SMART SYSTEM FOR PRESENTATIONS

USING GESTURE CONTROL

Submitted in partial fulfilment of the Requirements for the award of
Bachelor of Engineering Degree in n Computer Science and Engineering

By

JYOTHI RAGHAVENDRA REDDY BATCHU (39110420)
A. MAHESHWAR REDDY (39110042)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,

CHENNAI – 600119

April 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with ¯A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **JYOTHI RAGHAVENDRA REDDY (39110420) AND A.MAHESHWAR REDDY (39110042)** who carried out theproject entitled " **Smart system for presentations using gesture control"**under my supervision.

Internal Guide
Dr. S. JAYANTHI M.E., Ph.D

Head of the Department
Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on 20.04.2023

Internal Examiner **External Examiner**

DECLARATION

We, JYOTHI RAGHAVENDRA REDDY BATCHU (Reg.39110420) and A MAHESHWAR REDDY (Reg.39110042) Hereby declare that the project Report entitled "**Smart system for presentations using gesture control"** done by me under the guidance of **DR.S.JAYANTHI ., Ph.D.,** is submitted in partial fulfilment of the requirement for the award of bachelor of Engineering degree in computer science and Engineering

DATE

PLACE: Chennai **SIGNATURE OF CANDIDATES**

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. Tsasikala M.E., Ph. D, Dean,** School of COMPUTING**, Dr. L. Lakshmanan, M.E., Ph.D.,** Heads of the Department of Computer Science AndEngineering for providing us necessary support and details at the right time during the progressive reviews

We would like to express my sincere and deep sense of gratitude to our Project Guide **Dr.S.Jayanthi., Ph.D.,** Associate professor, Dept. of Computer science and Engineering, for her valuable guidance, suggestions and constant encouragement for the successful completion of our project work.

We wish to express my thanks toal Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The development of technology for supporting learning systems at this time takes place very rapidly. Human Computer Interaction provides users with the ability to control presentations in a natural way by their body gestures. wepropose a simple system that can be used to control presentation by hand gestures using computer vision. This system primarily employs a web camera to record or capture photos and videos, and this application regulates the system's presentation based on the input. The primary purpose of the system is to change its presentation slides, I also had access to a pointer that allowed me draw on slides, in addition to that erase. To operate a computer's fundamental functions, such as presentation control, we may utilise hand gestures. People won't have to acquire the often burdensome machine-like abilities as a result. These hand gesture systems offer a modern, inventive, and natural means of nonverbal communication. These systems are used widely in human computer interaction. This project's purpose is to discuss a presentation control system based on hand gesture detection and hand gesture recognition. A high resolutioncamera is used in this system to recognise the user's gestures as input. The main objective of hand gesture recognition is to develop a system that can recognise human hand gestures and use that information to control a presentation. With real-time gesture recognition, a specific user can control a computer by making hand gestures in front of a system camera that is connected to a computer. With the aid of Open CV Python and Media Pipe, we are creating a hand gesture presentation control system in this project. Without using a keyboard or mouse, this system can be operated with hand gesture

CHAPTER NUMBER	TITLE	PAGE NUMBER`
	ABSTRACT	V
	LIST OF FIGURES	18
1	**Chapter 1**	1
	Introduction	
2	**Chapter 2**	2 - 7
	2.1 Literature survey 2.2 Existing System	
3	**Chapter 3**	8
	3.1 Software Requirements Specification 3.2 Hardware Requirements Specification	
4	**Chapter 4**	9-12
	4.1 Problem Statement 4.2 Objectives 4.3 Scope of the project 4.4 Visualization	
5	**Chapter 5**	13-18
	5.1 Proposed Methodology 5.2 Block Diagram	
6	**Chapter 6**	19 - 22
	6.1 Simple Code	

7	**Chapter 7**	23 - 25
	Result	
8	**Chapter 8**	26
	Conclusion	
9	**Chapter 9**	27 - 28
	Reference	

CHAPTER 1
INTRODUCTION

With the massive influx and advancement of technologies, a computer system has become a very powerful machine which has been designed to make the human beings' tasks easier. Due to which the HCI (human – computer interaction) has become an important part of our lives. Now-a-days, the progress and development in interaction with computing devices has increased so fast that as a human being even we could not remained left with the effect of this and it has become our primary thing. The technologies has so much surrounded us and has made a place in our lives that we use it to communicate, shop, work and even entertain ourselves1.

There are many applications like media player, MS-office, Windows picture manager etc. which require natural and intuitive interface. Now-a-days most of the users uses keyboard, mouse, pen, Joysticks etc. to interact with computers, which are not enough for them. In the near future, these existing technologies which are available for the computing, communication and display will become a bottleneck and the advancement in these technologies will be required to make the system as natural as possible. Nevertheless the invention of mouse and keyboards by the researchers and engineers has been a great progress, there are still some situations where interaction with computer with the help of keyboard and mouse will not be enough

Vision-based and image processing systems has various applications in pattern recognition and moving robots navigation. It is a processing of input images producing output that is features or parameters related to images . Its application in robotics, surveillance, monitoring, tracking, and security systems makes it important and cover a wide range of applications worldwide. Object tracking is the main activity in computer vision and extracting its features is the basic principle. It has many applications in traffic control, human computer interaction, gesture recognition, augmented reality and surveillance

CHAPTER 2
2.1LITERA TURE SURVEY

A Review on the project is This paper aims to cover the various prevailing methods of deafmute communication interpreter system. The two broad classification of the communication methodologies used by the deaf –mute people are - Wearable Communication Device and Online Learning System. Under the Wearable communication method, there are Glove based system, Keypad method and Handicom Touch-screen. All the above mentioned three subdivided methods make use of various sensors, accelerometer. a suitable micro-controller. a text to speech conversion module, a keypad and a touch-screen

[1]

- **Title :** PYTHON BASED HAND GESTURE
- **Authors :** Ali A. Abed , Sarah A. Rahman
- **Year :** International Journal ,Volume 173 – No.4 , 2017
- **Algorithm:** Raspberry Pi
- **Drawback :** It founds convexity defects, which is the deepest point of deviation on contour By this, it can find the number of fingers extended and then it can perform different functions according to the number of fiextended
- **Result :** Controlling the number of fingersi n front of camera leads to command avoid obstacles in the way of the robot. A recognition rate of about98% is reached

[2]

- **Title :** HAND GESTURE MOVEMENT RECOGNITION SYSTEM
- **Authors :** Kundan Kumar Dubey , K. Narmatha
- **Year :** IRJCS/RS/Vol.06/Issue04/APCS10090 ,2019
- **Algorithm :** Convolution neural network
- **Drawback :** It is a frequent problem in machine learning For the proposed and recognition task, the region of interest is relatively small, causing the deceptive behaviors in the cnn learning, such as making an attempt to infer the hand gesture from non-related image areas

- **Result** : Network training is quite fast, requiring only ~50 minutes on hand gesture classification on asingle image using the propose network requires about 2.96 milliseconds (ms) on GPU. Class ifation runningtimes canbe substantially improved through run ning the network on pics batches (requiring 0.73 ms per photo witha ba tch size of 256)

[3]

- **Title:** Hand Gesture Recognition for Human Computer Interaction
- **Authors** : Aashni Haria , Shristi Poddar
- **Year** : Procedia Computer science , Volume 115 , pages 367-374 in 2017
- **Algorithm :** morphology,Pavildis
- **Drawback :** We were able to create a robust gesture recognition system that did not utilize any markers, hence making it more users friendly and low cost. In this gesture recognition system, we have aimed to provide gestures, covering almost all aspects of HCI such as system functionalities, launching of applicatio nns and opening some popular websits
- **Result :**In future we would like to improve the accuracy further and add more gestures to implement more functions. Finally, we target to extend our domain scenarios and apply ourtrackin g mechanism into a variety of hardware including digital and mobile devices. We also aim to extend this mechanism to a range of users including disabled users

[4]

- **Title** : HAND GESTURE RECOGNITION: A LITERATURE REVIEW
- **Authors** : Rafiqul Zaman Khan , Noor Adnan Ibraheem
- **Year** : IJAIA, Vol.3, No.4, July 2012

- **Algorithm :** neural networks
- **Drawback :** Orientation histogram method applied in [19] have some problems which are; similar gestures might have different orientation histograms and different gestures could have similar orientation histograms, besides that, the proposed method achieved well for any objects that dominate the image even if it is not the hand gesture
- **Result :**In this paper various methods are discussed for gesture recognition, these methods include from Neural Network, HMM, fuzzy c-means clustering, besides using orientation histogram for features representation. For dynamic gestures HMM tools are perfect and have shown its efficiency especially for robot control.

[5]

- **Title :** GESTURE CONTROL TECHNOLOGY
- **Authors :** Stephen M
- **Year** 2018
- **Algorithm :** practical
- **Drawback :** As mentioned in the 2011 Horizon Report, gesture-based computing is one of the key trends in education technology, and we are soon to see the implementation of this technology as it develops further. In higher education, gesture control technology could not only improve the learning experiences for the students, but also provide a new teaching method for lecturers
- **Result :** Gesture control technology has shown a great potential in education. Hui-Mei Hsu (2011) presented a complete analysis on the use of Kinect in education and concluded that it should really achieve the expected results, enhancing classroom interaction and participation, as well as improving the way that teachers can manipulate multimedia materials during the classes

[6]

- **Title :**HAND GESTURE RECOGNITION TOWARDS ENHANCING
- **Authors :**Tiago Cardoso , Joao Delgado
- **Year :** Procedia Computer science 67 ,419-429,2015
- **Algorithm :** 3rd degree polynomial
- **Drawback :** were developed for the Kinect SDK version The device used was the Xbox Kinect Sensor which has some differences in relation to its Windows counterpart (Kinect SDK was fully tested with Kinect Sensor for Windows which, besides API improvements, also implements a near mode
- **Result :** The implementation of Smart Cities concepts largely depend on the interaction means provided to citizens. The current proposal addresses gesture recognition, towards providing the ability of NUI concrete solutions for Smart Cities.The validation methods proved to be reliable in evaluating a template based approach and leaves no doubts of the good performance of the solution.

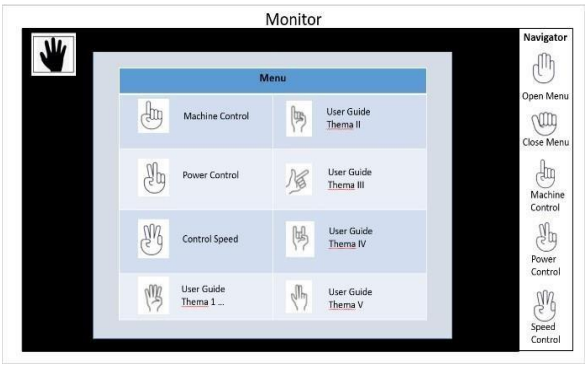
[7]

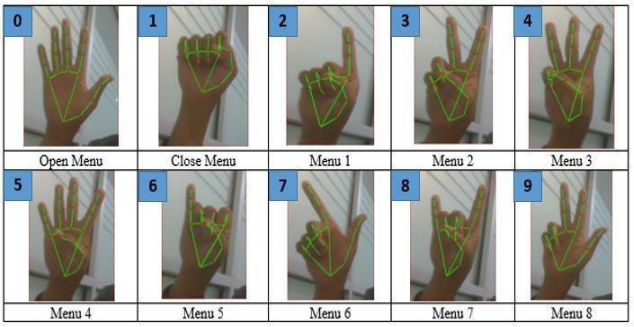
- **Title :** SMART PRESENTATION CONTROL BY HAND GESTURES
- **Authors :** Hajeera Khanum , Dr. Pramod H
- **Year :** Journal, Volume: 09 , 07 /July 2022
- **Algorithm :** polynomial
- **Drawback :**In order to generate a better result, we have implemented a Hand Gestures Recognition System. The webcam is turned on while the software is running, and the kind of gesture used to detect the shape of the hand and give us the desired output is static. This project uses the curve of the hand to regulate loudness. The system receives input, captures the item, detects it, and then recognises hand gestures
- **Result :** This project showcases a programme that enables hand gestures as a practical and simple method of software control. A gesture-

based presentation controller doesn't need any special markers, and it can be used in real life on basic PCs with inexpensive cameras since it doesn't need particularly high quality cameras to recognise or record the hand movements

[8]

- **Title** : Hand Gesture Recognition
- **Authors** :Moh.Harris , Ali Suryaperdana Agoes
- **Year** : Atlantis press,, volume 207,2021
- **Algorithm** : 3rd-degree polynomial equation
- **Drawback** :The main objective of the research is to recognize hand gestures to display one of the menus that a user has chosen through a Kinect. We used 10 captured hand gestures which each hand gesture directly set one menu out





I.

- **Result** : The measure of the performance on the model in machine learning used Confusion Matrix. In Python, we can use the library scikit-learn to develop a confusion matrix. Experiment datasets were obtained before we used them to predict the hand gestures. The Confusion Matrix was also used to observe an accuracy achieved for the model was made.

2.2 EXISTING SYSTEM

The author has developed an ANN application used for classification and gesture recognition, Gesture Recognition Utilizing Accelerometer. The Wii remote, which rotates in the X, Y, and Z directions, is essentially employed in this system. The author has utilised two tiers to construct the system in order to reduce the cost and memory requirements. The user is verified for gesture recognition at the first level. Author's preferred approach for gesture recognition is accelerometerbased.

That system signals are analysed at the second level utilising automata to recognise gestures (Fuzzy). The Fast Fourier technique and k means are then used to normalise the data. The accuracy of recognition has now increased to 95%. Recognition of Hand Gestures Using Hidden Markov Models - The author of this work has developed a system that uses dynamic hand movements to detect the digits 0 through 9. In this work, the author employed two stages. Preprocessing is done in the first phase, while categorization is done in the second. There are essentially two categories of gestures The author has employed inexpensive cameras to keep costs down for the consumers. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor. Although a kinect sensor's resolutions lower than that of other cameras, it is nevertheless capable of detecting and capturing large pictures and objects. Only the fingers, not the entire hand, are paired with FEMD to deal with the loud hand movements. This technology performs flawlessly and effectively in uncontrolled settings. The experimental result yields an accuracy of 93.2%.

The key gesture and the link gestures are employed in continuous gestures for the goal of spotting. Discrete Hidden Markov Model (DHMM) is employed for classification in this work. The Baum-Welch algorithm is used to train this DHMM. HMM has an average recognition rate range of 93.84 to 97.34%

CHAPTER 3

3.1 Hardware Requirements Specification Document

HARDWARE REQUIREMENTS:

- System : i5 processor
- Hard Disk : 500 GB.
- Monitor : 15'' LED O Input Devices ,Keyboard, Mouse
- Ram : 4GB.

3.2 Software Requirements Specification Document

SOFTWARE REQUIREMENTS:

- Operating system : Windows 10
- Coding Language : python
- Tool : Google colab , Jupyter notebook
- Database : MYSQL

CHAPTER 4

4.1 Problem Statement

Presentations are used widely in everyday lives. It is used in almost every single industry on a frequent basis.

But it is a hectic task for the speaker to navigate through the slides during the presentation. Either someone else has to operate the presentation or else the speaker should buy and carry the navigator remote with him all the time. He even has to make sure that the batteries are charged and the device isn't broken.

To remove all this friction we've come up with smart navigation using hand gestures.

4.2 Objectives

The project is about building a human-computer interaction systemusing hand gestures as a cheap alternative to depth cameras. We presenta robust , efficient and real-time technique using normal 2D -camera.We will be able to move the slides back and forth along with a pointer and drawing capabilities. And to make it more usable we will add an erasing gesture as well.

This will enhance the way of presenting compared to the current scenario.

4.3 Scope of the Project

In computer science and language technology, gesture recognition is an important topic which interpret human gesture through computer vision algorithms. There are various bodily motion which can originate gesture but the common form of gesture origination comes from the face and hands. The entire procedure of tracking gesture to their representation and converting them to some purposeful command is known as gesture recognition1. Various technologies has been used for the design and implementation of such kind of devices, but contact based and vision based technologies are two main types of technologies used for robust, accurate and reliable hand gesture recognition systems. Contact based devices like

accelerometers7, multi-touch screen, data glove9 etc. based on physical interaction of user who will be required to learn their usages. Whereas vision based devices like cameras has to deal with the prominent variety of gestures.

Gesture recognition involves to handle degrees of freedom4, 10 (DOF), variable 2D appearances, different silhouette scales (i.e. spatial resolution) and temporal dimension (i.e. gesture speed variability). Vision based gesture recognition further classified into two main categories, which are 3D model based methods and appearance based methods1.

3D based hand models4 describes the hand shapes and are the main choice of hand. gesture modeling in which volumetric analysis is done. In appearance based models4, the appearance of the arm and hand movements are directly linked from visual images to specific gestures. A large number of models belong to this group. We have followed one of these models i.e. silhouette geometry based models to recognize the gesture in our project.A fast, simple and effective gesture recognition algorithm for robot application has been presented which automatically recognizes a limited set of gestures. However, the segmentation process should be robust and required to be deal with temporal tracking, occlusion and 3D modelling of hand. The author of7 has used multi-stream Hidden Markov Models (HMMs) consisting of EMG sensors and 3D accelerometer (ACC) to provide user friendly environment for HCI.

However, there are some problems or limitations in ACC-based techniques and EMG measurement. In11, a method has been proposed which firstly store the human hand gesture into the disk, convert them into binary image by extracting frame from each video one by one and then creates 3D Euclidian space for binary image, for recognizing vision-based hand gesture. They have used back propagation algorithm and supervised feed-forward neural network based trainingfor classification

However it is suitable for only simple kind of gesture against the simple background. In12, a method for detecting finger from the detected hand, can be used as a non-

contact mouse, has been proposed. They have used skin color technique for segmentation and contour as the feature to locate the fingertip in hand. The authors in13 have used bag-of-features and multiclass SVM to detect and track bare hand, and to control an application using command generated by a grammar in a complex background, via skin detection and contour comparison algorithm.

They have also used K-means clustering algorithm and scale invariance feature transform (SIFT) to extract the main features from the trained images. However, the segmentation and localization method is unclear for the system and there is no rigorous geometric information of the object components. In14, the author has used Lucas KanadePyramidical Optical Flow algorithm to detect moving hand and K-means algorithm to find center of moving hand.

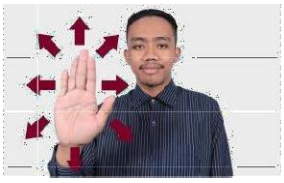
Here Principal Component Analysis (PCA) was used to extract features and then the extracted features were matched using K-nearest neighborh od. However, PCA

made whole system slower and required more memory. In15, a comparative analysis of different segmentation techniques and how to select an appropriate segmentation method for the system have been presented. It has also described Gaussian Model Classifier along with some other classification techniques

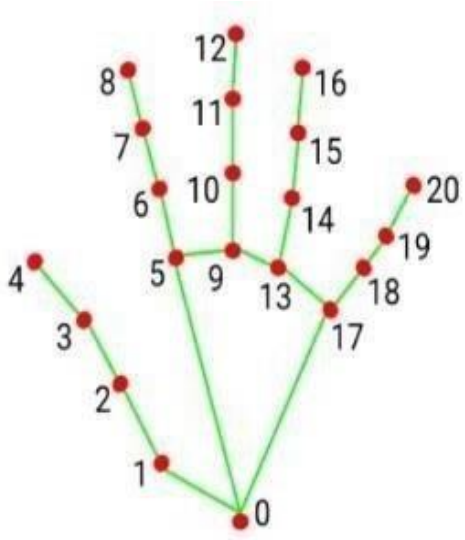
4.4 visualization

Accomplishes accurate crucial point clustering of 21 main points with only a 3D touch coordinates that is done within the identified hand areas and immediately

generates the coordinates predictor that is a representation of hand landmarks within MediaPipe

II.



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

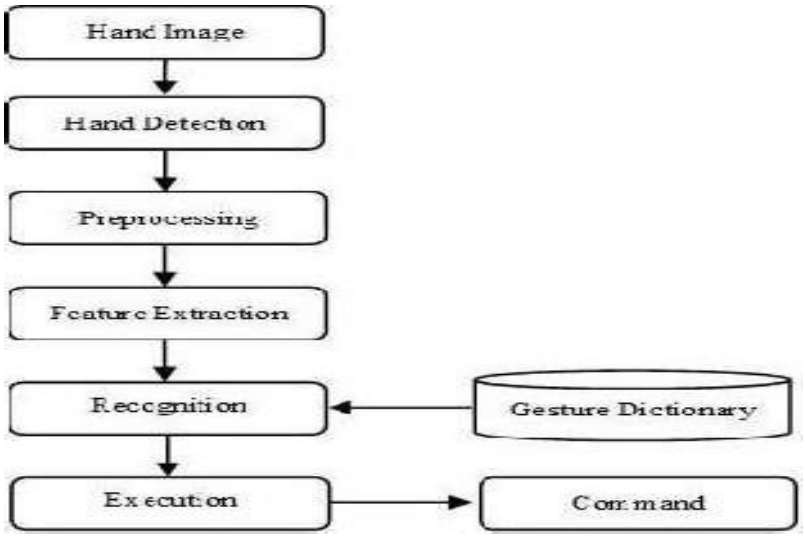
III.

IV.

CHAPTER 5

5.1 Proposed Methodology

The system we have proposed and designed for vision-based hand gesture recognition system contained various stages which we have explained through an algorithm. The working flowchart of gesture recognition system has also shown



V.

Region proposals (R-CNN, Fast R-CNN, Faster R-CNN, and cascade R-CNN) , the method proposes areas capable of containing the object and performs identification to save computational capacity.

we also use the CNN model to classify gestures. The goal of the algorithm is to detect gestures with real-time processing speed, minimize interference, and reduce the ability to capture unintentional gestures. The static gesture controls include on, off, up, and down inthis study

Fig. 1. Flowchart of hand gesture recognition.

VI.

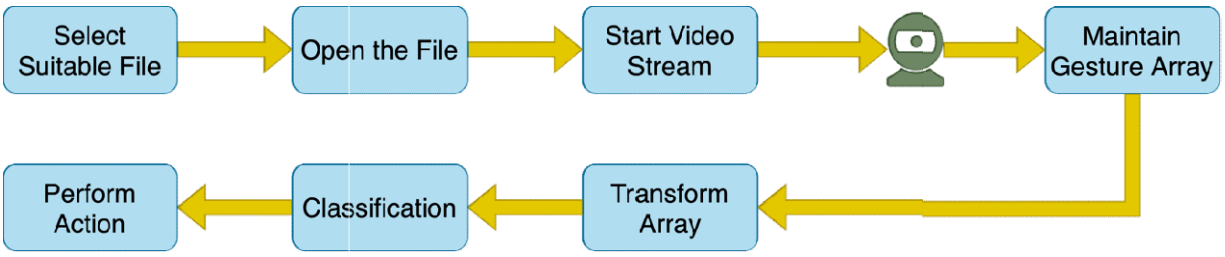
Fig. 2.Static hand gesture numbered class 1–6 in session 1 (recorded at a distance of 16cm approx.)

VII.

Fig. 3.Static hand gesture numbered class 1–6 in session 2 (recorded at a distance of 21cm. approx.)

VIII.

Step Diagram



IX.

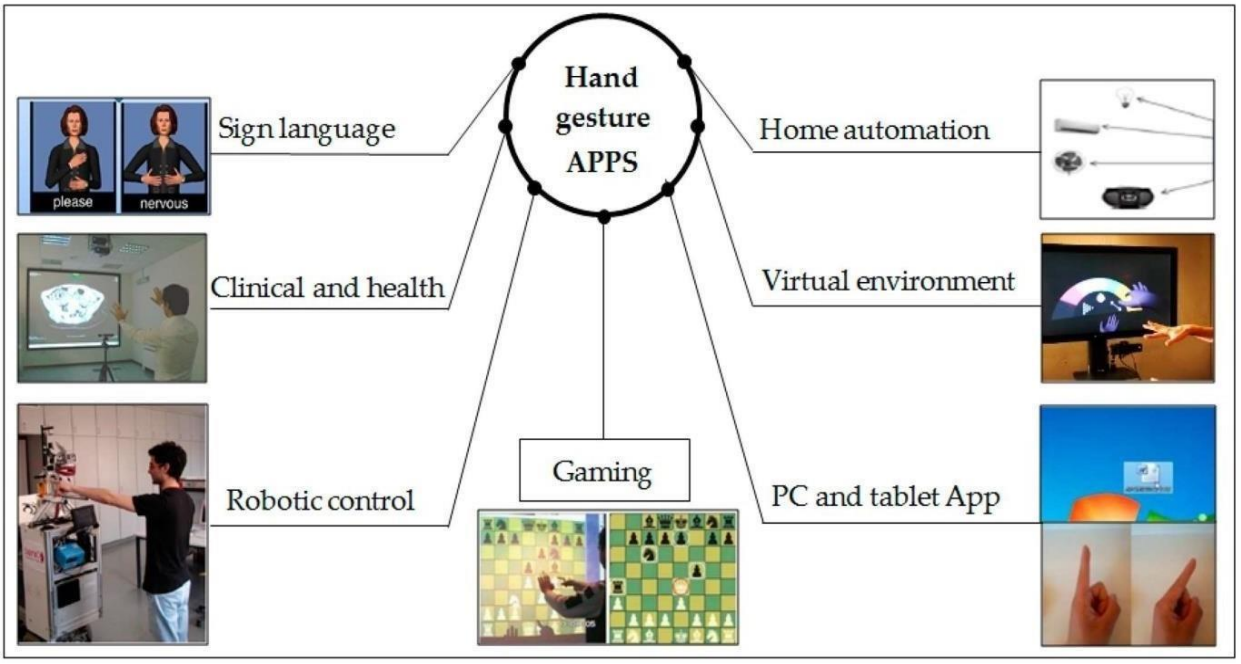
The methodology that we used for our project consists of different phases. The first and second phases include selecting and opening a PowerPoint file for presentation on PowerPoint Windows. The user selects the PowerPoint file to open, and our system will open the file for presentation. The user can select .ppt, pptx, .pptm files for presentation. After selecting, our program will automatically open the file.

In the third phase, the system starts a live video stream for detecting and recognizing the live gestures. A built-in or an external webcam will record this live stream. The gesturing will be recorded as an image array of size 20 in the fourth phase. This array will help detect a specific gesture. This array can be an entire performed gesture or action recorded frame by frame and fed to the network for

detection. This image array is an array of continuous frames where every frame is processed 20 times for gesture or action recognition. A transform function

transforms this array before predicting it for a specific action. The fifth phase transforms this array. The transform function used in our project is as follows

5.2 Block Diagram:



X.

(i) Toggle state switch is hand from spread state upwards, into grip state

(ii) Up order is hand from outstretched state up to left

(iii) Down order is hand from outstretched state up to right

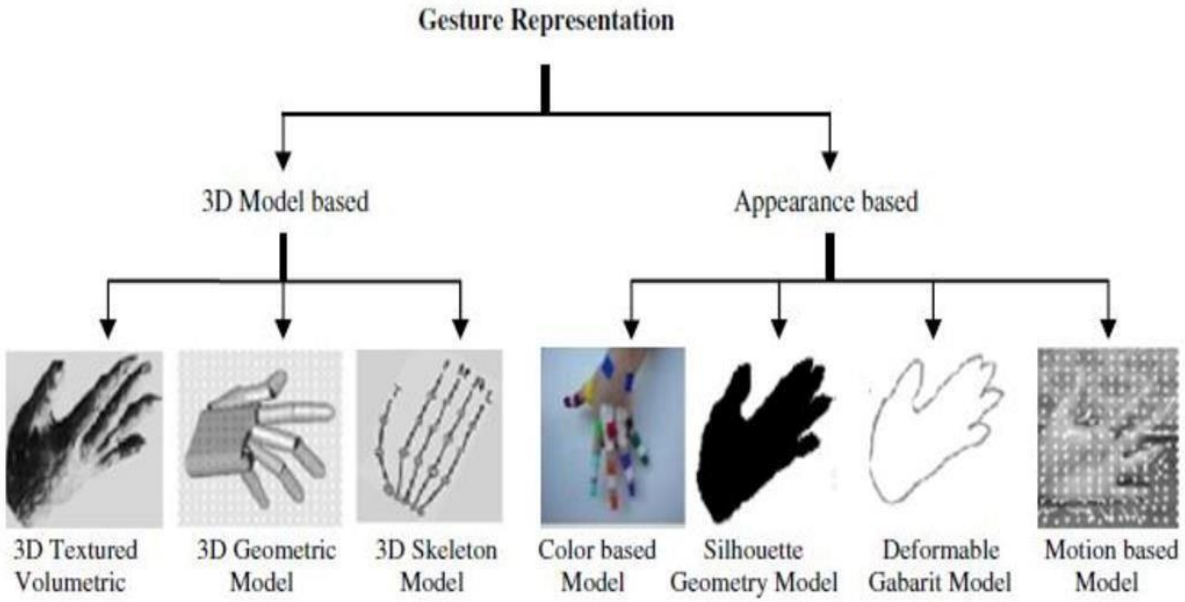
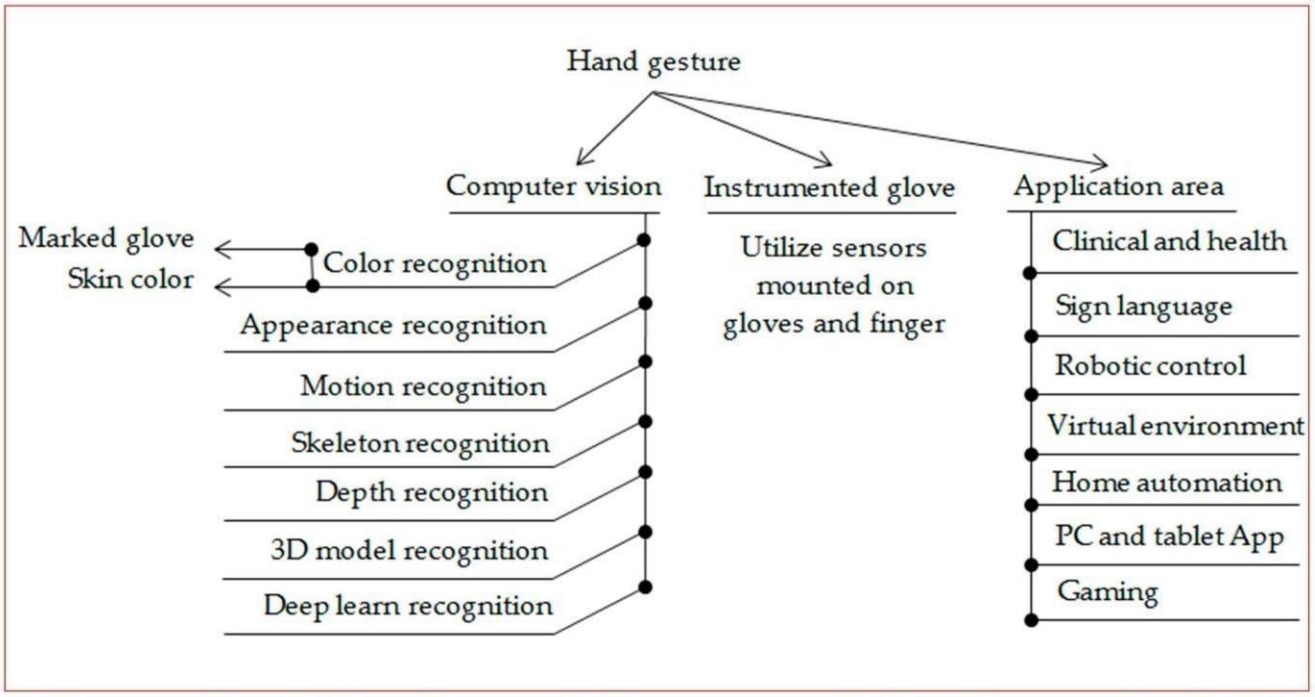
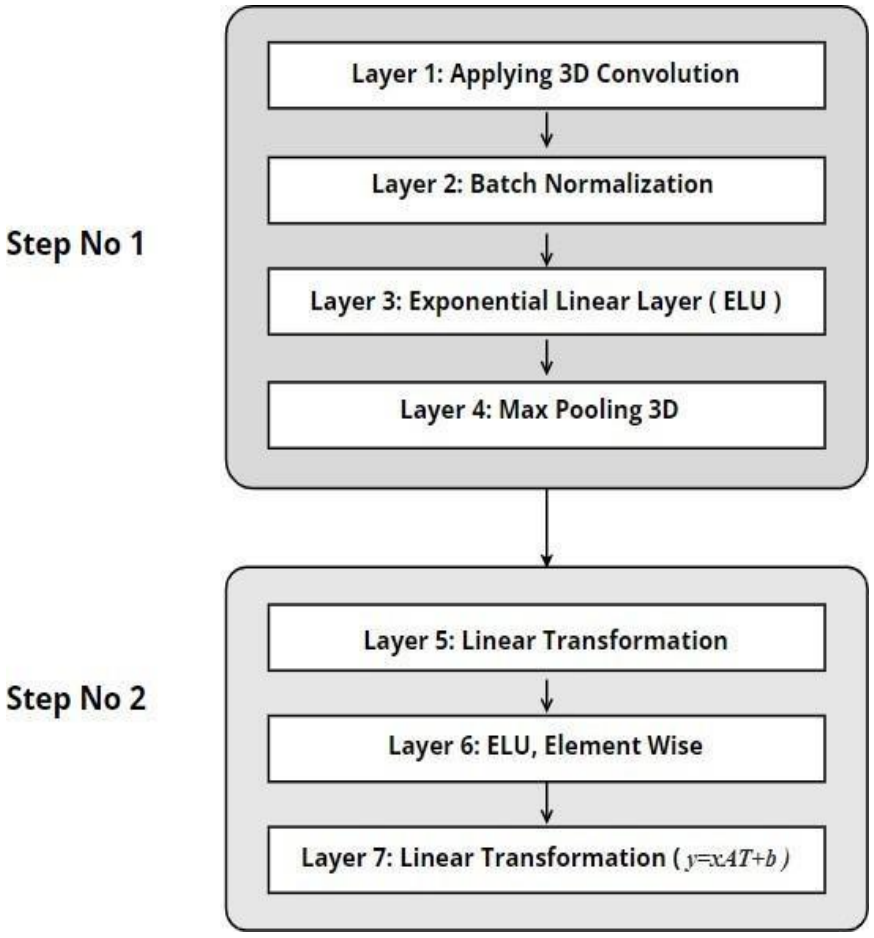


Fig. 3 Vision Based Hand Gesture Representation

XI.



XII.



XIII.

Zooming in with two fingers

CHAPTER 6

Simple Code

```
import cv2
import os
import numpy as np

# Parameters
width, height = 1280, 720
gestureThreshold = 300
folderPath = "Presentation"

# Camera Setup
cap = cv2.VideoCapture(0)
cap.set(3,   width)
cap.set(4, height)

# Hand Detector
detectorHand = HandDetector(detectionCon=0.8, maxHands=1)

#  Variables
imgList  =  []
delay = 30
buttonPressed = False
counter = 0
drawMode = False
imgNumber = 0
delayCounter = 0
annotations = [[]]
annotationNumber = -1
annotationStart = False
hs, ws = int(120 * 1), int(213 * 1) # width and height of small image
```

```
# Get list of presentation images
pathImages = sorted(os.listdir(folderPath), key=len)
print(pathImages)

while True:
# Get image frame
success, img = cap.read()
img = cv2.flip(img, 1)
pathFullImage = os.path.join(folderPath, pathImages[imgNumber])
imgCurrent = cv2.imread(pathFullImage)

# Find the hand and its landmarks
hands, img = detectorHand.findHands(img) # with draw
# Draw Gesture Threshold line
cv2.line(img, (0, gestureThreshold), (width, gestureThreshold), (0, 255, 0), 10)

if hands and buttonPressed is False: # If hand is detected

hand = hands[0]
cx, cy = hand["center"]
lmList = hand["lmList"] # List of 21 Landmark points
fingers = detectorHand.fingersUp(hand) # List of which fingers are up

# Constrain values for easier drawing
xVal = int(np.interp(lmList[8][0], [width // 2, width], [0, width]))
yVal = int(np.interp(lmList[8][1], [150, height-150], [0, height]))
indexFinger = xVal, yVal

if cy <= gestureThreshold: # If hand is at the height of the face
if fingers == [1, 0, 0, 0, 0]:
print("Left")
buttonPressed = True
if imgNumber > 0:
imgNumber -= 1
```

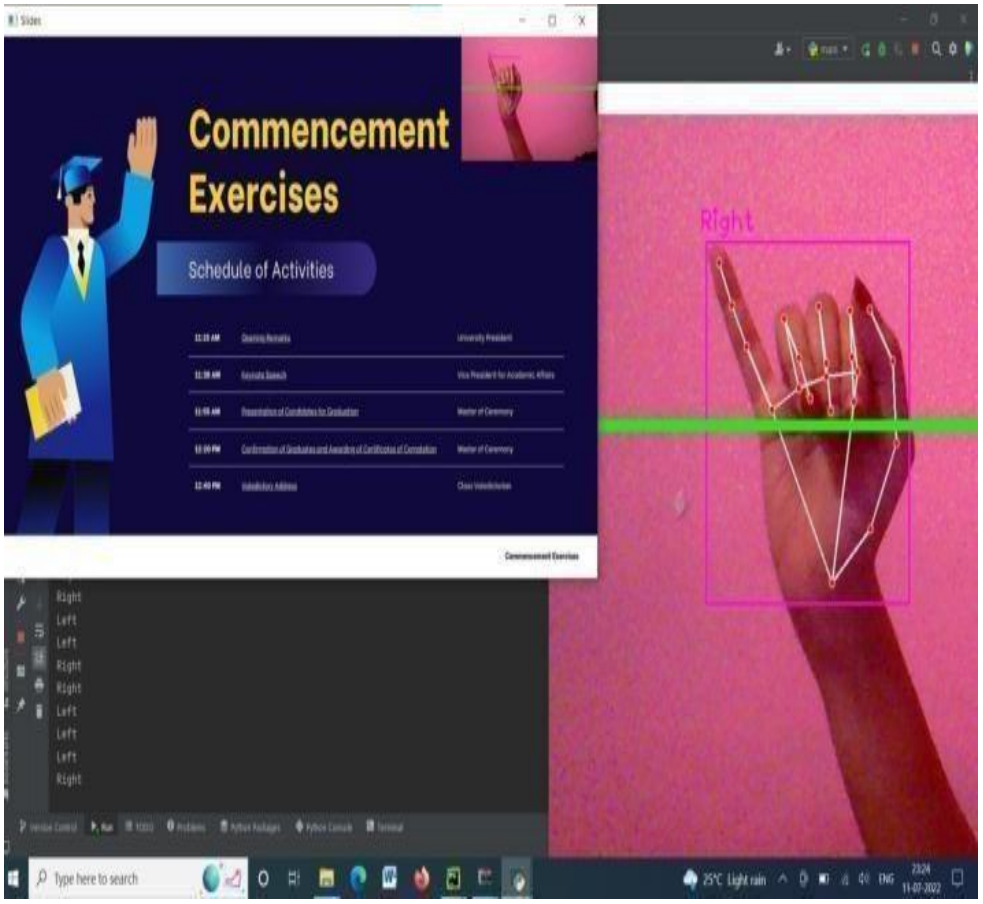
```
annotations = [[]]
annotationNumber = -1
annotationStart = False
if fingers == [0, 0, 0, 0, 1]:
print("Right")
buttonPressed = True
if imgNumber < len(pathImages) - 1:
imgNumber += 1
annotations = [[]]
annotationNumber = -1
annotationStart = False

if fingers == [0, 1, 1, 0, 0]:
cv2.circle(imgCurrent, indexFinger, 12, (0, 0, 255), cv2.FILLED)

if fingers == [0, 1, 0, 0, 0]:
if annotationStart is False:
annotationStart = True
annotationNumber += 1
annotations.append([])
print(annotationNumber)
annotations[annotationNumber].append(indexFinger)
cv2.circle(imgCurrent, indexFinger, 12, (0, 0, 255), cv2.FILLED)

else:
annotationStart = False

if fingers == [0, 1, 1, 1, 0]:
if annotations:
annotations.pop(-1)
annotationNumber -= 1
buttonPressed = True

else:
```

```
annotationStart = False

if buttonPressed:
counter += 1
if counter > delay:
counter = 0
buttonPressed = False

for i, annotation in enumerate(annotations):
for j in range(len(annotation)):
if j != 0:
cv2.line(imgCurrent, annotation[j - 1], annotation[j], (0, 0, 200), 12)

imgSmall = cv2.resize(img, (ws, hs))
h, w, _ = imgCurrent.shape
imgCurrent[0:hs, w - ws: w] = imgSmall

cv2.imshow("Slides", imgCurrent)
cv2.imshow("Image", img)

key = cv2.waitKey(1)
if key == ord('q'):
break
```

CHAPTER 7

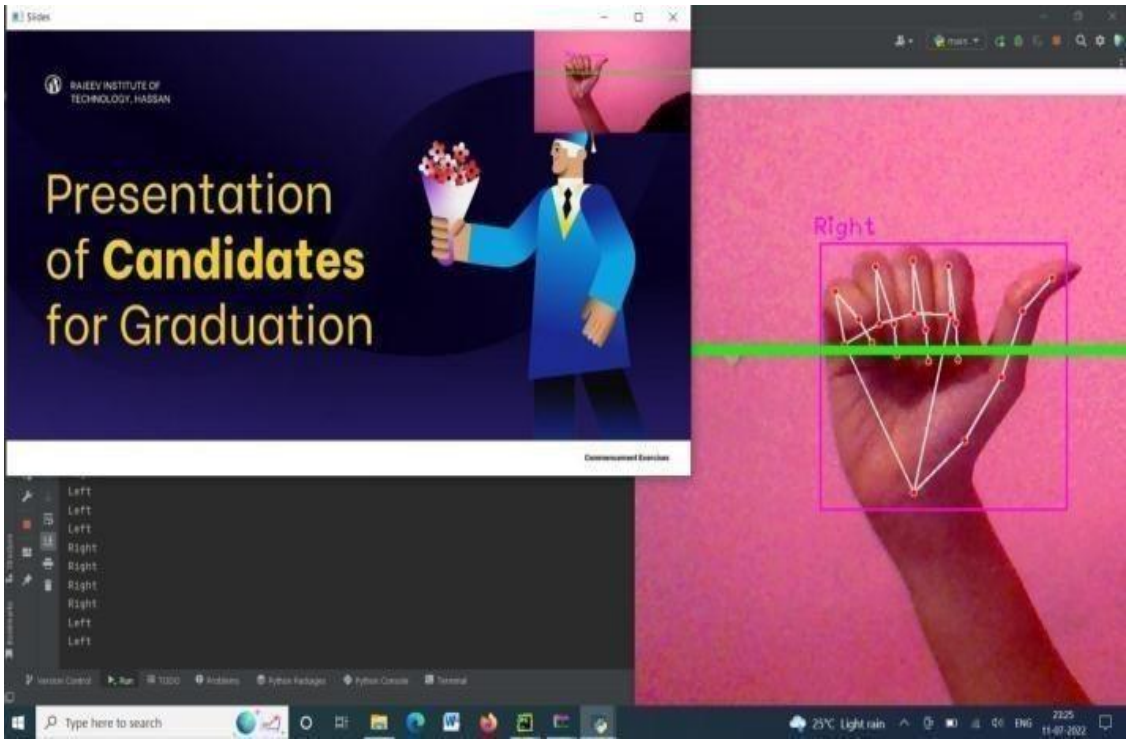
Result

A real-time simulation of the architecture with input from Gesture dataset (on left side) and real-time (online) classification scores of each gesture (on right side) are shown, where each class is annotated with different fingers



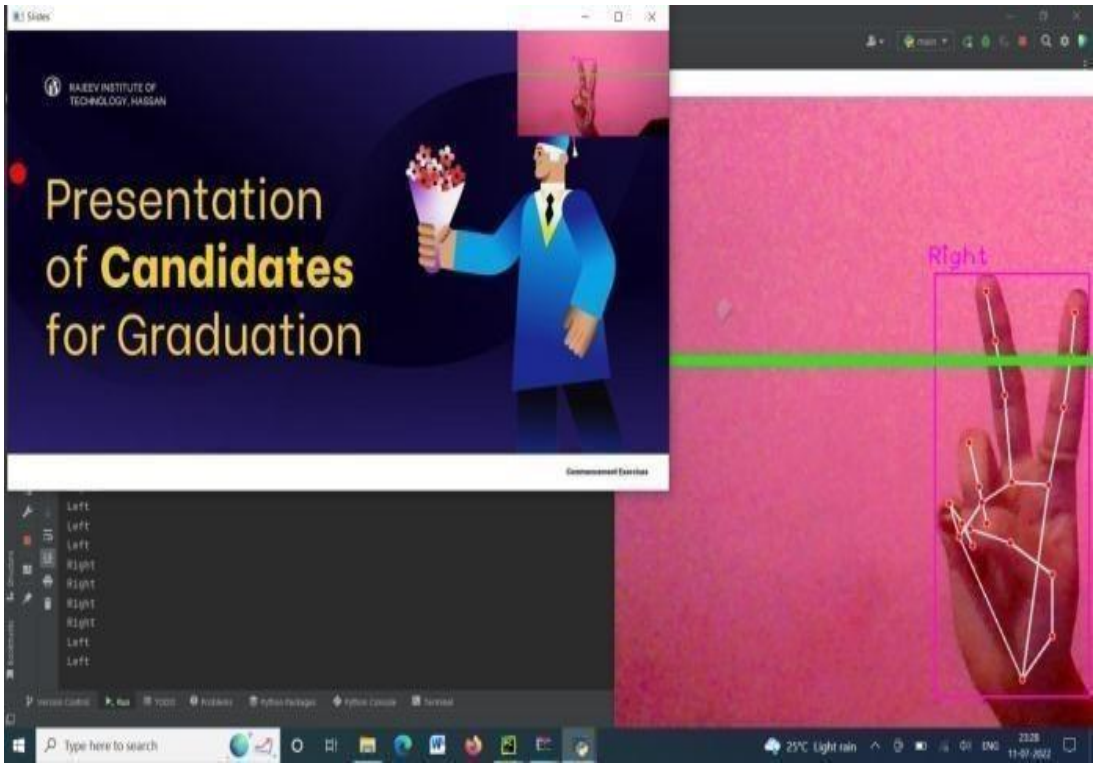
Hand Gesture to move on to the next slide

XIV.



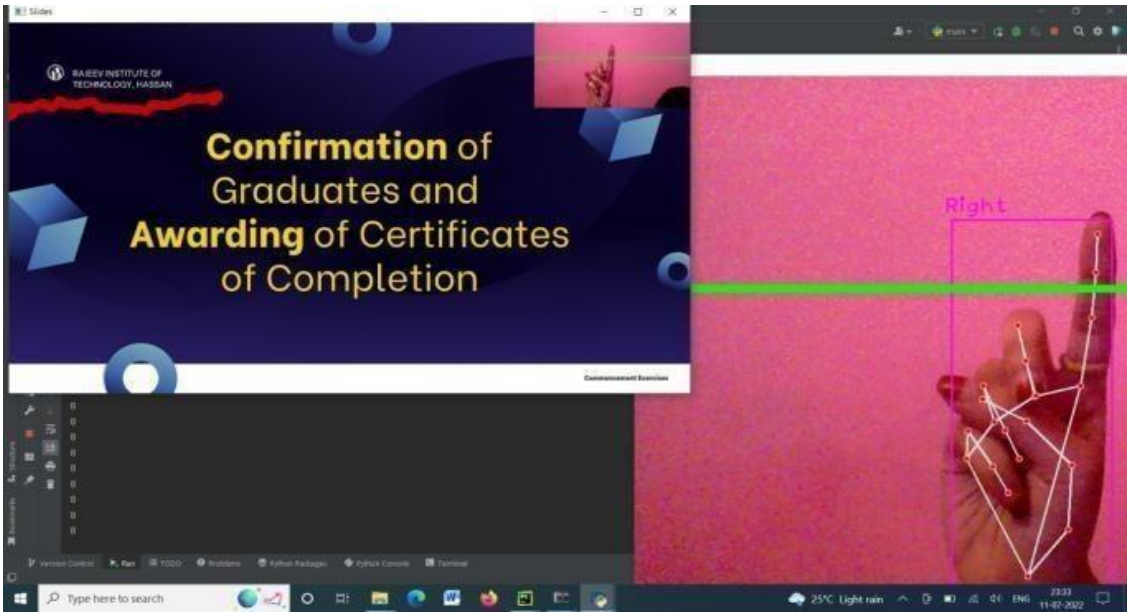
Hand Gesture for going back to previous slide

XV.



XVI.

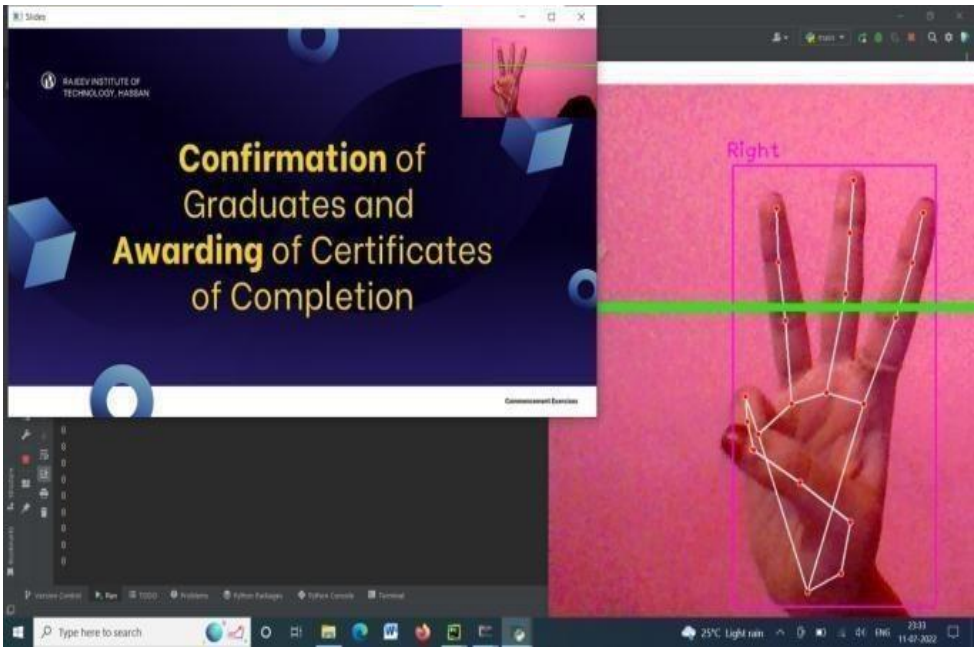
Getting a pointer on slide



Draw using that pointer

XVII.

Erase the drawing on slide



XVIII.

CHAPTER 8

Conclusion

This project showcases a programme that enables hand gestures as a practical and simple method of software control. A gesture-based presentation controller doesn't need any special markers, and it can be used in real life on basic PCs with inexpensive cameras since it doesn't need particularly high quality cameras to recognise or record the hand movements. The method keeps track of the locations of each hand's index finger and counter tips.

This kind of system's primary goal is to essentially automate system components so that they are easy to control. As a result, we have employed this method to make the system simpler to control with the aid of these applications in order to make it realistic. In this modern world, where technologies is at the peak, there are many facilities available for offering input to any applications running on the computer systems, some of the inputs can be offered using physical touch and some of them without using physical touch (like speech, hand gestures, head gestures etc.

Using hand gestures many users can handle applications from distance without even touching it. But there are many applications which cannot be controlled using hand gestures as an input. This technique can be very helpful for physically challenged people because they can define the gesture according to their need. The present system which we have implemented although seems to be user friendly as compared to modern device or command based system but it is less robust in detection and recognition as we have seen in the previous step.

We need to improve our system and try to build more robust algorithm for both recognition and detection even in the cluttered background and a normal lighting condition. We also need to extend the system for some more class of gestures as we have implemented it for only 6 classes. However we can use this system to control applications like power point presentation, games, media player, windows picture manager etc

.

.

CHAPTER9

References

1. A., A., & A., S. (2017). Python-based Raspberry Pi for Hand Gesture Recognition. International Journal of Computer Applications, 173(4), 18–24. https://doi.org/10.5120/IJCA2017915285
2. Al Saedi, A. K. H., & Al Asadi, A. H. H. (2020). A new hand gestures recognition system. Indonesian Journal of Electrical Engineering and Computer Science, 18(1), 49–55. https://doi.org/10.11591/IJEECS.V18.I1.PP49-55
3. Escalera, S., Gonzàlez, J., Baró, X., Reyes, M., Guyon, I., Athitsos, V., Escalante, H., Sigal, L., Argyros, A., Sminchisescu, C., Bowden, R., & Sclar off,
   S. (2013). Cha Learn multi-modal
   gesture recognition 2013: Grand challenge and workshop summary. ICMI 2013 - Proceedings of the 2013 ACM International Conference on Multimodal Interaction, 365– 370. https://doi.org/10.1145/2522848.2532597
4. Fourney, A., Terry, M. J., & Mann, R. A. (2010, September 1). Gesturing in the Wild:
   Understanding the Effects and Implications of Gesture-Based Interaction for Dynamic Presentations. Proceedings of the 2010 British Computer Society Conference on Human-
   Computer Interaction. https://doi.org/10.14236/EWIC/HCI2010.29
5. Haria, A., Subramanian, A., Asok kumar, N., Poddar, S., & Nayak, J. S. (2017). Hand Gesture Recognition for Human Computer Interaction. Procedia Computer Science, 115, 367–374. https://doi.org/10.1016/J.PROCS.2017.09.092
6. M, S. (2018). Static Hand Gesture Recognition for PowerPoint Presentation Navigation using Thinning Method. International Journal on Recent and Innovation Trends in Computing and Communication,6(4),187–189. https://doi.org/10.17762/IJRITCC.V6I4.1541
7. Ma, X., & Peng, J. (2018). Kinect sensor-based long-distance hand gesture recognition and fingertip detection with depth information. Journal of Sensors, 2018. https://doi.org/10.1155/2018/5809769
8. Marcel, S., Bernier, O., Viallet, J. E., & Collobert, D. (2000). Hand gesture recognition using input-output hidden Markov models. Proceedings - 4th IEEE

International Conference on Automatic Face and Gesture Recognition, \FG 2000, 456–461. https://doi.org/10.1109/AFGR.2000.840674

9. Marin, G., Dominio, F., & Zanuttigh, P. (2014). Hand gesture recognition with leap motionand kinect devices. 2014 IEEE International Conference on Image Processing, ICIP 2014, 1565–1569. https://doi.org/10.1109/ICIP.2014.7025313
10. Marin, G., Dominio, F., & Zanuttigh, P. (2015). Hand gesture recognition with jointlycalibrated Leap Motion and depth sensor. Multimedia Tools and Applications 2015 75:22, 75(22), 14991–15015. https://doi.org/10.1007/S11042- 015-2451-6
11. Materzynska, J., Berger, G., Bax, I., & Memisevic, R. (2019). The jester dataset: A large-scale video dataset of human gestures. Proceedings - 2019 International Conference on ComputerVisionWorkshop,ICCVW2019,2874–2882. https://doi.org/10.1109/ICCVW.2019.00349
12. Memo, A., Minto, L., & Zanuttigh, P. (2015). Exploiting silhouette descriptors and synthetic data for hand gesture recognition. Italian Chapter Conference 2015 - Smart Tools and Apps in Computer Graphics, STAG 2015, 15–23. https://doi.org/10.2312/STAG.20151288