AUGMENTED WHITEBOARD USING COMPUTER VISION

INTERIM REPORT II

Submitted to the APJ Abdul Kalam Technological University

in partial fulfilment of requirements for the award of degree

Bachelor of Technology

in

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

by

ATHUL NATH M (JCE20CS017)

EMMANUEL ELDHO (JCE20CS021)

MUHAMMED SHEHIN SALIM (JCE20CS035)

RAHUL N A (JCE20CS044)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JAWAHARLAL COLLEGE OF ENGINEERING & TECHNOLOGY

Lakkidi, Mangalam P.O, Ottapalam, Palakkad, Kerala – 679301

MAY 2024

AUGMENTED WHITEBOARD USING COMPUTER VISION

INTERIM REPORT II

Submitted to the APJ Abdul Kalam Technological University

in partial fulfilment of requirements for the award of degree

Bachelor of Technology

in

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

by

ATHUL NATH M (JCE20CS017)

EMMANUEL ELDHO (JCE20CS021)

MUHAMMED SHEHIN SALIM (JCE20CS035)

RAHUL N A (JCE20CS044)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JAWAHARLAL COLLEGE OF ENGINEERING & TECHNOLOGY

Lakkidi, Mangalam P.O, Ottapalam, Palakkad, Kerala – 679301

MAY 2024

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JAWAHARLAL COLLEGE OF ENGINEERING & TECHNOLOGY

2023-24

CERTIFICATE

This is to certify that the Project Report entitled **AUGMENTED WHITEBOARD USING COMPUTER VISION** submitted by **ATHUL NATH M** (JCE20CS017), **EMMANUEL ELDHO** (JCE20CS021), **MUHAMMED SHEHIN SALIM** (JCE20CS035) & **RAHUL N A** (JCE20CS044) to the APJ Abdul Kalam Technological University in partial fulfillment of the B.Tech. degree in DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING is a bonafide record of the project work carried out by him/her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Mr. AMBARISH A
(Project Guide)
Assistant Professor
Dept.of CSE
JCET
Lakkidi

Ms. DIVYA VISAKH
(Project Coordinator)
Assistant Professor
Dept.of CSE
JCET
Lakkidi

Dr. R. SATHEESH KUMAR
Professor and HOD
Dept.of CSE
JCET
Lakkidi

DECLARATION

We undersigned hereby declare that the project phase-I report **AUGMENTED WHITEBOARD USING COMPUTER VISION**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala, is a Bonafide work done by us under the supervision of **Mr. AMBARISH A** , Assistant Professor, Department of Computer Science and Engineering, Jawaharlal College of Engineering and Technology. This submission represents our ideas in our own words and where ideas or words of others have been included; we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

ATHUL NATH M
EMMANUEL ELDHO
MUHAMMED SHEHIN SALIM
RAHUL N A

Lakkidi, Mangalam P.O, Ottapalam,
Palakkad, Kerala – 679301
03-05-2024

Abstract

Augmented whiteboard tech merges physical and digital realms via computer vision, revolutionizing virtual whiteboards with motion tracking and augmented reality (AR) for engaging learning. The system relies on an advanced camera setup to track user hand motions in real time, converting them into digital ink on a virtual whiteboard as they move through the air. Immersive online learning is made possible by seamlessly integrating physical motions with digital technology. This augmented whiteboard's unique feature is real-time collaboration.Realtime drawings are shown to viewers, including instructors and students, encouraging interaction. Due to its quick response it is suitable for remote learning. Teachers can use it for classroom clarity, student engagement, and comprehension. It serves as an interactive meeting tool for team members to visualize ideas instantly. Architects and designers can swiftly share sketches with clients.

Acknowledgement

We thank the Almighty God for giving us blessings to complete our project phase-I. We sincerely appreciate the inspiration support and guidance of all those people who have been instrumental in making this project a success. We would like to sincerely thank **Adv. Dr. P Krishna Das**, Managing Tr ustee of Nehru college Educational and Charitable Trust for making the resources available at right time and providing valuable insights leading to the successful completion of project phase-I. We express our sincere thanks to **Dr. N Gunasekaran**, The Principal of our college for supporting all the way long. We express our special gratitude to **Dr. R. Satheesh Kumar**, Head of the Department of Computer Science and Engineering for providing us constant guidance and encouragement throughout the project phase-I. We extend our warm gratitude to the Project Coordinators, **Ms. Divya Visakh and Mr. Ambarish A**, Assistant Professor, Department of Computer Science and Engineering for her inspiration and timely suggestions. Also Thanking our Project guide **Mr. Ambarish A**, Assistant Professor, Department of Computer Science and Engineering for her guidance and support. We have to appreciate the guidance given by the panel members during the project phase-I presentation, thanks to their comments and advices. Last but not the least We place a deep sense of gratitude to our family members and our friends who have been constant source of inspiration during the preparation of this project phase-I.

ATHUL NATH M

EMMANUEL ELDHO

MUHAMMED SHEHIN SALIM

RAHUL N A

Contents

List of Figures

List of Tables

List of Abbreviations

HSV	Hue, Saturation, Value
OpenCV	Open Source Computer Vision
AR	Augmented Reality
RAM	Random Access Memory
VTK	Visualization Tool Kit

Chapter 1

Introduction

The advent of the COVID-19 pandemic in late 2019 triggered a seismic shift in various aspects of our lives, profoundly impacting businesses and educational institutions worldwide. Among the affected professions, teaching underwent a transformative evolution, transitioning from traditional classroom settings to a virtual realm necessitated by global lockdowns and the surge in online activities. This shift to online education posed significant challenges, particularly for educators accustomed to conventional teaching methods honed over years or even decades. The move to online teaching demanded adaptation to new tools and technologies, with educators grappling with the limitations of virtual platforms. Many found themselves relying on rudimentary tools like a computer mouse, resulting in a pedagogical experience characterized more by squiggles than coherent instruction. The absence of adequate equipment in students' households compounded the difficulties of teaching to larger audiences. In response to these challenges, there emerged a transformative solution: the Augmented Whiteboard, leveraging the power of computer vision technology. This innovative tool, exemplified by the AirBoard, offers a revolutionary approach to online teaching. By seamlessly integrating digital pads with screens, the 'Augmented Whiteboard' provides educators with a dynamic canvas for instruction. This not only facilitates a more organic and comprehensible teaching method but also significantly reduces the associated costs. The Augmented Whiteboard goes beyond traditional online teaching tools, allowing teachers to project their content onto displays while utilizing the air as an interactive canvas. This breakthrough not only enhances the quality of instruction but also addresses the financial constraints faced by practitioners. As a result, educators can approach their virtual classrooms with greater confidence, knowing that they have a powerful and intuitive tool at their disposal to engage and instruct students effectively. The following exploration will delve deeper into the functionalities and benefits of the Augmented Whiteboard in transforming the landscape of online education.

1.1 Motivation

The impact of the COVID-19 pandemic has been widespread, affecting individuals in various ways. While some have directly experienced the effects of the virus, others have faced the loss of loved ones. With the closure of numerous colleges and universities, students have encountered challenges in accessing the education they require. The augmented reality experiment on 'Augmented Whiteboard' holds particular relevance in this context, offering a potential solution for secure and efficient distance learning experiences. Access to educational resources has become increasingly difficult, especially for those reliant on physical classrooms and libraries. By providing a remote, immersive, and interactive learning platform, this initiative addresses these challenges. Moreover, the project addresses the issue of social distancing during the pandemic by enabling virtual communication between students and educators, irrespective of physical proximity. Ultimately, this endeavor reflects a commitment to creativity and innovation, essential for navigating the complexities presented by the pandemic. Through the development of new tools and technologies, we can better equip ourselves to overcome current challenges and pave the way for a brighter future.

1.2 Objective

This concept uses augmented reality, motion tracking, and computer vision to bridge the gap between physical gestures and digital capabilities. Improving virtual education and encouraging instantaneous industry collaboration are the main goals. The goal of fusing digital and physical technology is to produce an immersive, interactive learning environment that encourages participation and comprehension. Students who don't have access to typical educational materials or who require extra help in their studies may find this to be especially helpful. Furthermore, this idea has the power to completely change the way we work together in a variety of industries. Regardless of location or time zone, this can promote more productive and efficient teamwork by enabling real-time communication and concept visualization. All things considered, the project may have a favorable effect on both industry and education. This thus makes a learning environment that is more dynamic and engaging for both users and educators by utilizing cutting-edge technology like augmented reality and computer vision.

1.3 Background Study

The use of augmented reality (AR) in education has gained significant attention in recent years due to its potential to enhance student engagement, motivation, and learning outcomes. AR technology overlays digital information onto the physical world, allowing students to interact with virtual objects and concepts in a real-world setting.

Studies have shown that AR can improve learning outcomes in various subjects, such as mathematics, science, and language. For instance, a study by Kim et al. (2018) found that AR-based math lessons improved students' understanding and retention of mathematical concepts compared to traditional teaching methods. Another study by Ketelhut et al. (2017) found that AR-based science lessons increased students' engagement and motivation in learning science.

However, there are still challenges and limitations to the widespread adoption of AR in education. One of the main challenges is the cost and availability of AR technology, which can be prohibitively expensive for many schools and educators. Another challenge is the lack of high-quality AR content and apps that are specifically designed for educational use.

Therefore, this project on aims to address some of these challenges by offering a low-cost, accessible, and easy-to-use platform for creating AR content and experiences. By leveraging computer vision, motion tracking, and machine learning, your project can provide a more immersive and interactive learning experience for students, regardless of their location or background.

1.4 Problem Statement

Developing an augmented whiteboard system with motion tracking and AR to enhance online learning, collaboration, and real-time idea visualization.

1.5 Chapterwise Summary

The purpose of this project is to develop an Augmented Whiteboard using Computer Vision that enhances online learning experiences. This chapter provides an overview of the project objectives, methodology, and the potential benefits of the proposed system. In Chapter 2, we review the existing literature on computer vision and augmented reality in education, including their potential benefits and limitations. We also discuss the current state-of-the-art solutions for augmented whiteboards and their limitations. Chapter 3 describes the development process of the Augmented Whiteboard using Computer Vision, covering the hardware and software requirements, the data collection process, and the algorithm used for object detection and tracking. In Chapter 4, we present the results of the experiment conducted to evaluate the effectiveness of the system, discussing the performance metrics used to measure the success of the system and analyzing the results in detail. Finally, in Chapter 5, we conclude the project by summarizing the key findings and the potential benefits of the Augmented Whiteboard using Computer Vision, discussing the limitations of the system, and suggesting future directions for research and development.

Chapter 2

Literature Review

2.1 Augmented Reality for Studying Hands on the Human Body for Elementary School Students. [1]

This paper focuses on hands-on learning in elementary education with a specific emphasis on studying the human body. While the methodology is not explicitly stated, the authors highlight the engagement of elementary students with the human anatomy. The paper underscores the potential of augmented reality (AR) to enhance the educational experience for young learners.

2.2 Development of Augmented Reality (AR) for Innovative Teaching and Learning in Engineering Education. [2]

Enzai and team present an exploration of AR development for innovative teaching in engineering education. The methodology is not explicitly detailed, but the paper underscores the positive impact of AR on fostering innovation within engineering education. It offers insights into leveraging AR for more effective and engaging teaching methods.

2.3 Mobile Augmented Reality in Vocational Education and Training. [3]

This paper delves into the realm of mobile augmented reality (AR) applied in vocational education and training. While it highlights the advantages of utilizing AR in this context, the authors do not extensively discuss potential limitations related to the use of mobile technology in educational settings.

2.4 Enhancing the Attractiveness of Learning through Augmented Reality. [4]

Iftene and Trandabăț explore the role of augmented reality in increasing the attractiveness of learning materials. Despite lacking detailed information on potential challenges, the paper emphasizes the positive impact AR can have on enhancing the appeal of educational content.

2.5 Development of an Interactive Training Support System for Drawing and Painting Skill on a 3D Object with AR and Haptic Interface. [5]

Shima and Soga's paper focuses on interactive training for drawing and painting skills using AR and haptic interfaces. While emphasizing skill development on 3D objects, the paper does not extensively detail the effectiveness of the haptic interface in this context.

2.6 Mathematics and Geometry Education with Collaborative Augmented Reality. [6]

This paper explores the integration of collaborative augmented reality in mathematics and geometry education. While highlighting the positive enhancements in these subjects, the authors do not thoroughly discuss the collaborative aspects of the proposed methodology.

2.7 Augmented Reality in Teaching and Learning English Reading: Realities, Possibilities, and Limitations. [7]

Jamrus and Razali's paper explores the application of augmented reality (AR) in teaching English reading, with a predominant focus on its possibilities rather than a thorough examination of potential limitations. The authors extensively discuss the positive aspects of AR, emphasizing immersive learning experiences and heightened engagement. However, the paper lacks a comprehensive analysis of challenges that may arise during AR integration, such as technological constraints, costs, and potential distractions. A more balanced approach, considering both benefits and limitations, is crucial for informed decision-making by educators and stakeholders in the evolving landscape of AR in English reading education.

2.8 Crowd-board: An Augmented Whiteboard to Support Large-Scale Co-design. [8]

This paper introduces "Crowdboard," an augmented whiteboard explicitly crafted to streamline extensive co-design projects. By prioritizing robust support for large-scale collaborative efforts, the authors also conscientiously address potential challenges inherent in such expansive collaborations. The contribution of this paper extends beyond the introduction of Crowdboard by offering valuable insights into the dynamics of collaborative design projects and the innovative application of augmented reality to foster collective creativity.

Table 2.1: Literature Review Table 1

Paper Name	**Methodology**	**Merits and Demerits**	**Relevance On Project Area**
Augmented reality for studying hands on the human body for elementary school students. By Richi Ruslia, Darryl Arkan Nalandaa, Antonius Doff Valma Tarmidia, Kristien Margi Suryan-ingruma	**Marker Tracking**: Detect marker position and orientation using QR codes or images for 3D virtual world creation. **AR Startup**: Initiate the AR system upon marker recognition. **AR App Development**: Create augmented reality applications. **Data Management**: Implement marker tracking functionality and use qualitative methods for data collection and analysis.	**Merits**: 1. Interactive Learning 2. Deeper Insight: Qualitative data collection 3. Tangible Study Aid: hands-on tool. **Demerits**: 1. Marker Limitation: Marker dependency can restrict flexibility. 2. Technical Skill Needed: Designing AR apps and trackers demands technical expertise. 3. Generalizability Concern: A limited experimentation phase may affect broader applicability.	1. Relevant to the development of AR media as a learning medium for elementary school students. 2. Provides a hands-on learning experience for understanding body parts.

Table 2.2: Literature Review Table 2

Paper Name	**Methodology**	**Merits and Demerits**	**Relevance On Project Area**
Development of Augmented Reality (AR) for Innovative Teaching and Learning in Engineering Education. By Nur Idawati Md Enzai, Norhayati Ahmad, Mohd Amir Hamzah Ab. Ghani, Siti Sara Rais, Syazilawati Mohamed	1. The AR system was developed using the Assembler application. 2. It involves creating 3D images and AR markers.	**Merits**: 1. Suitable for online teaching. 2. Addresses facilities constraints and safety issues in engineering education. 3. Increases student interest. 4. Improves understanding of complex subjects. **Demerits**: 1. Limited by the free version of Assembler app. 2. Quality of 3D images can deteriorate with multiple conversions.	It is relevant to the project area, as it discusses the development of AR for engineering education, which aligns with the project's goals of utilizing AR for educational purposes.

Table 2.3: Literature Review Table 3

Paper Name	**Methodology**	**Merits and Demerits**	**Relevance On Project Area**
Mobile Augmented Reality in Vocational Education and Training. By Jorge Baccaa, Silvia Baldirisa, Ramon Fabregata, Kinshukb, Sabine Grafb	1. Using collaborative co-creation, the ”Paint-cAR” mobile AR app is developed for vocational car maintenance education, integrating the Universal Design for Learning (UDL) framework. 2. A cross-sectional evaluation study was conducted to validate the application.	**Merits**: 1. Helpful for vocational education. 2. Uses mobile AR markers to guide students through a complex process. Incorporates various learning activities, including tools and specifications. **Demerits**: 1. Attention dimension could be improved by addressing distractions in the workshop. 2. Limited time spent with the application hinders assessing the long-term applicability of concepts.	It uses Augmented Reality for Training and explaining complex concepts.

Table 2.4: Literature Review Table 4

Paper Name	**Methodology**	**Merits and Demerits**	**Relevance On Project Area**
Enhancing the Attractiveness of Learning through Augmented Reality. By Adrian Iftene, Diana Trandabă.	1. The paper introduces four augmented reality eLearning applications focusing on enhancing communication, collaboration, and learning in biology and geography. 2. Usability tests involving professors and students evaluate the impact of collaborative and non-collaborative settings on the effectiveness of these AR applications.	**Merits**: 1. Attractive user interfaces. 2. Positive feedback from professors and students 3. Reducing stress through game-based evaluations. **Demerits**: 1. Some minor performance and stability issues. 2. Confusion in the user interface for some users.	Highly relevant to eLearning, particularly in improving student engagement, commu-nication skills, and collaborative learning. The paper's findings suggest that AR applications can make learning more attractive and effective for students.

Table 2.5: Literature Review Table 5

Paper Name	**Methodology**	**Merits and Demerits**	**Relevance On Project Area**
Development of an Interactive Training Support System for Drawing and Painting Skill on a 3D Object with AR and Haptic Interface. By Haruki Sima, Masato Soga	1. The paper presents a system for teaching drawing and painting on virtual 3D objects. 2. It uses a haptic device for virtual brush control and augmented reality (AR) to overlay virtual and real 3D objects.	**Merits**: - 1. Effective use of haptic device for tactile feedback. 2. AR facilitates comparison between real and virtual objects. 3. Support for various painting methods. **Demerits**: - 1. Virtual brush operation might not fully replicate the feel of a real brush. 2. System complexity may require an adjustment period for users.	The paper's approach to teaching drawing and painting on 3D objects with haptic and AR technologies aligns with the project's focus on incorporating AR into art education.

Table 2.6: Literature Review Table 6

Paper Name	**Methodology**	**Merits and Demerits**	**Relevance On Project Area**
Mathematics and Geometry Education with Collaborative Augmented Reality. By Hannes Kaufmann, Dieter Schmalstieg	1. Used Construct3D, an AR-based 3D tool for math and geometry education, promoting spatial skills and enabling teacher-student collaboration. 2. It discusses its application in high school and university education.	**Merits**: 1. Encourages experimentation with geometric constructions. 2. Improves spatial skills. 3. Supports various teacher-student interaction scenarios. **Demerits**: 1. Limited to specific hardware setups. 2. Hand-eye coordination challenges. 3. Need for continuous development of educational content.	The Augmented Reality approach to make the education more immersive and interactive.

Table 2.7: Literature Review Table 7

Paper Name	Methodology	Merits and Demerits	Relevance On Project Area
Augmented Reality in Teaching and Learning English Reading: Realities, Possibilities, and Limitations. By Mohd Hafizuddin Mohamed Jamrus and Abu Bakar Razali	1. Prior research supports AR's potential for motivating English language learners. 2. This study assesses AR in English reading instruction, covering its concept, merits, challenges, and teacher preparedness. It offers guidance for potential classroom use. 3. It highlights AR's growing importance in education and suggests future research avenues for English reading instruction.	**Merits**: 1. AR adds interactivity, enhancing engagement and comprehension. 2. It aids 3D visualization for easy understanding 3. AR simulates real scenarios for practical learning. **Demerits**: 1. Unequal AR technology access can lead to educational disparities. 2. Implementing AR may require substantial technical resources and support, posing a barrier for some educational institutions	Augmented Reality (AR) in education holds a significant relevance in enhancing education

Table 2.8: Literature Review Table 8

Paper Name	**Methodology**	**Merits and Demerits**	**Relevance On Project Area**
Crowdboard: An Augmented Whiteboard to Support Large-Scale Co-Design. By Salvatore Andolina, Daniel Lee, Steven Dow	1. Node.js server and MongoDB database for system status sync using socket.io. 2. Web interface (Backbone.js) for the crowd. 3. Web interface (Backbone.js) for the team (projector-connected computer). 4. C# routine for depth-camera interaction. 5. Java and OpenCV module for whiteboard screenshot capture and upload.	**Merits**: 1. Involves complexity and cost. 2. Enhances real-time collaboration for holistic designs. 3. Includes remote participants to encourage diverse input. 4. Engages motivated stakeholders, driving innovation. **Demerits**: 1. Involves complexity and cost. 2. Raises data security and privacy concerns. 3. Managing input from many participants can overwhelm design teams.	The Crowdboard system is crucial for early-stage design, promoting real-time engagement with a diverse stakeholder community, enhancing inclusivity.

Chapter 3

System Development

3.1 System Architecture

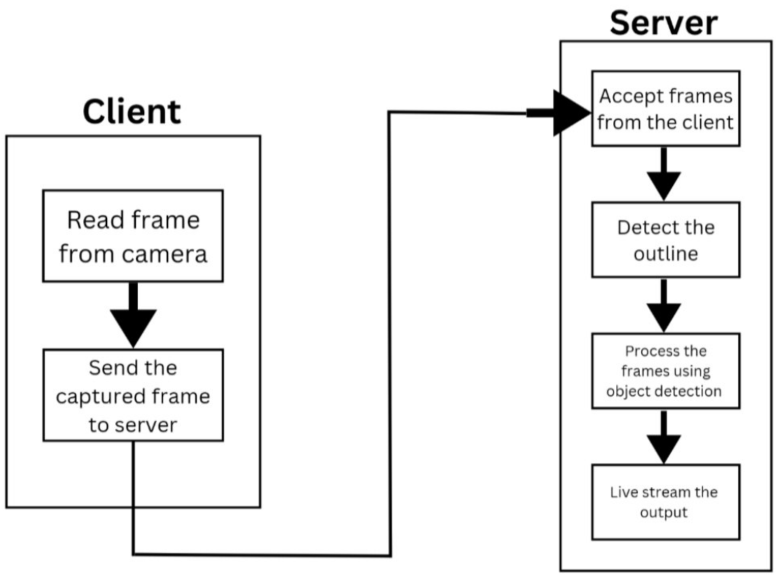


Figure 3.1: System Architecture

The diagram shows the process of sending frames from a client, such as a camera, to a server for real-time object detection. The camera captures frames and sends them to the server. The server receives the frames, detects the outlines of objects in the frames, processes the frames for object detection, and then live streams the output.

3.2 Dataflow Diagram

The Data Flow Diagram shows the working of an Augmented White Board Using Computer Vision. The Data Flow Diagram outlines the steps involved in the process of object detection and identification. It shows the process done at various levels.

3.2.1 Level 0

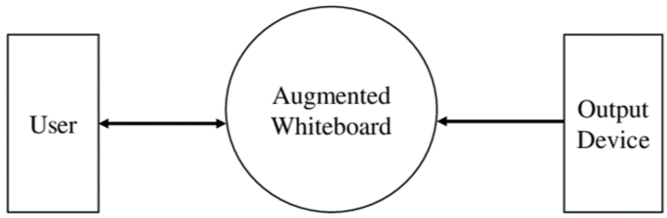


Figure 3.2: Level 0

The Level 0 consists of the following:

1. **User:** Users interact with the whiteboard.The user can draw by using the finger.
2. **Augmented Whiteboard:**The Augmented Whiteboard recognizes and tracks the movement of the finger to draw the content.It displays the user and the content drawn by the user through output devices.
3. **Output Device:** The output device shows the content drawn by the user through output device.

3.2.2 Level 1

The Level 1 shows the functions performed by the Augmented Whiteboard.It involves the following processes:

1. **Image/Frame Acquisition:** The input is taken through the camera as frames.Once the object is detected, this step involves capturing the object from the image. This can be done using various techniques such as cropping, masking, etc.

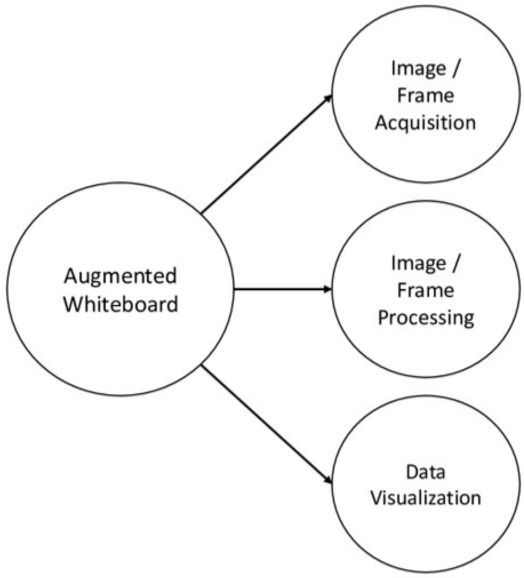


Figure 3.3: Level 1

2. **Image/Frame Processing:** The images/frames are processed after detecting outlines and objects, additional processing may be applied. This could include filtering noise, refining object boundaries, or any other enhancements to improve the accuracy of the augmented experience.

3. **Data Visualization:** The output device shows the content drawn by the user through output device.

3.2.3 Level 2.1

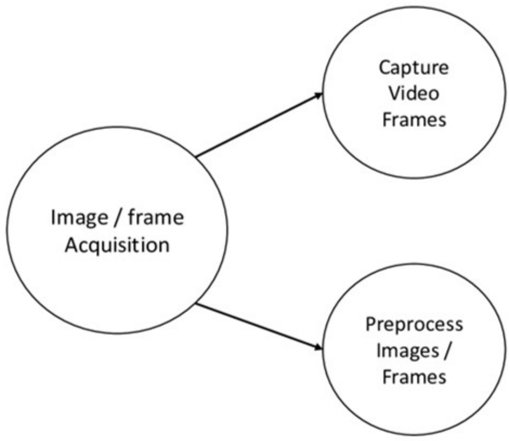


Figure 3.4: Level 2.1

The Level 2.1 involves the following processes:

1. **Capture Video Frames:** The input is taken through the camera as frames.Once the object is detected, this step involves capturing the object from the image. This can be done using various techniques such as cropping, masking, etc.

2. **Preprocess Images/Frames:** The images/frames are processed after detecting outlines and objects, additional processing may be applied. This could include filtering noise, refining object boundaries, or any other enhancements to improve the accuracy of the augmented experience.

3.2.4 Level 2.2

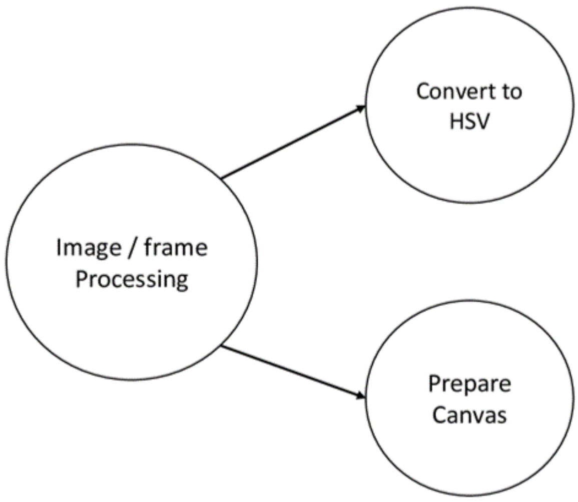


Figure 3.5: Level 2.2

The Level 2.2 involves the following processes:

1. **Convert to HSV:** The captured frames are converted into HSV (Hue Saturation Value) Frame for the ease of colour detection.

2. **Prepare Canvas:** The images/frames are processed after detecting outlines and objects.The canvas is prepared.

3.2.5 Level 2.3

The Level 2.3 involves the following processes:

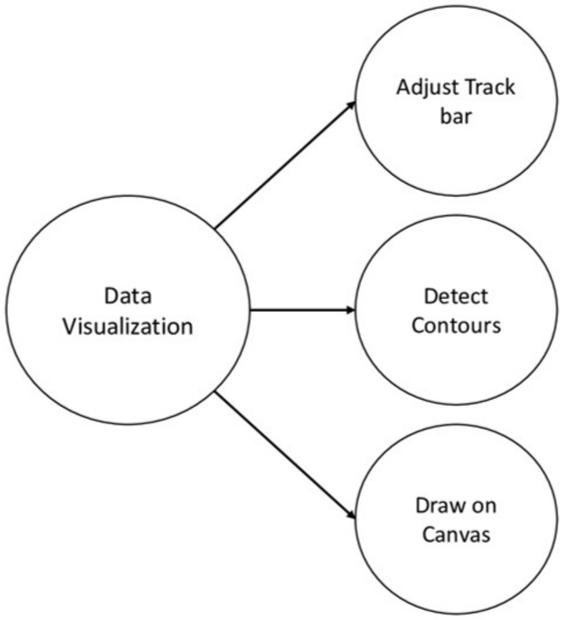


Figure 3.6: Level 2.3

1. **Adjust Track bar:** The track bar is adjusted to track the motion of the object. The motion tracker detects the tip of the finger.

2. **Detect Contour:** Computer vision algorithms, such as edge detection or contour finding, is employed to identify the boundaries of objects in the frames.

3. **Draw on Canvas:** The contents are drawn by the user on the canvas.The contents are displayed on the screen. The user can select the colour from the colour panel and draw on the canvas.

3.3 Module Description

3.3.1 Client Side

1. **Frame Capturing Module:** The process begins with a camera capturing video frames of the whiteboard or the surrounding area. The camera can be a standalone device or integrated into a computer or mobile device.

2. **Frames Sending Module:** The client, which can be a computer or any smart device, sends the captured video frames to a remote server for processing. This

step often involves establishing a network connection between the client and the server.

3.3.2 Server Side

1. **Frames Accepting Module:** The server receives the frames transmitted by the client. This might involve a server waiting for incoming connections or polling for data.

2. **Outline Detection Module:** The server processes each frame to identify the outlines of objects on the whiteboard. Computer vision algorithms, such as edge detection or contour finding, can be employed to identify the boundaries of objects.

3. **Object Detection Module:** Beyond simple outline detection, the server can use more advanced object detection technology to identify specific objects within the frames. This might involve the use of pre-trained models (such as YOLO, SSD, or Faster R-CNN) to recognize and locate objects in real-time.

4. **Frame Processing Module:** After detecting outlines and objects, additional processing may be applied. This could include filtering noise, refining object boundaries, or any other enhancements to improve the accuracy of the augmented experience.

5. **Output Module:** The server, now enriched with information about the whiteboard content, live streams the augmented frames back to the client. The live stream can be sent in real-time, allowing users to view an enhanced version of the whiteboard with additional information or annotations.

3.4 System Requirements

3.4.1 Hardware Requirements

1. **Camera:** A high-resolution camera (at least 720p) with a wide field of view is required to capture the user's hand movements and the surrounding environment.

2. **Processor:** A powerful processor (at least 1.8 GHz) is needed to process the video stream and perform the necessary computations for augmented reality. For this a CPU or an SoC (System-on-Chip) with a built-in GPU can be used.

3. **Memory:** At least 4 GB of RAM is required to store the video frames, computer vision algorithms, and AR content. More memory will allow for smoother performance and more complex AR effects.

4. **Storage:** A fast storage device (such as an SSD) is necessary to store the video files, computer vision models, and AR content. At least 16 GB of storage capacity is recommended.

5. **Display:** A high-resolution display (at least 1080p) is required to display the AR content and the user interface. You can use a monitor, TV, or a specialized display for augmented reality applications.

6. **Operating System:** A 64-bit version of Windows, macOS, or Linux is recommended.

3.4.2 Software Requirements

1. **Python:** A robust and versatile programming language like Python is essential for developing an AR application. You can use Python 3.x or higher for your project.

2. **OpenCV:** A computer vision library like OpenCV is necessary for image processing, object detection, and feature extraction. OpenCV provides a wide range of functions for these tasks, including image filtering, shape detection, and object recognition. For example, OpenCV's Haar cascades can be used for object detection and tracking.

3. **scikit-image:** A machine learning library like scikit-image is useful for image processing and feature extraction. It provides a variety of functions for image analysis, including edge detection, thresholding, and morphological operations.

4. **PyOpenGL:** An OpenGL library like PyOpenGL is required for creating 3D graphics and animations. PyOpenGL provides a high-level API for rendering

3D objects, lighting, and shading, which are critical for an immersive AR experience.

5. **VTK:** A 3D visualization library like VTK (Visualization Toolkit) is useful for creating interactive 3D visualizations and simulations. It provides a wide range of functions for visualizing data, including surface reconstruction, volume rendering, and mesh generation.

6. **NumPy and SciPy:** Numerical computing libraries like NumPy and SciPy are essential for scientific computing and data analysis. They provide a wide range of functions for linear algebra, optimization, statistics, and other tasks that are critical for an AR application.

7. **Matplotlib and Seaborn:** Data visualization libraries like Matplotlib and Seaborn are useful for visualizing data and creating interactive visualizations. They provide a variety of functions for creating graphs, charts, and other visualizations, as well as handling events and user input.

Chapter 4

Results and Discussion

4.1 Methodology

In the initial phase of this interactive airboard system, the process begins with Frame Initialization. This involves capturing video frames and subsequently converting them to the HSV color space, a crucial step in enhancing color-based processing capabilities. Following this, the system proceeds to Canvas and Ink Setup, where an augmented reality canvas is established, featuring ink buttons designed for color selection. This canvas serves as the virtual workspace for the ensuing creative interaction.

The third stage, Color Detection and Tracking, is pivotal in ensuring precision and responsiveness. During this phase, the system fine-tunes color detection parameters, enabling it to accurately identify and track the center coordinates of a marker. This meticulous tracking is essential for maintaining a seamless and intuitive user experience.

Finally, the culmination of these processes is observed in Real-time Drawing. Leveraging the tracked coordinates obtained from the Color Detection and Tracking phase, the system dynamically draws points on both the captured video frames and the augmented reality canvas. This integration results in an immersive and interactive airboard experience, where users can engage in real-time drawing with precision and fluidity. The synergy of frame initialization, canvas and ink setup, color detection, and real-time drawing contributes to the development of a sophisticated and engaging augmented reality platform.

4.2 Work Done

The project ”Augmented Whiteboard using Computer Vision” has made significant strides in its development, with a particular focus on the implementation of the front end. This crucial phase involved the creation and integration of user interfaces, ensuring a seamless and intuitive experience for the end-users. The development team successfully designed and implemented the graphical elements, user interactions, and overall layout of the augmented whiteboard application. Through meticulous attention to detail, the front-end development phase aimed to enhance user accessibility, responsiveness, and overall satisfaction. The utilization of computer vision technology adds a cutting-edge dimension to the traditional whiteboard, allowing for the recognition and interpretation of user input in real-time. As the project progresses into the development phase, the front-end achievements lay a solid foundation for the integration of the computer vision algorithms, promising an innovative and interactive augmented whiteboard solution. The collaboration between front-end development and upcoming back-end implementations is poised to deliver a comprehensive and user-friendly experience in the final product.

4.3 Algorithm

1. **START**:Initialize the program with required libraries and configurations.
2. **Importing Libraries**: The code starts by importing necessary libraries including OpenCV, Mediapipe, NumPy, and os.
3. **Setting up Video Capture**: It initializes the video capture from the webcam, sets the resolution, and creates a canvas for drawing.
4. **Loading Header Images**: It loads header images from a specified directory and stores them in a list.

5. **Setting Presets**: It initializes variables for drawing such as color, thickness, and fingertip IDs.

6. **Main Loop**: The code enters the main loop where it continuously captures frames from the webcam.

7. **Hand Detection and Landmarks**: It uses the MediaPipe Hands model to detect hands and landmarks in each frame.

8. **Hand Gesture Analysis**: It analyzes hand gestures to determine the user's intent:

   - **Selection Mode**: Detects when two specific fingers are up to change colors.
   - **Standby Mode**: Detects when the index and pinky fingers are open but not drawing.
   - **Draw Mode**: Detects when one finger is up and draws lines on the canvas.
   - **Clear Canvas**: Clears the canvas when all fingers are closed.
   - **Thickness Adjustment**: Adjusts the thickness of the drawing line based on the distance between the index finger and thumb.

9. **Drawing and Display**: It draws on the canvas based on the user's gestures and overlays the canvas on the camera frame. It also updates the header image based on the selected color.

10. **Display and Termination**: It displays the camera feed with the drawing overlay and waits for the user to press 'q' to exit the application.

11. **Cleanup**: Once the loop is exited, it releases the video capture and closes all OpenCV windows.

12. **STOP**:Terminate the program based on user input or predefined conditions.

4.4 Implementation

In the first phase of the project, the focus was on implementing the foundation for an "Augmented Whiteboard using Computer Vision." The project aimed to address the identified problem of enhancing online learning, collaboration, and real-time idea visualization through the development of an augmented whiteboard system. The initial steps involved a comprehensive literature review to understand existing solutions and technologies. Subsequently, an algorithm was prepared to integrate motion tracking and augmented reality (AR) features. The interface was also carefully designed to ensure user-friendly interaction. This phase laid the groundwork for the subsequent stages, providing a solid framework for the development and integration of computer vision capabilities into the augmented whiteboard system.The user can choose the colour from the panel by simply pointing the tip of finger on the colour, and draw with the tip of finger with the corresponding colour chosen.The tip of the finger is tracked by motion tracker and the digital ink is displayed. The Eraser button is used for erasing the drawn content.

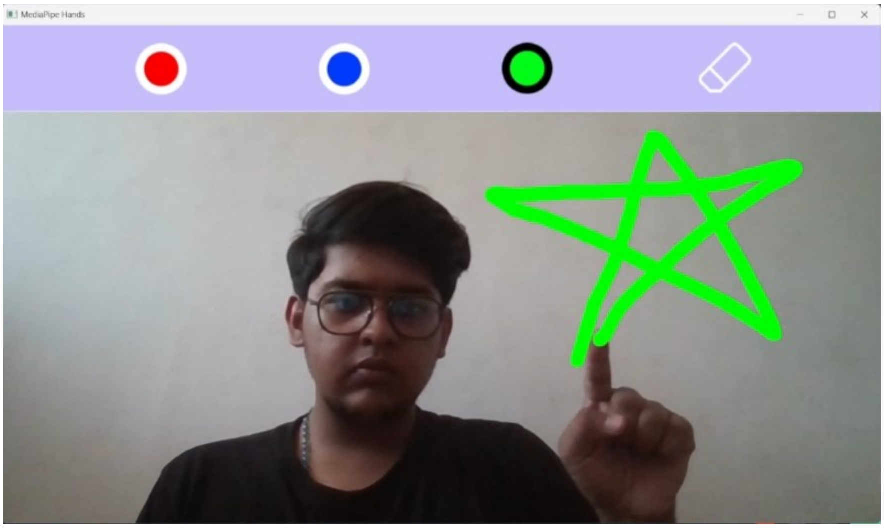


Figure 4.1: Draw

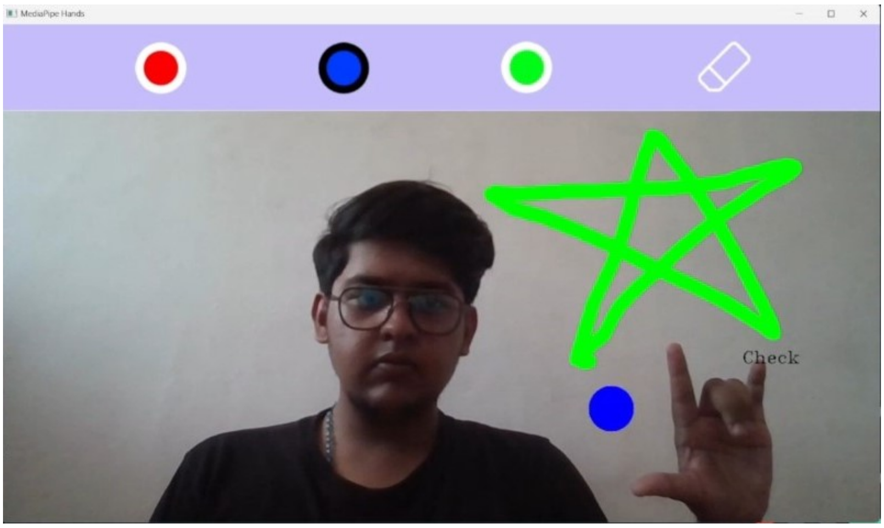


Figure 4.2: Check

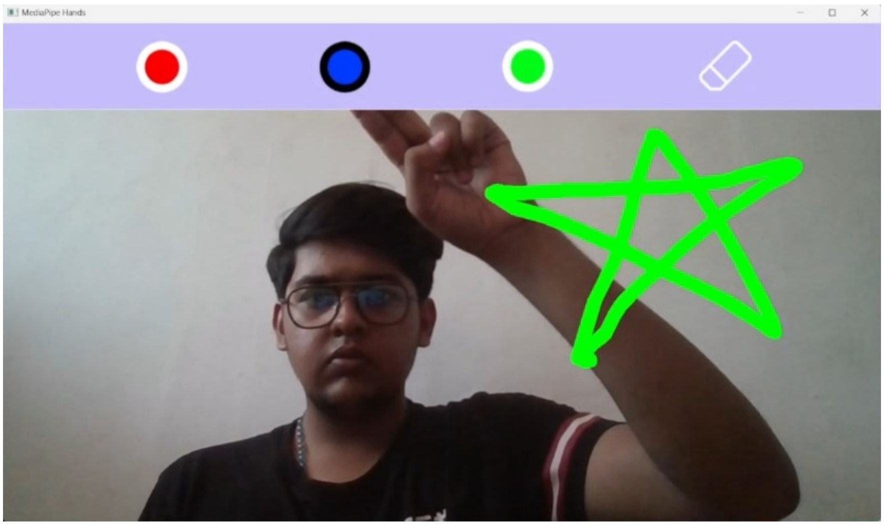


Figure 4.3: Select Colour

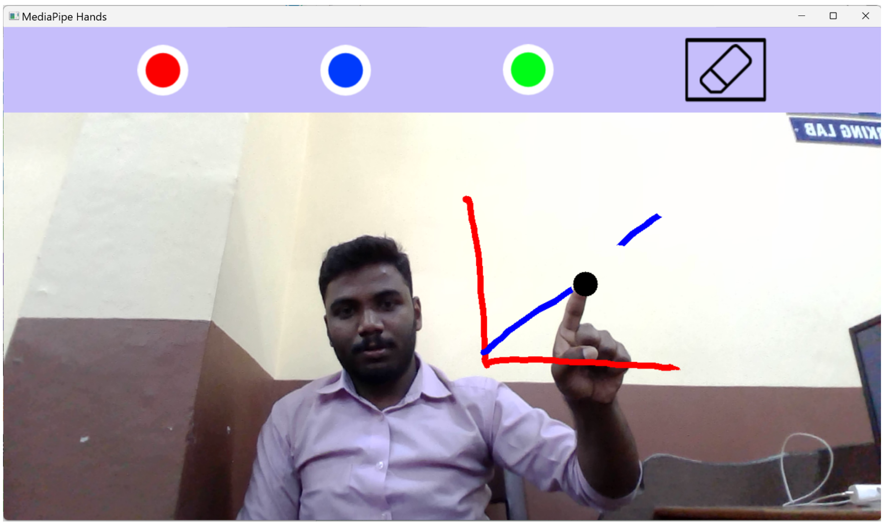


Figure 4.4: Erase

Chapter 5

Conclusion

The augmented whiteboard stands at the forefront of innovation, seamlessly merging the tangible and virtual realms through the synergistic integration of computer vision, motion tracking, and augmented reality technologies. This groundbreaking fusion holds transformative potential, particularly in the realm of education, where it can revolutionize the learning experience. At its core, the augmented whiteboard is a testament to the convergence of physical and digital worlds. By harnessing the power of computer vision, it can interpret and react to real-world actions, enabling a dynamic and interactive interface. Motion tracking further enhances its capabilities, allowing users to engage with content in ways previously unimaginable. Through the lens of augmented reality, information is overlaid onto the physical environment, creating an immersive and enriched learning space.In the educational landscape, the impact of the augmented whiteboard is profound. Real-time collaboration takes center stage, as students and educators alike can engage with content simultaneously. This not only fosters a more interactive learning environment but also promotes collaborative problem-solving and creativity. The traditional limitations of physical classrooms are transcended, opening up new possibilities for distance learning and global.

Future Scope of this Project

1. **Enhanced Interactivity**: Gesture recognition and voice commands could make interaction with the software more intuitive and efficient, reducing the need for traditional input methods like mouse and keyboard.

2. **Advanced Content Manipulation**: This feature would allow users to easily edit, resize, and animate elements within the software, empowering creativity and streamlining workflow.

3. **Integration with AI**: Automated content organization and predictive suggestions could save users time by anticipating their needs and helping them find relevant content more quickly.

4. **AR Collaboration**: Virtual collaboration in immersive environments could revolutionize teamwork by allowing users to collaborate in virtual spaces regardless of physical location.

5. **Cross-Platform Compatibility**: Access from various devices and platforms would ensure that users can work seamlessly across different devices and operating systems.

6. **Education Solutions**: Tailored features for educators and learners could include interactive lesson plans, virtual classrooms, and tools for assessing student progress.

7. **IoT Integration**: Interaction with smart devices and sensors could enable new possibilities for automation and data collection within the software.

8. **Data Visualization**: Advanced tools for analyzing complex data could help users gain insights from large datasets more easily and effectively.

9. **Accessibility Features**: Support for users with disabilities is essential for ensuring that the software is inclusive and usable by all individuals, regardless of their abilities.

References

[1] R. Rusli, D. A. Nalanda, A. D. V. Tarmidi, K. M. Suryaningrum, and R. Yunanda, "Augmented reality for studying hands on the human body for elementary school students," *Procedia Computer Science*, vol. 216, pp. 237–244, 2023.

[2] N. I. M. Enzai, N. Ahmad, M. A. H. A. Ghani, S. S. Rais, and S. Mohamed, "Development of augmented reality (ar) for innovative teaching and learning in engineering education," *Asian Journal of University Education*, vol. 16, no. 4, pp. 99–108, 2021.

[3] J. Bacca, S. Baldiris, R. Fabregat, S. Graf *et al.*, "Mobile augmented reality in vocational education and training," *Procedia Computer Science*, vol. 75, pp. 49–58, 2015.

[4] A. Iftene and D. Trandabăț, "Enhancing the attractiveness of learning through augmented reality," *Procedia Computer Science*, vol. 126, pp. 166–175, 2018.

[5] H. Shima and M. Soga, "Development of an interactive training support system for drawing and painting skill on a 3d object with ar and haptic interface," *Procedia Computer Science*, vol. 159, pp. 2015–2023, 2019.

[6] H. Kaufmann and D. Schmalstieg, "Mathematics and geometry education with collaborative augmented reality," in *ACM SIGGRAPH 2002 conference abstracts and applications*, 2002, pp. 37–41.

[7] M. H. M. Jamrus and A. B. Razali, "Augmented reality in teaching and learning english reading: realities, possibilities, and limitations," *International Journal of Academic Research in Progressive Education and Development*, vol. 8, no. 4, pp. 724–737, 2019.

[8] S. Andolina, D. Lee, and S. Dow, "Crowdboard: an augmented whiteboard to support large-scale co-design," in *Adjunct proceedings of the 26th annual ACM symposium on User interface software and technology*, 2013, pp. 89–90.

Appendix I

Certificate of Publication

International Journal of Computer Vision*

Date:	23 Apr 2024
To:	"Muhammed Shehin Salim" mhdshehinms@gmail.com
From:	"International Journal of Computer Vision (VISI)" yasotha.sujeen@springer.com
Subject:	VISI-D-24-00660 - Submission confirmation

Dear Mr Salim,

Thank you for submitting your manuscript, AUGMENTED WHITEBOARD USING COMPUTER VISION, to International Journal of Computer Vision.

The submission id is: VISI-D-24-00660
Please refer to this number in any future correspondence.

During the review process, you can keep track of the status of your manuscript.

Your username is: ********
If you forgot your password, you can click the 'Send Login Details' link on the EM Login page at https://www.editorialmanager.com/visi/

Should you require any further assistance please feel free to contact the Editorial Office by clicking on the "contact us" in the menu bar to send an email to us.

With kind regards,
Springer Journals Editorial Office
International Journal of Computer Vision

Figure I.1: Submission of Manuscript

*Subject to acceptance

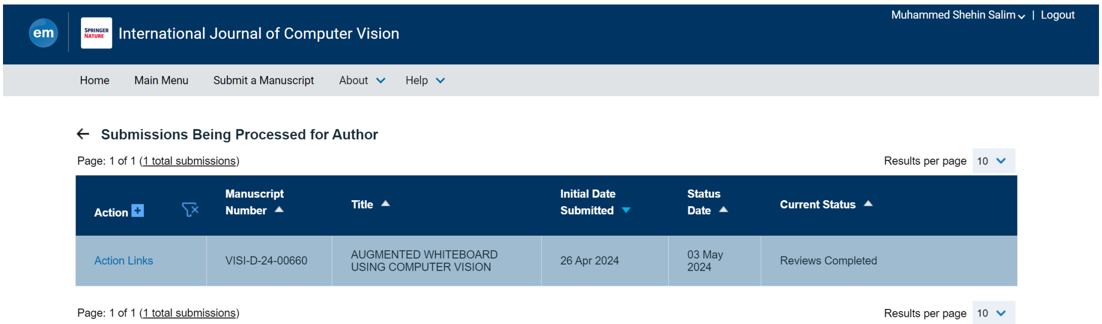


Figure I.2: Current Status

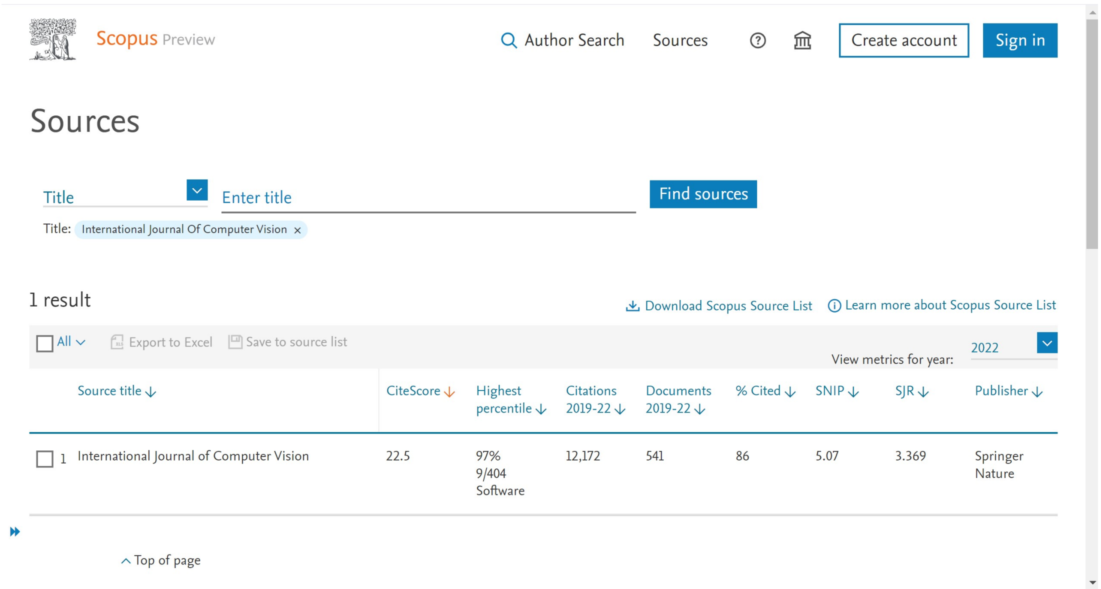


Figure I.3: Scopus Index

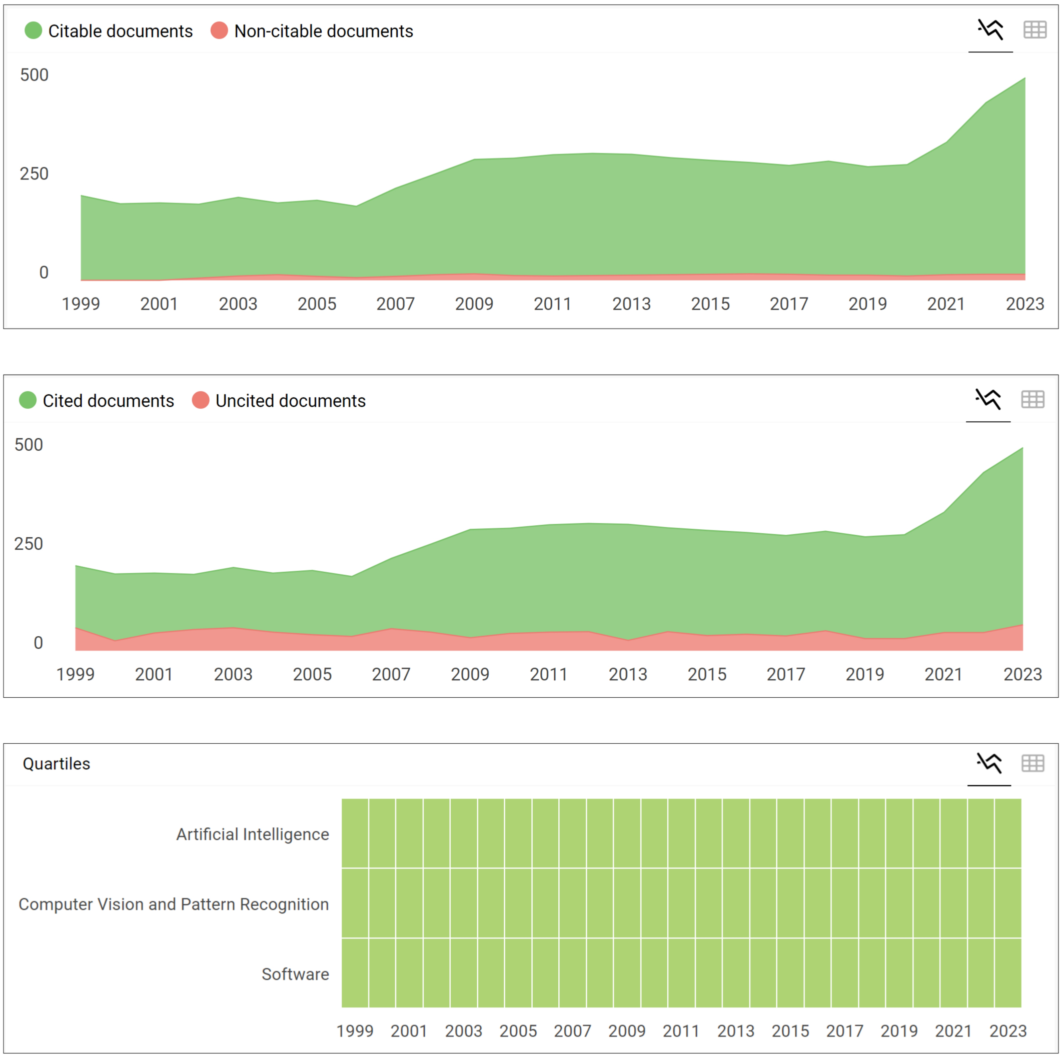
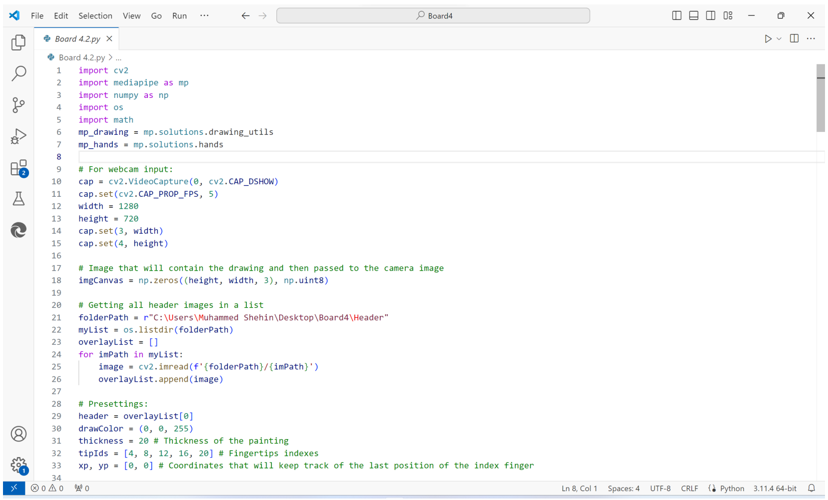


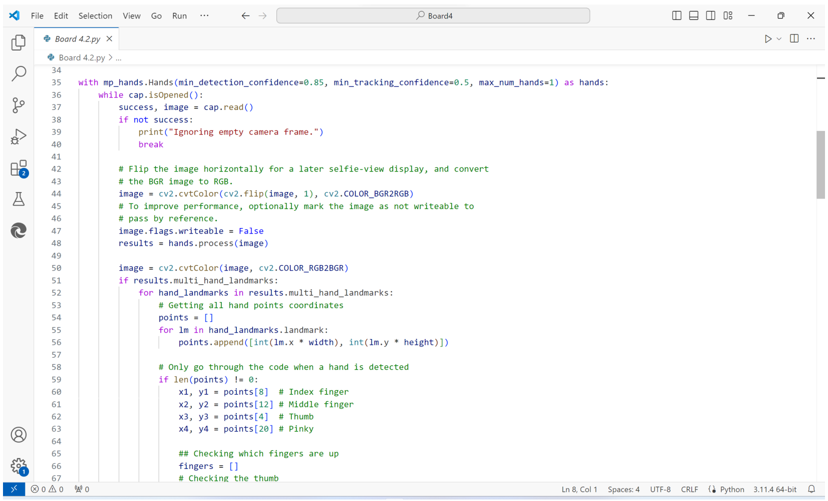
Figure I.4: Scopus Indexed Quartile 1

Appendix II

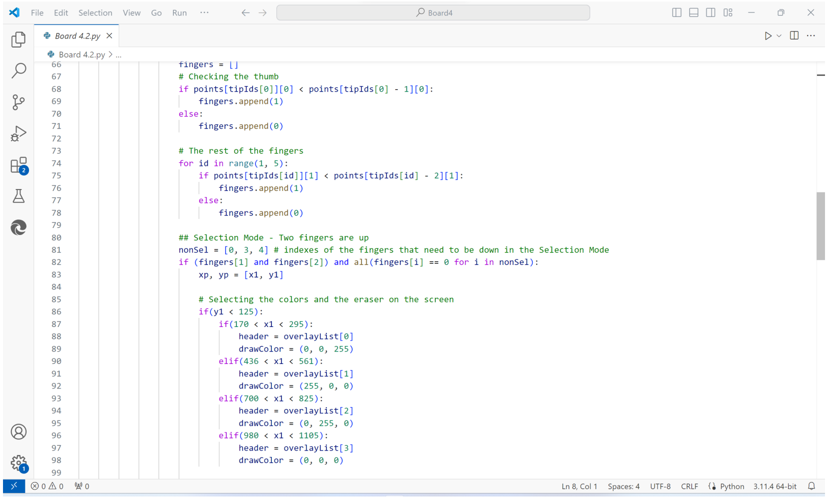
Source Code Snippets



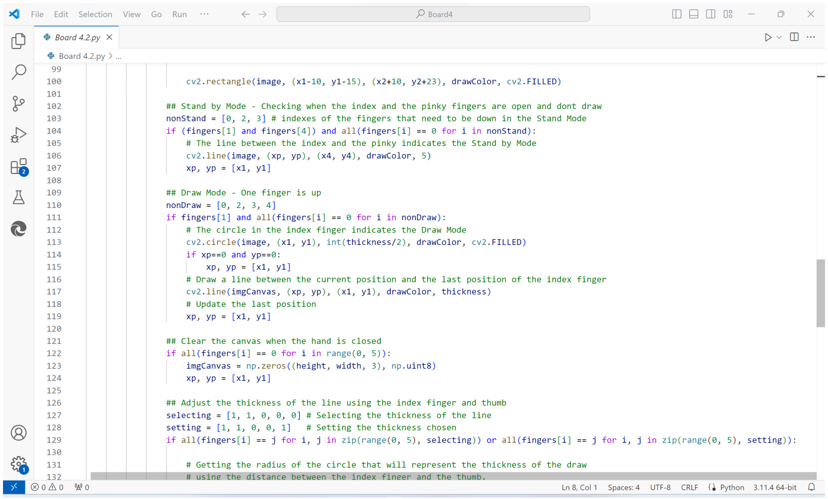
```python
import cv2
import mediapipe as mp
import numpy as np
import os
import math
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands

# For webcam input:
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
cap.set(cv2.CAP_PROP_FPS, 5)
width = 1280
height = 720
cap.set(3, width)
cap.set(4, height)

# Image that will contain the drawing and then passed to the camera image
imgCanvas = np.zeros((height, width, 3), np.uint8)

# Getting all header images in a list
folderPath = r"C:\Users\Muhammed Shehin\Desktop\Board4\Header"
myList = os.listdir(folderPath)
overlayList = []
for imPath in myList:
    image = cv2.imread(f'{folderPath}/{imPath}')
    overlayList.append(image)

# Presettings:
header = overlayList[0]
drawColor = (0, 0, 255)
thickness = 20 # Thickness of the painting
tipIds = [4, 8, 12, 16, 20] # Fingertips indexes
xp, yp = [0, 0] # Coordinates that will keep track of the last position of the index finger

```



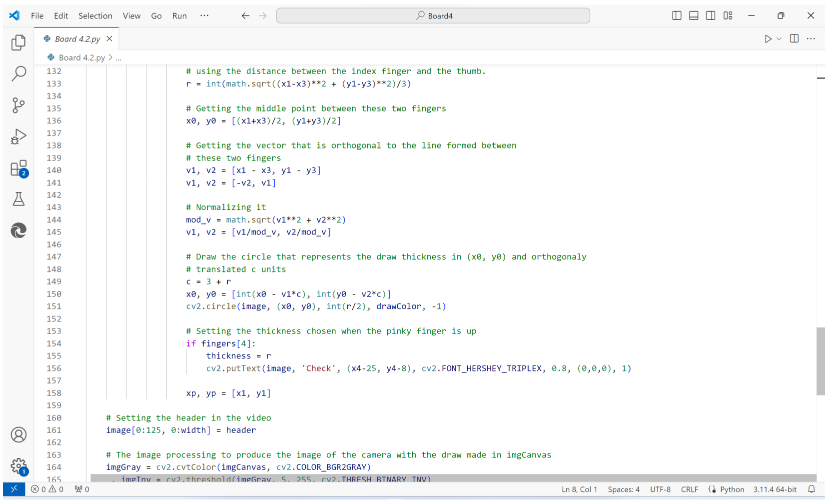
```python

with mp_hands.Hands(min_detection_confidence=0.85, min_tracking_confidence=0.5, max_num_hands=1) as hands:
    while cap.isOpened():
        success, image = cap.read()
        if not success:
            print("Ignoring empty camera frame.")
            break

        # Flip the image horizontally for a later selfie-view display, and convert
        # the BGR image to RGB.
        image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR_BGR2RGB)
        # To improve performance, optionally mark the image as not writeable to
        # pass by reference.
        image.flags.writeable = False
        results = hands.process(image)

        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
        if results.multi_hand_landmarks:
            for hand_landmarks in results.multi_hand_landmarks:
                # Getting all hand points coordinates
                points = []
                for lm in hand_landmarks.landmark:
                    points.append([int(lm.x * width), int(lm.y * height)])

                # Only go through the code when a hand is detected
                if len(points) != 0:
                    x1, y1 = points[8]  # Index finger
                    x2, y2 = points[12] # Middle finger
                    x3, y3 = points[4]  # Thumb
                    x4, y4 = points[20] # Pinky

                    ## Checking which fingers are up
                    fingers = []
                    # Checking the thumb
```



```python
fingers = []
# Checking the thumb
if points[tipIds[0]][0] < points[tipIds[0] - 1][0]:
    fingers.append(1)
else:
    fingers.append(0)

# The rest of the fingers
for id in range(1, 5):
    if points[tipIds[id]][1] < points[tipIds[id] - 2][1]:
        fingers.append(1)
    else:
        fingers.append(0)

## Selection Mode - Two fingers are up
nonSel = [0, 3, 4] # indexes of the fingers that need to be down in the Selection Mode
if (fingers[1] and fingers[2]) and all(fingers[i] == 0 for i in nonSel):
    xp, yp = [x1, y1]

    # Selecting the colors and the eraser on the screen
    if(y1 < 125):
        if(170 < x1 < 295):
            header = overlayList[0]
            drawColor = (0, 0, 255)
        elif(436 < x1 < 561):
            header = overlayList[1]
            drawColor = (255, 0, 0)
        elif(700 < x1 < 825):
            header = overlayList[2]
            drawColor = (0, 255, 0)
        elif(980 < x1 < 1105):
            header = overlayList[3]
            drawColor = (0, 0, 0)
```



```python
    cv2.rectangle(image, (x1-10, y1-15), (x2+10, y2+23), drawColor, cv2.FILLED)

## Stand by Mode - Checking when the index and the pinky fingers are open and dont draw
nonStand = [0, 2, 3] # indexes of the fingers that need to be down in the Stand Mode
if (fingers[1] and fingers[4]) and all(fingers[i] == 0 for i in nonStand):
    # The line between the index and the pinky indicates the Stand by Mode
    cv2.line(image, (xp, yp), (x4, y4), drawColor, 5)
    xp, yp = [x1, y1]

## Draw Mode - One finger is up
nonDraw = [0, 2, 3, 4]
if fingers[1] and all(fingers[i] == 0 for i in nonDraw):
    # The circle in the index finger indicates the Draw Mode
    cv2.circle(image, (x1, y1), int(thickness/2), drawColor, cv2.FILLED)
    if xp==0 and yp==0:
        xp, yp = [x1, y1]
    # Draw a line between the current position and the last position of the index finger
    cv2.line(imgCanvas, (xp, yp), (x1, y1), drawColor, thickness)
    # Update the last position
    xp, yp = [x1, y1]

## Clear the canvas when the hand is closed
if all(fingers[i] == 0 for i in range(0, 5)):
    imgCanvas = np.zeros((height, width, 3), np.uint8)
    xp, yp = [x1, y1]

## Adjust the thickness of the line using the index finger and thumb
selecting = [1, 1, 0, 0, 0] # Selecting the thickness of the line
setting = [1, 1, 0, 0, 1]   # Setting the thickness chosen
if all(fingers[i] == j for i, j in zip(range(0, 5), selecting)) or all(fingers[i] == j for i, j in zip(range(0, 5), setting)):

    # Getting the radius of the circle that will represent the thickness of the draw
    # using the distance between the index finger and the thumb.
```



```python
        # using the distance between the index finger and the thumb.
        r = int(math.sqrt((x1-x3)**2 + (y1-y3)**2)/3)

        # Getting the middle point between these two fingers
        x0, y0 = [(x1+x3)/2, (y1+y3)/2]

        # Getting the vector that is orthogonal to the line formed between
        # these two fingers
        v1, v2 = [x1 - x3, y1 - y3]
        v1, v2 = [-v2, v1]

        # Normalizing it
        mod_v = math.sqrt(v1**2 + v2**2)
        v1, v2 = [v1/mod_v, v2/mod_v]

        # Draw the circle that represents the draw thickness in (x0, y0) and orthogonaly
        # translated c units
        c = 3 + r
        x0, y0 = [int(x0 - v1*c), int(y0 - v2*c)]
        cv2.circle(image, (x0, y0), int(r/2), drawColor, -1)

        # Setting the thickness chosen when the pinky finger is up
        if fingers[4]:
            thickness = r
            cv2.putText(image, 'Check', (x4-25, y4-8), cv2.FONT_HERSHEY_TRIPLEX, 0.8, (0,0,0), 1)

        xp, yp = [x1, y1]

# Setting the header in the video
image[0:125, 0:width] = header

# The image processing to produce the image of the camera with the draw made in imgCanvas
imgGray = cv2.cvtColor(imgCanvas, cv2.COLOR_BGR2GRAY)
imgInv = cv2.threshold(imgGray, 5, 255, cv2.THRESH_BINARY_INV)
```

```python
                        v1, v2 = [v1/mod_v, v2/mod_v]

                        # Draw the circle that represents the draw thickness in (x0, y0) and orthogonaly
                        # translated c units
                        c = 3 + r
                        x0, y0 = [int(x0 - v1*c), int(y0 - v2*c)]
                        cv2.circle(image, (x0, y0), int(r/2), drawColor, -1)

                        # Setting the thickness chosen when the pinky finger is up
                        if fingers[4]:
                            thickness = r
                            cv2.putText(image, 'Check', (x4-25, y4-8), cv2.FONT_HERSHEY_TRIPLEX, 0.8, (0,0,0), 1)

                        xp, yp = [x1, y1]

        # Setting the header in the video
        image[0:125, 0:width] = header

        # The image processing to produce the image of the camera with the draw made in imgCanvas
        imgGray = cv2.cvtColor(imgCanvas, cv2.COLOR_BGR2GRAY)
        _, imgInv = cv2.threshold(imgGray, 5, 255, cv2.THRESH_BINARY_INV)
        imgInv = cv2.cvtColor(imgInv, cv2.COLOR_GRAY2BGR)
        img = cv2.bitwise_and(image, imgInv)
        img = cv2.bitwise_or(img, imgCanvas)

        cv2.imshow('MediaPipe Hands', img)
        if cv2.waitKey(3) & 0xFF == ord('q'):
            break
.release()
.destroyAllWindows()
```