A

Project Report on

"Draw In Air - Air Canvas Application using Opencv and Mediapipe"

Submitted in Partial Fulfilment of the Requirements

for the Degree of

Bachelor of Technology

in

Computer Science and Information Technology

by

Mr. Sanket Bodhale 1910029

Ms. Chinmai Arekar 2010026

Ms. Sejal Patil 2010028

Mr. Anshul Bhimnath 2010046

Under The Guidance of

Prof. Savita Patil

Department of Information Technology

K.E. Society's

Rajarambapu Institute of Technology, Rajaramnagar

(An Empowered Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

2023-2024

DECLARATION

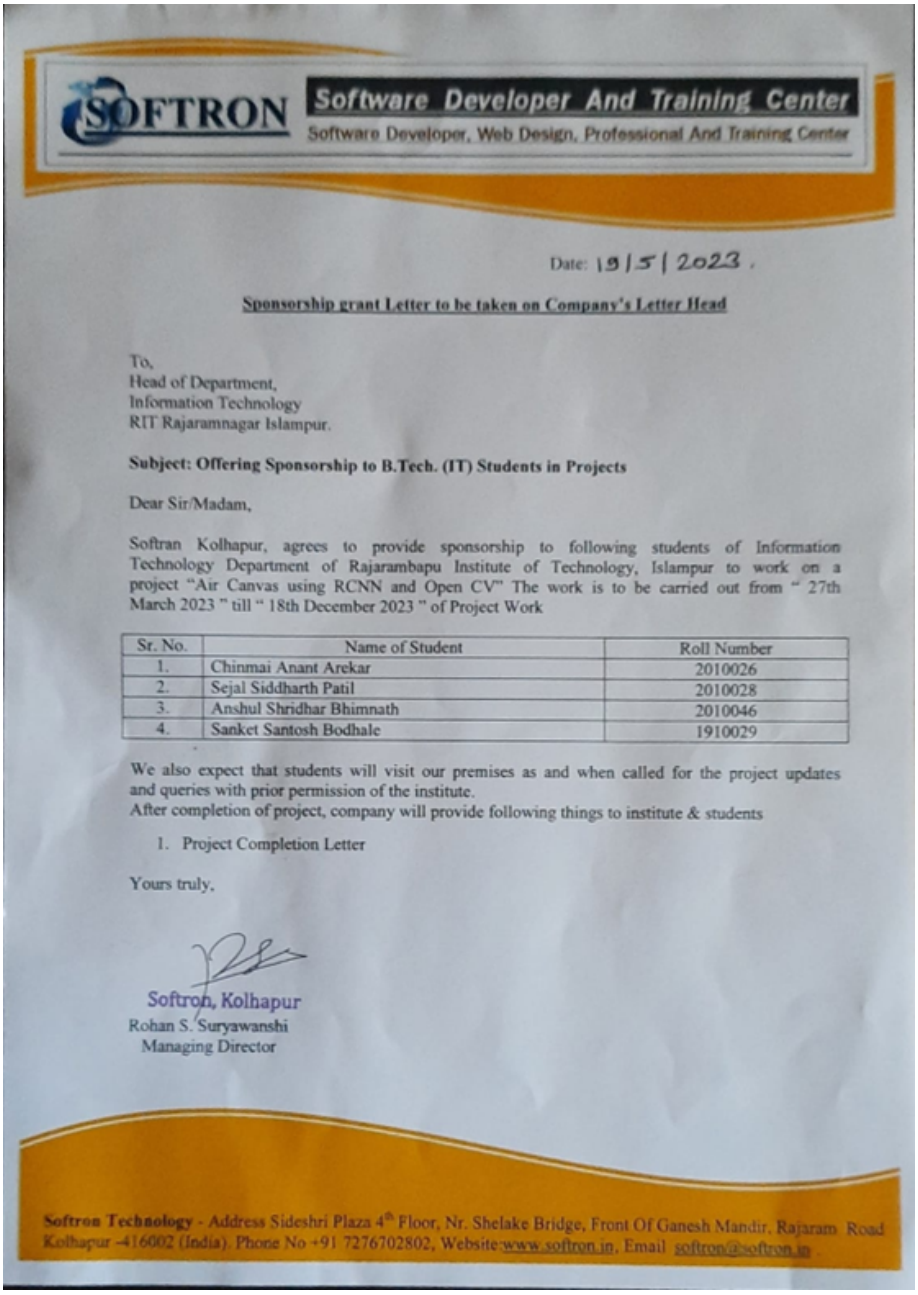
We declare that this report reflects our thoughts about the subject in our own words. We have sufficiently cited and referenced the original sources, referred or considered in this work. We have not misrepresented or fabricated or falsified any idea/data/fact/source in this submission. We understand that any violation of the above will be cause for disciplinary action by the Institute.

Sr. No	**Student Name**	**Roll No.**	**Signature**
1	Mr. Sanket Bodhale	1910029	
2	Ms. Chinmai Arekar	2010026	
3	Ms. Sejal Patil	2010028	
4	Ms. Anshul Bhimnath	2010046	

Date :

Place : RIT, Rajaramnagar

Sponsorship Letter



SOFTRON
Software Developer And Training Center
Software Developer, Web Design, Professional And Training Center

Date: 19/5/2023.

Sponsorship grant Letter to be taken on Company's Letter Head

To,
Head of Department,
Information Technology
RIT Rajaramnagar Islampur.

Subject: Offering Sponsorship to B.Tech. (IT) Students in Projects

Dear Sir/Madam,

Softran Kolhapur, agrees to provide sponsorship to following students of Information Technology Department of Rajarambapu Institute of Technology, Islampur to work on a project "Air Canvas using RCNN and Open CV" The work is to be carried out from " 27th March 2023 " till " 18th December 2023 " of Project Work

Sr. No.	Name of Student	Roll Number
1.	Chinmai Anant Arekar	2010026
2.	Sejal Siddharth Patil	2010028
3.	Anshul Shridhar Bhimnath	2010046
4.	Sanket Santosh Bodhale	1910029

We also expect that students will visit our premises as and when called for the project updates and queries with prior permission of the institute.
After completion of project, company will provide following things to institute & students

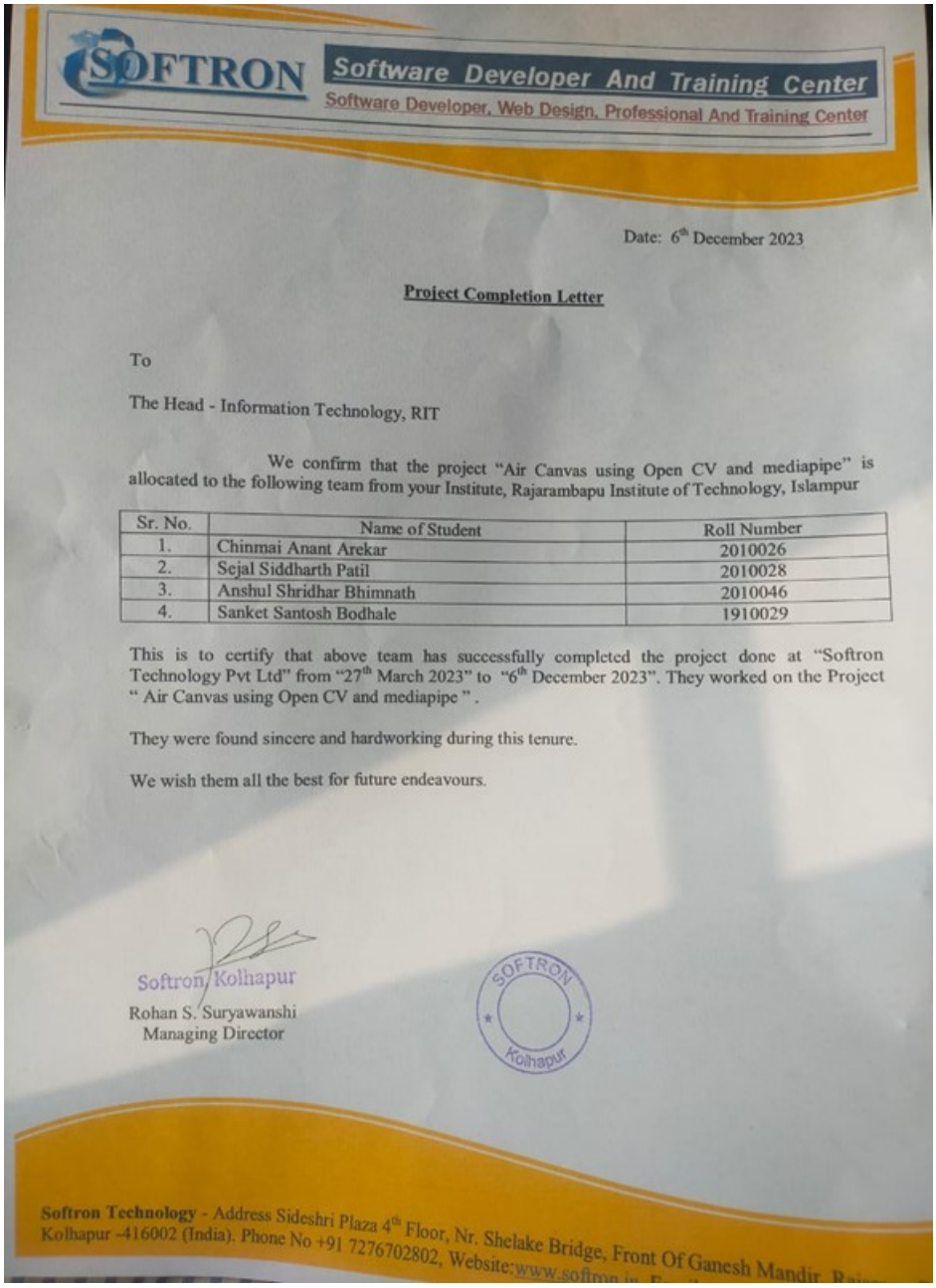
1. Project Completion Letter

Yours truly,

Softron, Kolhapur
Rohan S. Suryawanshi
Managing Director

Softron Technology - Address Sideshri Plaza 4th Floor, Nr. Shelake Bridge, Front Of Ganesh Mandir, Rajaram Road Kolhapur -416002 (India). Phone No +91 7276702802, Website:www.softron.in, Email softron@softron.in

Completion Letter



SOFTRON
Software Developer And Training Center
Software Developer, Web Design, Professional And Training Center

Date: 6th December 2023

Project Completion Letter

To

The Head - Information Technology, RIT

We confirm that the project "Air Canvas using Open CV and mediapipe" is allocated to the following team from your Institute, Rajarambapu Institute of Technology, Islampur

Sr. No.	Name of Student	Roll Number
1.	Chinmai Anant Arekar	2010026
2.	Sejal Siddharth Patil	2010028
3.	Anshul Shridhar Bhimnath	2010046
4.	Sanket Santosh Bodhale	1910029

This is to certify that above team has successfully completed the project done at "Softron Technology Pvt Ltd" from "27th March 2023" to "6th December 2023". They worked on the Project " Air Canvas using Open CV and mediapipe " .

They were found sincere and hardworking during this tenure.

We wish them all the best for future endeavours.

Softron, Kolhapur
Rohan S. Suryawanshi
Managing Director

SOFTRON Kolhapur

Softron Technology - Address Sideshri Plaza 4th Floor, Nr. Shelake Bridge, Front Of Ganesh Mandir, Kolhapur -416002 (India). Phone No +91 7276702802, Website:www.softron.in

Participation Certificates :

Participation Certificate in **Quantum 2K23**

organized by

Rajarambapu Institute of Technology, Islampur, Maharashtra

K. E. Society's

Rajarambapu Institute of Technology, Rajaramnagar

(NAAC A+ Graded Empowered Autonomous Institute Affiliated To Shivaji University, Kolhapur)

In association with

BHARAT FORGE

KALYANI BHARAT FORGE LIMITED

QUANTUM 2K23

A National Level Technical Symposium

Certificate of Appreciation

This is to certify that Mr./Ms. Chinmai Arekar

from Rajarambapu Institute of Technology, Rajaramnagar

Participated in project competition of CS & IT Engineering

"Quantum 2K23- A National Level Technical Symposium" held on 07th Dec. 2023 at RIT, Rajaramnagar, Sakharale.

Prof. Ranjitsingh Patil
Event Co-ordinator

Dr. S. S. Kumbhar
HOD, Civil Engg. Dept
Chief Event Co-ordinator

Dr. L. M. Jugulkar
Dean Student Development

Dr. P. V. Kadole
Director, RIT

K. E. Society's

Rajarambapu Institute of Technology, Rajaramnagar

(NAAC A+ Graded Empowered Autonomous Institute Affiliated To Shivaji University, Kolhapur)

In association with

BHARAT FORGE

KALYANI BHARAT FORGE LIMITED

QUANTUM 2K23

A National Level Technical Symposium

Certificate of Appreciation

This is to certify that Mr./Ms. Sejal Patil

from Rajarambapu Institute of Technology, Rajaramnagar

Participated in project competition of CS & IT Engineering

"Quantum 2K23- A National Level Technical Symposium" held on 07th Dec. 2023 at RIT, Rajaramnagar, Sakharale.

Prof. Ranjitsingh Patil
Event Co-ordinator

Dr. S. S. Kumbhar
HOD, Civil Engg. Dept
Chief Event Co-ordinator

Dr. L. M. Jugulkar
Dean Student Development

Dr. P. V. Kadole
Director, RIT

K. E. Society's

Rajarambapu Institute of Technology, Rajaramnagar

(NAAC A+ Graded Empowered Autonomous Institute Affiliated To Shivaji University, Kolhapur)

In association with

BHARAT FORGE

KALYANI BHARAT FORGE LIMITED

QUANTUM 2K23

A National Level Technical Symposium

Certificate of Appreciation

This is to certify that Mr./Ms. Anshul Bhimnath

from Rajarambapu Institute of Technology, Rajaramnagar

Participated in project competition of CS & IT Engineering

"Quantum 2K23- A National Level Technical Symposium" held on 07th Dec. 2023 at RIT, Rajaramnagar, Sakharale.

Prof. Ranjitsingh Patil
Event Co-ordinator

Dr. S. S. Kumbhar
HOD, Civil Engg. Dept
Chief Event Co-ordinator

Dr. L. M. Jugulkar
Dean Student Development

Dr. P. V. Kadole
Director, RIT

Participation Certificate in **Technovation 2023**

organized by

IEEE Bangalore Section

Technovation 2023

Serial No: T23018-1

Certificate of Participation

This is to certify that Chinmai Anant Arekar from the department of Computer Science and Information Technology, Rajarambapu Institute Of Technology, Islampur, Sangli, Maharashtra has participated and presented their project titled *OpenCV Air Canvas Application using python and Machine Learning* in Technovation 2023, a National level project exhibition organized by IEEE CMRIT SB on 28th June 2023.

Dr. Harsha B K
IEEE CMRIT SBC

Dr. Meenakshi R Patil
Convenor

Dr. Elumalai R
Advisor

Dr. Sanjay Jain
Principal CMRIT

ix.

Technovation 2023

Serial No: T23018-3

Certificate of Participation

This is to certify that Sejal Siddharth Patil from the department of Computer Science and Information Technology, Rajarambapu Institute Of Technology, Islampur, Sangli, Maharashtra has participated and presented their project titled " OpenCV Air Canvas application using python CV2 and Machine Learning. in Technovation 2023, a National level project exhibition organized by IEEE CMRIT SB on 28th June 2023.

Dr. Harsha B K
IEEE CMRIT SBC

Dr. Meenakshi R Patil
Convenor

Dr. Elumalai R.
Advisor

Dr. Sanjay Jain
Principal CMRIT

Technovation 2023

Serial No: T23018-1

Certificate of Participation

This is to certify that Anshul Shridhar Bhimnath from the department of Computer Science and Information Technology, Rajarambapu Institute Of Technology, Islampur, Sangli, Maharashtra has participated and presented their project titled "OPenCV Air Canvas Application using python and Machine Learning in Technovation 2023, a National level project exhibition organized by IEEE CMRIT SB on 28th June 2023.

Dr. Harsha B K
IEEE CMRIT SBC

Dr. Meenakshi R Patil
Convenor

Dr. Elumalai R
Advisor

Dr. Sanjay Jain
Principal CMRIT

ACKNOWLEDGEMENT

It is our foremost duty to express our deep sense of gratitude and respect to the guide **Prof. Savita Patil** sir for his uplifting tendency and inspiring us for taking up this project work successful. We are also grateful to **Dr. A.C. Adamuthe** (Head of Department, Information Technology) sir for providing all necessary facilities to carry out the project work and whose encouraging part has been a perpetual source of information.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Information Technology Department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.

ABSTRACT

Traditionally chalk and blackboard were used for communication and academic purposes. But, as technology advanced, digital writing pads became available. These pads can be used for an array of reasons, including education, presentation, and so on. Teachers must now sit in one area and utilise digital pads to educate their students. As we all know, digital writing pads must be carried everywhere, which is inconvenient, and they must be charged on a regular basis. In addition, the digital pads are not affordable. Teachers cannot keep an eye on their students while sitting in one area. That is where our OpenCV Air Canvas Application comes in. It allows you to sketch in the air, and your design shows on-screen. We explore two methods to develop Air Canvas Application that are Hand Tracking with mediapipe for hand landmarks, and Object Tracking with OpenCV for color-based detection. User gestures control actions like choosing colors or clearing the canvas.

Our application uses gesture recognition, a unique technology that combines sketching and pattern recognition. It allows users to sketch in the air using hand movements by utilising OpenCV and mediapipe. For precise tracking, we recommend hand tracking over object detection. Air Canvas is user-friendly, resolving the issues associated with digital pads. You select colours, clean the screen, and save your designs. It captures video using a camera, reads frames, and employs hand/object tracking algorithms. We have investigated two ways for developing the Air Canvas Application: Hand Tracking with Mediapipe for hand landmarks and Object Tracking with OpenCV for color-based detection. User movements control tasks such as colour selection and canvas cleaning. The programme records your hand motions in real time, so you can see your sketches on the live stream. It's a novel method to sketch digitally.

The output can be shown individually or as an overlay over the camera feed. In summary, the OpenCV Air Canvas Application is an innovative method to digital sketching, providing a cost-effective, portable, and user-friendly alternative to traditional writing pads. The integration of gesture recognition and colour tracking enhances user engagement, making it a great tool for educational institutions, presenters, and anybody looking for a unique approach to express ideas.

Keywords : Air Canvas, Gesture Recognition, Hand Tracking, Object Detection, OpenCV, mediapipe. .

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Before the digital age, communication and information sharing were grounded in simplicity. Traditionally, the mediums of writing were paper, chalk, or writing boards. People relied on basic tools like paper, chalk, and writing boards for writing and teaching. These traditional methods involved hands-on approaches, when technology had not yet transformed the way, we document and convey information.

Nowadays, instead of relying on traditional methods, many people use electronic writing pads. These are like smart notebooks that you can write on with a special pen. It is a bit like having a high-tech piece of paper that stores everything digitally. So, instead of using chalk or paper, we can use digital writing pads to draw, or write on a digital pad and it is also mainly used by teachers and presenters.

But the digital writing pads are a bit hectic to use in the educational institutions. They have certain limitations for the user to be more efficient. It is a lot expensive. It is difficult to carry (portable) around as it needs to be handled carefully. It needs to be charged always to use it. It has a wired connection with the laptop that allows user to roam around in limited space. User needs to hold it until he is done with it. Our proposed solution to this problem is Open Air Canvas Application. The air canvas system is user-friendly and a good solution for all the above-mentioned problems. It allows user to write in air and print on screen, to choose his or her choice of color while drawing on screen, to clear the screen when they wish and to save the paint window on their screen.

In recent years, there has been an interesting and challenging development at the intersection of drawing and pattern recognition. However, the digital age brought new possibilities, leading to the development of gesture recognition systems. As technology advances, the demand for natural human-computer interaction (HCI) systems to replace traditional methods

increases exponentially. Among these new features, Air Canvas stands out as an interesting technology, allowing users to capture air through gestures.

It can be implemented by either using hand tracking method or object detection method. The hand tracking method is implemented using OpenCV and mediapipe that provides accurate landmarks and smooth tracking. Whereas the object detection method focuses on color detection and tracking using the HSV (Hue, Saturation, Value) color space. The user can select different colours using trackbars, and the program detects and tracks the chosen color in the video feed from the webcam. Here we have preferred hand tracking method using OpenCV and mediapipe.

Air Canvas acts as an interactive system, giving users the ability to draw in mid-air with their hands. It combines a virtual canvas for detecting moving hands, choose the color and trace their movement. To bring this concept to life, the project uses OpenCV and mediapipe. The Python programming language forms the backbone of this process along with libraries such as NumPy, mediapipe, and OpenCV.

Chapter 2

Problem Life Cycle

2.1 Problems Identification

When the lectures in the class are going on it was observed that during conducting lectures, teachers use writing pad to write text on the screen. So, it restricts the teachers to stand at one place and conduct the lectures. They cannot roam around in the classroom and cannot carry the pad anywhere at the time it is in use.

2.2 Problem Selection

Traditional digital writing tablets, although common in educational institutions, come with their share of challenges. They are expensive, not portable, require constant charging, and limit users to limited areas due to the use of cables. To address these issues, our proposed solution, Open Air Canvas, offers a user-friendly alternative. So, our aim is to identify the correct technique that will address the above challenges.

2.3 Problem Definition

To solve the issues faced by the teachers while conducting the lectures, we have proposed a solution of Air Canvas which is a technology-based application that allows the user to draw on the air using hand movements. It is based on computer vision and media pipe that can detect hand movements, track their movement in real time and print likewise on screen.

2.3.1 Objectives:-

1. To identify different detection methods for the canvas application and selecting the best one.
2. To allow user to write in air and print on screen.
3. To allow user to choose the color of their choice.
4. To clear the screen as per the user wish.
5. To allow user to save the canvas screen.

2.3.2 Description:-

The problem that the air canvas project using OpenCV aims to address is the ability to draw in mid-air using hand movements. This technology allows users to draw without the need for physical materials such as paper, pencils, or tablets. The air canvas project is intended to create an interactive system that can detect hand gestures in real-time, track hand movements, and map them to a virtual canvas. The main challenge in developing an air canvas system is detecting and tracking hand gestures accurately in real time. This involves applying computer vision algorithms to identify the user's hand. The system must then track the hand movements and map them to the virtual canvas, ensuring that the drawing is accurate and consistent. Another challenge is the ability to draw each and every illustration shown by the user using hand movements. The system must be able to detect and interpret different hand movements and must be able to draw lines, circles, and curves. Finally, the air canvas system must be user-friendly and intuitive, allowing users to easily learn and use the system without extensive training. This requires developing a user interface that is easy to navigate and provides clear instructions on how to use the system. Hence, it mainly focuses on solving some major problems of society. Air writing quickly solve such issues. So, we are developing an Air Canvas Application that will detect hand/finger movements and print it on the screen simultaneously.

2.4 Problem Analysis

2.4.1 Fishbone Diagram

A Fishbone diagram, also known as a cause-and-effect diagram or an Ishikawa diagram, is a visual tool used to identify and analyse possible causes of a problem or to explore potential solutions. It helps teams brainstorm and organize ideas about what might be behind an issue. The main problem sits at the head of the fish, and the "bones" represent different categories of possible causes. It is a way to systematically explore and understand the root causes of a problem.

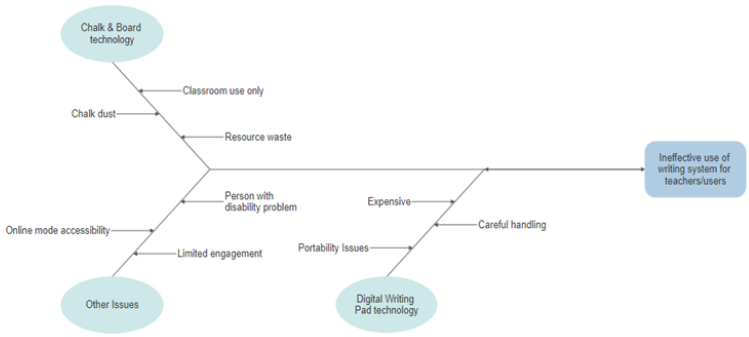


Figure 2.1: Fish Bone Diagram

2.4.2 Why-Why analysis

The "why-why analysis" is a technique used to identify the root causes of a problem or an issue by repeatedly asking "why" until the underlying cause is determined. In the case of the OpenCV Air Canvas application, which assumes to be an application that allows users to draw in the air using gestures and a camera-based input system, a why-why analysis could be conducted to understand the reasons behind any problems or shortcomings of the existing technologies and ineffective use of writing systems for teachers. Here is an example of how a why-why analysis could be applied:
Problem: Why there if ineffective use of existing writing systems for teachers?

1. Why are existing technologies contributing to the ineffective use of the writing system for teachers?

Because the chalk dust and costliness of digital writing tablets becomes the barrier and impact the effectiveness of teaching.

2. Why does chalk dust impact the effectiveness of teaching?
   Chalk dust causes health issues for teachers and students, leading to distraction and discomfort during lessons.

3. Why is the costliness of digital writing tablets a barrier for effective teaching?
   Teachers and educational institutions may find it financially challenging to invest in digital writing tablets, preventing them from accessing the benefits of modern technology.

4. Why does resource waste contribute to the ineffective use of the writing system?
   The resource waste leads to environmental pollution due to improper disposal of chalk and duster waste results in environmental pollution, affecting sustainability.

5. Why is the costliness of writing system equipment a barrier for effective teaching?
   The high cost of equipment makes it challenging for schools and teachers to afford, hindering effective implementation.

   By performing this why-why analysis, we have identified a possible root cause that are the health issues caused by chalk dust, the financial barriers associated with costly digital writing tablets, resource waste leading to environmental pollution, and the high cost of writing system equipment. This analysis suggests that enhancing the technological advancements and implementation of computer vision technologies could potentially lead to the effective use of existing writing systems, thus addressing the issues for improving the overall effectiveness and sustainability of teaching methods.

2.5 End Users

The "Air Canvas" application using OpenCV targets a diverse range of end users who are educators, students, artists, presenters and people with the disability to speak or hear.

1. **Educators and Students:** Interactive teaching through visual demonstrations and creative learning activities. Students can express ideas visually.

2. **Artists and Creative Individuals:** Provides a digital canvas for the creation of digital art using gestures and in a playful manner.

3. **People with disability to speak or hear:** Individuals to convey ideas visually, overcoming communication barriers for those with speech or hearing disabilities.

Chapter 3

Literature Survey and Motivation

- The "Air Canvas" application using OpenCV and NumPy in Python was introduced by S.U. Saoji, Nishtha Dua et.al.[1]. This project aimed to enhance user interaction on digital platforms through air gestures. Leveraging OpenCV, NumPy, and Python, the study achieved widespread accessibility and ease of implementation. However, it faced disadvantages such as limited precision, particularly for detailed gestures or complex applications.

- Aryan Gupta, Naman Chawla et.al.[2] contributed to touchless control through hand gestures using MediaPipe and OpenCV. Their project reduced the need for physical contact, emphasizing accessibility in public spaces. However, gestural precision may have required refinement, with potential variations in user adaptability.

- Palak Rai, Reeya Gupta et.al.[3] contributed to interactive learning experiences with a Virtual Canvas using OpenCV. Their project fostered collaboration and creativity, acting as an engaging educational tool. However, hardware limitations might have impacted accessibility, necessitating potential user interface improvements.

- Abhishek Kumar, Aman Kumar Jha et.al.[4] focused on creating a virtual drawing board using Python, capitalizing on the language's popularity and versatility. While leveraging Python's strengths, there was a need for user interface refinement for an optimal user experience.

- Shaurya Gulati, Ashish Kumar Rastogi et.al.[5] introduced a platform for artistic expression through a webcam, using MediaPipe and OpenCV. The project enabled webcam accessibility and real-time feedback for creative expression. However, challenges

arose in terms of accuracy variability, especially concerning webcam quality.

- Bhargav D V, Mayura Nandan M R et.al.[6] proposed a straightforward method for air-based scripting using computer vision. The project emphasized user-friendliness, allowing nine scripting tasks without complex setups. However, there were limitations in handling complex scripting tasks.

- Tamalampudi Hemaa Chandhan, Kavin Kumar R, Nalin Raj et.al.[7] developed an air canvas through accurate hand tracking technology using OpenCV and MediaPipe. This project enhanced precision for precise gestures and artistic expression but had a dependency on hand tracking, presenting potential challenges in false positives.

- R. Nitin Kumar et. al. [8] presented an innovative approach for hands-free text input through air writing gestures using MediaPipe and OpenCV. This project offered real-time recognition and diverse applications, including accessibility for differently-abled users. Nevertheless, the system exhibited environmental sensitivity, with reliability affected by factors like lighting conditions and background clutter.

- Subhash Chand Agarwal, Rajesh Kumar Tripathi et.al.[9] offered a virtual drawing tool with air-based input for creative freedom. While providing an interactive and intuitive interface, the project may have had limited features compared to professional design software.

- In a survey on virtual paint board systems using computer vision, Yasvi Vamja, Chaitali Patil et.al.[10] provided insights into different approaches and challenges. However, the survey may have had limitations in-depth analysis, and subjective survey responses might have limited the generalization of findings.

- Nethravati T L, Rakshita L Patil et.al. [11] explored AI-based virtual painting applications and the integration of OpenCV. Their project aimed to enhance creativity through AI-driven approaches, producing realistic artistic effects. However, the complexity of AI implementation may have impacted real-time performance, requiring resource-intensive processing.

Table 3.1: Literature Survey table

Paper	Concept	Technology used	Advantages	Disadvantages
Air Canvas Application Using OpenCV and NumPy in Python [1]	Aims to enhance user interaction with digital platforms by introducing air gestures, providing a more intuitive and engaging input method compared to traditional interfaces.	OpenCV NumPy Python	Widespread Accessibility: Utilizes widely used libraries (OpenCV, NumPy), making it accessible to a broad audience. Ease of Implementation: Simplicity in implementation allows for quick adoption and integration.	Limited Precision: May lack the precision required for detailed gestures or complex applications due to the simplicity of the approach.
Gesture Based Touchless Operations Leveraging MediaPipe and OpenCV [2]	Enables touchless control through hand gestures for various applications, emphasizing accessibility in public spaces.	OpenCV MediaPipe	Touchless Interaction: Reduces the need for physical contact, improving accessibility in public spaces.	Gestural Precision: May require refinement for precise gestural interactions, with potential variations in user adaptability.

Air Writing Recognition using MediaPipe and OpenCV [3]	Provides a hands-free alternative for text input, leveraging air writing gestures. Addresses the need for innovative input methods beyond traditional keyboards.	OpenCV Mediapipe	- Real-time Recognition: Capable of real time recognition, offering potential applications in accessibility and user convenience. Diverse Applications: Extends usability to various domains, accommodating differently-abled users.	Environmental Sensitivity: Reliability may be affected by factors such as lighting conditions and background clutter.
Virtual Air Board using Python [4]	Aims to create a virtual drawing board using Python, catering to users who prefer the Python programming language.	Python	Python Popularity: Leverages the popularity and ease of Python programming for wider adoption. Versatility: Offers a versatile application with potential for remote collaboration.	User Interface: The user interface may require refinement for an optimal user experience.

Virtual Canvas for Interactive Learning using OpenCV [5]	Enhances interactive learning experiences by providing a virtual canvas, fostering collaboration and creativity among users.	OpenCV	Collaborative Learning: Provides a platform for collaborative learning experiences, enhancing engagement. Educational Tool: Acts as an engaging educational tool with potential applications in various learning environments.	Hardware Limitations: Accessibility may be limited by hardware requirements, and user interface improvements may be necessary
Paint/Writing Application through WebCam using MediaPipe and OpenCV [6]	Provides a platform for artistic expression through a webcam, making creative activities accessible to a broader audience.	OpenCV MediaPipe	Webcam Accessibility: Enables widespread use with common webcams, facilitating accessibility. Real-time Feedback: Offers real-time feedback for creative expression.	Accuracy Variability: Accuracy may vary based on webcam quality, and precision might be a concern for detailed artwork.
A Simple Approach for Scripting in Air and Display using Computer Vision [7]	Proposes a straightforward method for air-based scripting, simplifying user interaction for scripting tasks in a novel and accessible way.	Computer Vision	User-Friendly: Simple implementation allows for scripting tasks without complex setups.	Limited Scripting Complexity: May have limitations in handling complex scripting tasks.
Air Canvas: Hand Tracking Using OpenCV and MediaPipe [8]	Develops an air canvas through accurate hand tracking technology, allowing users to create digital art using hand gestures.	OpenCV MediaPipe	Accurate Hand Tracking: Enhances precision for precise gestures and artistic expression.	Dependency on Hand Tracking: Relies on robust hand tracking, with potential challenges in false positives or misinterpretation of gestures.

Gesture-Based Touchless Operations Leveraging MediaPipe and OpenCV [8]	Enables touchless control through hand gestures for various applications, emphasizing accessibility in public spaces.	OpenCV Mediapipe	Touchless Interaction: Reduces the need for physical contact, improving accessibility in public spaces.	Gestural Precision: May require refinement for precise gestural interactions, with potential variations in user adaptability.
Virtual Drawing: An Air Paint Application [9]	Offers a virtual drawing tool with air-based input for creative freedom and innovative artwork.	Computer Vision Color detection and recognition Mask	Creative Freedom: Provides creative freedom for users with an interactive and intuitive interface.	- Feature Limitations: May have limited features compared to professional design software, potentially not suitable for advanced graphic design.
Survey on Virtual Paint Board Using Computer Vision [10]	Provides an overview of various virtual paint board implementations, identifies common challenges, and contributes to understanding trends .	Computer Vision	Comprehensive Overview: Offers insights into different approaches and challenges in virtual paint board systems.	Limited In-depth Analysis: May lack in-depth analysis, and subjective survey responses might limit the generalization of findings.
Virtual Painter using Artificial Intelligence and OpenCV [11]	Likely explores AI based virtual painting applications and the integration of OpenCV. Investigates advancements in AI driven creativity. Aims to enhance creativity in virtual painting through AI driven approaches, producing realistic artistic effects.	OpenCV Artificial Intelligence	- Realistic Artistic Effects: Potential for realistic effects through AI driven creativity.	Complexity: Implementation complexity in AI may impact real time performance, requiring resource-intensive processing.

As shown in the above Table 3.1, the literature survey comprises of diverse projects employing technologies such as OpenCV, NumPy, MediaPipe, Python, and Artificial Intelligence to advance user interaction, creative expression, and educational experiences. From air gestures for interactive learning to touchless operations and AI-driven virtual painting, each project demonstrates innovative digital interfaces. Whereas these projects lack in precision, the user interface in various applications may require refinement to ensure an optimal user experience, some projects may face limitations in accessibility due to specific hardware requirements, necessitating potential user interface improvements and certain projects may have limitations in handling complex scripting tasks. So, this gives us direction to come up with a new solution that will address all the above challenges and identify the best detection method to enhance the robustness, accessibility, and user experience of the air canvas application.

Chapter 4

Proposed System and Requirement Specification

4.1 Proposed Solution/ System & Methodology

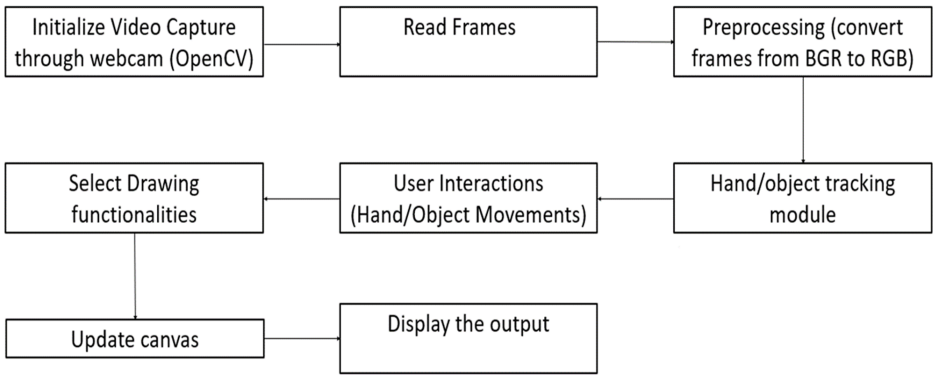


Figure 4.1: Block Diagram - Air Canvas Application

To address the challenges w.r. to existing technologies in writing on boards or usage of writing pads as discussed in previous chapters, OpenCV Air Canvas application is proposed to works as shown in Figure 4.1.1. The steps are discussed and explained below in detail.

1. **Initialize video capture through webcam:**
   To develop an air canvas application, we need to first initialize the video capture through webcam. Create a video capture object to access the camera feed. This method creates a Video Capture object named cap that is used to capture video frames from a specified video source. The value 0 refers to the default camera device on your system. If you have multiple cameras connected, you can use different indices to access them.
2. **Read Frames:**
   Continuously read frames from the video capture object. It reads a frame from the webcam using the cap VideoCapture object. This method returns two values which indicates whether a frame was successfully read or not, and frame, which contains the actual frame data.
3. **Preprocessing:**
   To improve the quality of the image or make processing easier, convert the collected frames or use additional preprocessing methods as necessary. In this project, preprocessing is primarily performed on the frame obtained from the webcam before feeding it into the hand detection model. The preprocessing involves converting the frame from BGR to RGB color space, flipping it vertically, and then passing it to the hands.process() method for hand detection.
4. **Hand/Object Tracking Module:**
   To develop an air canvas application, we have proposed two different detection methods that are hand tracking method and object tracking method as explained below.
   - **Hand Tracking method :** Hand detection technique is implemented to locate the user's hand of interest within the captured frame. This can be achieved using mediapipe, through the utilization of the Hands class from mediapipe.solutions.hands. This class is specifically designed to perform hand landmark detection, allowing for the identification of key points on the hand, such as fingertips and the hand centre. The process begins with the initialization of the mpHands module, configuring it to use the Hands class for hand tracking. The max_num_hands parameter is set to 1, due to which the system detects and tracks a single hand in each frame. Additionally, a minimum detection confidence level is specified to ensure the reliability of hand detection results.

- **Object Tracking method:** The object detection technique in this project relies on OpenCV and the integration of color tracking. The code initializes color tracking parameters using OpenCV functions, such as creating trackbars for upper and lower hue, saturation, and value (HSV) values. These trackbars allow the user to dynamically set the color thresholds for the object to be tracked. Color points are updated based on the tracked object's movements within the specified color range. Contours are identified in the segmented color mask, and the largest contour is selected as the object of interest. Object tracking is achieved by determining the object's position and drawing a circle around its centroid.
- **User Interactions:** In hand tracking the mapping between recognized gestures and specific drawing actions. For example, a closed fist gesture may correspond to the end of a drawing, while an open palm gesture correspond to start drawing and to choose colour. The gestures are mapped based on the positions of specific landmarks on the user's hand. The code primarily focuses on detecting the position of the thumb and forefinger to interpret gestures. The specific gestures are used to control the drawing behaviour on the virtual canvas. In object tracking, gestures are associated with actions based on the tracked object's movements and its position within designated regions of the frame. The system identifies user interactions by checking the object's position. The user can dynamically set color thresholds using trackbars, and the object's position is continuously tracked for real-time feedback. By leveraging color tracking and object positioning, gestures in the object detection module are mapped to actions such as selecting colours and interacting with the canvas in a manner distinct from hand tracking.
- **Select Drawing functionalities:** Add user controls, such as GUI elements, clear the canvas, saving screen, choosing colour or perform other actions. The system allows users to choose different drawing colors by positioning their hand over specific regions of the frame corresponding to colors such as blue, green, red, and yellow. The interactive canvas is continuously rendered based on the detected hand movements. As the user moves their hand, color points are updated, and lines are drawn on the canvas. The application also incorporates a "CLEAR" button on the frame, allowing users to clear the canvas when their hand is positioned over the designated region. The output

image including the canvas can be saved on user's device. Lines drawn on the live video frame mirror the actions on the canvas, enabling users to see their drawings directly on the video feed.

- **Update Canvas:** Continuously track the movements of the user's hand to canvas screen. The user's hand movements are tracked, and lines are drawn on the canvas based on those movements, creating a real-time interactive drawing experience. The color of the lines depends on the color currently selected by the user. the hands are tracked using the mediapipe library, and the canvas is printed by drawing lines on a virtual canvas based on hand movements. Based on the tracked movements and recognized gestures, update the canvas by drawing shapes.
- **Display the Output:** Show the output canvas on a separate window or overlay the drawing onto the real-time camera feed. cv2.imshow("Output", frame) displays the live video feed with hand landmarks and drawn lines on the frame.

4.2 Software Requirements Specification-SRS

The Software Requirements Specification is designed to document and describe the agreement between the customer and the developer regarding the specification of the software product requested. This Software Requirements Specification aims to describe the Functionality, External Interfaces, Attributes, and Design Constraints imposed on the Implementation of the software system described throughout the rest of the document.

4.2.1 Modules

Table 4.1: Modules

Sr. No	Module Name	Description
1	Canvas Module	Canvas module imprints the human action on the screen as per the chosen colour (by default blue colour).
2	Camera Module	It finds the hand in the visible area, tracks the hand found within that area and then maps out the detected hand movements on the canvas screen.

4.2.2 Functional Requirements

1. **Hand Detection:**
   Initial phase of the project is to detect the hand through Camera Module. First it detects the hand as per the mentioned number and then detects and displays different points on the hand and fingers. This module also maps out the hand movements on the canvas screen.
2. **Hand Impression:**
   When the hand is detected by the camera module the hand movements is imprinted over the canvas module.
3. **Canvas Display:**
   Canvas module imprints the hand movements on the canvas screen. Multiple colors can be chosen on the same screen to highlight the points or useit as per the requirement. In this module there are various features such as the application allow users to draw on a canvas in real-time, enable users to select colours, allow users to clean the entire canvas whenever they wish and save the drawing on the screen. This module prints the object movements on the screen accurately and provide output to the user.

4.2.3 Non-Functional Requirements

1. **Reliability:**
   The system should consistently and correctly identify the presence of a hand in the designated area and it should maintain accurate hand detection even when the hand is in motion.
2. **Performance:**
   Optimize the system's performance to achieve real-time object detection and tracking with minimal latency. Ensure that the drawing response is smooth and responsive to user input, providing a seamless and interactive experience.
3. **Usability:**
   Design a user-friendly interface that is intuitive and easy to navigate, allowing users to interact with the canvas and drawing tools effortlessly. Provide clear instructions or tutorials to guide users on how to operate the system effectively.

4.3 Significance of the project

- Educators can use the Air Canvas for interactive teaching, allowing them to visually demonstrate concepts and engage students in creative learning activities.
- The application can be integrated into online learning environments, providing a unique and interactive tool for remote education.
- Artists and creative individuals can utilize the Air Canvas as a digital canvas, enabling them to create digital art using intuitive hand gestures.
- Professionals can use the Air Canvas during business meetings and presentations to add dynamic drawings and annotations, making presentations more engaging and interactive.
- Individuals with speech or hearing disabilities can use the Air Canvas as a means of visual communication, overcoming barriers and expressing ideas effectively .
- This application can lead to cost savings by eliminating the need for traditional physical drawing materials, such as chalks, markers, and whiteboards.
- The "Air Canvas" relies on a standard webcam, which is commonly available in most laptops and computers. Users do not need to invest in additional hardware, making it a cost-effective solution.

4.4 Scope of Project

- The system's performance may be affected in low-light conditions.
- The complex or rapid hand movements might not be accurately captured.
- Users may need to adjust gestures or movement speed for optimal accuracy.
- Users may need to adapt to specific gestures or modify movement speed for improved system accuracy.
- Compatibility with future updates and technological advancements.

4.5 Deployment Requirements

4.5.1 Hardware Requirements

Table 4.2: Hardware Requirements

Sr. No.	Requirement	Specification
1	Processor	Intel i5
2	RAM	4 gb
3	Storage	1 gb or higher
4	Web Camera	640*480 resolution or higher

4.5.2 Software Requirements

Table 4.3: Software Requirements

Sr. No.	Requirement	Specification
1	Operating System	Windows 10
2	Programming Language - Python	Version 3.7 or higher
3	OpenCV	Version 4.0 or higher

4.6 Project Deliverable

Report (Hard and Soft Copy):
A comprehensive report documenting the project's objectives, methodology, findings, and recommendations. The report should include details about air canvas application model, its performance, and any relevant analysis.
Source Code & Executable Files:
Provide the complete source code of air canvas application. This includes all the scripts, modules, and libraries used in the development. Additionally, if applicable, provide any compiled executable files or deployment packages for running the model.
Certificates:
Included the project completion certificate received from the sponsored company.

4.7 Project Success

The success of an OpenCV Air Canvas application project can be defined based on various criteria and goals established by the developers or stakeholders. Here are some factors that can be considered when evaluating the success of the project:

1. **Functionality:** The application should fulfil its intended purpose effectively. It should accurately track hand movements, recognize gestures, and provide a smooth and intuitive drawing experience. The core functionality of the application should work as expected without significant issues or limitations.
2. **User Experience :** The application should provide a positive and satisfying user experience. This includes factors such as responsiveness, ease of use, intuitive controls, and visually appealing interface design. The users should find the application enjoyable and easy to navigate.
3. **Accuracy and Performance :** The hand tracking and recognition algorithms should perform accurately and reliably. The application should be able to track hand movements in real-time with minimal latency. Drawing gestures should be accurately translated into lines or shapes on the canvas without significant delays or inaccuracies.
4. **Stability and Reliability :** The application should be stable and reliable, with minimal crashes or unexpected behaviour. It should handle errors or exceptions

gracefully, providing appropriate feedback or recovery mechanisms when issues occur. The application should undergo thorough testing to ensure its stability and reliability.

5. **Compatibility :** The application should be compatible with the intended platforms and operating systems. It should work smoothly on various devices, including desktop computers. Compatibility with different screen resolutions, input methods, and device orientations should be considered.
6. **Performance Optimization :** The application should be optimized for performance to ensure smooth and responsive hand tracking and drawing. It should utilize efficient algorithms and take advantage of hardware acceleration when available. The application should aim for minimal resource consumption, such as CPU and memory usage.
7. **User Feedback :** User feedback and satisfaction can be an important measure of project success. Positive feedback, reviews, or user ratings indicate that the application meets user expectations and provides value. Feedback can be gathered through surveys and user testing.
8. **Timeliness and Budget :** Project success can also be measured by the adherence to project timelines and budgets. If the project is completed within the planned timeframe and allocated budget while meeting the desired functionality and quality, it can be considered a success from a project management perspective.
9. **Adoption and Impact :** The success of the application can also be evaluated based on its adoption and impact in the target user community. If the application gains popularity, attracts a significant number of users, or receives recognition in relevant communities or events, it can be seen as a measure of success.

Chapter 5

Design

5.1 Flow Chart

It is a diagram that shows step-by-step progression through a procedure or system especially using connecting lines and a set of conventional symbols as shown in given Figure 5.1.

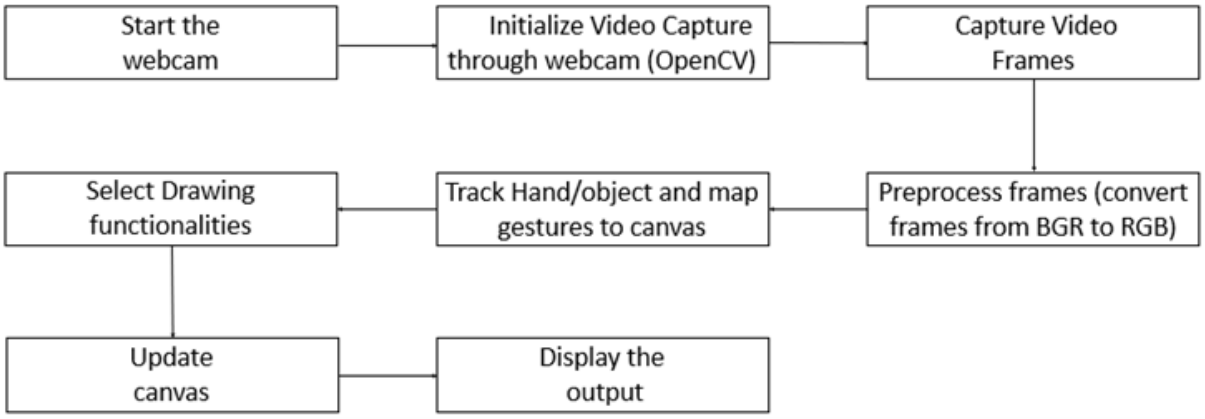


Figure 5.1: Flow chart

5.2 System Architecture

System architecture diagrams provide a visual illustration of a system's various components as shown in Figure 5.2.

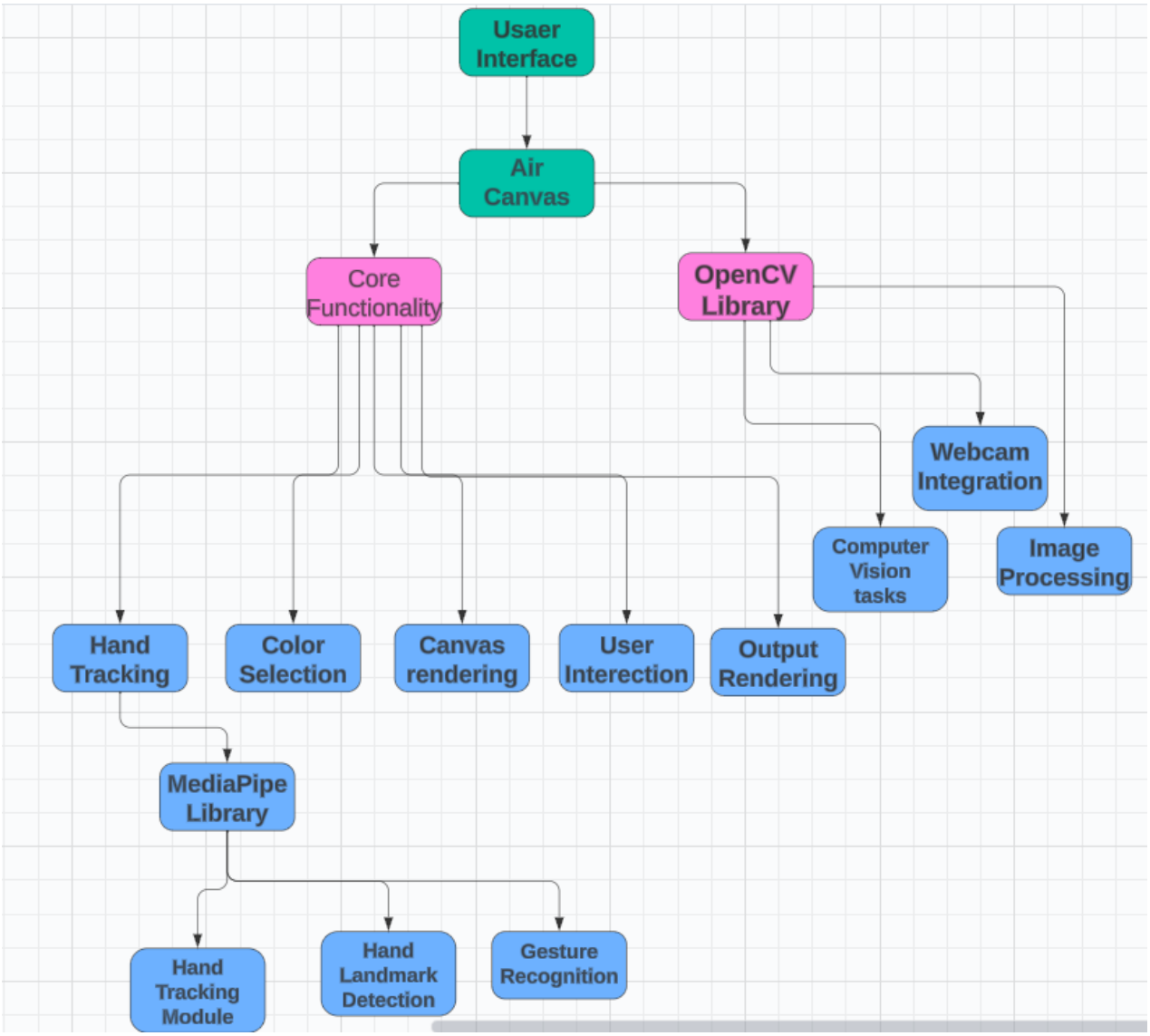


Figure 5.2: System Architecture

5.3 Data Flow Diagrams

5.3.1 DFD Level-0

DFD Level 0 is also called a Context Diagram. It is a basic overview of the whole system or process being analyzed or modelled as shown in Figure 5.3.

Figure 5.3: DFD Level-0

5.3.2 DFD Level-1

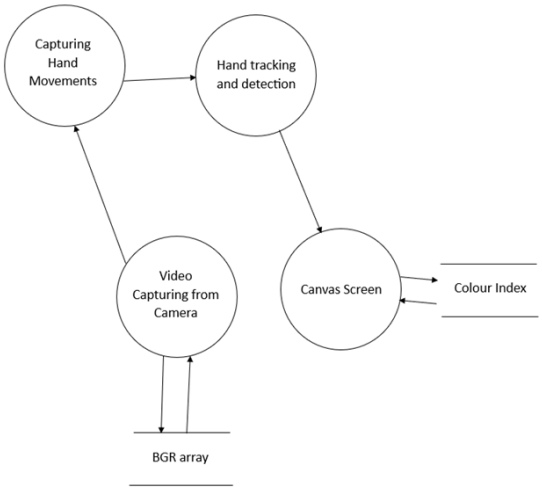


Figure 5.4: DFD Level-1

5.3.3 DFD Level-2

A Data Flow Diagram (DFD) Level 2 is a template that provides a more detailed view of a system. It identifies the individual processes that make up an information system and the data that flows between them. The processes given in Figure 5.5 forms the OpenCV Air Canvas Application.

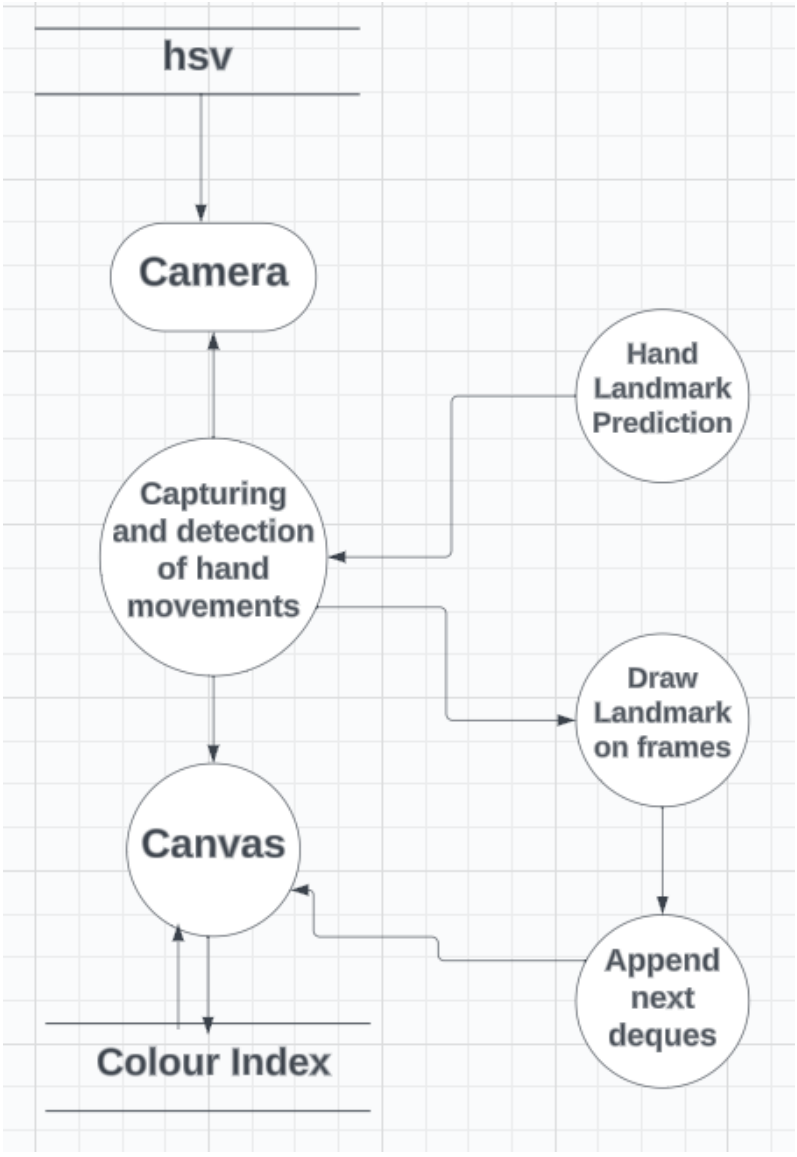


Figure 5.5: DFD Level-2

5.4 UML Diagrams

5.4.1 Class Diagram

Classes are shown by the rectangles in this illustration. Rectangle is broken down into three sections: class name, attribute, and operation. Upper elegance has a relationship with its subclass, as evidenced by the generalisation. The relationship is indicated by the arrow. Subclasses are used to connect interfaces in figure 5.6

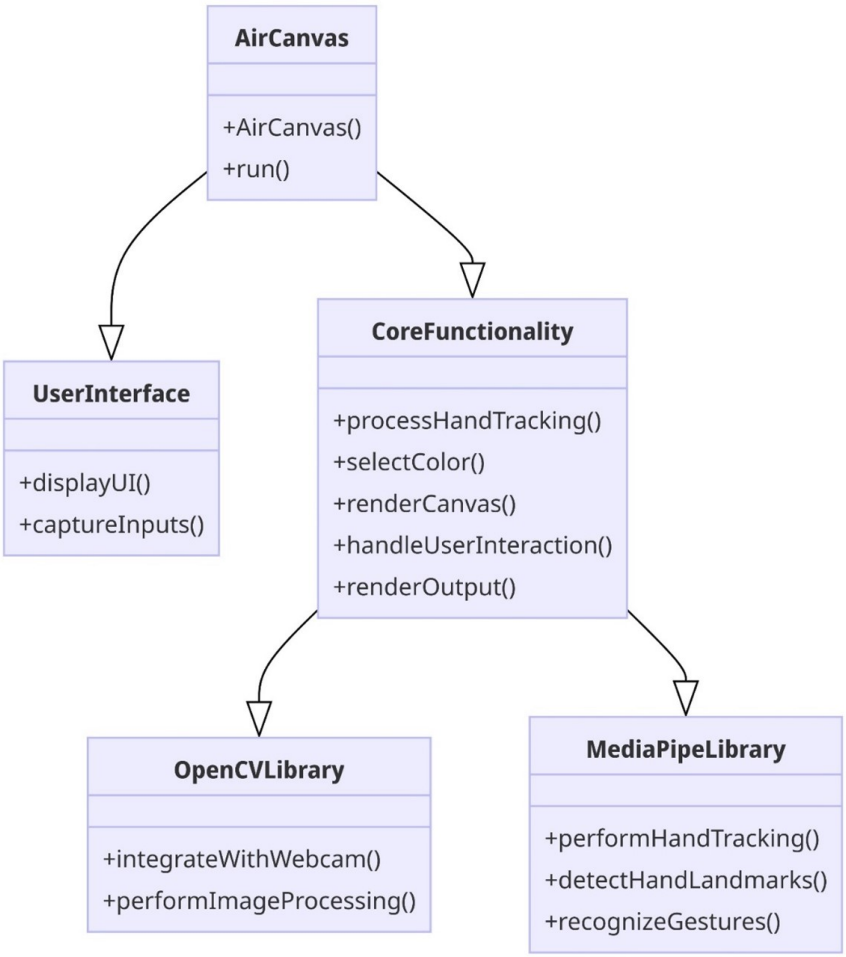


Figure 5.6: Class Diagram

5.4.2 Component diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system as shown in Figure 5.7.

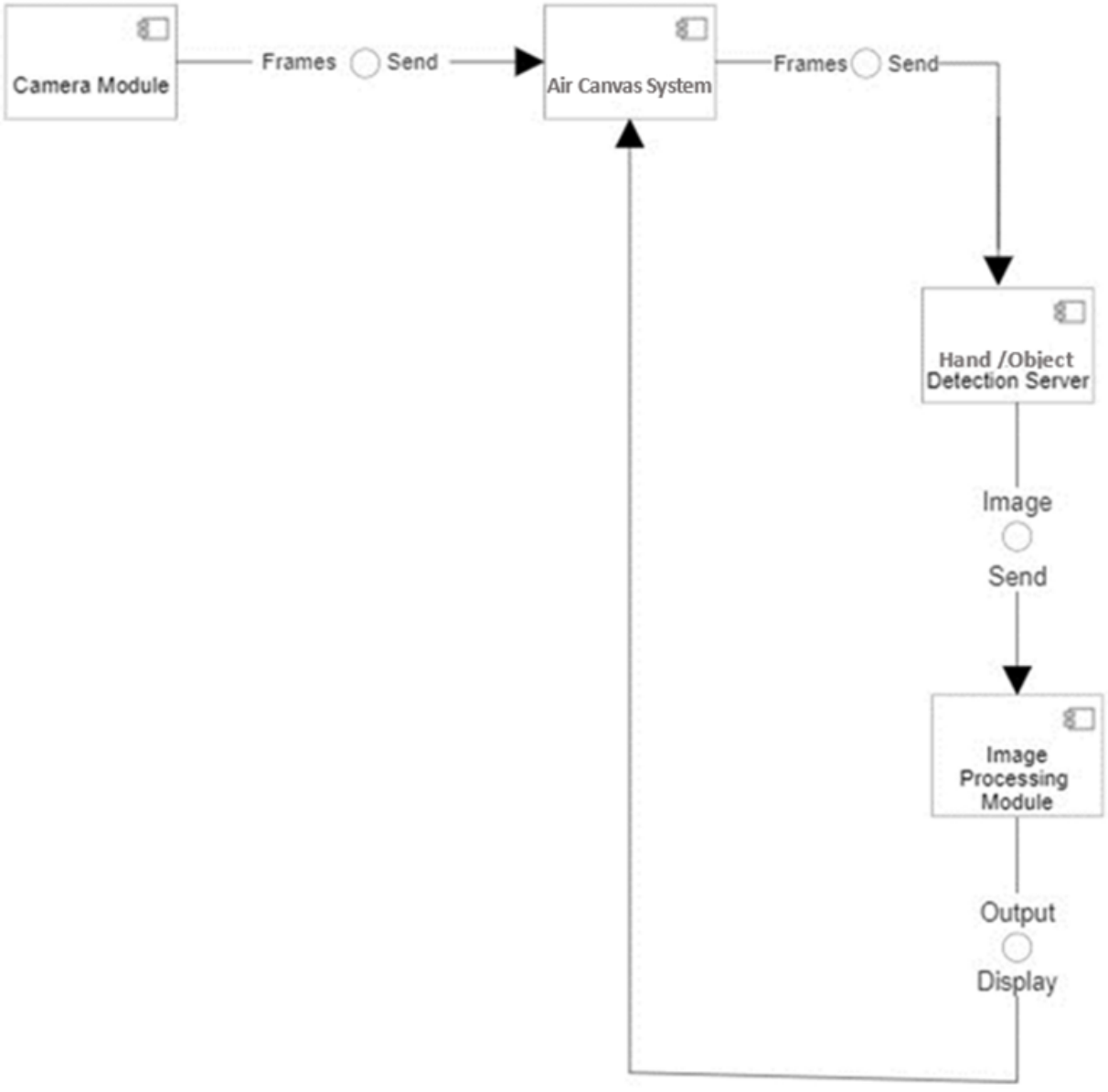


Figure 5.7: Component Diagram

5.4.3 Activity Diagram

Activity diagrams show the flow of one activity to another within a system or process. As shown in the Figure 5.8, it shows the various activities and their flow in the Air Canvas Application.

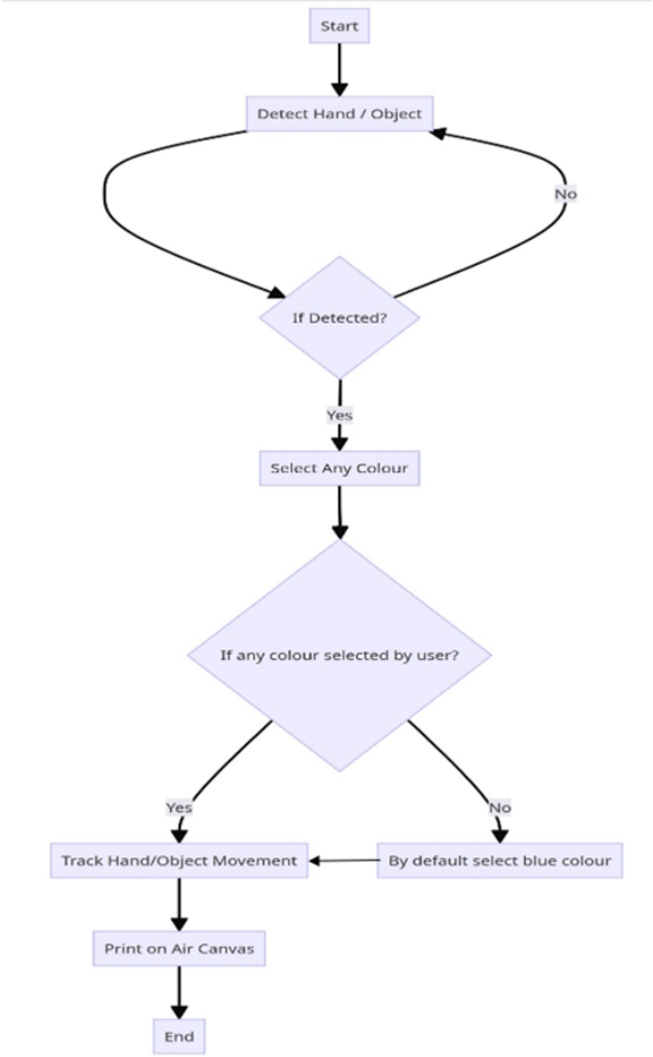


Figure 5.8: Activity Diagram

5.4.4 Sequence Diagram

A sequence diagram is a Unified Modelling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction as shown in Figure 5.9

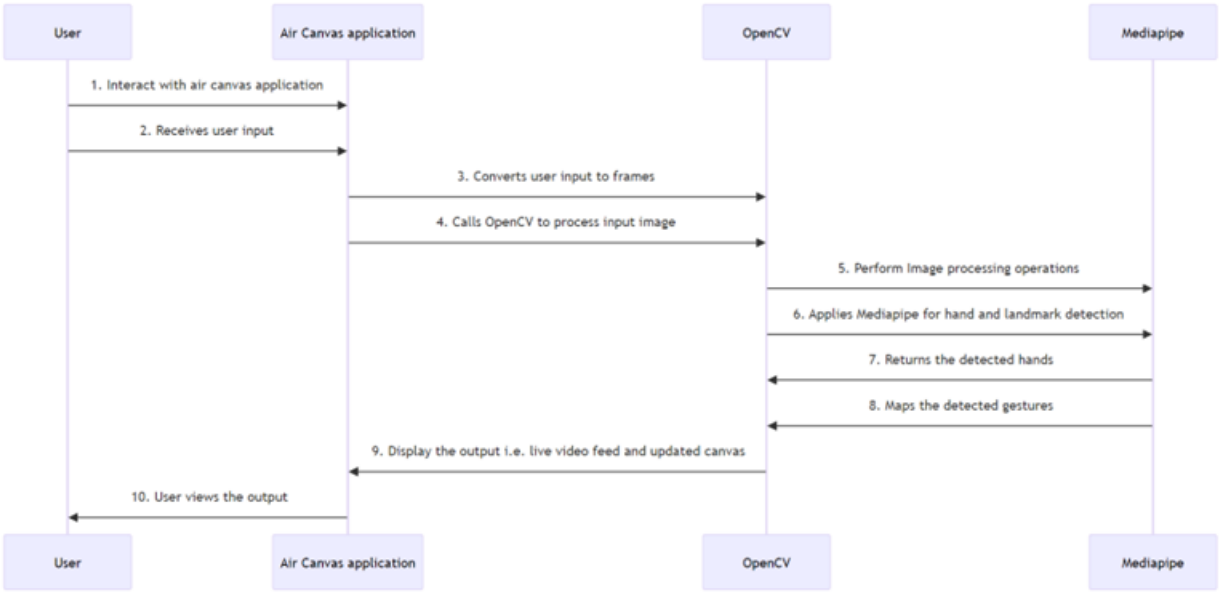


Figure 5.9: Sequence Diagram

5.4.5 Use Case Diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system as shown in Figure 5.10.

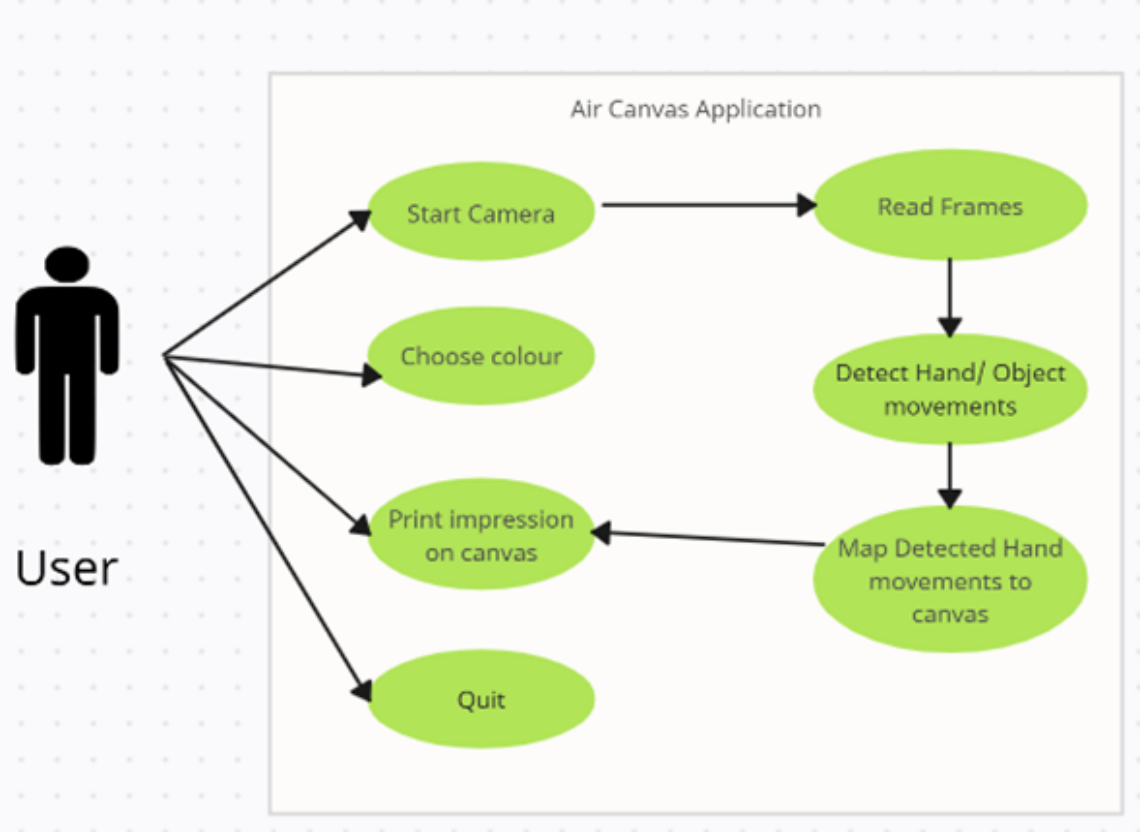


Figure 5.10: Use Case Diagram

Chapter 6

Development/Implementation Details

6.1 Installation

6.1.1 Visual Studio Code Installation:

- Visit the Visual Studio Code official website.
- Click "Download for Windows" and run the installer.
- Accept terms, choose installation location, and click Install.
- Open Visual Studio Code to start coding.

6.1.2 Install Python:

- Visit Python official website.
- Download Python 3.11.2 (64-bit) for Windows.
- Run the installer, check options for all users and PATH, then install.
- Verify installation by typing "python –version" in Command Prompt.

Python is a high-level, versatile programming language known for its readability and ease of use. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is widely used for computer vision applications, and it serves as the primary language for integrating the OpenCV

(Open-Source Computer Vision) library. Python has a vast and active community of developers, resulting in extensive libraries and resources that can be leveraged for different functionalities.

The Air Canvas application benefits from community-contributed modules, like the Mediapipe library for hand tracking. Additionally, Python is cross-platform, meaning the Air Canvas application can be developed and run on various operating systems such as Windows, macOS, and Linux without major modifications. This flexibility enhances the accessibility and usability of the Air Canvas application, allowing users across different platforms to enjoy the features seamlessly .

6.1.3 Python Libraries Framework:

- **OpenCV:** pip install OpenCV-python.
  OpenCV (Open-Source Computer Vision) is an open-source library that provides tools and functions for computer vision and image processing applications. It is written in C++ and has interfaces for various programming languages, including Python, making it widely accessible and utilized. OpenCV offers a rich set of functions for image and video processing. In the Air Canvas application, it is used to capture video frames from a webcam in real-time.
  OpenCV provides algorithms for feature detection, which is crucial for tracking hands in the Air Canvas application. Feature points on the hand are identified, allowing the system to track hand movements. OpenCV is cross-platform, meaning it can be used on different operating systems, including Windows, macOS, and Linux. This ensures that the Air Canvas application can be deployed on a variety of devices.OpenCV has a Python interface, making it accessible and easy to integrate with Python-based applications like the Air Canvas.

- **MediaPipe:** pip install mediapipe.
  Mediapipe is a library developed by Google that provides a simple and efficient way to implement various computer vision and machine learning applications. It offers a pre-built solutions for tasks like hand tracking, face detection, pose estimation, and more. Mediapipe makes it easier for developers to integrate complex computer vision functionalities into their applications without having to build these features from scratch. It provides a set of ready-to-use tools and models that simplify the development process, allowing developers to focus more on the

application's overall functionality rather than the intricate details of computer vision algorithms.

In the Air Canvas application, it can be utilized to detect and track the user's hand movements from the video stream captured by the webcam. The library can identify key landmarks on the hand, allowing for precise tracking of fingers and gestures. It seamlessly integrates with OpenCV, combining the strengths of both libraries for a comprehensive solution.

The mediapipe library detects the hand as shown in the Figure 6.1. In the visualization below, the red dots represent the localized hand landmarks, the green lines are simply connections between selected landmark pairs for visualization of the hand skeleton and the yellow dot represents that one selected or preferred landmark by user that will help in printing and the Figure 6.2 represents the different hand landmark number and their names.

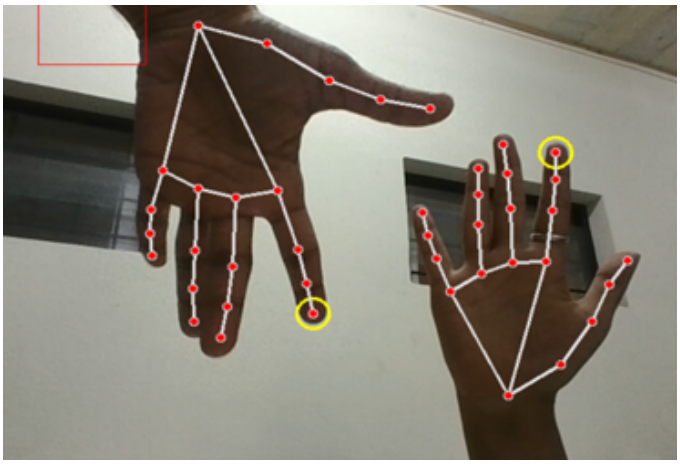
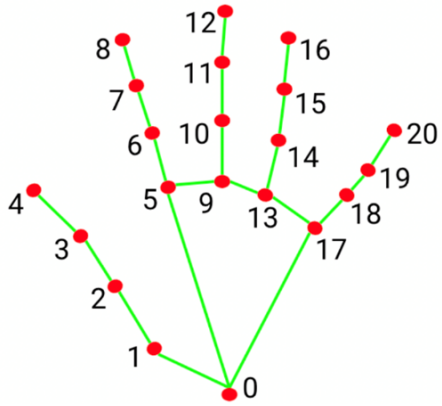


Figure 6.1: Hand tracking



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

Figure 6.2: Hand landmarks

- **NumPy:** pip install numpy.
  NumPy is a powerful numerical computing library for Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of high-level mathematical functions to operate on these arrays. NumPy enables efficient handling of image data in the form of arrays. In the Air Canvas application, NumPy can be used for tasks such as manipulating pixel values, creating image masks, and performing array operations on the video frames captured by the webcam.

6.1.4 Convert to Executable File:

- Install auto-py-to-exe: pip install auto-py-to-exe.
- Run auto-py-to-exe in the command line.
- Select script, output type, and configure options in the GUI.
- Click "Convert.py to .exe" to generate the executable file in the output folder.

6.2 Implementation

1. **Webcam Initialization and video processing:**
   To implement the OpenCV Air Canvas Application we need to first of all initialize the webcam and perform video processing through the webcam. The webcam is initialized using OpenCV's VideoCapture(0) method. A boolean variable

ret is used to check if the webcam is capturing frames successfully and it is set to True as shown in below code snippet. The frames are flipped vertically (cv2.flip(frame, 1)) to ensure that the video feed appears in the correct orientation. The main loop continuously reads frames from the webcam and processes them.

```
cap = cv2.VideoCapture(0) ret = True
```

2. **Image preprocessing:**
   The image preprocessing involves converting the BGR frame to RGB format, which is required for compatibility with the mediapipe library. The hand landmarks are detected using the hands.process() method from the mediapipe library as shown in below code snippet. Landmark coordinates are then extracted and processed further. The framergb variable stores the RGB version of the frame, which is then processed by the mediapipe hands model. The resulting landmarks are used for further processing.

```
# Convert BGR frame to RGB
framergb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
# Get hand landmark prediction
result = hands.process(framergb)
```

3. **Color Handling:**
   Different arrays (bpoints, gpoints, rpoints, ypoints) were employed to manage color points for blue, green, red, and yellow, respectively. The colors list stores RGB tuples for each color, and colorIndex tracks the current drawing color. Four deques are initialized to store color points for blue, green, red, and yellow. These deques will hold the drawing points for each color as bpoints = [deque(maxlen=1024)].

4. **Canvas Setup:**
   Then we have created a canvas (paintWindow) using NumPy arrays to facilitate drawing, defined rectangles and text for different color options and a "CLEAR" button on the canvas and displayed the canvas using OpenCV as shown in the code snippet below.

```
paintWindow = np.zeros((471, 636, 3)) + 255.

cv2.imshow("Output", frame) cv2.imshow("Paint", paintWindow).

frame = cv2.rectangle(frame, (40,1), (140,65), (0,0,0), 2)

frame = cv2.rectangle(frame, (160,1), (255,65), (255,0,0), 2)

cv2.putText(frame, "CLEAR", (49, 33), cv2.FONT_HERSHEY_SIMPLEX,
0.5, (0, 0, 0), 2, cv2.LINE_AA)

cv2.putText(frame, "BLUE", (185, 33), cv2.FONT_HERSHEY_SIMPLEX,
0.5, (0, 0, 0), 2, cv2.LINE_AA)
```

After implementing the above code snippet the obtained canvas screen is as shown in below Figure. 6.3.

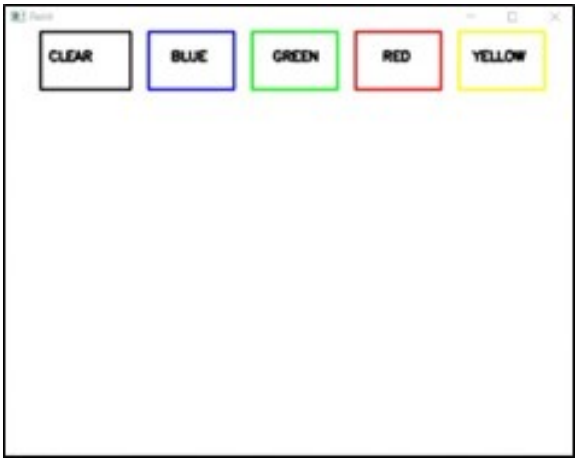


Figure 6.3: Canvas Screen

5. **Hand/Object Tracking and Detection:**

6. **Hand Tracking and Detection:**
   The Mediapipe hands module is initialized for hand tracking. The max_num_hands parameter is set to 1 for single-hand tracking as shown in the code snippet given below. Also utilized the hands model from Mediapipe to detect hand landmarks in each video frame. Then it Processes and draws hand landmarks on the video frame using the mpDraw.draw_landmarks function.

```
mpHands = mp.solutions.hands
hands = mpHands.Hands(max_num_hands=1, min_detection_confidence=0.7)
mpDraw = mp.solutions.drawing_utils
```

7. **Object Tracking and Detection:**
   The code starts by creating trackbars for adjusting upper and lower HSV (Hue, Saturation, Value) values. These values define the color range to be detected. A mask is created by applying an HSV range filter using the upper and lower values obtained from the trackbars. Contours are found in the mask, and the largest contour is selected. The center of the detected object is calculated, and based on its position, the color is identified. Drawing points are then appended to the corresponding deque. This is done as shown in the code snippet below.

```
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_APPROX_SIMPLE)
```

8. **Drawing on Canvas:**
   Determined the drawing color based on hand position and user input. It checks the position of the hand center in different regions and updates the colorIndex accordingly. Used deques (bpoints, gpoints, rpoints, ypoints) to store points drawn by the user in different colors. If a new color is selected or a specific gesture is detected, a new deque is created for that color as shown in code snippet below.

New deques (bpoints, gpoints, rpoints, ypoints) are appended when a specific condition (thumb position relative to the hand center) is met. Drew lines on both the video frame and the canvas based on these points. The lines are drawn using the cv2.line function between consecutive points for each color, resulting in both the live video frame (frame) and the canvas (paintWindow) being updated with the drawn lines.

```
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).
```

9. **User Interaction:**
   Allowed user interaction with the canvas by selecting different colors and clearing the canvas. Color selection is based on the user's hand position in specific regions of the frame. If the hand is positioned over a designated region, the colorIndex variable is updated accordingly. Canvas clearing is triggered when the hand is positioned over the "CLEAR" button region. If the hand is detected in this region, the canvas is reset by clearing the arrays used to store drawing points.

10. **Main Loop:**
    Created a main loop that continuously captured frames from the webcam as shown in the code snippet below. Processed hand landmarks, updated the canvas based on user input, and displayed the output frames. ret is a while loop that continues executing as long as the variable ret is True. The ret variable is typically used to check if the webcam is successfully reading frames. Inside the loop, the cap.read() function is used to read each frame from the webcam. The function returns two values - the first one is a boolean ret indicating whether the frame is successfully read, and the second one is the actual frame (frame).

```
while ret:
# Read each frame from the webcam
ret, frame = cap.read()
cv2.imshow("Output", frame)
cv2.imshow("Paint", paintWindow)
```

11. **Exiting the Program:**
    Program exit is triggered by pressing the 'q' key. The below given code snippet checks if the 'q' key is pressed. If the 'q' key is pressed, the loop will exit, and the program will stop capturing frames.

```
if cv2.waitKey(1) == ord('q'):
break
```

Chapter 7

Testing

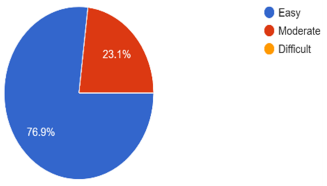
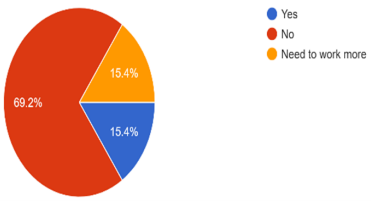
7.1 Test Cases

Software testing is a significant stage in the software development lifecycle that is crucial in guaranteeing an application's quality, functionality, and stability. The basic objective of software testing is to identify and fix faults, ensuring that the application satisfies defined criteria and delivers a seamless user experience. In this scenario, we used functional testing. Functional testing is a type of testing that confirms that each function of a software programme operates in accordance with the requirements and specifications. For this, we initially displayed and trailed the programme to 13 individuals, following which we sketched down some questions based on their experience with it. Based on these questions, the users rated the programme and recommended some changes.

Table 7.1: Testing Questions

Sr.No	Sketched Testing Questions	Option 1	Option 2	Option 3
1	Would you find easy to use??	Easy	Difficult	Moderate
2	Would you find easy to use?	Easy	Moderate	Difficult
3	How effective was the user interface, in your opinion?	Excellent	Moderate	Difficult
4	Did you need the assistance of a technical expert to use the system?	Yes	No	-
5	Regarding the drawing distance, how at ease are you?	Excellent	Moderate	Difficult
6	Do you prefer using the Air Canvas Application over Traditional Digital writing pads?	Yes	No	-
7	Do you think most people would soon learn how to use the system?	Yes	No	-
8	Would you like to make frequent use of the system?	Yes	No	-
9	How efficient would you find it to select the colour selections and erase the drawable?	Excellent	Moderate	Difficult

As per the above questions shown in Table 7.1, we have observed the below analysis given in Table 7.2:

Sr.No	Questions	Sr.No	Question
1.		2.	

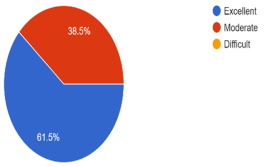
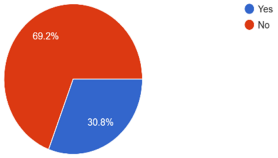
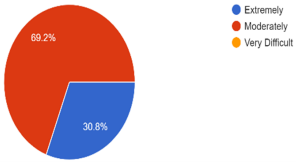
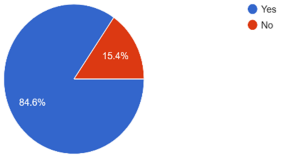
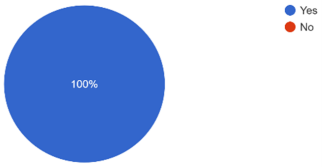
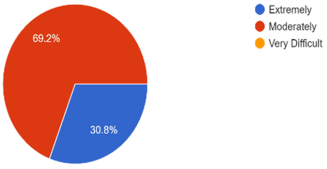
3.		4.	
5.		6.	
7.		8.	

Table 7.2: Analysis observation chart

One significant conclusion is that increased accuracy is something that everyone desires to see. Users frequently mention this as an area that requires attention, indicating a potential learning curve and a barrier to frequent use. Some users think that having technological skill is required, highlighting the need for a more user-friendly system that can accommodate users with varying degrees of technical proficiency. While the Air Canvas Application has received excellent comments, overcoming concerns with system complexity and accuracy, as well as extending creative possibilities such as colour choices and canvas size, might help to improve the overall user experience. The positive traits, such as simplicity of use and user interface effectiveness, create a solid foundation for developers to construct an even more user-friendly and feature-rich programme.

Chapter 8

Deployment

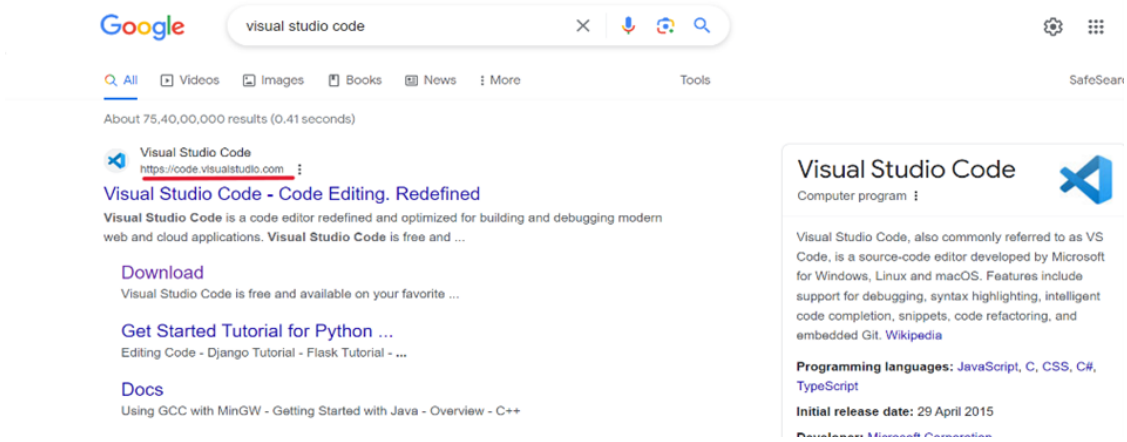
8.1 Readme File

- Install the Visual Studio code editor for the website **"https://code.visualstudio.com/".**
- Install Python programming from the website **"https://www.python.org/"** , preferably 3.10 or above.
- Install packages of OpenCV **"pip install opencv-python"**, MediaPipe **"pip install mediapipe"** & Numpy **"pip install numpy"** in terminal.

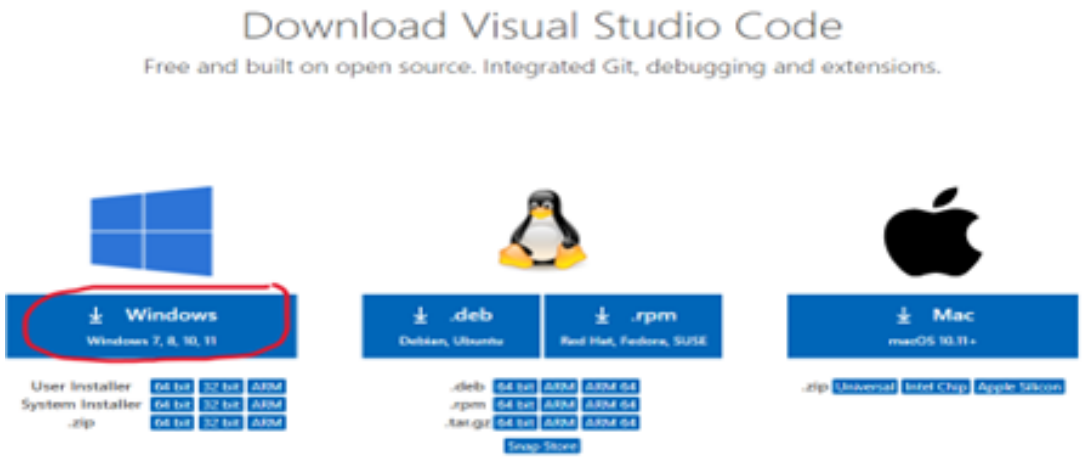
8.2 User Manual

8.2.1 Download & Package installation:

1. **Visual Studio Code Installation:**
   • **Step 1:** Visit the Visual Studio Code official website using any web browser such as Google Chrome, Microsoft Edge, and so on.

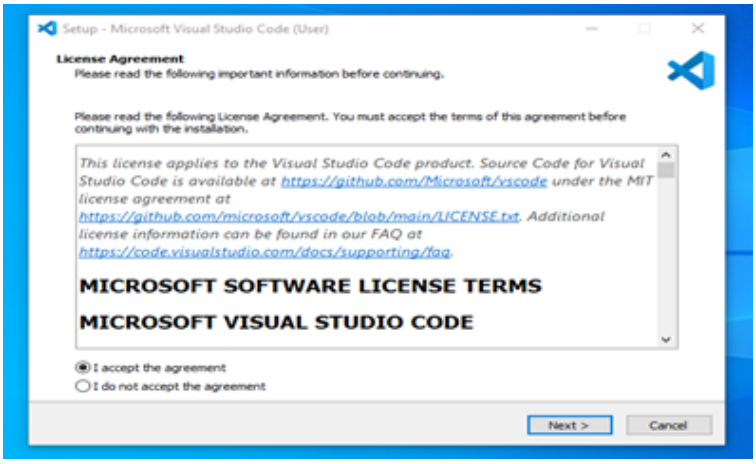


• **Step 2:** To begin the download of the Visual Studio Code Application, click the "Download for Windows" button on the website .

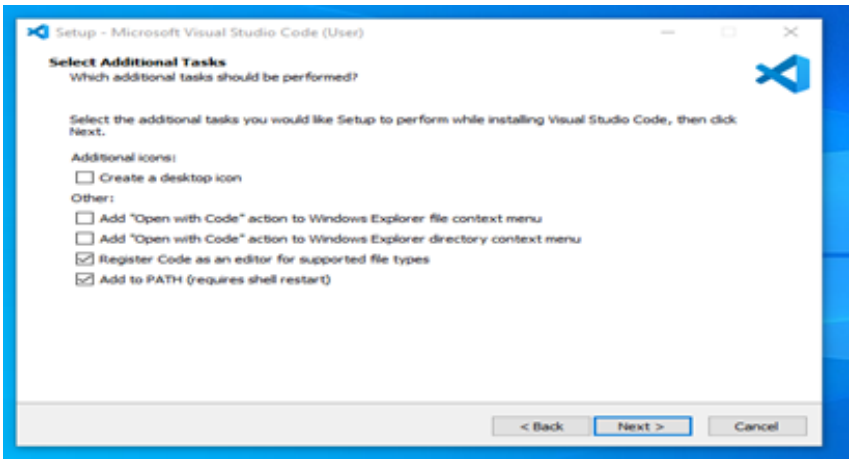


• **Step 3:** When the download is complete, the Visual Studio Code icon appears in the downloads folder. To begin the Visual Studio Code installation, click on the installer icon.

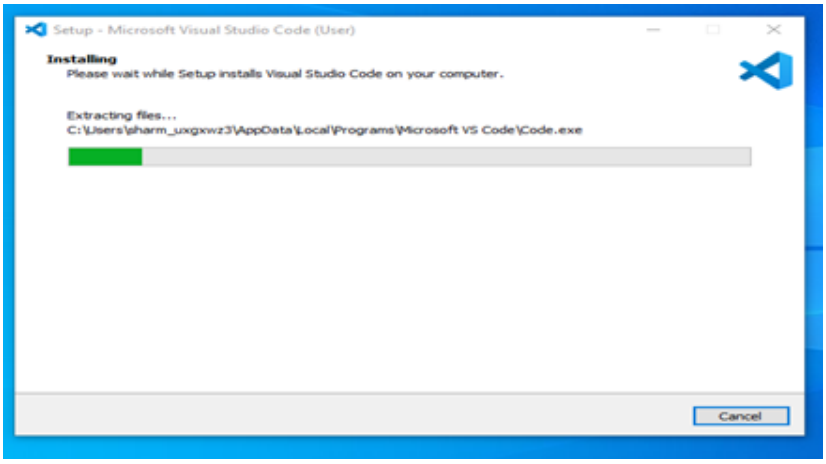
• **Step 4:** When the Installer launches, it will prompt you to accept the Visual Studio Code terms and conditions. Click I accept the agreement and then Next.



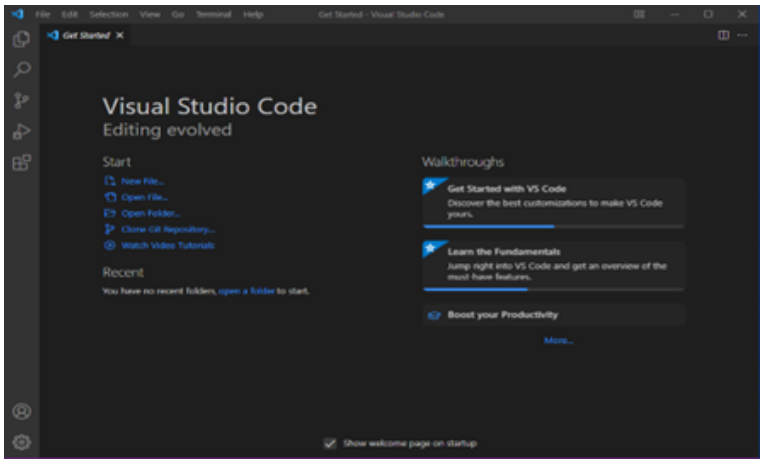
• **Step 5:** Select the location information to launch Visual Studio Code. It will then request that you browse the location. Then, press the Next button.



• **Step 6:** It will take about 1 minute to install Visual Studio Code on your Laptop once you click Install.

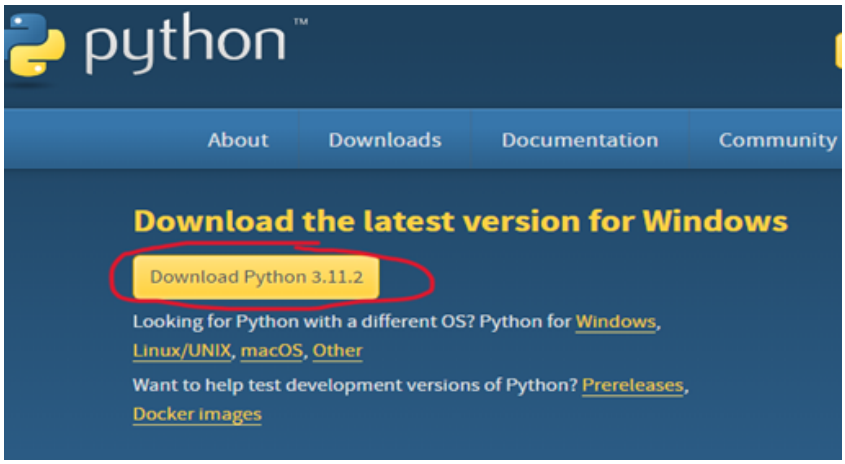


• **Step 7:** The Visual Studio Code window correctly opens following the previous step. You may now create a new file in the Visual Studio Code window and select a language to begin your programming journey!!

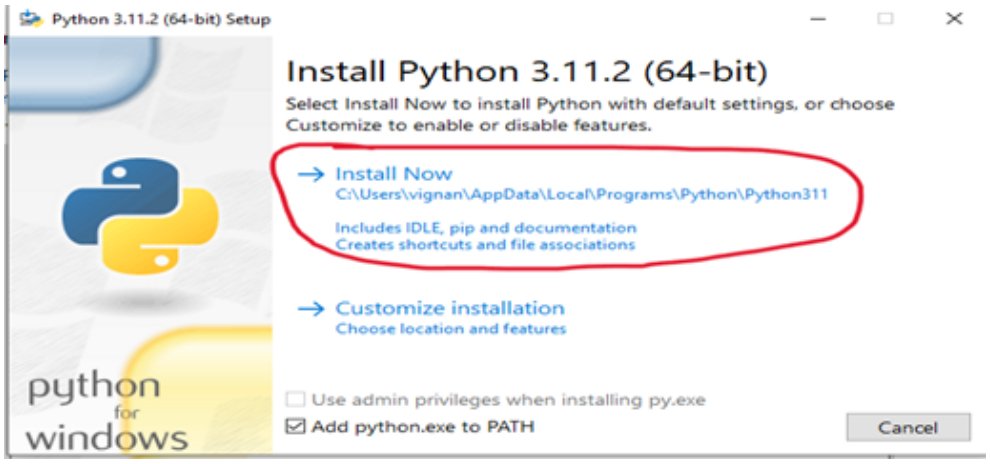


2. **Install Python:**

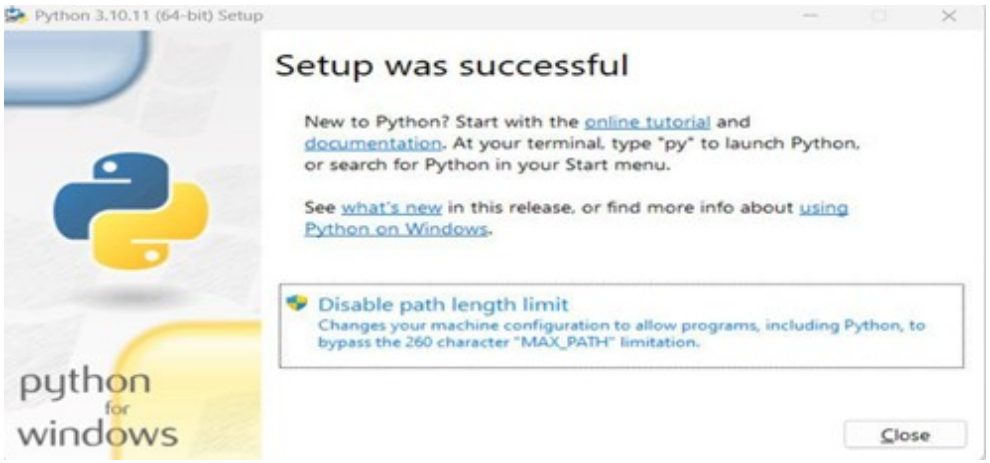
   • **Step 1:** Visit the python official website "https://www.python.org/downloads/". Find a dependable version of Python 3, preferably version 3.11.2.



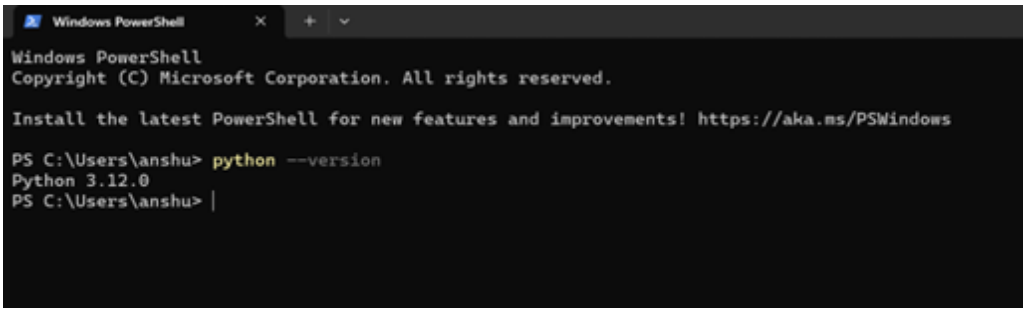
   • **Step 2:** Once you have downloaded the installer, double-click the.exe file, such as python-3.11.2 -amd64.exe, to launch the Python installation. Select the option to Install the launcher for all users by checking the relevant checkbox. By clicking the Add python.exe to PATH checkbox, users will be able to run Python from the command line.



• **Step 3:** After completing the setup. Python will be installed on your Windows machine. A successful message will be displayed. After Python gets completely installed, minimise the window.

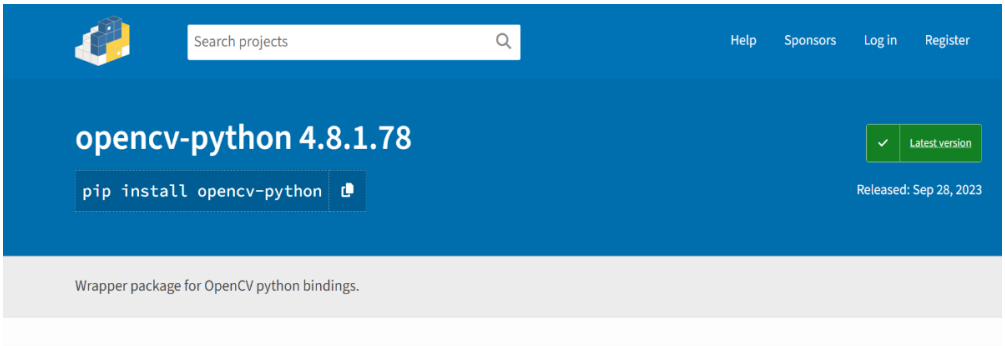


• **Step 4:** To go to the command line, go to the Start menu and type "cmd" into the search box. On Command Prompt, then type Python – version . Now Python is successfully installed on your computer.

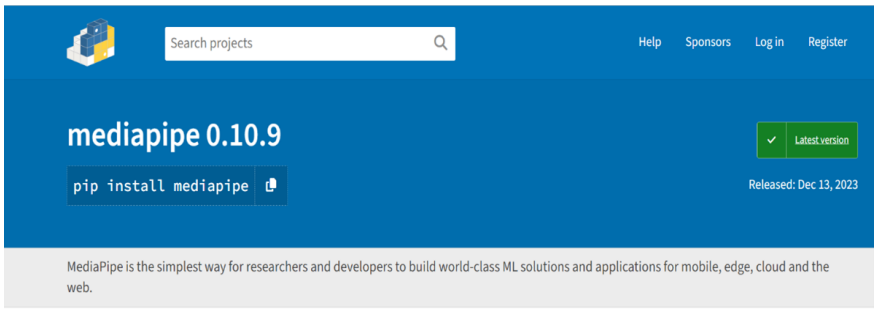


3. **Install Python Libraries:**

   – **OpenCV :**
   OpenCV is a Python open-source computer vision library used in artificial intelligence, machine learning, facial recognition, and other applications.
   Pip command – **pip install opencv-python**

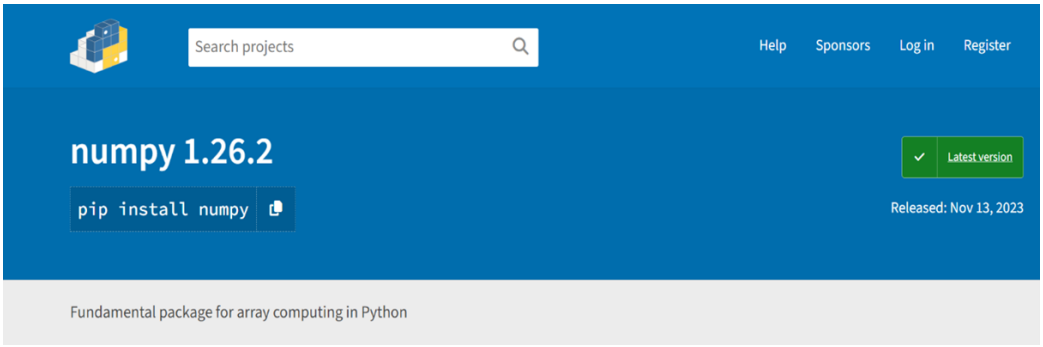


   – **MediaPipe:**
   Google's Mediapipe is a cross-platform library that delivers great ready-to-use machine learning solutions for computer vision workloads.
   Pip command – **pip install mediapipe.**



   – **Numpy:**
   NumPy is a Python library for manipulating arrays. It also has functions

for working with linear algebra, the Fourier transform, and matrices.
Pip command – **pip install numpy.**



4. **Convert Your Python Script to an Executable File:**

   - Python package that converts Python scripts to executable files is auto-py-to-exe. It is based on PyInstaller but has a Graphical User Interface (GUI) to make it easier to use.
   - **Step 1:** To use auto-py-to-exe, you need to install it first using pip: pip install auto-py-to-exe. Then, you need to run the following command to launch the GUI: auto-py-to-exe.
   - **Step 2:** This will open a window that looks like this-
   - **Step 3:** You may pick your script here by clicking the Browse button next to Script location. You may also pick whether you want a single-file or a one-folder executable by selecting One File or One Directory under Output Options.
   - **Step 4:** Other parameters, such as the icon, name, and terminal window, can be changed by selecting the Advanced tab and altering the fields under Additional Files or Window Based Options.
   - **Step 5:** Once you have finished configuring your options, click the Convert.py to.exe button at the bottom of the window. This will start the conversion process and display the results in the Console tab.
   - **Step 6:** When the conversion is complete, your executable file will be located in the output folder provided in Output Options. You may then run it or share it with others.

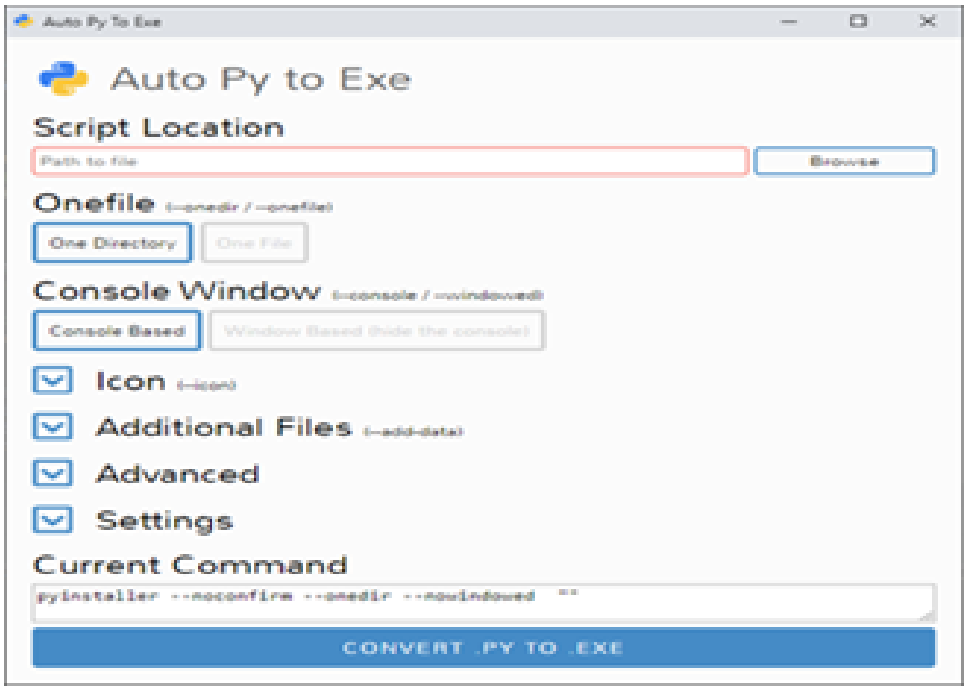


Figure 8.1: GUI to convert .py to .exe

8.2.2 Output Snapshots:

– **Step 1:** When we launch the.exe file, two panels appear as shown in Figure 8.2
  • The canvas Screen
  • The output Screen

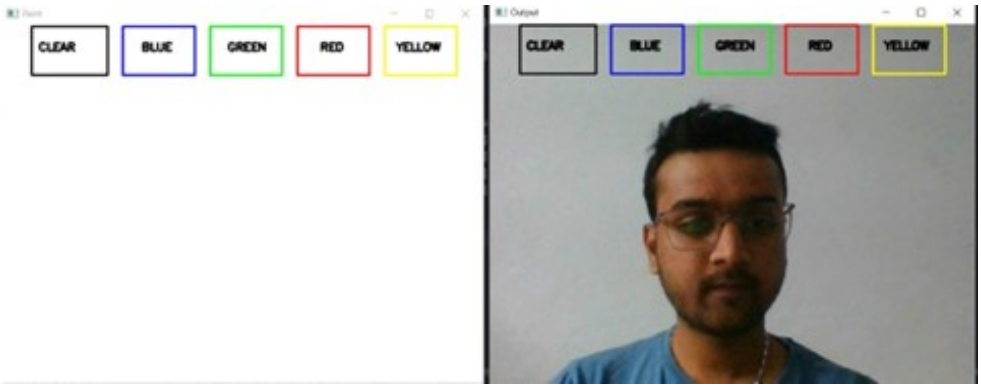


Figure 8.2: The canvas screen & output screen

– **Step 2:** In the output screen, our hand tips are recognised, and we may draw where the colour blue is initialised as shown in Figure 8.3.

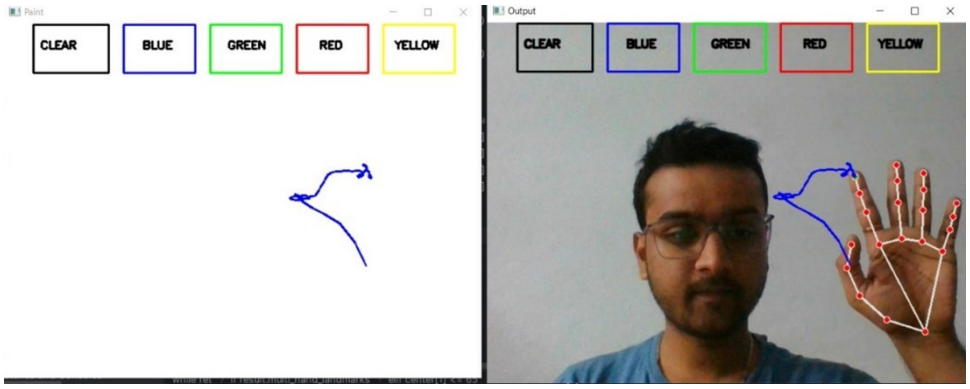


Figure 8.3: Hand detected with blue colour initialised

– **Step 3:** We may also modify the colour by bringing our index finger close to the green colour box and drawing anything we wish as shown in Figure 8.4.

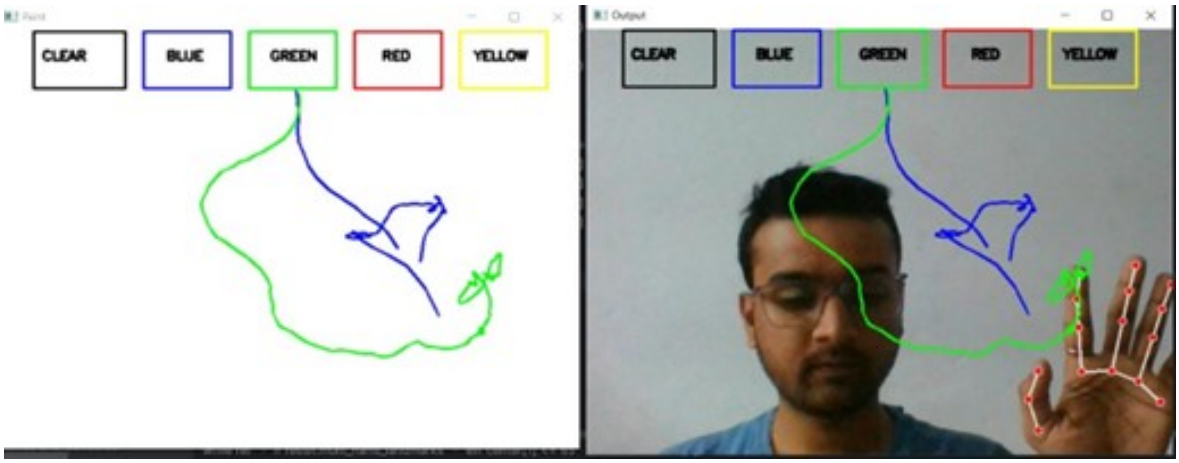


Figure 8.4: Choosing Green colour

– **Step 4:** By placing our index finger near the colour boxes on the output screen, we can flip the colour from red to green and vice versa as shown in Figure 8.5.

– **Step 5:** When we place our detected finger tip near the clear button on the output screen, everything written on the canvas & output screen is erased and a fresh clear screen is displayed as shown in Figure 8.6.

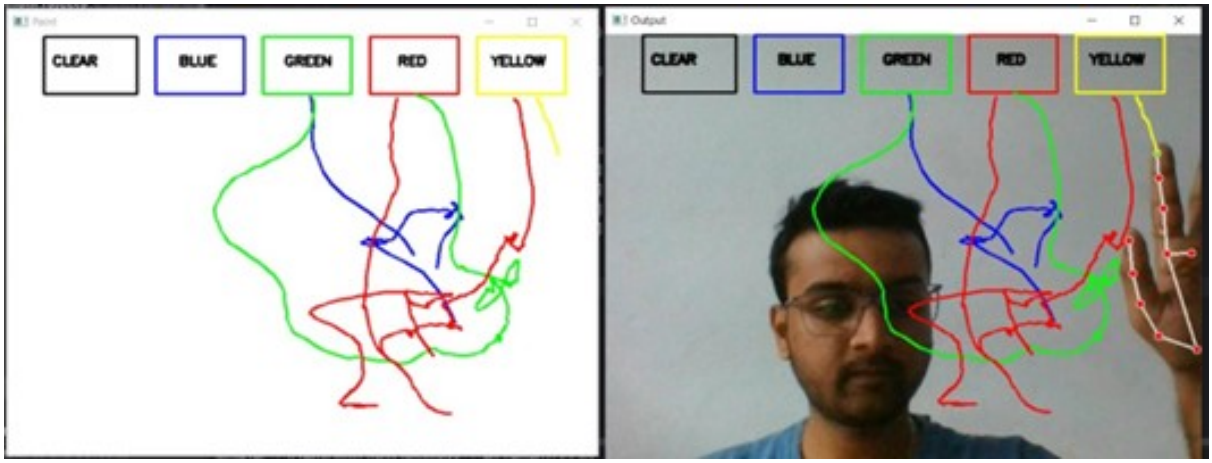


Figure 8.5: Flipping the colours from green to red & vice versa

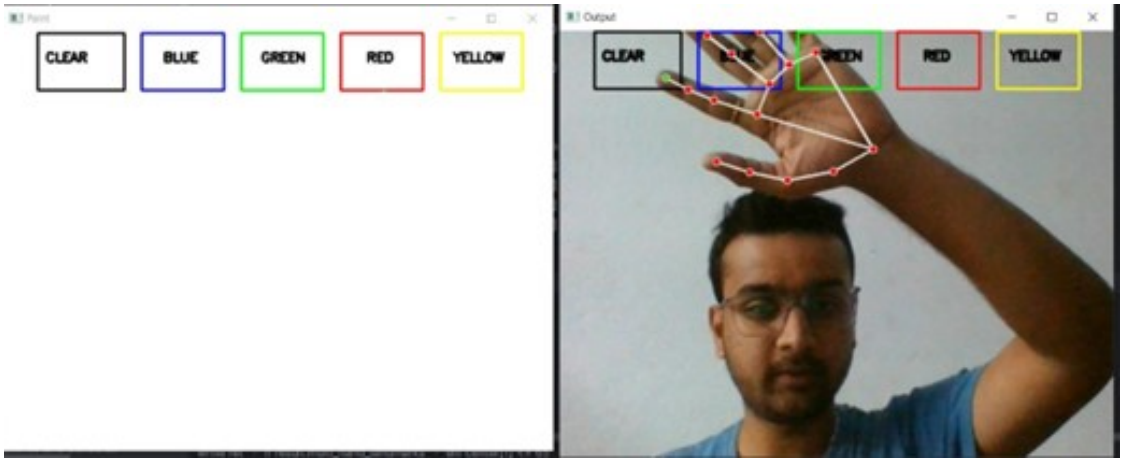


Figure 8.6: Erasing the content written on canvas as well as on output screen

– **Step 6:** By pressing Ctrl + S on the keyboard, the canvas screen in .png format as shown in Figure 8.7.

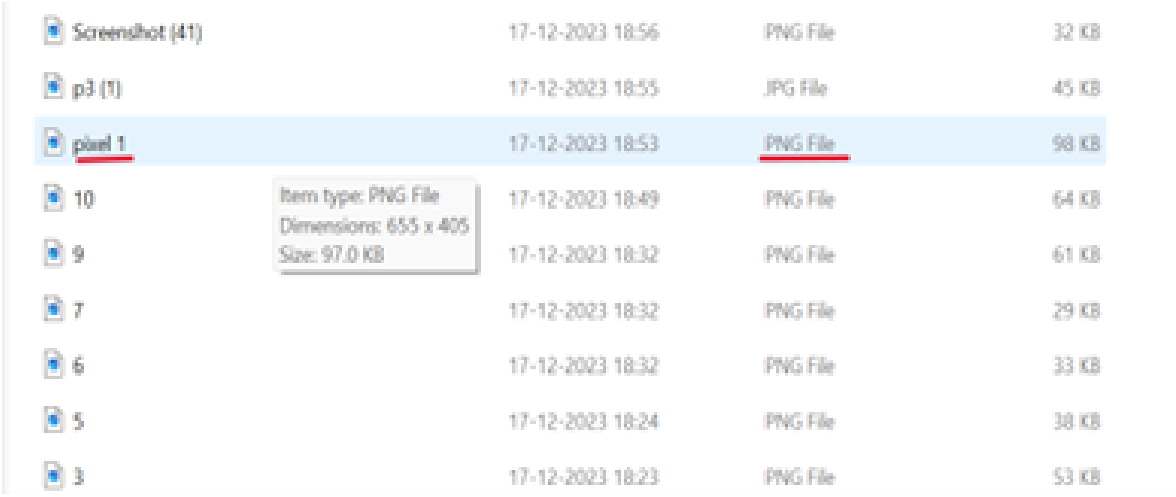


Figure 8.7: Saving the textual materials in .png format

Chapter 9

Results and Discussion

Digital writing pads are expensive, not portable, require constant charging, and limit users to limited areas. So, to overcome these shortcomings in digital writing pads we have identified different detection methods i.e. object detection method and hand tracking method and selecting the best one. The object detection method focuses on color detection and tracking using the HSV (Hue, Saturation, Value) color space and the hand tracking method uses OpenCV and Mediapipe library to implement this functionality. It involves video capturing, frames preprocessing, hand/object tracking, selection of drawing functionalities, canvas interaction, drawing on canvas, visualization, saving the screen and user interaction.

These are some of the comparative analyses between hand tracking module and object tracking module, we have done in the view of different aspects as shown in the Table 9.1.

After conducting thorough tests and evaluations, it became evident that the air canvas application, utilizing OpenCV and Mediapipe for hand tracking, outperformed an object detection approach.

Table 9.1: Comparative analysis of Hand Tracking and Object Tracking modules

Aspect	Hand Tracking Module	ObjectTracking Module
User Experience	Provides an immersive, natural interaction with hand gestures.	Offer a less intuitive experience, focusing on object tracking rather than natural hand movements.
Precision and Accuracy	High precision and accuracy in capturing detailed hand movements.	Precision may vary based on object characteristics, potentially lacking fine detail capture.
Responsiveness	More responsive, enabling real time tracking for immediate interaction.	Response time is influenced by object characteristics, leading to delays.
Implementation Simplicity	Relatively simpler implementation, adapting well to the natural hand interaction.	Involve additional complexities in mapping object movements to user interactions.
Gesture Recognition	Facilitates diverse gesture recognition through expressive hand movements.	Gesture recognition limited to the movements of tracked objects, potentially less diverse.
Overall Preference	Preferred choice, positively impacting user experience, precision, and implementation simplicity.	Less preferred due to potential challenges in achieving natural hand interaction and simplicity.

In conclusion, Hand tracking is the preferred choice guided by the positive impact on user experience, precision, accuracy, responsiveness, implementation simplicity, and gesture recognition for applications prioritizing natural and intricate hand interaction. Also, we have compared the OpenCV air canvas application with the existing technologies. We have made a comparative analysis of air canvas application and traditional writing mediums as given in below Table 9.2.

Table 9.2: Comparative analysis of existing technologies

Aspect	**Traditional Writing Mediums (Chalk, Duster, Blackboard)**	**Digital Writing Pad**	**Air Canvas Application**
User Interaction	Requires physical contact with chalk and blackboard; limited interactivity.	Offers digital interactivity but typically limited to the writing pad's surface.	Provides immersive, gesture based interaction without physical contact, enhancing user engagement.
Flexibility	Fixed medium; limited flexibility in terms of space and content.	Offers limited space to roam around due to wired connection.	Provides freedom of movement and a virtual canvas, offering unlimited space for creativity.
Erasing and Editing	Erasing can be messy with chalk and duster; limited editing capabilities.	Easy erasing and editing capabilities, but may still have limitations.	Effortless digital erasing and editing, enhancing user convenience and workflow.
Multifunctionality	Primarily designed for writing.	Expanded functionalities but may be constrained by the writing pad's features.	Multi-functional, supporting diverse drawing functionalities, color, and tools.
Resource Requirements	Requires resources like chalk and maintenance for blackboard cleaning.	Requires power for the writing pad; occasional replacement of pen nibs.	Relies on standard computing hardware with minimal maintenance. No resources needed.
Digital Advantages	Lacks digital advantages such as saving, sharing, or integrating with other digital tools.	Offers some digital features, but the extent may vary.	Fully digital, allowing easy saving, sharing, and integration with 54 other digital platforms.
User Experience	Traditional and may feel outdated in a digital-centric environment.	Blends traditional and digital aspects but may not provide a fully immersive experience.	Modern and immersive user experience, aligning with digital interaction expectations.
Cost Considerations	Relatively low-cost materials (chalk, duster, blackboard).	Moderate cost, varying based on the features of the writing pad.	Initial investment in technology but potential cost savings over time without resource expenses.

Based on the above analysis, the Air Canvas Application stands out as a modern and immersive solution, offering enhanced user interaction, flexibility, and digital advantages compared to traditional writing mediums and digital writing pads. While it in-

volves an initial investment, it provides a more versatile and cost-effective platform for creative expression and collaboration. By leveraging gesture-based interaction without physical contact, it enhances user engagement. The application provides freedom of movement with a virtual canvas.

Results:

After implementing the object detection module we have got the output as given below in Figure 9.1.

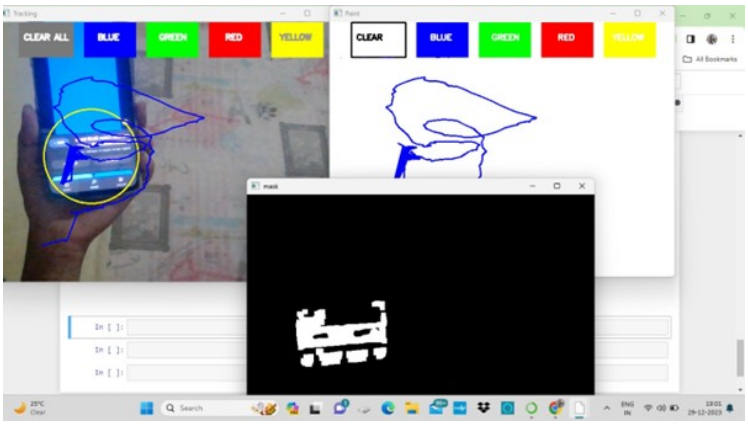


Figure 9.1: Object Detection Output Screens

After implementing the hand detection and tracking module we have got the output as given below in Figure 9.1.

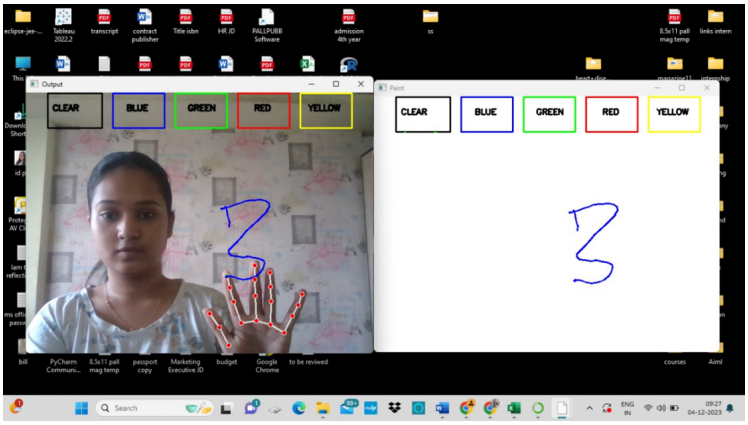


Figure 9.2: Hand Detection Output Screens

Chapter 10

Conclusion and Future Work

10.1 Conclusion

The OpenCV air canvas project provides an engaging and interactive drawing experience by leveraging computer vision techniques. By detecting and tracking a marker's movement in realtime, users can draw on a virtual canvas displayed on their screens. The project demonstrates the capabilities of OpenCV in image processing, object detection, and tracking. It also showcases the potential of computer vision in artistic and creative applications. The user is able to choose colour of their choice, clear the screen whenever they wish to and is able to save the canvas screen in the image format.

10.2 Future Work

1. Improved Hand Detection: Enhance the hand detection to handle different lighting conditions, complex backgrounds, distance between hand and camera, etc.

2. Multi-User Support: Implement multi-user support to allow multiple users to draw simultaneously on the canvas. This could involve incorporating multi-hand tracking and managing multiple hand tracing on canvas screen.

3. Gesture Recognition: Explore gesture recognition techniques to enable users to perform actions like changing color, or activating eraser mode through predefined hand gestures.

4. Collaboration and Sharing: Enable collaborative drawing by allowing multiple users to connect and draw together on a shared canvas. Implement features to share the canvas screen.

5. Integration with Other Platforms: Explore integrating the air canvas project with other platforms or devices, such as virtual reality (VR) headsets or interactive displays, to extend the project's reach and possibilities.

By focusing on these future work areas, the OpenCV air canvas project can continue to evolve, providing an immersive and dynamic drawing experience while pushing the boundaries of computer vision and interactive art applications.

Chapter 11

References/ Appendices /Bibliography

[1] Saoji, S. U., Dua, N., Choudhary, A. K., & Phogat, B. (2021). Air canvas application using OpenCV and NumPy in Python. *International Research Journal of Engineering and Technology (IRJET)*, 8(08).

[2] Nitin Kumar, R., Vaishnavi, M., Gayatri, K. R., Prashanthi, V., & Supriya, M. (2021, April). Air writing recognition using MediaPipe and OpenCV. In *International Conference on Ubiquitous Computing and Intelligent Information Systems* (pp. 447-454). Singapore: Springer Nature Singapore.

[3] Rai, P., Gupta, R., Dsouza, V., & Jadhav, D. (2022, October). Virtual Canvas for Interactive Learning using OpenCV. In 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT) (pp. 1-5). IEEE.

[4] Kumar, Abhishek, Aman Kumar Jha, Amrit Shandilya, Saurav Kumar Dubey, and A. Chaitra. "VIRTUAL AIR BOARD USING PYTHON."

[5] Gulati, S., Rastogi, A. K., Virmani, M., Jana, R., Pradhan, R., & Gupta, C. (2022, February). Paint/Writing Application through WebCam using MediaPipe and OpenCV. In 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM) (Vol. 2, pp. 287-291). IEEE.

[6] Bhargav, D. V., MR, M. N., PV, K. B., & Rekha, P. (2021, December). A Simple Approach for Scripting in Air and Display using Computer Vision. In 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC) (pp. 1-4). IEEE.

[7] Chandhan, Tamalampudi Hemaa, Nalin Raj, Neelam Nanda Kishore Reddy, and Mohammed Zabeeulla AN. "Air Canvas: Hand Tracking Using OpenCV and Medi-

aPipe." In 1st-International Conference on Recent Innovations in Computing, Science & Technology. 2023.

[8] Gupta, Aryan, Naman Chawla, Rachna Jain, Narina Thakur, and Ajantha Devi. "Gesture-Based Touchless Operations: Leveraging MediaPipe and OpenCV." NEU Journal for Artificial Intelligence and Internet of Things 1, no. 2 (2023).

[9] Agrawal, S. C., Tripathi, R. K., Bhardwaj, N., & Parashar, P. (2023, July). Virtual Drawing: An Air Paint Application. In 2023 2nd International Conference on Edge Computing and Applications (ICECAA) (pp. 971-975). IEEE.

[10] Vamja, Y., Patil, C., Matode, S., Kundale, N., & Tekade, P. (2023, February). Survey on Virtual Paint Board Using Computer Vision. In International Conference on Communication, Electronics and Digital Technology (pp. 675-684). Singapore: Springer Nature Singapore.

[11] TL, Mrs Nethravati, Rakshita L. Patil, S. Bhavana, Sairin Ray Choudhury, and S. Monisha. "VIRTUAL PAINTER USING ARTIFICIAL INTELLIGENCE AND OPENCV."

Activity Chart

Activity	Duration in Days	End Date	Month 2023										
			01	02	04	05	06	07	08	09	10	11	12
Team Formation project proposal/Problem identification	15	20/01/2023											
Title finalization and literature review	12	02/02/2023											
Self-study and topic research	20	20/04/2023											
Synopsis submission and presentation	15	09/05/2023											
System Requirements Specification	10	08/06/2023											
50% project completion	25	26/06/2023											
Project Phase-I evaluation	2	07/07/2023											
Participate in IEEE (Banglore Section) Technovation 2023	2	17/07/2023											
Sponsorship Proposal	5	19/08/2023											
Sponsorship Approval	7	26/09/2023											
Project Testing Phase	10	12/10/2023											
100% project completion	30	15/11/2023											
Participate in Quantum 2023	1	07/12/2023											
Report Preparation	10	10/12/2023											
Final report submission and presentation	1	30/12/2023											