A Project Report on

Whiteboard application using machine learning model

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science and Engineering

Submitted by

VENNAM ESHWAR
(20H51A05G0)
KOMMALAPATI SANDEEP
(20H51A0514)
SABA ZAREEN
(20H51A0574)

Under the esteemed guidance of
Mr. V. NARASIMHA
(Assistant Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC Autonomous)
*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A+ Grade
KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020 - 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project report entitled report entitled **"Whiteboard application using machine learning model"** being submitted by Vennam Eshwar (20H51A05G0), Kommalapati Sandeep (20H51A0514), Saba Zareen (20H51A0574) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr. V. Narasimha
Assistant Professor
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Mr. V. Narasimha, Assistant Professor**, Department of Branch Name for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of Dept Name, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Dept Name for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and Shri Ch Abhinav Reddy, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

Vennam Eshwar 20H51A05G0
Kommalapati Sandeep 20H51A0514
Saba Zareen 20H51A0574

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

The Whiteboard Application Using OpenCV is a digital platform designed to replicate the functionality of traditional physical whiteboards while harnessing the capabilities of computer vision technology. This application allows users to draw, write, and collaborate in a digital space, enhancing remote communication, creative expression, and interactive learning experiences. The application supports real-time collaboration, allowing multiple users to engage in synchronous drawing and annotation on a shared canvas. The integration of collaborative features promotes remote teamwork, virtual brainstorming, and interactive educational sessions. The application includes an undo/redo functionality that maintains a history of actions, enabling users to correct mistakes or explore different design iterations without limitations. The user interface (UI)is designed for ease of use and efficient navigation. The intuitive interface ensures that users, regardless of their technical proficiency, can seamlessly access and employ the application’s features.

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 Problem Statement

The project aims to develop a real-time hand tracking and drawing system utilizing OpenCV, offering an open-source, customizable, and accessible solution for interactive digital art creation. The primary objective is to design a system that enables users to draw and paint in the air, using their hand movements as input, in a natural and intuitive manner. The system should emphasize affordability, ease of use, and flexibility, catering to a wide range of creative applications and user preferences. It addresses the need for affordable and accessible hand tracking and drawing technology, while also offering a platform for innovation, experimentation, and user-driven improvements.

The whiteboard application is used to train our laptop or our screen to read whatever the user will be writing Infront of the screen in the air and that will be displayed on it. We will use some of the machine and deep learning libraries which includes open cv algorithms and functions are used for various tasks related to image processing, computer vision, and user interface development.

1.2 Research Objective

We will delve into the development of interactive features that enhance the creative process. Our research will explore mechanisms for color selection, including gesture-based color picking and interactive tools. We will investigate techniques for users to adjust the thickness of drawn lines, providing them with creative flexibility while maintaining an intuitive user experience. Eraser mode will be implemented through various interaction mechanisms, allowing users to correct or remove parts of their drawings effortlessly.

Additionally, we will explore multiplatform compatibility, adapting the project to different devices and platforms, ensuring performance and user experience optimization. Through these research objectives, we intend to deliver an engaging and cutting-edge interactive hand-tracking and drawing solution that pushes the boundaries of human-computer interaction and digital creativity.

1.3 Project Scope and Limitations

Implementing more advanced hand gesture recognition algorithms to improve accuracy and recognition speed. Integrating the system with other technologies, such as voice recognition or eye tracking, to create a more comprehensive human-computer interface. Collaborating with experts in fields such as robotics, computer vision, or human-computer interaction to advance the technology and develop new applications.

The project's performance and functionality may be limited by the capabilities of the hardware used, such as webcams or sensors. Higher-quality hardware may be required for optimal results. Hand tracking and gesture recognition accuracy may vary based on lighting conditions, background complexity, and the user's hand size and shape. Implementing real- time collaborative drawing with multiple users may introduce synchronization and communication challenges.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2
BACKGROUND WORK

2.1 Virtual Teaching Board using deep learning

2.1.1 Introduction:

The integration of deep learning technologies into the field of education has given rise to an extraordinary innovation known as the Virtual Teaching Board. This remarkable development harnesses the power of deep learning algorithms to revolutionize the way educators impart knowledge and engage with students. It represents a pivotal moment in the evolution of digital education, creating a dynamic and immersive learning environment.

At its heart, the Virtual Teaching Board is a digital whiteboard equipped with cutting-edge deep learning capabilities. These advanced algorithms enable the platform to understand, analyze, and adapt to the unique needs of each student. In real time, it can provide personalized content, recommendations, and feedback, transforming the educational experience into a highly individualized journey.

These platforms have the capacity to recognize and interpret students' learning patterns and behaviors. By leveraging this insight, educators can tailor their lessons to match each student's pace and preferred learning style, effectively bridging the gap between traditional classroom instruction and online education. Moreover, Virtual Teaching Boards equipped with deep learning can adapt to the evolving educational landscape. As new pedagogical techniques and insights emerge, the platform can integrate them seamlessly, ensuring that educators have access to the latest educational methodologies and technologies.

Till now we have different types of White board software for online teaching. Each software has its own pros and cons. One of the main disadvantages of these traditional whiteboard software is, they require hardware pointing devices or a touch sensitive screen for interaction like stylus, digital pens, mouse etc. So to avoid usage of such pointing devices and to make online learning more interesting and effective, we are introducing Virtual Teaching Board, which is a hand movement based writing in air software. It uses different hand gestures of the fingertips in air to write on the whiteboard instead of using external hardware devices like mouse, stylus, digital pens etc.

It provides users with an amusing environment. This project is divided into different modules namely Virtual board Module, Presentation Module and Calculation Module. In the Virtual board module, we are providing the user interface with functionalities selecting types of board (Normal and Blackboard) , Selecting colors for writing, Eraser mode, Presentation Mode and Calculator Mode. In the Virtual Board Module, By selecting writing mode and board type, users can write on the screen using different colors by just fingertip movements in air, and user can erase the board partially using eraser option or completely by waving five fingers on the screen And by selecting Presentation mode the user enters into Presentation module and there user can give presentation by moving slides forth and back just by hand gestures in air and user can also write or markup on the slides.

Sometimes a user may need to perform some mathematical operations, so to make his job done easily, we are providing a calculator module. User can simply select calculator mode and can perform mathematical operations. User can also have an option of changing fingertip pointer dimensions, means user can increase or decrease the size of the fingertip pointer for writing and also for erasing by using hand gestures. We used Open-cv, Mediapipe, NumPy libraries and Python Programming language in this Project.

Motivation

In the age of this digital world, the need for user-friendly whiteboard software is essential. The traditional whiteboard software requires a hardware pointing device or a touch screen for interaction.

These traditional whiteboard software requires high cost devices, Needs the latest device equipment which are not affordable to every user. So the main motive behind this project is to build a software that is user friendly and cost affordable.

Problem Definition

To create a virtual teaching software tool that may be used as an alternative to traditional ways of teaching. This is cost effective, it can be used by any teacher to make their teachings more informative and effective. We also sought to make it as simple and user-friendly as possible, with very minimum hardware requirements, so that even someone with no prior computer experience could use it.

Objective of the Project

To provide a virtual teaching board platform that uses hand gesture recognition in air to write on screen. To make the teaching process more effective we developed presentation mode, Calculation mode, different user interfaces with functionalities selecting types of board (Normal and Blackboard), Selecting colors for writing and Eraser for erasing with the help of Open CV, Media Pipe, NumPy libraries and python programming language in this Project.

2.1.2 Merits and Demerits:

Merits:

1. Personalization: Deep learning algorithms enable these boards to personalize education foreach student, adapting to their pace, style, and abilities. This tailored approach can enhance learning outcomes.
2. Accessibility: Virtual Teaching Boards using deep learning can be accessible from various devices and locations, making education more inclusive and flexible for students with diverse needs and circumstances.
3. Scalability: They can cater to a wide range of subjects and levels, from primary education to higher education and professional training, providing a versatile solution for various learning environments.
4. One of the most important merits of using deep learning for a virtual teaching board is its ability to provide personalized learning experiences tailored to individual students' needs and preferences.

Demerits:

1. Resource Intensive: Implementing deep learning technology can be resource- intensive in terms of computational power and infrastructure, which might pose challenges for schools with limited budgets.

2. Data Privacy: Handling student data and using deep learning algorithms for personalization can raise concerns about data privacy and security, especially in cases where sensitive information is involved.

3. Teacher Training: Educators need training to effectively use these platforms, understand the algorithms, and interpret data insights. The learning curve can be steep.

4. Technical Issues: Technical glitches or connectivity problems can disrupt lessons, causing frustration for both educators and students.

Challenges:

1. Integration and Compatibility: Incorporating Virtual Teaching Boards with deep learning into existing educational systems can be challenging, requiring time and resources for seamless integration.

2. Evolving Technology: Keeping up with the rapidly evolving field of deep learning and educational technology requires constant adaptation and investment.

3. Equity: Ensuring that all students, regardless of their socioeconomic status or access to technology, can benefit from these platforms remains a significant challenge.

2.1.3 Implementation:

To provide a virtual teaching board platform that uses hand gesture recognition in air to write on screen. To make the teaching process more effective we developed presentation mode, Calculation mode, different user interfaces with functionalities selecting types of board (Normal and Blackboard), Selecting colors for writing and Eraser for erasing with the help of OpenCV, Media Pipe, NumPy libraries and python programming language in this Project.

Cam Setup: The runtime operations are managed by the webcam of the connected laptop or desktop. To capture a video, we need to create a Video Capture object. Its argument can be either the device index or the name of a video file. Device index is just the number to specify which camera. Since, we only use a single camera we pass it as ‘0’. We can add additional camera to the system and pass it as 1,2 and so on. After that, you can capture frame-by-frame. But at the end, don’t forget to release the capture. We could also apply color detection techniques to any image by doing simple modifications in the code.

Capturing frames: The infinite loop is used so that the web camera captures the frames in every instance and is open during the entire course of the program. We capture the live feed stream, frame by frame. Then we process each captured frame which is in RGB (default) color space to HSV color space. There are more than 150 color-space conversion methods available in OpenCV. But we will look into only two which are most widely used ones, BGR to Gray and BGR to HSV.

Masking technique: The mask is basically creating some specific region of the image following certain rules. Here we are creating a mask that comprises of an object in red color. After that we perform a bitwise AND operation on the Input image and the Threshold image, which result in only the red colored objects are highlighted. This result of the AND operation is stored in res. We then display the frame, res and mask on 3 separate windows using imshow() function.

Hand detection is related to the location of the presence of a hand in a still image or sequence of images i.e. moving images. In case of moving sequence's it can be followed by tracking of the hand in the scene but this is more relevant to the applications such as sign language. The underlying concept of hand detection is that human eyes can detect objects which machines cannot with that much accuracy as that of a human. From a machine point of view it is just like a man fumble around with his senses to find an object. The factors, which make the hand detection task difficult to solve, are: Variations in image plane and pose so the hands in the image vary due to rotation, translation and scaling of the camera pose or the hand itself. The rotation can be both in and out of the plane. Skin Color and Other Structure Components for the appearance of a hand is largely affected by skin color, size and also the presence or absence of additional features like hairs on the hand further adds to this variability.

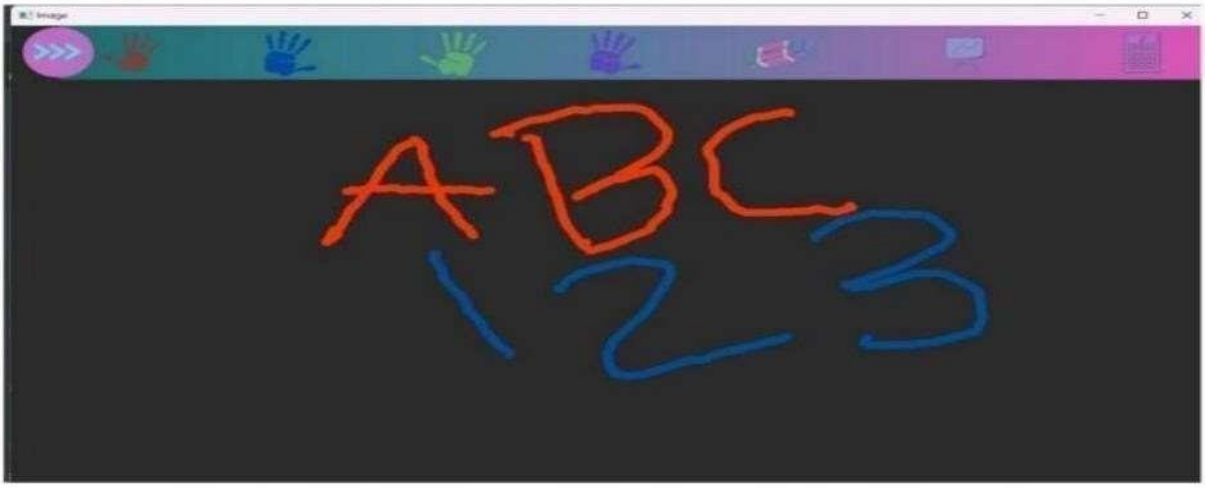


Fig 2.1.3.1: Writing in virtual black board

2.2 Virtual Whiteboard-A Gesture Controlled Pen-free Tool

2.2.1 Introduction:

The Virtual Whiteboard a Gesture-Controlled, Pen-Free Tool is a remarkable innovation that redefines the way we interact with digital information and collaborate in various professional and educational settings. This cutting-edge technology provides users with an interactive and versatile platform that simulates a traditional whiteboard while eliminating the need for physical markers, pens, or touch-sensitive screens.

At its core, the Virtual Whiteboard leverages gesture control to enable users to write, draw, and manipulate digital content using hand movements, finger gestures, or specialized peripherals like styluses. This approach offers a tactile and immersive experience, reminiscent of traditional whiteboards, while providing the advantages of digital technology.

One of the standout features of this tool is its pen-free nature, which eliminates the need for consumable writing tools and minimizes maintenance costs. This aspect not only reduces environmental impact but also enhances convenience, as users can write, erase, and manipulate content without concerns about ink, marker stains, or damaged writing surfaces.

The Virtual Whiteboard's gesture-controlled interface makes it a versatile and inclusive tool for a range of applications. It is ideal for collaborative work, brainstorming, teaching, and presentations, whether in physical spaces or remote settings. Users can simply use their hand movements to write or draw, making it accessible to individuals of all ages and abilities. This introduction sets the stage for exploring the multifaceted benefits and applications of the Virtual Whiteboard, offering a glimpse into how it has revolutionized the way we interact with digital content and collaborate in today's fast-paced and technology-driven world.

Virtual Whiteboard is a cutting-edge application that enables users to create digital drawings and presentations using hand gestures. This innovative technology provides a new way for users to interact with digital content, offering a more natural and intuitive user experience. The application uses advanced computer vision and machine learning algorithms to detect and interpret hand gestures, allowing users to create and manipulate content with ease. Virtual Whiteboard is designed to be a versatile and powerful tool for communication and collaboration. With a variety of tools and features, users can create and manipulate digital content to their liking. Drawing and writing tools, and PowerPoint controlling capabilities are all available in the application. The application's easy-to-use interface makes it accessible to users of all skill levels, allowing them to create and manipulate digital content with hand gestures quickly.

One of Virtual Whiteboard standout features is its PowerPoint controlling capabilities. Users can use hand gestures to control their PowerPoint presentations, eliminating the need for a clicker or keyboard. This feature makes it an ideal tool for public speakers, educators, and business professionals who want to engage their audiences in a more interactive and dynamic way. Users can advance or go back to slides, and even draw or annotate over the slides in real-time. A virtual whiteboard is a digital platform that mimics the features of a traditional whiteboard. It allows users to draw and write. A gesture-controlled virtual whiteboard eliminates the need for a stylus or a mouse and instead allows users to interact with the board using hand gestures.

2.2.2 Merits and Demerits:

Merits:

1. Interactivity: The tool enables natural and interactive interactions with digital content, making it engaging for both teaching and collaborative work. Users can draw, write, and manipulate content using gestures, which can enhance the learning and presentation experience.
2. Accessibility: Gesture control technology makes the Virtual Whiteboard accessible to a wide range of users, including those with physical disabilities.

3. Collaboration: The virtual whiteboard facilitates collaboration by allowing multiple users to interact with the board simultaneously from different locations. This feature is particularly useful for remote teams or classrooms where participants need to work together in real-time.
4. Cost-Effectiveness: Compared to traditional physical whiteboards, which require maintenance and replacement of markers and erasers, Merits Virtual Whiteboard offers a cost-effective solution. Once set up, users can access and utilize the virtual whiteboard without ongoing expenses for consumables.
5. Portability: Being a virtual tool, Merits Virtual Whiteboard is likely accessible across different devices such as computers, tablets, and interactive displays. This portability allows users to access and collaborate on the whiteboard from anywhere with an internet connection, fostering productivity and flexibility.

Demerits:

1. Learning Curve: Using gesture controls may have a learning curve, especially for individuals who are not familiar with the technology. Some users may find it challenging to adapt to this new way of interacting with digital content.
2. Limited Precision: Gesture control might not offer the same level of precision as using a physical writing tool. This can be a limitation for tasks that require intricate details or fine motor control.
3. Technical Limitations: Gesture control technology may not be flawless, and users may encounter technical issues such as inaccuracies in gesture recognition or lag in response time. These limitations could disrupt the user experience and impact productivity, especially in fast-paced collaborative environments.
4. Cost of Implementation and Maintenance: While virtual whiteboards can be cost-effective compared to traditional physical whiteboards in the long run, there may be initialcosts associated with setting up compatible hardware and software infrastructure.

5. Compatibility Issues: Compatibility with different devices, operating systems, or software platforms could pose challenges, particularly in heterogeneous environments where users utilize a variety of tools and technologies. Lack of seamless integration with existing workflows may hinder adoption and usage of the virtual whiteboard.

Challenges:

1. Standardization: The field of gesture control is still evolving, and standardization across different systems and platforms can be a challenge. Ensuring compatibility and a consistent user experience can be difficult.

2. Reliability: The reliability of gesture recognition systems is critical, especially in professional and educational settings where downtime can disrupt meetings, presentations, or lessons.

2.2.3 Implementation:

The objective is to create a free space where one can draw in air freely. The RGB camera detects the fingertip and tracks its motion throughout the screen. Whenever the hand comes in front of the camera, the initial thing to do is detect the fingertip.

Once the hand region and center of gravity have been successfully detected, the subsequent step is to track the movement of the fingertips on the screen. Previous research has shown that using the faster R-CNN handheld detector is computationally intensive, resulting in a lower frame rate that is below real-time performance. This algorithm first converts the detected fingertip into the HSV color space. Once the fingertip mask is identified in the air, the system performs various morphological operations to remove impurities from the masked image. The next critical step involves detecting the contours and drawing the line. To accomplish this, a python deque is created that memorizes the outline's position in each subsequent frame. The deque accumulates these points and uses them to create a line using OpenCV's drawing capabilities. This approach allows for precise and efficient fingertip tracking, even in real-time scenarios.

The first and second phases of the system involve selecting and opening a PowerPoint file, which will be presented on the PowerPoint window. The third phase involves transforming this array, while in the fourth phase, the system classifies the array of images using pre-trained weights that were previously loaded. This approach enables accurate recognition and classification of live gestures, which can be used to control various features in the PowerPoint presentation.

Fig 2.2.3.1: Performing Paint

2.3 Virtual Teaching Board using computer vision

2.3.1 Introduction:

Most of the people are familiar with various whiteboard software. In the digital social media where photos or images got much importance, the need of a user friendly whiteboard software is essential. The traditional whiteboard software requires a hardware pointing devices or a touch sensitive screen for interaction.

In most cases we need a hardware medium for interacting with the software system. Direct use of hands as an input device is an attractive method for providing natural human-computer interaction which has evolved from text-based interfaces through graphical based interfaces. hand movement recognition can be seen as a way for computers to begin understanding human body language, thus building a richer bridge between machines and humans.

It will be more user friendly if the computer system can be controlled using hand movements. This project propose a camera based Human Computer Interaction (HCI) system in which movements of user's hand is directly involved in creating and manipulation of art (Virtual board). Here we prefer the standard image processing library Open-CV.

The reason is by comparing various aspects of our proposed system, Open CV is more flexible than that of Matlab. In our system we have included interactive art for improving the educational experience. The related hand movement interactive mechanism include Interactive art system for multiple users based on tracking hand The proposed HCI system captures video of people's hand movements using a high resolution camera and then converted into necessary Diagrams.

One of the main disadvantages of these whiteboard software is the need of accurate pointing devices. Here we introduce the Smart virtual Board, which is a hand movements based writing software. It uses the hand movements for writing. User doesn’t need an external hardware pointing device to draw. Instead the different hand movements of the fingertips are used for writing purposes.

We designed this system as a primary level software product. The software can be further upgraded by improving the current features. The disadvantage of the traditional writing software is that they use traditional writing devices like mouse light, pens etc. Now it overcomes by Computer vision technology. The next evolutionary technology that will take over the world will be the hand movement technology. By using hand movements for communicating with system.

2.3.2 Merits and Demerits:

Merits:

1. The desktop computing paradigm limits the users' flexibility by forcing them to interact using a 2-Degree-Of Freedom device (the mouse), while they are used to interacting with the physical world in much more differentiated ways.

2. Hand movements allow the user to handle multiple points of input and even define several parameters at once.

3. Engagement and Motivation: The interactive nature of virtual teaching boards can increase student engagement and motivation, as it provides a more immersive and hands-on learning environment compared to traditional teaching methods.

Demerits:

1. As it is system based on the hand movements it is more important that hand movement should be proper so that it can properly understand by the camera.

2. Also it is important that for recognizing hands camera module requirement should be satisfied which in turns helps by eliminating the barrier between the camera and user with its meaningful actions.

3. Implementing computer vision technology for virtual teaching boards requires expertise in computer science and image processing, which may pose challenges for educators and institutions lacking technical resources.

4. Students and educators may require time to familiarize themselves with the virtual teaching board interface and its functionalities, potentially leading to a temporary decrease in productivity or engagement.

5. Not all students may have access to the necessary technology or internet connectivity required to participate effectively in virtual teaching board activities, exacerbating existing equity disparities in education.

Challenges:

1. Complexity of Implementation: Implementing Computer Vision in a Virtual Teaching Board involves intricate setup and calibration. Educators and IT staff need to be adequately trained to handle this complexity, which can lead to a steep learning curve.

2. Privacy Concerns: Camera-based Computer Vision raises privacy concerns. Ensuring that student data and identities are protected is crucial. Striking a balance between data collection for educational purposes and safeguarding privacy can be challenging.

2.3.3 Implementation:

The implementation of a Virtual Teaching Board using Computer Vision technology is a transformative process that promises to enhance the educational experience by leveraging advanced visual recognition capabilities. This endeavor begins with a clear understanding of the educational objectives and goals for which the Virtual Teaching Board will be employed. Defining these goals is paramount, as they serve as the guiding principles for subsequent steps in the implementation process.

Find Out Hand Landmark Position: There are 21 different Hand landmarks. Each ID have corresponding landmark, And landmark have x, y & z. So we are going to use x & y co-ordinate to find information or to find location for the landmark on hand but the values of Landmarks are in decimal point so location it should be in pixels. So it providing ratio of Images. So here we multiply it with the width and height and then we will get pixel value.

Fig 2.3.3.1: Virtual Writing Board

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1 Objective of Proposed Model:

The primary objective of this project is to develop a sophisticated computer application leveraging OpenCV and Python to transform a conventional screen into a dynamic notice board capable of drawing, writing, and erasing content using alternative methods. This innovative application aims to harness the power of computer vision and hand gesture recognition through live webcam input to provide users with an intuitive interface for interacting with the screen. With this program, you'll be able to move your hand in front of your webcam, and the program will follow your hand movements. This means you can draw lines, write words, and erase things directly on the screen just by moving your hand around.

By utilizing OpenCV's advanced color feature analysis, the application will accurately track the movement of the user's hand in real-time. This functionality enables users to seamlessly draw, write, and erase content directly onto the screen using intuitive hand gestures, thereby enhancing the overall user experience and eliminating the need for traditional input devices such as a mouse or keyboard. The project seeks to implement a range of interactive features, including drawing lines of varying thickness and color, writing text in different fonts and sizes, and erasing content with precision. Additionally, the application will support multi-touch gestures to facilitate collaborative editing and manipulation of content on the screen.

The project seeks to implement a range of interactive features, including drawing lines of varying thickness and color, writing text in different fonts and sizes, and erasing content with precision. Additionally, the application will support multi-touch gestures to facilitate collaborative editing and manipulation of content on the screen.

Through the integration of cutting-edge computer vision algorithms and Python scripting capabilities, the project aims to deliver a versatile and user-friendly solution for transforming any display into a dynamic notice board.

By empowering users to interact with digital content in a more natural and intuitive manner, the application has the potential to revolutionize the way individuals communicate, collaborate, and express ideas in both personal and professional settings. Application will support multi-touch gestures to facilitate collaborative editing and manipulation of content on the screen.

In summary, the overarching goal of this project is to leverage the capabilities of OpenCV and Python to create an innovative computer application that enables users to utilize hand gestures for drawing, writing, and erasing content on a screen, thereby providing a novel and engaging platform for digital interaction and communication.

3.2 Algorithms Used for Proposed Model:

3.2.1 Open CV:

OpenCV is an open-source computer vision library which is designed for computational efficiency and having a high focus on real-time image detection. OpenCV is coded with optimized C and can take work with multicore processors. If we desire more automatic optimization using Intel architectures.

One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure which helps people to build highly sophisticated vision applications fast. The OpenCV library, containing over 500 functions, spans many areas in vision. Because computer vision and machine learning often goes hand-in-hand. OpenCV also has a complete, general-purpose, Machine Learning Library (MLL).

This sub library is focused on statistical pattern recognition and clustering. The MLL is very useful for the vision functions that are the basis of OpenCV's usefulness, but is general enough to be used for any machine learning problem.

3.2.2 Media Pipe:

MediaPipe is a framework mainly used for building audio, video, or any time series data. With the help of the MediaPipe framework, we can build very impressive pipelines for different media processing functions.

Some ofthe major applications of MediaPipe.

- Multi-hand Tracking
- Face Detection
- Object Detection and Tracking
- 3D Object Detection and Tracking
- Automatic video cropping pipeline etc.

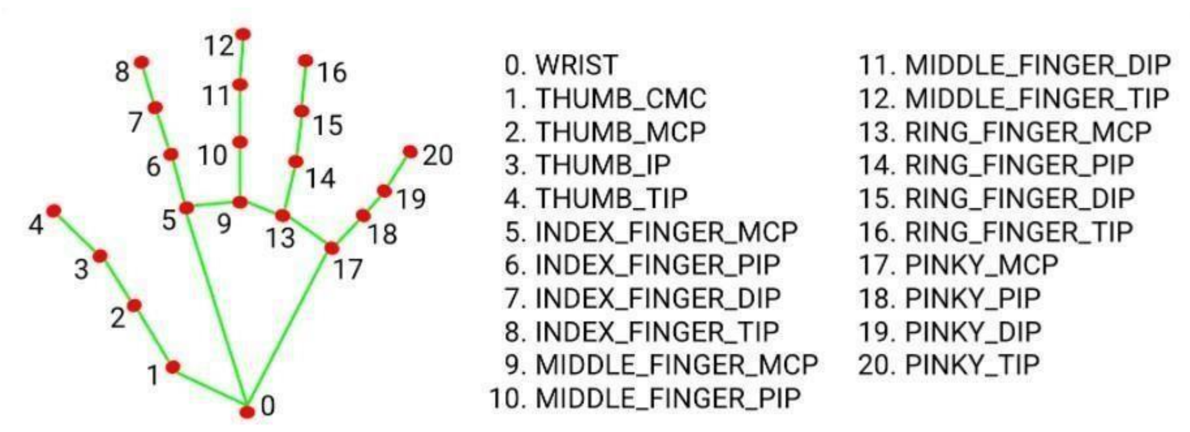


Fig 3.2.2.1: Hand Landmark Model

Basically, the MediaPipe uses a single-shot palm detection model and once that is done it performs precise key point localization of 21 3D palm coordinates in the detected hand region.

The MediaPipe pipeline utilizes multiple models like, a palm detection model that returns an oriented hand bounding box from the full image. The cropped image region is fed to a hand landmark model defined by the palm detector and returns high-fidelity 3D hand key points.

Tip of index and middle fingers:

we have a frame with skin colour regions only, but what we really want is to find the location of a fingertip. Using OpenCV you can find contours in a frame if you don't know what contour. Using contours you can find convexity defects, which will be potential fingertip location.

In my application, I needed to find the tip of a finger with which a user is aiming. To do this I determined the convexity defect, which is furthest from the centroid of the contour.

Finger up detection:

In this stage from all finger detection of index finger is calculated and if it is index finger then drawing option is activated.

Draw on screen:

Now that everything is all set and we are easily able to track our target object, it's time to use this object to draw virtually on the screen.

Now what we just have to do is use x,y location returned from cv2.boundingRect() function from a previous frame (F-1) and connect it with x,y coordinates of the object in the new frame (F). Byconnecting those two points we draw a line and we do that for each frame in the webcam feed, this way we'll see a real-time drawing with the pen.

3.2.3 Python:

Python is high level language and it is also integrated version of the program. Python is an object-oriented approach and its main aim to help programmers to write the code clearly, logical code for small and large scale of project.

The main goal of this programing language is simple, object-oriented programming language. The language and implementation should provide support for software engineering principles such as strong type library preset for different machine learning algorithm, and all other algorithm in simple manner. Coding will be smooth in python and the data analysis can be easily done in python.

This is so much so to the point where we now have modules and APIs at our disposal, and you can engage in machine learning very easily without almost anyknowledge at all of how it works. With the defaults from Scikit-learn, you can get 90-95% accuracyon many tasks right out of the gate.

3.2.4 Numpy as np:

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

At the core of the NumPypackage, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance.

Numpy arrays have a fixed size at creation, unlink Python lists (which can grow dynamically). Changing the size of an ndarray will create a new array and delete the original. Numpy arrays facilitates advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python’s built-in sequences.

A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays, though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays

3.3 Designing:

3.3.1 UML Diagrams:

A. Use Case Diagram: A use case diagram in the Unified Modeling Language (UML) is a type ofbehavioral diagram defined by and created from a Use-case analysis. Its purpose is to present agraphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose ofa use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

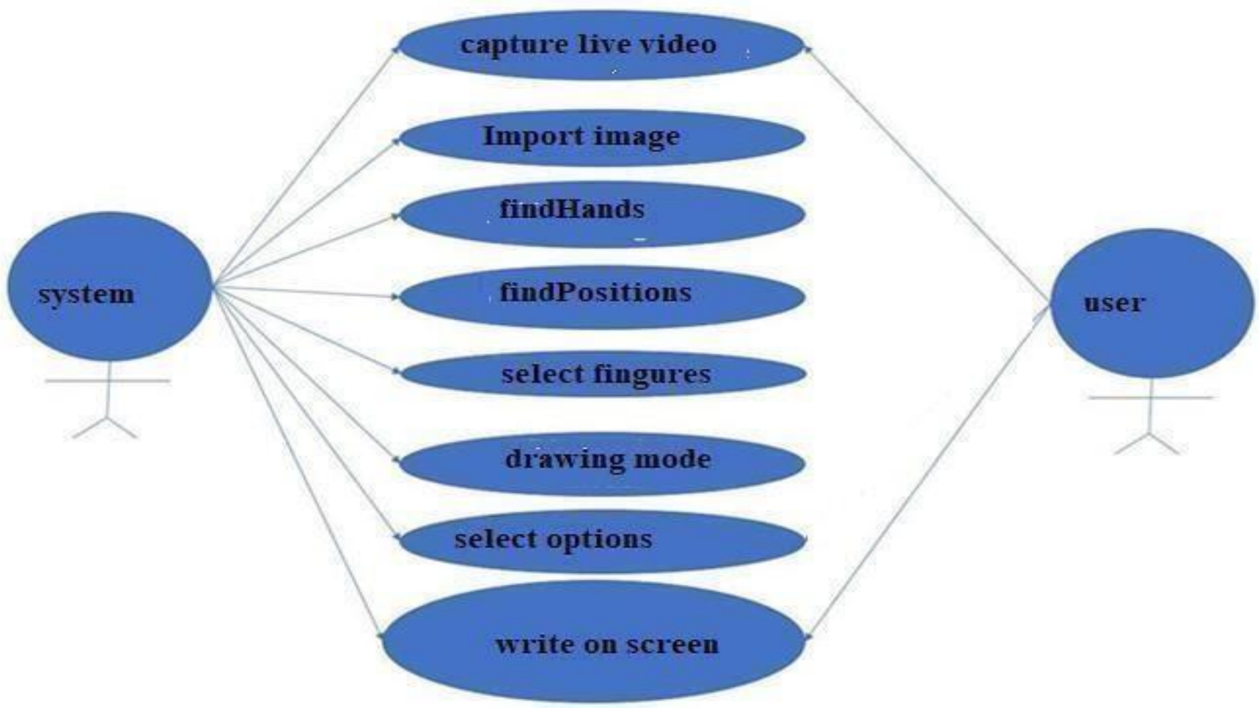


Fig 3.3.1.1: Use Case Diagram

B. Sequence Diagram: A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

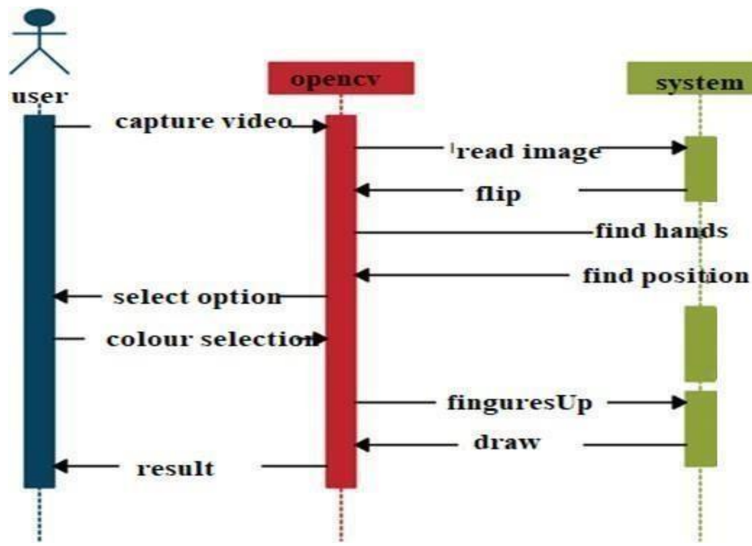


Fig 3.3.1.2: Sequence Diagram

C. Activity Diagram: This is graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

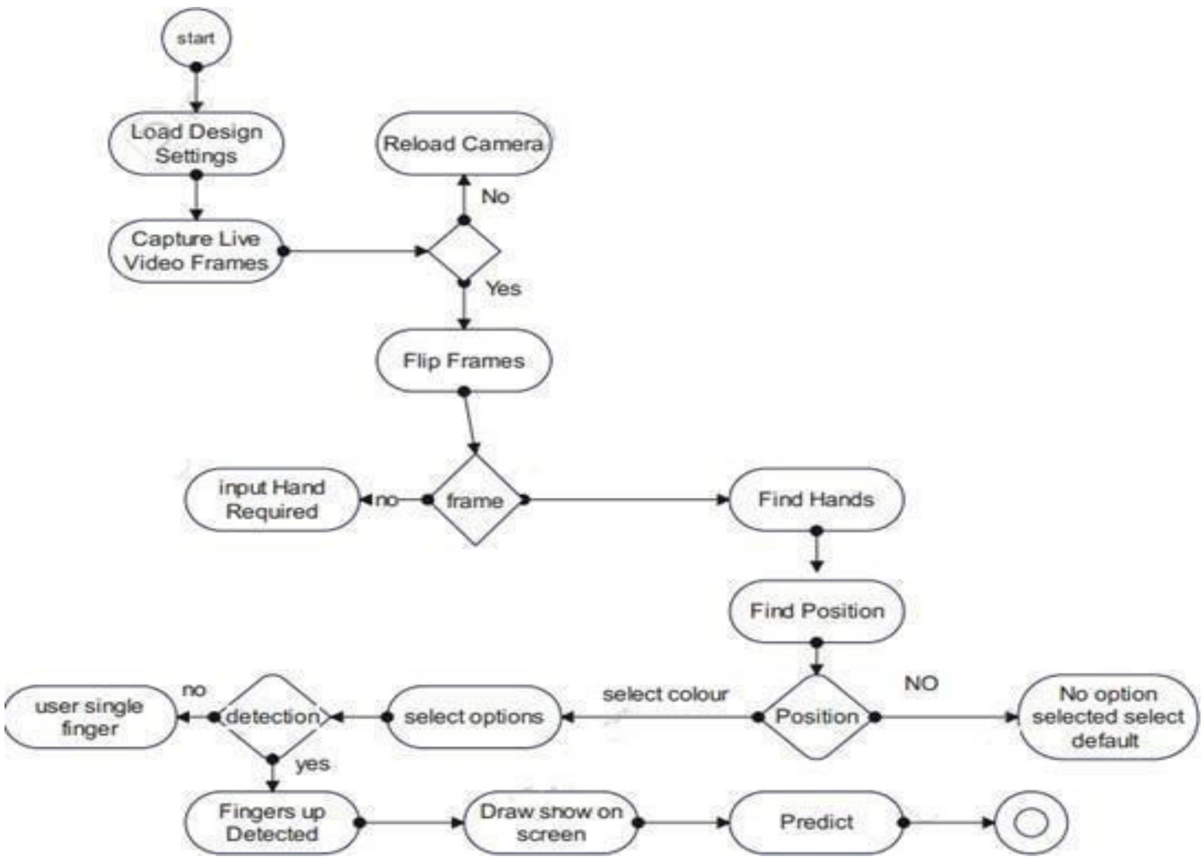


Fig 3.3.1.3: Activity Diagram

3.4 Stepwise Implementation and Code:

3.4.1 Implementation steps:

1. Create a header file for selection of pen and color and save in to list.
2. Declare variables for

   Header

   Color

   Bruch thickness

   Canvas

3. Capture live video using OPENCV library.
4. Detect hands (for this we use mediapipe library which has various functions for this project we are using hand detection features).
5. Initialize Media pipe variables
6. We declare an object called "hands" from *mp.solutions.hand* to detect the hands, in default, if you look inside the class "*Hands()*", the number of hands to detect is set to 2, minimum detection confidence is set to 0.5 and the minimum tracking confidence is set to 0.5. And we will use *mpDraw* to draw the key points.
7. We read the frames from the webcam and convert the image to RGB. Then we detect hands in the frame with the help of "*hands.process()"* function. Once the hands get detected we will locate the key points and then we highlight the dots in the keypoints using *cv2.circle,* and connect the key points using *mpDraw.draw_landmarks*.
8. Find Hands: The function *findHands* will accept an RGB image and detects the hand in the frame and locate the key points and draws the landmarks, the function *findPosition* will give the position of the hand along with the id.
9. Then the main function where we initialize our module and also we write a while loop to run the model. Here you can import this setup or the module to any other further related project works.

10. Detect Tip of index and middle fingers: In order to make this feature work initially, we have to take result from "hand recognition" modal so that we can import its features into tips detection. then we will define the parameters for the hand for which we can create a list. From created a list called TipID in which as an input which is feed to the landmark number of the tip of each finger of the hand.
11. Check which fingers are up: We read the webcam frame and initialize a variable *count*. This would be the final variable that will hold the fingers count.

The logic which we are using here is that if the y coordinate of the tip of any finger lies below the ycoordinate of the central landmark of that finger, that means the finger is closed.

For example: If the y coordinate of landmark 8 is less than the y coordinate of landmark 6 → This means the index finger is closed.

12. Selection mode: If 2 fingers are up, In this step finger-tip values are detected and if they are two fingers up then selection process will start. User can select color from the header and each selection will have different colors and eraser.
13. Drawing mode: Inthis step check index finger is up or not and find position of the finder and take input as color and if line- Index finger is up.

BITWISEAND:

cv2. bitwise_and() is a function that performs bitwise AND processing as the name suggests. The AND of the values for each pixel of the input images src1 and src2 is the pixel value of the output image.

BITWISEOR:

Bitwise operations are performed on an image when we need to extract only the required parts of the image. Consider a situation, in which we need to extract an irregular-shaped object from an image and paste it on another image. That's exactly when we use bitwise operations on the image to separate the foreground from the background.

3.4.2 CODE:

```
import cv2
import mediapipe as mp
import time

class handDetect():
  def __init__(self, mode=False, maxHands=2, detectionCon=0.5, trackCon=0.5):
    self.mode = mode
    self.maxHands = maxHands
    self.detectionCon = detectionCon
    self.trackCon = trackCon

    self.mpHands = mp.solutions.hands
    self.hands = self.mpHands.Hands(self.mode, self.maxHands, self.detectionCon, self.trackCon)
    self.mpDraw = mp.solutions.drawing_utils
    # self.hands = self.mpHands.Hands(self.mode, self.maxHands, self.detectionCon, self.trackCon)
    self.tipIds = [4, 8, 12, 16, 20]

  def findHands(self, img, draw=True):
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    self.results = self.hands.process(imgRGB)

    if self.results.multi_hand_landmarks:
      for handLms in self.results.multi_hand_landmarks:
        if draw:
          self.mpDraw.draw_landmarks(img,handLms, self.mpHands.HAND_CONNECTIONS)
    return img

  def findPositions(self, img, handNo=0, draw=True):
    self.lmList = []
    if self.results.multi_hand_landmarks:
      myHand = self.results.multi_hand_landmarks[handNo]
      for id,  lm in enumerate(myHand.landmark):
        h, w, c = img.shape
        cx, cy = int(lm.x * w), int(lm.y * h)
        self.lmList.append([id, cx, cy])
        if draw:
          cv2.circle(img, (cx, cy), 15, (255, 0, 255), cv2.FILLED)

    return self.lmList
```

```
def finguresUp(self):
      fingures = []
      #! Thumb
      if self.lmList[self.tipIds[0]][1] < self.lmList[self.tipIds[0] - 1][1]:
          fingures.append(1)
      else:
          fingures.append(0)

      #! Remaining 4 Fingures
      for id in range(1, 5):
          if self.lmList[self.tipIds[id]][2] < self.lmList[self.tipIds[id] - 2][2]:
              fingures.append(1)
          else:
              fingures.append(0)

      return fingures

  def main():
   pTime = 0
   cTime = 0
  cap = cv2.VideoCapture(0)
  detector = handDetect()
  while True:
      success, img = cap.read()
      img = detector.findHands(img)
      lmList = detector.findPositions(img)
      if len(lmList) != 0:
          print(lmList[4])

      cTime = time.time()
      fps = 1 / (cTime - pTime)
      pTime = cTime

      cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT_HERSHEY_PLAIN, 3, (255, 0, 255), 3)
      cv2.imshow("Image", img)

  if__name_== " main ":
  main()
```

```
import cv2
import numpy as np
import handtrackingmodule as htm
import time
import os

folderPath = "Header"
myList = os.listdir(folderPath)
# print(myList)
overlayList = []

for imgPath in myList:
  image = cv2.imread(f'{folderPath}/{imgPath}')
  overlayList.append(image)

# print(len(overlayList))

header = overlayList[0]
drawColor = (0, 0, 255)
brushThickness = 15
eraserThickness = 100
xp, yp = 0, 0
imgCanvas = np.zeros((720, 1280, 3), np.uint8)

cap = cv2.VideoCapture(0)

cap.set(3, 1280)
cap.set(4, 720)

detector = htm.handDetect(detectionCon=0.7, maxHands=1)

while True:
    #! 1. Import image
    success, img = cap.read()
    img = cv2.flip(img, 1)

    #! 2. Find Hand Landmarks
    img = detector.findHands(img)
    lmList = detector.findPositions(img, draw=False)
```

```
if len(lmList) != 0:

    #! Tip of index and middle fingers
    x1, y1 = lmList[8][1:]
    x2, y2 = lmList[12][1:]

    #! 3. Check which fingures are up
    fingures = detector.fingUresUp()

    #! 4. If selection mode - 2 fingures are up
    if fingures[1] and fingures[2]:
        xp, yp = 0, 0
        # print('selection')
        #! Checking for click
        if y1 < 125:
            if 250 < x1 < 450:
                header = overlayList[0]
                drawColor = (0, 0, 255)
            elif 550 < x1 < 750:
                header = overlayList[1]
                drawColor = (255, 0, 0)
            elif 800 < x1 < 950:
                header = overlayList[2]
                drawColor = (0, 255, 0)
            elif 1050 < x1 < 1200:
                header = overlayList[3]
                drawColor = (0, 0, 0)
        cv2.rectangle(img, (x1, y1 - 25), (x2, y2 + 25), drawColor, cv2.FILLED)
        cv2.putText(img, "Selection", (150, 170), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0,
0),2)
    #! 5. If drawing mode - Index fingure is up
    if fingures[1] and fingures[2] == False:
        cv2.circle(img, (x1, y1), 15, drawColor, cv2.FILLED)
        cv2.putText(img, "Drawing", (150, 170), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0),
2)
        # print('Drawing')
        if xp == 0 and yp == 0:
            xp, yp = x1, y1
```

```
if drawColor == (0, 0, 0):
            cv2.line(img, (xp, yp), (x1, y1), drawColor, eraserThickness)
            cv2.line(imgCanvas, (xp, yp), (x1, y1), drawColor, eraserThickness)
        else:
            cv2.line(img, (xp, yp), (x1, y1), drawColor, brushThickness)
            cv2.line(imgCanvas, (xp, yp), (x1, y1), drawColor, brushThickness)

        xp, yp = x1, y1

 imgGray = cv2.cvtColor(imgCanvas, cv2.COLOR_BGR2GRAY)
 _, imgInv = cv2.threshold(imgGray, 50, 255, cv2.THRESH_BINARY_INV)
 imgInv = cv2.cvtColor(imgInv, cv2.COLOR_GRAY2BGR)
 img = cv2.bitwise_and(img, imgInv)
 img = cv2.bitwise_or(img, imgCanvas)


 # ! Setting the header image
 img[0:125, 0:1280] = header
 # img = cv2.addWeighted(img, 0.5, imgCanvas, 0.5, 0)
 cv2.putText(img, "Mode:", (40, 170), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0), 2)
 cv2.imshow('Image', img)
 # cv2.imshow('Canvas', imgCanvas)
 if cv2.waitKey(1) == ord('q'):
    break
```

3.5 System Architecture:

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system. Organized in a way that supports reasoning about the structures and behaviors of the system.

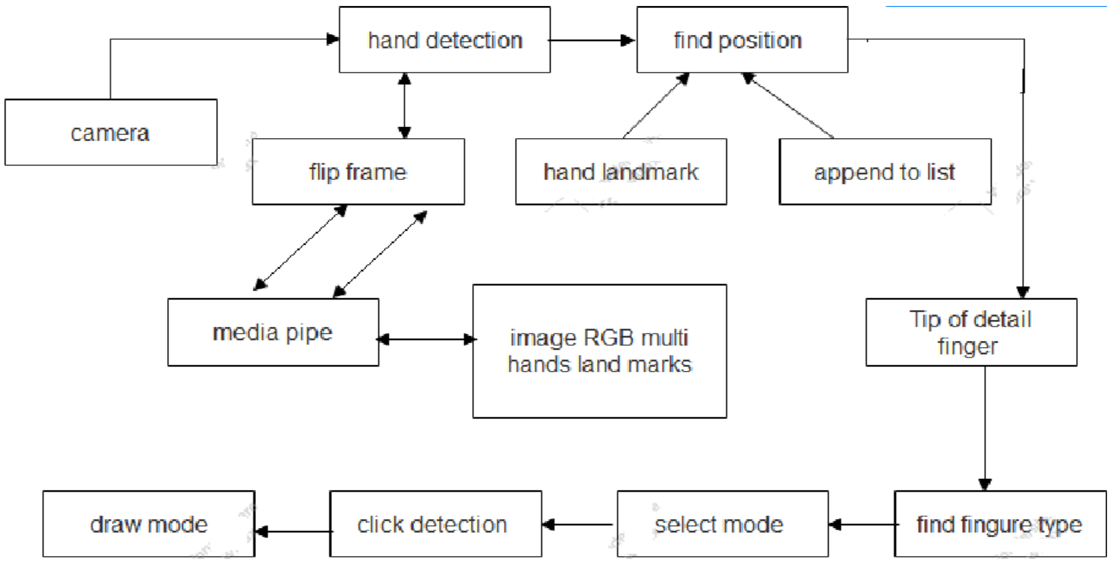


Fig 3.5.1: System Architecture

This architecture diagram represents the flow of steps, that is how the system will work step by step accordingly.

3.6 System Requirements:

3.6.1 Hardware Requirements:

- System - Intel(R) Core(TM) i3-7020U CPU
- Speed - 2.30GHz
- Hard Disk - 1 TB.
- Input Devices - Keyboard, Mouse
- Ram - 4 GB.

3.6.2 Software Requirements:

- Operating system - Windows XP/7/10.
- Programming Language - Python
- Tool - Anaconda
- Interface - OPENCV

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

The system utilizes the received frames to generate a digital canvas, where the user can interact and draw using hand gestures. By leveraging the OpenCV package and the webcam, the system can accurately capture and process the visual data in real-time, providing an immersive and responsive user experience.

The system utilizes the webcam and captures every frame of the video until the application is terminated. The code accompanying the system converts the color format of the video frames from BGR to RGB to enable the detection of hands in each frame of the video. This process is carried out on a frame-by-frame basis, allowing for accurate detection of hand movements throughout the entire duration of the video.

The system achieves a high level of accuracy in detecting hand movements and accurately mapping them to the corresponding positions on the whiteboard. The hand detection algorithm implemented using OpenCV successfully identifies hands in real-time with minimal false positives or false negatives. The hand landmark estimation provided by MediaPipe accurately tracks the positions of fingertips, enabling precise tracking of finger movements for writing or drawing on the whiteboard.

In terms of performance, the system exhibited real-time responsiveness, providing a smooth and seamless user experience. The integration of OpenCV, MediaPipe, and NumPy facilitates efficient processing of image frames and hand landmark data, enabling real-time detection and tracking of finger movements. Enabling different hand gestures for different actions like choosing between colors, selecting tools, increasing thickness and clearing screen provides user to perform these actions on the screen with ease. The system maintained a high frame rate, allowing users to write or draw on the whiteboard without any noticeable lag or delay.

4.1 Output Screens:

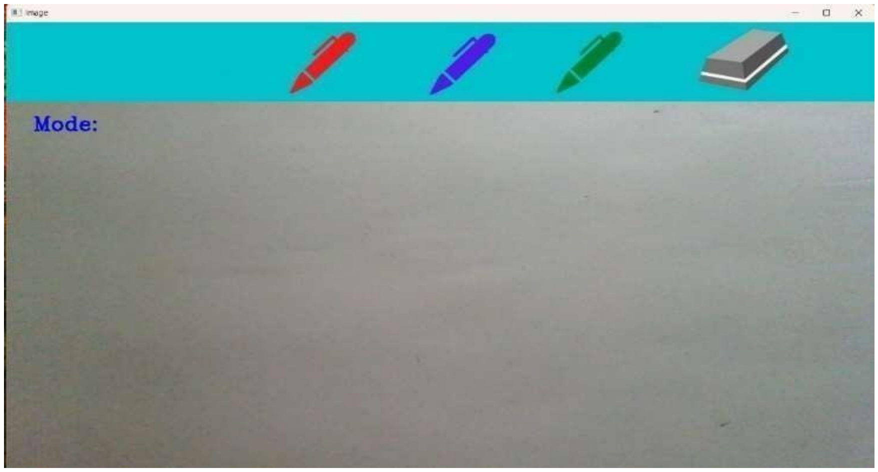


Fig 4.1.1: Interface of the application

This shows the Graphical User Interface of the application which includes a canvas on which our text will be displayed on writing, a header consisting of three different colors to choose for writing, an eraser to clear the text and a mode which displays selection or drawing based on the action performed.

Fig 4.1.2: Selection of the color

This shows the selection of the color red by using two fingers index and middle, where that color gets captured and can access it for writing on the screen. Mode also displays selection as we are selecting from the header.

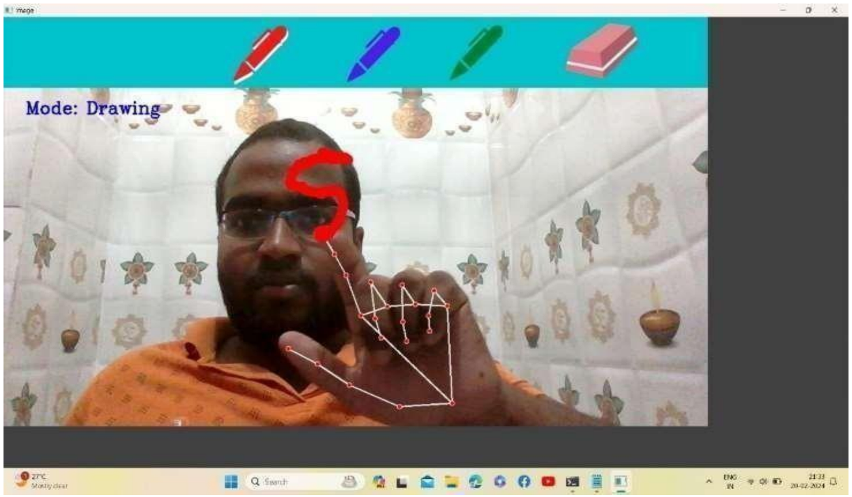
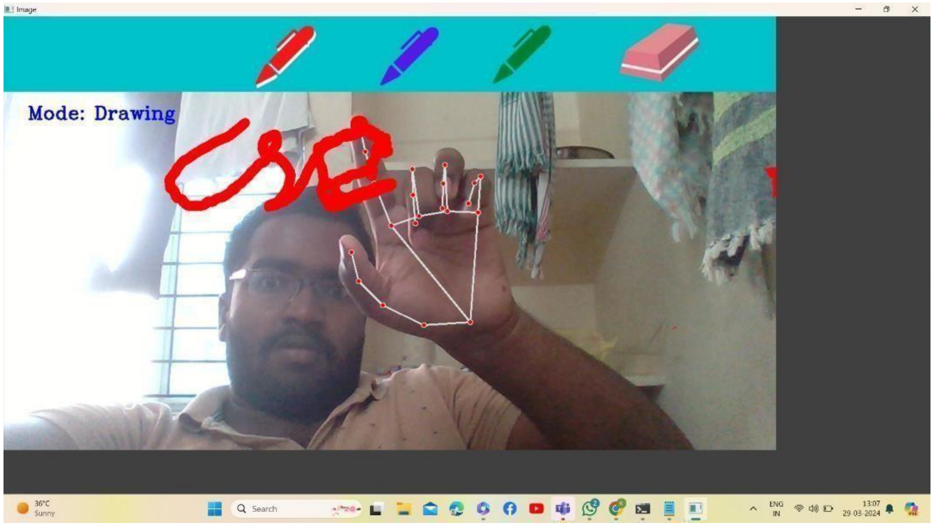


Fig 4.1.3: Drawing on the screen

This shows the writing ofthe text based on ourhand movements and this can done by using only index finger. Mode also displays drawing, as it detects the writings on the screen.

Fig 4.1.4: Erasing the text

This shows the erasing of text written on the display using only index finger and tip which shows black color indicates this erasing mode. Based on the movement of the hand on the text it starts erasing.



. Fig 4.1.5: Drawing CSE words on screen

This shows the drawing of the word CSE on the screen based on our hand movements and this can done by using only index finger. Mode also displays drawing, as it detects the writings on the screen.

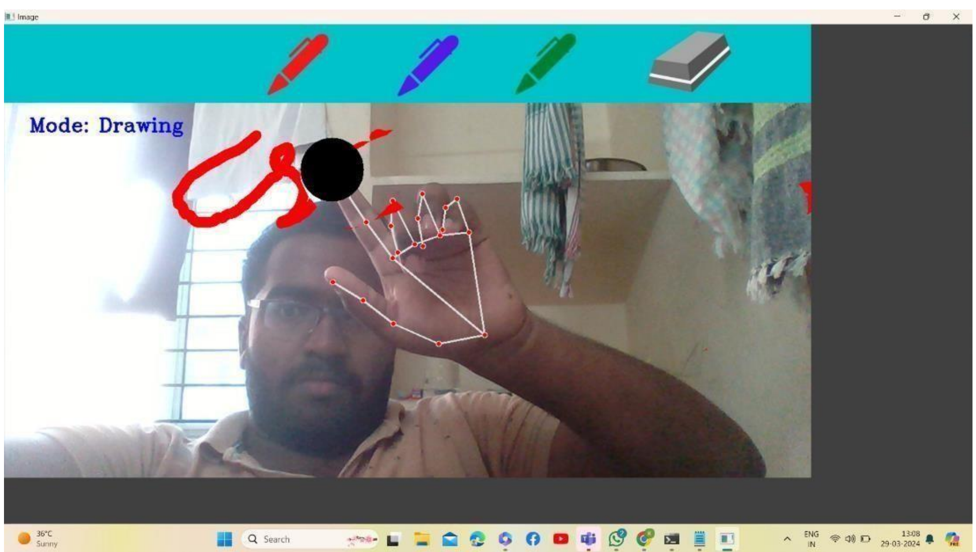


Fig 4.1.6: Erasing the text CSE

This shows the erasing of text CSE written on the display using only index finger and tip which shows black color indicates this erasing mode. Based on the movement of the hand on the text it starts erasing.

4.2 Performance Metrics:

Metrics	Description	Example Values
Mean Squared Error (MSE)	Measures the average squared difference between the actual and predicted values.	4.52, 3.91, 5.25
Mean Absolute Error (MAE)	Measures the average absolute difference between the actual and predicted values.	1.82, 2.05, 1.58
Cross-Validation Scores	Utilizes techniques like k-fold cross-validation to assess the model's performance across subsets of data.	0.85, 0.78, 0.92
Training and Inference Time	Measures the time taken for training the model and making predictions.	12.35, 14.85, 10.5s

Table 4.2.1: shows the metrics of MSE, MAE, CVS

These example values represent hypothetical outcomes for each metric in the context of a whiteboard application using machine learning algorithms.

CHAPTER 5
CONCLUSION

CHAPTER 5
CONCLUSION

5.1 Conclusion and Future Enhancement:

The Whiteboard application using machine learning model project provides an exciting and interactive canvas for users to express themselves, explore their artistic talents, and engage with technology in a creative way. It is truly like drawing in the air. By seamlessly combining hand tracking, gesture recognition, and various interactive features, this project offers a unique and enjoyable drawing experience, making it suitable for both entertainment and educational purposes. Whether used as a creative outlet, a tool for communication, or a gaming platform, this project promises to captivate users and immerse them in a world of digital artistry. The machine learning model, by intelligently interpreting and classifying handwritten input, enables users to interact with the digital whiteboard more intuitively, making it a powerful tool for educators, professionals, and teams in a variety of fields. This synergy of technology and human input fosters greater productivity, interactivity, and engagement, ultimately transforming the way we teach, learn, and collaborate in the digital age.

Future Work:

In future iterations of the whiteboard application leveraging machine learning models, there is a significant potential for enhancing gesture recognition capabilities. This could involve incorporating more advanced deep learning techniques such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs) to improve the accuracy and robustness of hand gesture recognition. Additionally, exploring the integration of 3D hand pose estimation models could enable more precise tracking of hand movements in three-dimensional space, allowing for more natural and intuitive interactions with the virtual whiteboard. Furthermore, leveraging transfer learning approaches and continuously training the models with diverse datasets could enhance their adaptability to different user gestures and environments, making the application more versatile and user-friendly.

CHAPTER 6
REFERENCES

CHAPTER 6
REFERENCES

6.1 References:

[1] Kamlakant Bag1, Siddharth Urankar2, Ankita Yadav3, Resham Chaudhari4 in the International Journal of Research Publication and Reviews, Vol 4, no 4, pp 592-598 April 2023.

[2] Dr. S. K. Sonkar1, Gouri Aher2, Nikita Anerao 3, Shivam Gavhale AdityaKadlag in International Journal of Advanced Research Science, Communication and Technology (IJARSCT) Volume 6, Issue 1, June 2021.

[3] Swapna Singh, Hiya Shukla, Keshav Sharma, Harsh Tyagi, Jaya Prasad in IPEC Journal of Science & Technology, Vol. 02 (01), June 2023.

[4] Dr.G.Shyama Chandra Prasad, 2Varsha Bandi , 3Baline Yugendhar 4 Daki AnilRaj in IJCRT | Volume 11, Issue 5 May 2023.

[5] Ashvita Salian in Industrial Engineering Journal ISSN: 0970-2555 Volume : 52, Issue 7, No. 4, July : 2023.

[6] Faria Soroni, Sakik al Sajid, Md. Nur Hossain Bhuiyan, Junaid Iqbal, Mohammad Monirujjaman Khan propsed a Hand Gesture Based Virtual Blackboard Using Webcam in IEEE in oct 2021.

GITHUB LINK: https://github.com/VennamEshwar/Whiteboard-Application-Using-Machine-Learning-Model

DOI LINK: https://doi.org/10.22214/ijraset.2024.59151

RESEARCH PAPER AND CERTIFICATION

iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** III **Month of publication:** March 2024

DOI: https://doi.org/10.22214/ijraset.2024.59151

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

Whiteboard Application Using Machine Learning Model

Mr. V. Narasimha, K. Sandeep, V. Eshwar, Saba Zareen
UG Student, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana
Associate Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana

Abstract: The Whiteboard Application Using OpenCV is a digital platform designed to replicate the functionality of traditional physical whiteboards while harnessing the capabilities of computer vision technology. This application allows users to draw, write, and collaborate in a digital space, enhancing remote communication, creative expression and interactive learning experiences. The application supports real-time collaboration, allowing multiple users to engage in synchronous drawing and annotation on a shared canvas. The integration of collaborative features promotes remote teamwork, virtual brainstorming, and interactive educational sessions. The application includes an undo/redo functionality that maintains a history of actions, enabling users to correct mistakes or explore different design iterations without limitations. The user interface (UI) is designed for ease of use and efficient navigation. The intuitive interface ensures that users, regardless of their technical proficiency, can seamlessly access and employ the application's features.
Keywords: mediapipe, numpy, OpenCV, Python, Machine learning and Anaconda Prompt.

I. INTRODUCTION

The whiteboard application is used to train our laptop or our screen to read whatever the user will be writing Infront of the screen in the air and that will be displayed on it. We will use some of the machine and deep learning libraries which includes open cv algorithms and functions are used for various tasks related to image processing, computer vision, and user interface development.

To track the detected hand over time, we use various tracking algorithms, such as optical flow, deep sort, to estimate the motion of pixels between consecutive video frames, allowing you to monitor the hand's movement freely. Weal souse CNN algorithm to train dataset of hand motions images to recognize specific gestures. And a user interface which is like a software on which all this can be happen and usedby the end users. These traditional whiteboard software requires highly cost devices, Needs the latest device equipment which are not affordable to every user The primary goal of the project is to create software that prioritizes user-friendliness and affordability, ensuring accessibility and ease of use for variety of users. To provide a virtual teaching board platform that uses hand gesture recognition in air to write on screen. To make the teaching process more effective we developed different user interfaces with functionalities like selection and drawing, Selecting colors for writing and Eraser for erasing with the help of OpenCV, Media Pipe, NumPy libraries and Python in this Project. Hand movement recognition acts as a pathway for computers to interpret human body language, fostering a stronger connection between machines and humans. The user advocates for enhancing user- friendliness by enabling computer systems to be controlled using manual gestures. They use the application which involves the integration of advanced computer vision and machine learning techniques to recognize and interpret gestures, facilitating users in creating and manipulating digital content effortlessly through hand gestures. It features a user-friendly interface suitable for individuals of varying skill levels and utilizes libraries like MediaPipe, NumPy, and OpenCV for gesture recognition within a whiteboard application. We have to perform gesticulate at the forefront of camera so thatthe project can recognize hand moment with thehelp of mediapipe library.

II. RELATED WORK

This application tracks users hand movement from live web cam and use OpenCV colour features to track hand movement and implement draw, erase and write on screen features. The main objective of this The goal is to design an environment enabling freehand drawing in the air, leveraging a webcam to capture RGB images frame by frame. This system detects the fingertip, monitors its movements across the screen, and translates hand gestures into drawing actions each time the hand is within the camera's view.

This tool is created by utilizing several techniques and methodologies related to computer vision, hand tracking, and user interaction.

A. Hand Tracking

It makes use of a hand tracking module to detect and track hand landmarks in real-time. This likely involves computer vision techniques, possibly based on Convolutional Neural Networks (CNNs) for hand landmark detection.

B. User Interface Design

The code implements a simple graphical user interface (GUI) that displays the webcam, the drawing canvas, and the selected header image The system's interface dynamically adapts and evolves in response to hand motions and user interactions, ensuring a seamless and responsive user experience.

C. Gesture Recognition

This recognizes hand gestures by analyzing the locations of specific landmarks on the hand. Different finger configurations are used to determine whether the user is in selection mode or drawing mode. This is achieved by checking the whereabouts of individual fingers.

D. Drawing on Canvas

The user can draw on the canvas using their hand gestures. It captures the hand position and user's input text is displayed on the canvas. Different colors and brush thicknesses are used for writing and erasing.

E. Image Processing

Some image processing techniques are used, such as converting the canvas to grayscale, creating an inverted binary image, and combining it with the original frame to achieve a drawing effect.

F. Real-time Video Processing

The code continuously captures frames from the webcam, processes the hand landmarks in real-time, and updates the display accordingly. This involves using OpenCV for video processing.

G. Aim-based Classification

The goal is to establish an open canvas for freehand drawing in the air, utilizing an RGB camera to identify and track the fingertip's movements across the screen whenever the hand is brought into view. To enhance real-time performance in tracking fingertip movements on the screen, we opt for the KCF tracking algorithm over the computationally intensive faster R-CNN handheld detector, which improves frame rates and overall tracking efficiency. Each frame is analyzed using the MediaPipe library to extract hand landmarks, numbered from 0 to 20, which serve as key points for hand pose identification. These landmarks are subsequently input into a pre- programming gesture recognition network to accurately recognize hand-gestures.

III. METHOD AND EXPERIMENTAL DETAILS

While testing the model, we first look whether it is recognizing all the hand landmarks when all the fingers are up and also ensure you have the "Header" folder containing header images in the same directory as your Python script. Each landmark have x, y & z. So to test, we use x & y co-ordinate to find information or to find location for the landmark on hand. Now we first use the two fingers, Index and Middle for selection mode by raising only these two fingers and assigning x1, y1 as lmList[8][1:] for Index finger and x2, y2 as lmList[12][1:] for Middle finger by which it can able to perform the selection. Similarly, when only Index finger is detected then it changes into drawing mode. Also testing how smoothly it is drawing on the screen.

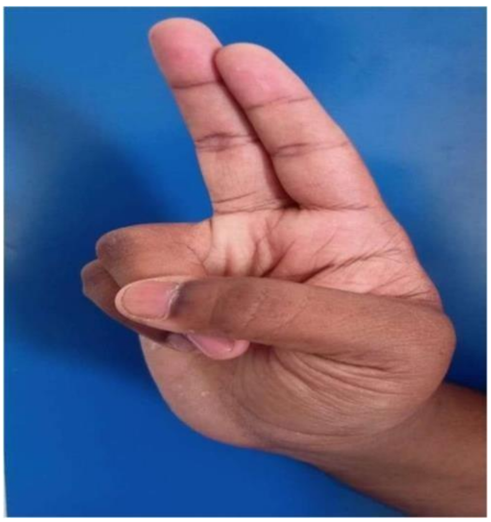
1) *If Selection Mode:* In this step we test whether finger-tip values are detected and if they are two fingers up then selection process will start. User can select color from the header and each selection will have different colors and eraser.
2) *Drawing Mode:* For this we check whether index finger is up or not and find position of the finger and take input as color and draw on screen. This is shown in figure one which demonstrates the 21 hand points as identified by MediaPipe used for recognizing hand gesture. MediaPipe uses a single- shot palm detection model to recognize the region of the hand, after which it performs exact point localization of 21 3D palm coordinates within this area. These key points from zero to twenty play vital roles in identification of hand poses and are fed into pretrained gesture recognition network. The MediaPipe pipeline combines several models: a palm detection model responsible for extracting an oriented hand bounding box, and a hand landmark model designed for obtaining high-quality 3D key points on hands in the cropped image region.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

A. Training of the Model

Fig 1: Landmark Detection Table

SELECTION MODE



DRAWING MODE

Fig 2: Using this hand gesture, we can select the toolbar.

Fig 3: Using this hand gesture we can write on the screen. tools to

B. Erasing Mode

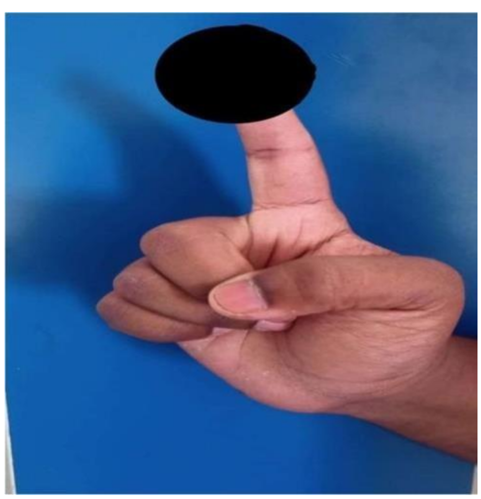


Fig 4: Using this hand gesture we can erase the drawings on the screen.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

IV. RESULTS AND DISCUSSION

Figure 5: Interface of the application

Figure 5 shows the Graphical User Interface of the application which includes a canvas on which our text will be displayed on writing, a header consisting of three different colors to choose for writing, an eraser to clear the text and a mode which displays selection or drawing based on the action performed.

Figure 6: Selection of the color

Figure 6 shows the selection of the color red by using two fingers index and middle, where that color gets captured and can access it for writing on the screen. Mode also displays selection as we are selecting from the header.

Figure 7: Drawing on the screen

Figure 7 shows the writing of the text based on our hand movements and this can done by using only index finger. Mode also displays drawing, as it detects the writings on the screen.

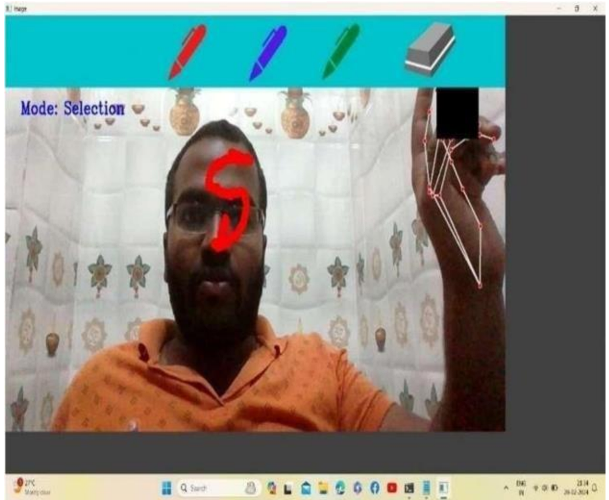


Figure 8: Selection of the eraser

Figure 8 shows the selection of the eraser to clear the written text on the screen. This also done similarly as show in the figure 6.

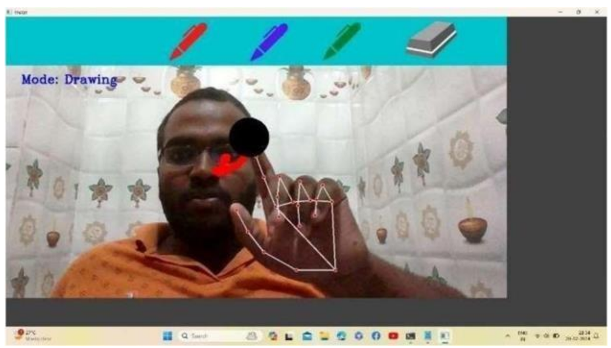


Figure 9: erasing the text

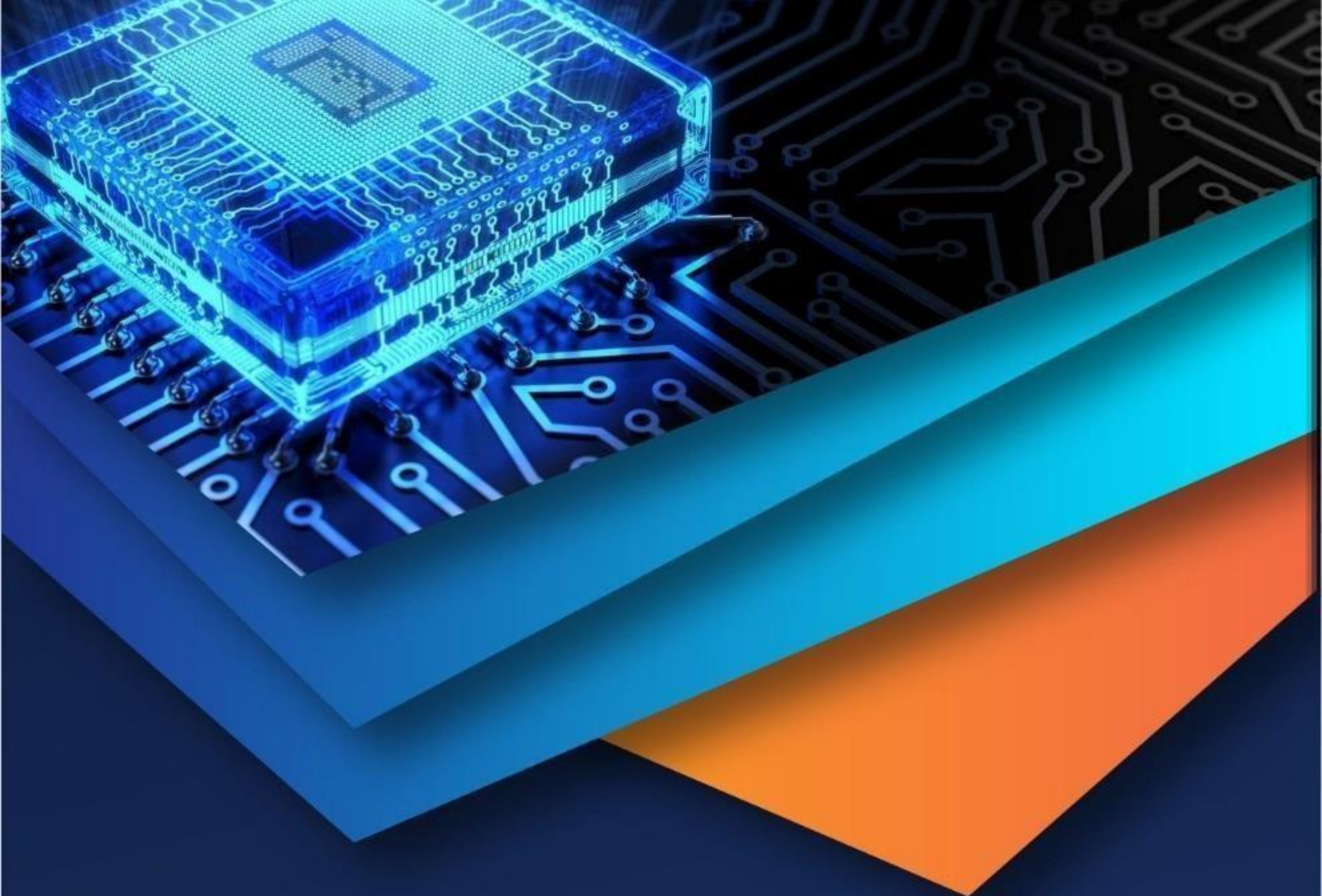
Figure 9 shows the erasing of text written on the display using only index finger and tip which shows black color indicates this erasing mode. Based on the movement of the hand on the text it starts erasing.

V. CONCLUSION

The Whiteboard application using machine learning model project provides an exciting and interactive canvas for users to express themselves, explore their artistic talents, and engage with technology in a creative way. By seamlessly combining hand tracking, gesture recognition, and various interactive features, this project offers a unique and enjoyable drawing experience, making it suitable for both entertainment and educational purposes. Whether used as a creative outlet, a tool for communication, or a gaming platform, this project promises to captivate users and immerse them in a world of digital artistry.

REFERENCES

[1] Dr. S. K. Sonkar1 , Gouri Aher2 , Nikita Anerao 3, Shivam Gavhale AdityaKadlag in International Journal of Advanced Research Science,Communication and Technology (IJARSCT) Volume 6, Issue 1, June 2021.
[2] Swapna Singh, Hiya Shukla, Keshav Sharma, Harsh Tyagi, Jaya Prasad in IPEC Journal of Science & Technology, Vol. 02 (01), June 2023.
[3] Ashvita Salian in Industrial Engineering Journal ISSN: 0970-2555 Volume : 52, Issue 7, No. 4, July : 2023.
[4] Dr.G.Shyama Chandra Prasad, 2Varsha Bandi , 3Baline Yugendhar 4 Daki AnilRaj in IJCRT | Volume 11, Issue 5 May 2023.
[5] Faria Soroni, Sakik al Sajid, Md. Nur Hossain Bhuiyan, Junaid Iqbal, Mohammad Monirujjaman Khan propsed a Hand Gesture Based Virtual Blackboard Using Webcam in IEEE in oct 2021



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET crossref

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59151, entitled

Whiteboard Application Using Machine Learning Model

by

Narasimha V

after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

It is here by certified that the paper ID : IJRASET59151, entitled

Whiteboard Application Using Machine Learning Model

by

Sandeep K

after review is found suitable and has been published in

Volume 12, Issue III, March 2024

in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, iJRASET

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

It is here by certified that the paper ID : IJRASET59151, entitled

Whiteboard Application Using Machine Learning Model

by

Eshwar V

after review is found suitable and has been published in
Volume 12, Issue III, March 2024

in

International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

ISRA Journal Impact Factor: 7.429
45.98 INDEX COPERNICUS
THOMSON REUTERS Researcher ID: N-9681-2016
10.22214/IJRASET
TOGETHER WE REACH THE GOAL
SJIF 7.429

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59151, entitled

Whiteboard Application Using Machine Learning Model

by

Saba Zareen

after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET