

# **Hand-Write Virtual Board**

Project report submitted in partial fulfillment of the requirement for the degree of  
Bachelor of Technology

in

**Computer Science and Engineering/Information Technology**

By

(HARSHIT YADAV (181469))  
& (PRATEEK SHARMA (181482))

Under the supervision of

(DR. PARDEEP KUMAR)

to



**Jaypee University of Information Technology Waknaghat, Solan-173234,  
Himachal Pradesh**

**Certificate**

**Candidate's Declaration**

I hereby declare that the work presented in this report entitled "**Hand-Write Virtual board.**" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2022 to May 2022 under the supervision of **(DR. PARDEEP KUMAR)**(Associate Professor in the Department of Computer Science & Engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.



(Student Signature)  
HARSHIT YADAV, 181469



PRATEEK SHARMA, 181482

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)  
Supervisor Name: Dr. Pardeep Kumar  
Designation: Associate Professor  
Department name: Computer Science & Engineering

## Acknowledgement

I, right off the bat, express my heartiest thanks and thankfulness to all-powerful God for His heavenly gift makes it feasible for us to finish the task work effectively.

I'm truly thankful and wish my significant obligation to Boss Dr Pardeep Kumar, Partner Teacher (SG), Branch of CSE Jaypee College of Data Innovation, Wagnaghat. His interminable persistence, academic direction, consistent consolation, steady and enthusiastic oversight, productive analysis, important exhortation, perusing numerous substandard drafts and adjusting them at all stages have made it conceivable to finish this task.

I should offer my heartiest thanks to Dr Pardeep Kumar, Part of CSE, for his generous help to finish my endeavor. I would moreover generously welcome each and every one of those individuals who have helped me straightforwardly or in a roundabout way in making this adventure a triumph. In this outstanding situation, I ought to thank the different staff individuals, both educating and non-preparing, which have cultivated their profitable help and worked with my undertaking. Finally, I ought to perceive with due respect the reliable assistance and patients of my people.

Harshit Yadav(181469)

Prateek Sharma(181482)

## Table of Content

S. NO.	NAME	PAGE NO.
1	INTRODUCTION	1
2	LITERATURE SURVEY	6
3	SYSTEM DEVELOPMENT <ul style="list-style-type: none"><li>● Requirements/Analysis/Design</li><li>● Product Functions</li><li>● Limitations</li><li>● Hardware Requirements</li><li>● Software Requirements</li><li>● Design and Planning</li><li>● Palm Detection Model</li><li>● Hand Landmark Model</li></ul>	8
4	PERFORMANCE ANALYSIS <ul style="list-style-type: none"><li>● Image Capturing</li><li>● Image Processing</li><li>● Gesture Recognition</li><li>● Using OpenCV</li></ul>	14

- |  |   |  |
|--|---|--|
|  | <ul style="list-style-type: none"><li>● Starting the app</li><li>● System Modules and Components</li><li>● Algorithm</li><li>● Glimpse of application</li></ul> |  |
|--|---|--|

## Abstract

In this project the gesture-based computer control system coupled with the dedicated touchless interactive virtual whiteboard is presented. The idea behind this project was to make online education easy and interactive. As we know that the last few years we not good in all respects and specifically for study and to cope up with problem online studies was the best reply to that. Online classes Google meet and other online platform were used for studies but to make these class more interactive we have made this project in which the teacher or the user of handy can draw and erase the text on the screen when the person is presenting on the screen and they can also record the class and the recording will be saved their pc that can be shared to the student late for their use. The framework designed empowers a client to control any top-most PC application by utilizing one hand signals. Initially, a survey of signal acknowledgment applications with an attention on strategies and calculations applied is given. Equipment and programming arrangement of the framework comprising a PC, camera, and screen. Picture handling techniques utilized and the fluffy rule-based signal acknowledgment idea are portrayed. The utilization of various python libraries, for example, Numpy for computation, OpenCV for picture catch and association with the equipment and media pipe for the acknowledgment of gestures of our hand that can be used to draw or highlight or erase and record anything on the screen. The application called Hand, enabling to draw shapes on a virtual whiteboard developed for use with the system is then

presented. The functionality of the virtual Interactive Whiteboard and the way of managing the contents with hand gestures is shown.

# **Chapter-1: Introduction**

## **1.1 Introduction**

This year has been a shocker in more ways than one. While the pandemic has left the world staggering from its effect, it has additionally left us at the cusp of an advanced upheaval. Right from our everyday lives to how organizations are run, innovation has taken the middle stage. The effect of this quick and unexpected shift is noticeable in pretty much every industry and the instruction area is no exemption.

The manner in which schooling is being granted and consumed has gone through an intense change since the beginning of the pandemic. With instructive foundations briefly shut attributable to somewhere safe worries, the idea of actual homeroom based learning is rapidly turning into a relic of days gone by.

Students and educators have now moved to virtual homerooms and web based learning has gotten forward momentum as of late with the pandemic further speeding up its development.

Due to its various applications in augmented experience, communication through signing acknowledgment, and PC games, hand motion acknowledgment is a critical report point in the field of human-PC collaboration. In spite of past endeavors, fostering a dependable hand motion recognition framework that can



be utilized in true applications stays a troublesome assignment. The nature of the information picture from optical cameras limits existing vision-based methods.

Likewise, these systems have failed to convey sufficient hand movement affirmation results. Hand area and sign affirmation are two inconvenient difficulties nearby movement affirmation. Hand signal execution has critical accommodation issues, for instance, quick reaction time, high distinctive verification precision, speed of learning, and client fulfillment, which could figure out why very few vision-based movement systems have progressed past their model stage.

A few scholastics have proposed utilizing a wearable camcorder to distinguish characters written in the air as an info component for penmanship in the air. Their review gives a letter input by utilizing a camcorder to catch the administrator's hand activity and doing picture examination on a PC. The hand-mounted camcorder, then again, is viewed as consistently worked by the framework. Various cameras have additionally been proposed for recognizing an individual's outline and removing it, as well as doing direction location, character distinguishing proof, and fingertip discovery. This strategy might be carried out by consolidating numerous cameras to gain a high appreciation rate (96.6%) and distinguish penmanship in the air.

## **1.2 Problem Statement**

The studies led during lockdown showed that while kids face issues in getting to instruction carefully, instructors face issues in conveying training through advanced mediums. A faltering 84% of educators announced confronting difficulties in conveying training carefully with near a portion of the instructors confronting issues connected with the web both sign issues and information expenses. Two out of each and every five educators come up short on vital gadgets to carefully convey schooling.

Such cases with respect to the absence of vital gadgets and their costs can be settled by utilizing Hand-Compose Virtual Board.

## **1.3 Objective of the Product**

The particular targets of the venture include:

- Auditing the current/current computerized training conveying philosophies,
- Thinking of a savvy and simple to utilize application;
- Concentrating on the previous models which could be utilized;
- Executing our thought utilizing the previous models;

## **1.4 Feasibility Study**

An attainability study is a significant level container rendition of the whole Framework investigation and Configuration Interaction. The review starts by grouping the issue definition. Possibility is to decide whether it merits doing. When an acknowledgment issue definition has been produced, the investigator fosters an intelligent model of the framework. A quest for choices is dissected cautiously. There are 3 sections in possibility study:

- Operational feasibility
- Technical feasibility
- Economical feasibility

### **1.4.1 Operational feasibility**

Functional plausibility is the proportion of how well a proposed framework tackles the issues, and makes the most of the amazing open doors distinguished during extension definition and how it fulfills the prerequisites recognized in the necessities examination period of framework advancement. The functional possibility appraisal centers around how much the proposed advancement projects finds a place with the current business climate targets as to improvement plan, conveyance date, corporate culture and existing business processes. To ensure success, desired functional results should be conferred

during plan and improvement. These incorporate such plan subordinate boundaries as unwavering quality, viability, acceptability, convenience, producibility, superfluity, manageability, moderateness and others. These boundaries are expected to be considered at the beginning phases of plan on the off chance that ideal functional ways of behaving are to be understood. A framework plan and advancement requires suitable and convenient utilization of designing and the board endeavors to meet the recently referenced boundaries. A framework might fill its expected need most really when its specialized and working qualities are designed into the plan. In this way, functional plausibility is a basic part of frameworks designing that should be a vital piece of the early plan stages.

### **1.4.2 Technical Feasibility**

This includes questions, for example, whether the innovation required for the framework exists, how troublesome it will be to assemble, and whether the firm has sufficient experience utilizing that innovation. The appraisal depends on frame plan of framework prerequisites concerning input, processes, yield, fields, projects and methodology. This can be qualified as far as volume of information, patterns, recurrence of refreshing to give a prologue to the specialized framework. The application is the way that it has been created on Windows XP stage and a high design of 1GB Slam on Intel Pentium Double center processor. This is actually plausible. The specialized possibility appraisal is centered around acquiring a comprehension of the current specialized assets of the association and their appropriateness to the normal requirements of the proposed

framework. It is an assessment of the equipment and programming and how it addresses the issues of the proposed framework.

### **1.4.3 Economical Feasibility**

Laying out the expense viability of the proposed framework i.e., in the event that the advantages don't offset the expenses, then, at that point, it does not merit going on. In the quick moving present reality there is an extraordinary requirement for online long range informal communication offices. Subsequently, the advantages of this venture in the ongoing situation make it monetarily plausible. The motivation behind the financial practicality appraisal is to decide the positive monetary advantages to the association that the proposed framework will give. It incorporates measurement and ID of the relative multitude of advantages anticipated. This appraisal regularly includes an expense/benefits investigation.

## **Chapter2: Literature Survey**

The capacity to see the shape and movement of hands can be a fundamental part in further developing the client experience across an assortment of mechanical areas and stages. For instance, it can shape the reason for communication through signing understanding and hand motion control, and can likewise empower the overlay of advanced content and data on top of the actual world in increased reality. While falling into place without a hitch for individuals, powerful constant hand insight is a distinctly difficult PC vision task, as hands frequently block themselves or one another (for example finger/palm impediments and handshakes) and need high difference designs.

Media Line Hands is a high-devotion hand and finger following arrangement. It utilizes AI (ML) to deduce 21 3D milestones of a hand from only a solitary edge. While present status of-the-workmanship approaches depend fundamentally on strong work area conditions for induction, this strategy accomplishes continuous execution on a cell phone, and even scales to various hands. We trust that giving this hand insight usefulness to the more extensive innovative work local area will bring about a development of inventive use cases, invigorating new applications and new exploration roads.

Media Line Hands uses a ML pipeline comprising of different models cooperating: A palm recognition model that works on the full picture and returns a situated hand bounding box. A hand milestone model that works on the trimmed picture locale characterized by the palm locator and returns high-constancy 3D hand keypoints.

Giving the precisely edited hand picture to the hand milestone model radically lessens the requirement for information expansion (for example pivots, interpretation and scale) and on second thought permits the organization to commit the greater part of its ability towards coordinate expectation exactness. Furthermore, in our pipeline the harvests can likewise be created in light of the hand milestones recognized in the past casing, and just when the milestone model could never again distinguish hand presence is palm discovery conjured to relocalize the hand.

### **Chapter-3: System Development**

## **3. Software Requirements Specification**

### **3.1 Purpose**

With the development of worldwide computing, current user interaction approaches with keyboard, mouse and pen are not sufficient. Due to the limitation of these devices the usable command set is also limited. Direct use of hands can be used as an input device for providing natural interaction.

Hand-Write Virtual board is a hand gesture recognition based whiteboard that allows you to write using a webcam. It uses various hand gestures to draw, erase and navigate around the app. It has features like different colored pens, eraser and a recording option that records your session and saves it in a folder. This virtual board is cost effective and also easy to use.

### **3.2 Product Perspective**

During the digitisation of education where on one hand keyboard and mouse are not sufficient for user interactions, stylus and pads are not affordable for everyone. Our product tries to satisfy this need by creating a virtual whiteboard which is cost effective and easy to learn and use. This service is usable from anywhere.

### **3.3 Product Functions**

- Record – Records the video while the user uses the whiteboard and saves the file.
- Draw/Write – Allows users to draw/write on the screen using hand gestures.



- Erase – Allows users to erase the previous drawings/writings using hand gestures.
- Change colors: Change pen colors using hand gestures.

### **3.4 Hardware Requirements**

- PC WITH 250 GB OR MORE HARD DISK
- AT LEAST 2GB RAM
- PC WITH PENTIUM 1 AND ABOVE

### **3.5 Software Requirements**

- Operating System - Windows XP
- Language - Python
- Libraries - OpenCV
- IDE - Visual Code

### **3.6 Design & Planning**

Hand gesture recognition system can be divided into following modules:

- Preprocessing
- Feature extraction of the processed image
- Real time classification

In the same way as other example acknowledgment assignments, pre-handling is fundamental for improving power and acknowledgment exactness. The preprocessing readies the picture arrangement for the acknowledgment, so prior to working out the corner to corner total and different calculations, a pre-handling step is performed to get the fitting picture, which is expected for continuous characterization. So it comprises of certain means. The net impact of this handling is to remove the hand just from the given information on the

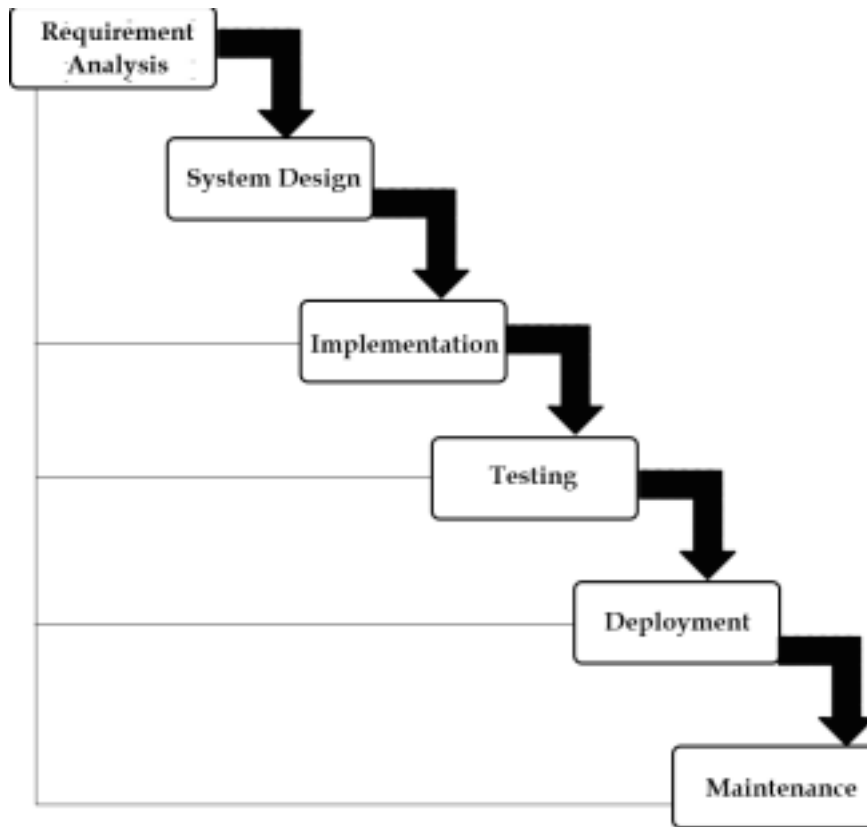
grounds that once the hand is distinguished from the given information it tends to be perceived without any problem.

## **3.8 MODELS**

### **3.8.1 WATERFALL MODEL**

The waterfall model was selected as the SDLC model due to the following reasons:

- Requirements were very well documented, clear and fixed.
- Technology was adequately understood.
- Simple and easy to understand and use.
- There were no ambiguous requirements.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and are view process.
- Clearly defined stages.
- Well understood milestones. Easy to arrange tasks.



### **3.8.2 PALM DETECTION MODEL**

To recognize beginning hand areas, a solitary shot indicator model streamlined for portable continuous purposes the Media pipelite model and full model to work across an assortment of hand sizes with a huge scope range (~20x) comparative with the picture outline and have the option to identify blocked and self-impeded hands. While faces have high differentiation designs, e.g., in the eye and mouth locale, the absence of such highlights in hands makes it relatively hard to recognize them dependably from their visual elements alone. All things being equal, giving extra setting, similar to arm, body, or individual elements, helps precise hand restriction.

Our technique tends to the above challenges utilizing various methodologies. In the first place, we train a palm indicator rather than a hand locator, since assessing bouncing boxes of inflexible articles like palms and clenched hands is essentially easier than identifying hands with enunciated fingers. Moreover, as palms are more modest items, the non-most extreme concealment calculation functions admirably in any event, for two-hand self-impediment cases, similar to handshakes. Also, palms can be demonstrated utilizing square bouncing boxes (secures in ML wording) overlooking other perspective proportions, and along these lines decreasing the quantity of anchors by an element of 3-5.

Second, an encoder-decoder include extractor is utilized for greater scene setting mindfulness in any event, for little articles (like the retina net methodology). In conclusion, we limit the central misfortune during preparing to help a lot of anchors coming about because of the great scope change.

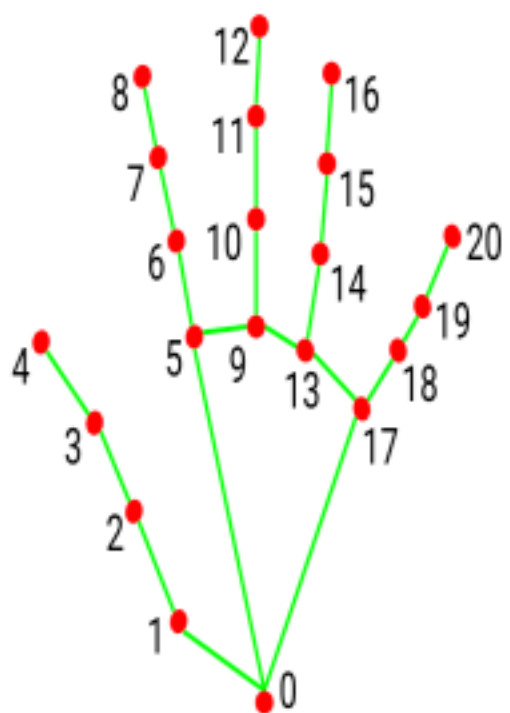
With the above strategies, we accomplish a typical accuracy of 95.7% in palm location. Utilizing an ordinary cross entropy misfortune and no decoder gives a benchmark of simply 86.22%.

### **3.8.2 HAND LANDMARK MODEL**

After the palm discovery over the entire picture our ensuing hand milestone model performs exact central issue restriction of 21 3D hand-knuckle facilitates inside the recognized hand locales through relapse, that is immediate direction forecast. The model learns a reliable inner hand present portrayal and is powerful even to somewhat noticeable hands and self-impediments.

To acquire ground truth information, we have physically commented on ~30K

certifiable pictures with 21 3D directions, as displayed beneath (we take Z-esteem from picture profundity map, assuming that it exists per relating coordinate). To more readily cover the conceivable hand presents and give extra oversight on the idea of hand calculation, we additionally render an excellent manufactured hand model over different foundations and guide it to the relating 3D directions.



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

## **Chapter-4: Performance Analysis**

In this section we will do analysis of technologies to use for implementing the project.

### **4.1 Image capturing using OpenCV**

OpenCV is a Python open-source library, which is utilized for PC vision in Man-made brainpower, AI, face acknowledgment, etc. In OpenCV, the CV is a truncation type of a PC vision, which is characterized as a field of study that assists PCs with understanding the substance of the advanced pictures like photos and recordings.

The motivation behind PC vision is to figure out the substance of the pictures. It separates the portrayal from the photos, which might be an article, a text depiction, a three-layered model, etc. For instance, vehicles can be worked with PC vision, which will actually want to distinguish various articles around the street, for example, traffic signals, walkers, traffic signs, etc, and act likewise.

We use OpenCV to capture hand gestures through the webcam frame by frame and further process them accordingly to draw or erase on the canvas.

### **4.2 Image processing using OpenCV**

Whenever we join every one of the focuses on the limit of an article, we get a form.

Commonly, a particular form alludes to limit pixels that have a similar variety and force. OpenCV makes it truly simple to find and attract shapes pictures. Switching the picture over completely to grayscale is vital as it readies the picture for the subsequent stage. Switching the picture over completely to a solitary channel grayscale picture is significant for thresholding, which thus is fundamental for the shape recognition calculation to work properly. While finding forms, first generally apply parallel thresholding or vigilant edge location to the grayscale picture. This converts the picture to highly contrasting, featuring the objects important to make things simple for the shape recognition calculation. Thresholding turns the line of the item in the picture totally white, with all pixels having a similar force. The calculation can now distinguish the boundaries of the articles from these white pixels.

Assuming we utilize single channels like R (red), G (green), or B (blue) rather than grayscale (thresholded) pictures, the form location calculation won't function admirably. As we examined already, the calculation searches for borders, and comparative force pixels to distinguish the shapes. A paired picture gives this data far superior to a solitary (RGB) variety channel picture.



### **4.3 Gesture Recognition using Mediapipe**

Media Line Hands uses a ML pipeline consisting of different models cooperating: A palm recognition model that works on the full picture and returns an arranged hand jumping box. A hand milestone model that works on the edited picture district characterized by the palm finder and returns high-devotion 3D hand keypoints.

Giving the precisely edited hand picture to the hand milestone model radically decreases the requirement for information increase (for example pivots, interpretation and scale) and on second thought permits the organization to commit the majority of its ability towards coordinate expectation exactness. Likewise, in our pipeline the yields can likewise be created in view of the hand milestones recognized in the past casing, and just when the milestone model could never again distinguish hand presence is palm location conjured to relocalize the hand.



### **4.4 Draw using OpenCV**



Now we start tracking coordinates of each and every point the center of the contour touches on the screen, along with its color. We store these sets of points of different colors in different deques. When the center of the contour touches one of the coloured boxes we store the points in its respective color deque.

So far we stored all the points in their respective color deques. Now we just join them using a line of their own color.

Once we join all the points in each and every frame with a line and put it on the window, it all fits perfectly to work like a paint application. After stopping to read data from the webcam, we release the camera and destroy all the windows.

## **USE OF VERSION CONTROL**

If you are a part of a software development life cycle as a developer, project management must have been a time where everybody working on the same project needs a central system to access, monitor, update, record or report progress of work; this is where version control comes in really handy.

There are a couple of project management tools available today, but we are opting for the free and popular gitHub. If we are very conversant and open source technology like gitHub we must have a project tab on a github repository. It has features consisting of pull requests, merge, fork etc.

If you are a graphic or web designer and want to keep every version of an image or layout (which you would most certainly want to), a Version Control System (VCS) is a very wise thing to use. It allows you to revert selected files back to a previous state, revert the entire project back to a previous state, compare changes

over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more. Using a VCS also generally means that if you screw things up or lose files, you can easily recover. In addition, you get all this for very little overhead.

## **4.5 Implementation of the system**

This gives an overview of the implementation and explains how users can navigate through the newly developed tool in order to use it easily.

### **4.5.1 Starting the app**

The app is started by running the Hand\_write.py file which opens up a new window which will act as a canvas for drawing and has multiple tabs for recording and different colored pens.

### **4.5.2 Recognized gestures**

- **Index finger up:** Lets the user draw on the canvas using index finger.
- **Index and middle finger up:** Lets users freely navigate throughout the canvas.
- **Palm:** Lets users erase already drawn things on the canvas.
- **Pinching (Using second hand):** Increases or decreases the pen or eraser size.

### **4.5.3 Recording videos using Handy**

There is a record button on top left of the canvas, while using the index and middle fingers

up user can freely navigate to the button and activate recording option which starts the recording and once the user wants to end recording they can stop and the recording will be stored in the recordings folder.

#### **4.5.4 Switching color of pens**

There are four different color options available on the top of the screen which are accessible using the free movement gesture just by placing the erect index and middle finger on them.

### **4.6 SYSTEM MODULES AND COMPONENTS**

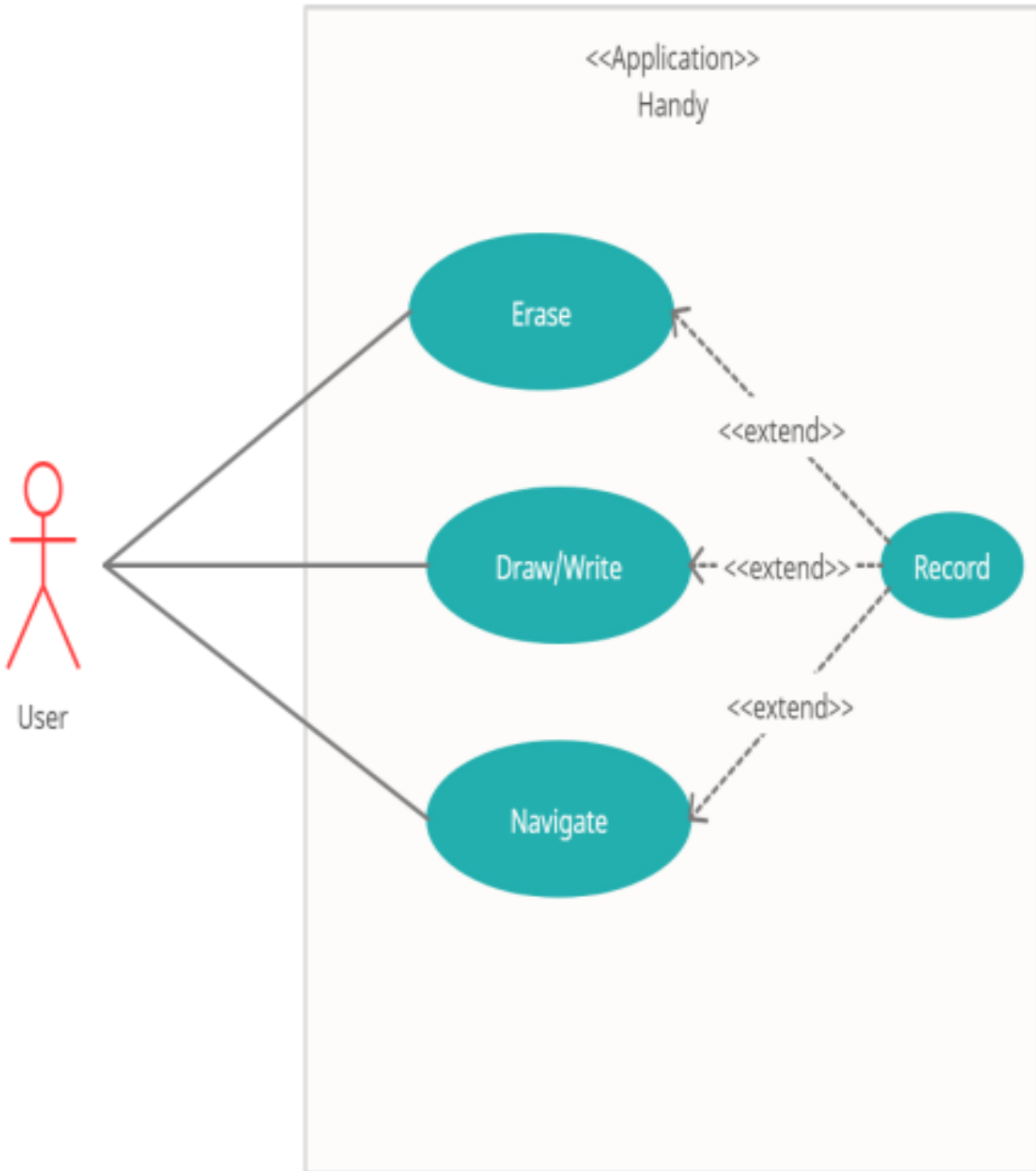
The system was implemented as a web-based virtual whiteboard that is used to make online teaching easy for the teacher and student. The language we use for development of this whiteboard is python and the libraries used are OpenCV, Media Pipe and Numpy. The environment in which we have developed this tool is visual studio. OpenCV is the library of python that is used to capture images of videos from the hardware of your device. Media Pipe is used for the gesture reorganization of our hand.

#### **4.6.1 Shortcomings with the system**

**The system implemented is hindered by following factors:**

- The app requires a camera so the user can't use their camera while on a google meet or zoom meeting while also using the app.
- If the user wants to use both the camera in the meeting and with the app then they need to attach an extra webcam.
- Getting used to writing with fingers requires a bit of practice.
- If the system on which the app is being used has less memory, then the functionalities might lag.

CASE DIAGRAM:



A utilization case chart is the essential type of framework/programming prerequisites for another product program immature. Use cases indicate the normal way of behaving (what),

and not the specific strategy for getting it going (how). Use cases once indicated can

be indicated both literary and visual portrayal (i.e., use case outline). A vital idea of purpose case demonstrating is that it assists us with planning a framework from the end client's viewpoint. It is a powerful method for imparting framework conduct in the client's terms by indicating all remotely noticeable framework conduct.

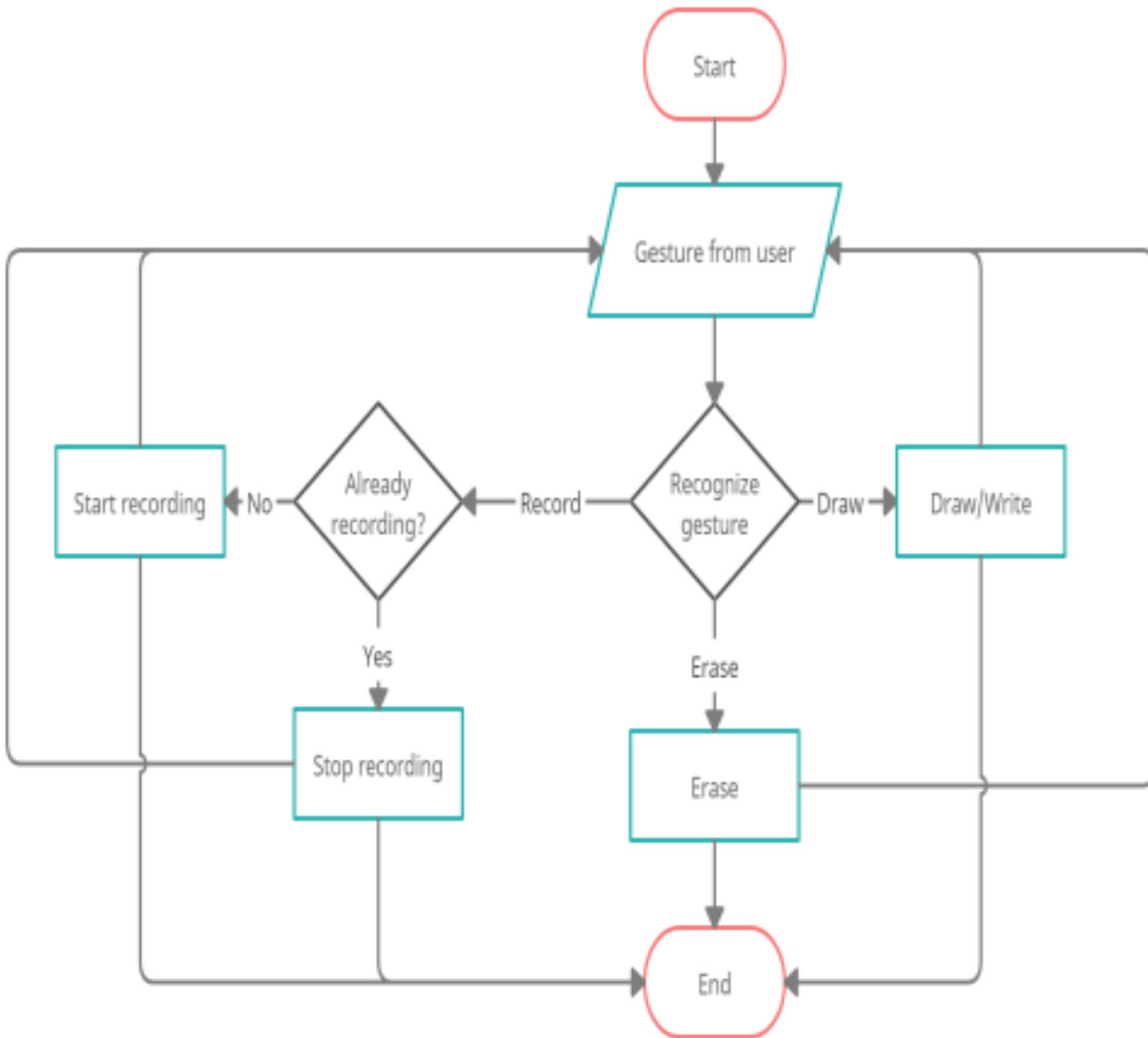
A utilization case chart is generally straightforward. It doesn't show the detail of the utilization cases:

- It just sums up a portion of the connections between use cases, entertainers, and frameworks.
- It doesn't show the request in which steps are performed to accomplish the objectives of each utilization case.

The User is the one and only actor here:

- Users can record their session/class.
- Write/draw on the canvas.
- Erase already drawn/written things on canvas.
- Navigate freely throughout the canvas.
- Change the color of pens.

## ACTIVITY DIAGRAM



Movement Graphs portray how exercises are composed to offer a support which can be at various degrees of reflection. Commonly, an occasion should be accomplished by certain tasks, especially where the activity is planned to accomplish various things that require coordination, or how the occasions in a solitary use case connect with each other, specifically, use situations where exercises might cover and require coordination.

- First, the user arrives.
- Webcam detects gestures and starts capturing frames.
- Individual frames are then processed and checked against the gestures to see if they are drawing/erasing/navigation.
- The task according to gestures is performed.
- The process of detecting, capturing and identifying gestures is repeated until the user exits the app.

### **Algorithm for hand\_write.py:**

```
import cv2
import numpy as np
import os
from datetime import datetime
from hand_detector_lib import hand_detector
sw,sh = 800,600
```



```
def main():  
  
    header_images=[]  
  
    for i in os.listdir('./assets'):  
        _img = cv2.imread(os.path.join("assets",i))  
        _img = cv2.resize(_img,(sw,int(sw*( _img.shape[0]/ _img.shape[1]))))  
  
        header_images.append( _img)  
  
    hh,hw,_ = header_images[0].shape  
  
    header = 0  
  
  
    cam = cv2.VideoCapture(0)  
    detector = hand_detector()  
  
  
    drawing_canvas = np.zeros((sh,sw,3),np.uint8)  
    color = (0,0,255)  
    thickness = 5  
    eraser_thickness = 80  
    x_prev,y_prev=0,0  
    record=False  
    out=None  
    while True:  
        status,frame = cam.read()  
        frame=cv2.resize(frame,(sw,sh))  
        frame = cv2.flip(frame,1)  
        frame = detector.find_hands(frame,draw=False)  
        lmList = detector.find_position(frame,draw=False)
```

**if lmList:**

**x1,y1 = lmList[8][1:]**

**x2,y2 = lmList[12][1:]**

**fingers = detector.fingers\_up()**

**if fingers[1] and fingers[2]==False:**

**cv2.circle(frame,(x1,y1),8,color,-1)**

**if x\_prev == 0 and y\_prev==0:**

**x\_prev,y\_prev = x1,y1**

**if color==(0,0,0):**

**et\_h = eraser\_thickness//2**

**cv2.rectangle(frame,(x1-et\_h ,y1- et\_h ),(x1+ et\_h ,y1+ et\_h ),(0,0,0),-1)**

**cv2.line(drawing\_canvas,(x\_prev,y\_prev),(x1,y1),color,eraser\_thickness)**

**else:**

**cv2.line(drawing\_canvas,(x\_prev,y\_prev),(x1,y1),color,thickness)**

**x\_prev,y\_prev = x1,y1**

**elif fingers[1] and fingers[2]:**

**x\_prev,y\_prev = 0,0**

**#cv2.rectangle(frame,(x1,y1-25),(x2,y2+25),(0,0,0),-1)**

**if y1<hh:**

**if 120<=x1<=240:**

**color = (0,0,255)**

**header = 0**

**elif 260<=x1<=380:**

**color = (255,0,0)**

**header = 1**

**elif 410<=x1<=530:**

**color = (0,255,0)**

**header = 2**

**elif 540<=x1<=660:**

**color = (156,0,210)**

**header = 3**

**elif x1>=670:**

**color = (0,0,0)**

**header = 4**

**elif x1<=100:**

**record = True**

**if not record:**

**out = None**

**elif record and not out:**

```

type = cv2.VideoWriter_fourcc(*'XVID' )
name = str(datetime.now()).split()
name= f"{name[0]}_{name[1][:name[1].index('.')]}_mp4"
name=name.replace("-","_")
name=name.replace(":","_")
out = cv2.VideoWriter(os.path.join("recordings",name),type , 30 , (sw,sh) )

```

```

img_gray = cv2.cvtColor(drawing_canvas,cv2.COLOR_BGR2GRAY)
_,img_inv = cv2.threshold(img_gray,10,255,cv2.THRESH_BINARY_INV)

```

```

img_inv = cv2.cvtColor(img_inv,cv2.COLOR_GRAY2BGR)

```

```

frame = cv2.bitwise_and(frame,img_inv)
frame = cv2.bitwise_or(frame,drawing_canvas)

```

```

frame[:hh,:hw]=header_images[header]

```

**if record and out:**

```

cv2.rectangle(frame,(0,0),(100,hh),(41,156,0),3)
out.write(frame)

```

```

cv2.imshow("Frame",frame)

```

```

if cv2.waitKey(1)==113:

```

30

```
cam.release()
```

```
if out:out.release()
```

```
break
```

```
cv2.destroyAllWindows()
```

```
main()
```

## Algorithm for hand\_detector lib.py:

```
import mediapipe as mp
```

```
import cv2
```

```
import time
```

```
class hand_detector():
```

```
def __init__(self,mode=False,maxHands=1,detectionConf=0.5,trackConf=0.5):
```

```
    self.mode=mode
```

```
    self.maxHands=maxHands
```

```
    self.detectionConf = detectionConf
```

```
    self.trackConf=trackConf
```

31

```
self.mpHands = mp.solutions.hands
```

```
self.hands=self.mpHands.Hands(self.mode,self.maxHands,self.detectionConf,self.trackConf)
```

```
self.mpDraw = mp.solutions.drawing_utils
```

```
self.tipIds = [4,8,12,16,20]
```

```
def find_hands(self,img,draw=True):
```

```
'''
```

**Finds the landmark of the hand(s)**

**Parameters:**

**img(numpy array): Image to detect hands on**

**draw(bool-optional): Draw the hand landmarks**

**Returns:**

**Image with hand landmarks drawn**

```
'''
```

```
imgRGB = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
```

```
self.results = self.hands.process(imgRGB)
```

```
if self.results.multi_hand_landmarks and draw:
```

```
    for handLms in self.results.multi_hand_landmarks:
```

```
self.mpDraw.draw_landmarks(img,handLms,self.mpHands.HAND_CONNECTIONS)
```

```
return img
```

```
def find_position(self,img,handNo=0,draw=True):  
    '''  
    Finds the position of the special points in hand  
    Parameters:  
        img(numpy array): Image to detect hands on  
        handNo(int-optional): Which hand?  
        draw(bool-optional): Draw the special points on hand  
  
    Returns:  
        List with coordinates of special points of the hand  
    '''  
    self.lmList = []  
    if self.results.multi_hand_landmarks :  
        myHand = self.results.multi_hand_landmarks[handNo]  
        for id,lm in enumerate(myHand.landmark):  
            h,w,c = img.shape  
            cx,cy = int(lm.x*w),int(lm.y*h)  
            self.lmList.append([id,cx,cy])  
  
            if draw:cv2.circle(img,(cx,cy),10,(255,0,255),-1)  
  
    return self.lmList  
  
def fingers_up(self):  
    '''
```

**Finds which fingers are up and straight****Returns:**

List of five element corresponding to five fingers of hand.

1-Lifted up, 0-Not lifted up

'''

```
fingers=[]
```

```
if self.lmList[self.tipIds[0]][1]< self.lmList[self.tipIds[0]-1][1]:
```

```
    fingers.append(1)
```

```
else:
```

```
    fingers.append(0)
```

```
for id in range(1,5):
```

```
    if self.lmList[self.tipIds[id]][2]<self.lmList[self.tipIds[id]-2][2]:
```

```
        fingers.append(1)
```

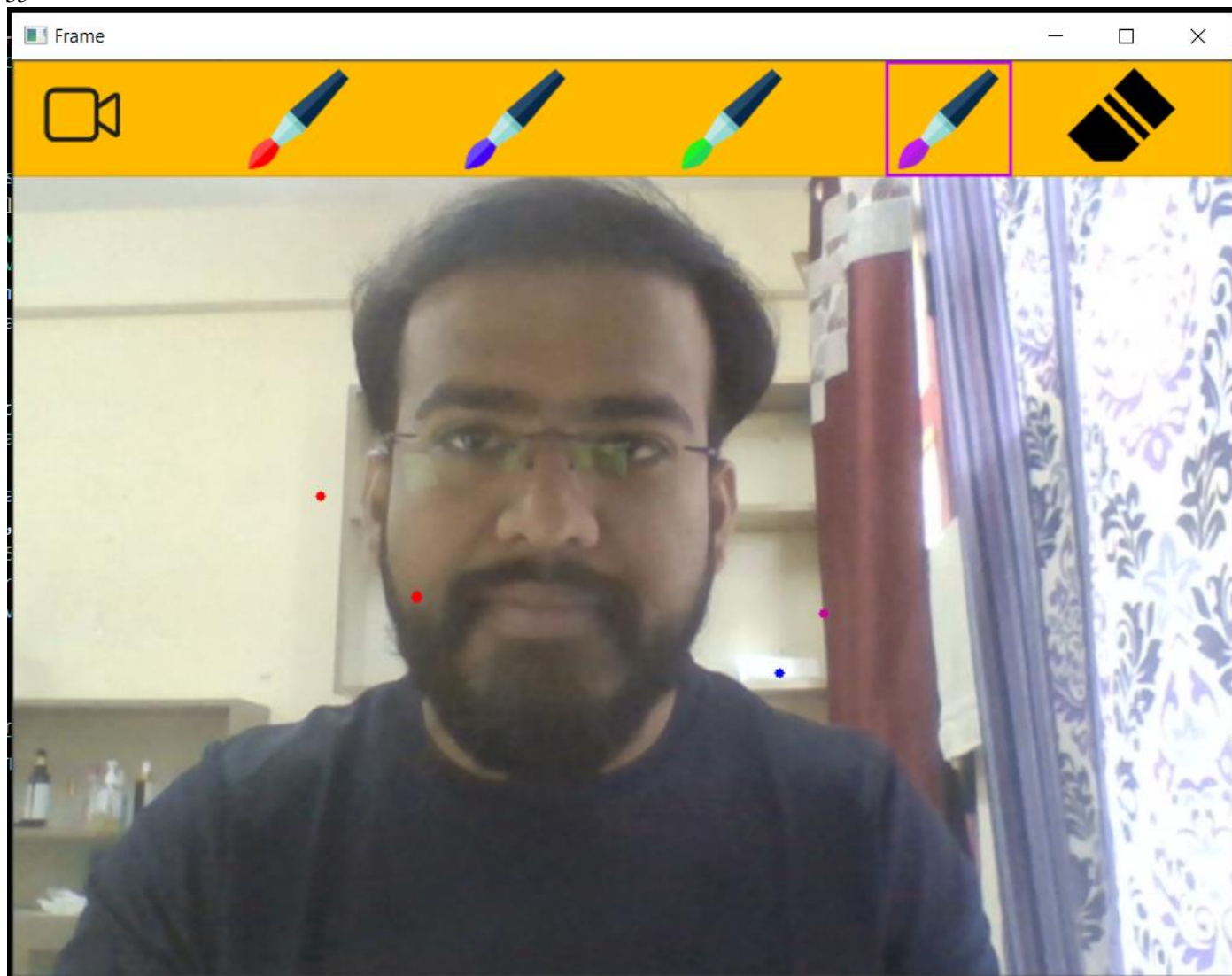
```
    else:
```

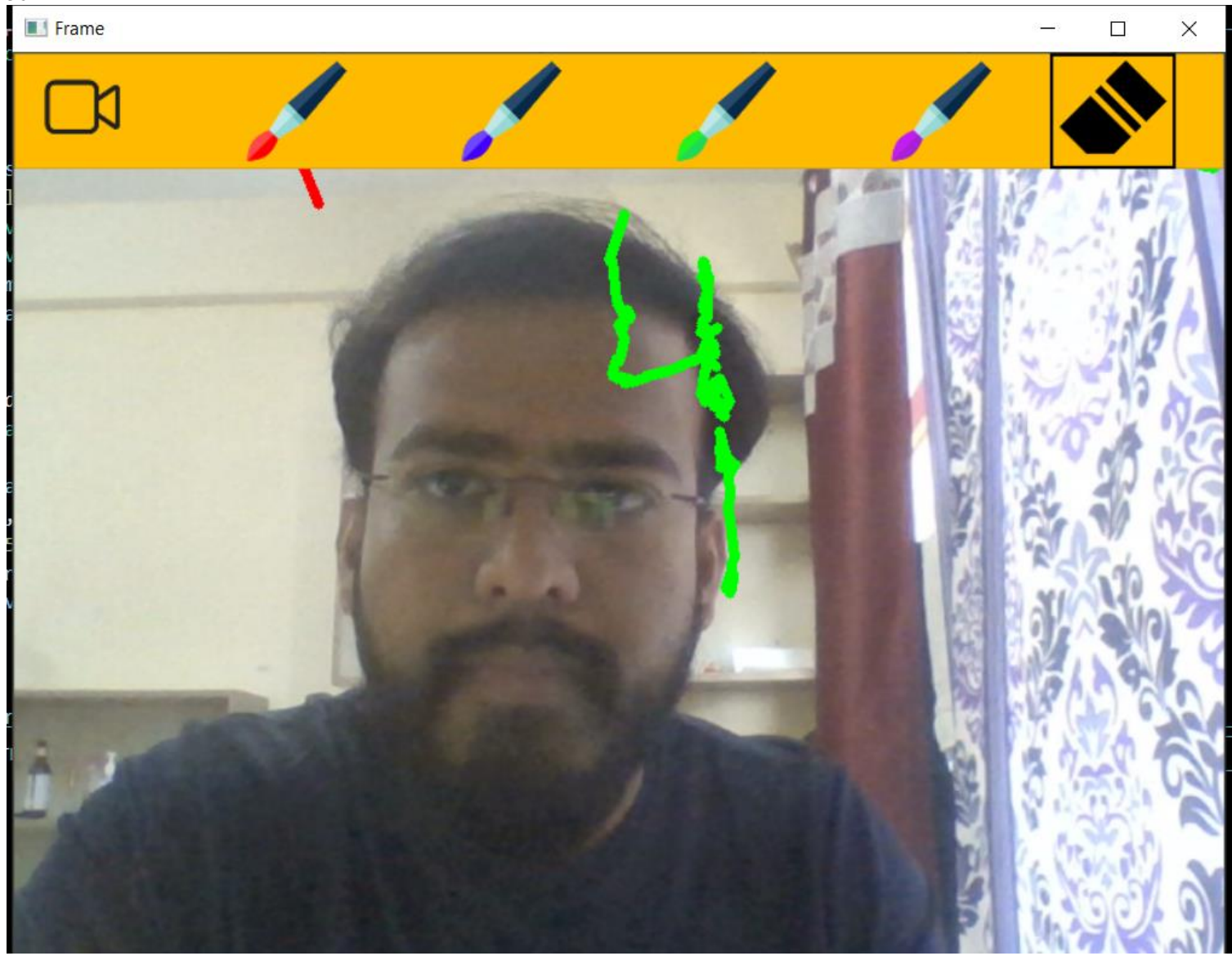
```
        fingers.append(0)
```

```
return fingers
```









## **Chapter-5: Conclusion**

The main aspect behind the product is that it enabled us to bring out the new ideas that were sustained within us for many days. This project offers teachers to teach, record and conduct classes easily through the internet. While it's easier to use it's also lighter on them economically which is an added benefit. This application has no competitors in the market yet so the technical advancement on these might be a little slow but as soon as one gets launched, the features and additions to it are extensive.

In analyzing, designing, implementing, and maintaining standards, we considered these characteristics as the foundation. These standards were made national, Hand-Write Virtual board will be an inexpensive, and less time consuming method once a system exhibiting national standards and the above mentioned characteristics is implemented.

After my research and my finalization of this project, I highly recommend that the Hand-Write Virtual Board serves to be the best to be put in use especially in the 21<sup>st</sup> century where human beings are embracing technology and where there is struggle for both the students as well as teachers regarding education. The adoption of this application should be started with lower classes and along with further improvements to the application we can promote it for higher classes and even colleges.

## **REFERENCES**

- [1] J. Francis and A. B K, "Significance of Hand Gesture Recognition Systems in Vehicular Automation-A Survey", International Journal of Computer Applications, vol. 99, no. 7, pp. 50-55, 2014.
- [2] S. Mitra and T. Acharya, "Gesture Recognition: A Survey", IEEE Trans. Syst., Man, Cybern. C, vol. 37, no. 3, pp. 311- 324, 2007.
- [3] A. S.Ghotkar and G. K. Kharate, " Hand Segmentation Techniques to Hand Gesture Recognition for Natural Human Computer Interaction", International Journal of Human Computer Interaction, vol. 3, no. 1, pp. 15-25, 2012.
- [4] Chen, F.-S., Fu, C.-M., & Huang, C.-L. (2003). Hand gesture recognition using a real-time tracking method and hidden Markov models. Image and vision computing, 21(8), 745-758.
- [5] Dipietro, L., Sabatini, A. M., & Dario, P. (2008). A Survey of Glove-Based Systems and Their Applications. Ieee transactions on systems, man, and cybernetics, part c (applications and reviews), 38(4), 461- 482.