Artificial Intelligence

Air Canvas

3rd-BSAI

Session:2023-2027

SUPERIOR UNIVERSITY

BSAI-Department of Software Engineering

The Superior University-Lahore

Fall 2024

Artificial Intelligence

Nature of project	Development		
Area of specialization	Artificial Intelligence		
Individual Project			
Roll_no	Student name	Email ID	Signature
076	Fatima Nadeem	Su92-bsaim-f23-076@superior.edu.pk	Fatima Nadeem Khan Niazi

Artificial Intelligence

Table of Contents

Table of Figures

Chapter:1

Introduction:

Air Canvas

Introduction:

The Air Canvas project is a piece of creativity and interactivity I've developed to allow users to draw digitally without using the traditional brushes or styluses. This creates an interactive tool whereby my project can actually detect the bead of a dark-colored pen, equivalent to the nib of a dark pen, in a process that traces movements to draw on a virtual canvas. This unique approach makes drawing intuitive and touch-free, transforming simple hand movements into digital art. The project uses Python as the back-end technology and powerful librarianship as **OpenCV**, and **NumPy** for processing video frames in real time and efficiently handling data.

Work by using:

The key concept of Air Canvas is to offer a modern drawing concept where users draw using a pen or an object with a dark-colored tip in front of a camera. OpenCV captures the live video feed coming through the webcam and detects the dark-colored nib by contrast with the surroundings. It tracks the movement of the nib and uses this data to draw lines or shapes on a virtual canvas displayed on the computer. This innovative solution

eliminates the need for physical contact, making the process both fun and hygienic.

Processor:

Technically, the Air Canvas works by processing the video feed from the webcam to track the dark-colored pen nib. A binary mask based on the HSV thresholds generates such an image which isolates the desired color and ignores the background noise. The nib position is highlighted, and the points are saved to the canvas and video feed. The implementation of this is in real time, making sure there are minimal delays between the movement of the nib and its corresponding strokes on the canvas. There is also a mask window within the application to better assist users in explaining how the pen nib is detected, giving them visual feedback as they adjust the settings.

1.3 Findings:

Air Canvas is indeed a wonderful demonstration of the concept that technology can turn everyday objects into creative tools. That digital drawing need no longer be associated with pricey, sophisticated hardware, a tool anyone can enjoy doing and learning from, means a lot if used for anything from fun to educational purposes up to creative projects. The Air Canvas creates a dynamic and entertaining user experience because it allows the tracking of movements, the offering of multiple colors, and effortless resetting of the canvas. This project distinguishes how innovative

programming can empower users to discover creativity in a digital environment by transforming a simple webcam into a versatile art tool.

Chapter:2

Tools and Technology

Technologies*:*

The following are the technology use in air canvas working.

OpenCV:

OpenCV is an integral part of the project Air Canvas, as it is used for real-time computer vision tasks. It will track and detect dark nib of any object movements, allowing for accurate gesture recognition to occur. OpenCV's library of extensive image processing functions makes it suitable for contour detection, color space transformations, and object tracking among other aspects. opencv is the most important library in the air canvas project almost all functionalities is working with build in function of this library.

NumPy:

It uses for numerical operations and efficient processing of array. It simplifies image handling since it is an array of multi-dimensional data. Its mathematical functions serve a crucial part in making some complex computations to quickly enhance its algorithms in the application of Air Canvas. numpy handle the array and it’s working smoothly. And make the work of air canvas excellent.

The following is the tool I used for air canvas working.

Visual Studio Code (VS Code):

The project is completed on Visual Studio Code as its development environment. Known for its lightweight nature and extent in personalization, VS Code, through its features such as debugging, syntax highlighting, and Python extension support, provides an easy way to improve the efficiency of one's development cycle.

Laptop detail:

This project is made on MacBook (Retina, 12-inch, Early 2015).processor 1.1 GHz Dual-Core Intel Core M, memory 8 GB 1600 MHz DDR3,graphics Intel HD Graphics 5300 1536 MB.total line of space taken by this project is 158, and memory is (0.88MB) per frame.

Chapter:3

Code of project

Line of code

```
from collections import deque #from this there will be stack in which we can add element from both side and also remove it from both side
import cv2
import numpy as np

def values(x):
pass

cv2.namedWindow("color detectors")#the following is perdefine perameter for a plot to detect colour
cv2.createTrackbar("upper hue","color detectors",153,180,values)
cv2.createTrackbar("upper saturation","color detectors",255,255,values)
cv2.createTrackbar("upper value","color detectors",255,255,values)
cv2.createTrackbar("lower hue","color detectors",64,180,values)
cv2.createTrackbar("lower saturation","color detectors",72,255,values)
cv2.createTrackbar("lower value","color detectors",49,255,values)
#as we use deque to store elements,now making array to handle colour
blue_points=[deque(maxlen=1024)]
green_points=[deque(maxlen=1024)]
red_points=[deque(maxlen=1024)]
yellow_points=[deque(maxlen=1024)]

#index points for colour
blue_index=0
green_index=0
red_index=0
yellow_index=0
```

```
#to use kernel ,small matrix for blurring, sharpening, embossing, edge detection.
kernel=np.ones((5,5),np.uint8)

colors=[(255,0,0),(0, 255, 0), (0, 0, 255), (0, 255, 255)]

color_index=0

#to start canvas

paintWindow=np.zeros((471,636,3)) +255
paintWindow=cv2.rectangle(paintWindow,(40,1),(140,65),(0,0,0),2)
paintWindow=cv2.rectangle(paintWindow,(160,1),(255,65),colors[0],-1)
paintWindow=cv2.rectangle(paintWindow,(275,1),(370,65),colors[1],-1)
paintWindow=cv2.rectangle(paintWindow,(390,1),(485,65),colors[2],-1)
paintWindow=cv2.rectangle(paintWindow,(505,1),(600,65),colors[3],-1)

cv2.putText(paintWindow,"erase all",
(49,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,(0,0,0),2,cv2.LINE_AA)
cv2.putText(paintWindow,"blue",(185,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,
(0,0,0),2,cv2.LINE_AA)
cv2.putText(paintWindow,"green",
(298,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,(0,0,0),2,cv2.LINE_AA)
cv2.putText(paintWindow,"red",(420,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,
(0,0,0),2,cv2.LINE_AA)
cv2.putText(paintWindow,"yellow",
(520,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,(0,0,0),2,cv2.LINE_AA)
cv2.namedWindow("paint",cv2.WINDOW_AUTOSIZE)

#connect to camera

cap=cv2.VideoCapture(0)

#making the loop for camaer to work untill user finished his/her work
while True:
ret , frame= cap.read()
frame=cv2.flip(frame,1)
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

```
u_hue=cv2.getTrackbarPos("upper hue","color detectors")
u_saturation=cv2.getTrackbarPos("upper saturation","color detectors")
u_value = cv2.getTrackbarPos("upper value", "color detectors")
l_hue = cv2.getTrackbarPos("lower hue", "color detectors")
l_saturation = cv2.getTrackbarPos("lower saturation", "color detectors")
l_value = cv2.getTrackbarPos("lower value", "color detectors")

upper_hsv=np.array([u_hue,u_saturation,u_value])
lower_hsv=np.array([l_hue,l_saturation,l_value])

#to add button on screen

frame=cv2.rectangle(frame,(40,1),(140,65),(122,122,122),-1)
frame=cv2.rectangle(frame,(160,1),(255,65),colors[0],-1)
frame=cv2.rectangle(frame,(275,1),(370,65),colors[1],-1)
frame=cv2.rectangle(frame,(390,1),(485,65),colors[2],-1)
frame=cv2.rectangle(frame,(505,1),(600,65),colors[3],-1)

cv2.putText(frame,"erase all",(49,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,
(0,0,0),2,cv2.LINE_AA)
cv2.putText(frame,"blue",(185,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,
(0,0,0),2,cv2.LINE_AA)
cv2.putText(frame,"green",(298,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,
(0,0,0),2,cv2.LINE_AA)
cv2.putText(frame,"red",(420,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,
(0,0,0),2,cv2.LINE_AA)
cv2.putText(frame,"yellow",(520,33),cv2.FONT_HERSHEY_TRIPLEX,0.5,
(0,0,0),2,cv2.LINE_AA)

#making a mask mean black mask layer to identifying the pointer
Mask = cv2.inRange(hsv, lower_hsv, upper_hsv)
Mask = cv2.erode(Mask, kernel, iterations=1)
Mask = cv2.morphologyEx(Mask, cv2.MORPH_OPEN, kernel)
Mask = cv2.dilate(Mask, kernel, iterations=1)

cnts,_=cv2.findContours(Mask.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPR
OX_SIMPLE)
center=None
```

```
if len(cnts)>0:
cnt=sorted(cnts,key=cv2.contourArea,reverse=True)[0]
((x,y),radius)=cv2.minEnclosingCircle(cnt)
cv2.circle(frame,(int(x),int(y)),int(radius),(0,255,255),2)
M=cv2.moments(cnt)
center=(int(M["m10"]/M["m00"]),int(M["m01"]/M["m00"]))

#to detect the button option
if center[1]<=65:
if 40<=center[0] <=140:#to clear
blue_points=[deque(maxlen=512)]
green_points=[deque(maxlen=512)]
red_points=[deque(maxlen=512)]
yellow_points=[deque(maxlen=512)]
blue_index=0
green_index=0
red_index=0
yellow_index=0
paintWindow[67:,:,:]=255
elif 160 <= center[0] <=255:
color_index=0
elif 275 <= center[0] <=370:
color_index=1
elif 390 <= center[0] <=485:
color_index=2
elif 505 <= center[0] <=600:
color_index=3

else:
if color_index ==0:
blue_points[blue_index].appendleft(center)
elif color_index==1:
green_points[green_index].appendleft(center)
elif color_index==2:
red_points[red_index].appendleft(center)
elif color_index==3:
yellow_points[yellow_index].appendleft(center)
```

```
else:
blue_points.append(deque(maxlen=512))
blue_index +=1
green_points.append(deque(maxlen=512))
green_index +=1
red_points.append(deque(maxlen=512))
red_index +=1
yellow_points.append(deque(maxlen=512))
yellow_index +=1

#now to draw anything
points=[blue_points,green_points,red_points,yellow_points]
for i in range(len(points)):
for j in range(len(points[i])):
for k in range(1,len(points[i][j])):
if points [i][j][k-1] is None or points[i][j][k]is None:
continue
cv2.line(frame,points[i][j][k-1],points[i][j][k],colors[i],2)
cv2.line(paintWindow,points[i][j][k-1],points[i][j][k],colors[i],2)
cv2.imshow("frame",frame)
cv2.imshow("paint window",paintWindow)
cv2.imshow("mask",Mask)
if cv2.waitKey(1 ) & 0xFF == ord('q'):
break
cap.release()
cv2.destroyAllWindows()
```

Chapter:4

Result of Code

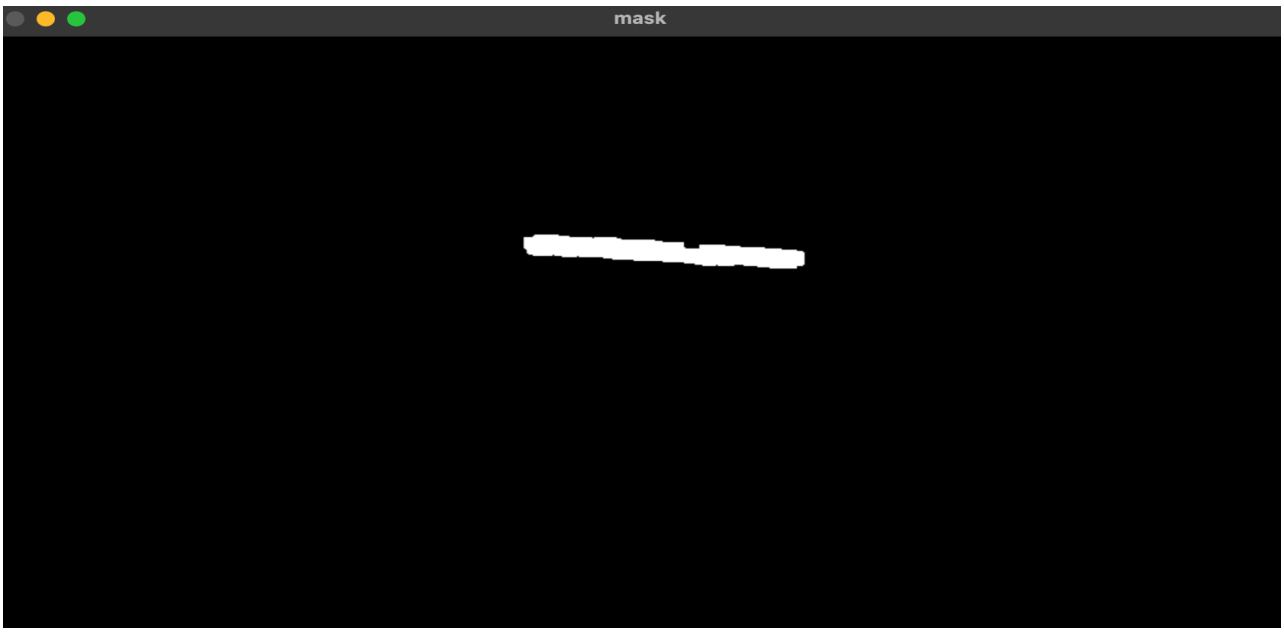

Figure 1: red color mask

In this figure 1 we can see the mask is detecting the pen to draw the things on canvas.

Figure 2: using blue color on air canvas

In this figure 2 we can see that blue using blue color we can draw any thing on air canvas.

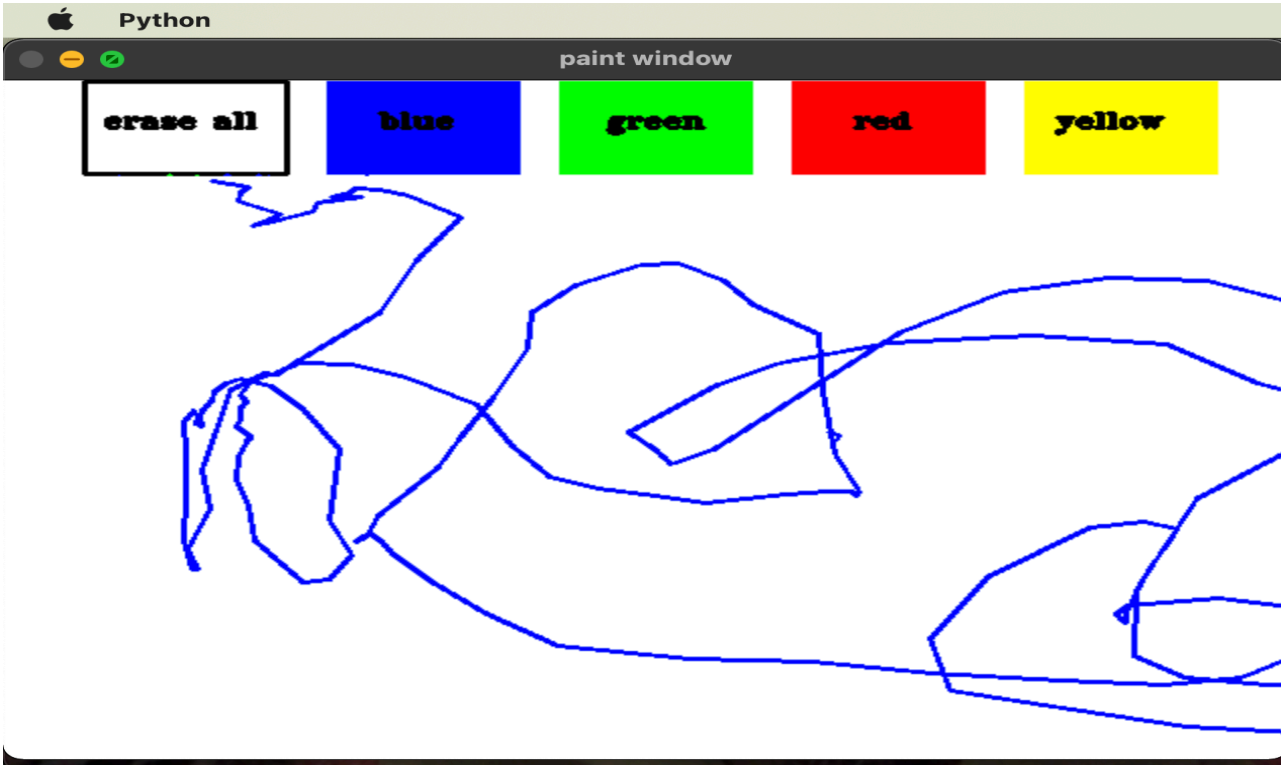

Figure 3: display work of blue color on canvas

In this a white canvas is used to show the drawing clearly, by using blue color we can do any thing but it depend upon the hand practices, so as we can see there are 3 ways screen to show the work, one is air canvas, one is mask and one is canvas.

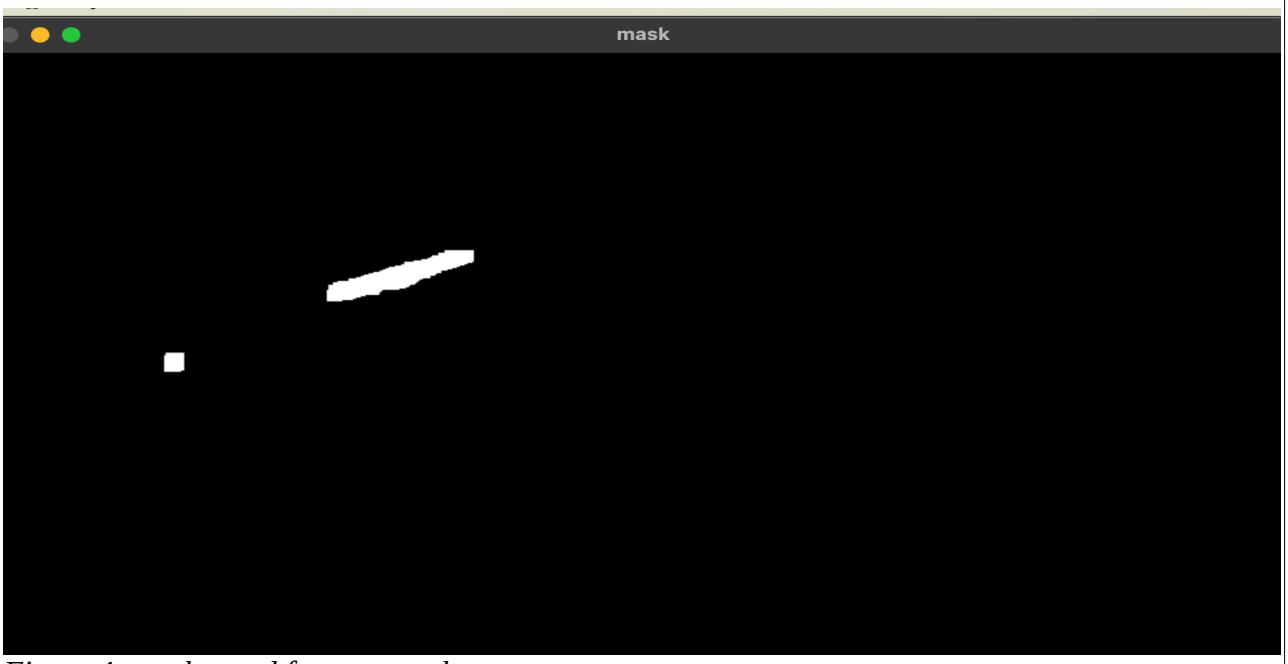

Figure 4: marks used for green color

Mask is used as same as in blue to draw things.

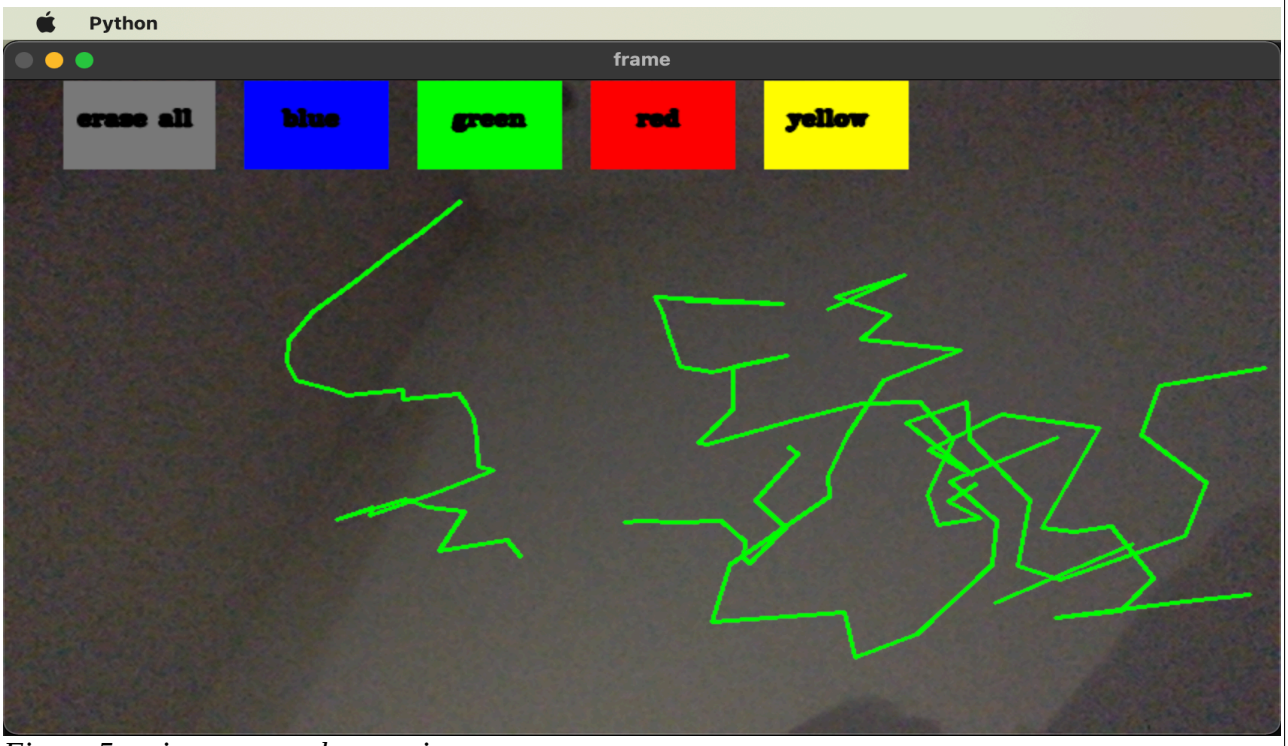

Figure 5: using green color on air canvas

using green color to draw thing same as blue.

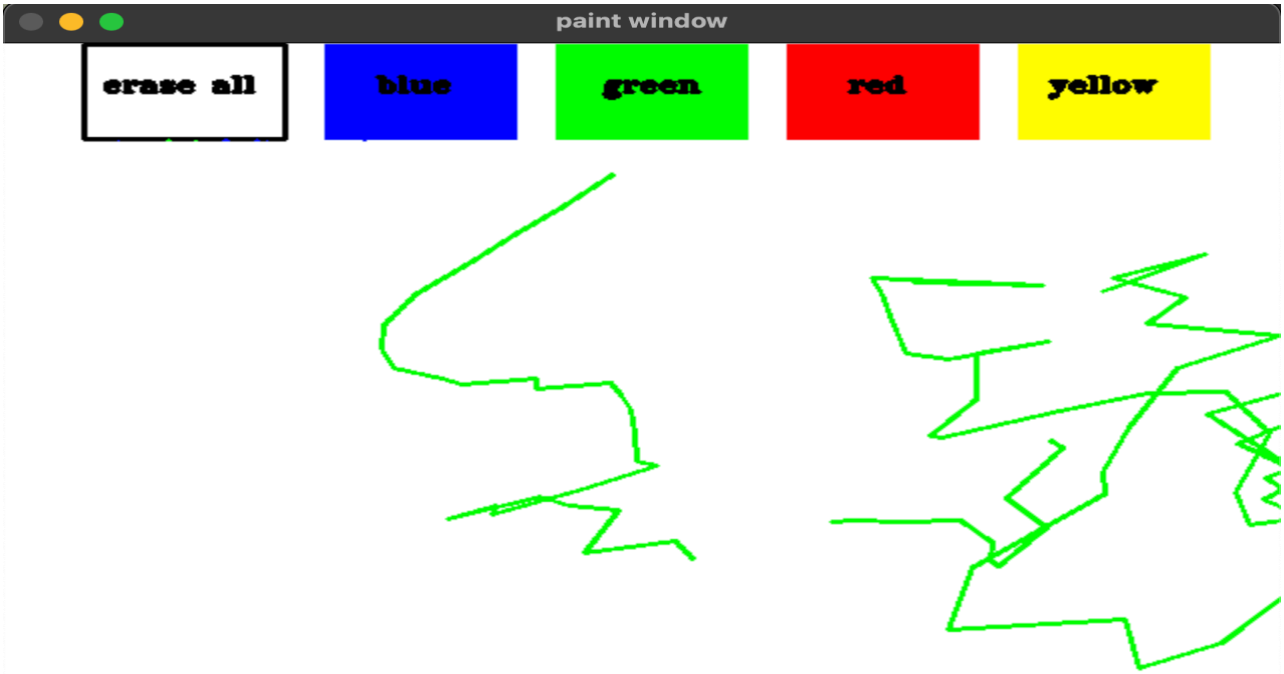

Figure 6: displaying work on canvas

canvas is used to show work same as blue.

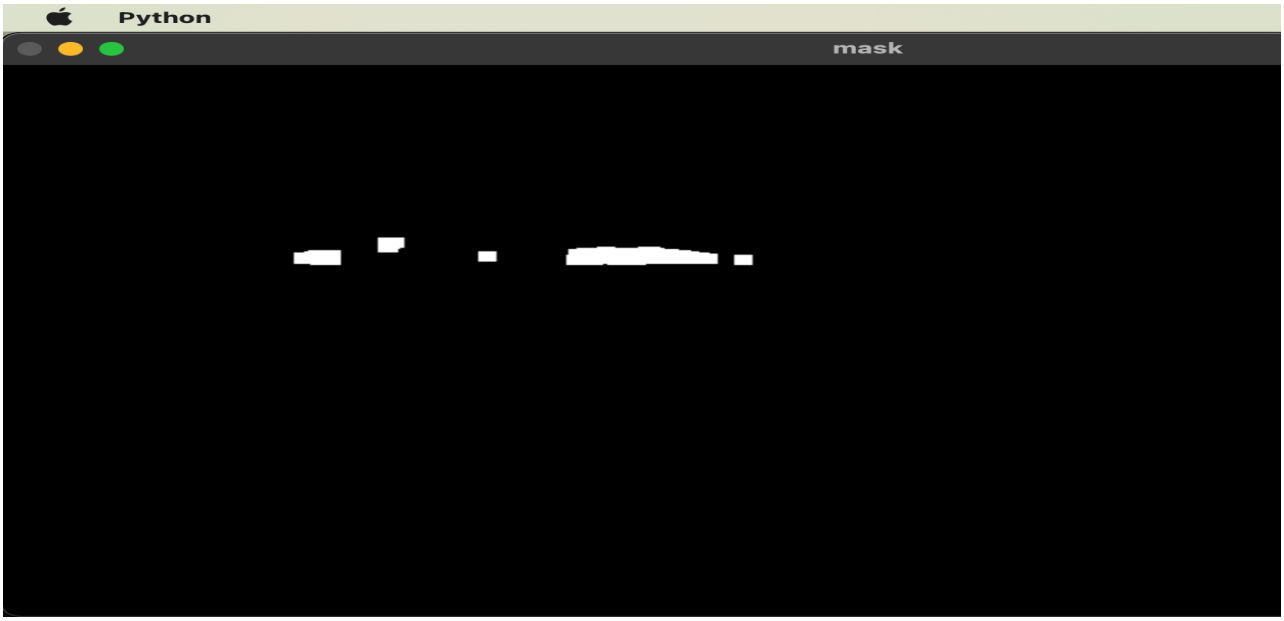

Figure 7: mask used for red pen

mask detecting pen to draw things to same as blue and green.

Figure 8: using red pen work on air canvas

By using red drawing things same as blue and green.

Figure 9: displaying work on canvas

showing work of red color on canvas, same as blue and green.

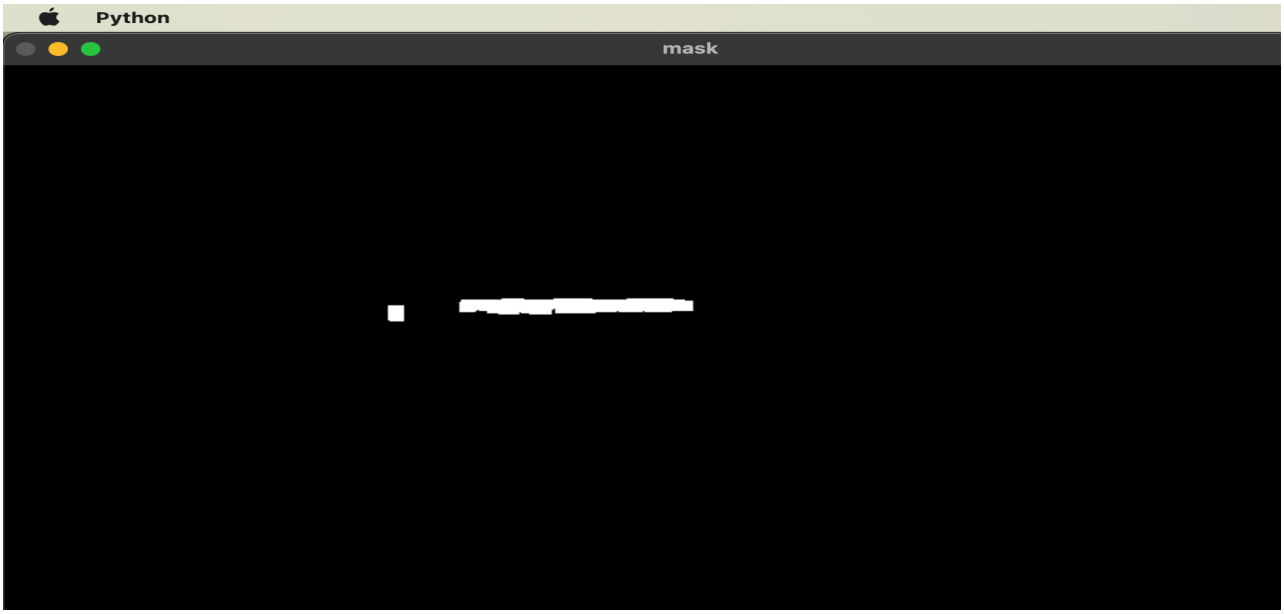

Figure 10: mask for yellow color

using pen drawing with yellow colour.

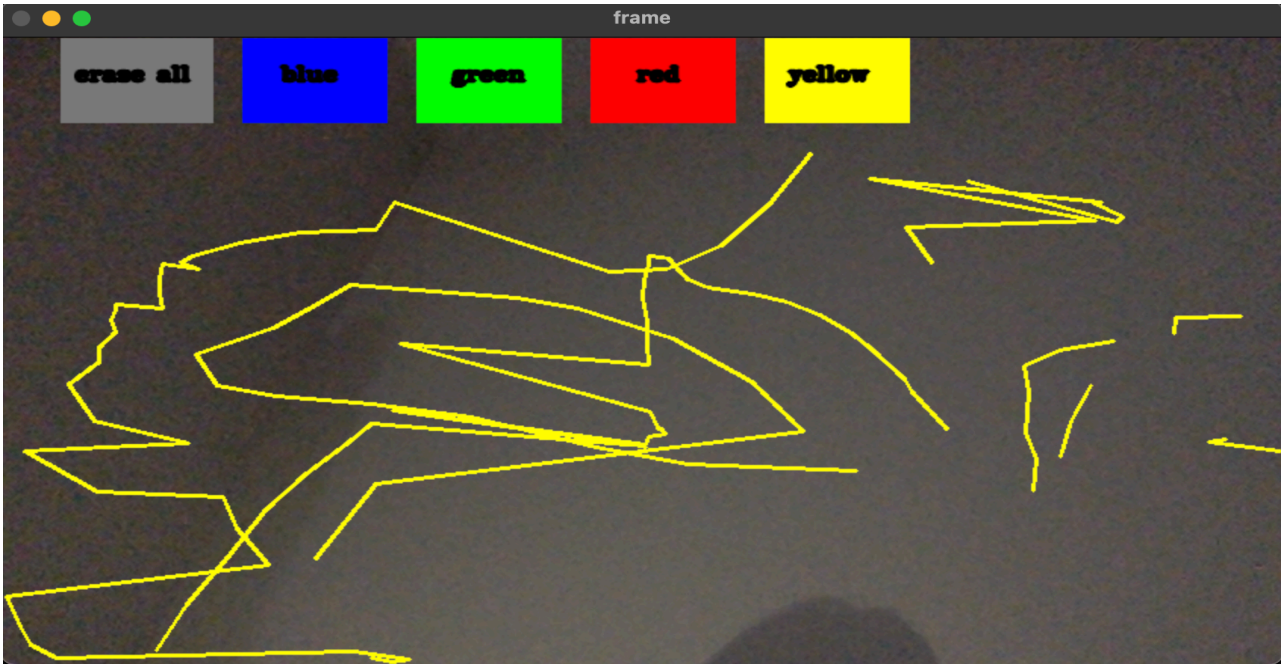

Figure 11: work on air canvas

work using yellow color on air canvas same as blue, green and red.

Figure 12: displaying work of yellow color

work on canvas using yellow color same as blue, green and red.

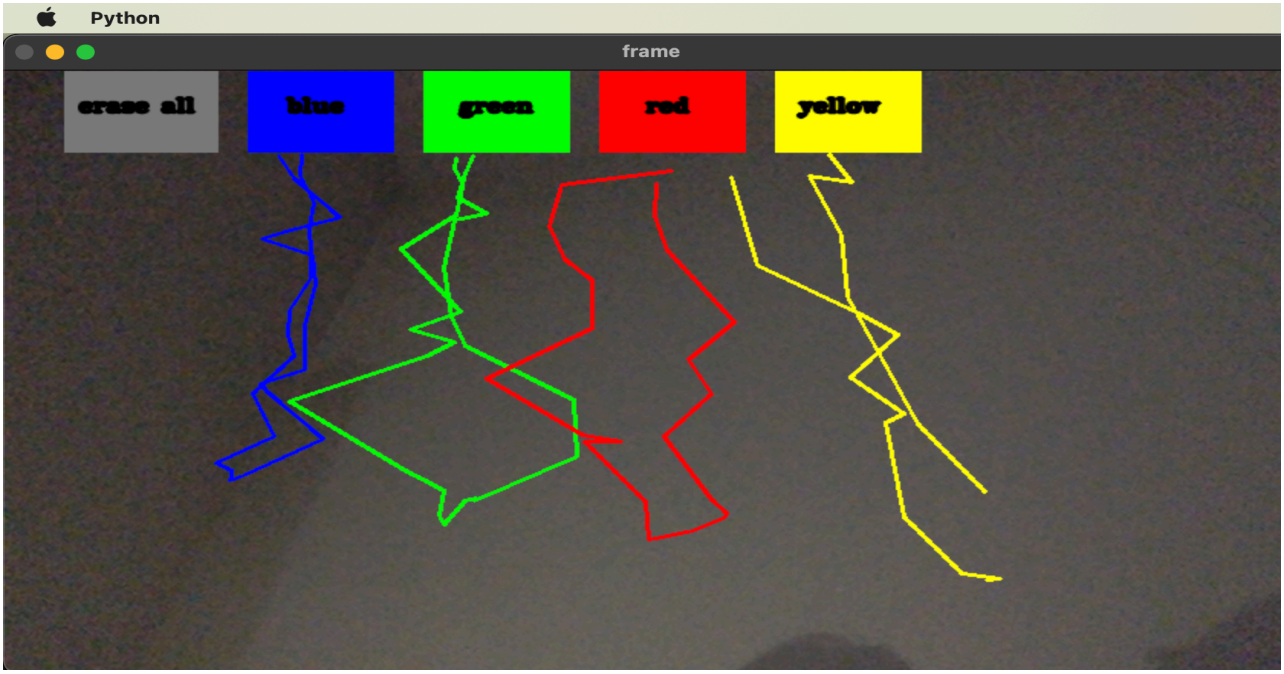

Figure 13: showing work of all colors

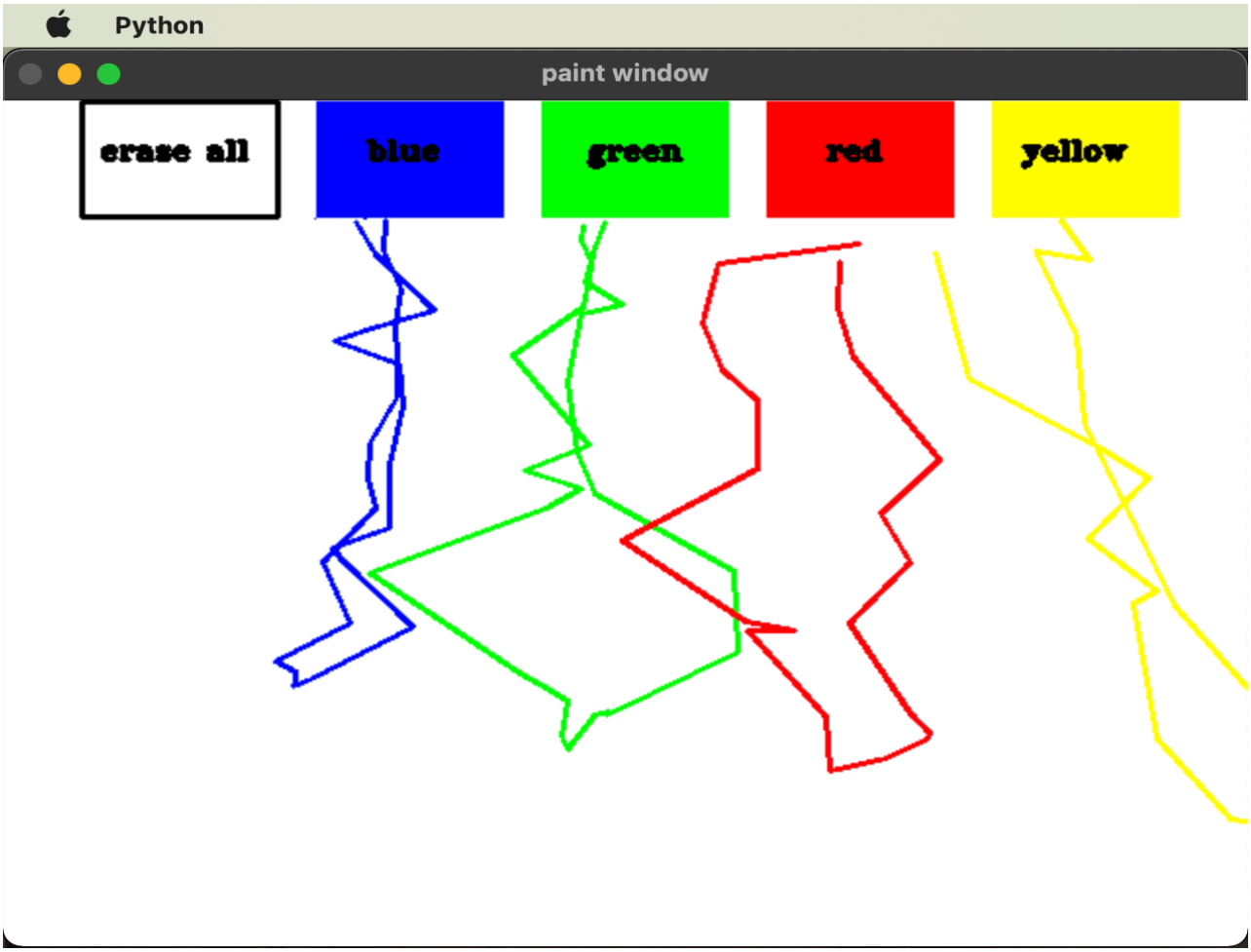

Figure 14: on canvas

As we can all the colors are working correctly.all things can be draw in air canvas. It is a fun technology to use. Without being bound to find things on paint canvas manually.

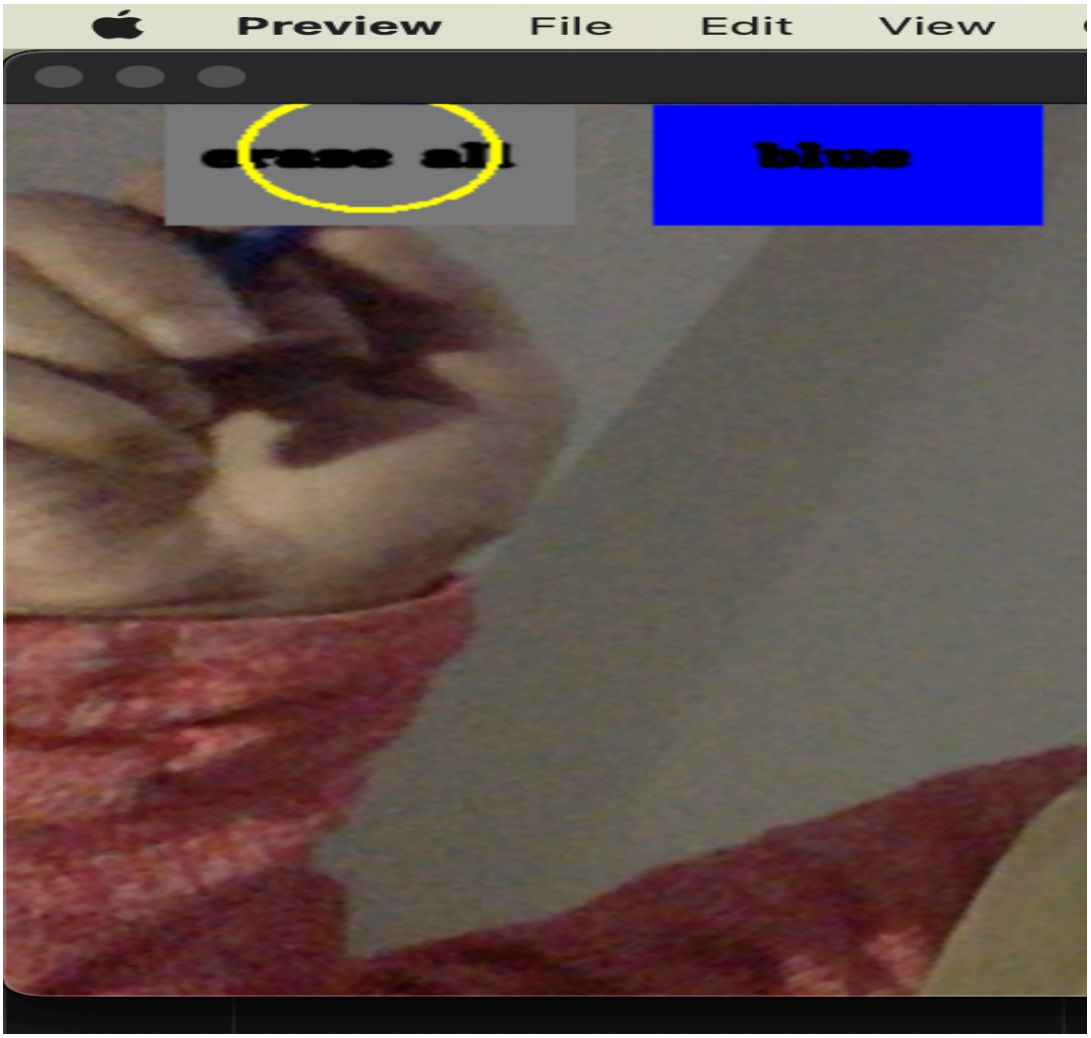

Figure 15: work of eraser

This erase option is used to clean the whole the work and drawing on cnavas.