A Project Report on

AIR CANVAS

Submitted to partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

AKANKSHA ANKAM	**18911A0506**
HARITHA RATHNAM	**18911A0523**
SAMHITHA REDDY	**18911A0537**
MEGHANA PUTTIGA	**18911A0598**

Under the Esteemed Guidance of

Mrs. Vijaya Lakshmi

Assistant Professor

Department of Computer Science & Engineering

VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Aziz Nagar Gate, C.B. Post, Hyderabad-500075

2021-22

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project report titled "**Air Canvas**" is being submitted by **Akanksha Ankam(18-506), Haritha Rathnam(18-523), Samhitha Reddy(18-537), Meghana Puttiga(18-598)** in partial fulfilment for the award of the Degree of Bachelor of Technology in **Computer Science & Engineering,** is a record of bonafide work carried out by him/her under my guidance and supervision. These results embodied in this project report have not been submitted to any other University or Institute for the award of any degree of diploma.

Internal Guide
Mrs. Vijaya Lakshmi
Asst. Professor

Head of Department
Dr. D.Aruna Kumari
Professor

External Examiner

DECLARATION

I, **Akanksha Ankam, Haritha Rathnam, Samhitha Reddy, Meghana Puttiga,** hereby declare that the project entitled, **"AIR CANVAS"** submitted for the degree of Bachelor of Technology in Computer Science and Engineering is original and has been done by me and this work is not copied and submitted anywhere for the award of any degree.

Date:

Place: HYDERABAD

AKANKSHA ANKAM(18-506)

HARITHA RATHNAM(18-523)

SAMHITHA REDDY(18-537)

MEGHANA PUTTIGA(18-598)

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to the project guide, **Mrs. Vijaya Lakshmi**, Asst. Professor, Vidya Jyothi Institute of Technology, Hyderabad for his/her timely cooperation and valuable suggestions while carrying out this work. It is his/her kindness that made me learn more from him.

I am grateful to **Dr. D.Aruna Kumari**, Professor and HOD, department of CSE, for her help and support during my academic year.

I whole-heartedly convey my gratitude to Principal **Dr. A. Padmaja** for her constructive encouragement.

I would like to take this opportunity to express my gratitude to our Director **Dr. E. Sai Baba Reddy** for providing necessary infrastructure to complete this project.

I would thank my parents and all the faculty members who have contributed to my progress through the course to come to this stage.

AKANKSHA ANKAM(18-506)

HARITHA RATHNAM(18-523)

SAMHITHA REDDY(18-537)

MEGHANA PUTTIGA(18-598)

ABSTRACT

Writing in air has been one of the most fascinating and challenging research areas in field of image processing and pattern recognition in the recent years. It contributes immensely to the advancement of an automation process and can improve the interface between man and machine in numerous applications.
Several research works have been focusing on new techniques and methods that would reduce the processing time while providing higher recognition accuracy. Object tracking is considered as an important task within the field of Computer Vision. The invention of faster computers, availability of inexpensive and good quality video cameras and demands of automated video analysis has given popularity to object tracking techniques.
Generally, video analysis procedure has three major steps: firstly, detecting of the object, secondly tracking its movement from frame to frame and lastly analysing the behaviour of that object.
For object tracking, four different issues are taken into account; selection of suitable object representation, feature selection for tracking, object detection and object tracking. In real world, Object tracking algorithms are the primarily part of different applications such as: automatic surveillance, video indexing vehicle navigation etc.
The project takes advantage of this gap and focuses on developing a motion-to-text converter that can potentially serve as software for intelligent wearable devices for writing from the air. This project is a reporter of occasional gestures. It will use computer vision to trace the path of the finger. The generated text can also be used for various purposes, such as sending messages, emails, etc. It will be a powerful means of communication for the deaf. It is an effective communication method that reduces mobile and laptop usage by eliminating the need to write

INDEX

CHAPTER-1

INTRODUCTION

Technology is constantly growing and changing our ways of living. It makes life easier at times, and more interesting too. When one would think of art in the past, usually computers and technology did not come to mind.
Now because of modern technology, the digital age is uncovering vast ways to create amazing works of art through computer tools and software.

Now a days everything is being digitalized, we now have digital classrooms and online education. We require certain tools to make this online education efficient, Air canvas can be on of that tool.

Air canvas is an application software which uses the technology of image processing and works as a digital canvas on which we can draw or write anything we want hands free, we just have to gesture anything that we want to draw or write in front of the webcam of our computer.

This functionality can also be achieved by using digital slate attached to your computer, but why use additional components if we can achieve it with what we have.

The basic goal of Air Canvas is to map the coordinates of the user's pointer finger to the screen, where coloured circles are drawn and connected to simulate a brush stroke.

User can choose among various colours and can undo everything with the help of clear option without even touching the computer, buttons are provided for every colour and user have to point it using the coloured object at the tip of his finger.

CHAPTER-2

LITERATURE SURVEY

2.1 Soft computing

Soft computing is the reverse of hard (conventional) computing. It refers to a group of computational techniques that are based on artificial intelligence(AI) and natural selection. It provides cost-effective solutions to the complex real-life problems for which hard computing solution does not exist.

Some characteristics of Soft computing

- Soft computing provides an approximate but precise solution for real-life problems.
- The algorithms of soft computing are adaptive, so the current process is not affected by any kind of change in the environment.
- The concept of soft computing is based on **learning from experimental data**. It means that soft computing does not require any mathematical model to solve the problem.
- Soft computing helps users to solve real-world problems by providing approximate results that conventional and analytical models cannot solve.
- It is based on Fuzzy logic, genetic algorithms, machine learning, ANN, and expert systems.

Fig -Soft computing process

Applications of soft computing

There are several applications of soft computing where it is used. Some of them are listed below:

- It is widely used in **gaming products like Poker and Checker**.
- In kitchen appliances, such as **Microwave and Rice cooker**.
- In most used home appliances - **Washing Machine, Heater, Refrigerator, and AC** as well.
- Apart from all these usages, it is also used in **Robotics work** (Emotional per Robot form).
- **Image processing and Data compression** are also popular applications of soft computing.
- Used for handwriting recognition.

As we already said that, soft computing provides the solution to real-time problems and here you can see that. Besides these applications, there are many other applications of soft computing.

.

Need of soft computing

Sometimes, conventional computing or analytical models does not provide a solution to some real-world problems. In that case, we require other technique like soft computing to obtain an approximate solution.

- Hard computing is used for solving mathematical problems that need a precise answer. It fails to provide solutions for some real-life problems. Thereby for real-life problems whose precise solution does not exist, soft computing helps
- When conventional mathematical and analytical models fail, soft computing helps, e.g., You can map even the human mind using soft computing.
- Analytical models can be used for solving mathematical problems and valid for ideal cases. But the real-world problems do not have an ideal case; these exist in a non-ideal environment.
- Soft computing is not only limited to theory; it also gives insights into real-life problems.
- Like all the above reasons, Soft computing helps to map the human mind, which cannot be possible with conventional mathematical and analytical models.

2.2 Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.
It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

For our application to work we need the access of webcam and to deal with the computer vision problems, so python has the libraries called OpenCV and numpy.

2.3 Computer Vision

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

2.4 OpenCV

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as Numpy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.
The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

CHAPTER-3

FEASIBILITY STUDY

Feasibility Study is associate degree assessment of the utility of a planned set up or technique. It gives an analysis of a proposed project to work out whether it's feasible and will plow ahead. This analysis is completed to urge a quick idea of the project. This study tells about the cost and various resources required for the system. The technologies are to be compared and accordingly the system has to be prepared.

The feasibility study are often analyzed in four ways-

1. Economic Feasibility
2. Technical Feasibility
3. Operational Feasibility
4. Schedule Feasibility

3.1 Economic Feasibility – This is often about the analysis. This study gives the highest management the economic justification for the new system. It gives the particular comparison of costs and benefits. This is often the point of reference to match actual cost because the project progresses.

3.2 Technical Feasibility – The difficult part is to analysis the technical issues. The project has many things to review just like the performance and therefore the output of various inputs. The text can't be detected if the image is blurred. If the image doesn't contain any text the system gives the output in several manners. The various technologies need to be studied and compared to development the program. The acceptable technology should be decided and worked upon to create the system.

3.3 Operational Feasibility – This analysis deals with control, efficiency and services. The project should have good performance and efficiency to detect correct text. The program should be easy to use and hospitable many users.

3.4 Schedule Feasibility – This study analysis about timeline estimations and optimizing resources. The appliance needs many software and tools.

CHAPTER-4

SYSTEM REQUIREMENTS

A Software Requirements Specification (SRS) – a requirements specification for a software is also an entire description of the behavior of a system to be developed. It includes a group of use cases that describe all the interactions the users will have with the software. Additionally, to use cases, the SRS also contains non-functional requirements. Nonfunctional requirements are those requirements which impose constraints on all the planning or implementing (such as performance engineering requirements, quality standards, or constraints of design).
is accountable for analyzing the business wants of their shoppers and stakeholders to help determine business issues and propose solutions. inside the systems development life cycle domain, the BA generally performs a liaison perform between the business facet of AN enterprise and thus the info technology department or external service suppliers.

Hardware Requirements:

- Operating system: Windows 7 or newer, 64-bit macOS 10.9+, or Linux.
- System architecture: 64-bit x86, 32-bit x86 with Windows or Linux.
- CPU: Intel Core 2 Quad CPU Q6600 @ 2.40GHz or greater.
- RAM: 4 GB or greater.

Software Requirements:

- Pycharm
- Python (3.8)
- Dlib, CMake

REQUIREMENTS DEFINITION:

After the severe continuous analysis of the problems that rose in the existing system, we are now familiar with the requirement that is required by the current system. The requirements that the system need is categorized into the functional and non-functional requirements. These requirements are listed below:

Functional Requirements:

The functional requirements describe the core functionality of the applying. Functional requirements are key for product features or functions which developers

must implement to enable users to accomplish their specific tasks. So, it's important to form them clear both for the event team and also the stakeholders. Generally, functional requirements describe system behavior under specific conditions

- Digital drawing will be difficult to execute, for that we would like to hold a digital slate everywhere
- Unlimited Possibilities. Limitless possibilities can result in creative paralysis.
- It are often printed on textured paper but it's not the identical as an ingenious painting.
- No Original Copy. No original physical copy. Too Easy. Some say the undo button makes digital art too easy but I feel that's a misconception.
- It still takes a good level of skill when it involves the basics of art to provide good digital work.
- More importantly, Digital drawing isn't that efficient. Easier to induce started and work quickly but it is an upscale approach.
- It is more forgiving and as nothing is permanent after you have, but cannot use the undo button efficiently.
- It also provides more chance of exploration with as many as unlimited experimental possibilities which can be wastage of your time.
- Easier duplication. Ideal for working with clients but needs to face some issues when it involves drawing simple objects.
- To improve many issues while drawing with a cursor, we've used OpenCV which is an open-source library and is freed from cost.
- As compared to other libraries, it's fast since it's written in C/C++.
- It works better on System with lesser **RAM**
- This also supports most of the Operating Systems like Windows, Linux and MacOS which helps all the users using different operating systems.

Interface Requirements

- User Interface: - The interface of this method may be a user-friendly Python Graphical Computer program.
- Hardware Interfaces: - The interaction between the user and therefore the console is achieved through python capabilities.
- Software Interfaces: - The specified software is PYTHON
- The color screen detector accepts the user inputs by the movement of the chosen pointer.
- The computer screen also shows the available different color options for the user to differentiate between the beads used till now.
- Operating Environment: - Windows XP, Linux.

Non-Functional Requirements:

In System engineering and Requirement of Engineering, (NFR) Non Functional requirements are requirements which specifies the factors which may be used for judging the operations of a system which are different from specific behavior. They're differ from functional requirement's which define specific behavior' or function's. System architecture Document contains all the small prints of non-functional requirements(NFR) plan's because of they are usually architecturally significant.

Definition:

Functional requirements are how the system must do whereas non-functional requirements are said to be how the system must be.
Non-functional requirements are also called as quality attributes of system however there's a difference between the 2 . Whenever we say a system should be secure, highly available, portable, scalable and so on, we are talking about its quality attributes.
These requirements describe operational qualities instead of behavioral qualities.

The NFR Qualities are divided into two major categories, they are

1. **Execution qualities**
   - Safety.
   - security
   - usability
   - observable things during operation

2. **Evolution qualities**
   - Testability
   - Maintainability
   - Extensibility
   - Scalability.

Brief about qualities:

Performance & scalability: -

- How fast does the system return result's?
- what proportion will this performance change with higher workloads?

Portability: -

- Which hardware, OS, browser's, and their version's does the software run on

Does it conflict with other applications and processes within these environments?
compatibility: -
Reliability, availability, maintainability. How often does system experience's critical-failure's and how much time does it available to user's against down-times?
Security: -

- How are system and its data protected against an attacks?

Localization: -

- Does the system match local specifics.

Usability: -

- How simple for a customer to use the system.

Performance: -

- defines how briskly a software system or its particular piece responds to certain users' actions under certain workload. In most cases, this metric's explain what quantity a user must wait before the target operation happens (the page renders, a transaction is processed, etc. given the overall number of users at the instant. But it's

not always like that. Performance requirements describe background processes invisible to user.

Scalability: -

- assesses the workloads under which the system will still meet the performance-requirements.

Other terms for non-functional requirements are

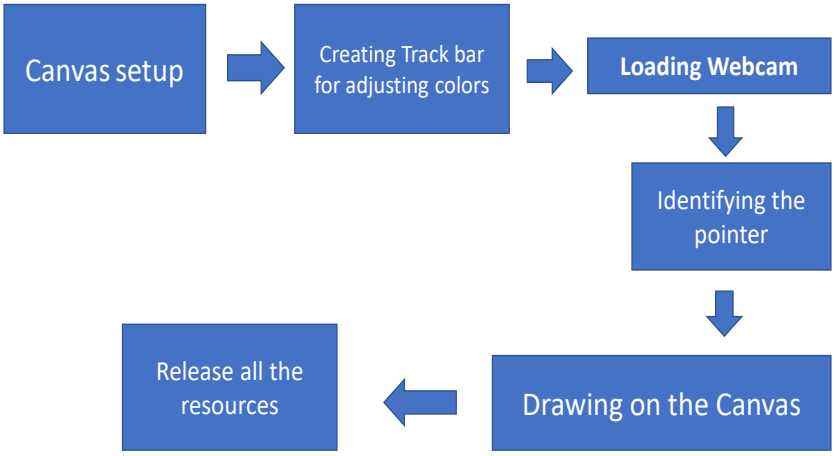
- qualities
- quality goals
- quality of service requirements
- constraints
- non-behavioral requirement

CHAPTER-5

SYSTEM DESIGN

SYSTEM ARCHITECTURE

The system architectural design is the design process for identifying the subsystems making up the system and framework for subsystem control and communication. The goal of the architectural design is to establish the overall structure of the software system.

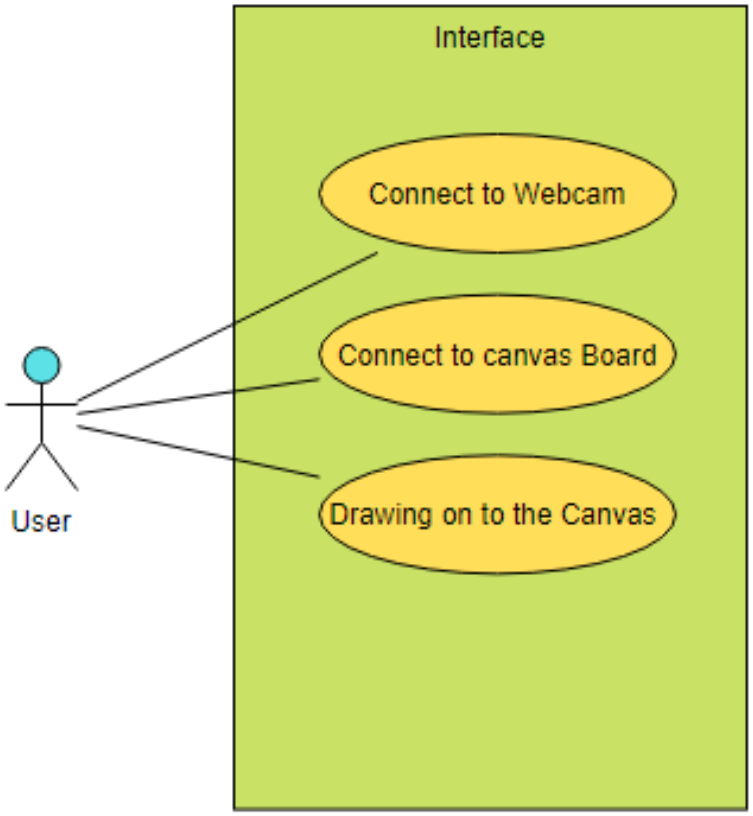


USECASE DIAGRAM

Use case diagrams are generally used to represent the overall scenario of the current system. A scenario is basically a sequence of steps describing an interaction between a user and the system.

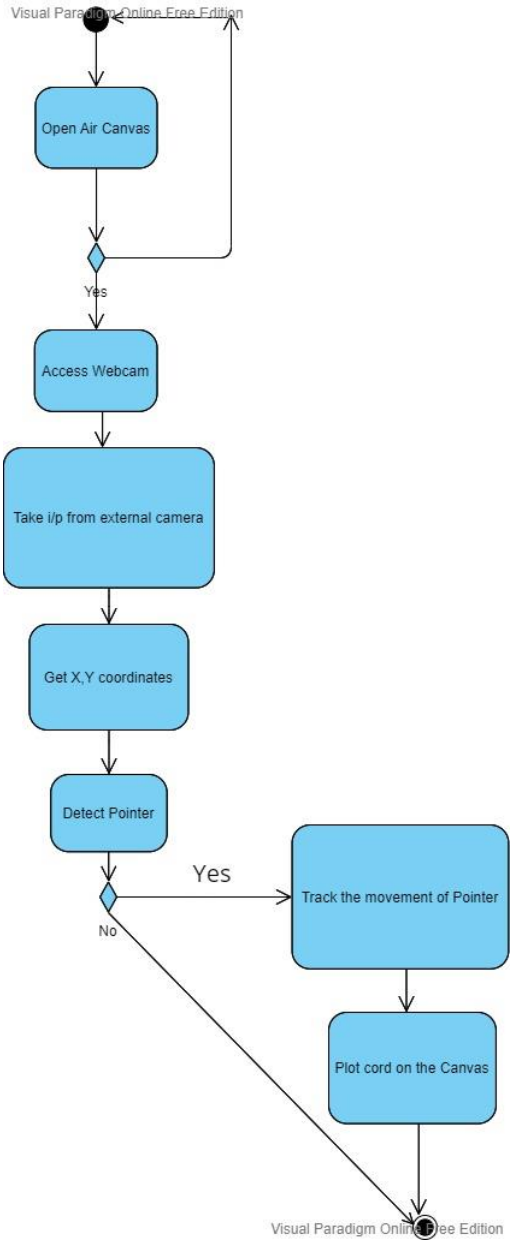
Thus a use case is a set of scenarios tied together by some goal. The use case diagrams are drawn for exposing the functionalities of the system.

The purpose of a use case diagram is to capture the dynamic aspect of a system. They provide a simplified graphical representation of what the system should do in a use case.



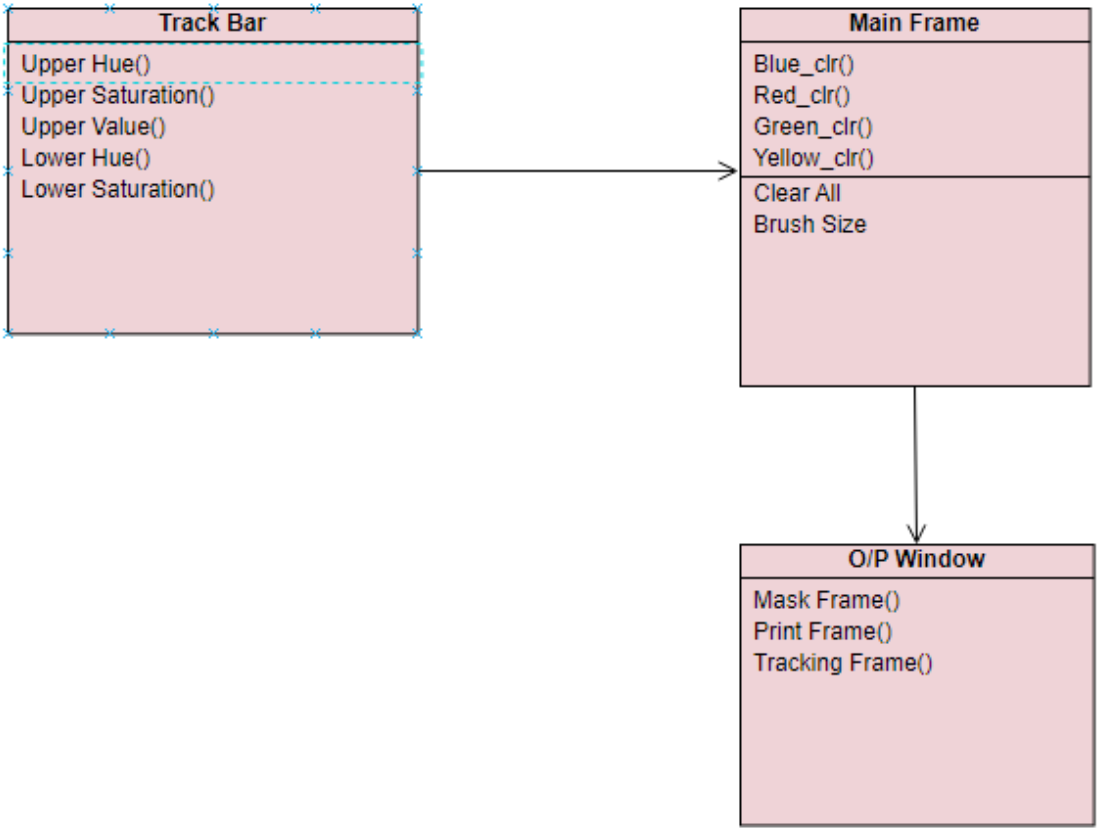
ACTIVITY DIAGRAM

The activity diagram is a graphical representation for representing the flow of interaction within specific scenarios. It is similar to a flowchart in which various activities that can be performed in the system are represented.



CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.


Track Bar
Upper Hue()
Upper Saturation()
Upper Value()
Lower Hue()
Lower Saturation()
Main Frame
Blue_clr()
Red_clr()
Green_clr()
Yellow_clr()
Clear All
Brush Size
O/P Window
Mask Frame()
Print Frame()
Tracking Frame()

CHAPTER-6

IMPLEMENTATION

The implementation of the project is done with the help of python language. To be particular, for the purpose of image processing and color tracking pycharm IDE is being used.

Pycharm is one of several Python distributions. Pycharm is an Integrated development environment, it is generally used for python language in specific. It is developed by Jet Brains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

On Python technology, we found out Pycharm to be easier. Since it helps with the following problems:

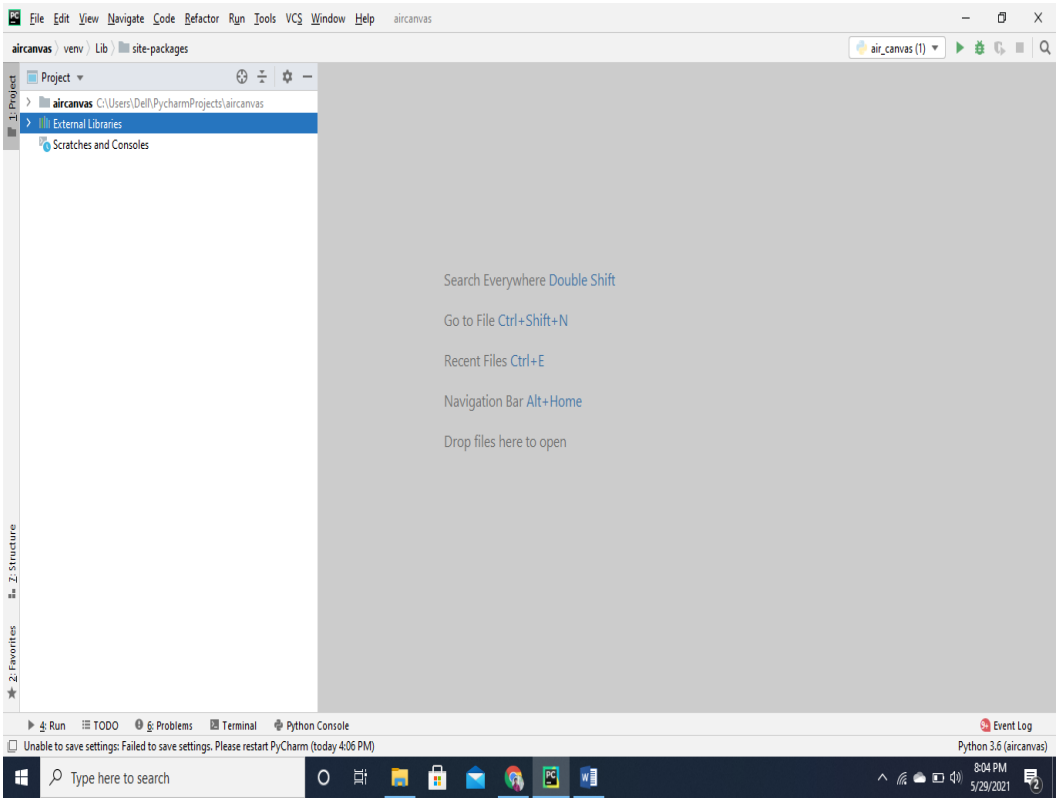
- Installing Python on multiple platforms.
- Inbuilt Graphical debugger
- An Integrated unit tester
- A lot of packages
- Separating out different environments.
- Dealing with not having correct privileges.
- Getting up and running with specific packages and libraries.

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as Numpy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

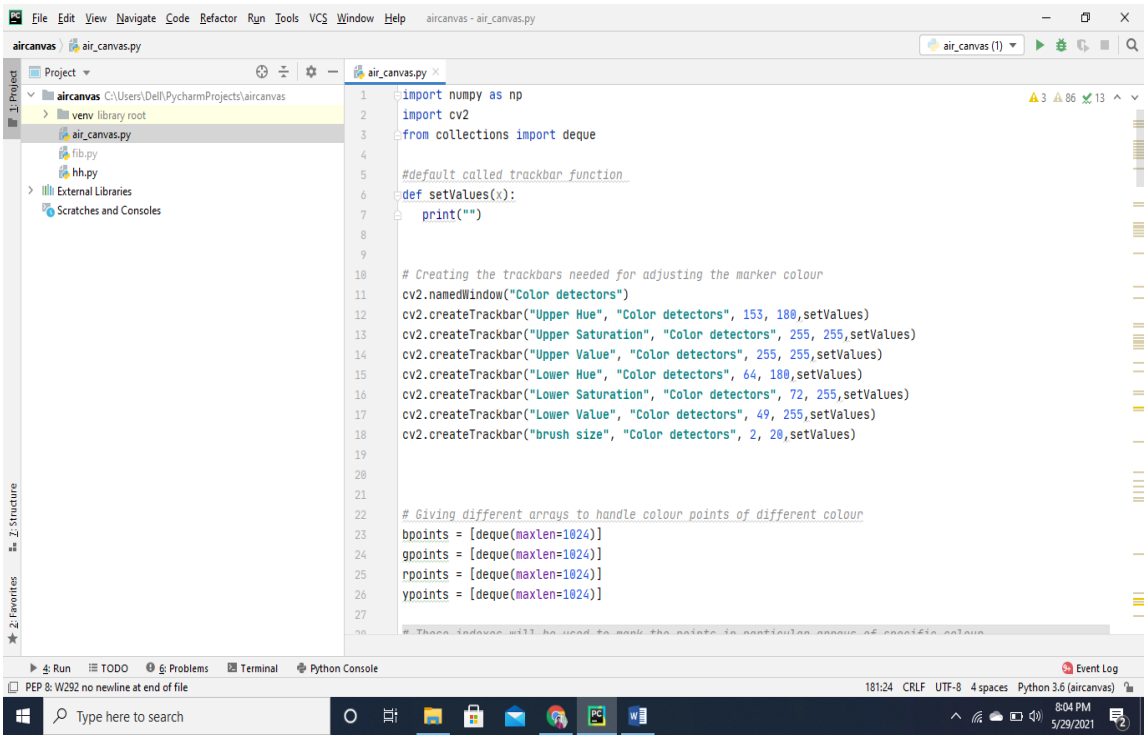
The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

Steps to run this project:

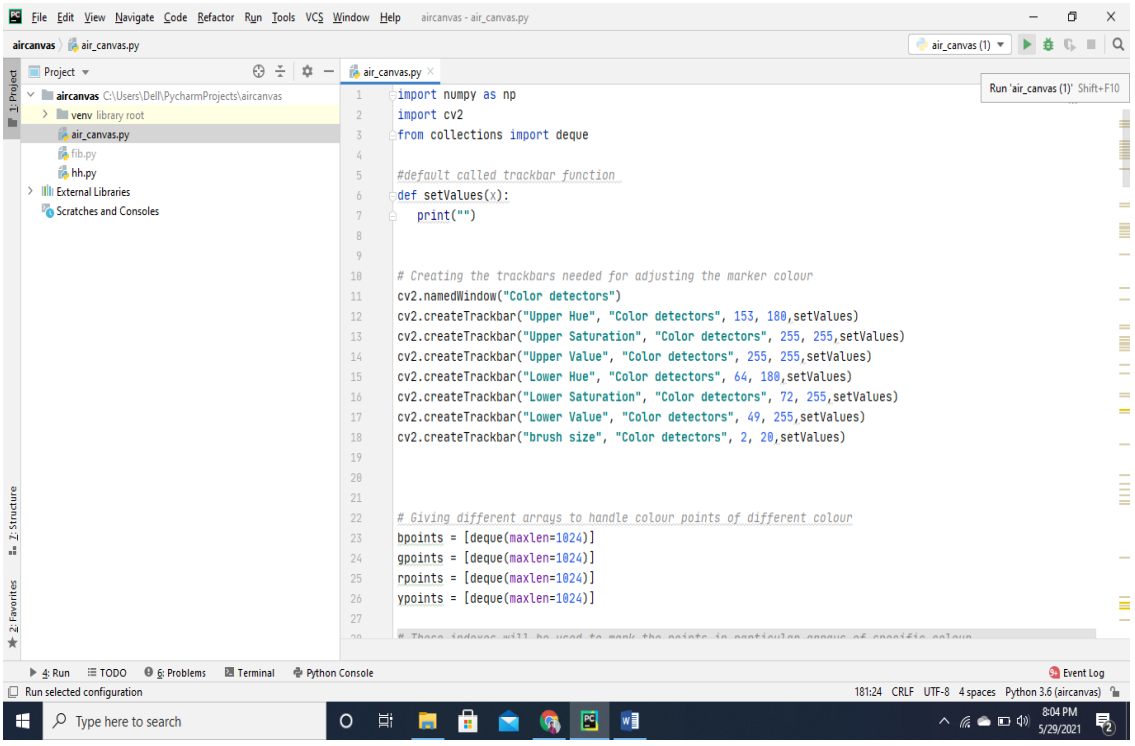
1. first open the pycharm IDE.



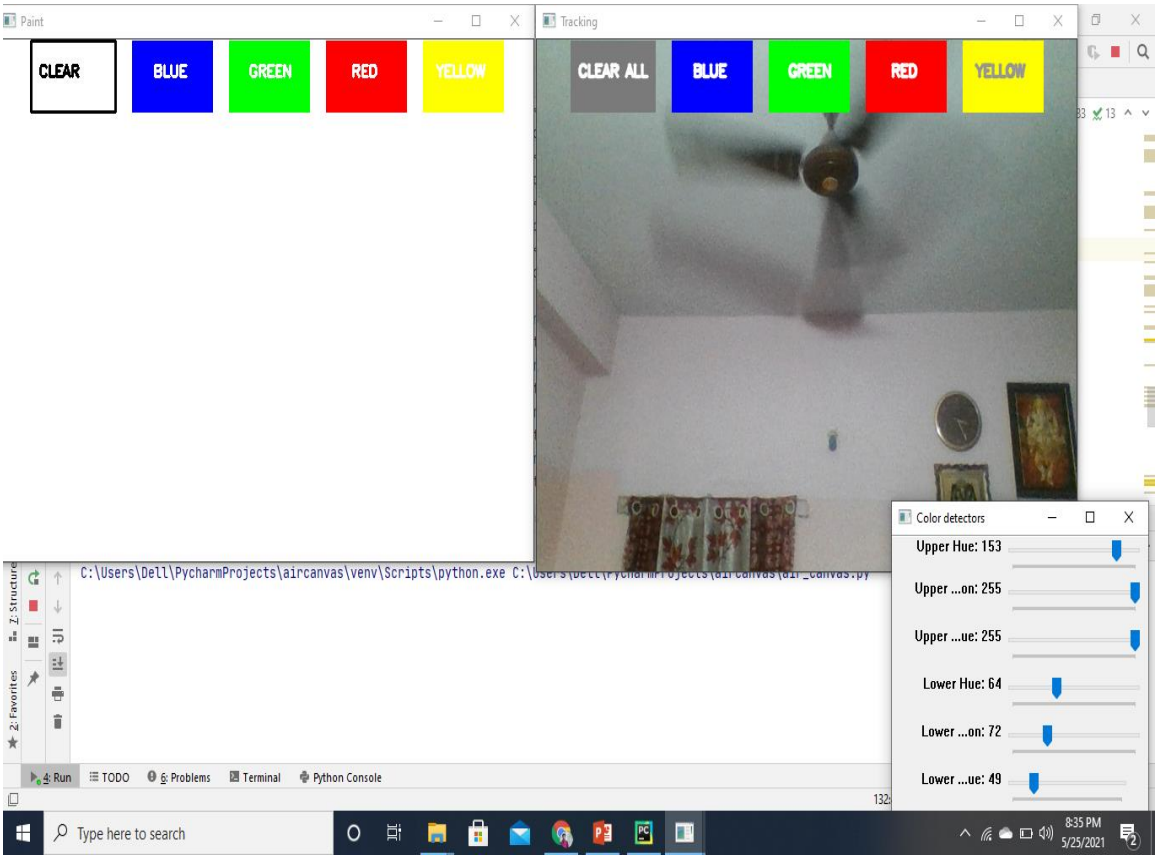
2. Open the file project and select on the file named “Air Canvas”.



3. Then on the right top corner, click on the RUN button on the right top corner.



3. Now there will be few windows displayed on the screen and the user can write or draw on to the screen.



Implementation Details:

For the purpose of implementation of the project image processing, object detection and color detection plays an important role.
The OpenCv library generally comes with inbuilt modules for image processing and object detection.
These modules are trained internally and comes with a large variety of applications. There are several functions in the modules which are embedded for different functionality. The OpenCv module is named as CV2.
We can import this module with the help of the import function as “import CV2”.
The basic functions involved in image processing are cvtColor, imread, imshow, canny, Blur, Gaussian Blur, erode, dilate etc.

cvtColor:

This function is generally used to convert the normal image into any other color. The syntax of function which is used to convert the color of image into another color is as follows.

```
grayImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

Imread:

This function is used to read the image or capture the live video from the external camera.

```
image = cv2.imread('/content/Screenshot (14).png')
```

Imshow:

This function is used to show the output to the user.

```
cv2_imshow(grayImage)
```

Canny:

This function is used for the purpose of edge detection. In edge detection, it gives an outline of the image by extracting edges from the image. We are using a canny filter to perform this task. Based on the threshold values, a canny filter detects the edges. The higher thresholds

give cleaner images compared to lower thresholds gives a clumsy output.

```
edge_det = cv2.Canny(img,100,200)
```

Smoothing techniques:

The image smoothing technique is performed using a filter. By convolving the image, it reduces the noise in the image by adding a blurring effect on the edges. There are different types of smoothing techniques we perform depending on the input image.

Blur:

In this technique, depending on the size of the filter convolution is performed but it averages the pixels under the filter and assigns the value to the centre pixel under the filter.

```
blur = cv2.blur(img, (5,5))
cv2_imshow(blur)
```

under

CODE:

```
import numpy as np
import cv2
from collections import deque

#default called trackbar function
def setValues(x):
   print("")


# Creating the trackbars needed for adjusting the marker
colour
cv2.namedWindow("Color detectors")
cv2.createTrackbar("Upper Hue", "Color detectors", 153,
180,setValues)
cv2.createTrackbar("Upper Saturation", "Color detectors",
255, 255,setValues)
cv2.createTrackbar("Upper Value", "Color detectors", 255,
255,setValues)
cv2.createTrackbar("Lower Hue", "Color detectors", 64,
180,setValues)
cv2.createTrackbar("Lower Saturation", "Color detectors",
72, 255,setValues)
cv2.createTrackbar("Lower Value", "Color detectors", 49,
255,setValues)
cv2.createTrackbar("brush size", "Color detectors", 2,
20,setValues)



# Giving different arrays to handle colour points of
different colour
bpoints = [deque(maxlen=1024)]
gpoints = [deque(maxlen=1024)]
rpoints = [deque(maxlen=1024)]
ypoints = [deque(maxlen=1024)]



# These indexes will be used to mark the points in
particular arrays of specific colour
blue_index = 0
green_index = 0
red_index = 0
yellow_index = 0

#The kernel to be used for dilation purpose
kernel = np.ones((5,5),np.uint8)

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

```
# Here is code for Canvas setup
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, "CLEAR", (49, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0), 2, cv2.LINE_AA)
cv2.putText(paintWindow, "BLUE", (185, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
cv2.LINE_AA)
cv2.putText(paintWindow, "GREEN", (298, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
cv2.LINE_AA)
cv2.putText(paintWindow, "RED", (420, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
cv2.LINE_AA)
cv2.putText(paintWindow, "YELLOW", (520, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (150,150,150), 2,
cv2.LINE_AA)
cv2.namedWindow('Paint', cv2.WINDOW_AUTOSIZE)



# Loading the default webcam of PC.
cap = cv2.VideoCapture(0)

# Keep looping
while True:
    # Reading the frame from the camera
    ret, frame = cap.read()

    #Flipping the frame to see same side of yours
    frame = cv2.flip(frame, 1)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    #creating the trackbar
    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")
    bs = cv2.getTrackbarPos("brush size", "Color detectors")

    Upper_hsv =np.array([u_hue,u_saturation,u_value])
    Lower_hsv =np.array([l_hue,l_saturation,l_value])


# Adding the colour buttons to the live frame for colour
access
    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, "CLEAR ALL", (49, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
cv2.LINE_AA)
    cv2.putText(frame, "BLUE", (185, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
cv2.LINE_AA)
    cv2.putText(frame, "GREEN", (298, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
cv2.LINE_AA)
    cv2.putText(frame, "RED", (420, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
cv2.LINE_AA)
    cv2.putText(frame, "YELLOW", (520, 33),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (150,150,150), 2,
cv2.LINE_AA)


    # Identifying the pointer by making its mask
    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)

    # Find contours for the pointer after idetifying it
    cnts,_ = cv2.findContours(Mask.copy(),
cv2.RETR_EXTERNAL,
       cv2.CHAIN_APPROX_SIMPLE)
    center = None

    # Ifthe contours are formed
    if len(cnts) > 0:
```

```
       # sorting the contours to find biggest
        cnt = sorted(cnts, key = cv2.contourArea, reverse =
True)[0]
        # Get the radius of the enclosing circle around the
found contour
        ((x, y), radius) = cv2.minEnclosingCircle(cnt)
        # Draw the circle around the contour
        cv2.circle(frame, (int(x), int(y)), int(radius), (0,
255, 255), 2)
        # Calculating the center of the detected contour
        M = cv2.moments(cnt)
        center = (int(M['m10'] / M['m00']), int(M['m01'] /
M['m00']))

        # Now checking if the user wants to click on any
button above the screen
        if center[1] <= 65:
            if 40 <= center[0] <= 140: # Clear Button
                bpoints = [deque(maxlen=512)]
                gpoints = [deque(maxlen=512)]
                rpoints = [deque(maxlen=512)]
                ypoints = [deque(maxlen=512)]

                blue_index = 0
                green_index = 0
                red_index = 0
                yellow_index = 0

                paintWindow[67:,:,:] = 255

            elif 160 <= center[0] <= 255:
                    colorIndex = 0 # Blue
            elif 275 <= center[0] <= 370:
                    colorIndex = 1 # Green
            elif 390 <= center[0] <= 485:
                    colorIndex = 2 # Red
            elif 505 <= center[0] <= 600:
                    colorIndex = 3 # Yellow
        else :
            if colorIndex == 0:
                bpoints[blue_index].appendleft(center)
            elif colorIndex == 1:
                gpoints[green_index].appendleft(center)
            elif colorIndex == 2:
                rpoints[red_index].appendleft(center)
            elif colorIndex == 3:
                ypoints[yellow_index].appendleft(center)
    # Append the next deques when nothing is detected to
avois messing up
    else:
        bpoints.append(deque(maxlen=512))
        blue_index += 1
        gpoints.append(deque(maxlen=512))
```

```
            green_index += 1
            rpoints.append(deque(maxlen=512))
            red_index += 1
            ypoints.append(deque(maxlen=512))
            yellow_index += 1
        #print(bs,l_value)
        # Draw lines of all the colors on the canvas and frame
        points = [bpoints, gpoints, rpoints, ypoints]
        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], bs)
                    cv2.line(paintWindow, points[i][j][k - 1],
points[i][j][k], colors[i], bs)

        # Show all the windows
        cv2.imshow("Tracking", frame)
        cv2.imshow("Paint", paintWindow)
        cv2.imshow("mask",Mask)

           # If the 'q' key is pressed then stop the application
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break

# Release the camera and all resources
cap.release()
cv2.destroyAllWindows()
```

CHAPTER-7

TESTING

Software "testing can be stated as the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by it's design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary" cases.
The "process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy and usability. It mainly aims at measuring specification, functionality and performance of a software program" or application.
Software testing can be divided into two steps:
1. **Verification:** it "refers to the set of tasks that ensure that software correctly implements a specific function".
2. **Validation:** it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.
Verification: "Are we building the product right?"
Validation: "Are we building the right product?"

Software Testing can be broadly classified into two types:

1. **Manual Testing:** Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.
Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

2. **Automation Testing:** Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.
Apart from regression testing, automation testing is also used to test the application from load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing.

1.Black Box Testing: The technique of testing in which the tester doesn't have access to the source code of the software and is conducted at the software interface without concerning with the internal logical structure of the software is known as black box testing.

2. **White-Box Testing:** The technique of testing in which the tester is aware of the internal workings of the product, have access to its source code and is conducted by

making sure that all internal operations are performed according to the specifications is known as white box testing.

Types of Testing:-

1. Unit Testing:

It focuses on the smallest unit of software design. In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs.
Example:
a) In a program we are checking if loop, method or

function is working fine

b) Misunderstood or incorrect, arithmetic precedence.

c) Incorrect initialization

2. Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components is combined to produce output.
Integration testing is of four types: (i) Top-down (ii) Bottom-up (iii) Sandwich (iv) Big-Bang
Example
(a) Black Box testing: - It is used for validation.

In this we ignore internal working mechanism and

focuses on **what is the output?**

(b) White Box testing: - It is used for verification.

In this we focus on internal mechanism.

3. Regression Testing

Every time a new module is added leads to changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program.
Example
In school record suppose we have module staff, students and finance combining these modules and checking if on integration these module works fine is regression testing

4. Smoke Testing

This test is done to make sure that software under testing is ready or stable for further testing
It is called a smoke test as the testing an initial pass is done to check if it did not catch the fire or smoke in the initial switch on.

Example:
If project has 2 modules so before going to module
make sure that module 1 works properly

5. Alpha Testing

This is a type of validation testing. It is a type of *acceptance testing* which is done before the product is released to customers. It is typically done by QA people.
Example:
When software testing is performed internally within the organization

6. Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for a limited number of users for testing in a real-time environment
Example:
When software testing is performed for the limited
number of people

7. System Testing

This software is tested such that it works fine for the different operating systems. It is covered under the black box testing technique. In this, we just focus on the required input and output without focusing on internal working.
In this, we have security testing, recovery testing, stress testing, and performance testing
Example:
This include functional as well as non-functional
testing

8. Stress Testing

In this, we give unfavorable conditions to the system and check how they perform in those conditions.
Example:
(a) Test cases that require maximum memory or other
resources are executed

(b) Test cases that may cause thrashing in a virtual operating system

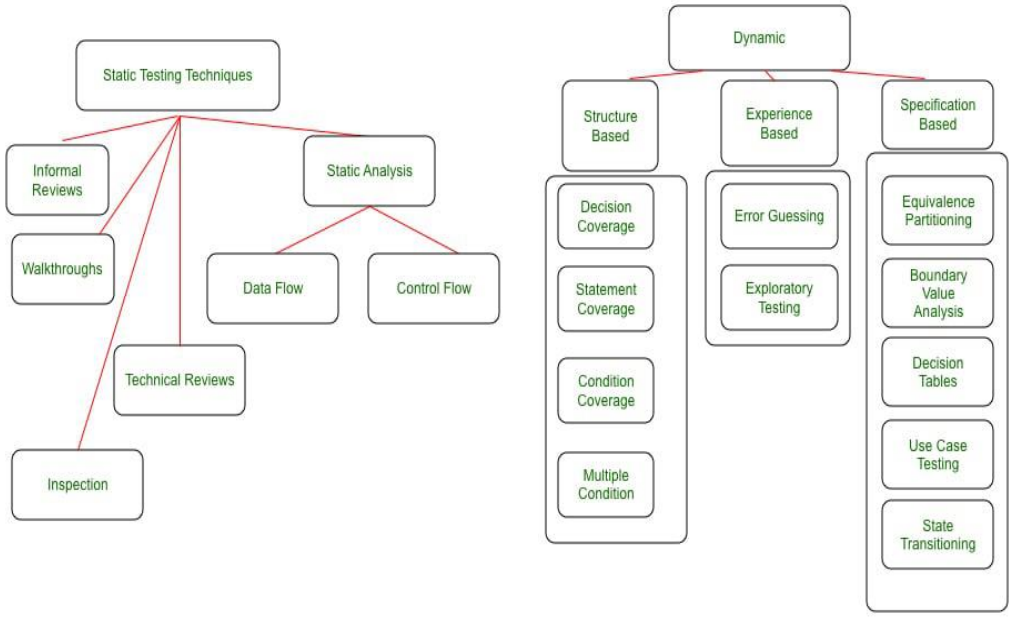
Principles Of Testing:

- Below are the principles of software testing:
- All the tests should meet the customer requirements.
- To make our software testing should be performed by a third party
- Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
- All the test to be conducted should be planned before implementing it
- It follows the Pareto rule (80/20 rule) which states that 80% of errors come from 20% of program components.
- Start testing with small parts and extend it to large parts.

Types Of Software Testing Techniques:

There are two main categories of software testing techniques:

1. **Static Testing Techniques** are testing techniques which are used to find defects in Application under test without executing the code. Static Testing is done to avoid errors at an early stage of the development cycle and thus reducing the cost of fixing them.
2. **Dynamic Testing Techniques** are testing techniques that are used to test the dynamic behavior of the application under test, that is by the execution of the code base. The main purpose of dynamic testing is to test the application with dynamic inputs- some of which may be allowed as per requirement (Positive testing) and some are not allowed (Negative Testing).



We did testing after completion of each module and several tests after integration of all the modules.

That is the efficient way of testing and finding the errors, if you test each and every part separately it will be easy to find the where the errors and can dealt with immediately.

If you directly move to integration testing without any unit testing, then it will become very hard to find the errors and bugs.

We have done several tests in different areas:

i. **Testing of trackbar**: We create track bar to track the specific color of the bead using HSV (Hue Saturation Value) space. We have to set the proper hsv value to track the specific color of the mask, if the value is not correspondent to the color of that mask then no tracking will take place.

ii. **Testing of camera**: We need to use a proper webcam for the proper function of the application, the camera should be able to distinguish among the different colours.

iii. **Canvas setup**: We need the proper setup of white sheet canvas on which everything we draw on air is tracked and projected.

iv. **Addition of color buttons:** We need to add no of rectangle boxes on the sheet which work as buttons, we need to test and give proper coordinates for the boxes, so as we move our mask towards the coordinates of the color, the color of the brush should change to that particular color.

v. **Frame reading**: We have setup the camera and the program should read every frame of the video and setup the lower and upper HSV values and add the rectangle boxes which work as buttons to every frame of video.

vi. **Testing of Identification of mask**: Identification of the mask is one of the crucial part. We need to apply proper morphological operations to remove the impurities around the mask, the two main operations are dilation and erosion, we did proper testing to find the appropriate HSV values and proper detection of the color of the mask.

vii. **Test for change of colours:** After identification of centre of the contour, then the coordinates of it is checked with the coordinates of the colour buttons and the matching coordinates of the colour leads to the change of color or else it just draws the lines on to the canvas.

Like this we performed several unit testing and after that we integrated all the modules and performed the integrated testing to conform the proper functioning of the tool.

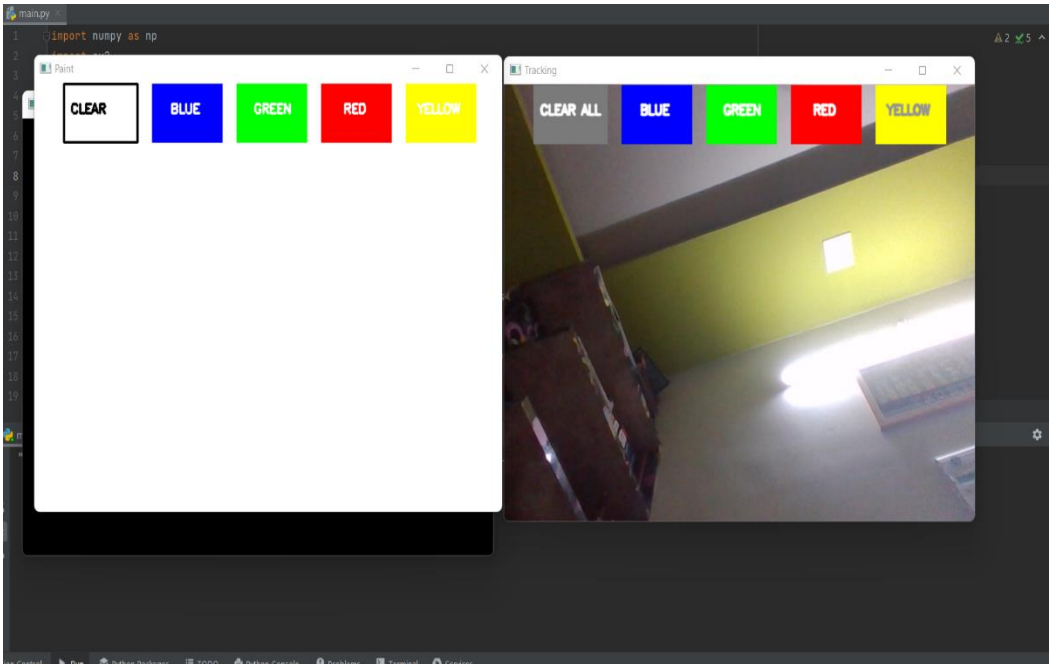
Test table:

Test cases and Output:

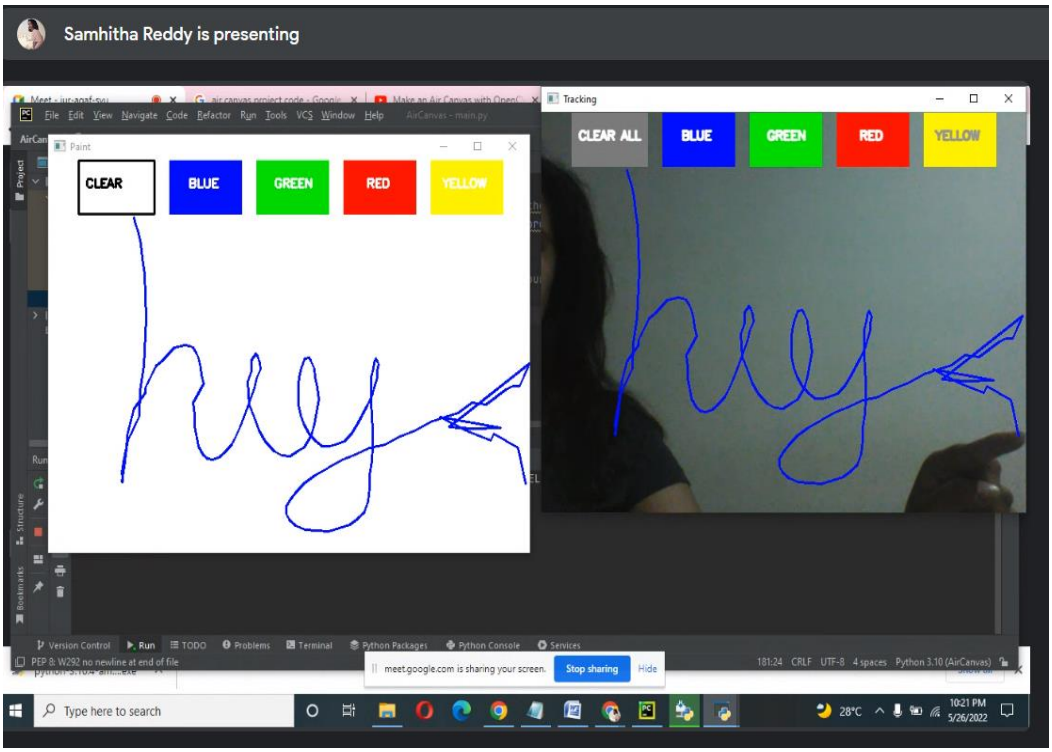
Functionality	Result
Drawing smoothly	pass
Selection of colors	pass
Erase the content	pass
Adjustment of Track bar	pass
Increasing of brush size	pass
Decreasing of brush size	pass
Identifying the pointer accurately	pass

CHAPTER-8
RESULTS AND OUTPUT SCREENS

OUTPUT SCREEN 1:



OUTPUT SCREEN 2:



CHAPTER-9
CONCLUSION

- Air canvas are very useful in drawing.
- It will be useful in teaching purposes, they should not have to carry a digital slate everywhere, they'll use this feature to put in writing anything and project just by writing within the air in front of the camera.
- This project will take the method of digital learning and digital drawing to its next step if implemented properly.
- This project will be considered jointly of the important applications of image processing, object tracking and color detection.

We will be able to consider that our project to be an overall success. With Air Canvas, we have achieved a hands-free drawing program that also uses OpenCV to detect the user's pointer finger. Colorful lines will be drawn wherever the user desires and therefore the brush can even be modified. It's truly like drawing within the air, isn't it.Yes, Air Canvas really has many flaws that might also be interesting divisions of research within the future. The primary is the issue of frame rate: image processing stalled the camera feed and produced a cumbersome lag that impedes on the usability of the program. It'd be best optimized with multicore functionality, which we attempted during this project. If the timing problems with queueing data between processes may be managed such that specified frame information is passed so as, perhaps Air Canvas is upgraded to run authentically in real time. Moreover, we relied on open source OpenCV code for hand recognition, which had its own issues that we worked hard to avoid.

Artificial Intelligence and also the technology that is being used are one side of the life that always gives interest and amazes us with the new ideas, features, innovations, products, schemas etc. AI continues to be not implemented because the films representing it(i.e. intelligent robots), however there are many important tries to achieve the amount and to compete in market, like sometimes the robots that they show in TV. Nevertheless, the hidden projects and also the development in industrial companies.

At the end, we've been during this research through the AI definitions, brief history, applications of AI publicly, applications of AI in military, ethics of AI, and therefore the three rules of robotics. this is often not the tip of AI, there's more to come back from it, who knows what the AI can do for us within

the future, maybe it'll be an entire society of robots.The initial motivation was a requirement for a dustless classroom for the scholars to check in. i do know that there are some ways like touch screens and more but what about the faculties which can’t afford it to shop for such huge, large screens and teach on them sort of a T.V. So, i believed why not can a finger be tracked, but that too at a initial level without deep learning. Hence it absolutely was OpenCV which came to the rescue for these computer vision projects..

We will be using the pc vision techniques of OpenCV to create this project. the popular language is Python thanks to its exhaustive libraries and simple to use syntax but understanding the fundamentals it may be implemented in any OpenCV supported language. Here Color Detection and tracking are employed in order to attain the target. the color marker is detected, and a mask is produced. It also includes the further steps for all morphological operations on the mask produced which are found to be Erosion and Dilation. Erosion reduces the impurities present within the mask and dilation further restores the eroded main mask.

CHAPTER-10

REFERENCES

- Finger Detection by Izane: https://github.com/lzane/Fingers-Detection-using-OpenCV-and-Python
- Finger Detection and Tracking by Amesh Prakash Pandey: https://dev.to/amarlearning/finger-detection-and-tracking-using-opencv-and-python-586m
- OpenCV contours: https://docs.opencv.org/3.4.2/d4/d73/tutorial_py_contours_begin.html
- AutoTurret by : https://courses.ece.cornell.edu/ece5990/ECE5725_Spring2018_Projects/fy57_xz522_AutoTurret/index.html
- "Guide to Set Up Pi Camera & Some OpenCV Image Processing Basics" by Xitang Zhao: https://blackboard.cornell.edu/bbcswebdav/pid-4087974-dt-content-rid-24070270_1/courses/12153_2019SP/12153_2019SP_ImportedContent_20190408043030/Pi_camera_and_OpenCV_v1%281%29.pdf
- Create aircanvas using python and openCV by gfg: https://www.google.com/amp/s/www.geeksforgeeks.org/create-air-canvas-using-python-opencv/amp/
- Air canvas project by bgithub: https://github.com/infoaryan/Air-Canvas-project
- http://infoaryan.com/blog/air-canvas-computer-vision-project/
- https://www.pantechsolutions.net/air-canvas-using-opencv-and-python
- https://www.codespeedy.com/create-an-air-canvas-using-opencv-python/