Oriental Institute of Science & Technology, Bhopal

Gesture-Based Paint Application

A

Project Work

Submitted as Minor Project in Partial fulfillment for the award of Graduate Degree in Bachelor of Engineering in Computer Science & Engineering.

Submitted to

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA
BHOPAL (M.P)

Submitted By

Faiza Khan (0105CS181039)

Under the Guidance of

Prof. Bhavana Choubey

(Department of Computer Science & Engineering)

Oriental Institute of Science & Technology, Bhopal

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JAN-JUN 2021

ORIENTAL INSTITUTE OF SCIENCE & TECHNOLOGY, BHOPAL

CERTIFICATE

This is to certify that the Minor project entitled "Gesture-Based Paint Application" being submitted by Faiza Khan (0105CS181039) student of VIth Semester, Department of Computer Science & Engineering in partial fulfillment for the award of Bachelor of Technology degree from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) in session 2020-21. It has been found to be satisfactory and hereby approved for submission. This record of bonafide work is carried out by her under our supervision.

Prof. Bhavana Choubey
Guide
Department of
Computer Science & Engineering

Prof. Atul Barve
Head
Department of
Computer Science & Engineering

ACKNOWLEDGEMENT

I take the opportunity to express my cordial gratitude and a deep sense of indebtedness to my guide **Prof.** Bhavana Choubey for the valuable guidance and inspiration throughout the project duration. I feel thankful to her for her innovative ideas, which led to the successful submission of this minor project work. I feel proud and fortunate to work under such an outstanding mentor in the field of **Gesture-Based Paint Application**. She has always welcomed my problem and helped me to clear my doubt. I will always be grateful to her for providing me moral support and sufficient time.

I owe sincere thanks to **Dr. *Rajesh Shukla*,** Director OIST, for providing me with moral support and necessary help during my project work in the Department.

At the same time, I would like to thank **Prof. Atul Barve (HOD, CSE)** and all other faculty members, and all non-teaching staff of the Department of Computer Science & Engineering for their valuable cooperation.

I would also thank my Institution, faculty members, and staff without whom this project would have been a distant reality. I also extend my heartfelt thanks to my family and well-wishers.

Faiza Khan
0105CS181039

APPROVAL CERTIFICATE

This is to certify that the project entitled **Gesture-based paint application** being submitted by **Faiza Khan 0105CS181039** student of Sixth Semester, Bachelor of Technology in Computer Science and Engineering have done her work as Minor Project for Partial fulfillment of the degree from RGPV, Bhopal (M.P.).

Prof. Bhavana Choubey

Date: 02/07/2021

Signature

Table of Contents

LIST OF FIGURES

ABSTRACT

Our project “Gesture-based paint application” allows user to draw on screen with full ease using real time hand gestures. The application also allows user to switch between brush size and paint colours. It also contains many other features. Hand gestures are the best way to communicate with a computer system. Almost all paint programs are more or less similar on the basis of their functionality and the way the user gives input. We thought this will give EDGE to our project.

In the modern digital age of social media where the photos and images have much importance, there is a need for user -friendly painting tool. The intention of this painting tool is to provide an accurate and convenient pointing device. This project aims to bridge the gap between machines and humans. Controlling the computer system using movements detected in camera is more user friendly than traditional methods. The OpenCV library has been preferred for this method.

1. Introduction

i. Overview

A paint program is a software graphics program that allows the user to draw or paint bitmapped images on a computer. The aim of this project is to make the process of painting more user-friendly by using the webcam of a computer to track the motion of an object and use it as a paintbrush. Given the real-time webcam data, this paint-like python application uses the OpenCV library to track an object of interest and allows the user to draw by moving the object, which makes it both awesome and challenging to draw simple things. This project allows user to draw on the screen with full ease using real-time hand gestures. The application also allows user to switch between brush size and paint colors. It also contains many other features.

ii. Motivation

In the current scenario, MS Paint or different paint softwares are used for editing or drawing images on the screen with traditional pointing devices. For editing images or drawing shape users do not have any other option except keyboard, mouse to use paint software. As device or hardware interface is required to use software which impacts on the cost of the project or system. So to avoid this we are proposing a Smart paint software which can be used without any hardware interface. Hand gestures are best way to communicate with computer system. Almost all paint programs are more or less similar on the basis of their functionality and the way user gives input. We either use a mouse, drawing pad or touch screen to feed input. But there are other ways to give input, which are less frequently used or unknown to people and are easier for user as compared to previous ways. One of them is gesture-based way of giving input to the paint application. This is the driving force behind this project. So this project provides a more convenient way to the users, to feed input to the paint application that is through hand gestures.

2. Project Definition And Scope

i. Definition

This project uses python language along with OpenCV library to track an object in the frame and uses the detected object to draw colored lines. To perform video tracking an algorithm analyzes sequential video frames and outputs the movement of targets between the frames. Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing.

ii. Scope

The proposed system helps the user to perform hand gesturing in an efficient and cost effective manner. After the system is set up, the user interaction can be increased drastically, which makes it even easier for a user because they can interact easily and effectively with the system, rather than the usage of traditional software products. Using Machine learning, we can eliminate cost factors as well as noise or any type of barrier easily. With this approach, we can easily improve the scope of our software. With small modifications to the proposed system, it can be used in many other fields. In other words it can replace every existing user interacting devices. With small modification to the proposed system it can be used in many other fields. In other words it can replace every existing user interacting devices.

3. Literature Survey

The following papers show a survey of different proposed hand gesture and computer-based approaches to eliminate hardware interfaces that costs less.

1. **Hand Gesture Real-Time Paint Tool Box: Machine Learning Approach [1]**, they discussed about gesture based paint tool box which has 6 gestures to draw line, circle. This paper states various approaches through which a paint tool box accuracy can be achieved. To achieve more accuracy than any other approach they have used machine learning approach. According to the survey Machine learning approach gives 96% accurate result. They have Haar-Like classifier which is used to track the hands and they have also used edge detection for calculating threshold value of the object which is in front of the camera. They implement this by using simple hand gestures and webcam of laptop or desktop. Haar-Like classifier used to compare the image which is captured to the gesture present in current dataset, if matched resulted action performed on the screen. Isolated hand are captured and a classifier used to separate them with background image and other body parts. For color selection they have used some conversion method which is Gray conversion if selected color is in Black and White or in RGB color form, for color selection they have used disjointing the color set used to analyze the color and draw colored shaped object.

2. **Gloved and Free Hand Tracking based Hand Gesture Recognition [2]**, they discussed about continuous hand gesture recognition. It reports the robust and efficient hand tracking as well as segmentation algorithm where a new method, based on wearing glove on hand is utilized. We have focused on another tracking algorithm, which is based on the skin color of the palm part of the hand i.e. free hand tracking. A comparative study between the two tracking methods is presented

in this paper. A fingertip can be segmented for the proper tracking in spite of full hand part. Waving goodbye is a gesture. Pressing the key on a keyboard is not the gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. So the gloved hand is tracked and the value stored is active to recognition. According to glove based recognition technique it is easy to find color selection by identifying the color of the glove worn by the user. Free hand tracking reduces the cost estimation of time and space of the algorithm.

3. **Human Computer Interaction Using Face and Gesture Recognition [3]**, they discussed and presented a face and gesture recognition based human-computer interaction (HCI) system using a single video camera. Different from the conventional communication methods between users and machines, they combined head pose and hand gesture to control the equipment. They can identify the position of the eyes and mouth, and use the facial center to estimate the pose of the head. Two new methods are presented: automatic gesture area segmentation and orientation normalization of the hand gesture. It is not mandatory for a user to keep gestures in upright position, the system segments and normalizes the gestures automatically. The user can control multiple devices, including robots simultaneously through wireless network.

4. Proposed Approach

In order to make the painting tool more user friendly we are proposing a system which will help the user to draw on screen using movement detection of a cursor using camera of a computer.

A video is composed of infinite frames at different instant. Live stream video content will be taken as input and the program will detect objects which appear to be similar to the cursor. The program will proceed to draw contours around the objects and in turn select the one which is largest and most prominent among all. Then the movement of the cursor is tracked and the user can thus draw on the screen using movement detection of the cursor.

The program will consist of a GUI which will show the blank canvas (which is to be painted) and different configuration options of the paint application eg. clearing the screen, change brush size/shape, change colour of brush.

The program will also have another window, which will show the live video stream input and will also show how are the frames getting captured and masked to detect the cursor for the painting brush.

Contours is a Python list of all the contours in the image. Each individual contour is a Numpy array of (x,y) coordinates of boundary points of the object.

5.Explanation of the implementation

1. Camera Operation: The camera of the computer is used to capture the video. For this, we will have to create a VideoCapture object. Its argument can be index of device camera or the name of the video file being used. It is thus a way to define which camera is being used. If there is only one camera we pass 0. After this, each frame is captured. At the end, the video capture is released.

2. Capturing frames: The infinite loop is used so that the camera captures the frames in every instance and is open during the entire course of the program. After capturing the live stream frame by frame we convert the color space of the frame.

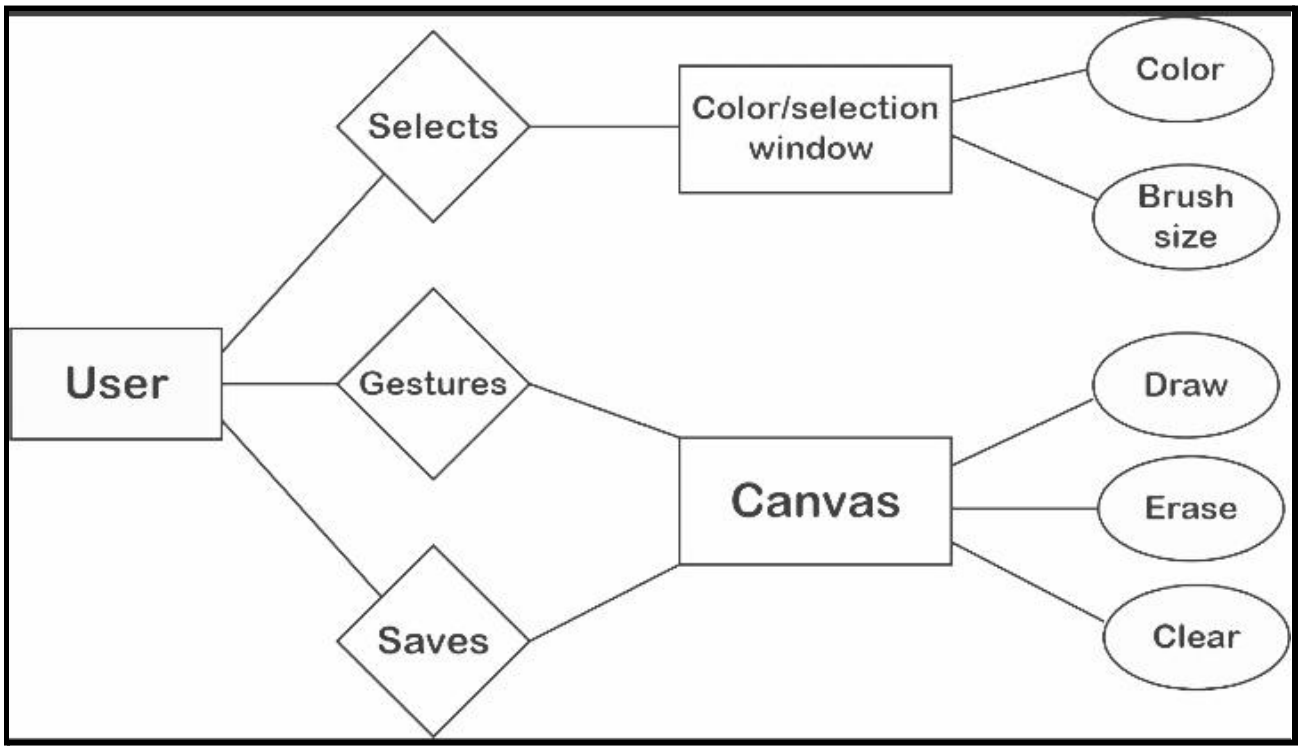
3. Masking Technique: The mask is basically creating some specific region of the image following certain rules. Then the frame, res, and mask can be displayed on each window separately.

4. Display the frame: The video input will be displayed on the screen along with the canvas on which we have to draw the figures. For this imshow() function will ge used. The function waitKey() waits for key event , it will be used as HighGui cannot process window events like redraw resizing, input event, so we use it for delay. There will also be integration of keyboard keys for some functionalities.

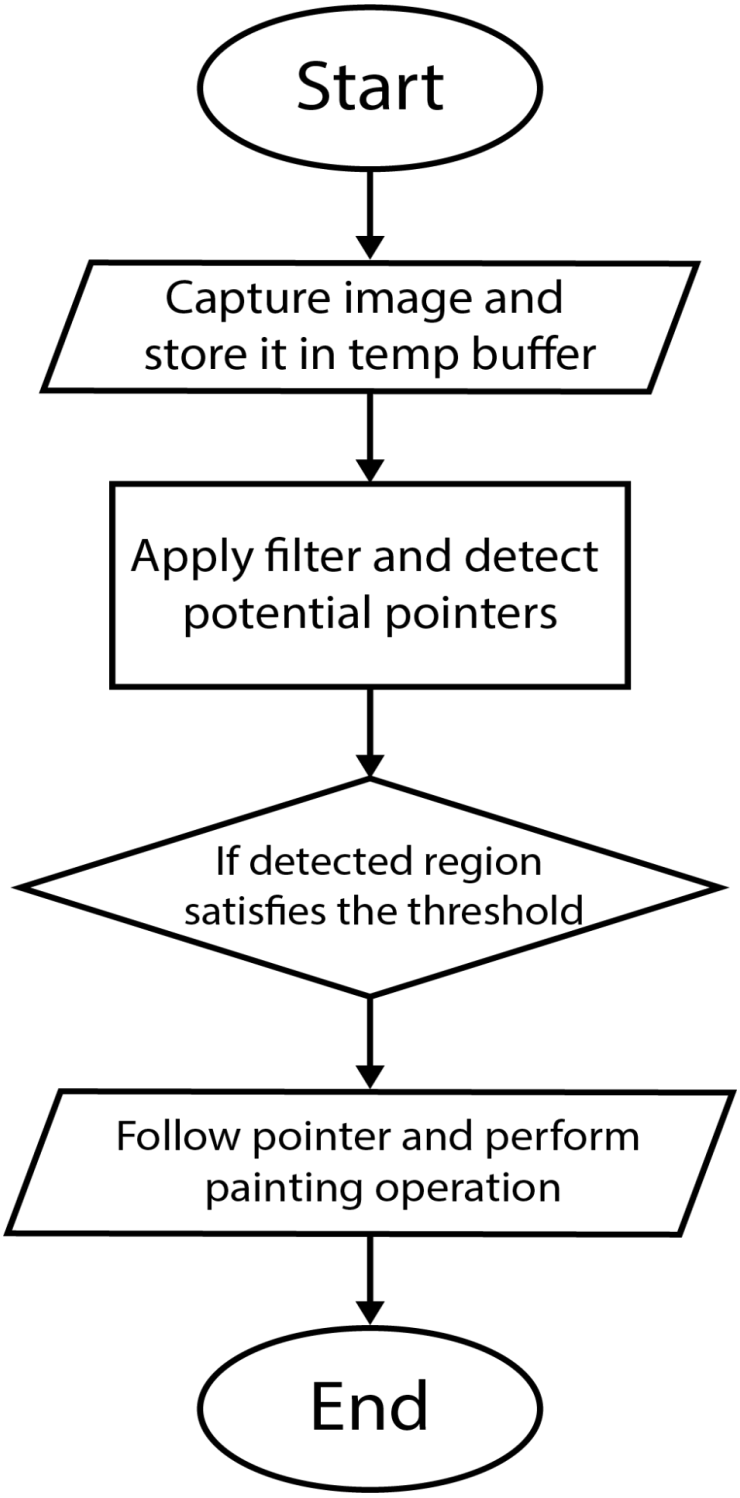
5. Functionality to save the image: The program will be able to save the painted image.

6.Design

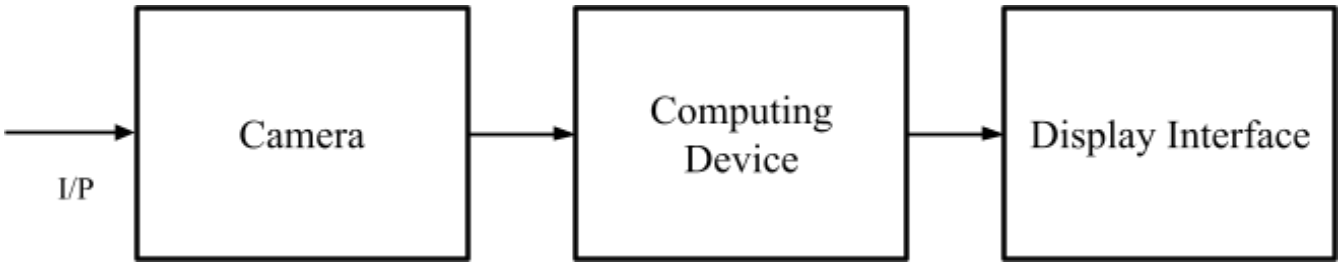
6.1 E-R diagram



6.2 Data flow diagram



6.3 General view of painting tool



7. Requirements

Hardware Requirements :

- Computer
- Camera

Software Requirements :

- Python3
- Open-CV
- Other python modules and packages

8. Coding

i. Code for selection of colors/clear button/eraser:

```
def what_menu_button_pressed(self,center):
    if center[0]>65:
        return None
    if 40 <= center[1] <= 140:  # Clear All
        return "CLEAR"
    elif 160 <= center[1] <= 255:
        return "BLUE"
    elif 275 <= center[1] <= 370:
        return "GREEN"
    elif 390 <= center[1] <= 485:
        return "RED"
    elif 505 <= center[1] <= 600:
        return "YELLOW"
    elif 601 <= center[1] <= 700:
        return "ERASER"
    return None
```

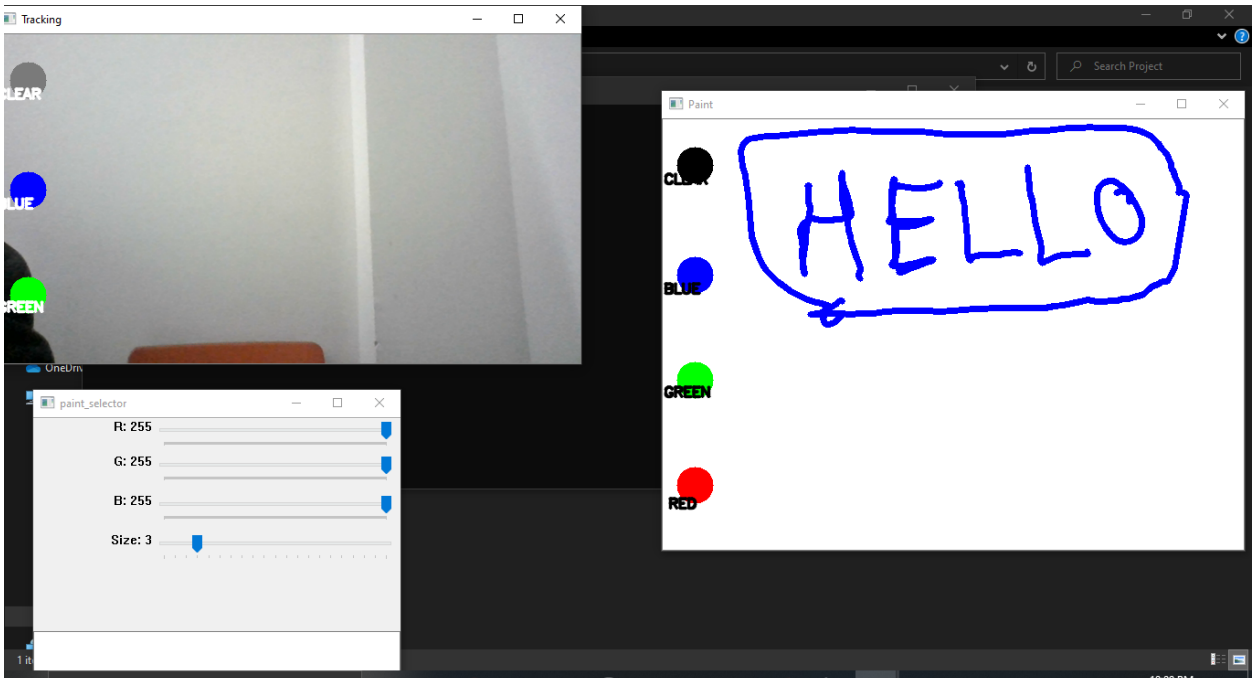
ii. Code for finding largest contour of all contours:

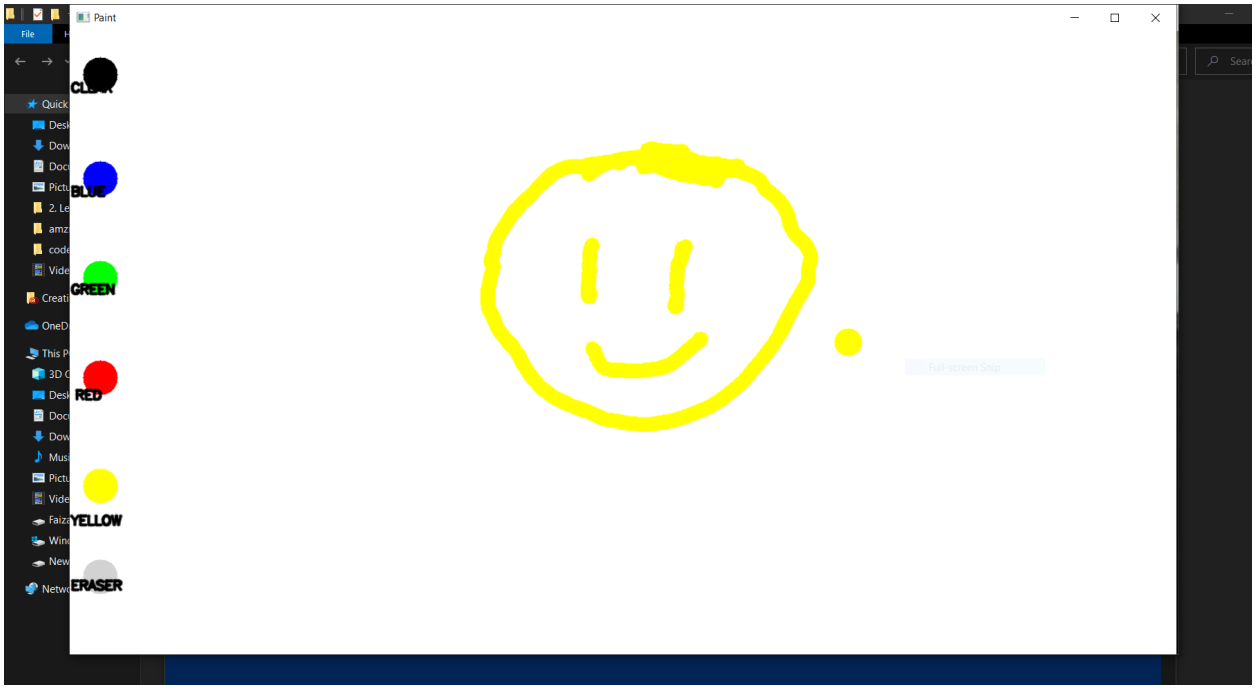
```
def find_largest_contour(self, contours):
    number_of_contours = len(contours)
    largest_contour_area = -1
    index_of_largest_contour=-1
    if number_of_contours > 0:  # if you have found atleast 1 countours
        # loop through the contours and find the largest one which we consider is the hand
        for i in range(number_of_contours):
            current_contour_area = cv2.contourArea(contours[i])
            if current_contour_area > largest_contour_area:
                largest_contour_area = current_contour_area
                index_of_largest_contour = i
        largest_contour = contours[index_of_largest_contour]
        return largest_contour
    else:
        return None
```

iii. Code for key mappings for various functions:

```
k = cv2.waitKey(1) & 0xFF
img1 = self.img.copy()
img1[:] = [self.b, self.g, self.r]

if k == ord("s"):
    if self.drawing == True:
        self.drawing = False
    else:
        self.drawing = True
    pcenter = None
    first=False

elif k == ord("q"):
    break

elif k == ord("p"):
    cv2.imwrite("frame%d.jpg" % self.imageindex, self.paintWindow)
    self.imageindex += 1

elif k == ord("m"):
    if self.size < 50:
        self.size += 2

elif k == ord("n"):
    if self.size > 2:
        self.size -= 2
```

9. Screen Layouts





10. Expected Results and Discussion

In the modern digital age of social media where the photos and images have much importance, there is a need for user friendly painting tool. The traditional painting software requires a hardware pointing device such as a mouse. In most cases we need a hardware medium for interacting with the software system. The direct use of movements captured by a camera is an attractive method for providing natural human computer interaction. It is an excellent extension of computer vision and human computer interaction fields.

Thus, the intention of this painting tool is to provide accurate and convenient pointing device. This project aims to bridge the gap between machines and humans. Controlling the computer system using movements detected in camera is more user friendly than traditional methods. Open CV library has been preferred for this method.

11. Activity Time Chart

PROJECT ACTIVITY	DURATION
Learn Python and its libraries such as numpy, collections, etc.	Week 1- Week 2
Analyzing requirements and setting up the environment	Week 3
Learning and testing some basic programs in OpenCV.	Week 4
Implementing the paint application window and object detection.	Week 5
Adding basic features to the project, eg: switching between colors, changing brush size, etc.	Week 6
Adding advanced features and testing the project in different environments.	Week 7

Fig. 11 Activity time chart table

12. Future Enhancements

- Working on User Interface and User Experience. Improving the look of the palette and erasing tools to make it more user-friendly and easy to use. Making the user interface more interactive and give users a better experience by combining the 3 different tabs into 1 tab that fits the screen.
- Increasing the accuracy in detecting contours such that the pointer on the canvas does not get distracted or deflected by an object similar in color to the contour taken by the user
- Fill color function.

13. References

[1] Vandit Gajjar, Viraj Mavani, and Ayesha Gurnani “Hand Gesture Real Time Paint Tool-Box: Machine Learning Approach,” In IEEE International Conference on Power, Control, Signals and Instrumentation Engineering.

[2] Dharani Mazumdar, Anjan Kumar Talukdar, Kandarpa Kumar Sarma “Gloved and Free Hand Tracking based Hand Gesture Recognition” Dept. of Electronics & Communication Engineering in ICETACS 2013.

[3] Yo-Jen Tu, Chung-Chieh Kao, and Huei-Yung Lin “Human Computer Interaction Using Face and Gesture Recognition” In Department of Electrical Engineering, National Chung Cheng University, Chia-Yi 621, Taiwan, R.O.C.

14. Bibliography

- https://www.pyimagesearch.com/2018/07/19/opencv-tutorial-a-guide-to-learn-opencv/
- https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
- https://en.wikipedia.org/wiki/OpenCV
- https://www.researchgate.net