

PROFESSIONAL TRAINING REPORT

at

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University)

**Submitted in partial fulfilment of the requirements for the
award of Bachelor of Technology in**

Information Technology

By

SIVAKUMAR S

(REG NO 38120098)



DEPARTMENT OF Technology

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

NOVEMBER 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)



Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI– 600119

www.sathyabama.ac.in

DEPARTMENT OF Information Technology

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Sivakumar S(38120098)** who carried out the project entitled “**Virtual Drag and Drop control using Hand Tracking Module in Machine Learning**” under my supervision from June 2021 to November 2021.

Internal Guide

Dr.Sendurusrinivasulu, M.E., Ph.D.,

Head of the Department

Dr.R.Subhashini, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I, **Sivakumar S(38120098)** hereby declare that the Project Report entitled as **“Virtual Drag and Drop control using Hand Tracking Module in Machine Learning”** done by me under the guidance of **Dr.Sendurusrinivasulu, M.E., Ph.d.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology Degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D.,Dean**, School of Computing, **Dr.R.Subhashini M.E., Ph.D.,Head of the Department** of Information Technologyfor providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Sendurusrinivasulu, M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

Training Certificate

ABSTRACT

This project is developed to control the volume of a computer using real-time hand gesture recognition using Machine Learning modules. The main aim of the project is to create a positive human-computer interaction with augmented reality technology.

Hand gesture recognition is very significant for human-computer interaction. In this work, we present a novel real-time method for hand gesture recognition for Volume control in a personal computer. In our framework, the hand region is extracted from the background with the background subtraction method. Then, the palm and fingers are segmented so as to detect and recognize the fingers. Finally, a rule classifier is applied to predict the labels of hand gestures. The gestures are recognised and according to that, the volume of the system is controlled. The experiments on the data set of images show that our method performs well and is highly efficient.

Moreover, our method shows better performance than a state-of-art method on another data set of hand gestures. The volume control using hand gesture is where the people can use more efficiently. It is very useful in many ways for various kinds of people, it can be used in a conference or in meeting to make it easier to control the audio, it is very helpful for old people to control the audio without a physical interaction. We created this project to make easier way to control the audio of the system and to utilize the hand gesture recognition in an efficient way.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	Abstract	i
	List of Abbreviations	iv
1.	INTRODUCTION	1
	1.1 Outline of The Project	1
	1.2 Purpose of The Project	1
	1.3 Problem in Existing System	2
	1.4 Proposed System	2
	1.5 Aim of the Project	3
	1.6 Scope of the Project	3
2	SOFTWARE and HARDWARE REQUIREMENTS	4
	2.1 Software Requirements	4
	2.2 Network Requirements	8
	2.3 Hardware Requirements	8
3	FRAMEWORKS and METHODS	9
	3.1 Frameworks and Libraries	9
	3.2 API Methods	10

4.	RESULTS and DISCUSSIONS	12
	4.1 Test and Results	12
5.	CONCLUSION	13
	5.1 Conclusion	13
	5.2 Future work	13
	APPENDIX	14
	Screen shots	14
	Source code	20
	REFERENCE	34

LIST OF ABBREVIATIONS

ACRONYM	EXPANSION
ML	Machine Learning
HTM	Hand Tracking Module
MP	Media Pipe
CV2	OpenCV
HCI	Human Computer Interaction
NP	NumPy
HTTP	HyperText Protocol
WWW	World Wide Website
VS CODE	Visual Studio Code

CHAPTER - 1

INTRODUCTION

1.1 Outline of the Project

As we know, the vision-based technology of hand gesture recognition is an important part of human-computer interaction (HCI). In the last decades, keyboard and mouse play a significant role in human-computer interaction. However, owing to the rapid development of hardware and software, new types of HCI methods have been required. In particular, technologies such as speech recognition and gesture recognition receive great attention in the field of HCI. HCI are used successfully in many areas today, particularly in industrial production, military operations, deep sea drilling, and space exploration. This success drives the interest in the feasibility of using computer in human social environments, particularly in the care of the aged and the handicapped. In social environments, humans communicate easily and naturally by both speech (audio) and gesture (vision) without the use of any external devices (like keyboards) requiring special training. Computer have to adapt to human modes of communication to promote a more natural interaction with humans. Given a choice between speech and gesture, some researchers have opined that gesture recognition would be more reliable than speech recognition because the latter would need a greater number of training datasets to deal with the greater variability in human voice and speech. This project is about implementing the control of a volume through simple hand gestures. The main motivation is the desirability of developing robots that can interact smoothly with humans without the need of any special devices.

1.2 Purpose of the Project

The main purpose of the project is to create an easier way to communicate with the system for controlling the volume of the media being played and also to create a better human-computer interaction (HCI) for next level of technology. Through this project, many people are getting an advantage to avoid a physical contact with the hardware systems (mouse or keyboard). Hand gesture recognition is

very significant for human-computer interaction. In this work, we present a novel real-time method for hand gesture recognition for Volume control in a personal computer. It is very useful in many ways for various kinds of people, it can be used in a conference or in meeting to make it easier to control the audio, it is very helpful for old people to control the audio without a physical interaction.

1.3 Problem in the Existing System

The previous system of use the HCI to create various specifications using Hand Tracking Module (HTM), but fails to reach the expected output in volume control. The previous system is not upgraded till date. Its is being is developed by the previous modules of machine learning. Accuracy of the prediction will be in intermediate level. The hand tracking system is not predicted as expected and the software will run only when the application in opened. Facilities of the system are less, like the prediction accuracy, nature of response, etc.

1.4 Proposed System

The proposed system is to create a volume control for a computer using gestures with hand tracking modules. The system revolves around the concept of Augmented reality and Human Computer Interaction. It uses the Machine Learning modules; Media Pipe, cvzone, OpenCV, HTM, NumPy and math.

1.4.1 Advantages

- Efficient way to communicate with computer
- Time saving
- No physical involvement
- End to end acceleration

1.5 AIM OF THE PROJECT

The aim of the project is to create an easier way to communicate with the system for controlling the volume of the media being played and also to create a better HCI. We created this project to make easier way to control the audio of the system and to utilize the hand gesture recognition in an efficient way. The further

aim to save time and energy of the people and to avoid physical contact with the hardware devices.

1.6 SCOPE OF PROJECT

1.6.1 GOALS

- To avoid physical interaction with devices.
- Efficient way to control audio.
- Creating virtual environment
- Ease human computer interaction.

CHAPTER – 2

IMPLEMENTATION

2.1 Fingers and Palm Segmentation

The output of the hand detection is a binary image in which the white pixels are the members of the hand region, while the black pixels belong to the background. An example of the hand detection result is shown in Figure 1. Then, the following procedure is implemented on the binary hand image to segment the fingers and palm.

2.1.1 Palm Point.

The palm point is defined as the centre point of the palm. The block city distance is used to measure the distances between the pixels and the nearest boundary pixels. The found palm point is marked with the point of the green color in Figure 2.



Fig.1

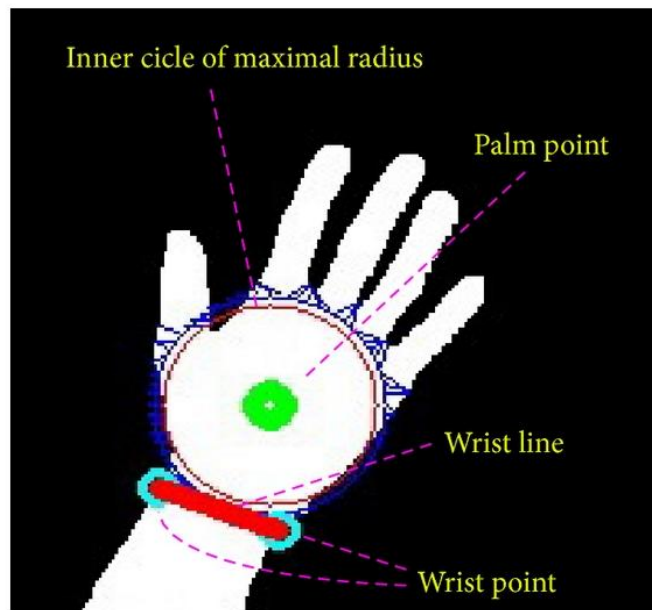


Fig.2

2.1.2 Inner Circle of the Maximal Radius.

When the palm point is found, it can draw a circle with the palm point as the center point inside the palm. The circle is called the inner circle because it is included inside the palm. The radius of the circle gradually increases until it reaches the edge of the palm. That is the radius of the circle stops to increase when the black pixels are included in the circle. The circle is the inner circle of the maximal radius which is drawn as the circle with the red color in Figure2.



Fig.3

2.2. Fingers Recognition

In the segmentation image of fingers, the labelling algorithm is applied to mark the regions of the fingers. In the result of the labelling method, the detected regions in which the number of pixels is too small is regarded as noisy regions and discarded. Only the regions of enough sizes are regarded as fingers and remain. For each remained region, that is, a finger, the minimal bounding box is found to enclose the finger. A minimal bounding box is denoted as a red rectangle in Figure 4. Then, the center of the minimal bounding box is used to represent the center point of the finger.



Fig.4

2.2.1 Thumb Detection and Recognition.

The centers of the fingers are lined to the palm point. Then, the degrees between these lines and the wrist line are computed. If there is a degree smaller than, it means that the thumb appears in the hand image. The corresponding center is the center point of the thumb. The detected thumb is marked with the number 1. If all the degrees are larger than, the thumb does not exist in the image.

2.2.2 Detection and Recognition of Other Fingers.

In order to detect and recognize the other fingers, the palm line is first searched. The palm line parallels to the wrist line. The palm line is searched in the way: start from the row of the wrist line. For each row, a line paralleling to the wrist line crosses the hand. If there is only one connected set of white pixels in the intersection of the line and the hand, the line shifts upward. Once there are more than one connected sets of white pixels in the intersection of the line and the hand, the line is regarded as a candidate of the palm line. In the case of the thumb not detected, the line crossing the hand with more than one connected sets of white pixels in their intersection is chosen as the palm line. In the case of the thumb existing, the line continues to move upward with the edge points of the palm instead of the thumb as the starting point of the line. Now, since the thumb is taken away, there is only one connected set of pixels in the intersection of the line and the hand. Once the connected set of white pixels turns to 2 again, the palm line is found. The search of the palm line is shown in Figure 5.



Fig.5

After the palm line is obtained, it is divided into 4 parts. According to the horizontal coordinate of the center point of a finger, it falls into certain parts. If the finger falls into the first part, it is the forefinger. If the finger belongs to the second part, it is the middle

finger. The third part corresponds to the ring finger. The fourth part is the little finger. The result of finger recognition of Figure 1 is demonstrated in Figure 6. In the figure, the yellow line is the palm line and the red line parallels to the wrist line.



Fig.6

2.3 Computer vision and Digital Image Processing

The sense of sight is arguably the most important of man's five senses. It provides a huge amount of information about the world that is rich in detail and delivered at the speed of light. However, human vision is not without its limitations, both physical and psychological. Through digital imaging technology and computers, man has transcended many visual limitations. He can see into far galaxies, the microscopic world, the sub-atomic world, and even “observe” infra-red, x-ray, ultraviolet and other spectra for medical diagnosis, meteorology, surveillance, and military uses, all with great success. While computers have been central to this success, for the most part man is the sole interpreter of all the digital data. For a long time, the central question has been whether computers can be designed to analyse and acquire information from images autonomously in the same natural way humans can. According to Gonzales and Woods [2], this is the province of computer vision, which is that branch of artificial intelligence that ultimately aims to “use computers to emulate human vision, including learning and being able to make inferences and tak[ing] actions based on visual inputs.” The main difficulty for computer vision as a relatively young discipline is the current lack of a final scientific paradigm or model for human intelligence and human vision itself on which to build a infrastructure for computer or machine learning [3]. The use of images has an obvious drawback. Humans perceive the world in 3D, but current visual sensors like cameras capture the world in 2D images. The result is the natural loss of a good deal of information in the captured images. Without a proper paradigm to explain the mystery of human vision and perception, the recovery of lost information (reconstruction of the world) from 2D images represents a difficult hurdle for machine vision [4]. However, despite

this limitation, computer vision has progressed, riding mainly on the remarkable advancement of decades-old digital image processing techniques, using the science and methods contributed by other disciplines such as optics, neurobiology, psychology, physics, mathematics, electronics, computer science, artificial intelligence and others. Computer vision techniques and digital image processing methods both draw the proverbial water Real-Time Hand Gesture Detection and Recognition Using Simple Heuristic Rules Page 3 of 57 from the same pool, which is the digital image, and therefore necessarily overlap. Image processing takes a digital image and subjects it to processes, such as noise reduction, detail enhancement, or filtering, for the purpose of producing another desired image as the end result. For example, the blurred image of a car registration plate might be enhanced by imaging techniques to produce a clear photo of the same so the police might identify the owner of the car. On the other hand, computer vision takes a digital image and subjects it to the same digital imaging techniques but for the purpose of analysing and understanding what the image depicts. For example, the image of a building can be fed to a computer and thereafter be identified by the computer as a residential house, a stadium, high-rise office tower, shopping mall, or a farm barn. [5] Russell and Norvig [6] identified three broad approaches used in computer vision to distil useful information from the raw data provided by images. The first is the feature extraction approach, which focuses on simple computations applied directly to digital images to measure some useable characteristic, such as size. This relies on generally known image processing algorithms for noise reduction, filtering, object detection, edge detection, texture analysis, computation of optical flow, and segmentation, which techniques are commonly used to pre-process images for subsequent image analysis. This is also considered an “uninformed” approach. The second is the recognition approach, where the focus is on distinguishing and labelling objects based on knowledge of characteristics that sets of similar objects have in common, such as shape or appearance or patterns of elements, sufficient to form classes. Here computer vision uses the techniques of artificial intelligence in knowledge representation to enable a “classifier” to match classes to objects based on the pattern of their features or structural descriptions. A classifier has to “learn” the patterns by being fed a training set of objects and their classes and achieving the goal of minimizing mistakes and maximizing successes through a step-by-step process of improvement. There are many techniques in artificial intelligence that can be used for object or pattern recognition, including statistical pattern recognition, neural nets, genetic algorithms and fuzzy systems. The third is the reconstruction approach, where the focus is on building a geometric model of the world suggested by the image or images and which is used as a basis for action. This corresponds to the stage of image understanding, which represents the highest and most complex level of computer vision processing. Here the emphasis is on enabling the computer vision system to construct internal models based on the data supplied by the images and to discard or update these internal Real-Time Hand Gesture Detection and Recognition Using Simple Heuristic Rules Page 4 of 57 models as they are verified against the real world or some other criteria. If the internal model is consistent with the real world, then image understanding takes place. Thus, image understanding

requires the construction, manipulation and control of models and at the moment relies heavily upon the science and technology of artificial intelligence.

2.4 OpenCV

OpenCV is a widely used tool in computer vision. It is a computer vision library for real-time applications, written in C and C++, which works with the Windows, Linux and Mac platforms. It is freely available as open-source software from <http://sourceforge.net/projects/opencvlibrary/>. OpenCV was started by Gary Bradsky at Intel in 1999 to encourage computer vision research and commercial applications and, side-by-side with these, promote the use of ever faster processors from Intel [7]. OpenCV contains optimised code for a basic computer vision infrastructure so developers do not have to re-invent the proverbial wheel. The reference documentation for OpenCV is at the basic tutorial documentation is provided by Bradsky and Kaehler. According to its website, OpenCV has been downloaded more than two million times and has a user group of more than 40,000 members. This attests to its popularity. A digital image is generally understood as a discrete number of light intensities captured by a device such as a camera and organized into a two-dimensional matrix of picture elements or pixels, each of which may be represented by number and all of which may be stored in a particular file format (such as jpg or gif) [8]. OpenCV goes beyond representing an image as an array of pixels. It represents an image as a data structure called an `IplImage` that makes immediately accessible useful image data or fields, such as:

- width – an integer showing the width of the image in pixels
 - height – an integer showing the height of the image in pixels
 - image Data – a pointer to an array of pixel values
 - channels – an integer showing the number of colors per pixel
 - depth – an integer showing the number of bits per pixel
 - width Step – an integer showing the number of bytes per image row
 - image Size – an integer showing the size of in bytes
- Real-Time Hand Gesture Detection and Recognition Using Simple Heuristic Rules Page 5 of 57
- Roi – a pointer to a structure that defines a region of interest within the image [9]. OpenCV has a module containing basic image processing and computer vision algorithms. These include:
 - smoothing (blurring) functions to reduce noise,
 - dilation and erosion functions for isolation of individual elements,
 - floodfill functions to isolate certain portions of the image for further processing,
 - filter functions, including Sobel, Laplace and Canny for edge detection,
 - Hough transform functions for finding lines and circles,
 - Affine transform functions to stretch, shrink, warp and rotate images,

- Integral image function for summing subregions (computing Haar wavelets),
- Histogram equalization function for uniform distribution of intensity values,
- Contour functions to connect edges into curves,
- Bounding boxes, circles and ellipses,
- Moments functions to compute Hu's moment invariants,
- Optical flow functions (Lucas-Kanade method),
- Motion tracking functions (Kalman filters), and
- Face detection/ Haar classifier.

OpenCV also has an ML (machine learning) module containing well known statistical classifiers and clustering tools. These include:

- Normal/ naïve Bayes classifier,
- Decision trees classifier,
- Boosting group of classifiers,
- Neural networks algorithm, and
- Support vector machine classifier.

2.5 Pattern Recognition and Classifiers

In computer vision a physical object map to a particular segmented region in the image from which object descriptors or features may be derived. A feature is any characteristic of an image, or any region within it, that can be measured. Objects with common features may be grouped into classes where the combination of features may be considered a pattern. Object recognition may be understood to be the assignment of classes to objects based on their respective patterns. The program that does this assignment is called a classifier. The most important step is the design of the formal descriptors because choices have to be made on which characteristics, quantitative or qualitative, would best suit the target object and in turn determines the success of the classifier. In statistical pattern recognition, quantitative descriptions called features are used. The set of features constitutes the pattern vector or feature vector, and the set of all possible patterns for the object form the pattern space X (also known as feature space). Quantitatively, similar objects in each class will be located near each other in the feature space forming clusters, which may ideally be separated from dissimilar objects by lines or curves called discrimination functions. Determining the most suitable discrimination function or discriminant to use is part of classifier design. A statistical classifier accepts n features as inputs and gives 1 output, which is the classification or decision about the class of the object. The relationship between the inputs and the output is a decision rule, which is a function that puts in one space or subset those feature vectors that are associated with a particular output. The decision rule is based on the

particular discrimination function used for separating the subsets from each other. The ability of a classifier to classify objects based on its decision rule may be understood as classifier learning, and the set of the feature vectors (objects) inputs and corresponding outputs of classifications (both positive and negative results) is called the training set. It is expected that a well-designed classifier should get 100% correct answers on its training set. A large training set is generally desirable to optimize the training of the classifier, so that it may be tested on objects it has not encountered before, which constitutes its test set. If the classifier does not perform well on the test set, modifications to the design of the recognition system may be needed.

CHAPTER 3

SOFTWARE AND HARDWARE REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

3.1.1 Visual Studio Code Editor

Introduction and layout

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). VS Code has a powerful command line interface (CLI) which allows you to customize how the editor is launched to support various scenarios. At its heart, Visual Studio Code is a code editor. Like many other code editors, VS Code adopts a common user interface and layout of an explorer on the left, showing all of the files and folders you have access to, and an editor on the right, showing the content of the files you have opened.

Extensions

The features that Visual Studio Code includes out-of-the-box are just the start. VS Code extensions let you add languages, debuggers, and tools to your installation to support your development workflow. VS Code's rich extensibility model lets extension authors plug directly into the VS Code UI and contribute functionality through the same APIs used by VS Code.

Version Control System

Visual Studio Code has integrated source control and includes Git support in-the-box. Many other source control providers are available through extensions on the VS Code Marketplace. VS Code has support for handling multiple Source Control providers simultaneously. For example, you can open multiple Git repositories alongside your TFS local workspace and seamlessly work across your projects. The SOURCE CONTROL PROVIDERS list of the Source Control view (Ctrl+Shift+G) shows the detected providers and repositories and you can scope the display of your changes by selecting a specific provider. VS Code ships with a Git source control manager (SCM) extension. Most of the source control UI and work flows are common across other SCM extensions, so reading about the Git support will help you understand how to use another provider.

3.1.2 LANGUAGES

Python

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

3.2 NETWORK REQUIREMENTS

A stable, high-speed (wired or wireless) Internet connection is required.

3.3 HARDWARE REQUIREMENTS

- CPU: x86 64-bit CPU(Intel / AMD Architecture)
- RAM: 4GB
- Minimum space: 5 GB free disk space

CHAPTER – 4

CONCLUSION AND FUTURE WORKS

4.1 CONCLUSION

- Developed a system for “Virtual Drag and Drop control using Hand Tracking Module in Machine Learning”The project helps the users to control volume by using hand gesture which can be used in many situations and in any place.
- It can be a step for time and energy saving for people and next move towards virtual communication.
- Anyone can use this facility and the system can run in background as well.
- The project is completed with the objective of positive intent only.

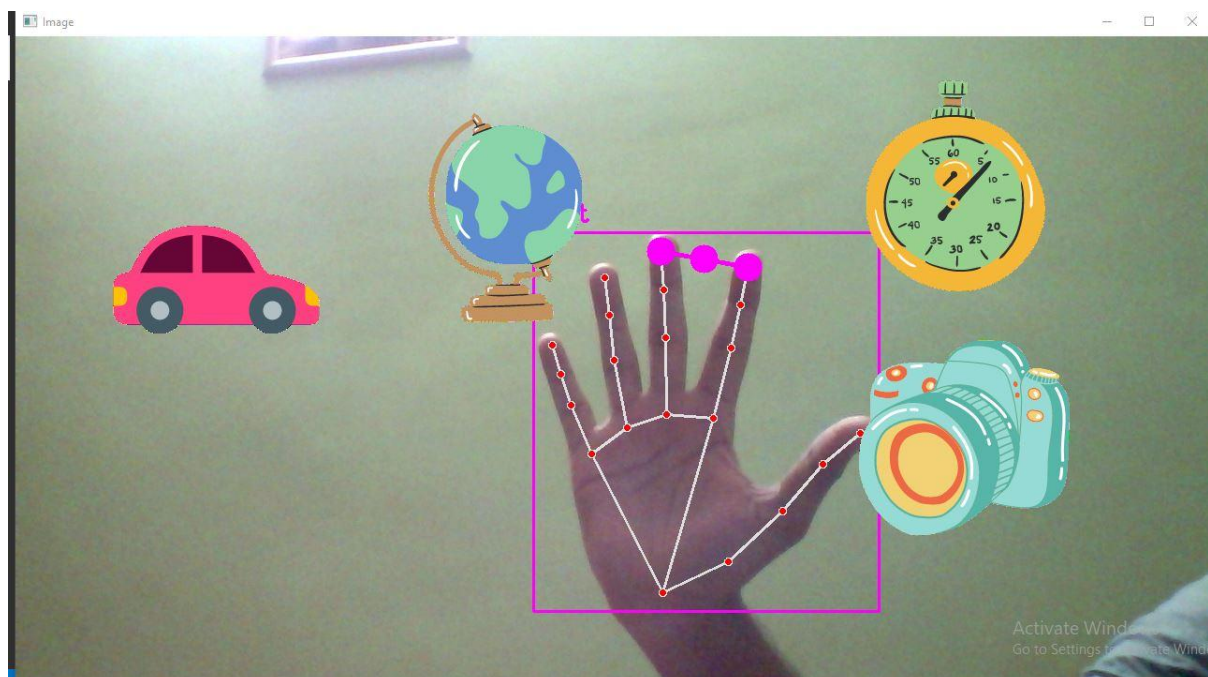
4.2 FUTURE WORKS

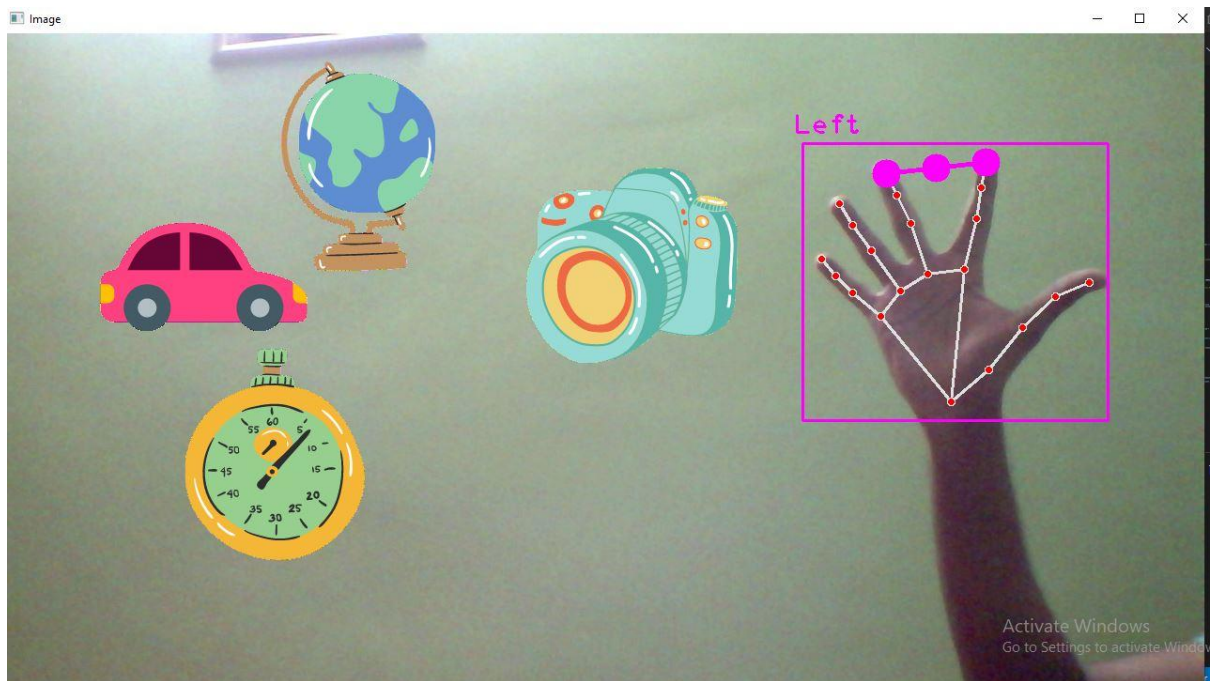
The future work of the project goes by the aim of successfully attaining a complete virtual environment with no hardware devices connected for any operation to be done. Complete gesture control for mouse, keyboard and other controls of a computer without any physical interaction.

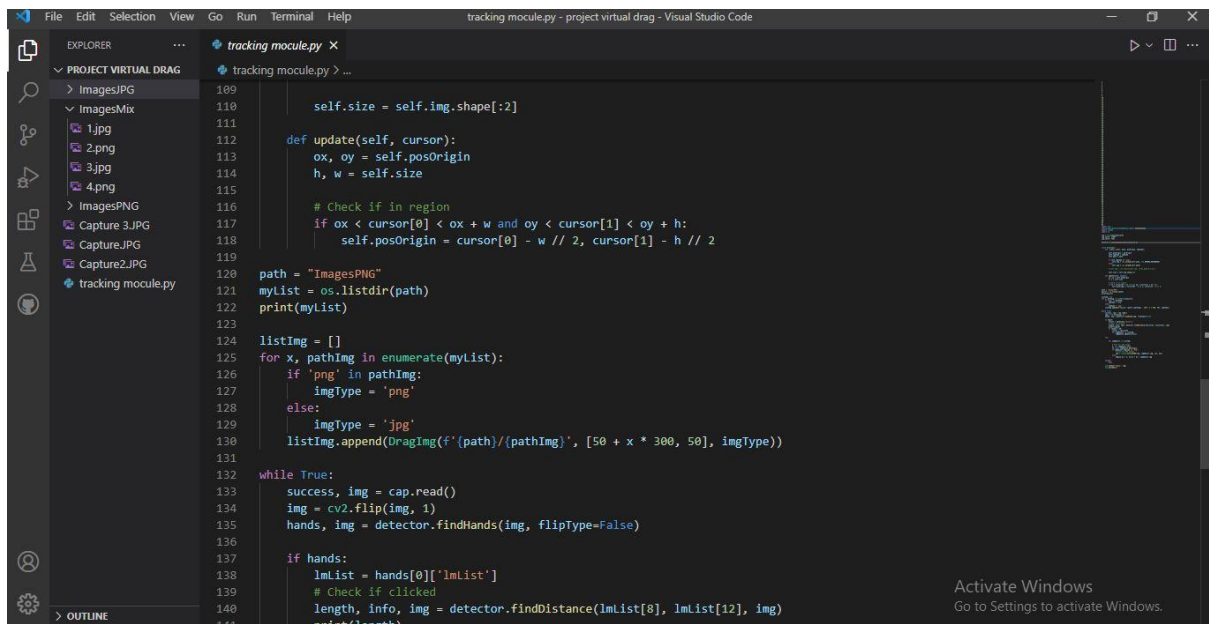
CHAPTER – 5

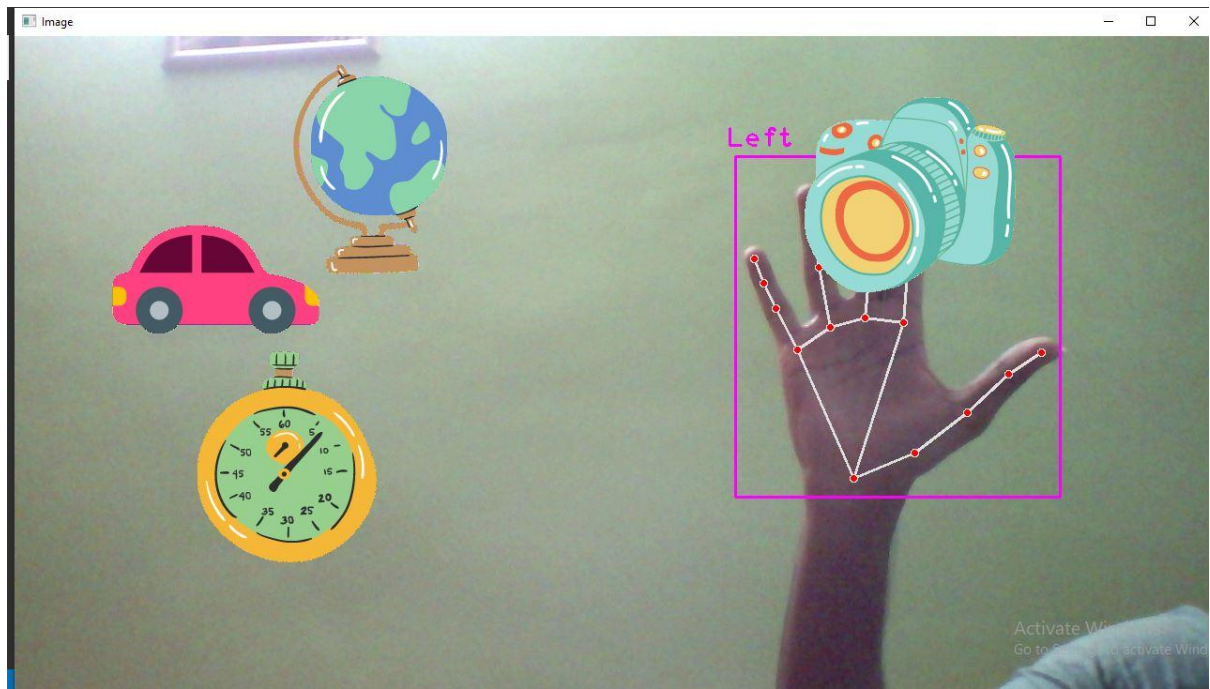
APPENDIX (RESULTS AND DISCUSSION)

Result snapshots:









```

import cv2
from cvzone.HandTrackingModule import HandDetector
import cvzone
import os

cap = cv2.VideoCapture(0)
cap.set(3, 1280)
cap.set(4, 720)

detector = HandDetector(detectionCon=0.8)

class DragImg():
    def __init__(self, path, posOrigin, imgType):

        self.posOrigin = posOrigin
        self.imgType = imgType
        self.path = path

        if self.imgType == 'png':
            self.img = cv2.imread(self.path, cv2.IMREAD_UNCHANGED)
        else:
            self.img = cv2.imread(self.path)

        # self.img = cv2.resize(self.img, (0,0), None, 0.4, 0.4)

        self.size = self.img.shape[:2]

    def update(self, cursor):
        ox, oy = self.posOrigin
        h, w = self.size

        # Check if in region
        if ox < cursor[0] < ox + w and oy < cursor[1] < oy + h:
            self.posOrigin = cursor[0] - w // 2, cursor[1] - h // 2

path = "ImagesPNG"
myList = os.listdir(path)
print(myList)

listImg = []
for x, pathImg in enumerate(myList):
    if 'png' in pathImg:
        imgType = 'png'
    else:
        imgType = 'jpg'
    listImg.append(DragImg(f'{path}/{pathImg}', [50 + x * 300, 50], imgType))

```

```

while True:
    success, img = cap.read()
    img = cv2.flip(img, 1)
    hands, img = detector.findHands(img, flipType=False)

    if hands:
        lmList = hands[0]['lmList']
        # Check if clicked
        length, info, img = detector.findDistance(lmList[8], lmList[12], img)
        print(length)
        if length < 60:
            cursor = lmList[8]
            for imgObject in listImg:
                imgObject.update(cursor)

    try:

        for imgObject in listImg:

            # Draw for JPG image
            h, w = imgObject.size
            ox, oy = imgObject.posOrigin
            if imgObject.imgType == "png":
                # Draw for PNG Images
                img = cvzone.overlayPNG(img, imgObject.img, [ox, oy])
            else:
                img[oy:oy + h, ox:ox + w] = imgObject.img

    except:
        pass

    cv2.imshow("Image", img)
    cv2.waitKey(1)

```

CHAPTER – 6

REFERENCES

1. Zhigang, F. Computer gesture input and its application in human computer interaction. *Mini Micro Syst.* 1999, 6, 418–421.
2. Mitra, S.; Acharya, T. Gesture recognition: A survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* 2007, 37, 311–324. [CrossRef]
3. Ahuja, M.K.; Singh, A. Static vision based Hand Gesture recognition using principal component analysis. In *Proceedings of the 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, Amritsar, India, 1–2 October 2015; pp. 402–406.
4. Kramer, R.K.; Majidi, C.; Sahai, R.; Wood, R.J. Soft curvature sensors for joint angle proprioception. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, 25–30 September 2011; pp. 1919–1926.
5. Jespersen, E.; Neuman, M.R. A thin film strain gauge angular displacement sensor for measuring finger joint angles. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, New Orleans, LA, USA, 4–7 November 1988; pp. 807–vol.
6. Fujiwara, E.; dos Santos, M.F.M.; Suzuki, C.K. Flexible optical fiber bending transducer for application in glove-based sensors. *IEEE Sens. J.* 2014, 14, 3631–3636. [CrossRef]
7. Shrote, S.B.; Deshpande, M.; Deshmukh, P.; Mathapati, S. Assistive Translator for Deaf & Dumb People. *Int. J. Electron. Commun. Comput. Eng.* 2014, 5, 86–89.
8. Gupta, H.P.; Chudgar, H.S.; Mukherjee, S.; Dutta, T.; Sharma, K. A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. *IEEE Sens. J.* 2016, 16, 6425–6432. [CrossRef]
9. Lamberti, L.; Camastra, F. Real-time hand gesture recognition using a color glove. In *Proceedings of the International Conference on Image Analysis and Processing*, Ravenna, Italy, 14–16 September 2011; pp. 365–373.

