A

Project Report

on

Virtual Interactive Board

submitted for partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

in

Computer Science

By

Vaibhav Singh (2000290120183)
Vaibhav Mittal (2000290120182)
Sarthak Srivastava (2000290120136)

Under the Supervision of

Mrs. Arushi Gupta
Assistant Professor

Department of Computer Science
KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

May 2024

DECLARATION

We hereby declare that this submission is our work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Signature of Student
Name: Vaibhav Mittal
Roll No.: 2000290120182

Signature of Student
Name: Vaibhav Singh
Roll No.: 2000290120183

Signature of Student
Name: Sarthak Srivastava
Roll No.: 20002901201

CERTIFICATE

This is to certify that the Project Report entitled “**Virtual Interactive Board”** which is submitted by **Vaibhav Mittal, Vaibhav Singh and Sarthak Srivastava** in partial fulfillment of the requirement for the award of degree B. Tech. in the Department of Computer Science of Dr A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidate’s own work carried out by him under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date:

Supervisor Signature

Mrs. Arushi Gupta
Assistant Professor
Department of
Computer Science

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the synopsis of the B.Tech Mini Project undertaken during B.Tech. Third Year. We owe a special debt of gratitude to Mr. Pardeep Tyagi (Assistant Professor), Department of Computer Science, KIET Group of Institutions, Delhi- NCR, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his/her cognizant efforts that our endeavours have seen the light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kumar Shrivastava, Head of the Department of Computer Science, KIET Group of Institutions, Delhi- NCR, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution to the completion of the project.

Signature of Student
Name: Vaibhav Mittal
Roll No.: 2000290120182

Signature of Student
Name: Vaibhav Singh
Roll No.: 2000290120183

Signature of Student
Name: Sarthak Srivastava
Roll No.: 2000290120136

TABLE OF CONTENTS

ABSTRACT

One of the most intriguing and difficult developments in the fields of image processing and pattern design recently has been the ability to write in the air. It primarily advances an automated process and can enhance the interface between a machine and a human in a variety of applications. New strategies and procedures that help in cutting processing time and offering high recognition efficiency and accuracy have been the focus of effort in a number of research fields. In the realm of computer vision, object tracking is viewed as a crucial task.

It entails first identifying the item, then following its motion from frame to frame, and then analysing the object's behaviour. We will follow the finger's movement using computer vision. It will be a potent technique of communication and a successful way to lessen the need for writing. As Everyone is aware that painters produce paintings on canvases. Consider the possibility that we could paint in the air merely by waving our hands.

Therefore, in this project, we'll use Python and OpenCV to create an air canvas. Computer vision, hand and real-time gesture control, air writing, and object identification are some related terms. Creating an air canvas using Python and OpenCV involves leveraging computer vision techniques to track hand movements in real-time and interpret them as drawing actions on a virtual canvas.

By combining elements like Computer Vision, Hand and Gesture Recognition, Object Identification, Real-Time feedback, developers can create an innovative and interactive air canvas system that enables users to draw and write in the air using hand gestures, opening up new possibilities for human-machine interaction and creative expression.

In essence, by enabling the act of writing in the air, this project not only showcases the potential of computer vision but also paves the way for a seamless fusion of human creativity and technological innovation.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S.No.	Abbreviations	Full Form
I	ML	Machine Learning
II	HSV	Hue/Saturation/Value
III	RGB	Red/Green/Blue
IV	LED	Light Emitting Diode
V	HCI	Human Computer Interaction

CHAPTER 1
INTRODUCTION

1.1 Introduction to Project

The emergence of the digital age has brought about a significant shift in the always changing field of artistic expression. Once a mainstay of the spread of culture, modern literature now finds itself in the midst of the emergence of digital art. Digital painting, interactive installations, and other kinds of art are all included in the broad category of digital art, which is defined by its production and distribution via digital media. Traditional artistic approaches, which were mostly based on tangible materials and analog processes, have had to be reevaluated in light of this change.

Traditional art has long been the basis of human creativity and expression. It includes visual, auditory, audio-visual, and audio-visual fantasy arts. For millennia, the use of pen and paper writing, chalk drawing on a chalkboard, and other manual techniques has been essential to artistic pursuits. But as technology develops, conventional and digital art are entwining more and more, each influencing and enhancing the other in a mutually beneficial connection.

Even with the rise of digital art, conventional methods are still respected and important in the creative community. Many modern artists combine the infinite possibilities of digital tools with the tactile properties of physical medium, incorporating old approaches into their workflow with ease. This combination of analog and digital methods emphasizes how creative activities are connected across media and how the creative process is always evolving.

There is a growing need for natural Human-Computer Interaction (HCI) solutions as the lines between digital and traditional art become more blurred. By bridging the gap between the intentions of the artist and the digital medium, these solutions seek to improve the creative experience. The possibilities of digital art creation tools are enhanced by methods like color detection, which makes it possible to identify particular hues within the HSV color space, and image segmentation, which makes it easier to mark pixels with shared features.

In real terms, to accomplish their goals, projects in this field frequently make use of strong computer hardware and software libraries. For example, a project may use a 64-bit operating system and a high-performance CPU, such as the Intel(R) Core (TM) i5 10500H, clocked at 4.0GHz, in addition to a sufficient amount of RAM to ensure seamless operation. Programming languages and software libraries, such NumPy, OpenCV, and Python, offer fundamental tools and frameworks for processing and manipulating images, allowing developers and artists to bring their creative ideas to life digitally.

1.2 Project Category

Digital art's primary objective goes beyond simple production; it also includes the investigation of novel interfaces that unite technology capabilities with human expression. Here, the creation of a hand gesture recognition system for digital writing is an innovative project at the nexus of technology, art, and human-computer interaction.

While writing instruments like keyboards, touchscreens, and styluses can produce digital art, our solution stands out for combining computer vision with hand gesture detection, all controlled by Python programming. A true human-machine interface may now be created thanks to the confluence of disciplines, allowing artists to effortlessly translate their gestures and movements into digital expressions.

This system is based on the integration of state-of-the-art technology. Real-time recognition and conversion of hand motions recorded by sensors or cameras into digital inputs is made possible by computer vision algorithms. The foundation of the program is Python, which offers a flexible and user-friendly framework for handling data processing, creating intricate algorithms, and enabling smooth interaction with other software libraries and frameworks.

Such a system has far-reaching ramifications not only for digital art but also for virtual reality, augmented reality, and human-computer interaction. This technology creates new opportunities for interactive experiences, cooperative projects, and intuitive and natural interactions with digital worlds.

Furthermore, the creation of a system for recognizing hand gestures highlights how digital art has developed into a dynamic, multidisciplinary subject. It emphasizes how crucial it is for technicians, engineers, and artists to work together to push the frontiers of innovation and creativity. Therefore, our system is a monument to the transformative power of interdisciplinary collaboration in influencing the future of human expression, as well as a significant achievement in digital art.

1.3 Objectives

1.3.1. Creation of a Hand Gesture Recognition System: The main goal of this project is to create and put into place a reliable system for hand gesture recognition that is especially suited for digital writing. Users will be able to interact with digital writing interfaces in an intuitive manner because to this system's usage of computer vision algorithms, which will precisely recognize and interpret hand motions.

1.3.2. Integration of Python Programming and Computer Vision: Creating a full human-machine interface also requires the smooth integration of Python programming with computer vision techniques. The research attempts to achieve real-time hand motion recognition and translation into digital inputs for writing applications by merging these disciplines.

1.3.3. Improving the Creative Experience: By offering a simple and natural method of digital expression, the initiative aims to improve the creative experience for both users and artists. The goal of the hand gesture detection system is to enable users to discover new creative and innovative opportunities by bridging the gap between traditional and digital art forms.

1.3.4. Investigation of Novel Interfaces: In addition to digital writing, the project's goal is to investigate how the hand gesture detection system may be used in other artistic projects including interactive installations, augmented reality, and virtual reality. The initiative intends to push the boundaries of human-computer interaction and artistic expression by examining innovative interfaces.

1.3.5. Encouraging interdisciplinary collaboration among artists, technicians, engineers, and programmers is one of the project's main goals. The project intends to use the combined knowledge and abilities of its team members to accomplish its goals and develop digital art and technology by bringing together a variety of viewpoints and areas of expertise.
Illustration of the Transformative Potential of Teamwork: The project's ultimate goal is to demonstrate the revolutionary potential of interdisciplinary cooperation in shaping the course of human expression. The project aims to encourage the next generation of artists, technologists, and innovators to push the limits of creativity and innovation by demonstrating the power of teamwork.

1.4. Structure of Report

In Chapter 1, the report begins with an overview of the project, highlighting the emergence of digital art in the context of the evolving landscape of artistic expression. It discusses the fusion of traditional and digital art forms, emphasizing the need for natural Human-Computer Interaction (HCI) solutions to bridge the gap between analog and digital mediums. The chapter also introduces the project's objectives, scope, and significance in addressing the evolving needs of the creative community.
Chapter 2 presents a comprehensive review of relevant literature related to the project. It explores existing research, methodologies, and technologies in the fields of digital art, HCI, and computer vision. The literature survey examines the historical evolution of artistic expression, the impact of technology on creative processes, and recent advancements in digital art creation tools. It also discusses key concepts such as color detection, image segmentation, and their applications in enhancing the creative experience.

In Chapter 3, the proposed system for bridging traditional and digital art through natural HCI solutions is introduced. The chapter outlines the project's conceptual framework, including the integration of computer vision techniques, hardware specifications, and software libraries. It presents an overview of the system architecture, highlighting its components, functionalities, and anticipated outcomes. Additionally, Chapter 3 discusses the rationale behind the chosen approach and its alignment with project objectives.

Chapter 4 focuses on requirement analysis and system specification for the proposed project. It details the functional and non-functional requirements, user needs, and technical constraints. The chapter defines the scope of the project, outlining specific features, functionalities, and performance criteria. Additionally, it presents a detailed system specification, including hardware and software requirements, data flow diagrams, and use case scenarios.

Chapter 5 delves into the implementation phase of the project, describing the practical steps taken to develop the proposed system. It discusses the software development process, coding methodologies, and integration of computer vision algorithms. The chapter also addresses challenges encountered during implementation and strategies employed to overcome them. Additionally, Chapter 5 provides insights into the iterative nature of software development and the evolution of the system architecture.

In Chapter 6, the focus shifts to testing and maintenance activities for the developed system. The chapter outlines the testing methodologies, including unit testing, integration testing, and user acceptance testing. It discusses the identification and resolution of bugs, performance optimization, and system refinement based on user feedback. Additionally, Chapter 6 emphasizes the importance of ongoing maintenance and support to ensure the long-term viability and effectiveness of the system.

Chapter 7 presents the results of the implemented system and provides a detailed analysis of its performance and effectiveness. It discusses key findings, observations, and insights gained from testing and user feedback. The chapter also compares the outcomes against predefined objectives and discusses implications for future development and refinement. In the final chapter,

Chapter 8, the report concludes with a summary of key findings, contributions, and implications of the project. It reflects on lessons learned, challenges encountered, and opportunities for future research and development. Additionally, the chapter outlines potential avenues for further exploration, including enhancements to the system, integration with emerging technologies, and broader applications in the field of digital art and HCI.

CHAPTER 2
LITERATURE SURVEY

2.1 Literature Review

Strong hand recognition system using the Kinect sensor. The suggested technique in makes advantage of the Kinect sensor 's depth and colour data to determine the hand's shape. With the Kinect sensor, gesture recognition presents a challenging issue to be acknowledged. This particular Kinect sensor only has a 640x480 resolution. It is effective for tracking huge objects like the human body. However, even a finger 's size is difficult.

The movement of a finger with an LED. the authors have suggested a technique where An LED is mounted on the user's finger, and a web camera is used to monitor it. Characters from the plot and the database are compared with one another. The returning alphabets are all those that match the sketched pattern. It needs an aiming red LED light source that is installed on the finger. Assume that only LED light is in view on the web camera and that no other red objects are nearby.

Improved Desktop User Interface An improved segmented desktop interface concept for interactivity was put forth in [5]. You can use desktop apps with this system, which uses a projector and a charged device (CCD) camera that you can use with your fingertips. Each component of this system completes a task that is wholly distinct from the others. Radial menus are selected with the left hand, while manipulative objects are chosen with the right hand. His method involves the use of an infrared camera. The algorithm creates search windows for the fingertip because it is computationally difficult to identify the fingertip.

2.2 Research Gaps

2.2.1. The method used for fingertip identification only works with your fingers; highlighters and other such gadgets are not compatible. Without a sophisticated equipment like a depth sensor, it is extremely difficult to recognize and identify an object like a finger from an RGB image.

2.2.2. The technology uses an RGB camera to begin writing and does not allow for pen up and down movement. The inability to detect depth prevents tracking of the pen's up and down movement. As a result, the fingertip 's complete trajectory is traced, and the resulting image is useless and unrecognizable to the model. Figure illustrates the distinction between handwritten and air written doodle, write, draw, etc., we use a lot of paper. However, in this system, we employ hand gesture detection along with machine learning algorithms and Python programming to produce a genuine human-machine interface.

2.2.3. Our initiative primarily focuses on addressing these significant issues:

2.2.3.1. Deaf people: Sign language is a useful tool for communication even if we sometimes take hearing and listening for granted. Without a translator in the middle, the majority of people in the world are unable to grasp their emotions.

2.2.3.2. One of the primary focuses of our virtual interactive board is to reduce paper wastage. By providing an alternative to traditional paper-based methods, this technology can contribute to saving trees and reducing the environmental impact associated with paper production and disposal.

2.2.3.3. Accessibility: to a wide range of users, including those from underserved communities and regions with limited access to traditional educational resources.

2.2.4. Control of real -time systems It takes a lot of coding attention to change a system's status in real- time using hand gestures. In order to completely master his plan, the user must also be familiar with numerous motions.

2.3 Problem Formulation

It's quite amazing to think that something as basic as waving a finger in the air can have such a significant impact on producing lifelike images. When incorporated into computer vision projects, this seemingly insignificant action assumes a new relevance as a crucial input for deciphering human intentions and interactions with digital environments.

Understanding and interpreting hand gestures in computer vision opens up a world of possibilities for improving creative expression and human-computer interaction. Through the acquisition and examination of hand movement dynamics in both space and time, computer vision algorithms are able to deduce a user's gestures and convert them into significant commands or inputs in a digital environment.

This feature has significant ramifications for visual expression and digital art. Imagine an artist creating digital landscapes with the dexterity and fluidity of real-world motions by manipulating virtual brushes and canvases using hand gestures. Artists are able to go beyond the constraints of conventional input devices and interact naturally and effortlessly with their creations thanks to gesture-based interaction's intuitive nature.

Furthermore, the incorporation of hand gestures into computer vision projects highlights how multidisciplinary modern technology is. Through the integration of concepts from several domains, including computer science, machine learning, and human-computer interaction, scientists and engineers can open up novel pathways for creativity and innovation.

The development of aerial web technologies has also increased the accessibility and adaptability of gesture-based interaction by allowing the detection and tracking of hand motions without the need for additional equipment or physical touch. This non-intrusive method not only makes the user experience easier, but it also broadens the range of contexts in which gesture recognition can be used, from immersive virtual worlds to interactive art installations.

Essentially, the combination of computer vision and hand gestures signifies a meeting point between the real and virtual worlds, wherein human motions are utilized as an effective means of modifying and sculpting digital media. As this technology develops further, it has the potential to completely transform not only the ways in which we make and use digital imagery, but also the ways in which we view and interact with the environment.

CHAPTER 3
PROPOSED SYSTEM

3.1 Proposed System

Ever wished you could just raise your finger in the air and catch your imagination? here, we'll use the webcam to record the movement of the colour marker to build an aerial canvas that can be used to sketch anything on it. A marker is an object of colour that is placed on the tip of the finger. In this project, computer vision techniques are aided by OpenCV. Though knowing the fundamentals can be used in any language that OpenCV supports, Python is the ideal choice due to its abundance of libraries and simple syntax. Here, colour detection and tracking are employed to accomplish our objective. Once a colour marker is found, a mask is created. It contains the subsequent phases of the product mask's morphological activity, including erosion and expansion.

3.2 Unique Features of The System

1. Has the ability to track any particular person's necessary colour pointer.

2. Users may easily and quickly draw with four distinct colours, and they can even switch them at any time.

3. To clean the board one piece at a time, choose the Clear option at the top of the display.

4. After the software has launched, there is no need to make contact with the computer.

3.2.1. Colour tracking of object at fingertips: To identify coloured objects, the webcam captured image must first be transformed to the HSV colour system. This code helps with colour space for colour tracking by converting the entering image to HSV space, which is fantastic. Now that we have the coloured item on our fingers, we will make track bars to align the HSV values into the necessary colour range. After track bars are configured, we obtain the track bars' current value and construct a range. A NumPy struct called range is used to transmit data into the cv2.inrange*()* function for additional processing. The coloured object and mask are returned by this function. This mask is incorporated in a black and white image that has white pixels where the necessary colour is needed.

3.2.2. Detecting the mask contour of the coloured object: It's time to locate the mask 's centre in order to draw the line after finding it in the air canvas. Here, we manipulate the mask morphologically in the code below to remove imperfections and make the outlines easier to see.

3.2.3. Drawing the Line by using the Contour position. The actual reasoning behind this Computer Vision project is that we'll execute a Python deque on a data structure. The contour's position on each frame will be stored in the deque, and we will use these points to create a line using OpenCV methods. This contour's location aids in our ability to decide whether to draw on the sheet or to press a button. Some of the buttons at the top of the canvas have been implemented; if the pointer lands in their area, it will assist in initiating their function. On the canvas made with OpenCV, there are four buttons. By dequeue, clear the entire screen. Red: Use the palette to change the ink's colour to red. Green: Use the palette to change the colour to green. Yellow: Make the hue yellow.

3.2.4. We will now draw all the points on the locations stored in the dequeue, with the corresponding colour.

CHAPTER 4
REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

4.1 Feasibility Study

Cost Analysis: The costs associated with hardware and software procurement, development efforts, and potential collaborations should be considered. While hardware costs are relatively fixed, software development costs may vary depending on the complexity of the project and the need for specialized expertise.

Return on Investment (ROI): The ROI for natural HCI solutions in bridging traditional and digital art lies in enhancing the creative experience for artists and potentially expanding the market for digital art creation tools. Increased usability and accessibility could lead to greater adoption and revenue generation for software developers and artists alike.

User Acceptance: The success of natural HCI solutions in bridging traditional and digital art depends on user acceptance and adoption. User testing and feedback loops during development are essential for ensuring that the system meets the needs and expectations of artists.

Training and Support: Providing adequate training and support for artists transitioning to digital mediums is crucial for operational success. Tutorials, documentation, and online communities can facilitate learning and troubleshooting.

Scalability: Natural HCI solutions should be designed with scalability in mind to accommodate future advancements in technology and changes in user preferences. Flexible architectures and modular design principles can support scalability and adaptation over time.

4.2 Software Requirement Specification

4.2.1 Functional Requirement

Operating System: Windows 10 or above

Client Script: Python

4.2.2 Performance Requirement

Processor: Any modern x86 based processor

Hard Disk: 10 MB

RAM: 8 GB

4.3 SDLC Model Used

The waterfall model is a sequential and linear approach to software development that divides the project lifecycle into distinct phases, with each phase dependent on the deliverables of the previous one. It is one of the earliest methodologies used for software development and follows a structured progression from requirements gathering to deployment and maintenance.

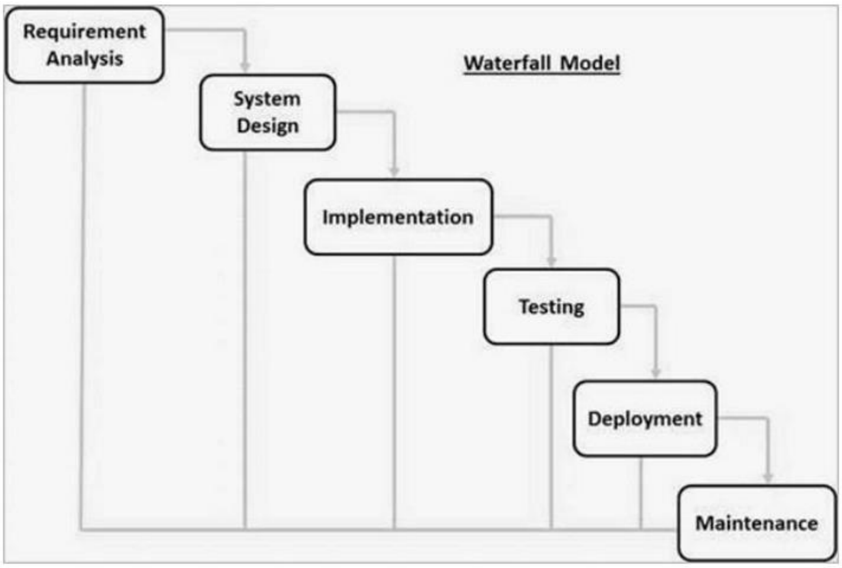

Figure 4.1 Waterfall SDLC Model

Reasons to use Waterfall Model:

Clear Requirements: If the requirements for the project are well-defined and unlikely to change significantly, the waterfall model can be effective. It relies on a thorough upfront analysis of requirements, making it ideal when there is a clear understanding of what needs to be delivered.

Resource Availability: If resources, such as skilled personnel or specialized tools, are readily available and unlikely to change throughout the project, the waterfall model can provide a straightforward framework for resource allocation and planning.

Minimal Iterations: If the project scope is well-defined and there is little need for iterative development or frequent changes based on user feedback, the waterfall model can be effective. It minimizes the need for revisiting and revising work completed in earlier phases.

4.4 System Design

The finger detection model of the system demands a data collecting. The main goal of the fingertip model is that a single-color stylus or air pen can be used to write with the fingertip sensor model for air writing. The technique, however, makes use of fingertips. Without the need for a stylus, this recorded movement is captured through drawing characters that individuals can write in the air. To create a list of coordinates, we employed deep learning algorithms to find the fingertip in each image. Technique for creating finger recognition dataset:

1. Video to Frames: This technique captures hand motions in two-second snippets captured at different angles. Diagram 3 then shows how these movies are divided into 30 different pictures, for a total of 2000 images. This dataset was manually labelled using Labelling. The best model with this dataset has an accuracy of 99%. However, because all thirty of the frames were produced from the same movie and location, the dataset is incredibly repetitious. Consequently, the model exhibits poor performance for discrete backgrounds that differ from the backgrounds found in the data set.

2. Take pictures against different backgrounds: We added a new dataset to address the drawback of the earlier method's lack of diversity. This time, we are conscious of the fact that controlling the system requires a few motions. Consequently, we gathered four distinct hand positions, which are displayed in Figure 4.

3. Training the finger recognition model: The dataset is split into training and development groups (85% to 15%) after it is ready and labelled. Our dataset was trained using pre -trained faster Single Shot (SSD) and RCN detector models. When it comes to precision, Fast RCN outperforms SSD. Information needs to be verified, so we do that. Two common detection modules one that categorizes regions and the other that makes suggestions are combined in the SSD.

The primary goal was to enable the model to distinguish between the four fingers' fingertips. As a result, the user can now control the system using the number of fingers that he represents. With the help of their index finger, they can now type rapidly. Two fingers can be used to convert that writing motion into electronic text, three fingers can add space, five fingers can be used to press the backspace key, four fingers can be used to make consecutive predictions, and three fingers can be used to select the first, second, or third prediction based on the requirement. Exit the prediction mode by displaying five fingers. There are roughly 1800 photos in this dataset. The prior model is intended to automatically label this dataset using this type of script. After that, we offer a new template and fix the incorrectly labelled photos. 94% accuracy is attained. Unlike the previous model, this one performs effectively in a variety of settings.

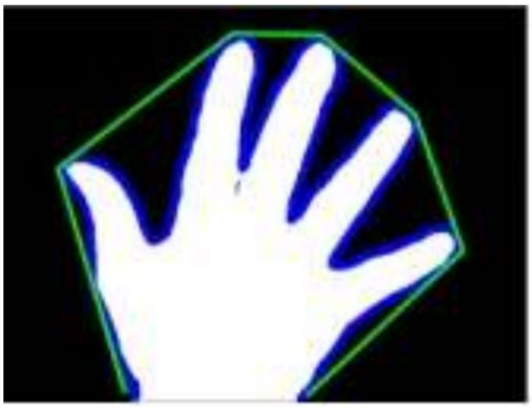

Figure 4.2 Detection of Finger Tips Using OpenCV

A. Play back the captured image and convert it to the HSV colour space (which makes colour easy to identify).

B. Create the canvas and embellish it with the matching-coloured buttons.

C. Modify the trace bar values in order to identify the colour highlight mask.

D. Mask pre-treatment using morphological adjustments.

E. Locate the contours, note the coordinates of the largest contour's centre, and continue storing them in an array for the upcoming frames. (Tables for doing sketches on the painting)

F. Lastly, sketch the points that are kept in the table.

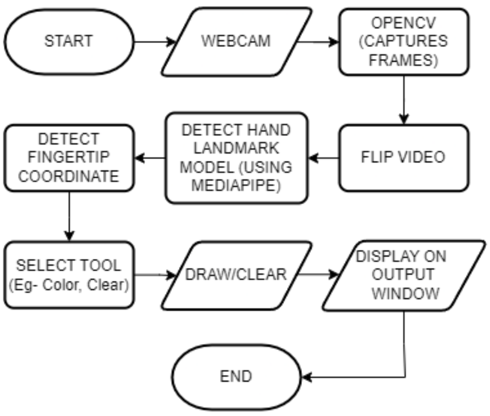

Figure 4.3 Flow Chart representation

4.4.1 Data Flow Diagram Level 0

At the highest level of abstraction, the DFD Level 0 provides an overview of the system and its interaction with external entities. It illustrates the flow of data between the main processes within the system.

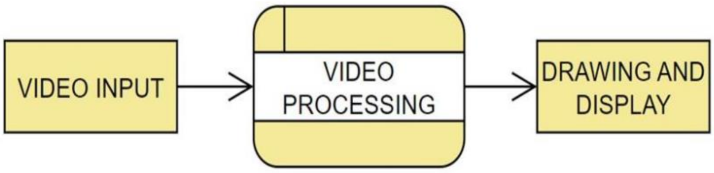

Figure 4.4 Level 0 data flow diagram of the program's primary function

Data Flow Diagram Level 1

DFD Level 1, also known as Level 1 Data Flow Diagram, dives deeper into the system compared to the high-level overview of a Level 0 DFD (Context Diagram) Elements of DFD level 1 processes, data flows, data stores and external entitites.

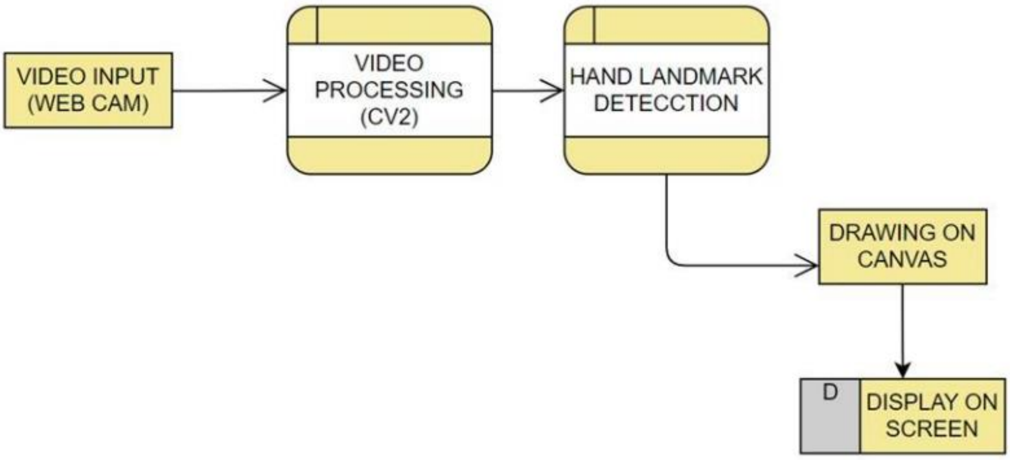

Figure 4.5 Level 1 data flow diagram of the program

4.4.2 Use Case Diagram

A use case diagram is a visual representation of how users (or external systems) interact with a system to achieve specific goals. It's a core concept in Unified Modeling Language (UML), a standard for software design.

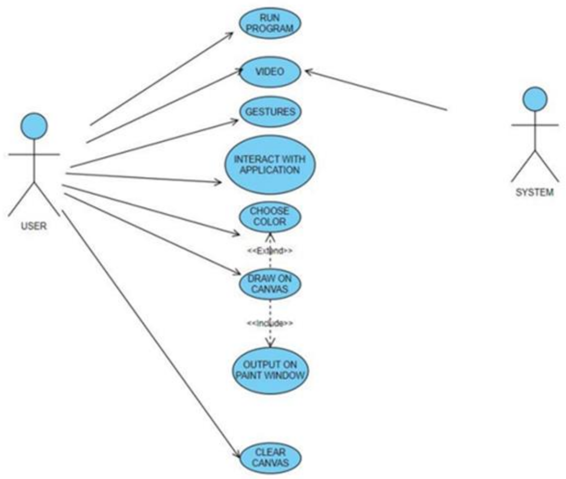

Figure 4.6 Use Case Diagram of the program

CHAPTER 5
IMPLEMENTATION

5.1 Introduction to Tools and Technologies used

Python 3.10 is the language chosen for this project, particularly for its suitability in the field of machine learning (ML) and computer vision tasks. Several factors contribute to Python's popularity and effectiveness in these domains:

5.1.1. Ease of Use: Python is known for its simplicity and readability, making it accessible to both beginners and experienced developers alike. Its clean syntax and straightforward structure allow for faster development and easier debugging, which is crucial in the iterative process of ML model building and experimentation.

5.1.2. Abundance of Libraries: Python boasts a vast ecosystem of libraries and frameworks specifically tailored for ML and computer vision tasks.

5.1.3. Community Support: Python has a vibrant and active community of developers, researchers, and enthusiasts who contribute to its ecosystem through open-source projects, forums, and online resources. This wealth of community support fosters collaboration, knowledge sharing, and continuous improvement, ensuring that developers have access to the latest advancements and best practices in ML and computer vision.

5.1.4. Flexibility and Extensibility: Python's flexibility allows developers to seamlessly integrate ML models and computer vision algorithms with other components of the project, such as data pipelines, user interfaces, and deployment frameworks. Its compatibility with various platforms and technologies further enhances its versatility, enabling developers to build scalable and interoperable solutions.

Overall, Python's combination of simplicity, robust libraries, community support, flexibility, and performance optimization makes it an ideal choice for developing ML and computer vision solutions.

5.2 Technologies Used

• **MediaPipe**

MediaPipe is an open-source framework developed by Google for building cross-platform applications involving perceptual computing tasks such as hand tracking, pose estimation, object detection, and facial recognition. It provides ready-to-use machine learning models and pipelines for various tasks, allowing developers to integrate complex computer vision functionalities into their projects with ease.

MediaPipe is designed for real-time performance and efficiency, making it suitable for applications requiring low-latency processing, such as augmented reality, virtual reality, and interactive installations. The framework offers both Python and C++ APIs, catering to developers with different preferences and requirements. MediaPipe's modular architecture and pre-trained models simplify the development process, enabling rapid prototyping and experimentation in the domain of computer vision and machine learning.

• **OpenCV**

OpenCV is a widely-used open-source library for computer vision and image processing tasks. It provides a comprehensive set of functions and algorithms for tasks such as image loading, manipulation, filtering, feature detection, object recognition, and more. OpenCV is written in C/C++ and has Python bindings, making it accessible to developers working in Python.

It is known for its high performance, efficiency, and cross-platform compatibility, with support for Windows, Linux, macOS, Android, and iOS. OpenCV is extensively used in various applications, including robotics, augmented reality, surveillance systems, medical imaging, and of course, digital art projects like the one described in the context of this discussion.

• **NumPY**

NumPy is a fundamental library for numerical computing in Python. It provides support for multidimensional arrays, mathematical functions, linear algebra operations, random number generation, and more. NumPy's array objects are more efficient than Python's built-in lists for handling large datasets and performing mathematical operations.

It serves as the foundation for many other Python libraries in the scientific computing and data analysis ecosystem, enabling seamless integration and interoperability.
In the context of machine learning and computer vision, NumPy is often used for data preprocessing, feature extraction, matrix operations, and statistical analysis.

CHAPTER 6
TESTING AND MAINTENANCE

6.1 Testing Techniques and Test Cases Used

Table 6.1: Test Cases

Test Case	Test Objective	Test Data	Expected Result	Actual Result	Pass /Fail
1	System Initialization	Internal Program Code	Webcam turns on and displays the OpenCV frame with buttons for colors and clear.	Webcam turns on and displays the OpenCV frame with buttons for colors and clear.	Pass
2	Hand landmarks recognition	Webcam feed containing hand of the user	Hand landmarks are detected by the mediapipe and the positions of the fingers are obtained	Hand landmarks are detected by the mediapipe and the positions of the fingers are obtained	Pass
3	Ability to Draw	Index finger tracking from	The Index finger is tracked accurately and the drawing is visible	The Index finger is tracked accurately and the drawing is visible	Pass

		webcam feed			
4	Ability to change colors	Index finger position on the color button	Different colors can be selected and the drawing with selected color is visible on the canvas	Different colors can be selected with index finger and the drawing with selected color is visible on the canvas	Pass
5	Ability to clear the canvas	Index finger position on the clear button	The user is able to clear the canvas by selecting ‘CLEAR’ button	The user is able to clear the canvas by selecting ‘CLEAR’ button with the index finger	Pass
6	Ability to move hand freely	Presence of individual index finger	The user is able to move hand freely by performing a pinching gesture and not being able to draw on the canvas while in this gesture	The user is able to move hand freely by performing a pinching gesture and not being able to draw on the canvas while in this gesture	Pass

7	Closing the program	Internal program code	The user being able to close the program by pressing the ‘q’ button on the keyboard ensuring the program is not running in the background	The user being able to close the program by pressing the ‘q’ button on the keyboard ensuring the program is not running in the background	Pass
8	Detecting multiple hands	Multiple hands on webcam feed	The program detects multiple hands and assigns a 3D landmark model to each of them	The program was not able to detect any hand after the initial hand	Fail
9	Visibility of hand	Position of the hand on webcam feed	The program detects the hand accurately no matter the position on the screen	The program fails to detect the hand if the angle of the hand is not exclusively facing the webcam in an upright position	Fail

Table 6.2: Decision Table

Conditions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Index Finger upright	False	True	False	True
Pinch Gesture	False	False	True	True
Output(e/a)	Invalid gesture	Draw/Select	Standby	Invalid gesture

CHAPTER 7
RESULTS AND DISCUSSIONS

7.1 User Interface

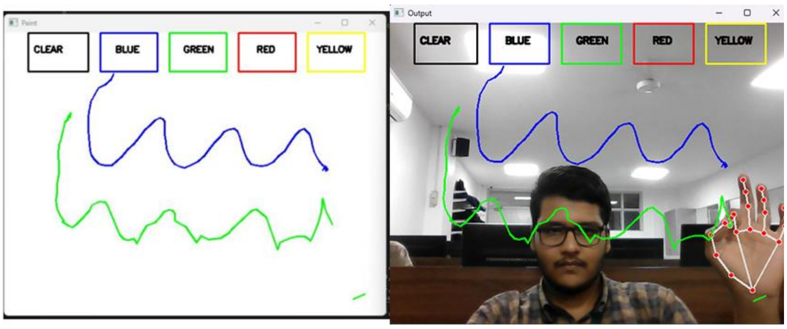

Figure 7.1 Drawing Functionality on A Virtual Canvas

User can draw in the air while their webcam is on, the left window represents the drawing on Virtual canvas and the right window represents the user and their actions being tracked 1:1 on both windows

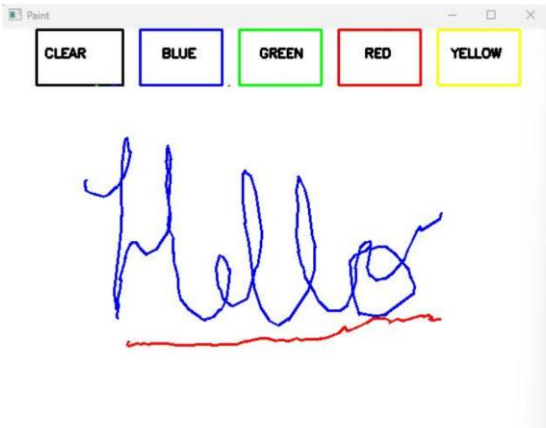

Figure 7.2 UI of Virtual Interactive Board

CLEAR – clears the canvas window

BLUE – lets the user select the blue color

GREEN – lets the user select the green color

RED – lets the user select the red color

YELLOW – lets the user select the yellow color

7.2 Libraries Used

```python
# All the imports go here
import cv2
import numpy as np
import mediapipe as mp
from collections import deque
```

Figure 7.3 Python libraries of Virtual Interactive Board

• MediaPipe (MP)

Using MediaPipe here to generate a hand landmark model

• OpenCV (CV2)

OpenCV allows creation of canvas and webcam capture window

• NumPY (NP)

NumPy is used to manage data generated in form of matrices during hand capture

CHAPTER 8
CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

The technology has the ability to transform standard writing methods in ways that go well beyond convenience; it can potentially help with particular communication issues that people with impairments or those who have trouble utilizing traditional input methods confront. The technology gives users an additional method of communication—like hand gesture recognition—that makes interacting with digital surroundings more inclusive and easily accessible.

Furthermore, people who may have trouble using a traditional keyboard, such as the elderly or those with limited dexterity, will find the device especially helpful due to its user-friendly interface. The method makes it easier for users to interact with digital interfaces by utilizing common gestures and movements like finger tracing and hand waving.

Moreover, new opportunities for smooth integration and automation arise from the system's ability to extend control to Internet of Things (IoT) devices. With simple hand gestures, users may easily operate appliances, smart home technology, and other IoT-enabled equipment, simplifying everyday chores and improving convenience overall.

In the future, a number of improvements could improve the functionality and strength of the system even further. First of all, handwriting recognition technology would greatly expedite word-for-word writing and enable users to produce text more quickly. This enhancement would be dependent on sophisticated algorithms that can recognize handwritten input with accuracy.

Second, improving the system's capacity to identify hand gestures that include pauses will improve its usability and responsiveness in practical situations. Temporal signals and context-aware algorithms could help the system distinguish intentional gestures from accidental movements more accurately, lowering the possibility of false positives or unwanted activities.

Thirdly, improving the security features of the system would lessen the possibility of unwanted access or manipulation. Examples of these features include changing the initial state of fingertip recognition and adding user-specific control mechanisms. This would protect against any misuse or exploitation by guaranteeing that the system only reacts to the gestures of the intended user.

Furthermore, YOLOv3 (You Only Look Once version 3) and other cutting-edge object detection algorithms could be used to increase the accuracy and speed of fingertip recognition. The system might adapt and change over time, becoming more dependable and efficient in a variety of settings and usage scenarios by utilizing artificial intelligence and machine learning.

In conclusion, the system has a great deal of promise for the future due to its ability to improve accessibility, facilitate communication, and enable smooth interaction with IoT devices. Aerial writing systems have the potential to become increasingly useful tools for digital expression and interaction as AI and technology continue to improve.

8.2 Future Scope

8.2.1. Improved User Interface: To further simplify communication, the system may eventually include sophisticated user interface (UI) components including voice commands and gesture customization. Users could benefit from a more intuitive and personalized experience if the interface was customized to suit their unique needs and preferences.

8.2.2. Customization of Gesture Recognition: The system may allow users to alter gesture recognition parameters to meet their own needs and preferences. Users can customize the system to fit their own workflow by defining custom gestures, altering sensitivity levels, and generating individualized shortcuts for frequently used tasks.

8.2.3. Multi-Modal Interaction: In the future, the system may experiment with other ways to combine hand gestures with other input modalities as voice instructions, eye tracking, or facial expressions. The system might provide a more flexible and adaptive interface by utilizing numerous input channels, allowing users with a range of demands and preferences.

8.2.4. Integrating Augmented Reality (AR) with Virtual Reality (VR): By expanding its capabilities to accommodate AR and VR environments, the project may let users produce and engage with digital content in immersive virtual environments. The system may open up new avenues for interactive storytelling, collaborative workspaces, and artistic expression by utilizing AR and VR technology.

8.2.5. Collaborative Drawing: Upcoming iterations of the system may include features that enable several users to work together on the same digital canvas at once. This could make it easier for creative teams and artists to collaborate remotely, allowing them to work together in real time regardless of where they are physically located.

8.2.6. Accessibility Features: The system could include voice prompts, screen readers, and haptic feedback for users with visual or aural disabilities in order to guarantee inclusion. By giving accessibility top priority during the design and development phase, the system could guarantee that every user may take part completely in digital drawing activities.

8.2.7. Cloud-based Collaboration: To facilitate the easy sharing and synchronization of drawings amongst devices and users, the system may make use of cloud computing and collaborative platforms. This would enable users to work together in real-time, access their products from any location, and switch between devices without losing their work.

REFERENCES

[1] Huang, Yichao, Xiaorui Liu, Xin Zhang, and Lianwen Jin. "A pointing gesture based egocentric interaction system: Dataset, approach and application." In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 16-23. 2016.

[2] Mehtab, Sidra. "Object Detection and Tracking Using OpenCV in Python."

[3] Saoji, S., Nishtha Dua, Akash Kumar Choudhary, and Bharat Phogat. "Air canvas application using Opencv and numpy in python." IRJET 8, no. 08 (2021).

[4] Satpute, Suraj, Harshad Shende, Vikas Shukla, and Bharti Patil. "Real time object detection using deep-learning and OpenCV." International Research Journal of Engineering and Technology (IRJET) 7, no. 4 (2020): 3243-3246.

[5] Chang, Yuan-Hsiang, and Chen-Ming Chang. Automatic hand-pose trajectory tracking system using video sequences. IntechOpen, 2010.

[6] Gurav, Ruchi Manish, and Premanand K. Kadbe. "Real time finger tracking and contour detection for gesture recognition using OpenCV." In 2015 International Conference on Industrial Instrumentation and Control (ICIC), pp. 974-977. IEEE, 2015.

[7] Chandhan, Tamalampudi Hemaa, Nalin Raj, Neelam Nanda Kishore Reddy, and Mohammed Zabeeulla AN. "Air Canvas: Hand Tracking Using OpenCV and MediaPipe." In 1st-International Conference on Recent Innovations in Computing, Science & Technology. 2023.

[8] Tiwari, Satyam V., Deep A. Vartak, and Ms Soniya Khatu. "Virtual Hand Gesture Painting."

[9] Vasavi, R., Nenavath Rahul, A. Snigdha, K. Jeffery Moses, and S. Vishal Simha. "Painting with Hand Gestures using MediaPipe." Int. J. Innov. Sci. Res. Technol. 7, no. 12 (2022): 1285-1291.

[10] Khanum, Hajeera, and H. B. Pramod. "Smart Presentation Control by Hand Gestures Using computer vision and Google's Mediapipe." International Research Journal of Engineering and Technology (IRJET) (2022): 2657.

[11] Agrawal, Subhash Chand, Rajesh Kumar Tripathi, Neeraj Bhardwaj, and Prashun Parashar. "Virtual Drawing: An Air Paint Application." In 2023 2nd International Conference on Edge Computing and Applications (ICECAA), pp. 971-975. IEEE, 2023.

[12] KANTER, JORDAN A., and KAMIL QUINTEROS. "GESTURAL DESIGN-HAND TRACKING FOR DIGITAL DRAWING." Architecture and Planning Journal (APJ) 28, no. 3 (2023): 3.

[13] DAHANAYAKA, DTDM, AR LOKUGE, JADE JAYAKODY, and IU ATTHANAYAKE. "Smart Drawing for Online Teaching." Instrumentation 8, no. 2 (2021): 56-66.

[14] Gunda, Chidvika, Manohar Maddelabanda, and Hariharan Shanmugasundaram. "Free hand text displaying through hand gestures using mediapipe." In 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), pp. 996-1000. IEEE, 2022.

[15] Osama, NourEldin, Yousr Ahmed, Hussein Mohamed, Seif Eldin Hesham, Youssef Ahmed, Eman K. Elsayed, and Dalia Ezzat. "Virtual Control System for Presentations by Real-Time Hand Gesture Recognition Based on Machine Learning." In International Conference on Advanced Intelligent Systems and Informatics, pp. 327-335. Cham: Springer Nature Switzerland, 2023.

PROOF OF PATENT PUBLICATION

(12) PATENT APPLICATION PUBLICATION
(21) Application No.202411030532 A

(19) INDIA

(22) Date of filing of Application :16/04/2024
(43) Publication Date : 10/05/2024

(54) Title of the invention : INNOVATIVE AND SMART VIRTUAL INTERACTIVE BOARD

(51) International classification	:G06F0003010000, G06Q0010100000, G06N0020000000, G06F0003048150, G06N0003080000
(86) International Application No Filing Date	:NA :NA
(87) International Publication No	: NA
(61) Patent of Addition to Application Number Filing Date	:NA :NA
(62) Divisional to Application Number Filing Date	:NA :NA

(71)**Name of Applicant :**
1)KIET Group of Institutions
Address of Applicant :KIET Group of Institutions, Delhi-NCR, Ghaziabad-Meerut Road, Ghaziabad, Uttar Pradesh, INDIA, 201206 ----------- -----------
2)Dr. Ruchita Gautam
Name of Applicant : NA
Address of Applicant : NA
(72)**Name of Inventor :**
1)Pardeep Tyagi
Address of Applicant :KIET Group of Institutions, Delhi-NCR, 13 KM Stone, NH 58, Ghaziabad, Uttar Pradesh, India-201206 Ghaziabad ----------- -----------
2)Vaibhav Mittal
Address of Applicant :KIET Group of Institutions, Delhi-NCR, 13 KM Stone, NH 58, Ghaziabad, Uttar Pradesh, India-201206 Ghaziabad ----------- -----------
3)Vaibhav Singh
Address of Applicant :KIET Group of Institutions, Delhi-NCR, 13 KM Stone, NH 58, Ghaziabad, Uttar Pradesh, India-201206 Ghaziabad ----------- -----------
4)Sarthak Srivastava
Address of Applicant :KIET Group of Institutions, Delhi-NCR, 13 KM Stone, NH 58, Ghaziabad, Uttar Pradesh, India-201206 Ghaziabad ----------- -----------

(57) Abstract :
This invention represents a groundbreaking advancement in human-computer interaction, offering an immersive and intuitive platform for digital engagement. This innovation combines the power of Python, OpenCV, Mediapipe, and Numpy to create a dynamic virtual board that responds to natural gestures and movements. The system's primary objectives encompass enhancing remote education by fostering engagement and knowledge retention, facilitating seamless collaboration in remote work environments, and leveraging advanced technologies in computer vision and machine learning to create a realistic and versatile interactive experience. With applications spanning education, teamwork, creative endeavors, and beyond, the Virtual Interactive Board redefines the boundaries of digital interaction, promising a future where users can effortlessly manipulate digital content in ways that mirror their physical world interactions.

No. of Pages : 12 No. of Claims : 4