CHAPTER 1
INTRODUCTION

1.1 Problem Statement

In today's modern era when the technology is developing at highest speed, people are moving towards the technology to get their work done easily and save as much of their time as possible.
But the use of such technology is not possible for everyone. The older generation still are learning how to use smartphones, which makes it almost impossible for them to use laptop or PCs for their personal work. This makes them work for much more time or use old classic methods to do their work which requires lots of work and so much hassle. There are also many other people than old people who still can't use the computers. Such people seek help from others or a simple way for them to use PC's and Laptops, so they can use them for some basic works.

Computer Vision, Machine Learning and Gesture Free technology are the new era of the technology. Most of the markets and people are moving towards them because they make the work easy, efficient, saves time and are very easy to use. These touchless technology enables people to use advanced technology with some basic gestures which makes it so much easy to use that almost everyone can use it. Computers which can work with only hand gestures makes them easy to use for new people and makes Computer/PC's more user friendly.

So, recently there have been few approaches towards touchless use of computer using computer vision and gesture control but they have only been limited to one use like – either they can work to move pointer, it can play pause the music only or it can increase/decrease the volume.

So, with all these limitations in mind we developed a software which helps people to simple tasks with only hand gestures making it very easy to use by anyone which can help you play, pause, seek forward/backward a video, save an opened file, lock the computer screen etc., all with just Hand Gestures.

1.2 Motivation

1) As in the modern era computers are becoming more advanced and capable of doing more processing as compared to earlier computers. The hand gesture will allow us leverage in controlling the computer as there is less contact and more effectiveness.
2) With gestures we can operate the computer from farther distances, making it is much easier to use and user friendly.
3) Computers can be easy to use with gestures for new people, making them more user friendly.
4) And as the world is advancing to the era of technology and IoT, making a software to make computers work from hand gesture is a great approach to make computers easy to use and increase the productivity.
5) After going through the literature and seeing projects on gesture recognition, we found that there has not been any significant project built on controlling the Computer using hand gesture.
6) Making a Hand gesture controller for computer will be of great benefit to make the technology more usable by everyone.

1.3 Project Overview

We are developing a software which will recognize our hand gestures through the computer vision and with the help of neural networks and different python libraries and performs different tasks related to them. It will check which gesture is being done by the user and then performs that gesture.

This software is based on computer vision and AI/ML and can perform the given tasks –

- Play/Pause – A Video or Song.
- Close current window.
- Save the currently opened file.
- Volume up/down, mute/unmute.
- Zoom in/out.
- Lock windows.
- Minimize/Maximize all opened windows.

Anyone can just run the software in background allowing it to use the camera and just simply use hand gestures to perform these different tasks in any laptop which have a camera in it. Making this software to be used in almost any laptop and making it very easy to use and it doesn't require much hardware requirements.

Therefore, we bring a unique, easy to use, simple and user-friendly software which can be use by anyone or any person new to technology or computers. It will help old people as well as people with no technology exposure to use Computer/Pc's with just their hand gestures.

Fig. 1.3.1 Play/Pause with Open Hand Gesture through Webcam

1.4 Objectives

- To be able to use multiple apps, media players or even give presentations with just a wave and Gestures of your hand. (No mouse or remote controllers needed making the person to use their hand freely and express themselves more freely when speaking.)
- People who have difficulty using the hardware can use the hand gesture controller to easily navigate and use the computer with the help of their hands.
- To be able to control the computer from farther distances and while doing other work like - Cooking, Workout, drawing etc.

1.5 Expected Outcome

The project uses the computer vision on Laptops/PC's to perform different actions by recognizing the hand gestures. Making the software easy to use by anyone. To encourage people to use technology easily without any trouble.

This software can utilize the webcam of PC's and allow users to use their hands for gesture control. The software can recognize different gestures and perform actions related to the gestures (like – play, pause, next page etc.). It will be able to use multiple apps, media players or even give presentations with just a wave and Gestures of your hand.

This software can recognize static as well as dynamic gestures which makes it to use gestures that are more natural to human beings. This makes the software more user friendly and it will encourage people to use this software in their day-to-day routine. Just allows users to work efficiently, give presentation without any remote requirement, do housework, and change music with just a wave of hand.

So, overall, it's an innovative software solution which helps people to use computer without touching it with just their hand gestures.

1.6 Hardware and Software Requirements

Hardware Requirements-

- Webcam/Camera (with access)
- Enough Memory space to install software.
- Camera specs:
    - Camera resolution should be 480p or more.

Camera angle:

- Normal camera has a field of view of 78 degrees.
- The camera angle should be on hand level (in a way that it captures the whole hand movement).

Software Requirements-

- Windows 7 or higher
- Python

1.7 Report Outline

Chapter 2 talks about the related technology and uses of gesture recognition in different products and industries which make use of Gesture Recognition and AI.

Chapter 3 focuses on the proposed model, how it proceeds, what are the requirements, who are the users and what all will be the methodology to build our solution.

Chapter 4 provides the results and experimental analysis of our system.

Chapter 5 testing the proposed system and getting the test results.

Chapter 6 Concluding the paperwork and what's the future scope of proposed system.

CHAPTER 2
LITERATURE SURVEY

2.1 Existing Gesture Controlling Applications

With so much advancement in the technology and AI/ML, there are lots and lots of gesture recognition applications. But there is only 1 official application which performs action after recognizing those gesture, which is named as "ZESTURE" which also does not provide many gestures to be recognized.

There are lot of uses in today's world where Gesture recognition is used. Some of them are –

1. **<u>Video Games</u>** – There are lot of games in the market which uses hand gesture recognition to play the games and make users interact more with gesture recognition instead of playing it with controller.
2. **<u>Automobiles</u>** – There are smart cars in the market which has gesture enabled functionality to make it safer for the driver to use media with just their hands.
3. **<u>Drone Controllers</u>** – Some drone companies have developed the gloves which is based on gestures through which people can control the drones.
4. **<u>Medical Operations</u>** – In medical field, doctors are using gesture enabled devices which help them perform tasks with their hands while performing surgery or operations.
5. **<u>Smart Homes/Home Automation</u> –** Smart homes are the new way the homes are being built with technology making them use gestures and voice to turn on/off lights or electric appliances.
6. **<u>Virtual Reality</u>** – Gesture interaction plays important part in the VR. The gesture interaction helps users to interact in VR easily and in more natural way.
7. **<u>Consumer Electronics</u>** – Consumer electronics like – smart watches etc., are also getting gesture recognition features included in them for better user experience.

2.2 Existing Gesture Recognition Models

Table 2.2.1 Pre-existing Gesture Recognition Model

Author	Title	Year	Methodology	Remarks
Hasan, Haitham, et al.	A survey on human-computer interaction technologies and techniques	2016	It focuses on several application fields that use hand gestures for efficient interaction and summarizes surveys conducted in human–computer interaction (HCI) studies.	This paper reviews several other papers based on Hand gesture HCI and provides all information regarding the HCI tools & methodology necessary.
Molchanov, Pavlo, et al.	Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks.	2016	Use Recurrent 3D Convolutional Neural Networks(R3DCNN), NVIDIA dataset, 3DCNN for Spatio-temporal feature extraction, SoftMax layer for gesture probabilities and RNN for global temporal modelling.	This gesture recognition system attains 83.8% accuracy & also beats the perfect algorithms with 88.4% human accuracy.
G. Li, et al.	Hand gesture recognition based on convolutional neural networks.	2017	Uses CNN for gesture recognition to avoid feature extraction and reduce the number of parameters for training. Also uses Support Vector Machine (SVM).	CNN gives 98.52% of accuracy rate. LSTM is better than CNN in longer dependence while CNN uses different feature maps to capture several times.
Joshi, Anand.	Real Time Monitoring of CCTV Camera images using Object detectors and scene classification for retail and surveillance applications.	2017	Uses Inception-ResNet-v2 CNN. Uses datasets of Knives, Firearms, EgoHands, Hand and ImageNet. Evaluates: AlexNet & Caffe, Google Net & TensorFlow Inception-ResNet-V2 and TensorFlow	Approves Inception-ResNet-v2 with TensorFlow to be the best among these un terms of accuracy and analysis of models. It detected with accuracy and analysis of models. It detected with accuracy the images of interest.
H.Jiang, et al.	Face Detection with the Faster R-CNN	2017	Proposes Faster R-CNN for face detection on WIDER, FDDB, & IJB-A face dataset. While using the ImageNet model and VGG16 classifiers.	The best performance is detected in the WIDER dataset. Effectiveness of Faster R-CNN is due to Region Proposal Network (RPN) module.

Xu, Pei.	A real-time hand gesture recognition and human-computer interaction system.	2017	Uses only one monocular camera, CNN classifier and Kalmann Filter to estimate position of cursor w.r.t the point of hand detector.	This system supports only the static gestures, not the dynamic ones. Uses Kalmann filter to smoothen the cursor motion. And the use of CNN reduces the manual work of extracting features.
A. A. Alani, et al.	Hand gesture recognition using an adapted convolutional neural network with data augmentation	2018	Uses ADCNN (Adapted Deep Convolutional Neural Networks) to enhance CNN's performance for Hand Gesture Recognition	Uses 3750 images to train & test the model. ADCNN has 99.73% accuracy whereas CNN shows 95.73% accuracy.
Z.Zhou, et al.	Image Segmentation Algorithms based on Convolutional Neural Networks	2018	CNN for feature extraction and RCNN & OverFeat for object recognition. AlexNet as well as SVM model is used.	A few image segmentation algorithms are implemented. BP and CNN can be improved, increasing the structural risk. Accuracy has improved from 97.94% to 98.12%
Y Chao, et al	Rethinking the Faster R-CNN Architecture for Temporal Action Localization	2018	Proposes Tal-Net (Temporal Action Localization) to be in use rather than Faster R-CNN.	Tal-Net performs better than Faster R-CNN methods. Late feature fusion, context feature extraction and improved receptive alignment of field.
D. Nguyen, et al.	Hand segmentation under different viewpoints by combination of Mask R-CNN with tracking.	2018	Proposes to use Mask R-CNN for Hand segmentation on Microsoft COCO dataset, and MeanShift for Hand Tracking.	Though it uses Mask R-CNN, it explores temporal information of the hand motion & also tracks it. Depends on the camera image views. Improvement of 5% - 9%, more than the normal Mask R-CNN.

M.Dani, et al	Mid-Air Fingertip-Based User Interaction in Mixed Reality.	2018	SCUT-Ego-Finger benchmark Dataset is used. YOLOv2, Faster R-CNN & regression CNN models for training. Here, MobileNet is also used.	YOLO V2 shows 98.9% accuracy followed by Faster RCNN with VGG16 showing 98.1%.
Köpüklü, Okan, et al.	Real-time hand gesture detection and classification on using convolutional neural networks.	2019	CNN for gesture recognition and 3D CNN for analysis of video. Dynamic hand gesture datasets are Ego Gesture & Nv Gesture. Also uses ResNext-101 classifier.	They were using the levenshtein metric, giving 91.04% Ego Gesture and 77.39% Nv Gesture accuracies.
S., Rubin Bose, et al.,	Hand Gesture Recognition using Faster R-CNN Inception V2 Model.	2019	Uses Faster R-CNN with Inception v2 architecture, ADAM, Momentum and RMSprop optimizer.	Learning rate of ADAM & Momentum is 0.0002 and 0.0004 for RMSprop. ADAm is a better optimizer with avg. precision 0.794, avg. recall 0.833, F1 score 0.813.
W.B.Dou, et al.	Hand Gesture Communication using Deep Learning based on Relevance Theory.	2020	It makes sense of the Gestures based on the surroundings. Uses Mask R-CNN for feature detection and extraction of gestures & objects; Declarative Memory Recurrent Neural (DMRN) Model to keep learning and make sense of the gestures based on objects.	The results of this system show that such a system would work well only when trained well. The accuracy and precision of the result isn't mentioned.

M. Srividya,et al.	Hand Recognition and Motion Analysis using Faster RCNN.	2019	It is a Faster R-CNN based model using VGG16 classifier. Uses a CNN based detector. Trained on ImageNet Dataset and own dataset of guns of 3000 images.	Faster RCNN has the best performance in database 5 with recall 100% & F1 score of 91.43%. Best potential to detect the object was in low quality YouTube videos.
G.Lingyun, et al.	Hierarchic al Attention-Based Astronaut Gesture Recognition: A Dataset and CNN Model.	2020	Uses Mask R-CNN for image segmentation, Hierarchical Attention Single-Shot Detector Network (HA-SSD) for gesture recognition, MobileNet for feature extraction & DSSL Astronaut Gesture Dataset	YOLO V3 detects the hands very well in DSSL, but on a single DSSL it is terrible. HA=SSD could be improved using the deburred and super-resolution model for image processing & to have better performance.
Virender Singh, et al.	Real-time anomaly recognition through CCTV using neural networks.	2020	They have trained several models increasing the anomalies each time with more and varied datasets and real-life scenarios. It uses Inception v3 CNN and then RNN.	The best model classifies 13 anomalies; it has Adam optimizer, joining 8 frames forming a chunk; Errorfunction-categorical_crossentropy; Activation Function: Sigmoid, SoftMax & Relu. It provides the best accuracy of 97.23%.
Shrey Srivastava, et al.	Comparative analysis of deep learning image detection algorithms.	2021	It compares the 3 image processing algorithms: SSD, Faster R-CNN, YOLO using Microsoft COCO dataset based on their performances, strengths, limitations. Analysis is based on the parameters of precision accuracy and F1 score.	YOLO V3, SSD and then Faster R-CNN; performance is ranked in this order respectively. Best performances include YOLO V3 - To analyze a live video feed; SSD - ensures a good balance of accuracy & speed. Faster R-CNN - not for real-time results, needs small datasets.

X. Gao, et al.	Accuracy Analysis of Triage Recommendation Based on CNN, RNN and RCNN Models	2021	Proposes to analyze & compare CNN, RNN and RCNN on 20,000 texts of patient symptom descriptions (in Mandarin). Uses CBOW model for word vectors.	RCNN attains the highest accuracy of 76.51% while comparing it to RNN and CNN. Take longer time for execution due to time complexity & heavy computation. Gao, et al., intend to reduce the complexity
Mujahid, Abdullah, et al.	Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model.	2021	Uses YOLO v3 and DarkNet53 as CNNs for Hand gesture recognition. Later compares them with the SSD (Single Shot Detector) model and VGG16 (Visual Geometry Group).	Attains the accuracy of 97.68, precision of 94.88, recall of 98.66, and an F-1 score of 96.70%. In comparison with SSD & VGG16, accuracy between 82 - 85 is attained.

Limitations of previous applications -

- ZESTURE application just performs action on multimedia player. No interaction with the computer and very limited gestures.
- Other relative application only recognizes static gestures.
- No application for computer which provides more than 5 gestures.

2.3 Proposed System

The proposed software is named “HandMotion Pro”, which is the hand gesture control software for everyone to use in daily life. It provides different workings like –

- Play/Pause – A Video or Song.
- Close current window.
- Save the currently opened file.
- Volume up/down, mute/unmute.
- Zoom in/out.
- Lock windows.
- Minimize/Maximize all opened windows.

Development of the project is done in the python language with the help of different tools and libraries like TensorFlow, Mediapipe etc. It takes live video input from the camera of the PC/laptop, matches the current gesture being made by the user and then performs the task associated with it. Our approach first builds a rectangle around hand and then captures the points location with help of Mediapipe, then we have normalized the distance and convert the point in 2D from 3D and then with the data we trained our model to tell the different gestures and then we integrated different computer functionalities with the model to perform actions by recognizing different gestures.

2.4 Feasibility Study

As previously discussed, there are only very few applications that perform actions by using the computer vision and recognizing hand gestures, but these apps either recognize static hand gestures or perform very functionalities. These apps don't provide large functionality which makes them very less useful and not so user friendly. The 'HandMotion Pro' tries to provide both static as well as dynamic gestures and provide many functionalities. Some of its features are:

- ✓ Simple to use => It is very simple to use, just open the app and let it run in the background.
- ✓ User Friendly => It is very user friendly as it recognizes the most common gestures and help you use computer with just hand gestures.
- ✓ No Hardware Requirements => Our app doesn't need any additional hardware to work, it just needs access to the webcam of the laptop.
- ✓ No internet requirement => Our app doesn't require any internet to work, it completely works offline.
- ✓ Dynamic Hand Gesture Recognition => Our app even recognize dynamic hand gesture movements.
- ✓ Large Number of Gestures => Our app provides 14 gesture recognition including both dynamic and static hand gestures.

Thus, in this way, this apps overcomes the problems faced by the previous applications, so it can be the new all-in-one go to app for everyone who wants to use computer easily and in touchless manner.

CHAPTER 3:
SYSTEM DESIGN AND ANALYSIS

3.1 Software Development Life Cycle/ Model:

We have chosen to develop our software through the **Agile Model.**

The advantages of the Agile Model are as follows −

- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements.
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required.
- Easy to manage.
- Gives flexibility to developers.

Fig. 3.1.1 Agile Model

3.2 Methodology

3.2.1 Flow Diagram – Working of Hand Gesture Recognition Model

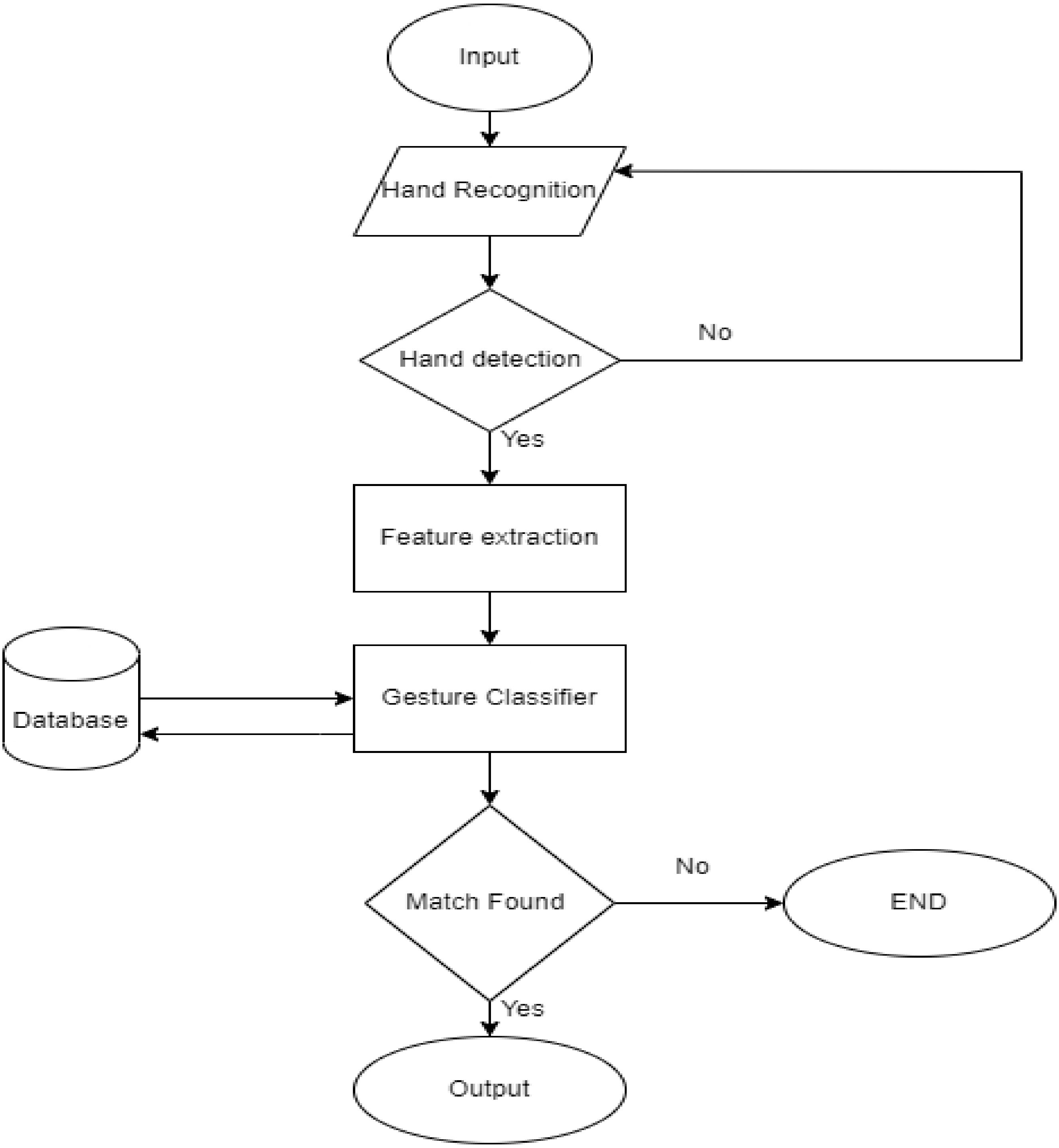

Fig. 3.2.1.1 Hand Gesture Recognition Model Flowchart

3.2.2 Implementation:

- First step is to initialize the camera and start capturing frames. These frames can then be passed through the Mediapipe hand tracking module, which will identify the hand's key points and generate a 3D hand pose estimation.
- Using the 3D hand pose estimation, OpenCV can be used to extract features from the hand, such as the position of the fingers and the orientation of the palm.
- Then the 3D model was normalized to 2D.
- We took wrist point as the origin and we reduce it from other hand point, and then convert them to in range of 0-1.
  wrist point = (600,300) is taken as origin
  Max value = 250
  thumb tip = (350,200)
  After normalization => (350-600), (200-300)
  => -220, -100
  => -1, -0.4

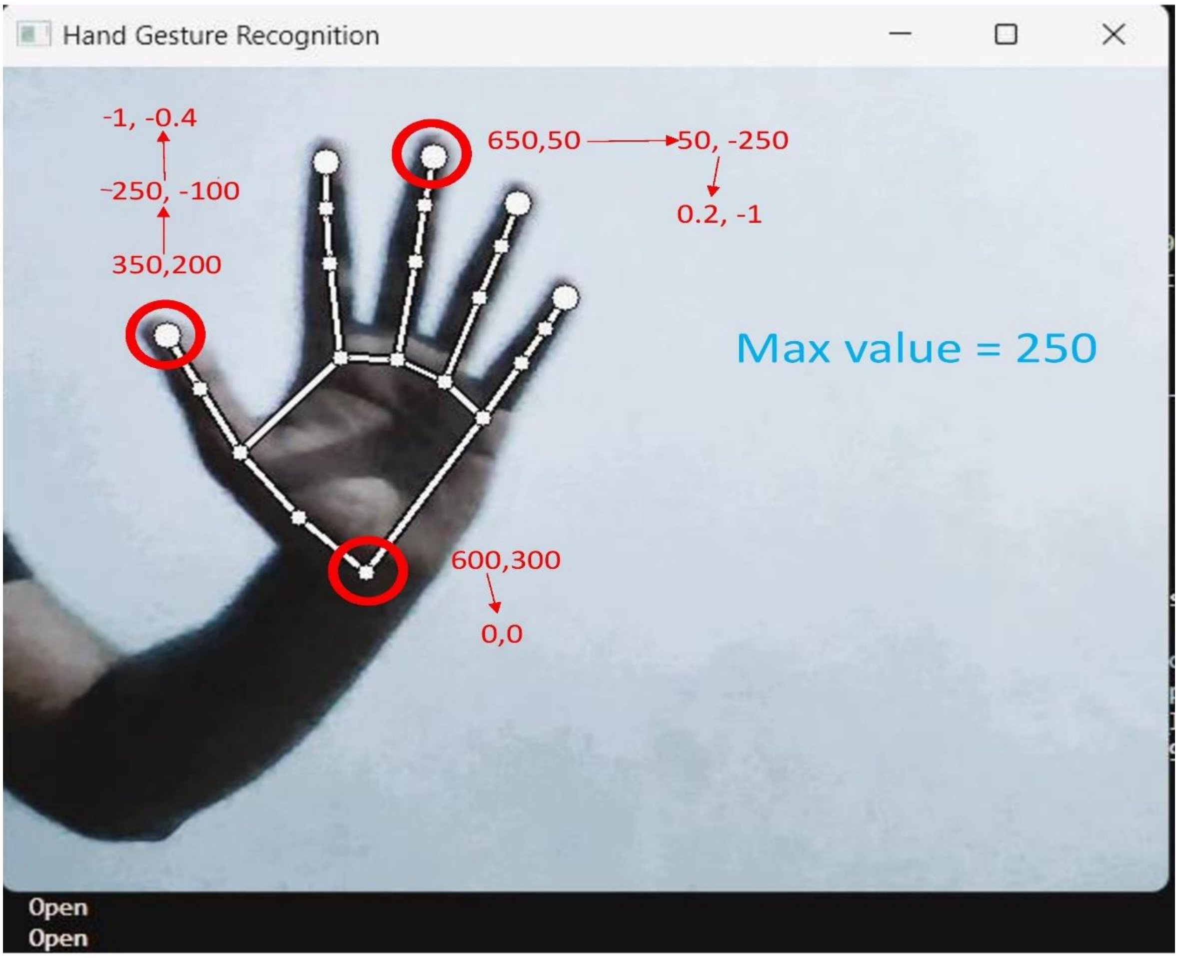

Fig. 3.2.2.1 Normalization of the hand points

- Then, these normalized values are used to record different gestures and create dataset, and then a model is trained to identify these gestures.
- Once the recognised gesture is made the most in span of 2 sec is considered as the gesture being made by the user, then the related OS function is performed.

3.3 Different Libraries and Techniques Used

3.3.1 Mediapipe

This module deals with the recognition of hand movements in the live image.

- We have used Mediapipe hands.
- Hand recognition model is divided into two parts -
  - Palm detection model
  - Hand landmark model

One of the most popular libraries for hand gesture recognition is Mediapipe, a cross-platform framework for building multimodal applied ML pipelines. Mediapipe provides a wide range of pre-built modules for hand tracking, pose estimation, face detection, and more. It also supports real-time processing, making it ideal for applications that require low latency, such as gaming. In addition to Mediapipe, OpenCV is another popular library for computer vision that can be used for hand gesture recognition.

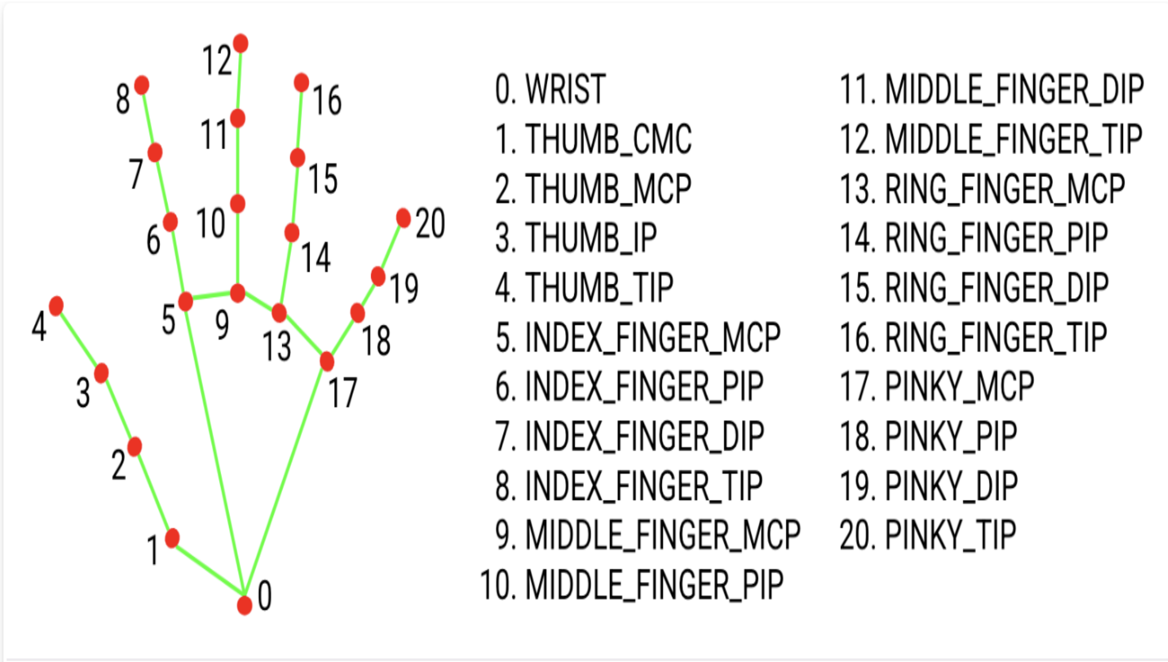

Fig. 3.3.1.1 Mediapipe Hand detection point names

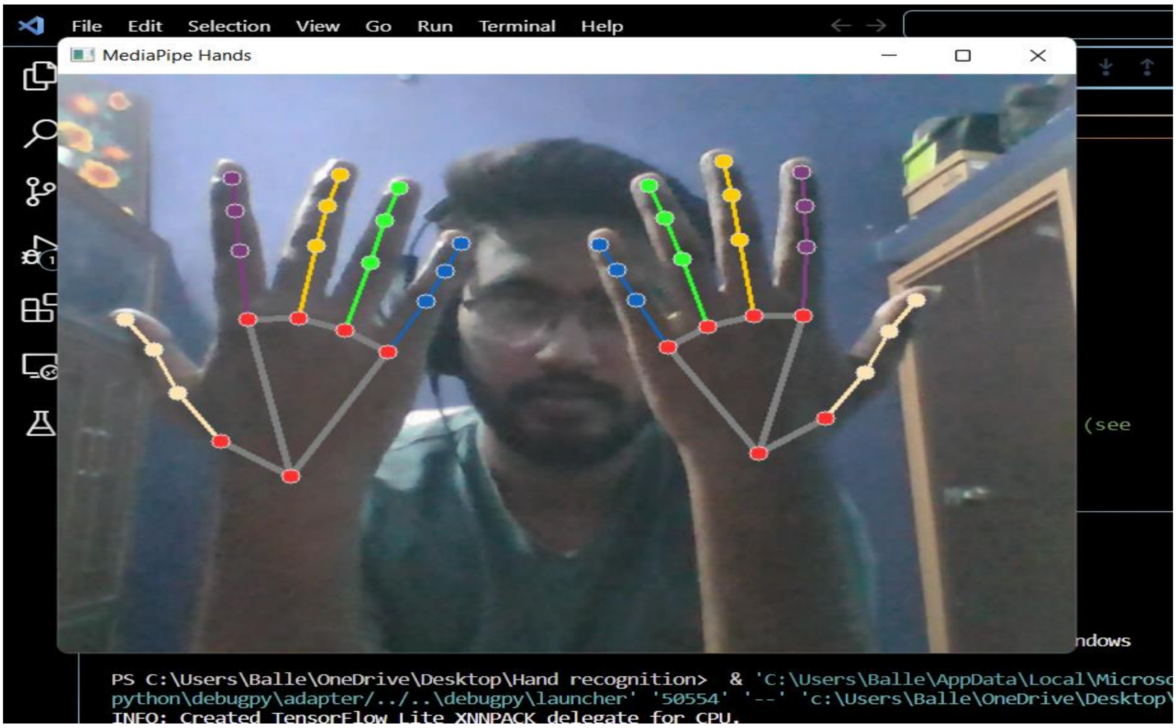

Fig. 3.3.1.2 Mediapipe Hand Detection Open

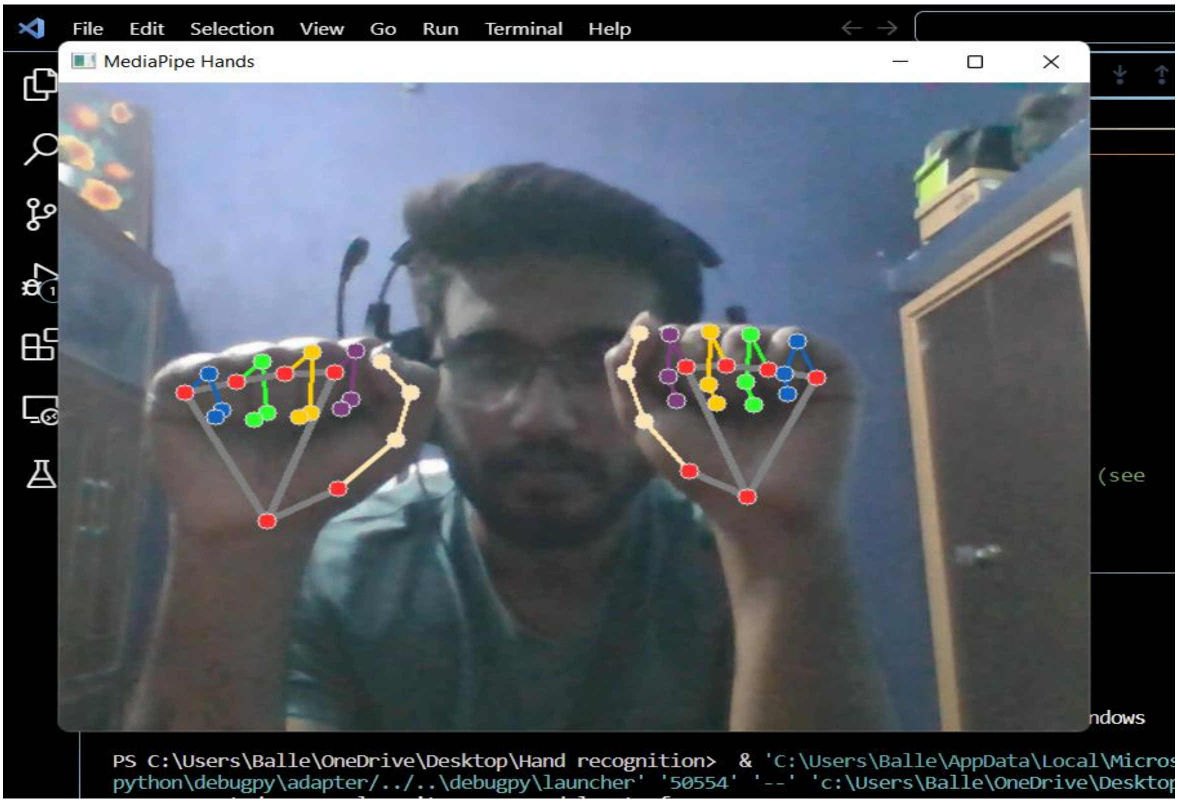

Fig. 3.3.1.3 Mediapipe Hand Detection Close (2 hands)

3.3.2 OpenCV

OpenCV, is a very large open-source library which is used for processing the images, used in machine learning, and for computer vision. It is currently the most important part of real-time image and video processing. It plays a very important part in modern systems. With help of this, anyone can analyse images and any video stream to find and differentiate different people, different objects, and even any human handwriting. Programming Language like- Python can handle the data captured and stored in the form of array data structure for the analyzation when different other libraries, like- NumPy are used with it. The vector space is deployed and some mathematical operations are applied to the characteristics which helps in identification of the different visual patterns and all of their distinct features.

OpenCV provides a wide range of functions for image processing, feature detection, and object tracking. Combined with Mediapipe's hand tracking module, OpenCV can be used to accurately recognize and interpret hand gestures in real-time.

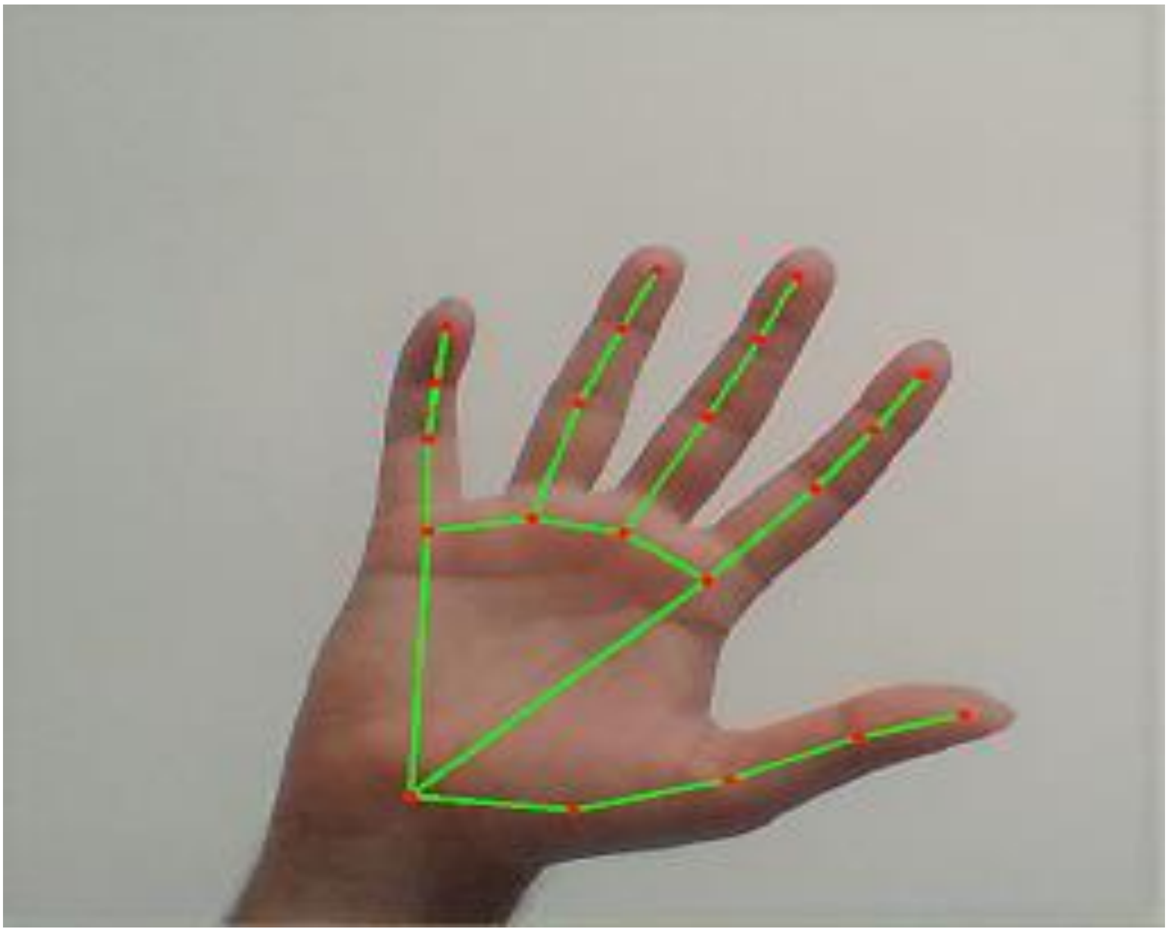

Fig. 3.3.2.1 OpenCV hand detection

3.3.3 PyAutoGUI

Python scripts can be automated including interactions with other apps by controlling the mouse and keyboard using PyAutoGUI. The API is intended to be user-friendly and easy to use. Python 2 and 3 are required for PyAutoGUI to function on Windows, macOS, and Linux.

Some of PyAutoGUI's features include:

1. Clicking and dragging the mouse across other application windows.
2. Keystrokes being sent to applicants (for example, to fill out forms).
3. Take screenshots of the screen whenever you come across an image, such as a button or checkbox.
4. You can close, minimize all window tabs, or current tab, resize, move, or maximize an application's window by finding it. (Windows-only, currently).
5. Display message and alarm boxes.

CHAPTER 4:
Result and Output

4.1 Proposed Model Outputs

The proposed software can recognize many gestures including both Static as well as Dynamic Gestures. Till now, it can recognize total of 14 gestures i.e., 8 static gestures and 6 dynamic gestures.

Static Gestures -

1. Open Hand
2. Close Hand
3. OK Sign
4. Peace Sign or V Sign
5. Thumbs Up
6. Thumbs Down
7. Pinch In
8. Pinch Out

Dynamic Gestures- (1 Finger (Used for dynamic gesture recognition activation)

1. Clockwise
2. Counter Clockwise
3. Move Left
4. Move Right
5. Move Up
6. Move Down

4.2 Gesture and Their Working

Table 4.2.1 Gesture Name and Their Corresponding Functions

Open Hand	Play/Pause
Close Hand	Close the current window
OK Sign	Save
Peace Sign or V Sign	Mute / Unmute
Thumbs Up	Top of page
Thumbs Down	Bottom of page
Pinch In	Zoom Out
Pinch Out	Zoom In
Clockwise	Go to main screen || Minimize all windows/maximize all windows
Counter Clockwise	Lock Windows
Move Left	Previous Slide / Seek video backward
Move Right	Next slide / Seek video forward
Move Up	Volume Up
Move Down	Volume Down

4.3 Outputs – Hand Gesture Recognition

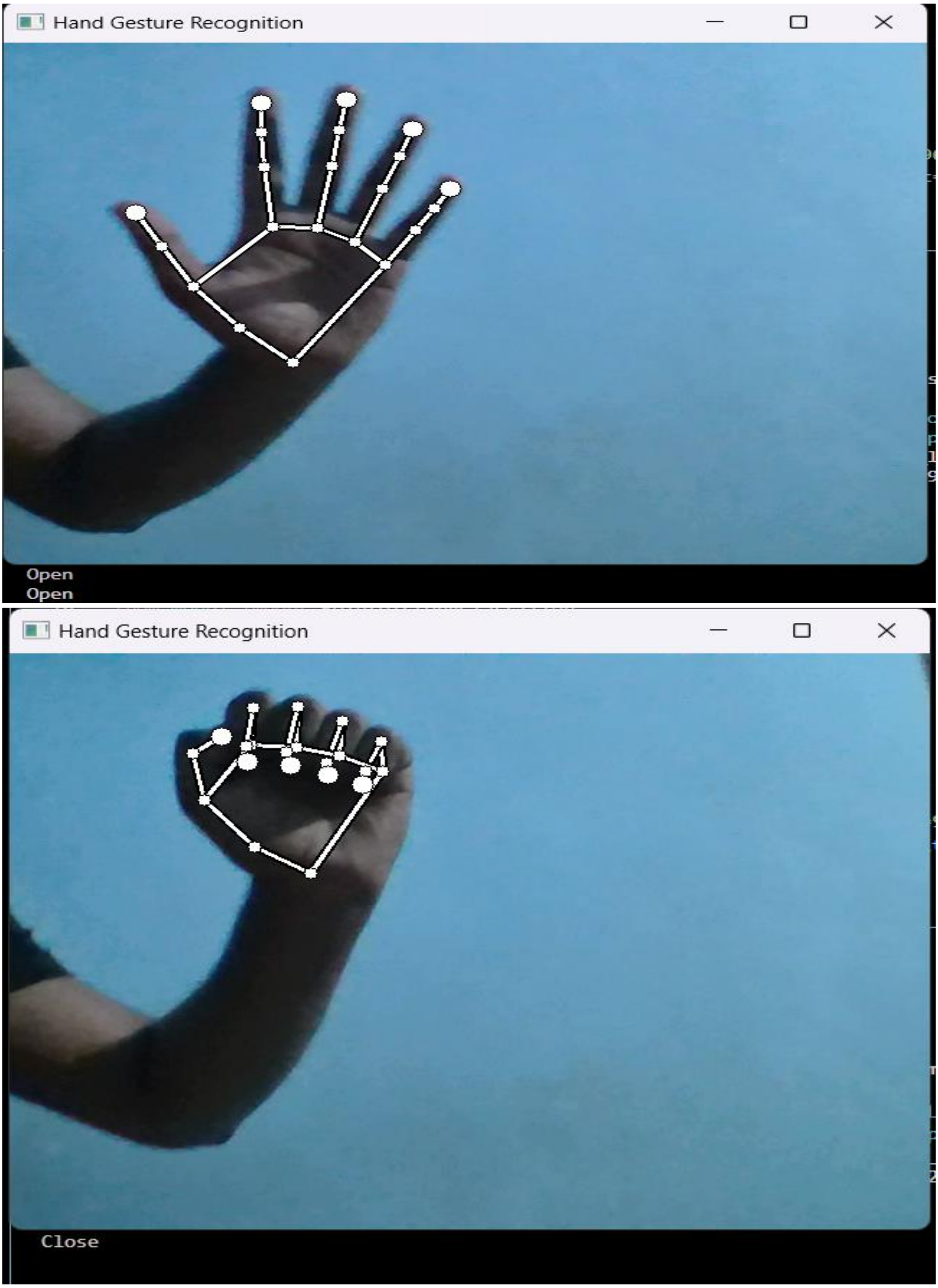

Fig. 4.3.1 Static Gesture – Open Hand

Fig. 4.3.2 Static Gesture – Close Hand

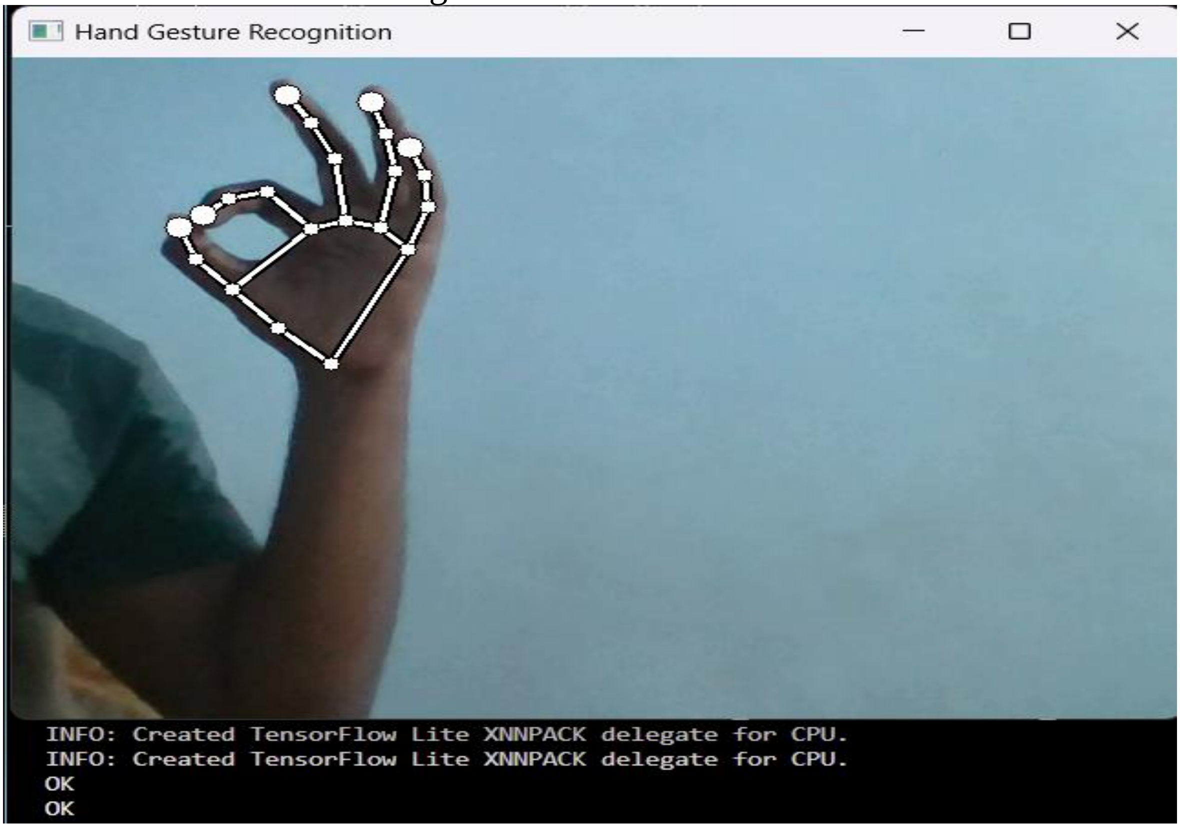

Fig. 4.3.3 Static Gesture – OK Sign

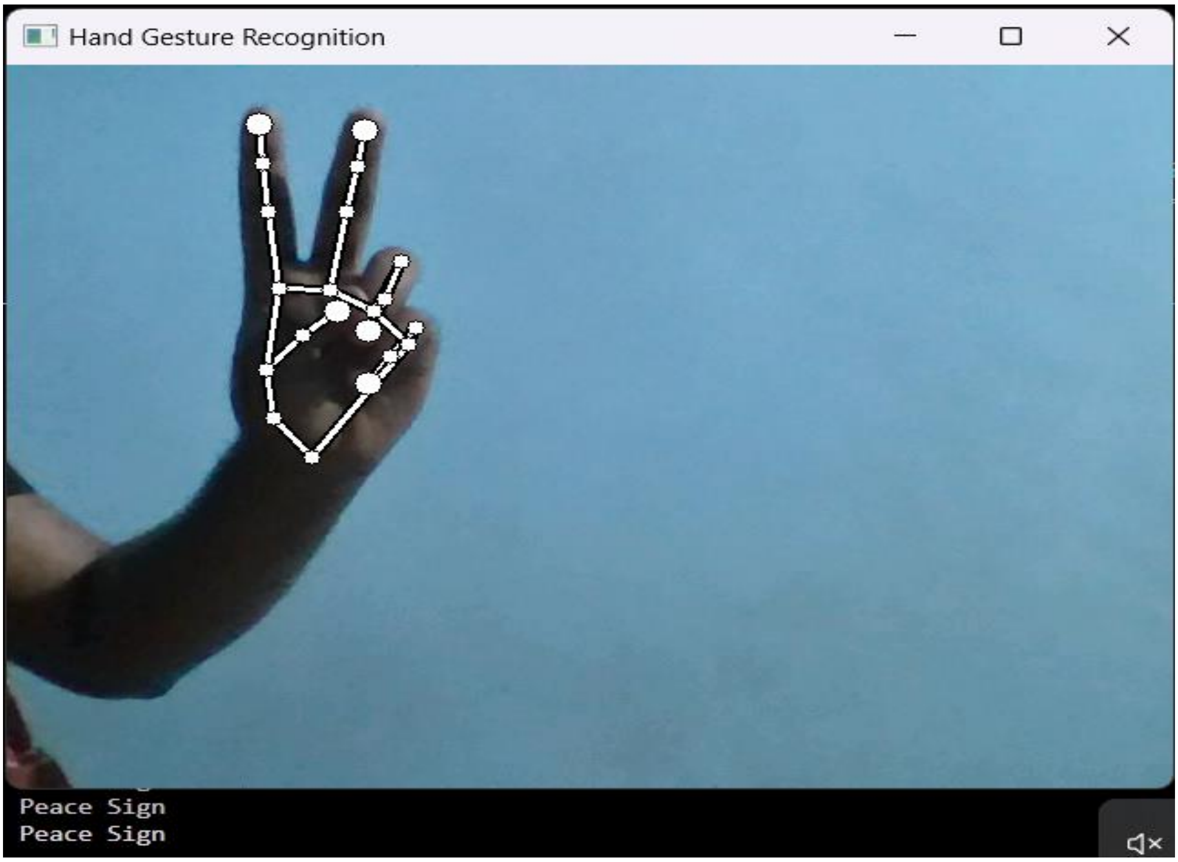

Fig. 4.3.4 Static Gesture – Peace Sign/V-Sign

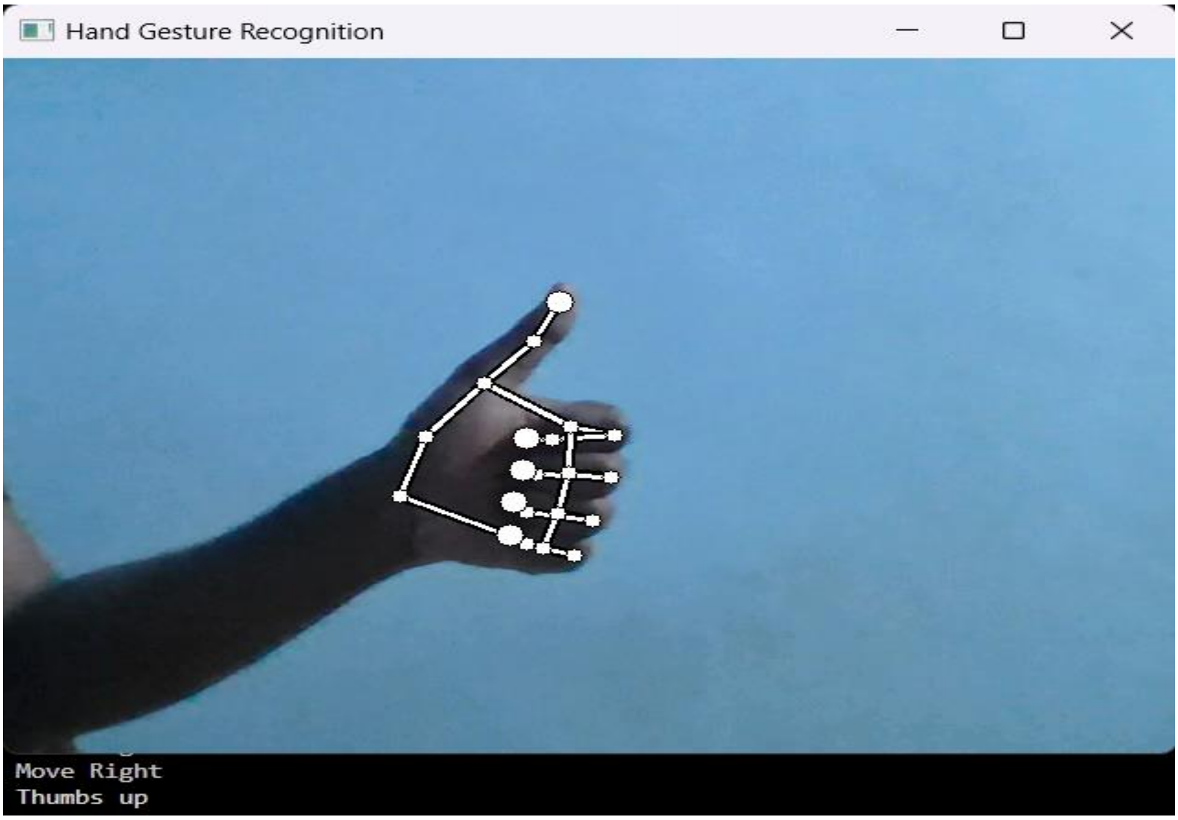

Fig. 4.3.5 Static Gesture – Thumbs Up

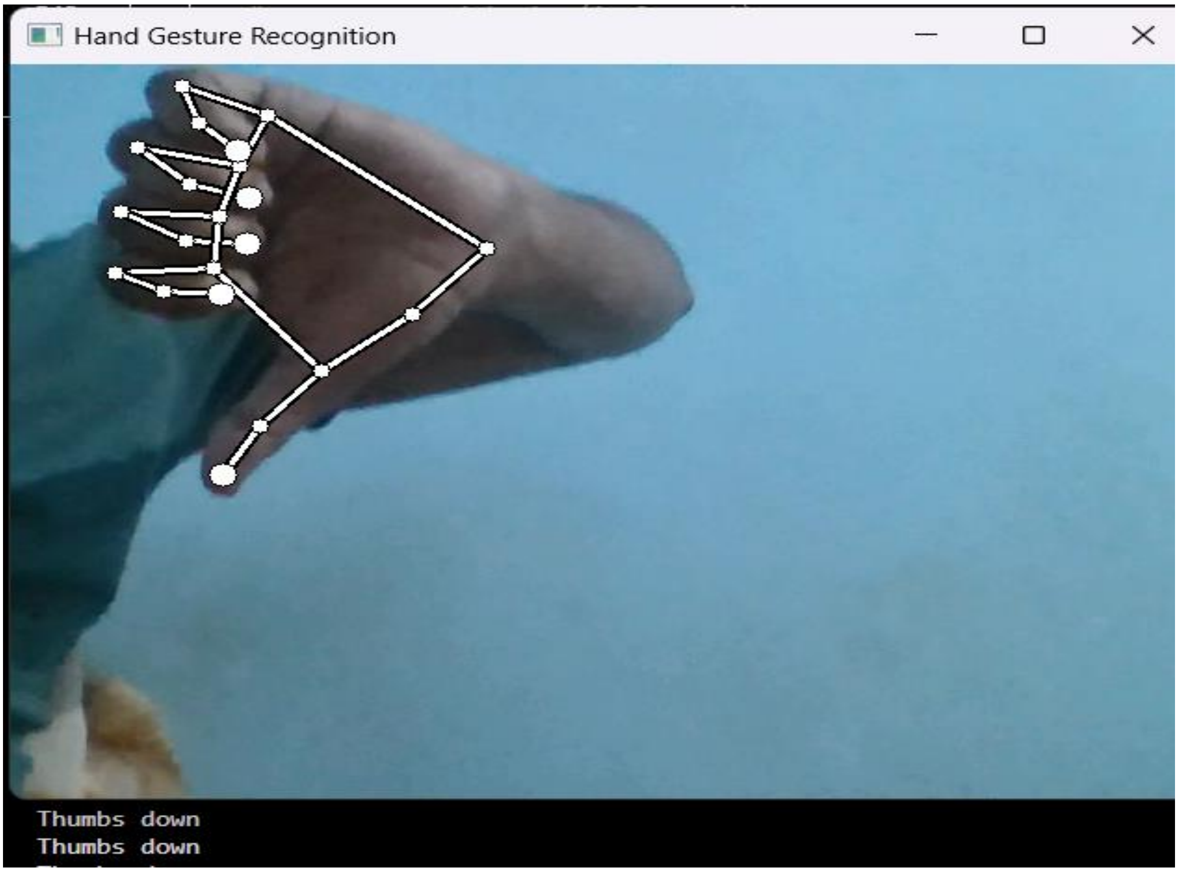

Fig. 4.3.6 Static Gesture – Thumbs Down

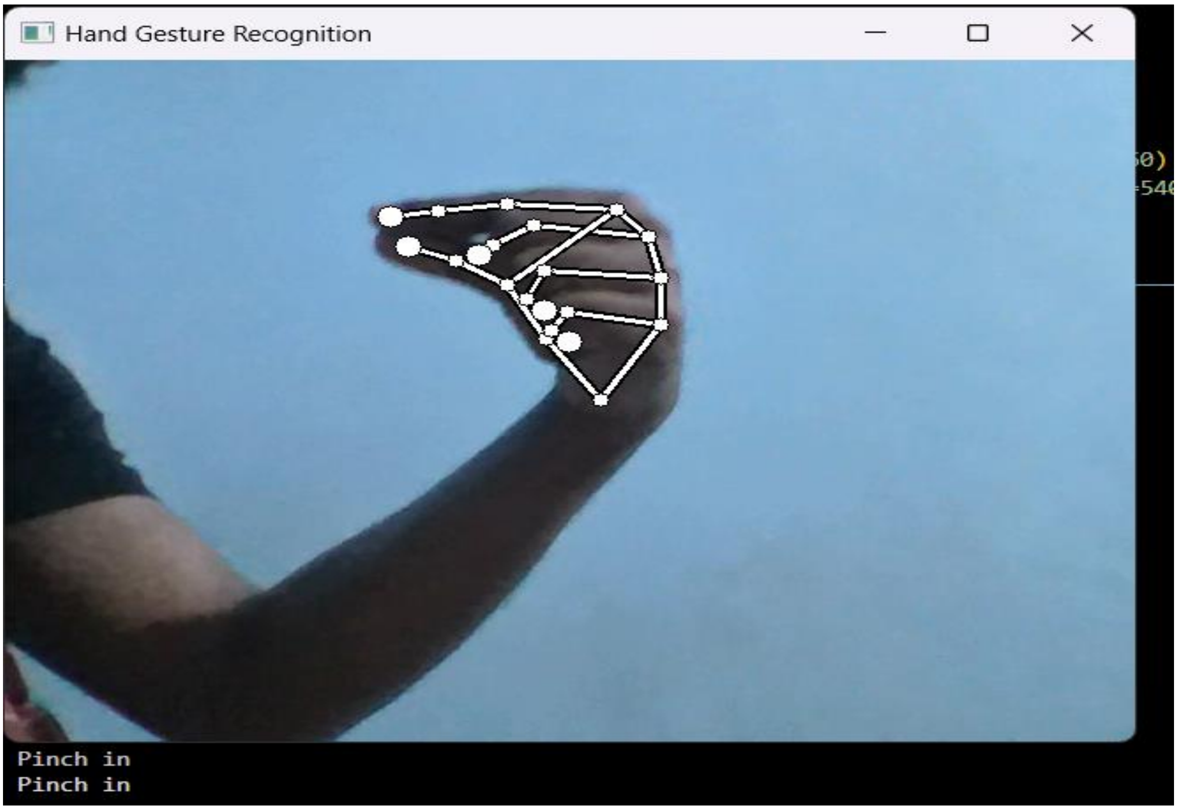

Fig. 4.3.7 Static Gesture – Pinch In

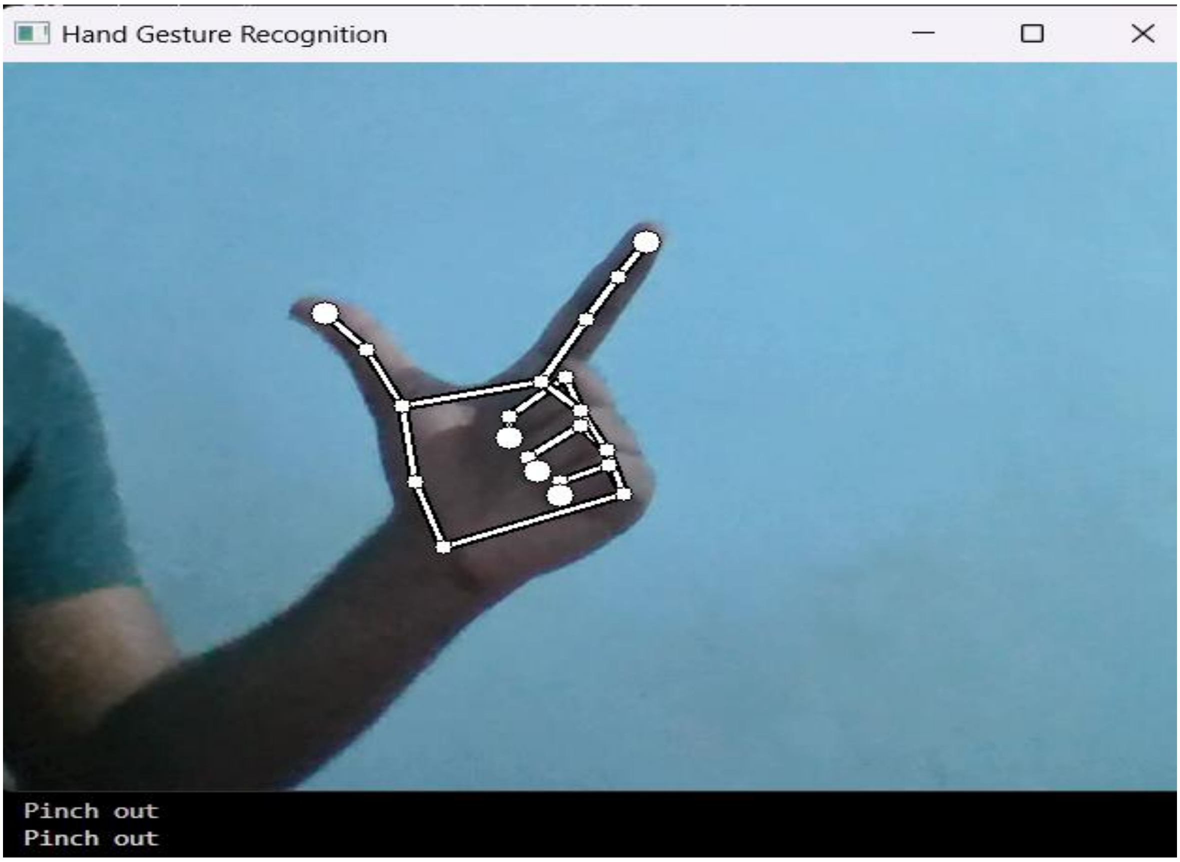

Fig. 4.3.8 Static Gesture – Pinch Out

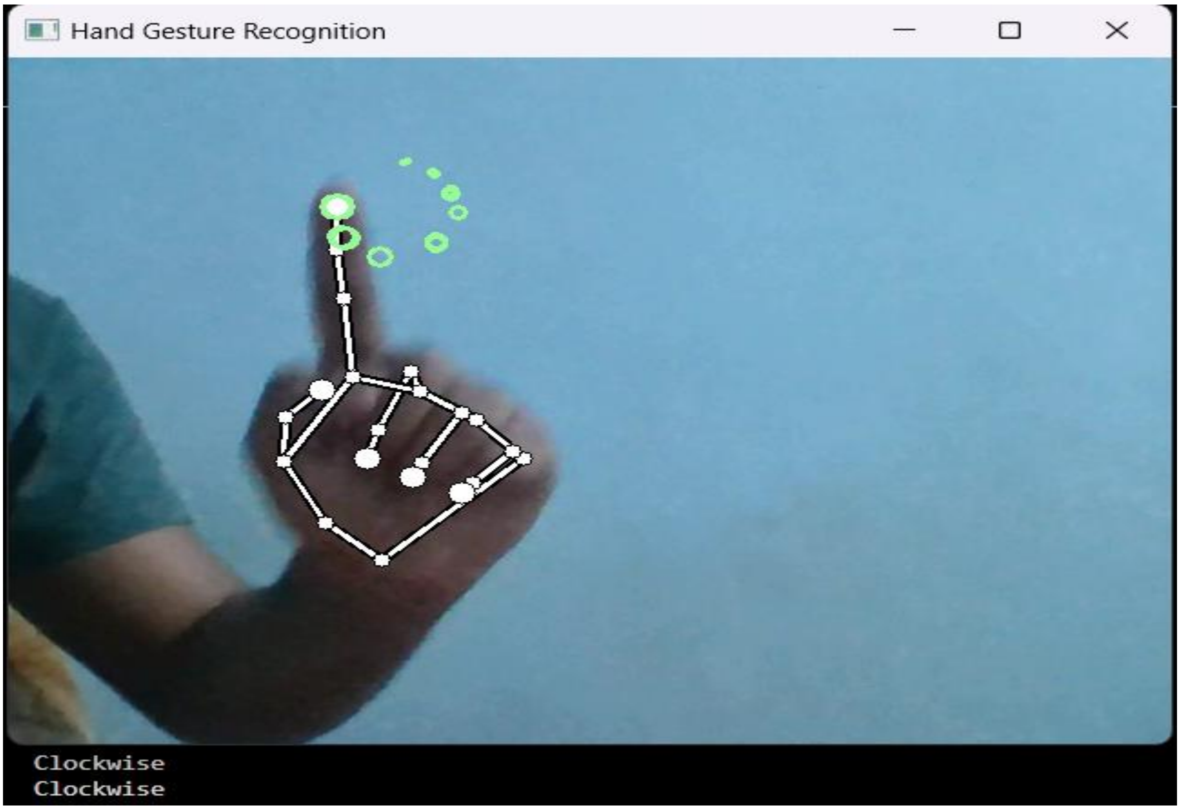

Fig. 4.3.9 Dynamic Gesture – Clockwise

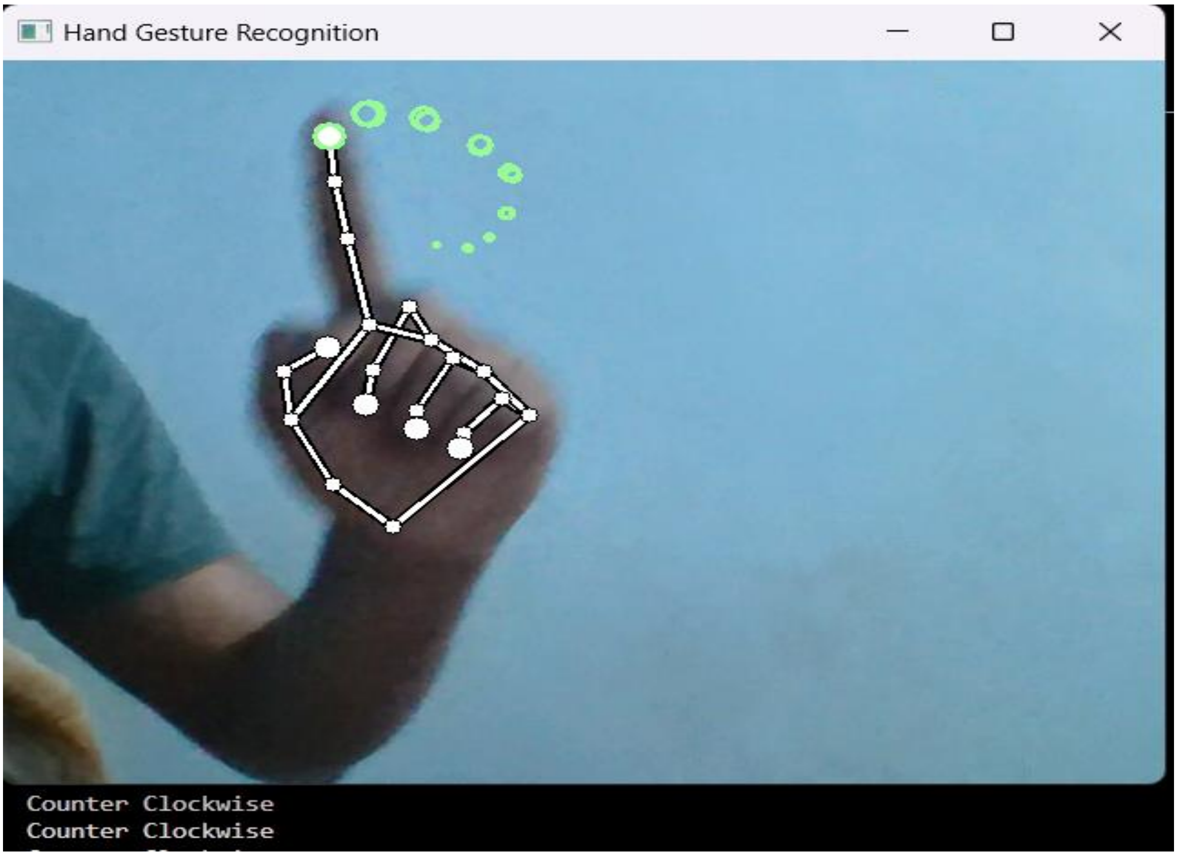

Fig. 4.3.10 Dynamic Gesture – Counter clockwise

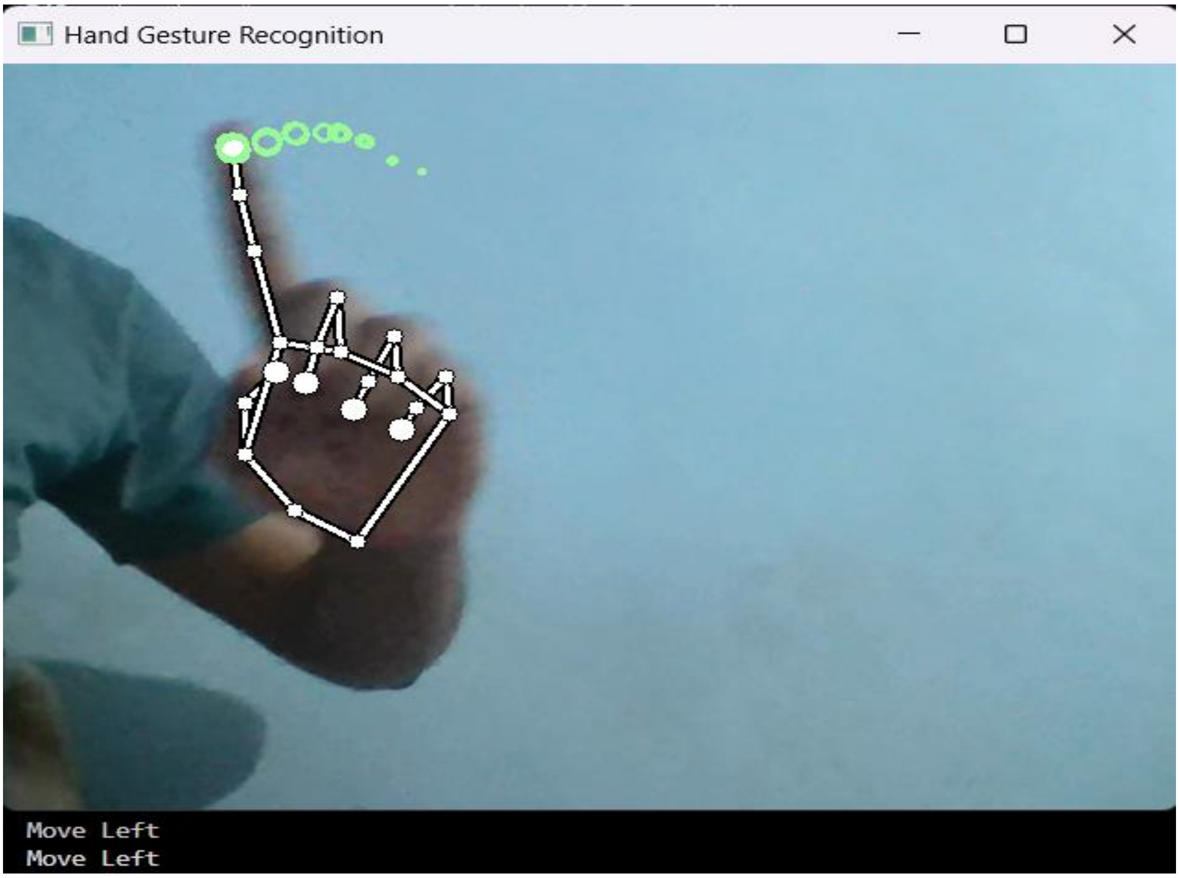

Fig. 4.3.11 Dynamic Gesture – Move Left

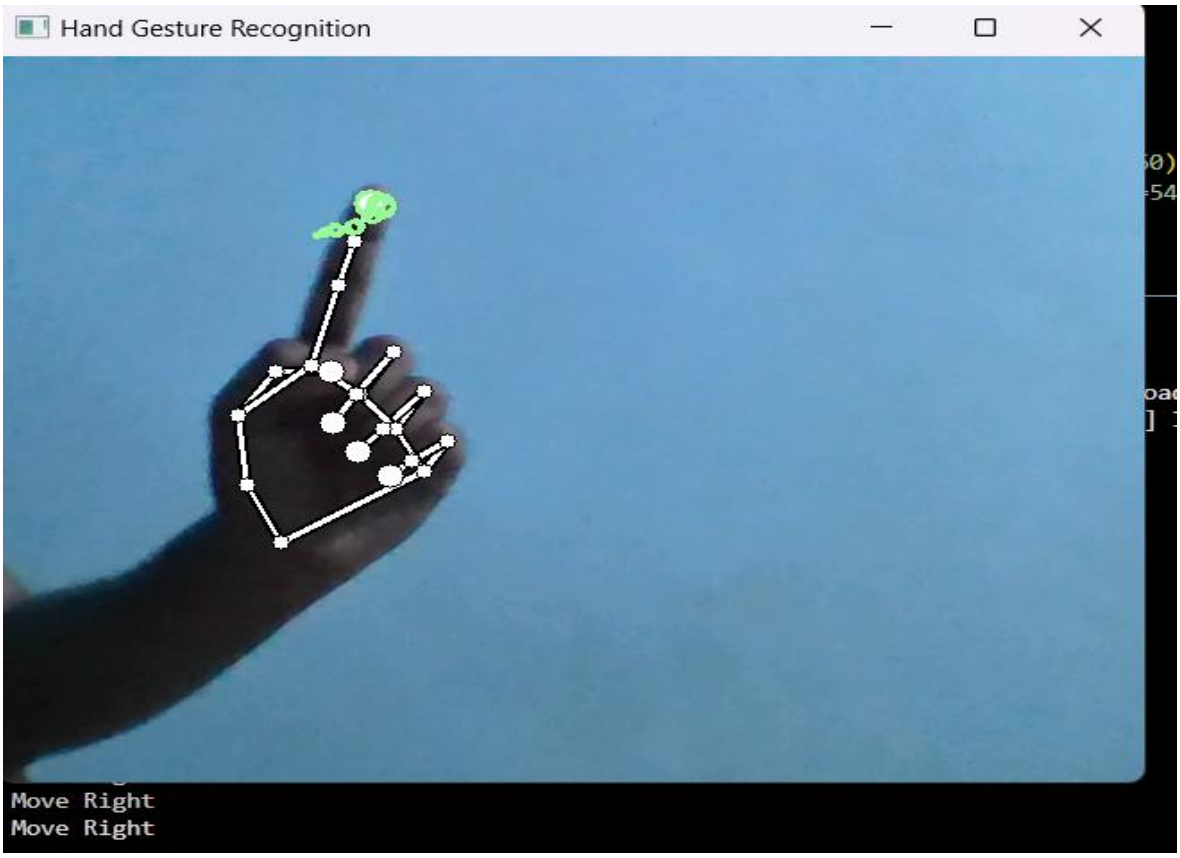

Fig. 4.3.12 Dynamic Gesture – Move Right

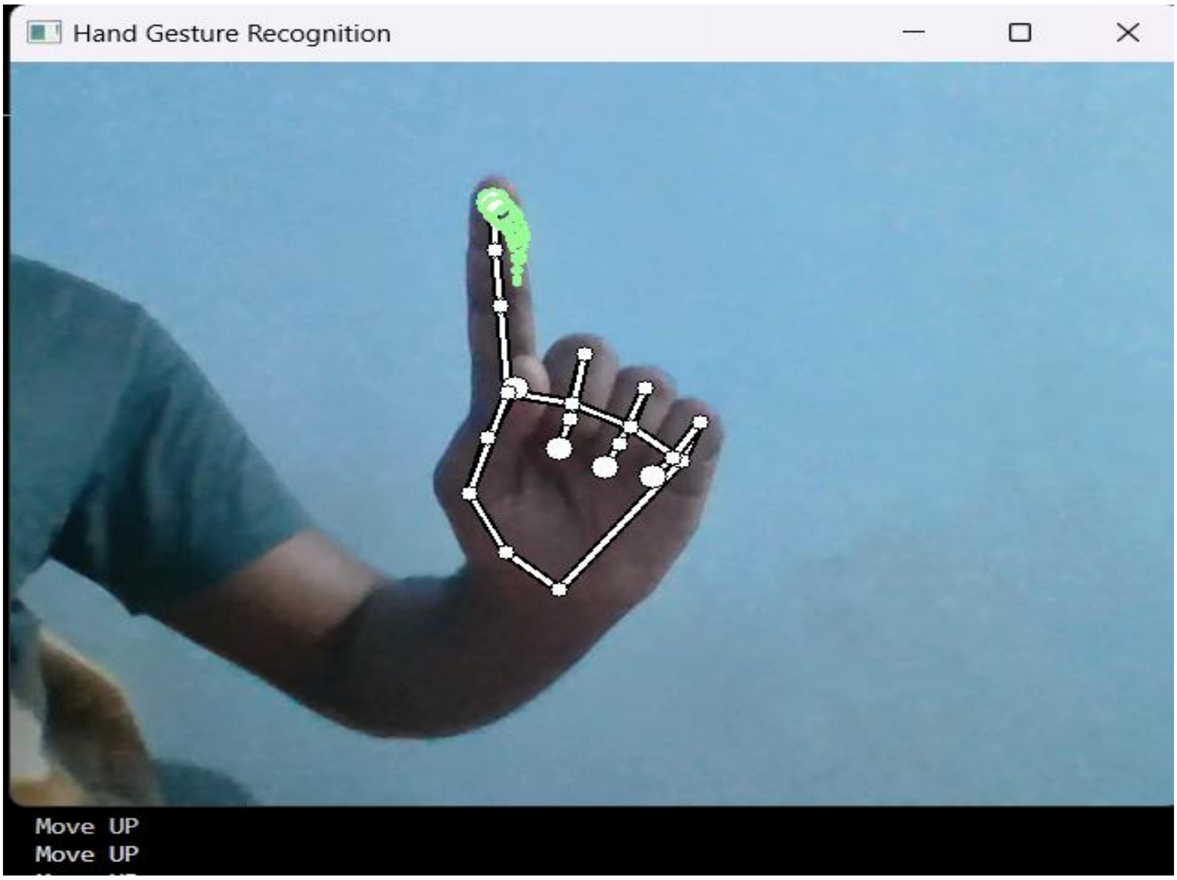

Fig. 4.3.13 Dynamic Gesture – Move Up

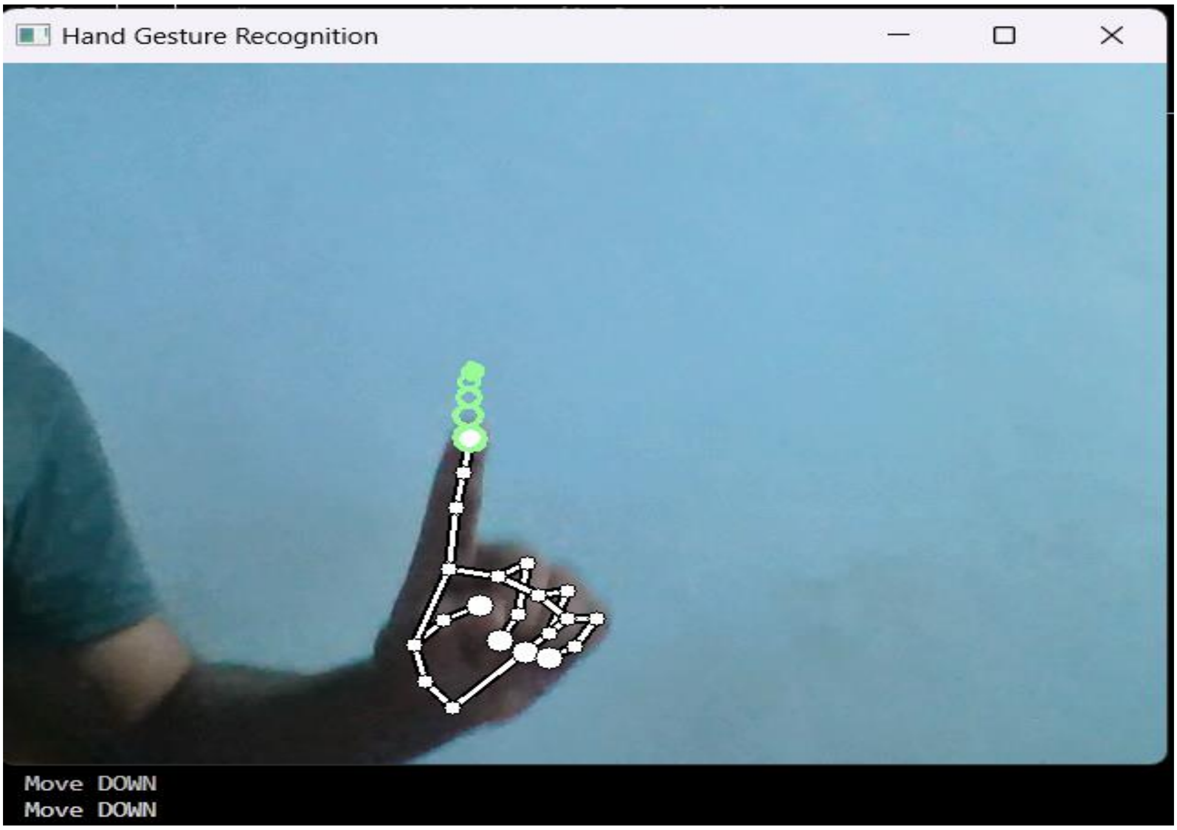

Fig. 4.3.14 Dynamic Gesture – Move Down

4.4 Outputs – Functions Performing on Gesture Recognition

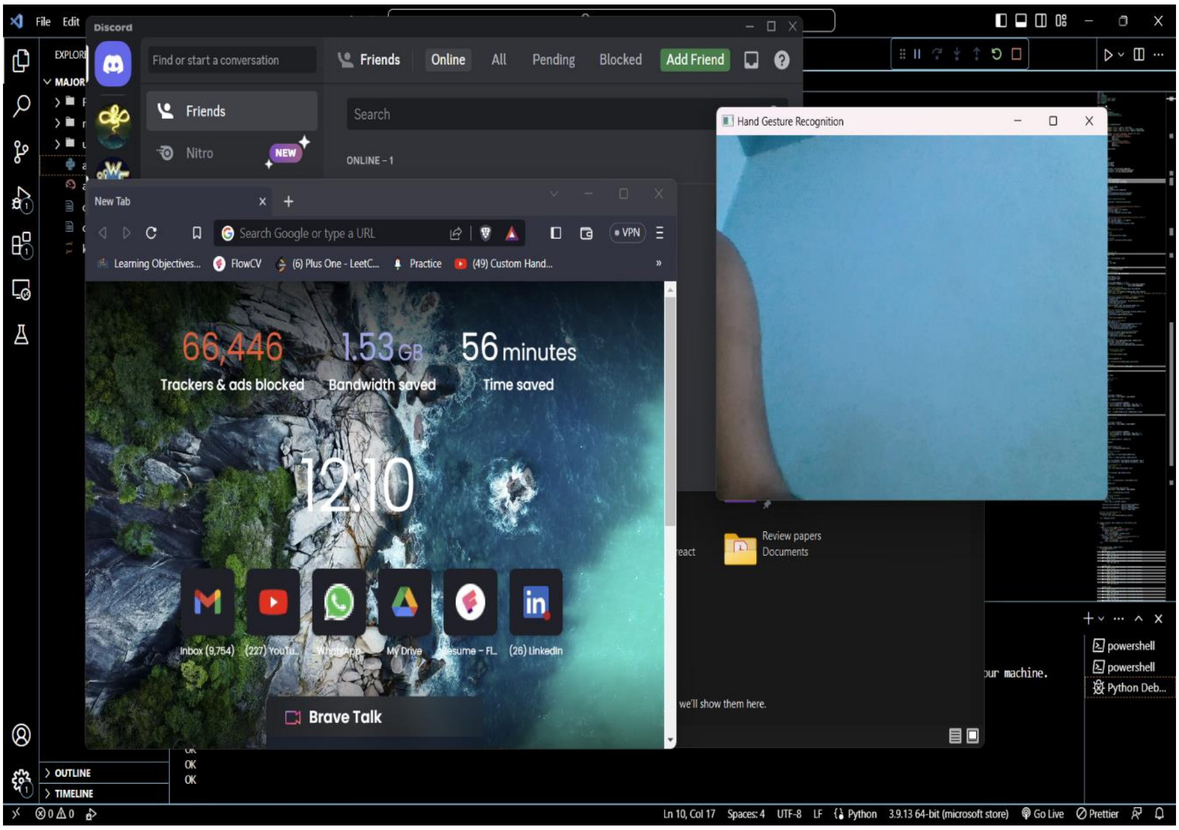

Fig. 4.4.1 Multiple opened windows

Fig. 4.4.2 All windows minimized (after clockwise gesture)

Fig. 4.4.3 Windows locked(after counter-clockwise gesture)

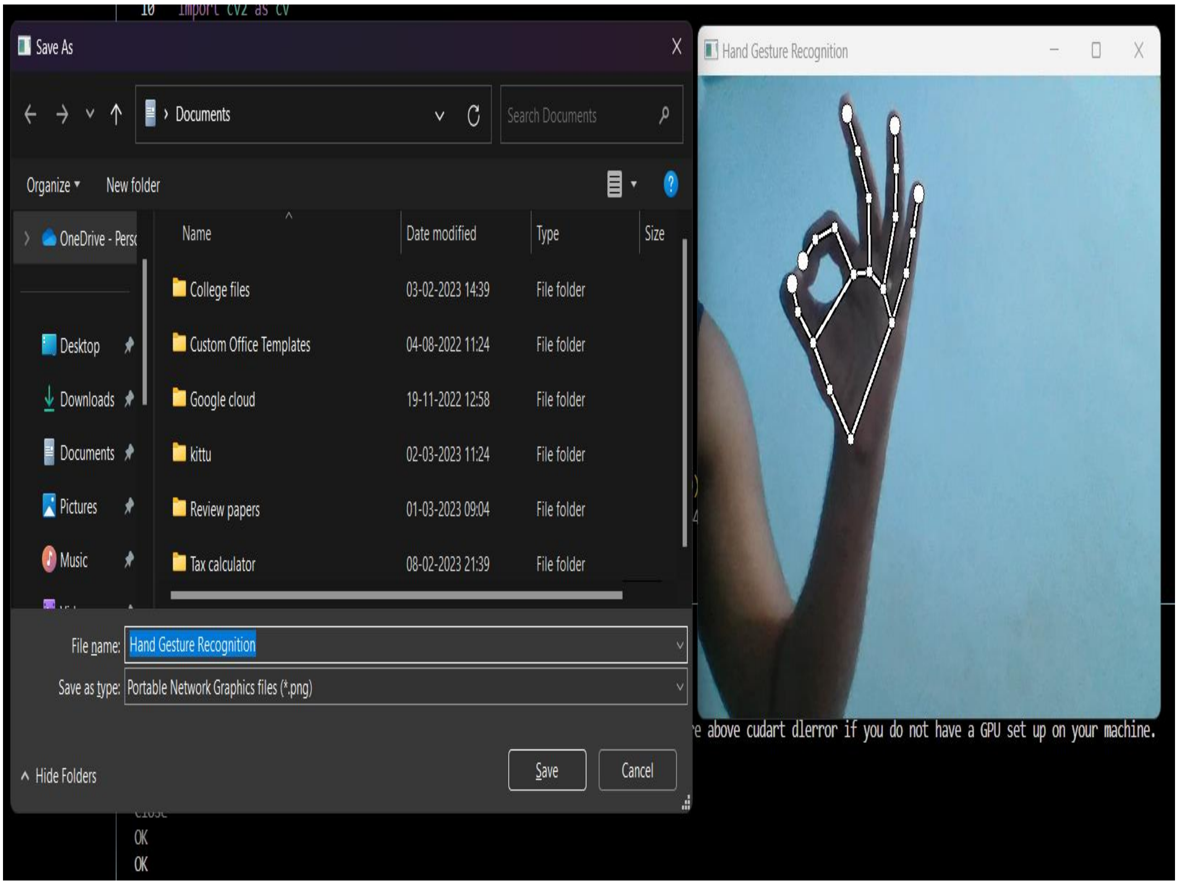

Fig. 4.4.4 Save pop-up after OK gesture

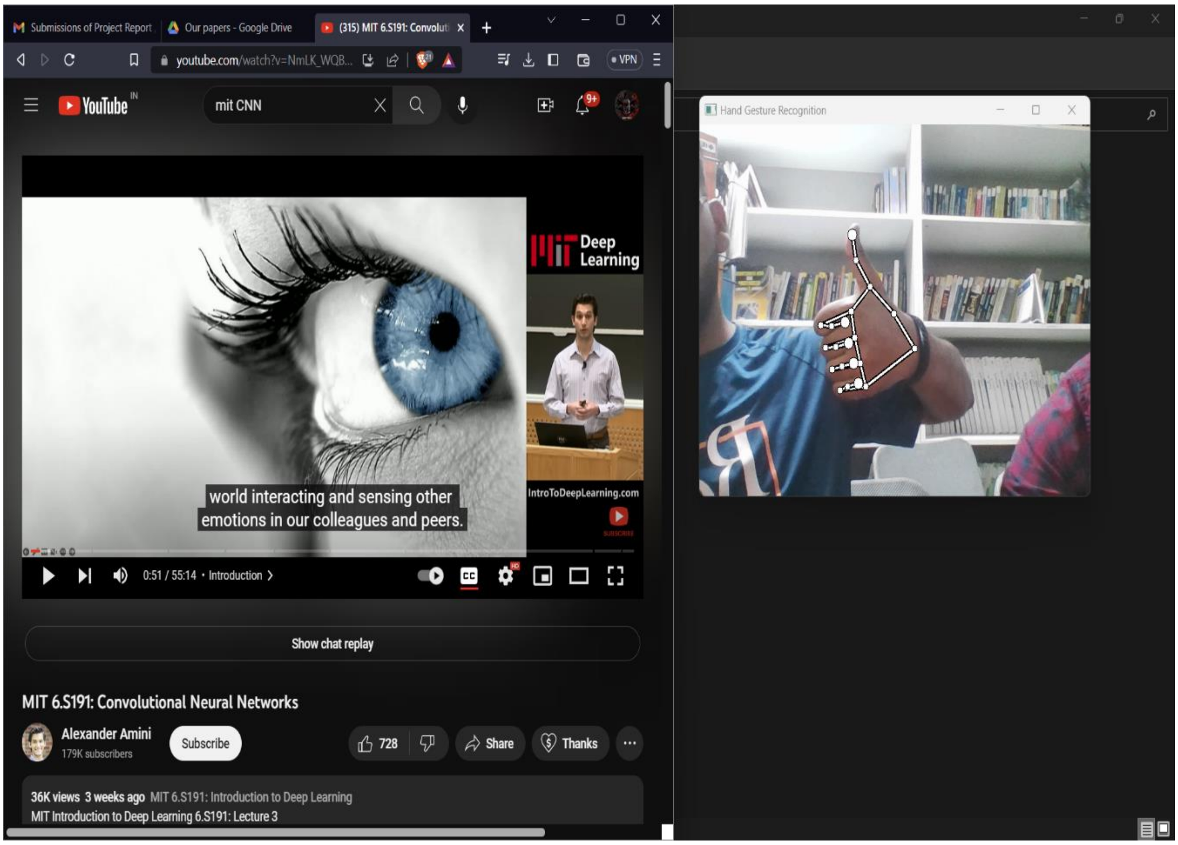

Fig. 4.4.5 Moving to Page top with Thumbs Up Gesture

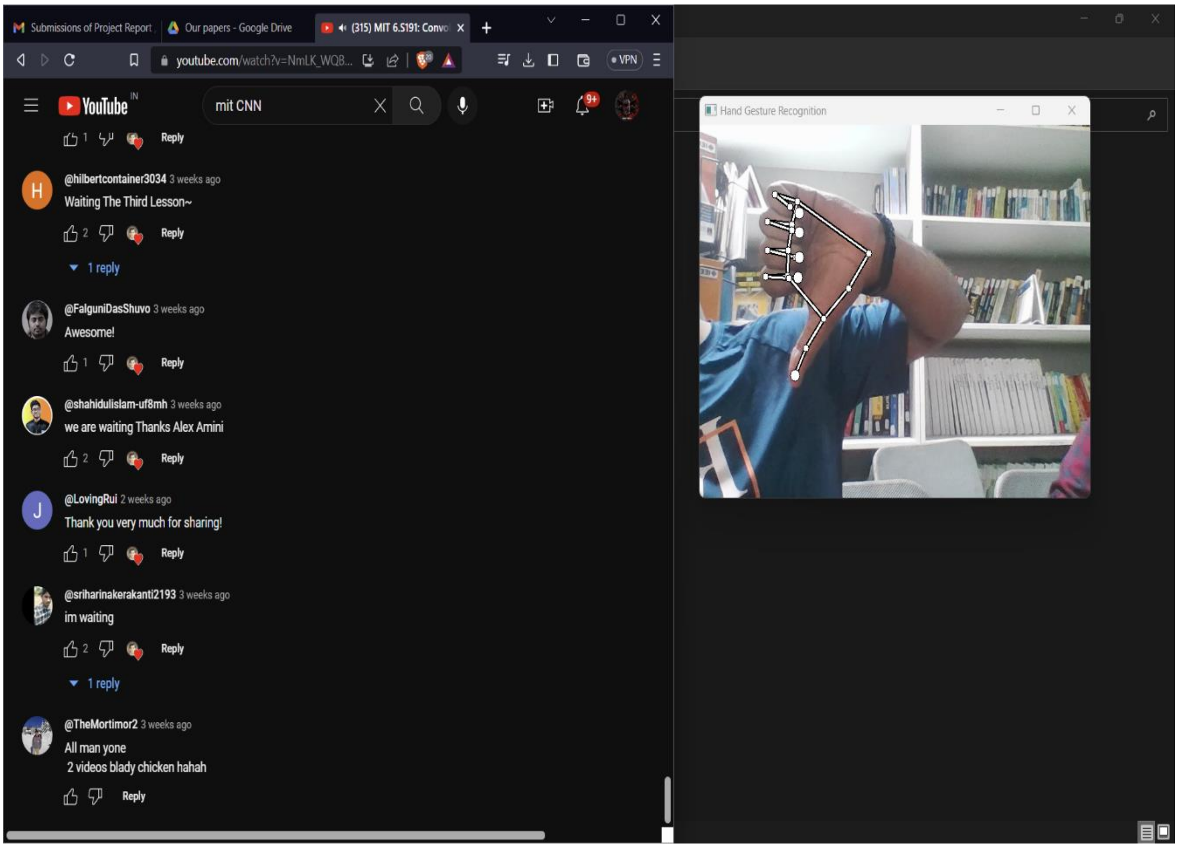

Fig. 4.4.6 Moving to Page bottom with Thumbs Down Gesture

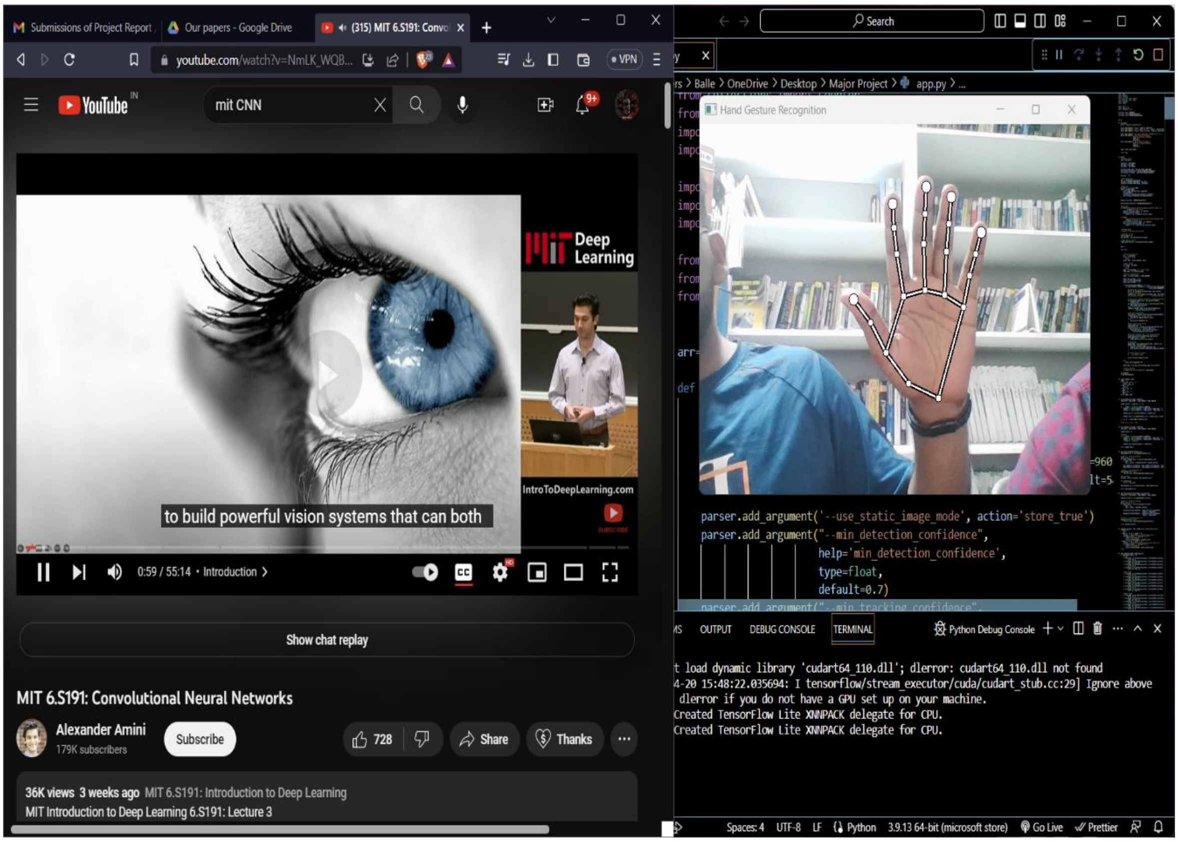

Fig. 4.4.7 Video Play with Open Hand Gesture

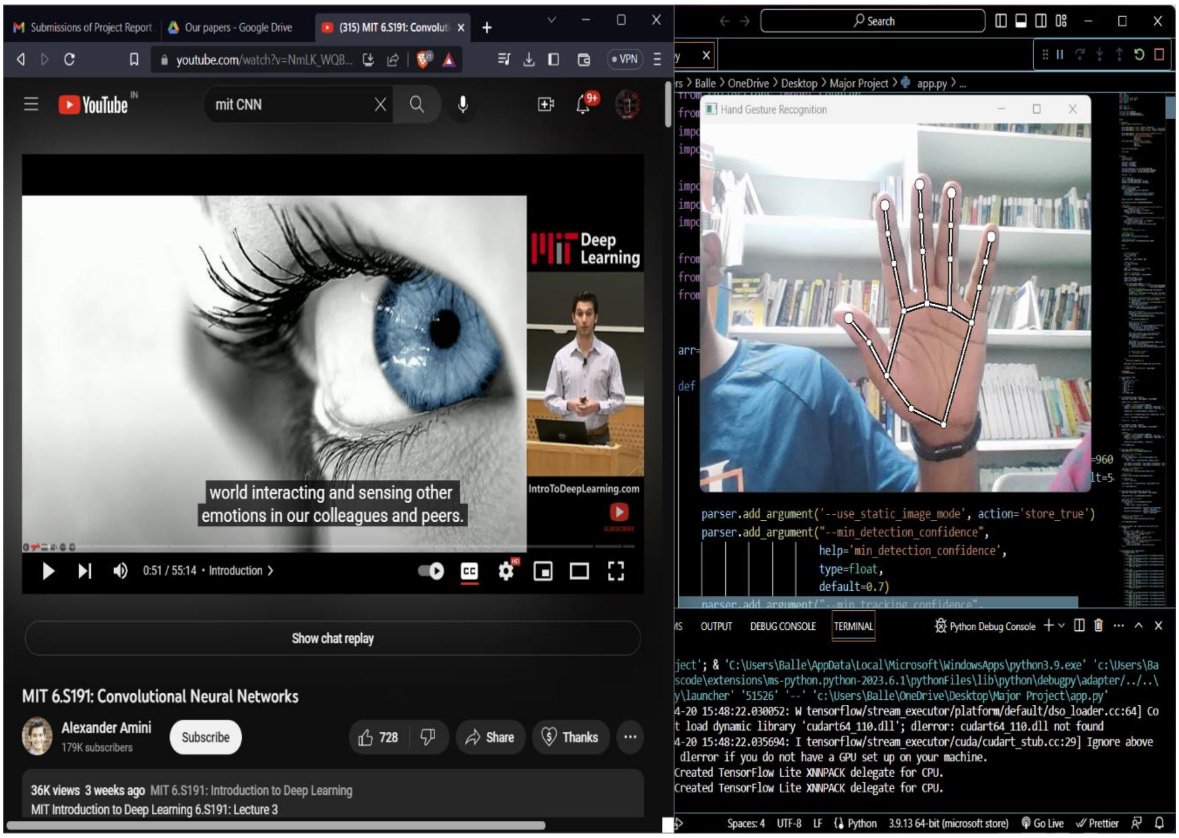

Fig. 4.4.8 Video Pause with Open Hand Gesture

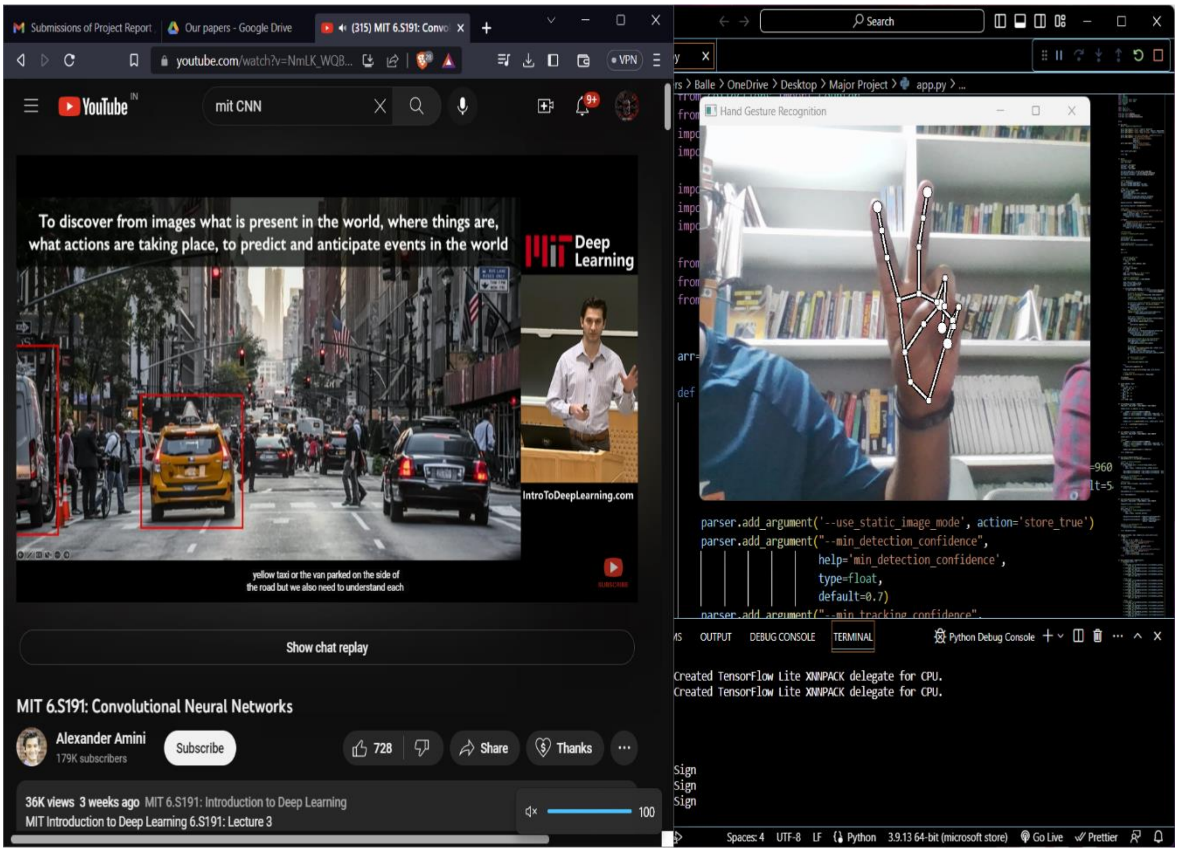

Fig. 4.4.9 Mute with V-sign or Peace Gesture

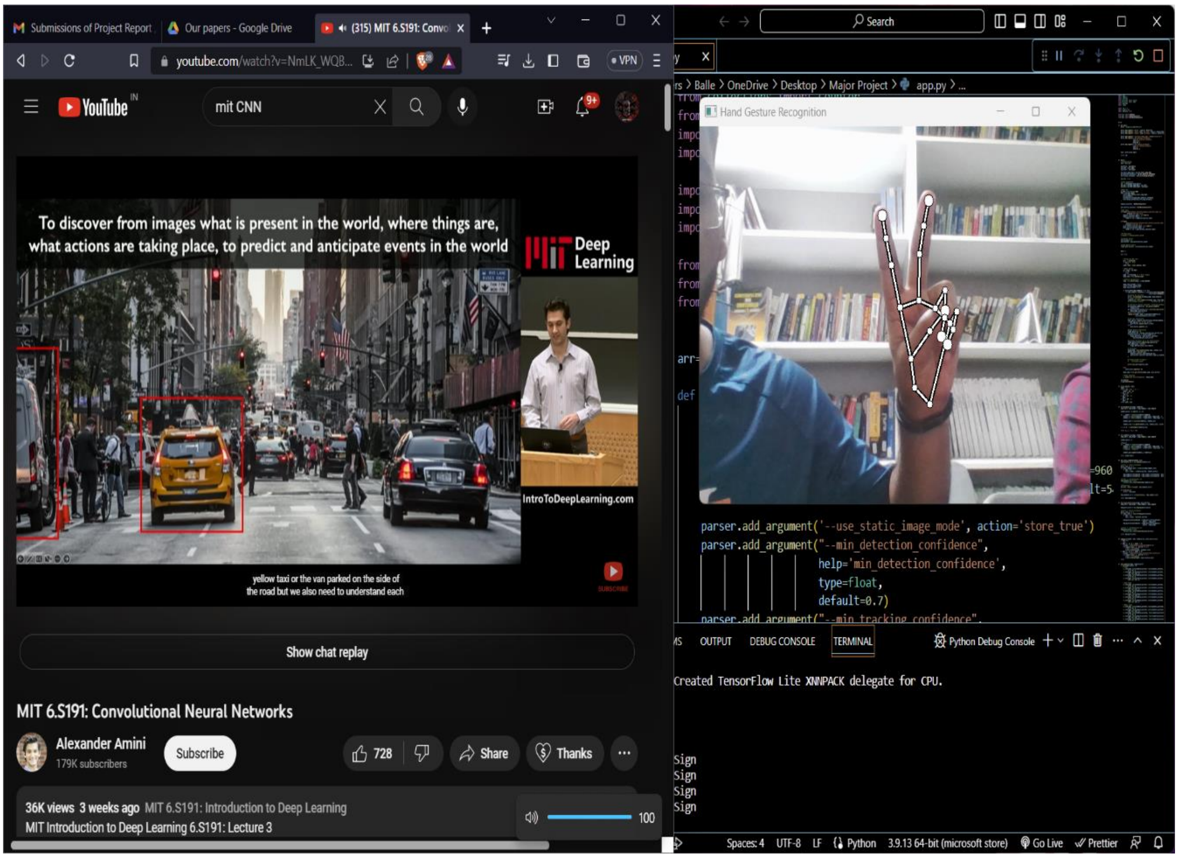

Fig. 4.4.10 Unmute with V-sign or Peace Gesture

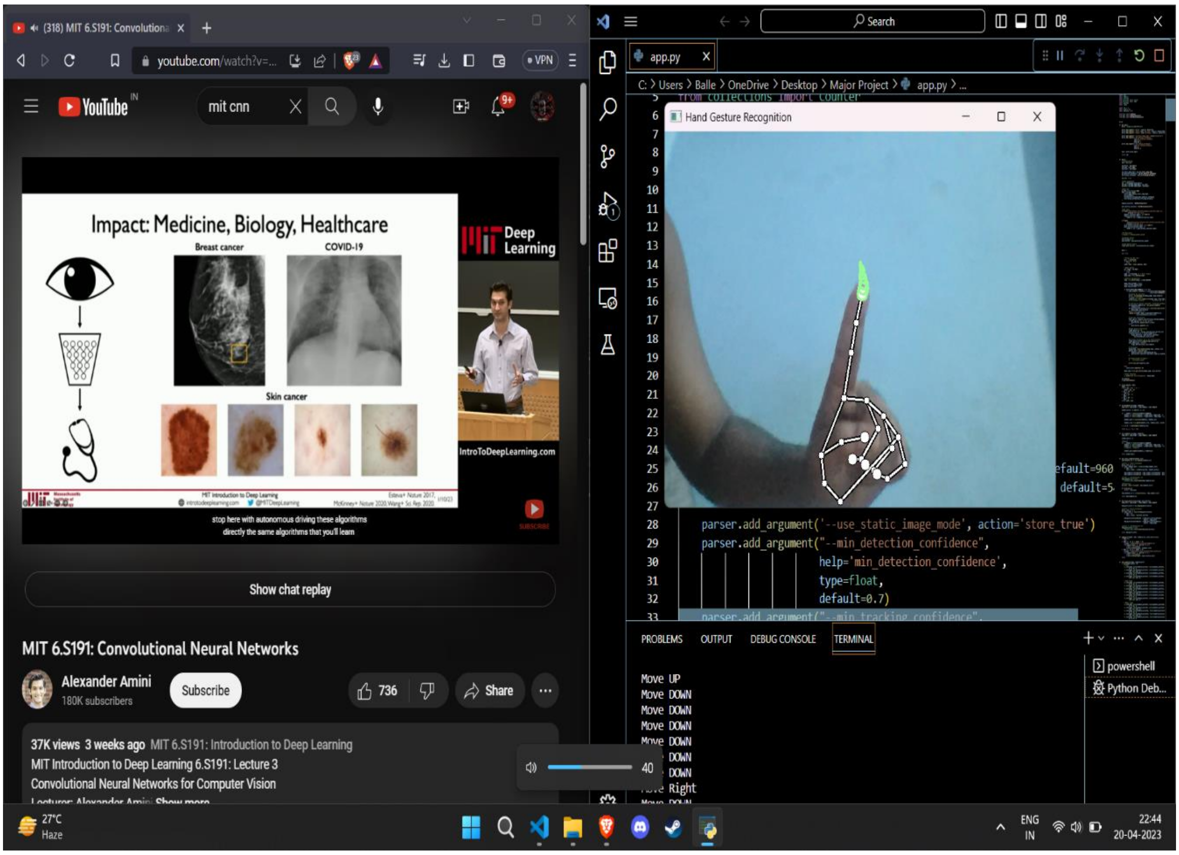

Fig. 4.4.11 Volume Decrease with Move Down Gesture

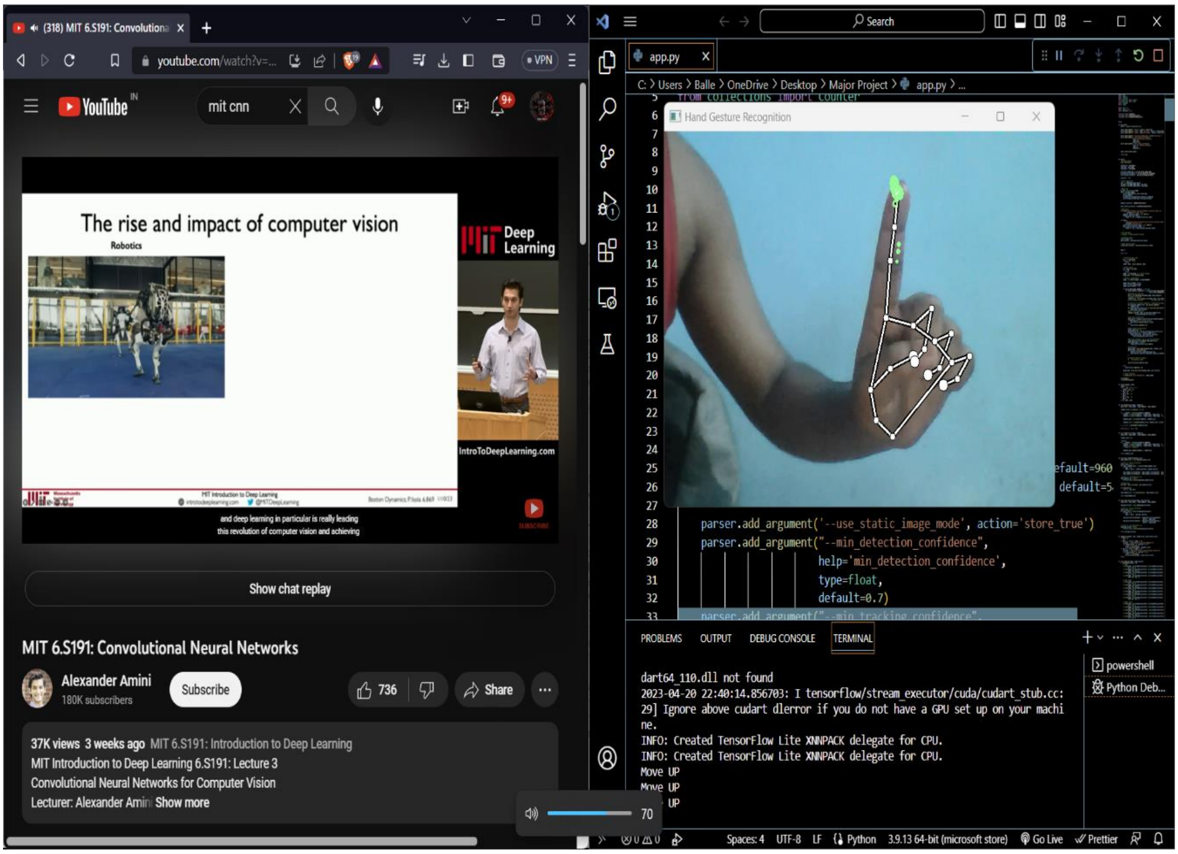

Fig. 4.4.12 Volume Increase with Move Up Gesture

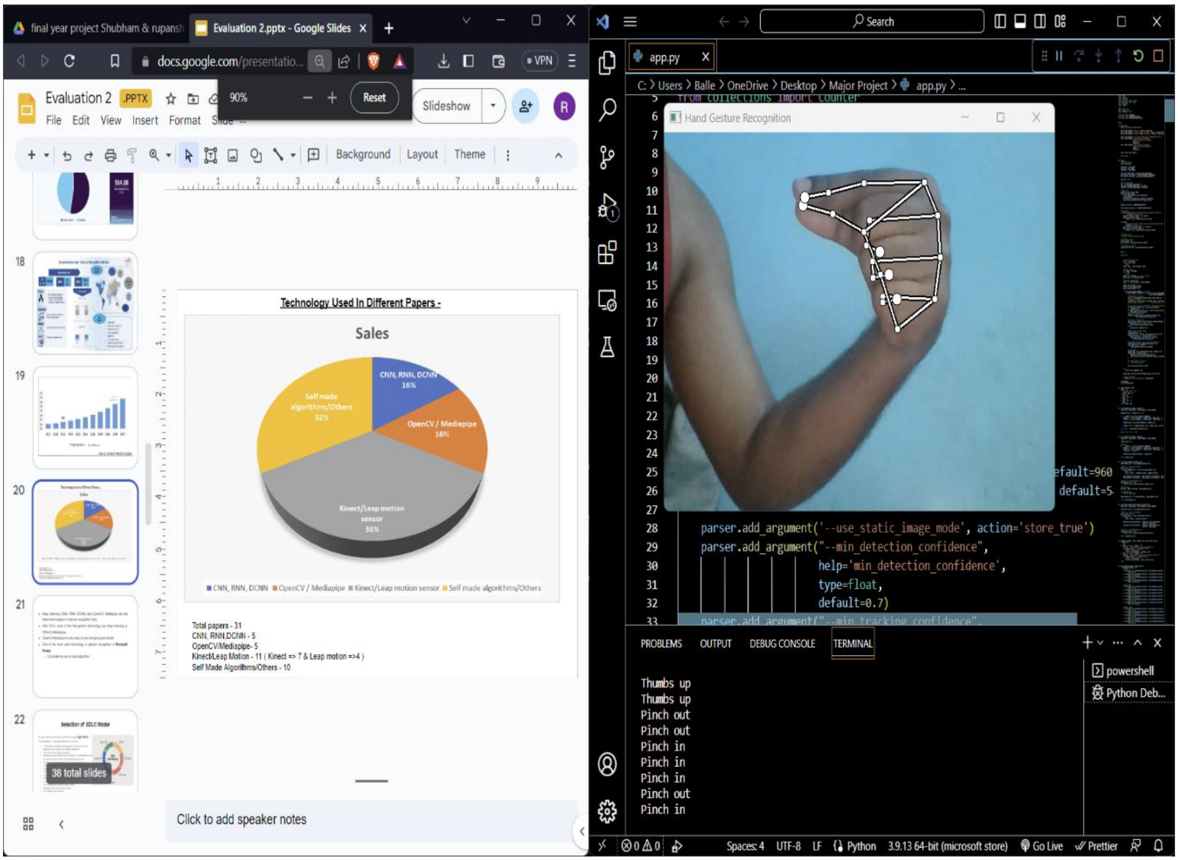

Fig. 4.4.13 Zoom Out with Pinch In Gesture

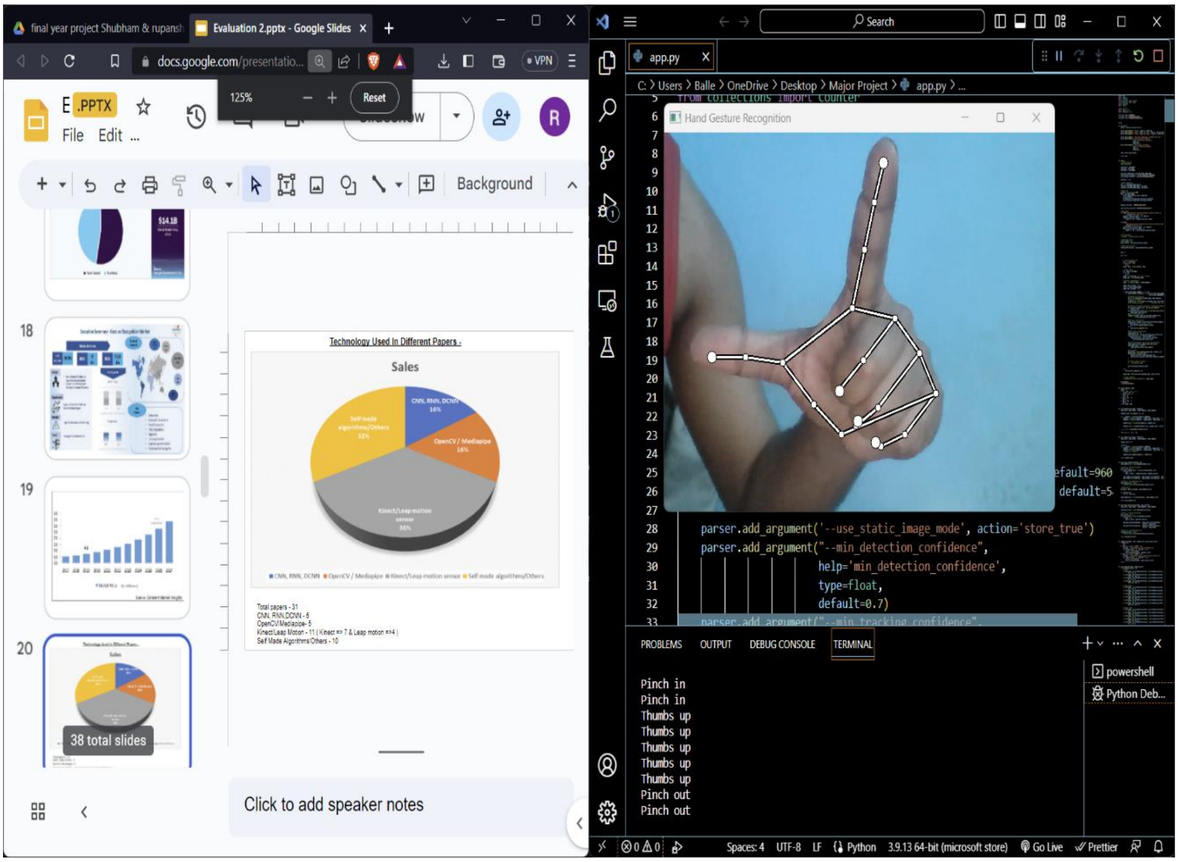

Fig. 4.4.14 Zoom In with Pinch Out Gesture

CHAPTER 5

Testing Process

5.1 Software Testing

Software testing is essential for making sure that software apps are reliable and of high quality. It entails running a software system or one or more of its components to find bugs and ensure that the system satisfies the criteria.
Before the programme is released to end consumers, its main objective is to find any flaws or errors. Software defects can be fixed before a product is published with the aid of testing, which also gives developers input on their work.
Software testing is a crucial step in the life cycle of software creation that serves to guarantee the dependability and quality of software products.

5.2 Unit Testing

In this instance, each module is assessed separately. The criteria used to define unit test modules were chosen to help find modules with essential functionality. A module might be a single person or a process.

The different unit testing functions that were tested are mentioned below:

- Working of Mediapipe
- Hand detection model
- Static Gesture Recognition model
- Dynamic Gesture Recognition model
- Using PyAutoGUI
- Using computer functionalities with Hand

5.3 Integration Testing

During the planning phase of integration, pertinent components are combined and collectively analysed. The integrated testing framework is created by applying integration test plan tests to bigger aggregates of unit-tested components, such as data, that have already undergone unit testing.

5.4 Validation Testing

This method is used to check whether the software meets the requirements at the beginning or conclusion of the manufacturing process.

5.5 Test Cases

Table 5.5.1 Test case for working of Mediapipe

Test case ID	1
Test Case Name	Working of Mediapipe
Test Case description	Detection of video
Steps	1. Starting the camera 2. Retrieve the video from the camera
Expected Results	Camera detected and opened.
Actual Results	As expected

Table 5.5.2 Test case for working of Hand detection model

Test case ID	2
Test Case Name	Hand detection model
Test Case description	Detection of the user’s hand
Steps	1. The palm model detects the palm region of the hand. 2. Model detects key points in detected hand region.
Expected Results	Whole hand is detected with key points
Actual Results	As expected

Table 5.5.3 Test case for working of Static Gesture Recognition Model

Test case ID	3
Test Case Name	Static Gesture Recognition model
Test Case description	Recognizing different hand gestures
Steps	1. Create the dataset to train the model for different hand gestures. 2. Train the model on the dataset to recognize different static hand gestures through a live video input. 3. The recognized static gesture is displayed as output.
Expected Results	Different static hand gestures being recognized and their name displayed as output.
Actual Results	As expected

Table 5.5.4 Test case for working of Dynamic Gesture Recognition model

Test case ID	4
Test Case Name	Dynamic Gesture Recognition model
Test Case description	Recognizing different dynamic hand gestures
Steps	1. Create the dataset to train the model for dynamic hand gestures. 2. Train the model on the dataset to recognize different dynamic hand gestures through a live video input. 3. The recognized dynamic gesture is displayed as output.
Expected Results	Different dynamic hand gestures being recognized and their name displayed as output.
Actual Results	As expected

Table 5.5.5 Test case for using of PyAutoGUI library

Test case ID	5
Test Case Name	Using PyAutoGUI
Test Case description	Using different shortcut keys using PyAutoGUI.
Steps	Using PyAutoGUI to test different shortcut keys with keystrokes to perform different actions using python script.
Expected Results	Different shortcut keys working through a python script.
Actual Results	As expected

Table 5.5.6 Test case for Using computer functionalities with Hand

Test case ID	6
Test Case Name	Using computer functionalities with Hand
Test Case description	Using hand gesture recognition to trigger PyAutoGUI commands.
Steps	1. Using trained model to recognize gesture. 2. Using recognized gesture to trigger a PyAutoGUI command. 3. Perform the shortcut key actions on the PC.
Expected Results	Perform different shortcut key actions by recognizing hand gestures through video input.
Actual Results	As expected

CHAPTER 6
CONCLUSION

In summary, the "Controlling computer functionality with hand gestures" project sought to create a system that enables users to communicate with their computers using hand gestures. A camera is used to record hand motions, and computer vision algorithms are utilized to identify and interpret those movements into instructions that may be used to operate different computer operations.

The project required the installation of several parts, such as hardware and software, as well as testing and assessing how effectively the system worked. The test findings showed that the system could correctly identify hand gestures and carry out the associated orders.

Overall, the research serves as an effective illustration of how computer vision and machine learning methodologies may be used to create creative and cutting-edge human-computer interaction systems. The method created for this research has the potential to give users a more natural and intuitive way to interact with their computers, particularly in situations when using standard input devices like keyboards and mouse might not be as practical or comfortable. The initiative also demonstrates the possibilities for additional study and development in the field of computer vision as well as the continual improvements in that sector.

In addition to its potential practical applications, this study also has larger effects in the area of computer-human interaction. A larger trend towards the creation of more organic and intuitive interfaces may be seen in the usage of hand gestures as an input method. Designing user-friendly interfaces that cater to a wide range of people with various requirements and skills is crucial as computers becomes more widespread in our everyday lives.

The study also emphasizes how crucial multidisciplinary cooperation is to create cutting-edge technological answers. Computer vision, machine learning, and hardware design experts were called upon for the project. The project's ability to develop a system that is both technically complex and useful was made possible by combining expertise from several fields.
The project also raises significant ethical questions about the usage of gesture-based user interfaces. It is crucial to take privacy, security, and consent into account when using any technology that collects and processes personal data.

Future researchers and developers in this field will need to be proactive in resolving these problems and making sure that gesture-based interfaces are created in a responsible and moral way.

Overall, the "Controlling computer functionality with hand gestures" project shows how gesture-based interfaces have the potential to give people more comfortable and natural methods to interact with their computers. It also emphasizes how crucial multidisciplinary cooperation and ethical development methods are to the creation of cutting-edge technological solutions.

6.1 System Usability

Any human-computer interaction system must consider system usability, and the project "Controlling computer functionality with hand gestures" is no exception. Usability is the degree to which a system is simple to understand, simple to use, and offers a pleasurable user experience.

Usability was taken into consideration when designing the system for this project, with the aim of producing an interface that is easy to use and effective. In order to assess the system's usability, participants in a user study were required to carry out a number of activities using the gesture-based user interface. Data on a variety of usability parameters, such as job completion time, mistake rate, and user satisfaction, were gathered for the research.

The study's findings demonstrated the gesture-based interface's excellent usability, with participants expressing high levels of pleasure and saying that it was simple to understand and operate. The low mistake rate and quick job completion durations were further indications of the system's general usefulness.

The system's usefulness was aided by the use of basic hand motions that were easy to understand. Participants were able to pick up on the gestures fast since they were made to be simple to memorize and use. Additionally, the technology gave users visual cues to assist them comprehend which motions were recognized and which orders were being carried out.

The responsiveness of the system was another crucial element. The system was intended to react to human motions fast, giving immediate feedback and lowering the possibility of user annoyance or confusion. The system's responsiveness was made possible by the employment of hardware elements with low latency and computer vision algorithms that were optimized.
Overall, "Controlling computer functionality with hand gestures" serves as an excellent example of the value of usability in human-computer interaction systems. The research has demonstrated how gesture-based interfaces may provide people a more natural and intuitive way to interact with their computers by developing a system that is both simple to use and effective. The study's findings also emphasize how crucial user testing and assessment are to making sure that technological solutions are both efficient and useful.

6.2 Future Scope

1. Facial recognition can be added with gesture recognition, so only the recognized user's hand gesture is captured.
2. Different security techniques can be included to make it more secure.
3. More dynamic and static gestures can be added.
4. Size of training dataset can be increased to increase the accuracy of the model.

REFERENCES

[1] R. P. Sharma and G. K. Verma, "human computer interaction using hand gestures".

[2] M. Covarrubias, M. Bordegoni, and U. Cugini, "A hand gestural interaction system for handling a desktop haptic strip for shape rendering," *Sens Actuators A Phys*, vol. 233, pp. 500–511, Aug. 2015, doi: 10.1016/j.sna.2015.07.024.

[3] H. S. Adeen *et al.*, "RemoAct: Portable projected interface with hand gesture interaction," *Journal of Computer Science*, vol. 11, no. 5, pp. 738–749, 2015, doi: 10.3844/jcssp.2015.738.749.

[4] V. K. Adhikarla, G. Jakus, and J. Sodnik, "Design and evaluation of freehand gesture interaction for light field display," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9170, pp. 54–65. doi: 10.1007/978-3-319-20916-6_6.

[5] N. Lalithamani, "Gesture Control Using Single Camera for PC," in *Physics Procedia*, 2016, vol. 78, pp. 146–152. doi: 10.1016/j.procs.2016.02.024.

[6] L. Zhao, "Gesture Control Technology: An investigation on the potential use in Higher Education," 2016. [Online]. Available: http://mgestyk.com/

[7] A. Joshi, "Real Time Monitoring of CCTV Camera Images Using Object Detectors and Scene Classification for Retail and Surveillance Applications," 2017.

[8] S. Song, D. Yan, and Y. Xie, "Design of control system based on hand gesture recognition," in *ICNSC 2018 - 15th IEEE International Conference on Networking, Sensing and Control*, May 2018, pp. 1–4. doi: 10.1109/ICNSC.2018.8361351.

[9] E. Togootogtokh, T. K. Shih, W. G. C. W. Kumara, S. J. Wu, S. W. Sun, and H. H. Chang, "3D finger tracking and recognition image processing for real-time music playing with depth sensors," *Multimed Tools Appl*, vol. 77, no. 8, pp. 9233–9248, Apr. 2018, doi: 10.1007/s11042-017-4784-9.

[10] K. K. Dubey, A. Jha, A. Tiwari, and K. Narmatha, "HAND GESTURE MOVEMENT RECOGNITION SYSTEM USING CONVOLUTION NEURAL NETWORK ALGORITHM," *IRJCS:: International Research Journal of Computer Science*, vol. VI, pp. 154–160, 2019, doi: 10.26562/IRJCS.2019.APCS10090.

[11] T. Vuletic, A. Duffy, L. Hay, C. McTeague, G. Campbell, and M. Grealy, "Systematic literature review of hand gestures used in human computer interaction interfaces," *International Journal of Human Computer Studies*, vol. 129, pp. 74–94, Sep. 2019, doi: 10.1016/j.ijhcs.2019.03.011.

[12] O. Köpüklü, A. Gunduz, N. Kose, and G. Rigoll, "Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks," Jan. 2019, [Online]. Available: http://arxiv.org/abs/1901.10323

[13] V. Singh, S. Singh, and P. Gupta, "Real-Time Anomaly Recognition Through CCTV Using Neural Networks," in *Procedia Computer Science*, 2020, vol. 173, pp. 254–263. doi: 10.1016/j.procs.2020.06.030.

[14] S. K. Sriram and N. Sinha, "Gestop: Customizable Gesture Control of Computer Systems," in *ACM International Conference Proceeding Series*, Jan. 2020, pp. 405–409. doi: 10.1145/3430984.3430993.

[15] V. Niranjani, R. Keerthana, B. Mohana Priya, K. Nekalya, and A. K. Padmanabhan, "System application control based on Hand gesture using Deep learning," in *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021*, Mar. 2021, pp. 1644–1649. doi: 10.1109/ICACCS51430.2021.9441732.

[16] A. Mujahid *et al.*, "Real-time hand gesture recognition based on deep learning YOLOv3 model," *Applied Sciences (Switzerland)*, vol. 11, no. 9, May 2021, doi: 10.3390/app11094164.

ANNEXURE 1

ANNEXURE 2