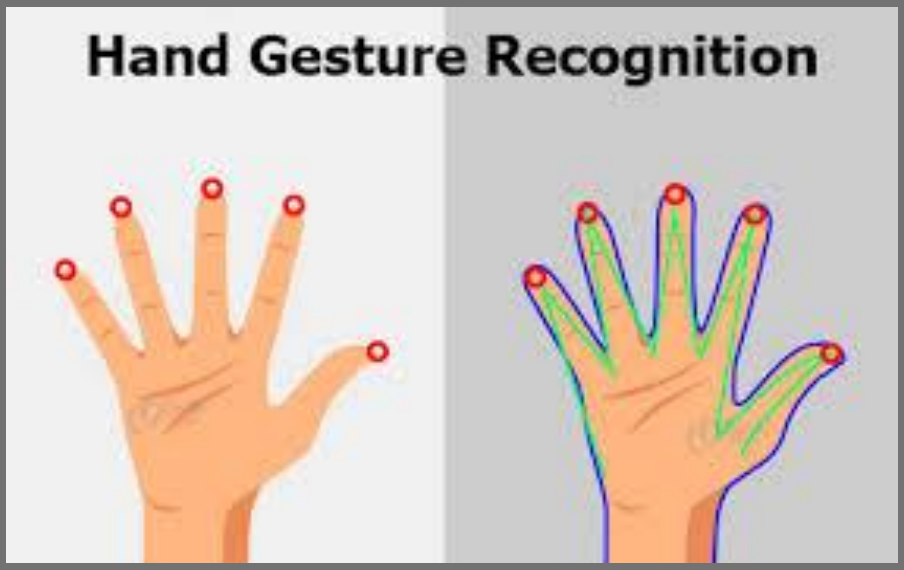
Hand Gesture Recognition



Team Members:
Yuvraj Mangla(102383073)
Paramjeet Singh(102383050)
Yuvraj Singh(102383074)
Joshvir Singh(102383046)
Palak(102383048)

Abbreviations Used

GUI	Graphical User Interface
HCI	Human Computer Interaction
MHI	Motion History Images
IDE	Integrated Development Environment
OpenCV	Open Source Computer Vision
NUI	Natural User Interface

Libraries Used

```
import cv2
import mediapipe as mp
import pyautogui
import math
from enum import IntEnum
from ctypes import cast, POINTER
from comtypes import CLSCTX_ALL
from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume
from google.protobuf.json_format import MessageToDict
import screen_brightness_control as sbcontrol
```

- **CV2: OpenCV-Python** is a library of Python bindings designed to solve computer vision problems.
- **Mediapipe:** It is a cross platform library developed by Google that provides amazing ready-to-use ML solutions for computer vision tasks.
- **Pyautogui: PyAutoGUI** is a cross-platform GUI automation **Python** module for human beings. Used to programmatically control the mouse & keyboard.
- **Math: Python** has a built-in **module** that you can use for **mathematical** tasks. The **math module** has a set of methods and constants.
- **Enum:** Python's enum module provides the Enum class, which allows you to create enumeration types.

Introduction

Hand gesture recognition is a software that allows users to give virtual hand inputs to a system without using an hardware device. To the extreme it can also be called as hardware because it uses an ordinary web camera. This method is usually operated with multiple input devices, which may include an actual mouse or a computer keyboard. Hand gesture recognition uses web camera and works with the help of different image processing techniques.

In this the hand movements of a user are mapped into mouse inputs. A web camera is set to take images continuously. Most laptops today are equipped with webcams, which have recently been used in security applications utilizing face recognition. In order to harness the full potential of a webcam, it can be used for vision-based CC, which would effectively eliminate the need for computer hardware. The usefulness of a webcam can also be greatly extended to other HCI application such as a sign language database or motion controller. Over the past decades there have been significant advancements in HCI technologies for gaming purposes, such as the Microsoft Kinect and Nintendo Wii. These gaming technologies provide a more natural and interactive means of playing videogames. Motioncontrols is the future of gaming and it have tremendously boosted the sales of video games, such as the Nintendo Wii which sold over 50 million consoles within a year of its release. HCI using hand gestures is very intuitive and effective for one-to-one interaction with computers and it provides a Natural User Interface (NUI). There has been extensive research towards novel devices and techniques for cursor control using hand gestures. Besides HCI, hand gesture recognition is also used in sign language recognition, which makes hand gesture recognition even more significant.

Literature Survey

As modern technology of human computer interactions become important in our everyday lives, we can take an example of mouse. Varieties of mouse with all kind of shapes and sizes were invented, from a casual office mouse to a hard-core gaming mouse. However, there are some limitations to these hardware as they are not environment friendly. For example, the physical mouse requires a flat surface to operate, not to mention that it requires a certain area to fully utilize the functions offered. Furthermore, some of the hardware are completely useless when it comes to interact with the computers remotely due to the cable lengths limitations, rendering it inaccessible.

The current system is comprised of a generic mouse and trackpad monitor control system, as well as the absence of a hand gesture control system. The use of a hand gesture to access the monitor screen from a distance is not possible. Even though it is primarily attempting to implement, the scope is simply limited in the Hand gesture recognition field. The existing Hand gesture recognition control system consists of simple mouse operations using a hand recognition system, in which we can control the mouse pointer, left click, right click, and drag, and so on. The use of hand recognition in the future will not be used. Even though there are a variety of systems for hand recognition, the system they used is static hand recognition, which is simply a recognition of the shape made by the hand and the definition of action for each shape made, which is limited to a few defined actions and causes a lot of confusion. As technology advances, there are more and more alternatives to usinga mouse.

A special sensor (or built-in webcam) can track head movement to move the mouse pointer around on the screen. In the absence of a mouse button, the software's dwell delay feature is usually used. Clicking can also be accomplished with a well-placed switch.

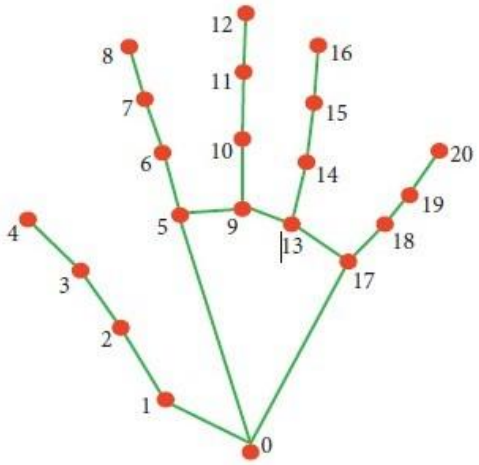
System Development

The various functions and conditions used in the system are explained in the flowchart of the real-time AI Hand gesture recognition system in figure.

Camera Used in the AI Hand gesture recognition System. The proposed AI Hand gesture recognition 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, as shown in Figure. The web camera captures and passes the frames to the AI virtual system.

Capturing the Video and Processing. The AI Hand gesture recognition system uses the webcam where each frame is captured till the termination of the program. The video frames are processed from BGR to RGB colour space to find the hands in the video frame by frame as shown in the following code:

```
def findHands(self, img , draw = True):
imgRGB = cv2.cvtColor(img , cv2.COLOR_BGR2RGB)
self.results = self.hands.process(imgRGB)
```



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

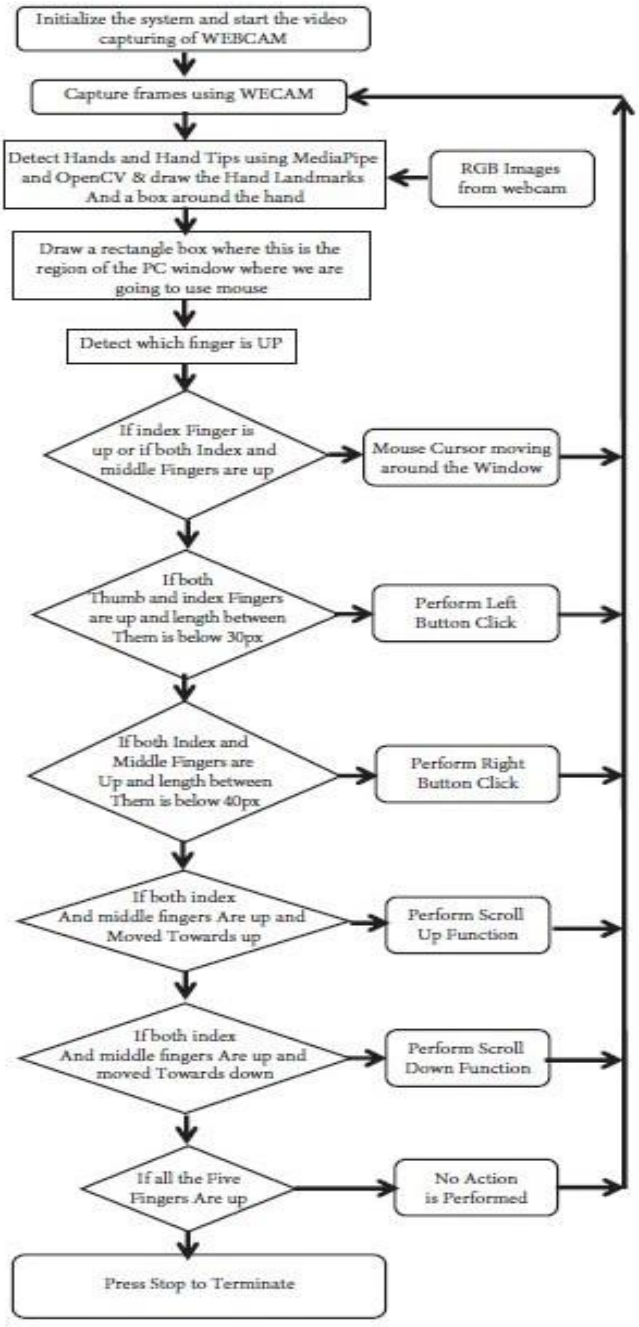
Rectangular Region for Moving through the Window. The AI Hand gesture recognition system makes use of the transformational algorithm, and it converts the coordinates of fingertip from the webcam screen to the computer window full screen for controlling the mouse. When the hands are detected and when we find which finger is up for performing the specific mouse function, a rectangular box is drawn with respect to the computer window in the webcam region where we move throughout the window using the mouse cursor.

Detecting Which Finger Is Up and Performing the Particular Mouse Function. In this stage, we are detecting which finger is up using the tip Id of the respective finger that we found using the **MediaPipe** and the respective co-ordinates of the fingers that are up, and according to that, the particular mouse function is performed.

Mouse Functions Depending on the Hand Gestures and Hand Tip Detection Using Computer Vision for the Mouse Cursor Moving around the Computer Window. If the index finger is up with tip Id = 1 or both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up, the mouse cursor is made to move around the window of the computer using the **AutoPy** package of Python.

For the Mouse to Perform Left Button Click. If both the index finger with tip Id = 1 and the thumb finger with tip Id = 0 are up and the distance between the two fingers is lesser than 30px, the computer is made to perform the left mouse button click using the pynput.

Flow chart of AI Hand gesture recognition

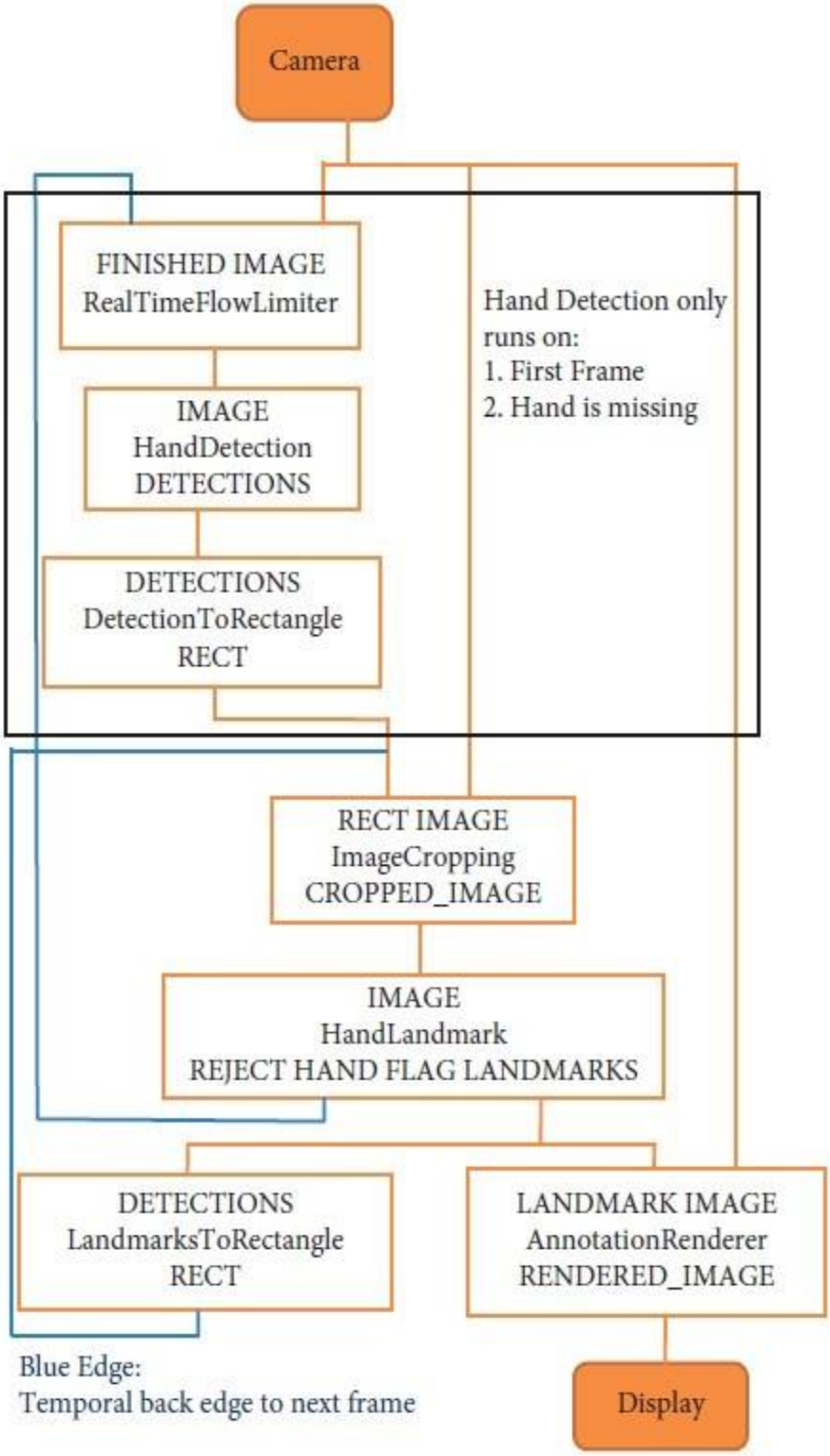


Algorithm and Techniques Used

For the purpose of detection of hand gestures and hand tracking, the MediaPipe framework is used, and OpenCV library is used for computer vision. The algorithm makes use of the machine learning concepts to track and recognize the hand gestures and hand tip.

MediaPipe is a framework which is used for applying in a machine learning pipeline, and it is an opensource framework of Google. The MediaPipe framework is useful for cross platform development since the framework is built using the time series data. The MediaPipe framework is multimodal, where this framework can be applied to various audios and videos. The MediaPipe framework is used by the developer for building and analyzing the systems through graphs, and it also been used for developing the systems for the application purpose. The steps involved in the system that uses MediaPipe are carried out in the pipeline configuration. The pipeline created can run in various platforms allowing scalability in mobile and desktops. The MediaPipe framework is based on three fundamental parts; they are performance evaluation, framework for retrieving sensor data, and a collection of components which are called calculators, and they are reusable. A pipeline is a graph which consists of components called calculators, where each calculator is connected by streams in which the packets of data flow through. Developers are able to replace or define custom calculators anywhere in the graph creating their own application. The calculators and streams combined create a data-flow diagram; the graph is created with MediaPipe where each node is a calculator and the nodes are connected by streams.

Single-shot detector model is used for detecting and recognizing a hand or palm in real time. The single-shot detector model is used by the MediaPipe. First, in the hand detection module, it is first trained for a palm detection model because it is easier to train palms. Furthermore, the non-maximum suppression works significantly better on small objects like palms or fists. A model of hand landmark consists of locating joint or knuckle co-ordinates in the hand region,
OpenCV is a computer vision library which contains image-processing algorithms for object detection. OpenCV is a library of python programming language, and real-time computer vision applications can be developed by using the computer vision library. The OpenCV library is used in image and video processing and also analysis such as face detection and object detection.


Camera
FINISHED IMAGE
RealTimeFlowLimiter
IMAGE
HandDetection
DETECTIONS
DETECTIONS
DetectionToRectangle
RECT
Hand Detection only
runs on:
1. First Frame
2. Hand is missing
RECT IMAGE
ImageCropping
CROPPED_IMAGE
IMAGE
HandLandmark
REJECT HAND FLAG LANDMARKS
DETECTIONS
LandmarksToRectangle
RECT
LANDMARK IMAGE
AnnotationRenderer
RENDERED_IMAGE
Blue Edge:
Temporal back edge to next frame
Display

Performance Analysis

In the proposed AI Hand gesture recognition system, the concept of advancing the human- computer interaction using computer vision is given.
Cross comparison of the testing of the AI Hand gesture recognition system is difficult because only limited numbers of datasets are available. The hand gestures and fingertip detection have been tested in various illumination conditions and also been tested with different distances from the webcam for tracking of the hand gesture and hand tip detection. An experimental test has been conducted to summarize the results shown in the Table.

The test was performed 25 times by 4 persons resulting in 600 gestures with manual labelling, and this test has been made in different light conditions and at different distances from the screen, and each person tested the AI Hand gesture recognition system 10 times in normal light conditions, 5 times in faint light conditions, 5 times in close distance from the webcam, and 5 times in long distance from the webcam, and the experimental results are tabulated in Table.

Mouse function performed	Success	Failure	Accuracy(%)
Mouse movement	100	0	100%
Left button click	98	2	98%
Right button click	99	1	99%
Scroll function	93	7	93%
Brightness control	95	5	95%
Volume control	96	4	96%
No action performed	100	0	100%
Result	681	19	97.28%

From the Table, it can be seen that the proposed AI Hand gesture recognition system had achieved an accuracy of about 97%. From this 97% accuracy of the proposed AI Hand gesture recognition system, we come to know that the system has performed well. As seen in Table, the accuracy is low for “Scroll function” as this is the hardest gesture for the computer to understand. The accuracy for scroll function is low because the gesture used for performing the particular mouse function is harder. Also, the accuracy is very good and high for all the other gestures. Compared to previous approaches for Hand gesture recognition, our model worked very well with 97% accuracy. The graph of accuracy is shown in Figure:

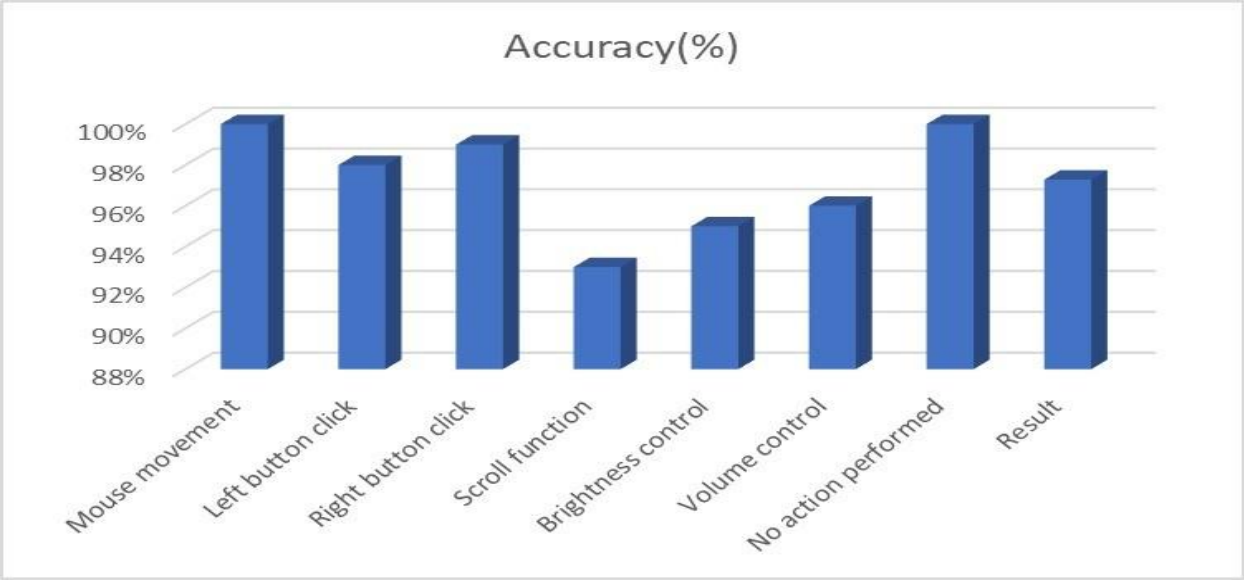
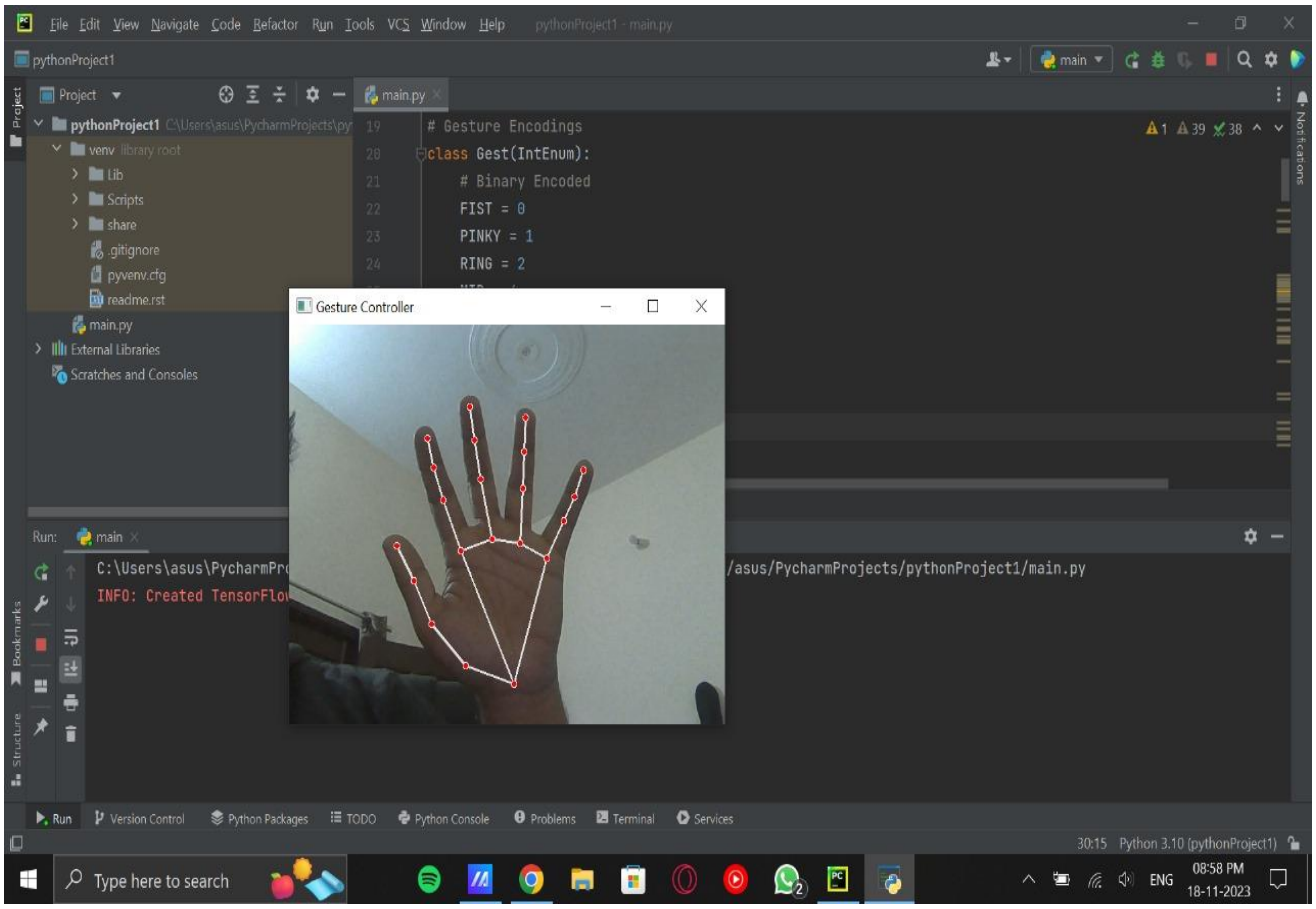


Image of our Working Project



Conclusion and Future Scope

Due to accuracy and efficiency plays an important role in making the program as useful as an actual physical mouse, a few techniques had to be implemented. After implanting such type of application there is big replacement of physical mouse i.e., there is no need of any physical mouse. Each & every movement of physical mouse is done with this motion tracking mouse (Hand gesture recognition).

There are several features and improvements needed in order for the program to be more user friendly, accurate, and flexible in various environments. The following describes the improvements and the features required:

a) Smart Movement: Due to the current recognition process are limited within 25cm radius, an adaptive zoom in/out functions are required to improve the covered distance, where it can automatically adjust the focus rate based on the distance between the users and the webcam.

b) Better Accuracy & Performance: The response time are heavily relying on the hardware of the machine, this includes the processing speed of the processor, the size of the available RAM, and the available features of webcam. Therefore, the program may have better performance when it's running on a decent machine with a webcam that performs better in different types of lightings.

c) Mobile Application: In future this web application also able to use on Android devices, where touchscreen concept is replaced by hand gestures.

Test Cases

Test case id	Scenario	Boundary Value	Expected Result	Actual Result	Status
1	Used in normal environment.	>90%	In normal environment hand gestures can be recognized easily.	Hand gestures got easily recognized and work properly.	**Passed**
2	Used in bright environment.	>60%	In brighter environment, software should work fine as it easily detects the hand movements but in a more brighter conditions it may not detect the hand gestures as expected.	In bright conditions the software works very well.	**Passed**
3	Used in dark environment	<30%	In dark environment, It should work properly.	In dark environment software didn't work properly in detecting hand	**Failed**

				gestures.	
4	Used at a near distance (15cm) from the web cam.	>80%	At this distance, this software should perform perfectly.	It works fine and all features works properly.	**Passed**
5	Used at a far distance (35cm) from the web cam.	>95%	At this distance, this software should work fine.	At this distance, it is working properly.	**Passed**
6	Used at a farther distance (60cm) from the web cam.	>60%	At this distance, their will be some problem in detecting hand gestures but it should work fine.	At this distance, The functions of this software works properly.	**Passed**

References

1) OpenCV Website – www.opencv.org

2) MSDN Microsoft developers network – www.msdn.microsoft.com