VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi: 590 018

Digital Image Processing Mini Project report on

"VIRTUAL PAINT"

Submitted in partial fulfillment of the requirement for the award of Degree of

BACHELOR OF ENGINEERING
IN
ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

By

ABISHEK R	**1AY20AI002**
GAGANDEEP Y	**1AY20AI019**

Under the guidance of

Mrs. Kavitha Nair

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING
ACHARYA INSTITUTE OF TECHNOLOGY
(Affiliated to Visvesvaraya Technological University, Belagavi)

2022-2023

ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)
Soladevanahalli, Bangalore – 560090

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

ACHARYA
INSTITUTES

CERTIFICATE

Certified that the Digital Image Processing Laboratory with Mini Project entitled **Virtual Paint** is a bonafide work carried out by **Abishek R (1AY20AI002) & Gagandeep Y(1AY20AI019)** of sixth semester in partial fulfillment for the award of degree of **Bachelor of Engineering in Artificial Intelligence & Machine Learning** of the **Visvesvaraya Technological University**, **Belagavi**, during the year **2022-2023.** It is certified that all corrections/ suggestions indicated for internal assessments have been incorporated in the Report deposited in the departmental library. The Mini Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the **Bachelor of Engineering Degree**.

Signature of Faculty in Charge **Signature of H.O.D**

Name of the examiners **Signature with date**

1.

2.

ACKNOWLEDGEMENT

I express my gratitude to our institution and management for providing us with good infrastructure, laboratory, facilities and inspiring staff whose gratitude was of immense help in completion of this mini-project successfully.

I am grateful to the institute Acharya Institute of Technology and management with its ideas and inspiration for having provided us with the good infrastructure, laboratory, facilities and inspiring staff which has made this seminar report complete successfully.

I would like to express my sincere gratitude to **Dr. Rajath Hegde M M**, Principal, AIT for all the facilities that he has extended throughout my work.

I heartily thank and express my sincere gratitude to **Dr. Vijayashekhar S Sankannavar**, Associate Professor and Head, Dept. of AIML, AIT for his valuable support and a constant source of enthusiastic inspiration to steer us forward.

I would like to express my sincere gratitude to the Mini-project Coordinators **Mrs. Kavitha Nair R**, Assistant Professor, Dept. of AIML, AIT and **Mrs. Soumya Santhosh**, Assistant Professor, Dept. of AIML, AIT for their valuable guidance and support.

Finally, I would like to express my sincere gratitude to my parents, all teaching and non-teaching faculty members and friends for their moral support, encouragement and help throughout the completion of the mini-project.

Abishek R **1AY20AI002**
Gagandeep Y **1AY20AI019**

ABSTRACT

Virtual painting is an innovative technology that has been developed in recent years to provide a new and engaging way of creating and experiencing art. It involves the use of digital tools and techniques to simulate the process of painting in a virtual environment, allowing users to create artworks in a way that is both intuitive and expressive. This mini project explores the concept of virtual painting and its implementation using digital image processing techniques. The project focuses on developing an interactive system that enables users to create virtual paintings using a variety of digital tools such as brushes, colours, textures and eraser. It offers a range of benefits such as eliminating the need for physical paint and canvas, making it a more environmentally friendly option, and providing artists with greater flexibility and control over their creative process.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In recent years, computer vision techniques have been used to create a wide range of innovative applications, including facial recognition, object detection, and augmented reality. This project aims to leverage computer vision techniques to create a virtual painting application that provides an interactive and engaging way for users to create digital artwork. The project utilizes libraries such as OpenCV and Mediapipe HandTrackingModule in Python to detect and track hand movements, and provides users with a range of colors and brush sizes to choose from.

CHAPTER 2

PROBLEM DEFINITION

Artists use canvases to create art, but not everyone has access to them or the space to use them. Additionally, physical canvases have limitations in size and durability. With the rising popularity of digital art and the increasing accessibility of computer vision technologies, there is a need for a more intuitive and natural way for artists to create and express their ideas. Therefore "Virtual Paint“ finds its potential which aims to provide a digital canvas that allows users to paint and draw using only their hands, creating a more natural and immersive experience that enhances the artist's creativity and expression.

CHAPTER 3

LITERATURE SURVEY

3.1. LITERATURE REVIEW

S.N	PAPER TITTLE & PUBLICATION DETAILS	NAME OF THE AUTHORS	TECHNICAL IDEAS / ALGORITHMS USED IN THE PAPER & ADVANTAGES	SHORTFALLS/DISADVANTAGES & SOLUTION PROVIDED BY THE PROPOSED SYSTEM
1	Virtual Painting with OpenCV Using Python [2020] IJSRST	Yash Patil, Mihir Paun, Deep Paun, Karunesh Singh, Vishal Kisan Borate	Python,OpenCV, hand detection and tracking algorithms, video processing algorithms.	Inaccuracy, No UI and limited info on user experience.
2	Virtual Air Canvas Application using OpenCV and Numpy in Python [2020] IJARIIE	Asst Prof.Jahnavi S, K Sai Sumanth Reddy, Abhishek R, Abhinandan Heggde, Lakshmi Prashanth Reddy	Python, OpenCV, Numpy, Matplotlib.	Limited Scope, No data on Performance.

3	Virtual Paint Application Using Hand Gestures [2022] IRJET	Niharika M, Neha J, Mamatha Rao, Vidyashree K P	Python, Numpy, Haar Cascades, OpenCV, Machine Learning Algorithms	Not implemented in real time.

Table 3.1 Literature Survey

3.2 PROPOSED SOLUTION

The aim is to develop an immersive virtual painting application that allows users to unleash their creativity without the need for physical tools or materials. The application will feature a user-friendly interface with a canvas where users can paint using various virtual brushes, colors, and effects. Motion tracking technology will be employed to track the user's hand movements and translate them into brush strokes on the virtual canvas, providing a natural and intuitive painting experience. Users will have the option to customize brush settings, select colors from the palette, and work with multiple layers to create complex artwork. This project uses a gesture system where we use two fingers for selecting tools and one finger to draw.

CHAPTER 4

REQUIREMENTS SPECIFICATION

Requirements specification is a specification of software requirements and hardware requirements required to do the project.

4.1 Hardware Requirements Specification

Hardware Requirements are the hardware resources that are need to do the project work. These resources are a computer resource provides functions and services to do the project. Hardware resources required for our project are shown below.

- Processor : Intel Core i5 or above
- RAM : >=8GB
- Hard disk : Minimum 10 GB

4.2 Software Requirements Specification

Software Requirements are the software resources that are need to do the project work. These resources are installed on a computer in order to provide functions, services, hardware accessing capabilities to do the project.

In our project we used the following software resources.

- Jupyter Notebook
- Necessary Library Packages

4.3 Functional Requirements

Functional requirements specify a function that system or a system component must be able to perform. It can be documented in various ways.

- The system should be able to gather the data from the camera. The system should be able to collect input feed from the camera of the device.
- The system should be able to process the data captured by the camera and give a corresponding output.
- The system should be able to detect and track the hand movements.
- The system should be able to detect hand gestures for selecting various tools.
- High Accuracy Level: Upon recognizing hand movements the system should provide corresponding outputs accurately.

4.4 Non-Functional Requirements

- Realibilty: The virtual paint application is reliable as it can consistently performs its functions without any errors.
- Availabilty: The project will be deployed on a public shared server so it will be available all the time and will be accessible anywhere of the world using internet.
- Maintainabilty: It is very easy to maintain the system. The system has been developed on jupyter notebook so anyone who has the knowledge of jupyter notebook, can easily maintain the system.
- Portability: Yes this system is portable and we can switch the servers very easily.
- Browser Compatibilty: The project being web based required compatibility with at least the popular web browsers. Microsft windows XP and above, Linux and Macintosh being the current popular operating system and Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari and Google Chrome being the currently popular web browsers.

CHAPTER 5

SYSTEM DESIGN

5.1 Data Flow Diagram

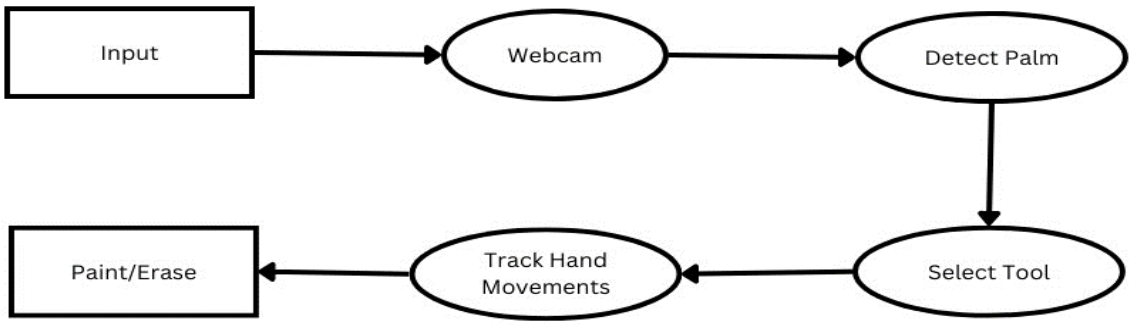


Figure 5.1 Data Flow diagram

5.2 Use case Diagram

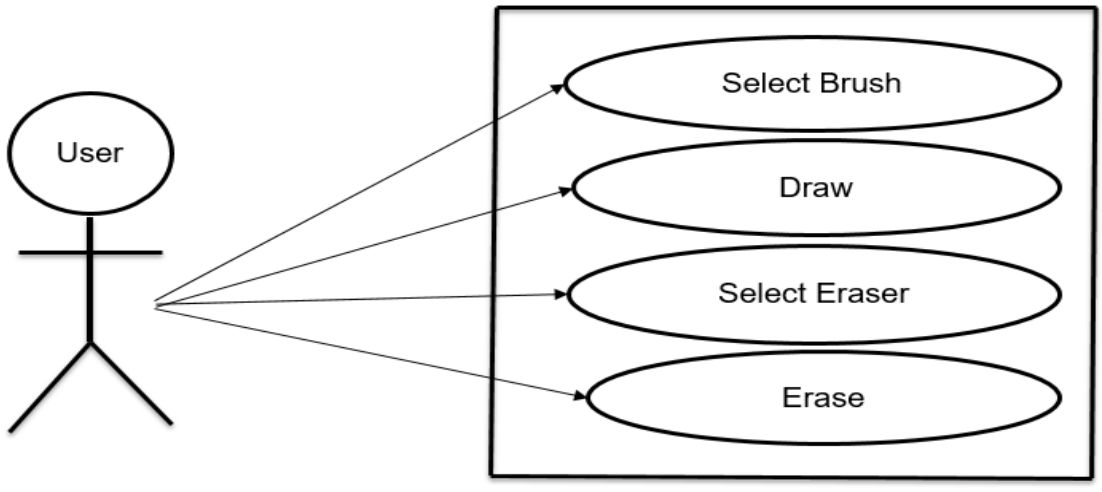


Figure 5.2 Use Case diagram

5.3 Module Description

Importing Libraries: The necessary libraries are imported, including OpenCV, MediaPipe, NumPy, and the os module for file operations.

Setting up Webcam: The code initializes the webcam to capture video frames. The resolution and frame rate of the webcam are set, and the initial canvas for drawing is created.

Loading Header Images: A folder containing header images is specified, and the code loads these images into a list. These header images will be displayed at the top of the video frame.

Hand Landmark Detection: The code processes each frame and uses MediaPipe's Hands module to detect hand landmarks in the frame. The hand landmarks consist of 21 points representing different parts of the hand.

Gesture Recognition: The code analyzes the detected hand landmarks to recognize different hand gestures. It checks the finger positions to determine the mode of operation: Selection Mode, Standby Mode, or Draw Mode. It also detects when the hand is closed to clear the canvas.

Displaying Output: The code combines the video frame, header image, and the drawn canvas into a final image, which is displayed in a window using OpenCV's imshow function

CHAPTER 6

IMPLEMENTATION

6.1 Tools and Technologies Used

Jupyter: Jupyter is an open-source web application that allows users to create and shared documents containing live code, visualizations, and narrative text. It provides an interactive computing environment where users can write and execute code in different programming languages, including Python, R, and Julia.

OpenCV: OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being an Apache 2 licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects.

Numpy: A library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions.

MediaPipe: MediaPipe is a machine learning solution framework developed by Google. MediaPipe has various pre-trained models inbuilt. MediaPipe is used for video processing and the HandTrackingModule is used to detect hand gestures and movements.

6.2 Algorithms / Methodologies Used

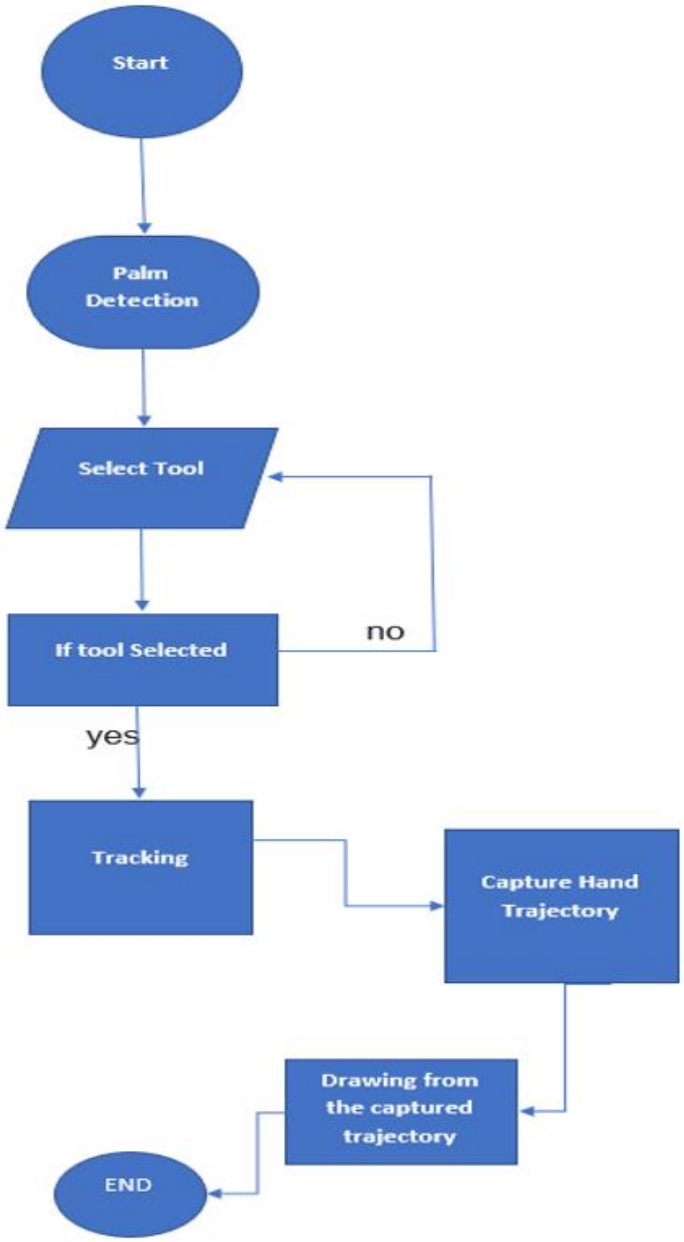


Figure 6.2 Use Case diagram

CHAPTER 7

SYSTEM TESTING

SL No.	Test condition	Expected result	Result
1	Using two fingers to select the brush tool with a particular color.	Brush selected.	Successful
2	Moving across the camera to draw on the canvas.	Strokes appears on the canvas.	Successful
3	Using two fingers to select the eraser.	Eraser selected.	Successful
4	Moving across the camera to erase on the canvas.	Erase the strokes.	Successful
5	On making a fist.	Clear the canvas.	Successful

Table 7.1 Test Cases

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Virtual Paint offers a transformative and immersive experience, bridging the gap between traditional art and technology. By leveraging virtual reality and advanced digital tools, it empowers artists to unleash their creativity in unprecedented ways. With its intuitive interface and limitless possibilities, Virtual Paint has the potential to revolutionize the art world and redefine how we perceive and interact with art. It opens doors to new artistic expressions, encourages experimentation, and fosters a sense of community among artists. Embracing the power of technology, Virtual Paint redefines the canvas, allowing artists to create masterpieces that transcend boundaries and inspire generations to come.

The future scope for Virtual Paint is promising and opens up a world of possibilities. By leveraging virtual reality (VR) and augmented reality (AR) technologies, the project can revolutionize the way art is created and experienced. With further development, the project has the potential to provide a realistic and immersive virtual painting experience, allowing users to unleash their creativity without the need for physical materials or space. The integration of machine learning algorithms could enable the system to analyze user strokes and provide real-time feedback, enhancing the learning process for aspiring artists. Additionally, the project can extend its application beyond art by incorporating collaborative features, enabling multiple users to paint together in a shared virtual environment, fostering creativity and collaboration on a global scale.

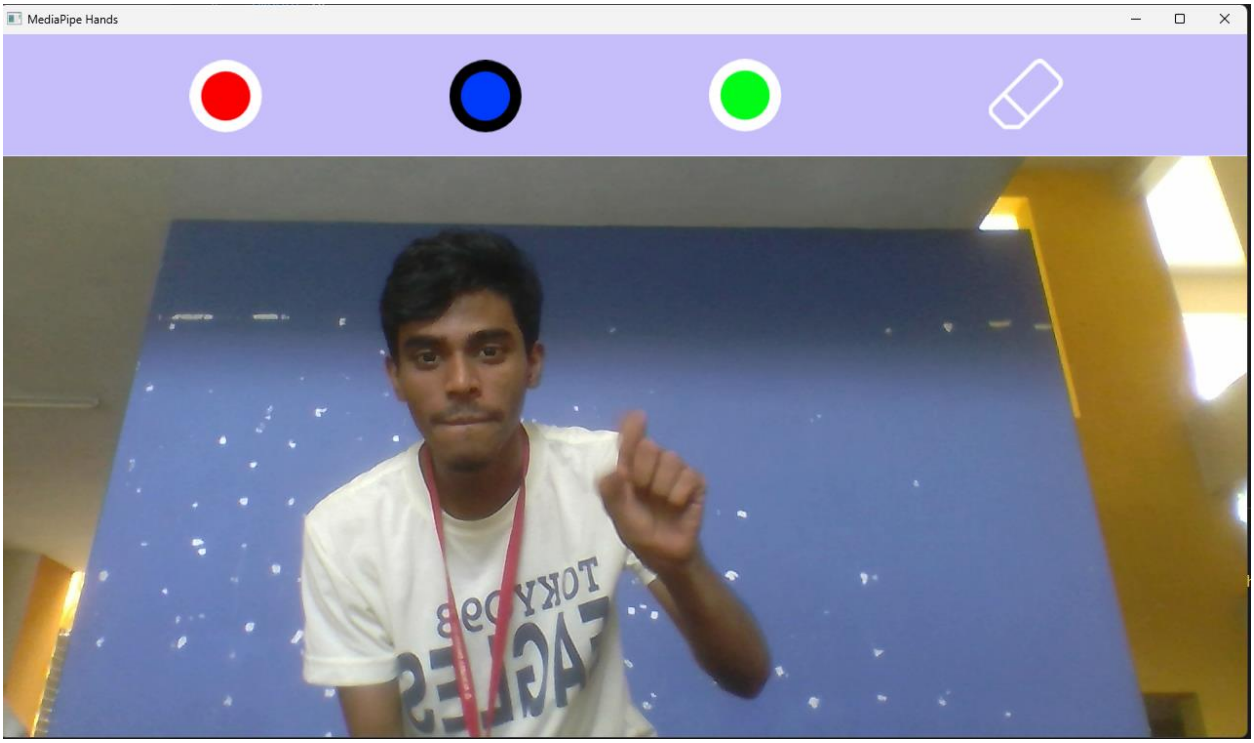
APPENDICES

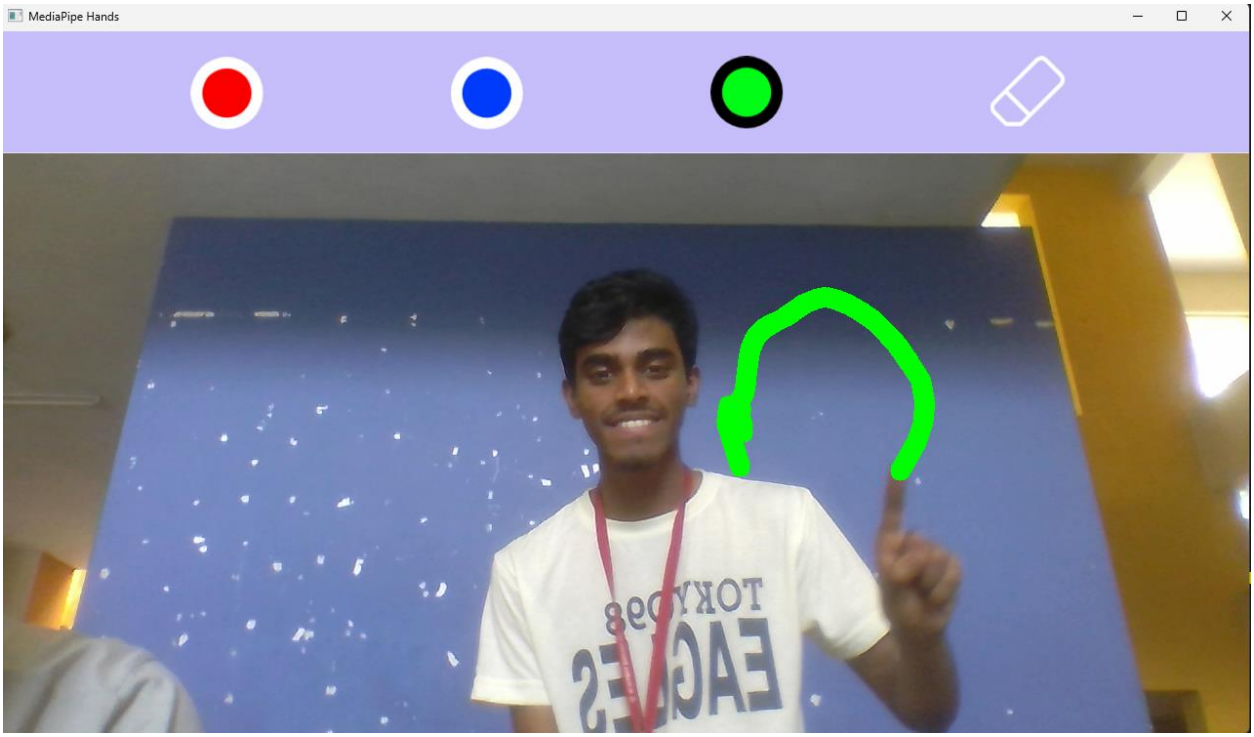
A. SAMPLE CODE

```
import cv2
import mediapipe as mp
import numpy as np
import os
import math
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands

# For webcam input:
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
cap.set(cv2.CAP_PROP_FPS, 5)
width = 1280
height = 720
cap.set(3, width)
cap.set(4, height)

# Image that will contain the drawing and then passed to the camera image
imgCanvas = np.zeros((height, width, 3), np.uint8)

# Getting all header images in a list
folderPath = 'Hand Tracking Project\Header'
myList = os.listdir(folderPath)
overlayList = []
for imPath in myList:
    image = cv2.imread(f'{folderPath}/{imPath}')
    overlayList.append(image)

# Presettings:
header = overlayList[0]
drawColor = (0, 0, 255)
thickness = 20 # Thickness of the painting
tipIds = [4, 8, 12, 16, 20] # Fingertips indexes
xp, yp = [0, 0] # Coordinates that will keep track of the last position of the index finger

with mp_hands.Hands(min_detection_confidence=0.85, min_tracking_confidence=0.5, max_num_hands=1) as hands:
    while cap.isOpened():
        success, image = cap.read()
        if not success:
            print("Ignoring empty camera frame.")
```

```
## Selection Mode - Two fingers are up
nonSel = [0, 3, 4] # indexes of the fingers that need to be down in the Selection Mode
if (fingers[1] and fingers[2]) and all(fingers[i] == 0 for i in nonSel):
    xp, yp = [x1, y1]

    # Selecting the colors and the eraser on the screen
    if(y1 < 125):
        if(170 < x1 < 295):
            header = overlayList[0]
            drawColor = (0, 0, 255)
        elif(436 < x1 < 561):
            header = overlayList[1]
            drawColor = (255, 0, 0)
        elif(700 < x1 < 825):
            header = overlayList[2]
            drawColor = (0, 255, 0)
        elif(980 < x1 < 1105):
            header = overlayList[3]
            drawColor = (0, 0, 0)

    cv2.rectangle(image, (x1-10, y1-15), (x2+10, y2+23), drawColor, cv2.FILLED)

## Stand by Mode - Checking when the index and the pinky fingers are open and dont draw
nonStand = [0, 2, 3] # indexes of the fingers that need to be down in the Stand Mode
if (fingers[1] and fingers[4]) and all(fingers[i] == 0 for i in nonStand):
    # The line between the index and the pinky indicates the Stand by Mode
    cv2.line(image, (xp, yp), (x4, y4), drawColor, 5)
    xp, yp = [x1, y1]

## Draw Mode - One finger is up
nonDraw = [0, 2, 3, 4]
if fingers[1] and all(fingers[i] == 0 for i in nonDraw):
    # The circle in the index finger indicates the Draw Mode
    cv2.circle(image, (x1, y1), int(thickness/2), drawColor, cv2.FILLED)
    if xp==0 and yp==0:
        xp, yp = [x1, y1]
    # Draw a line between the current position and the last position of the index finger
    cv2.line(imgCanvas, (xp, yp), (x1, y1), drawColor, thickness)
    # Update the last position
```

B. SNAPSHOTS





REFERENCES

[1] Asst Prof.Jahnavi S, K Sai Sumanth Reddy, Abhishek R, Abhinandan Heggde, Lakshmi Prashanth Reddy: Virtual Air Canvas Application using OpenCV and Numpy in Python: IJARIIE-ISSN(O)-2395-4396 Vol-8 Issue-4 2022

[2] Yash Patil, Mihir Paun, Deep Paun, Karunesh Singh, Vishal Kisan Borate: Virtual Painting with Opencv Using Python: Print ISSN: 2395-6011 | Online ISSN: 2395-602X Volume 5 Issue 8, November-December-2020

[3] Niharika M, Neha J, Mamatha Rao, Vidyashree K P :VIRTUAL PAINT APPLICATION USING HAND GESTURES:e-ISSN: 2395-0056 p-ISSN: 2395-0072 Volume: 09 Issue: 04 | Apr 2022

[4] Digital image processing Third edition,2015, Rafael C. Gonzalez, Richard E. Woods

[5] Digital Image Processing- S.Jayaraman, S.Esakkirajan, T.Veerakumar, TataMcGraw Hill 2014.