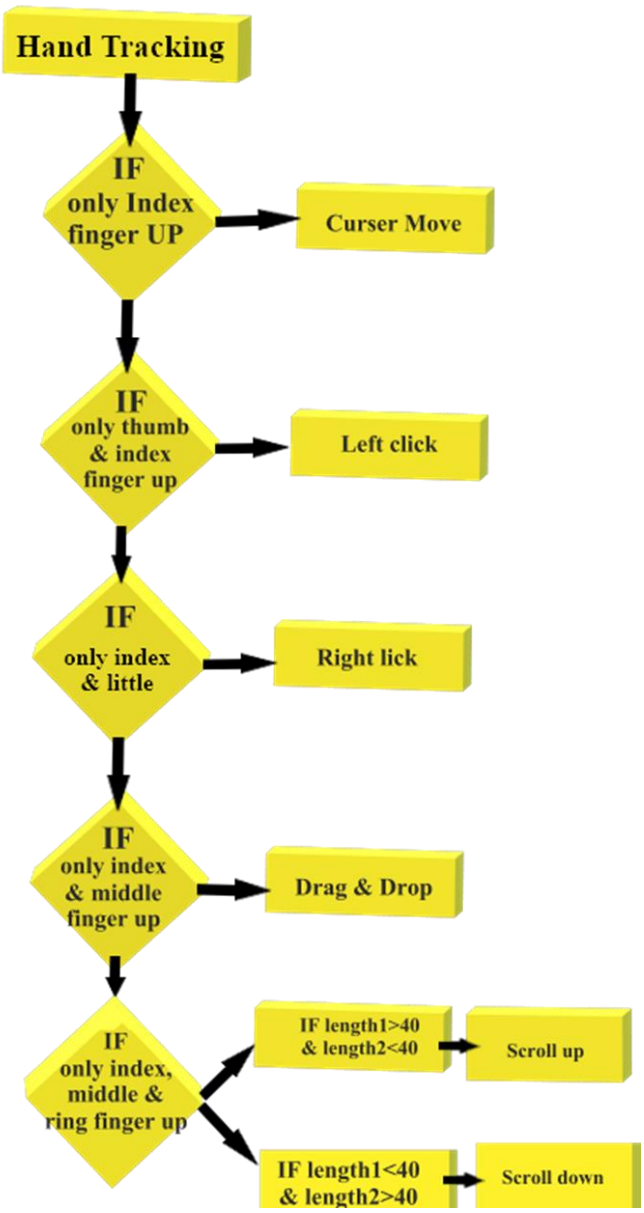


Figure 3.21: Virtual Mouse system

- **Curser Movement**

  For the mouse curser movement we use only index finger up. The system will check for which finger is up and down, if only the index finger is up then mouse curser movement is started. The mouse curser movement will be done by accessing landmark locations of index finger. By using location of tip of index finger curser will move in same direction as index finger moves

- **Mouse left click**

  when the system check for fingers up and found only index and thumb fingers are up, then the mouse left click will happen. For mouse left click operation the HGR sends land mark locations of all 21 points then the system only access index tip and thumb tip. The distance between index finger and thumb finger calculated, if the distance is less than 40 pixels then the left click will happen using pyautogui.

- **Mouse right click**

  when the system check for fingers up and found only index and little fingers are up, then the mouse right click will happen. For mouse right click operation the HGR sends land mark locations of all 21 points then the system only access index tip and little finger tip. The distance between index finger and little finger calculated, if the distance is less than 50 pixels then the right click will happen using pyautogui.

- **Mouse drag and drop**

  when the system check for fingers up and found only index and middle fingers are up, then the mouse drag and drop will happen. For mouse drag and drop operation the HGR sends land mark locations of all 21 points then the system only access index finger tip and middle finger tip. The distance between index finger and middle finger calculated, if the distance is less than 40 pixels then the mouse drag will started and end when the distance is more than 40 pixels, it will happen using pyautogui.

- **Mouse scroll up and down**

  Mouse scroll happens when index, middle and ring fingers are up, when the distance between index and middle is more than 40 pixel and distance between middle and ring finger is less than 40 pixels then mouse scroll up will happen, when the distance between index and middle is less than 40 pixel

and distance between middle and ring finger is more than 40 pixels then mouse scroll down will happen.

3.2 MEDIAPIPE

MediaPipe powers revolutionary products and services we use daily. Unlike power-hungry machine learning Frameworks, MediaPipe requires minimal resources. It is so tiny and efficient that even embedded IoT devices can run it. In 2019, MediaPipe opened up a whole new world of opportunity for researchers and developers following its public release.

3.2.1 WHAT IS MEDIAPIPE?

MediaPipe 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.

3.2.2 MEDIAPIPE TOOLKIT

MediaPipe Toolkit comprises the Framework and the Solutions. The following diagram shows the components of the MediaPipe Toolkit.

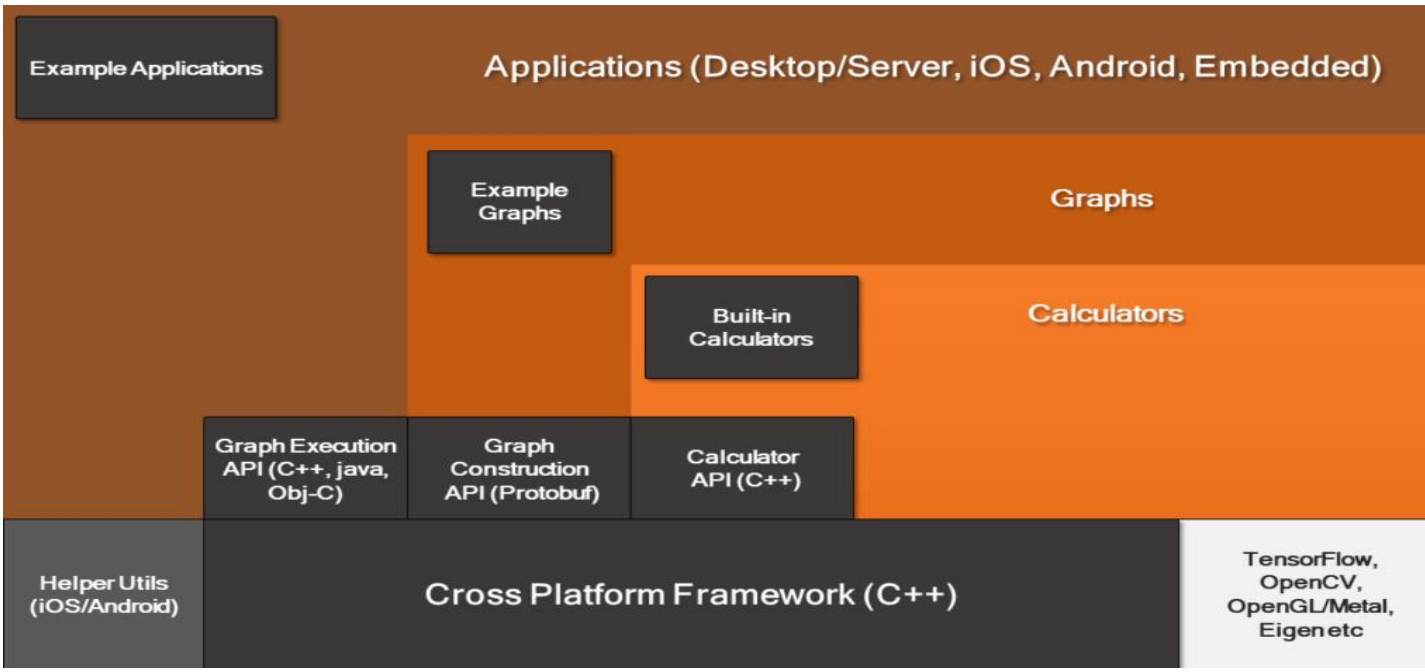


Figure 3.22: Mediapipe Toolkit

3.2.3 FRAMEWORK

The Framework is written in C++, Java, and Obj-C, which consists of the following APIs.

- Calculator API (C++).
- Graph construction API (Protobuf).
- Graph Execution API (C++, Java, Obj-C).

3.2.4 GRAPHS

The MediaPipe perception pipeline is called a Graph. Let us take the example of the first solution, Hands. We feed a stream of images as input which comes out with hand landmarks rendered on the images.

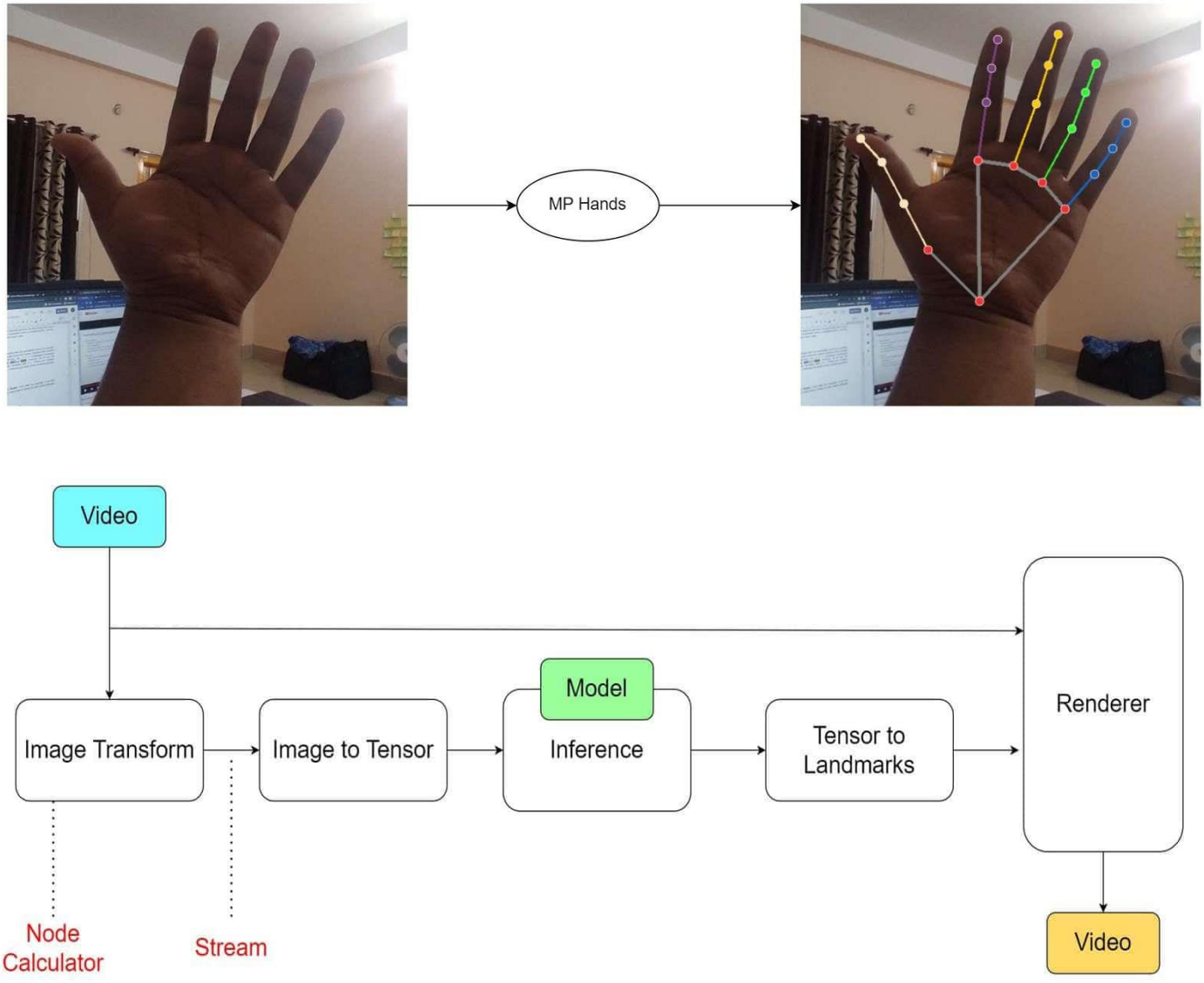


Figure 3.23: Mediapipe Hands Solution Graph

3.2.5 MEDIAPIPE SOLUTIONS

Solutions are open-source pre-built examples based on a specific pre-trained TensorFlow or TFLite model. You can check Solution specific models here. MediaPipe Solutions are built on top of the Framework. Currently, it provides sixteen solutions as listed below.

- Face Detection
- Face Mesh
- Iris
- Hands
- Pose
- Holistic
- Selfie Segmentation
- Hair Segmentation
- Object Detection
- Box Tracking
- Instant Motion Tracking
- Objectron
- KNIFT
- AutoFlip
- MediaSequence
- YouTube 8M

The solutions are available in C++, Python, JavaScript, Android, iOS, and Coral. As of now, majority of the solutions are available only in C++ (except KNIFT and IMT) followed by Android, with Python not too far behind. The other wrapper languages, too, are growing fast with a very active development state. As you can see, even though MediaPipe Framework is cross-platform, that does not imply the same for the solutions. MediaPipe is currently at alpha version 0.7. We can expect the solutions to get more support with the beta releases. Following are some of the solutions provided by MediaPipe.

MediaPipe Python solutions are the easiest for beginners because of the simplicity of the setup process and the popularity of the Python programming language. The modularity of the MediaPipe Framework enables customization. But before plunging into customization, we recommend getting comfortable with various pre-built solutions. Understand internal APIs associated with them and then tweak the outputs to create your exciting applications.

CHAPTER 4

IMPLEMENTATION

The structure of project file shown below:

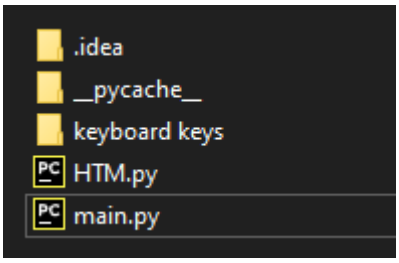


Figure 4.1: project file directory

There is one main folder called keyboard keys and two python files called HTM.py and main.py in the project directory. Folder called keyboard keys consists of all keyboard keys image files in png or jpg format to add in the video. This image keys have pixel size of 40x40 except Caps lock, Shift, Backspace and Enter of size 60x40 and Space of size 120x40. This images are name in numeric order from 01 to 60 for easy placing in screen.

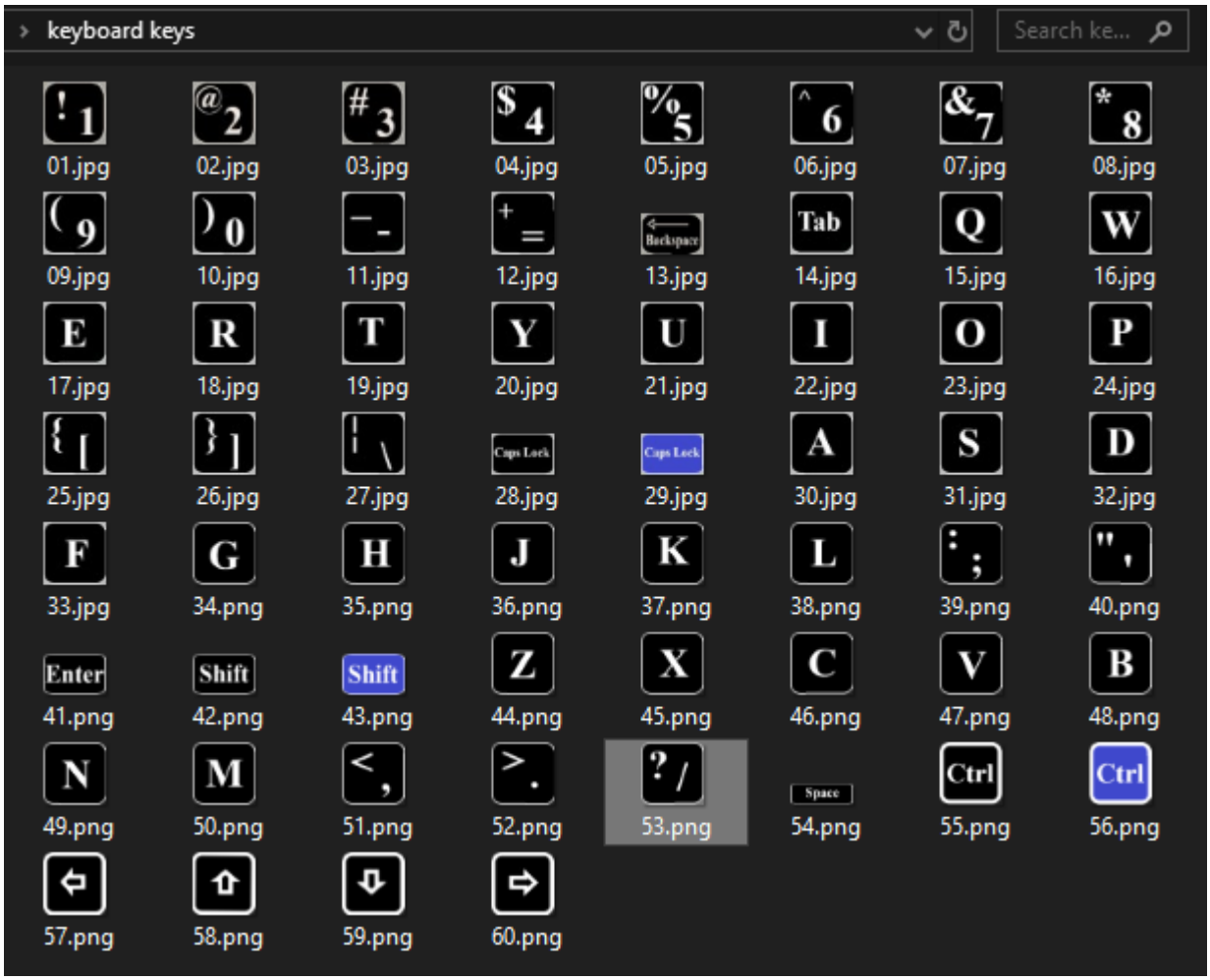


Figure 4.2: keyboard key images in file directory

- **__init__**

HTM.py python file consists of Hand Tracking Module which is used for Hand detection and Tracking. This python module consists of three functions in single class called HandDetector, the first function is __init__ function, this function is used to initialize the Maximum number of hands to detect in the video, Minimum Detection Confidence Threshold and Minimum Tracking Confidence Threshold for a detection of hand in the video and this accept four parameters.

```
def __init__(self, mode=False,maxHands=2, detectionCon=0.5, minTrackCon=0.5):
    """
```

Figure 4.3: __init__ function in HTM

- **findHands**

Next function is findHands function is used to Hand detection and tracking, this function accept three parameters and two of them is already initialized.

```
def findHands(self, img, draw=True, flipType=True):

    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    self.results = self.hands.process(imgRGB)
    allHands = []
    mylmList=[]
    HandType=[]
    myHand = {}
```

Figure 4.4: findHands function in HTM

findHands accept img that is recorded video from web camera, and draw=True for drawing palm sketch in the video else draw=False. The flipType represents the plam is faced towards web-cam or not, if palm faced towards then flipType is True else its Flase. The frames of images convert BGR to RGB color for palm detection. After palm detection it assign all 21 points in Hand and add those locations to mylmList string, Hand Type to HandType string. After appending data to strings system will calculate center of palm to draw bounding box to detected hand.

- **fingersUp**

fingerUp function is used to find which finger is up and which finger is down. This function accept myHand string which consists of Hand land marks, Hand Type and image of detected hand. This function will calculate which finger is up or down using tip of finger and bottom of finger. This methods returns zero for down fingers and one for up fingers.

IF fingers tips Y co-ordinte > fingers bottom Y co-ordinate:
finger is up
ELSE:
finger is down

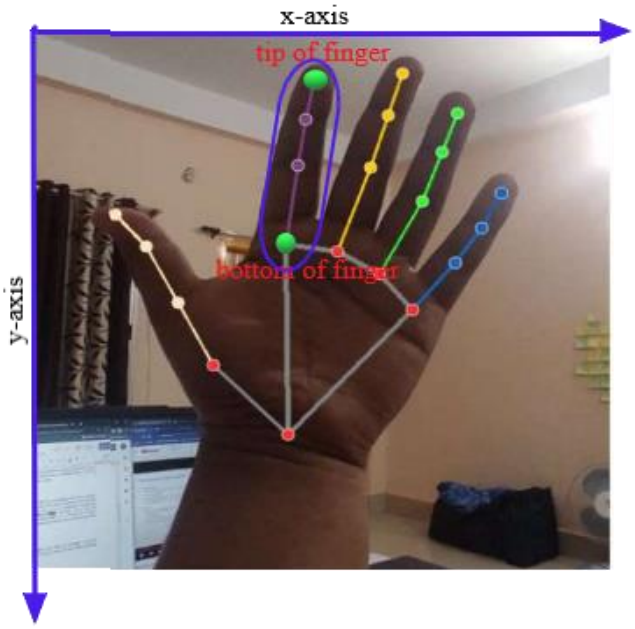


Figure 4.4: finger up calculating method

The next one is main.py file, in this file where user import all module and implemented design of the system. First in main.py file user import all needed python packages.

- import cv2
  cv2 is a openCV library function used to access image frames from the web-cam, write text on video, reading images from files, drawing shapes like rectangle, circle and other, coloring texts and font styles.
- import pyautogui
  pyautogui mainly used for keyboard key pressing.
- import mouse
  mouse package is used for mouse curser movement from one location to another location, mouse left click in particular location, mouse right click on particular location, mouse drag from one location and drop it in another location and mouse scroll up and down function.
- import time
  time module used to sleep the system for number of miliseconds.
- import numpy as np
  numpy is used to read and write large number of data in a stings.

- from HTM import HandDetector
  HTM is used for hand detection and Tracking.
- import os
  os package is used to access file directory in the system for images reading in folder.

The next step is to record a video from web- cam for this cv2 package is used.

```
cap = cv2.VideoCapture(0)
    cap.set(4, 1080)
        cap.set(6, 720)
            while True:
                success, img = cap.read()
                cv2.imshow("Image",  img)
                cv2.waitKey(1)
```

The next step is to assign images of keys path to a array that is done using os python module. The sample code shown below where keyboard keys is folder name where all images present. Then all paths are assigned to overlayls list.

```
Folder="keyboard keys"
    myList=os.listdir(Folder)
        overlayls=[]
            for imPath in myList:

image=cv2.imread(f'{Folder}/{imPath}')
    overlayls.append(image)
```

After the video recording next step is to find hand in hands in the recorded video using Hand Tracking modules findHand function by calling it. It can see in below code, Where it accepts img as image frames and it returns drawing image, oldLMlist of 21 landmark locations, Hand type detected and myHand array of all data.

```
img,oldLMlist,MyHandType,myHand = detector.findHands(img)
```

Using handType then add separate systems i.e, Virtual keyboard for left hand and virtual Mouse for Right hand using if statement. The code is shown below,

```
if MyHandType=="Right":
        // Virtual Mouse
        elif MyHandType==”Left”:
            //Virtual Keyboard
```

After hand type classification for left hand assign virtual keyboard system, in first step of virtual keyboard system is to draw images in video screen for that opencv will be helpful as below given code.

```
for i in range(0, 12):
```

img[20:60, 20 + (45 * i):60 + (45 * i)] = overlayls[i]

in above code the first row of keys will be seen in the video that is all 12 buttons, where 20:60 is y1 and y2 i.e, starting & ending locations of y-axis, 20 + (45 * i):60 + (45 * i) are x1 and x2 locations are starting and ending locations of x-axis. By using this method all keyboard keys will be drawn.

The next step is to highlight keyboard keys with respect to index finger by using corner rectangle method, i.e, drawing around the keyboard key which user placed in keyboard key. For corner rectangle method accepts starting location of image and ending location of image (x1,y1,x2,y2).

Then for click of any key user has to place index finger above that key and keep the distance less than 40 pixels between the index and middle finger. The distance is calculated using below formula.

$$D(c_{2(x,y)}, c_{1(x,y)}) = \sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2}.$$

Where Xmin and Ymin are the first finger locations Xmax and Ymax are the second finger locations.

After keyboard design then move on to Mouse design, by using the same hand detected module and Land mark lists of the hand. Where mouse can perform the following operations.

- Mouse move
  For mouse movement the user has to use only index finger must up, then were user moves index finger same direction mouse curser will move. For this below code will used,

  clocX = plocX + (x3 - plocX) / 10
  clocY = plocY + (y3 - plocY) / 10
  mouse.move(clocX, clocY)

  in the above code clocX, clocY are current locations of mouse curser this initially will be zero plocX, plocY are previous locations of mouse curser where x3 and y3 will be calculated using below code and 10 is the smoothing value used to easily move the mouse curser. Mouse.move is mouse moving function from mouse python package.

  x3 = np.interp(x1, (frameR, wCam - frameR), (0, wScr))
  y3 = np.interp(y1, (frameR, hCam - frameR), (0, hScr))

  frameR is framerate that is 90 wCam, hCam are width and height of camera screen i.e, 1080 & 720. wScr, hScr are width and height of desktop screen i.e, 1400 & 780 its depend upon user PC or Laptop screen. x1, y1 are index finger location in screen.

- **Mouse left click**

  For mouse left click the user has to put only thumb and index finger up. Mouse left click will happen where mouse curser is currently present. This happens using below function. Where mouse.click is function from mouse python package for mouse clicking.

  mouse.click('left')

- **Mouse right click**

  For mouse left click the user has to put only index and little finger up. Mouse right click will happen where mouse curser is currently present. This happens using below function. Where mouse.click is function from mouse python package for mouse clicking.

  mouse.click('right')

- **Mouse drag and drop**

  For mouse drag and drop the user has to put only index and middle finger up. Mouse drag will happen when the distance between index and middle finger is less than 40 pixel, then the drop will happen when distance is more than 40 pixels. This happens using below function. Where mouse.drag is function from mouse python package for mouse drag.

  mouse.drag(0, 0, drag_to_X - drag_from_X, drag_to_Y - drag_from_Y, duration=0.1)

  where 0,0 is starting position of dragging and drag_to_X - drag_from_X, drag_to_Y - drag_from_Y is the ending position of dragging and duration is the total time of dragging that is 0.1 second.

- **Mouse scroll up**

  For mouse scroll up operation user has to lift index, middle and ring fingers must up and the distance between index and middle finger is more than 40 pixels and the distance between middle and ring finger is less than 40 pixels. This will happen using below function.

  mouse.wheel(-1)

- **Mouse scroll down**

  For mouse scroll down operation user has to lift index, middle and ring fingers must up and the distance between index and middle finger is less than 40 pixels and the distance between middle and ring finger is more than 40 pixels. This will happen using below function.

  mouse.wheel(1)

CHAPTER 5

RESULTS

Snapshot of Instruction page:

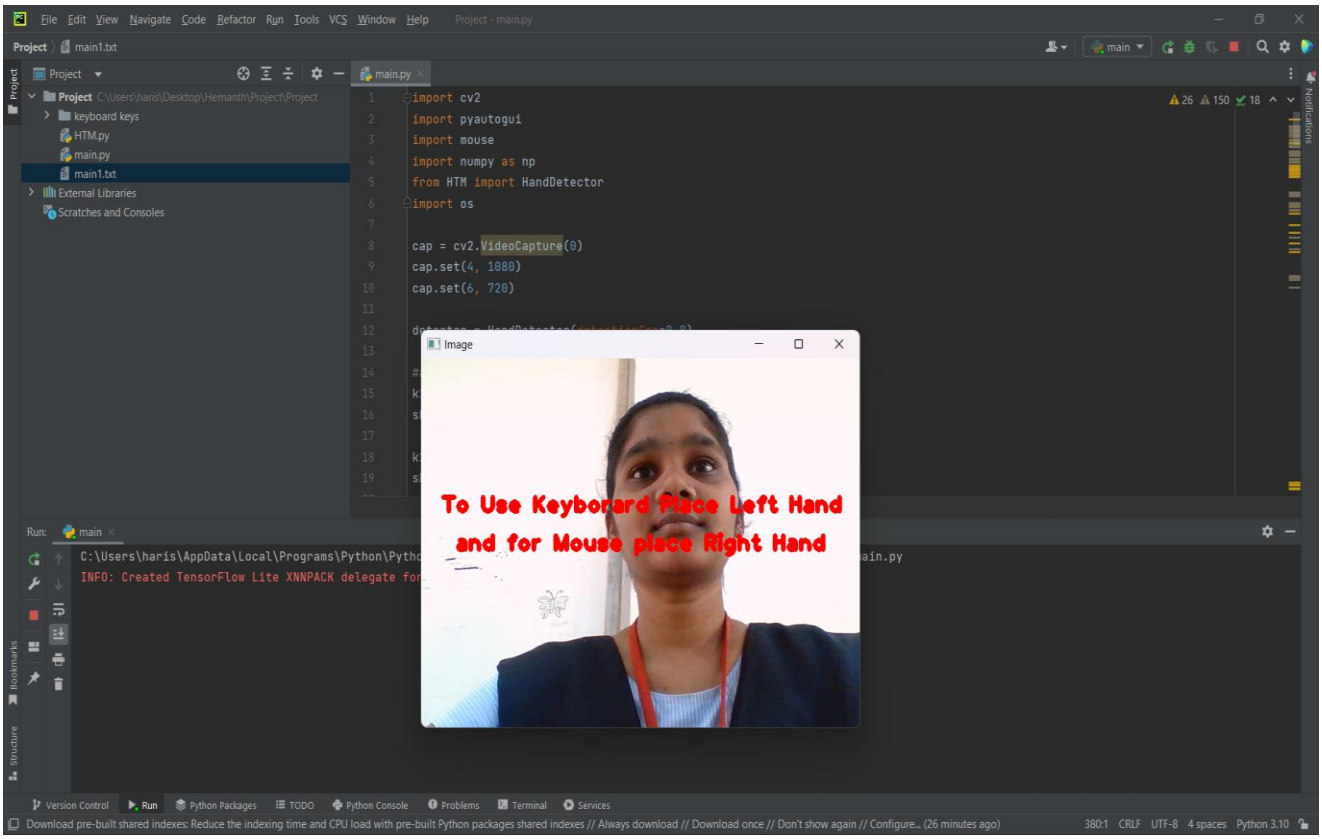


Figure 5.1: Instruction Page

The above Figure 5.1 shows the instruction to us in the project for the people understanding.

- If the user wants to use keyboard, then user has to use his left hand
- If the user wants to use the mouse , then user has to use this right hand.
- If the user wants to stay still then user has to either show both the hands or show no hands at all.

Snapshot of Hand detection:

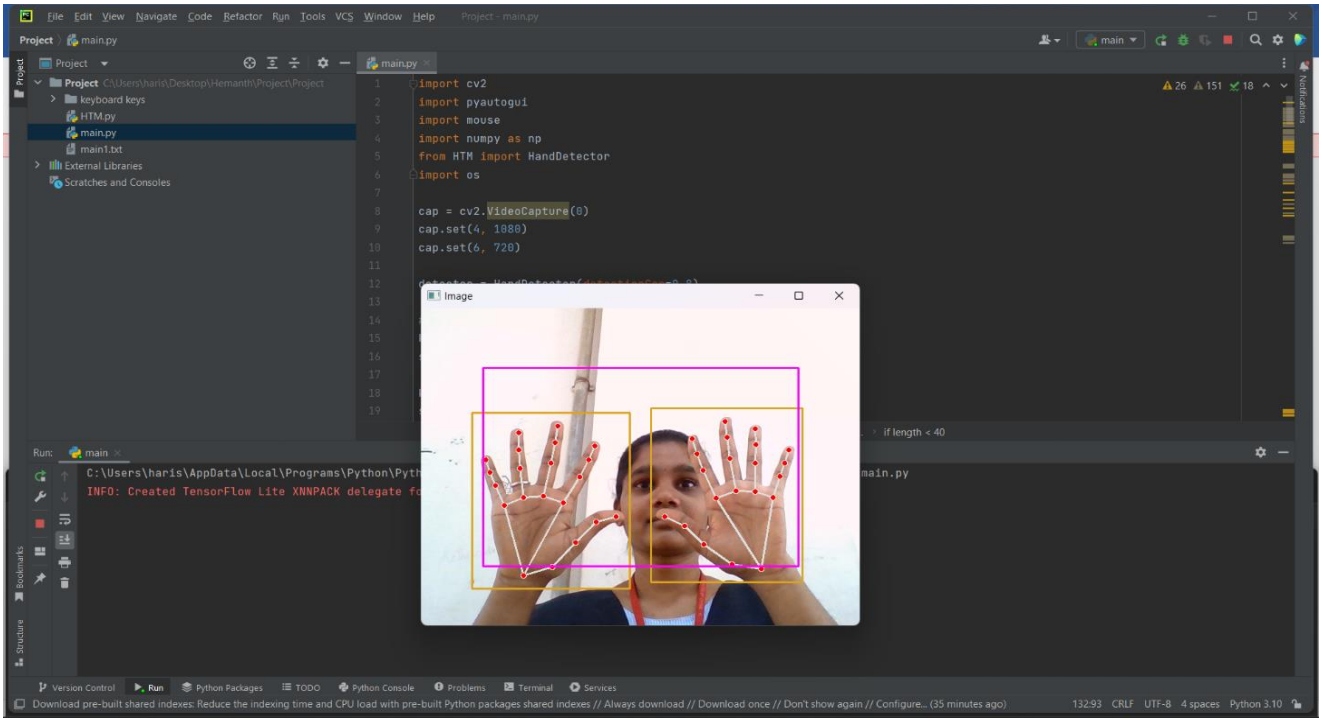


Figure 5.2: Both Hand Detection

The above Figure 5.2 shows when the user uses both the left and the right hand which is detected and kept at still position as both hands are shown to the system.

Snapshot of Keyboard:

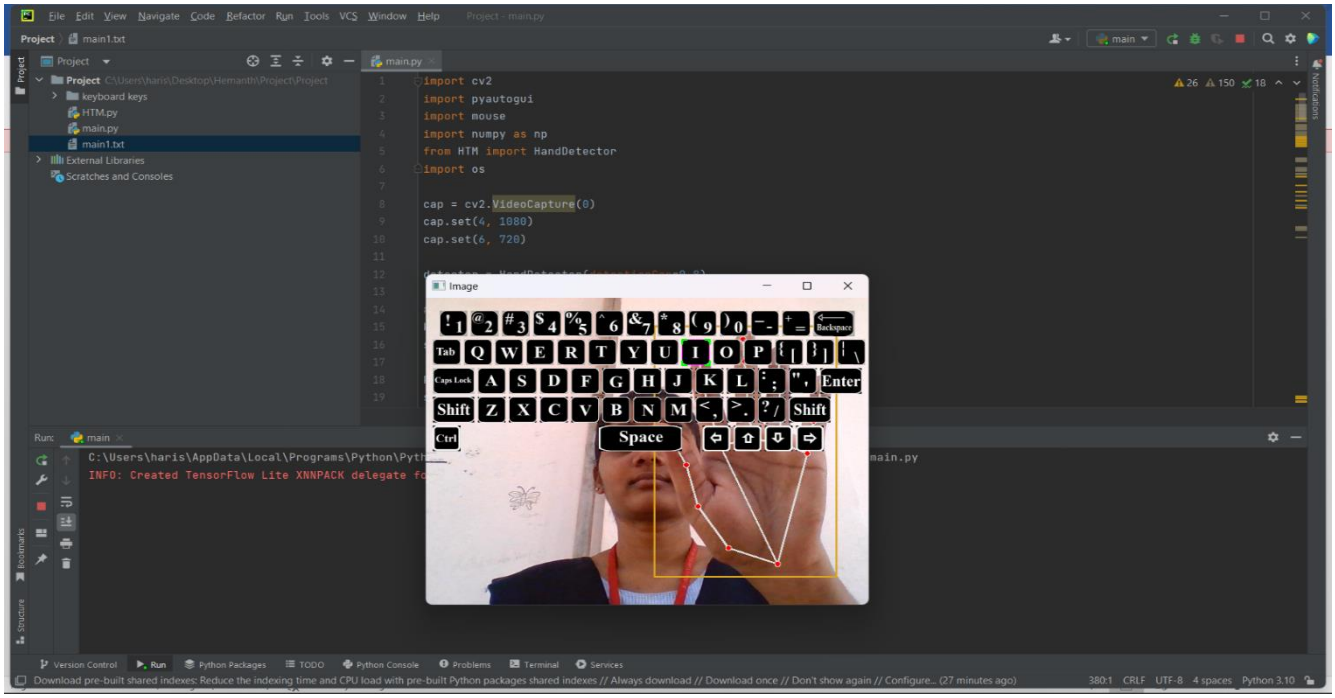


Figure 5.3: Keyboard Implementation

The above Figure 5.3 shows the implementation of keyboard using the left hand.

Snapshot of Using the keyboard in real time :

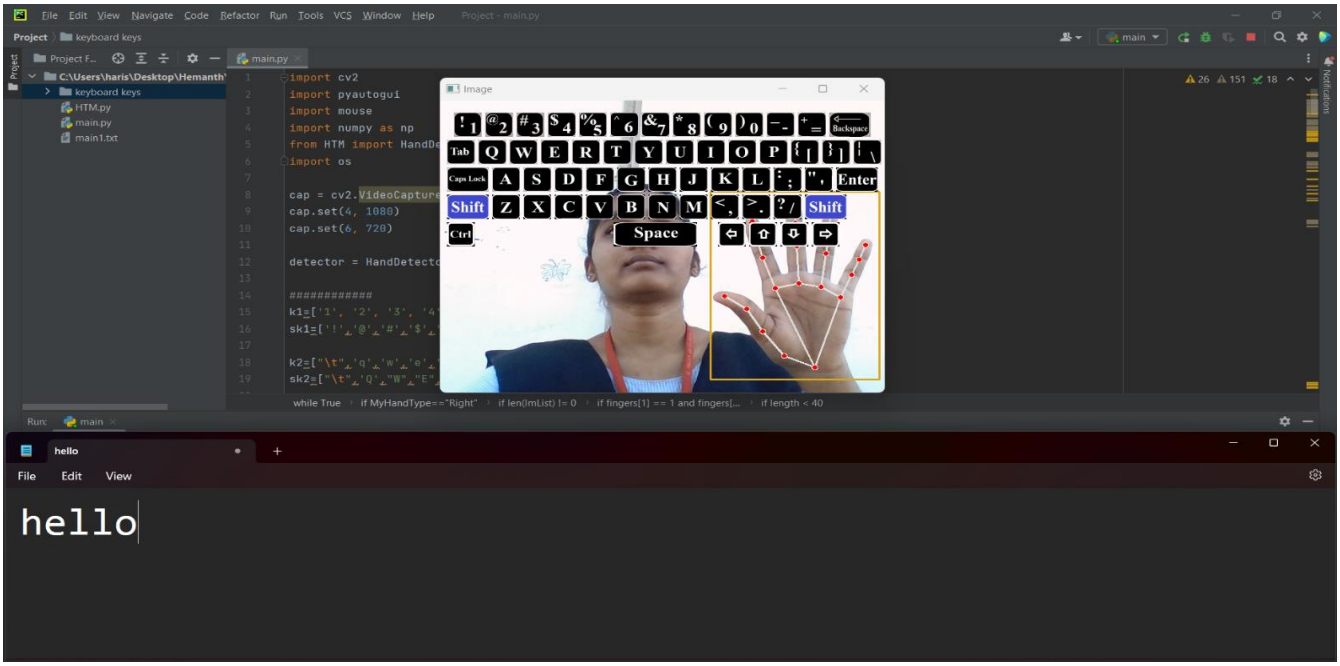


Figure 5.4: Using of Keyboard in Notepad

The above Figure 5.4 shows the usage of keyboard which can be used in any type of typing platform using the left hand .Here the user can see that the highlighted key.

Snapshot of Using Special Characters:

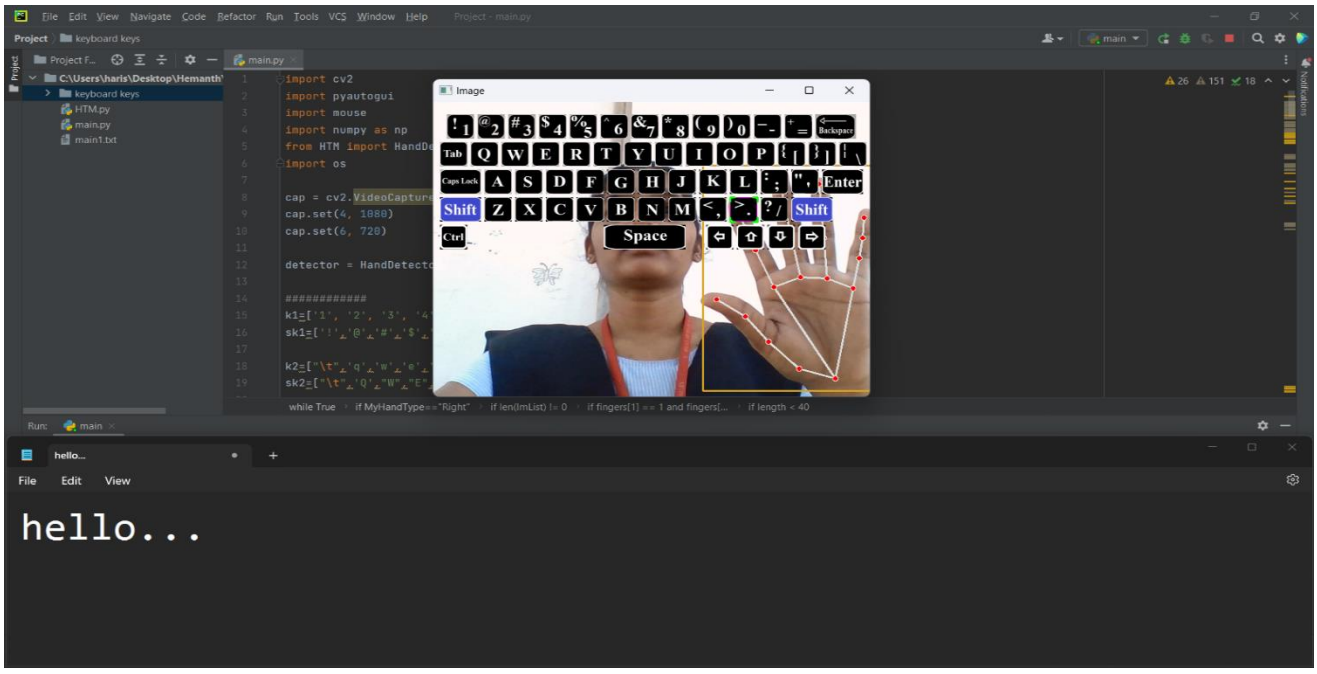


Figure 5.5: Usage of Special Characters

The above Figure 5.5 shows the implementation of using the special characters as well as the wordings to give more results as the layout was taken as reference from the original layout of keyboard.

Snapshot of using Right Hand for Mouse:

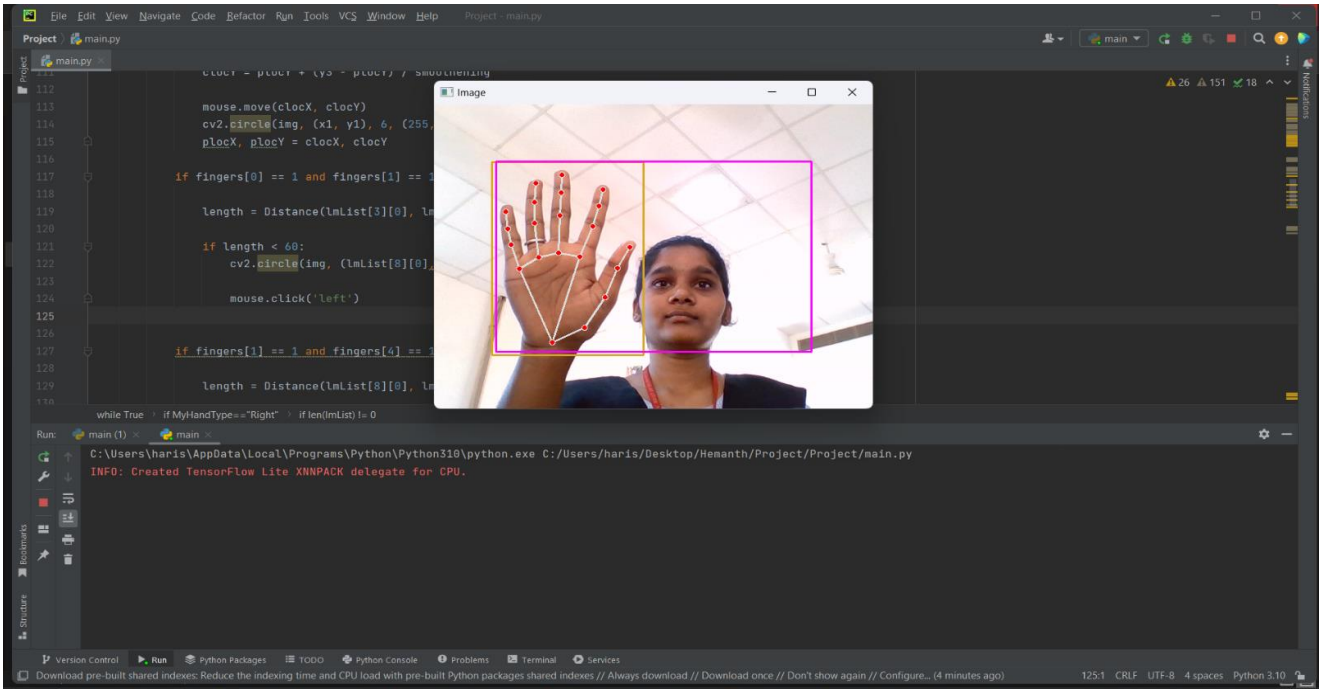


Figure 5.6: Using Right Hand for Mouse

The above Figure 5.6 shows that just by shifting from left hand to right hand of the user they can shift from keyboard to mouse .

Snapshot of using Index Finger for Mouse Pointer:

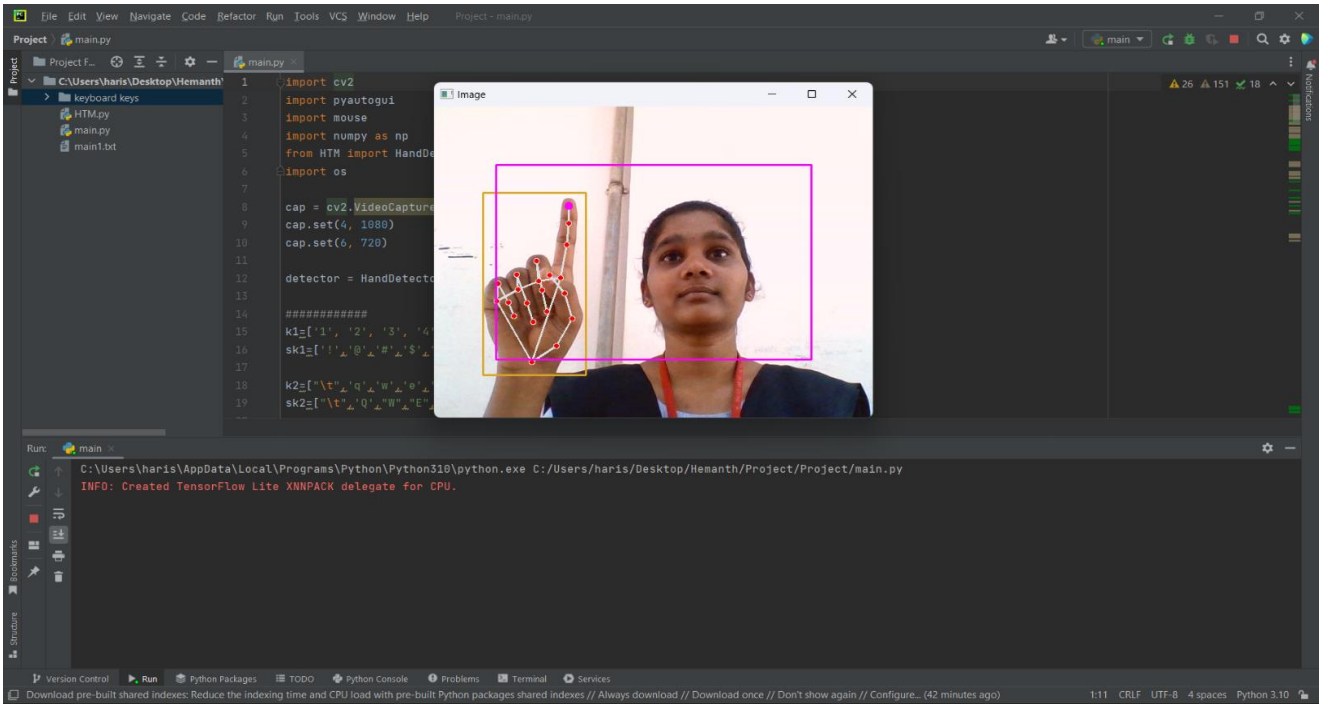


Figure 5.7: Using Index Finger for Mouse Pointer

The above Figure 5.7 shows the usage of mouse pointer where the user has to show index finger to point the place or position of the pointer for the mouse to work.

Snapshot of Drag and Drop operation in Mouse:

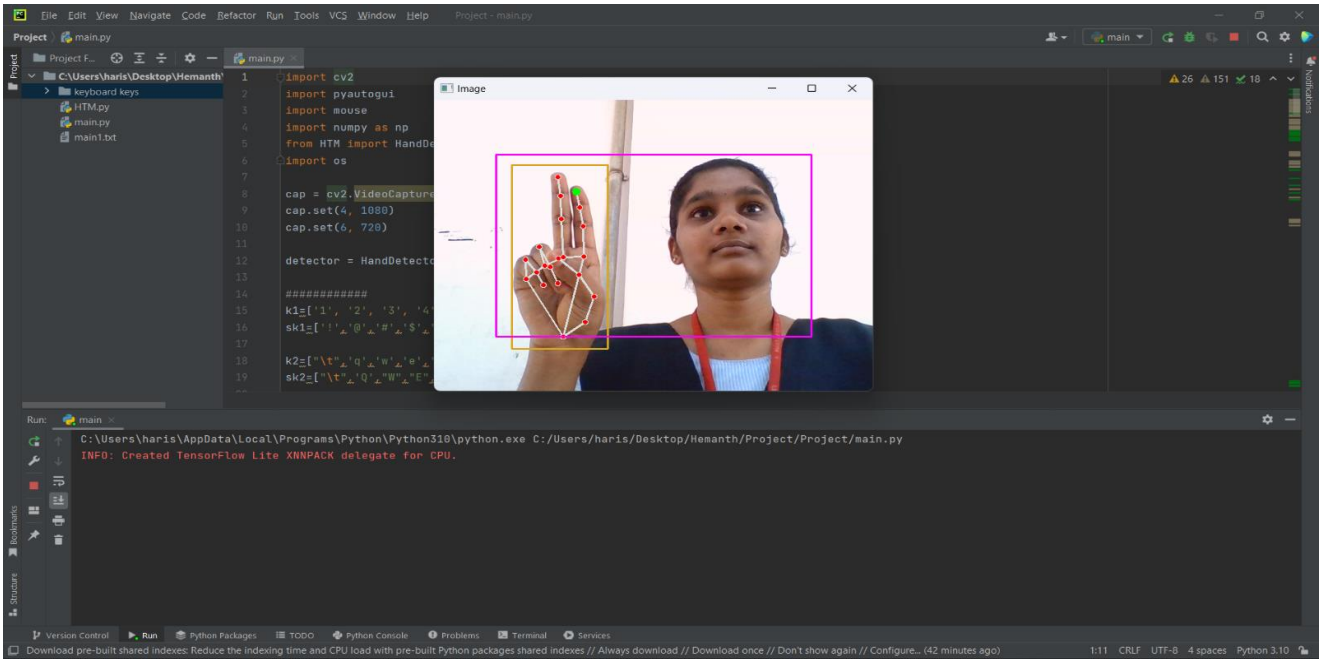


Figure 5.8: Using Both Index Finger and Middle Finger for Drag and Drop Operation

The above Figure 5.8 shows that by using both the index finger and the middle finger the mouse pointer act on drag and drop operation for files or as for selection of content to be dragged .

Snapshot of Right Click operation in Mouse:

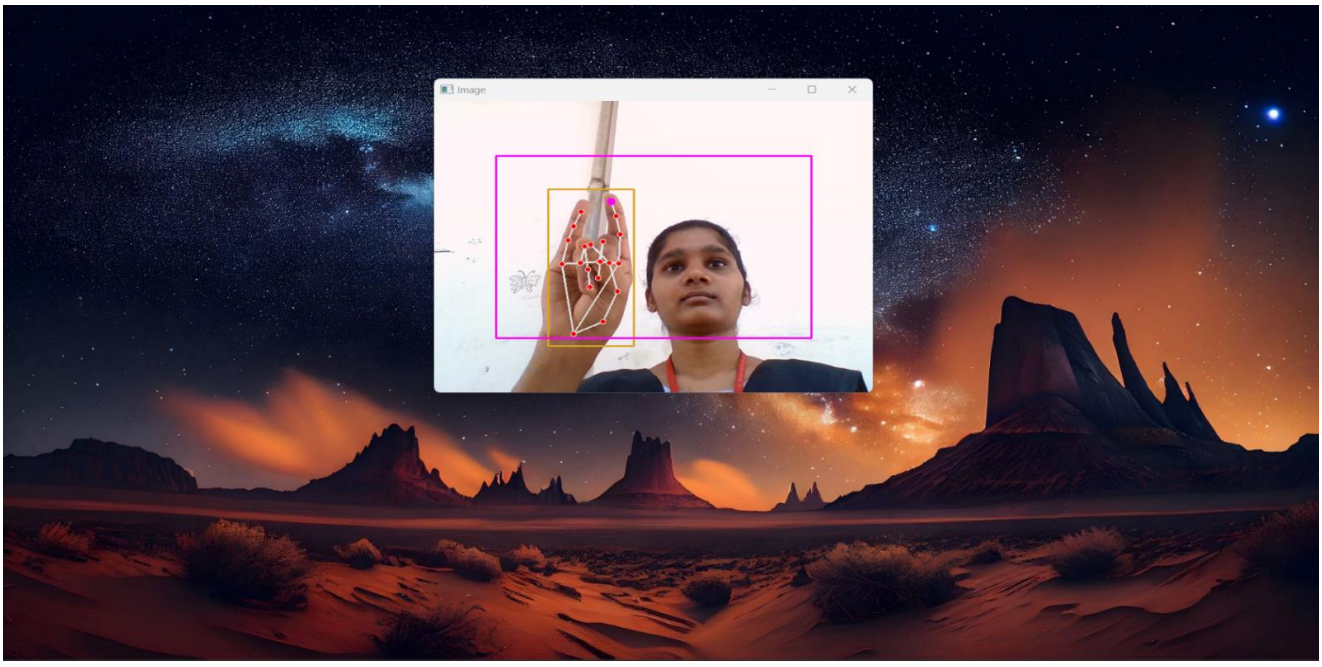


Figure 5.9: Using Both Index Finger and Little Finger for Right Click Operation

The above Figure 5.9 shows that by using both the index finger and the little finger the mouse pointer act on right click operation to either select or rename a file .

Snapshot of Left Click operation in Mouse:

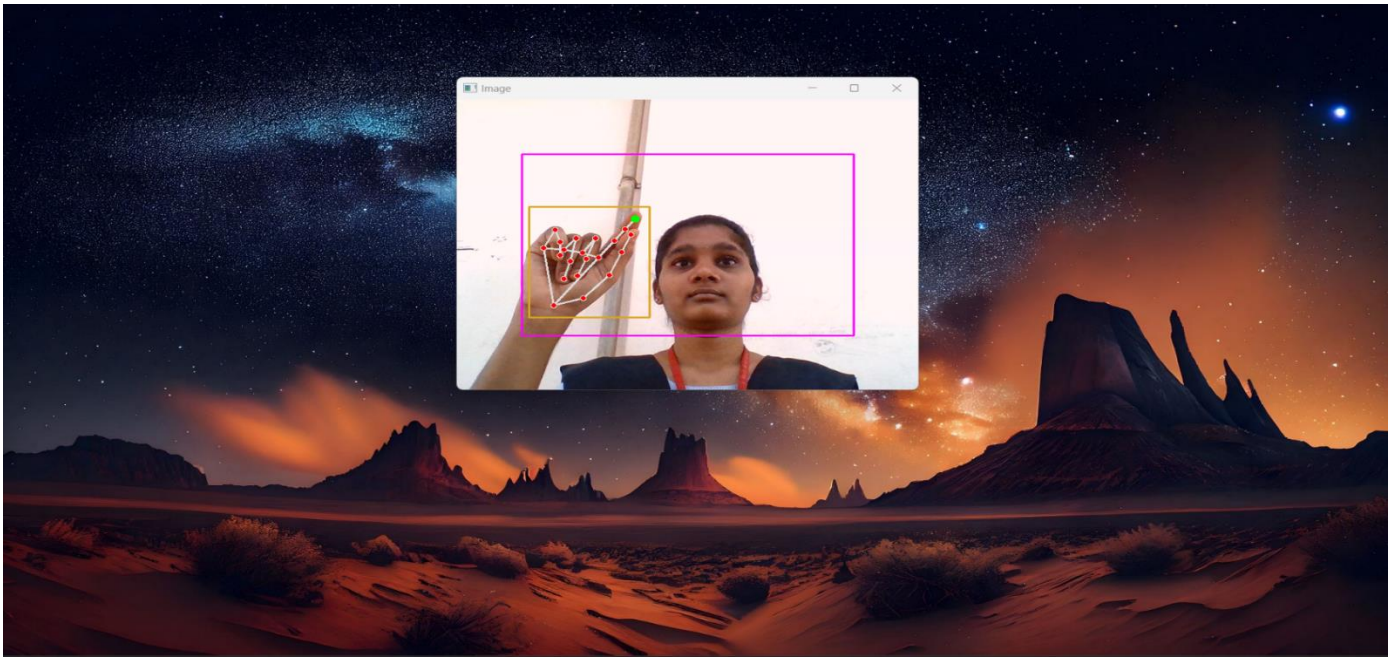


Figure 5.10: Using Both Index Finger and Thumb Finger for Left Click Operation

The above Figure 5.10 shows that by using both the index finger and the thumb finger the mouse pointer act on left click operation to either select or rename a file .

Snapshot of Scroll up operation in Mouse:

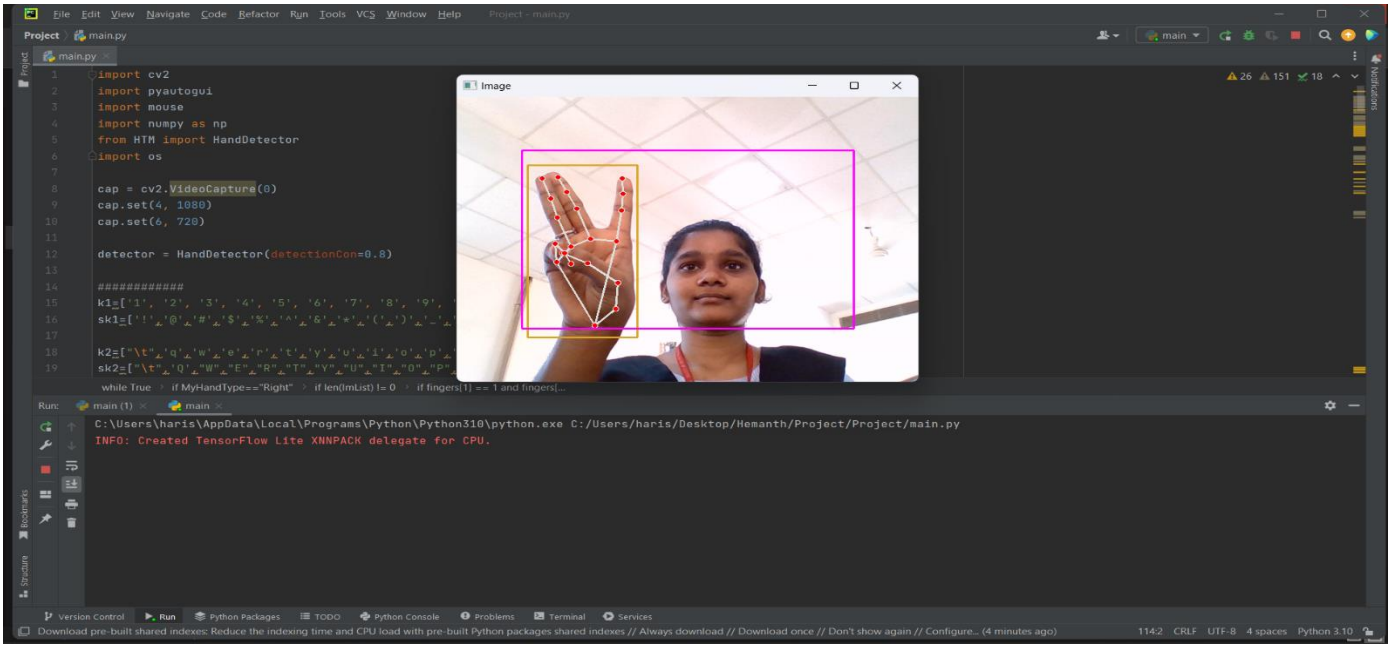
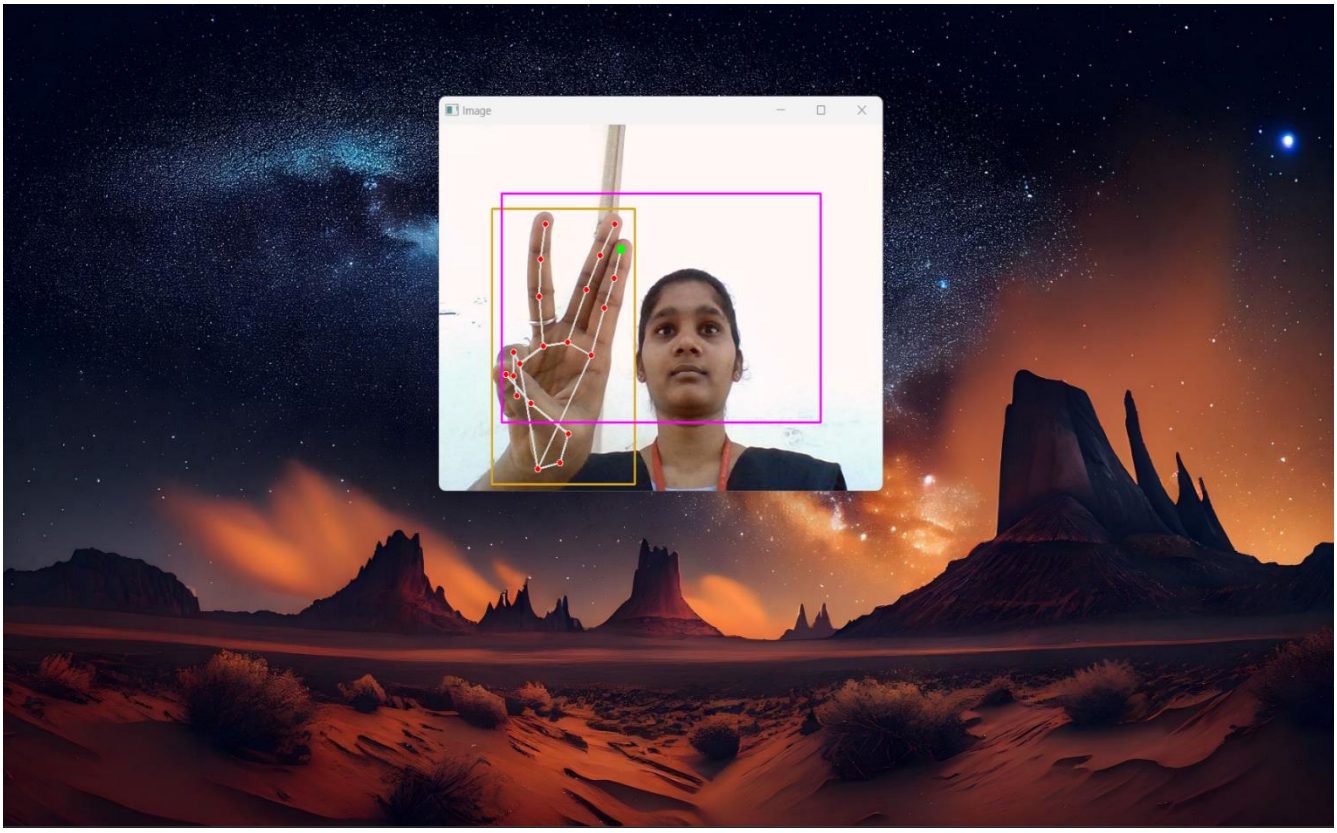


Figure 5.11: Using Index , Middle and Ring Finger for Scroll Up Operation

The above Figure 5.11 shows that using index , middle and ring finger the mouse pointer act on scroll up operation but for this to happen the distance between index and middle finger should be greater than 40 pixel and the ring finger and middle finger distance should be less 40 pixels.



Snapshot of Scroll Down operation in Mouse:

Figure 5.12: Using Index , Middle and Ring Finger for Scroll Down Operation

The above Figure 5.12 shows that using index , middle and ring finger the mouse pointer act on scroll down operation but for this to happen the distance between the ring finger and middle finger should be greater than 40 pixel and the index finger and middle finger distance should be less 40 pixels.

CHAPTER 8

CONCLUSION

In this project, we implemented a virtual keyboard and mouse by using Camera and Image Processing. Implementing a virtual keyboard system gave us a better understanding of the trade one must consider when choosing image analysis techniques in the context of a larger system. We are trying to characterize what typing tasks users currently perform on their touchscreens and then examine whether those tasks can be performed effectively with a virtual keyboard. For mouse analysis, AI virtual mouse system is to control the mouse cursor functions by using the hand gestures instead of using a physical mouse. The proposed system can be achieved by using a webcam or a built-in camera which detects the hand gestures and hand tip and processes these frames to perform the particular mouse functions. From the results of the model, we can come to a conclusion that the proposed AI virtual keyboard and mouse system has performed very well and has a greater accuracy compared to the existing models and also the model overcomes most of the limitations of the existing systems.

FUTURE SCOPE

The hand gesture recognition is moving at tremendous speed for the futuristic products and services and major companies are developing technology based on the hand gesture system and that includes companies like Microsoft, Samsung, Sony and it includes the devices like laptop, hand held devices, professional and led lights. • in the medical fields hand gesture may also be experienced in terms of robotic nurse and medical assistance. As the technology is always revolving and changing the future is quite unpredictable but we have to be certain the future of gesture recognition is here to stay with more and eventful and life touching experiences.

REFERENCES

1. V. Bazarevsky and G. R. Fan Zhang. On-Device, MediaPipe for Real-Time Hand Tracking. https://github.com/google/mediapipe
2. K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, "Realtime computer vision with openCV," Queue, vol. 10, no. 4, pp. 40–56, 2012. https://doi.org/10.1145/2184319.2184337
3. Haria, A. Subramanian, N. Ashok kumar, S. Poddar, and J. S. Nayak, "Hand gesture recognition for human computer interaction," Procedia Computer Science, vol. 115, pp. 367–374, 2017.
4. K. P. Vinay, "Cursor control using hand gestures," International Journal of Critical Accounting, vol. 0975–8887, 2016.
5. J. Katona, "A review of human-computer interaction and virtual reality research fields in cognitive Info Communications" Applied Sciences, vol.11, no.6, https://doi.org/10.3390/app11062646
6. D.-S. Tran, N.-H. Ho, H.-J. Yang, S.-H. Kim, and G. S. Lee, "Real-time virtual mouse system using RGB-D images and fingertip detection," Multimedia Tools and Applications Multimedia Tools and Applications, vol. 80, no. 7, pp. 10473–10490, 2021