Final Year Project Report

on

“Real time object detection using ML”

Submitted to the

Savitribai Phule Pune University

In partial fulfillment for the award of the Degree of

Bachelor of Engineering

in

Information Technology

by

Shruti Bhabad [190138513]
Kshitij Bhalerao [190138516]
Prerna Nagare [190138550]
Divya Shinde [190138560]

Under the guidance of

Prof. Pragati V. Pandit

Department of Information Technology

K.K. Wagh Institute of Engineering Education and Research, Nashik -3

2023 - 2024

Department of Information Technology

K.K. Wagh Institute of Engineering Education and Research, Nashik-3

2023-2024

CERTIFICATE

This is to certify that the project report entitled **"Real Time Object Detection using ML"** being submitted by **Shruti Bhabad** [190138513], **Kshitij Bhalerao** [190138516], **Prerna Nagare** [190138550], **Divya Shinde** [190138560] is a record of bonafide work carried out by him/her under the supervision and guidance of **Prof. Pragati V. Pandit** in partial fulfilment of the requirement for **BE (Information Technology Engineering) 2019 course** of Savitribai Phule Pune University, Pune in the academic year 2023-2024.

Date:
Place: Nashik

Prof. Pragati V. Pandit	Dr. Preeti D. Bhamre
Guide	Head of the Department

This project report has been examined by us as per the Savitribai Phule Pune University, Pune, requirements at K. K Wagh Institute of Engineering Education and Research, Nashik on ....

(Name & Signature)	(Name & Signature)
Internal Examiner	External Examiner

ACKNOWLEDGEMENT

I would like to thank to Head of my Department Prof. Dr. Preeti D. Bhamre for her valuable and constructive suggestions during the development of this project. Then I would like to thank our Project guide Prof. Pragati P. Pandit for her valuable guidance which has been the one that helped me patch this project and make it successful her suggestions and instruction has served as the major contributor towards the completion of the project. Then I would like to thank our teachers who helped me with many difficulties to make this project successful. Then I would like to thank my friend who has helped us with their valuable suggestion and guidance has been helpful in various phases of the completion of the project.

(Shruti Bhabad)
(Kshitij Bhalerao)
(Prerna Nagare)
(Divya Shinde)

ABSTRACT

The Dynamic Object Tracking System presented in this project revolutionizes real-time object tracking through the fusion of Machine Learning (ML), computer vision, and projection technology. The system addresses the escalating need for accurate tracking in diverse scenarios. It seamlessly integrates a projection module, camera network, and advanced object detection algorithms to monitor both tangible and projected objects. By leveraging ML and computer vision, the system captures live video feeds from strategically positioned cameras. Employing state-of-the-art object detection algorithms such as YOLO or Faster R-CNN, it precisely identifies and localizes objects within the environment. Some critical challenges, including data synchronization and real-time processing, are tactfully managed through optimized strategies. The applications are far-reaching, encompassing interactive training platforms, immersive entertainment experiences, surveillance, and human-computer interaction research. This system's convergence of ML, computer vision, and projection engenders a powerful and versatile platform for real-time object tracking, offering unparalleled accuracy and innovation across an array of domains.

Keywords: *Dynamic Object Tracking, Machine Learning(ML), Real-time Tracking, Projection Module, Object Detection Algorithms*

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Real Time Object Detection using ML

1.1 Introduction

Step into a realm where reality and virtual play collide with the Interactive Projection Ball Game! An immersive blend of physical throws and digital enchantment, this experience breaks all the conventional gaming rules. Picture tossing a ball and witnessing it dance with virtual counterparts on the screen— that's the magic this project is conjuring.

The mission is clear: unleash the joy. Throw a ball, let the tech do its dance, and rack up points as your real-world tosses spark a digital adventure. It's not about complexity; it's about pure, unadulterated fun. Join the journey where every throw becomes a brushstroke in the canvas of the Interactive Projection Ball Game's exhilarating escapade.

1.2 Aim / Motivation

The project aims to:

- To develop a cutting-edge Dynamic Object Detection System that revolutionizes real-time object detection by integrating Machine Learning (ML), computer vision, and protuberance technology. This system seeks to address the increasing demand for accurate object tracking across various scenarios.

- To seamlessly integrate a protuberance module, camera network, and advanced object detection algorithms to detect both physical and projected objects. By leveraging ML and computer vision, the system aims to capture live video feeds from strategically placed cameras and employ state-of-the-art object detection techniques to precisely identify and localize objects in real-time. Critical challenges such as data synchronization and real-time processing will be addressed through optimized strategies.

- To have far-reaching applications, including interactive training platforms, immersive entertainment experiences, surveillance, and industrial automation. Ultimately, the goal is to create a versatile platform for real-time object detection that offers unparalleled accuracy and innovation across a wide range of fields.

1.3 Scope

- Dynamic Object Detection System: Create a system that adeptly tracks moving physical objects, enhancing user interaction by precisely following the trajectory of thrown balls, adding dynamism to the Interactive Projection Ball Game.

- Virtual Object Integration: Integrate seamlessly projected virtual objects with physical ones, forming a cohesive and immersive environment where the tangible and virtual realms coalesce, enriching user experience.

- Object Detection Algorithms: Implement advanced algorithms for precise identification of physical objects in real-time, ensuring accuracy and responsiveness in recognizing the user's interactions within the Interactive Projection Ball Game.
- Multi-camera Synchronization: Achieve synchronized operation across multiple cameras, enhancing the system's spatial awareness and enabling a comprehensive view of user interactions, optimizing the overall immersive experience.
- Real-time Processing: Employ real-time processing capabilities to swiftly interpret and respond to user actions, minimizing latency and ensuring instantaneous feedback, thus maintaining the fluidity of the Interactive Projection Ball Game.

1.4 Objective

The project sets out to achieve several key objectives:

- Integrate Computer Vision Algorithms: Implement and refine computer vision algorithms for dynamic object tracking.
- Incorporate Projection Technology: Explore the integration of projection technology, possibly inspired by the studies of Schoning, Kruger, Rohs, and Lochtefeld, to enhance the interactive space for users and provide innovative ways of interacting with physical objects.
- Collaborative and Interactive Gameplay: Emphasize collaborative gameplay and interaction with physical objects.
- User Experience Enhancement: Prioritize the enhancement of user experiences and interactive interfaces, showcasing the potential for cost-effective and immersive interactions in diverse contexts.
- Tangible Interface Integration: Merge the spectacle of modern video games with tangible interfaces, offering a novel and immersive gaming experience.

- Accessibility: Emphasize the use of accessible technology to create engaging and interactive entertainment applications.

1.5 Report Organization

The report is structured into eight chapters:

Chapter 1: Introduction to Real Time Object Tracking using ML and covering the projects Motivation, contribution, aim, scope, objectives, contribution.

Chapter 2: The Literature Review, explores related work and back-ground in the field.

Chapter 3: Methodology

Chapter 4: Project Requirement Specifications delves into hardware and software requirements, functional and non-functional specifications.

Chapter 5: High-Level Design of Project,includes UML diagrams and system design, while.

Chapter 6: Experimental Setup and Simulation

Chapter 7: Results and Evaluation

Chapter 8: Project planning for Sem I and Sem II

Chapter 9: Conclusion summarizing the project acheivements and key takeaways

Chapter 2

Literature Review

A literature review is a comprehensive analysis of existing research and publications related to a specific topic. In the context of this project, it involves summarizing and evaluating prior studies, findings, and technologies in areas such as object detection, projection mapping, multi-camera systems, human-computer interaction, and real-time processing.

2.1 Related Work Done

The integration of Machine Learning (ML), Computer Vision, and Projection Technology in real-time object-tracking systems has been a topic of significant research and development in recent years. Several studies have explored various aspects of similar projects, paving the way for innovative applications and advancements. In this literature review, we highlight key findings from related works that have contributed to the development of our Dynamic Object Tracking System.

William T. Freeman, David B. Anderson, Paul A. Beardsley,Chris N. Dodge had studied Computer Vision for Interactive Computer Graphics. In this research, various vision algorithms were developed and applied for interactive graphics applications. The study focused on fundamental visual measurements, including large object tracking, shape recognition, motion analysis, and small object tracking. These measurements were utilized to create vision-based interfaces for computer games and interactive devices like toy robots and televisions. The research team also designed a specialized image detector/processor, reducing implementation costs. This work demonstrates the practical applications of computer vision in enhancing user experiences and interactive interfaces, showcasing the potential for cost-effective and immersive interactions in diverse contexts.

Johannes Schoning, Antonio Kruger, Michael Rohs, Markus Lochtefeld had studied on LittleProjectedPlanet: An Augmented Reality Game for Camera Projector Phones. This project explores the integration of miniaturized projection technology, specifically pico projectors, into mobile devices. This integration enables these devices to project large-scale information onto real-world surfaces, expanding the interaction space of mobile devices to physical objects in the environment. This advancement opens up possibilities for innovative interaction concepts that are not achievable on traditional desktop computers. The paper discusses the potential applications of camera projector phones through a mobile adaptation of the Playstation 3 game LittleBigPlanet. The camera projector unit augments users' hand drawings by overlaying virtual objects and simulating their physical interactions with the real world. Players can create a 2D world on paper or use existing physical objects, allowing the physics engine to simulate interactions and achieve game goals. This research showcases the transformative potential of integrating projection technology into mobile devices, paving the way for novel interactive experiences.

Richard Colvin, Ted Hung, David Jimison, Benjamin Johnson, Eben Myers, Tina Blaine had studied A Dice Game in Third-Person Augmented Reality. This project explores a prototype entertainment application utilizing the Augmented-Reality Toolkit. In this innovative system, players engage in a fantasy dice game where the dice, marked with glyphs, are interpreted by a computer. The system provides graphical and auditory feedback using consumer-grade equipment: a USB webcam, a projector, and a standard desktop computer with surround speakers. Notably, unlike many Augmented-Reality Toolkit applications, players do not wear head-mounted displays. Instead, face-to-face gameplay is integrated with the physicality of a traditional dice game, with results displayed on a shared projection screen from a third-person perspective. This unique approach merges the spectacle of modern video games with a tangible interface, offering a novel and immersive gaming experience. The study showcases the potential of augmented reality in enhancing traditional gaming forms, emphasizing the use of accessible technology to create engaging and interactive entertainment applications.

Alessandro Dal Corso,Gudmundur Einarsson, H. M. Kjer,M. Olsen had studied on VirtualTable: a projection augmented reality game. In the VirtualTable project, an interactive tower defense game is created using projection augmented reality. Players engage in the game by preventing virtual stylized soot balls from reaching the cheese on a table. Any object placed on the table becomes part of the game, serving as a wall, obstacle, or tower, depending on its type. The game encourages strategic thinking and collaboration, rather than relying solely on the quantity of objects. This continuous installation allows any number of players to join for an optional period, fostering an interactive and engaging experience for participants. The project's focus on collaborative gameplay and the integration of physical objects enhances the immersive nature of the game, making it an attractive and innovative attraction for users.

2.2 Existing System

The existing system landscape in the realm of real-time object tracking and interactive experiences encompasses a wealth of research and applications. Notable contributions include object detection, projection mapping, multi-camera systems, and human-computer interaction studies. One such example is the VirtualTable project, which showcases the fusion of augmented reality and projection mapping to create engaging interactive gaming experiences. In this context, users are immersed

in a tower defence game where virtual stylized soot balls must be prevented from reaching a target, utilizing physical objects as barriers and towers. Additionally, studies in multi-camera systems have emphasized the importance of synchronized data from multiple sources. These systems ensure coherent processing of information, especially in dynamic environments where objects are in motion. Human-computer interaction studies have contributed valuable insights into user behaviour and interface design. These findings emphasize the significance of user-friendly interactions, guiding the development of intuitive interfaces for interactive systems.
The existing systems underscore the significance of collaborative gameplay, encouraging strategic thinking and cooperation among players rather than relying solely on the quantity of objects. This approach does not provide dynamic object tracking to a wide range of practical applications, showcasing its versatility and adaptability. However, the proposed system will provide integration of various technologies and offer an innovative solution that goes beyond individual components. It represents a holistic approach to dynamic object tracking, leveraging machine learning, computer vision, and projection technology for a seamless and immersive user experience.

Summary

This chapter provides a literature review which is the comprehensive survey of existing scholarly work related to the research topic. It serves as a foundation for the subsequent chapters and contributes to the overall understanding of the research area. In the next chapter, Proposed Architecture upon the insights gained from the literature review are discussed in detail.

Chapter 3

System Architecture

In the previous chapter, problem definition, scope and objective of this project and report organization have been discussed. Also,came across with the related work to this project. In this chapter, Methodology for Implementation of this Project and Project Architecture, detailed Architecture Diagram along with the methodological steps have been discussed.

3.1 Problem Statement / Definition

In the realm of entertainment and education, traditional tools often encounter difficulty in fully engaging modern audiences accustomed to dynamic, interactive experiences. Recognizing this challenge, there arose an opportunity to bridge the gap between conventional methods and contemporary preferences. This led to the inception of an innovative solution, a groundbreaking interactive Augmented Reality (AR) experience that seamlessly integrates virtual elements with tangible, physical interactions.

Through the integration of ML technology, this initiative revolutionizes the way audiences perceive and engage with content. By superimposing computer-generated sensory inputs like sound, video, and graphics onto the real-world environment, the experience transcends the limitations of traditional mediums. It enables users to interact actively with the amalgamation of virtual and physical elements, fostering a deeper level of immersion and captivation.

This interactive AR experience is designed to be both entertaining and educational, catering to diverse audiences across various sectors. Whether applied in educational settings to enhance learning experiences or employed in entertainment for immersive storytelling, its adaptability and versatility promise an unparalleled level of engagement. Through the fusion of virtual and real-world interactions, this innovative approach redefines entertainment and educational paradigms, ushering in a new era of captivating, interactive experiences for modern audiences.

3.2 Methodology

3.2.1 Data Collection and Preparation:

- Collect a diverse dataset of physical objects representing typical usage scenarios.
- Pre-process and augment the dataset to suit training and testing machine learning models.

3.2.2 Machine Learning Model Development:

- Implement and fine-tune advanced ML algorithms for object detection and tracking, using transfer learning to adapt pre-trained models to specific tracking requirements.

3.2.3 Projection Technology Integration:

- Develop seamless integration of virtual objects onto physical surfaces, ensuring accurate alignment through computer vision and projection technologies.

3.2.4 Real-time Processing Optimization:

- Optimize algorithms for efficiency and parallel processing to achieve real-time performance.
- Consider hardware acceleration options like GPUs for enhanced speed.

3.2.5 User Interaction Design:

- Design an intuitive user interface for controlling and interacting with tracked objects, refining it iteratively based on user testing and feedback.

3.2.6 Testing and Validation:

- Rigorously test the system in various real-world environments, collecting data on accuracy, reliability, and real-time performance in diverse scenarios.

3.2.7 Potential Expansion Analysis:

- Explore avenues for future expansion, such as integrating more advanced ML models or additional sensors for enhanced object tracking, considering scalability and adaptability for future developments.

3.3 Block Diagram

The integration of virtual objects into the real world has gained significant interest in various applications, ranging from augmented reality to virtual product placement. To achieve seamless integration, a robust methodology is essential. This paper presents a stepwise approach to address this challenge.

- The first step i.e. 'Data Collection and Preparation', involves gathering a wide range of physical objects that the system will encounter in real-world scenarios. Diversity ensures the model's ability to generalize across various objects and environments. But before the show starts, some preparation is

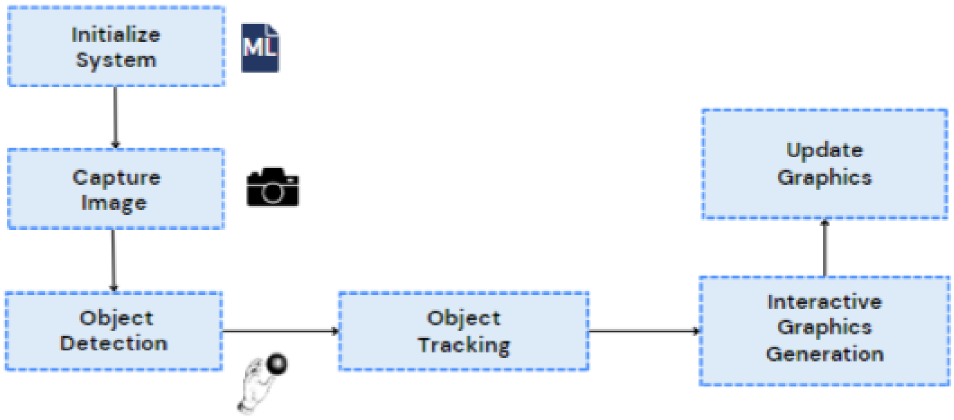

Figure 3.1: Block Diagram

necessary. Before feeding the data into machine learning models, it needs to be pre-processed, which may involve tasks like resizing images, normalizing colors, and removing noise.

- The second step involves 'Machine Learning Model Development', where advanced algorithms like object detection become the detectives, learning to identify and locate specific objects within the scene. In image segmentation, the artist carefully separates the object from its background, allowing clear virtual interactions. Pose estimation, the choreographer, determines the object's orientation and position in space, ensuring smooth virtual placement. This step focuses on training and fine-tuning algorithms like object detection, image segmentation, and pose estimation. Transfer learning is used to leverage pre-trained models and accelerate the development process.

- The third step involves 'Projection Technology Integration', this step involves seamlessly blending virtual objects with the real world using techniques like computer vision and spatial mapping. Computer vision and projection technologies work hand-in-hand to ensure precise alignment, placing virtual objects exactly where they should be. Calibration and alignment are crucial to ensure accurate placement of virtual objects.

- The fourth step involves 'Real-time Processing Optimization', efficient real-time processing is achieved through algorithmic streamlining for maximum speed. Parallel processing is employed to allow multiple tasks to occur simultaneously, ensuring instantaneous reactions and minimizing delays. This step is crucial for creating a responsive and dynamic virtual environment.

- The fifth step i.e. 'User Interaction Design', this step designs the user interface, i.e. a control panel for manipulating the virtual world. An intuitive and user-friendly interface is essential for interacting with the system. This involves designing controls for manipulating virtual objects, considering ergonomics and user experience. User experience is the director's feedback, incorporating user testing and feedback to refine the interface and make it even more user-friendly.

- The sixth step involves 'Testing and Validation', the system is rigorously tested in various real-world scenarios to evaluate its accuracy, reliability, and performance under different conditions. User feedback is actively collected and analyzed to identify areas for improvement and refinement.

- The seventh step is 'Potential Expansion Analysis'. The project looks towards the future by exploring potential improvements and new features. This involves considering the integration of more advanced machine learning models, additional sensors, or expanding the system's capabilities to new use cases. This forward-looking step ensures the adaptability and scalability of the proposed methodology.

Summary

This chapter provides the proposed system which involves an innovative AR experience bridging traditional tools with interactive AR technology. The methodology includes ML-based object detection, integration of projection technologies, user interaction design, and real-time processing optimization. The block diagram outlines the system's architecture, capturing images, detecting and tracking objects, and generating interactive graphics based on real-world interactions for an engaging user experience. In the next chapter, software requirements and specifications, functional requirements are discussed in detail.

Chapter 4

Project Requirement Specifications

In the previous chapter the paper shows the system architecture of Real Time Object Detection System using ML.
In this chapter requirement specification of Real Time Object Detection System using ML has been discussed

4.1 Hardware/Software Requirements

4.1.1 Hardware Requirements:

- Cameras:
  High-resolution cameras with low-light capabilities for clear and accurate image capture. Multiple cameras are strategically placed to cover the entire tracking area for comprehensive object monitoring.

- Projection Equipment:
  High-quality projectors capable of projecting virtual objects with clarity and precision onto physical surfaces.

- Physical Ball:
  A high contour color ball is used for interacting with the virtually projected environment.

4.1.2 Software Requirements:

- Operating System:
  Windows, Linux, or macOS operating system based on the preference and compatibility of software tools.
- Programming Language:
  Python for implementing machine learning algorithms, computer vision tasks, and system integration.
- Machine Learning Libraries:
  PyGame or other machine learning frameworks for developing and deploying object detection models. OpenCV for image and video processing tasks and integration with machine learning models.

4.2 Functional/Non-Functional Requirements

4.2.1 Functional Requirements:

- Object Detection: The system must accurately detect and identify physical and virtual objects using machine learning algorithms
- Real-time Processing: Achieve real-time processing of video feeds for immediate object-tracking responses
- Projection Integration: Integrate virtual objects onto physical surfaces in a synchronized manner with detected physical objects
- Synchronization: Ensure synchronization between camera feeds and projection outputs for accurate spatial alignment

- User Interface: Design an intuitive user interface allowing users to control the system and interact with tracked objects.

4.2.2 Non-functional Requirements:

- Performance: The system should handle a minimum of 30 frames per second for real-time tracking
- Accuracy: Achieve at least 90% accuracy in object detection and tracking.
- Scalability: The system should be scalable to accommodate additional cameras and projection devices if required
- Usability: The user interface should be user-friendly, ensuring ease of interaction and control.
- Reliability: The system should be reliable, with minimal downtime and robust error-handling mechanisms.
- User Training: Specify user training requirements, including user guides and training materials.
- Testing and Quality Assurance: Specify testing criteria and quality assurance standards to ensure that the system meets performance and reliability expectations.

Summary

This chapter provides information about the software requirements specification, Functional Requirements, and Non-Functional Requirements. In the next chapter, the high-level design of the project,UML Diagrams, class diagram Activity, and Sequence Diagrams are discussed in detail.

Chapter 5

High-Level Design of Project

In the previous chapter, Requirement specification have been discussed.
In this chapter, Detailed design of project using various UML diagram is been discussed.
High-Level Design refers to the process of creating abstract representations of a system before the actual implementation begins. In the context of a project, it involves creating diagrams that provide an overview of the system's structure and functionality. These diagrams collectively provide a comprehensive view of the project at a high level, aiding in communication among stakeholders and serving as a blueprint for the subsequent stages of development.

5.1 UML Diagram

Unified Modeling Language (UML) diagrams serve as a visual language for understanding, designing, and communicating complex systems. Widely employed in software engineering, UML diagrams provide a standardized and systematic way to represent various aspects of a system's architecture and behavior. These diagrams serve as a bridge between technical and non-technical stakeholders, facilitating effective communication and comprehension throughout the software development lifecycle.

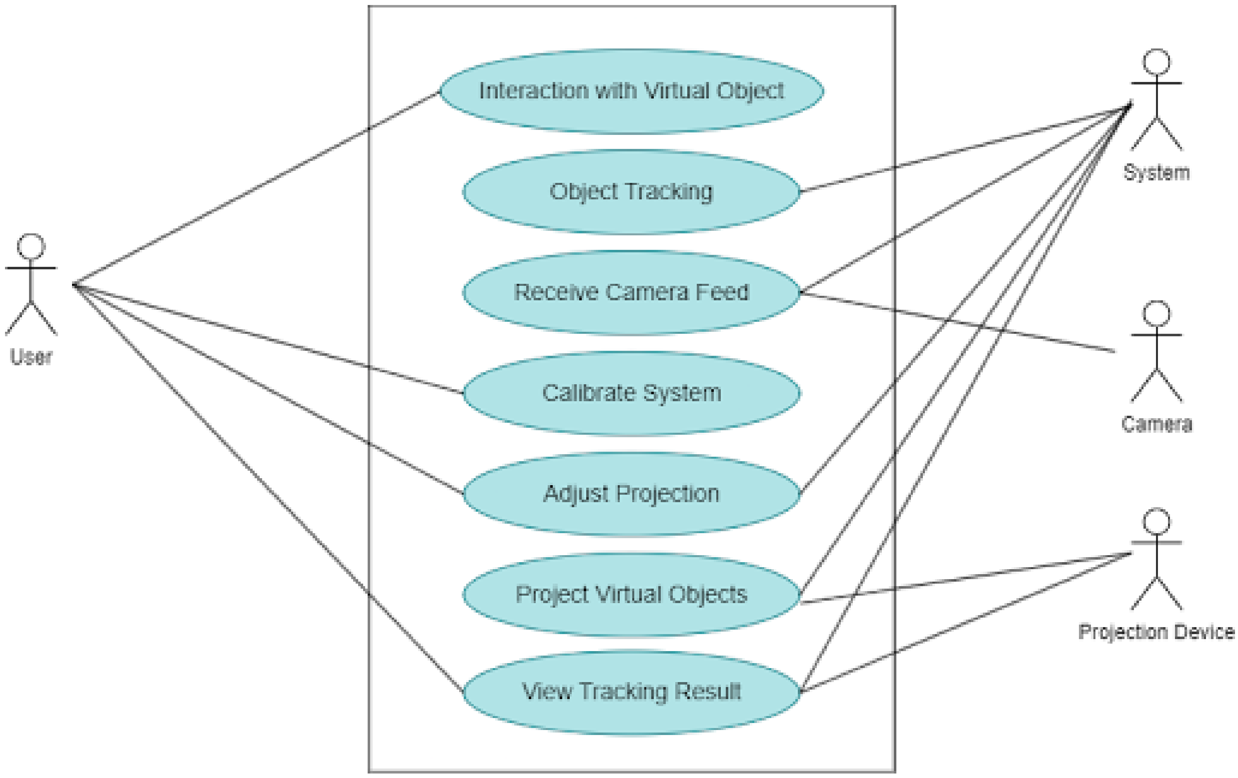

Figure 5.1: Use Case

5.1.1 Use Case Diagram

The Use Case Diagram serves as the project's high-level blueprint, offering a comprehensive view of the system's functionalities from an end-user perspective. It outlines the interactions between external actors and the system, defining the various use cases that propel our project's workflow. Fig. 5.1 serves as a visual guide, outlining how different actors interact with the system and the key functionalities they engage with. It highlights the user's control over the system, the continuous tracking of objects, and the dynamic projection of virtual elements onto physical surfaces. It visually encapsulates the dynamic interplay between actors and use cases in the context of the Dynamic Object Tracking System.

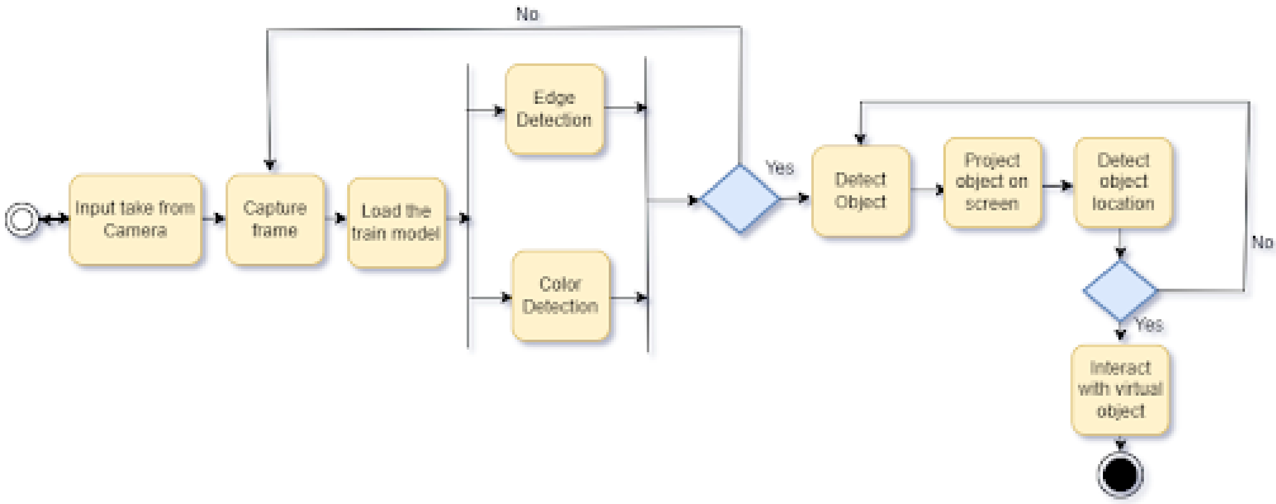

Figure 5.2: Activity Diagram

5.1.2 Activity Diagram

An activity diagram is a UML (Unified Modeling Language) diagram that visually represents the flow of activities within a system or process. It uses nodes and arrows to illustrate the sequence of tasks, decision points, and concurrent activities, aiding in understanding and designing complex workflows.The activity diagram begins with camera input, followed by frame capture. The trained model is loaded for edge and color detection, identifying objects. The system projects the detected objects on the screen, then locates and interacts with virtual objects, demonstrating the sequential flow of operations.

5.1.3 Class Diagram

A class diagram is a type of static structure diagram in the Unified Modeling Language (UML) that represents the structure and relationships of a system's classes and their interactions.This class diagram, represents an Object Detection and Interaction System that combines camera input, image processing, machine learning, and user interaction. The goal of this system is to detect physical objects, such as balls, thrown onto a projected screen and interact with these objects in real-time, scoring points when the objects collide with moving balloons on the screen. The Camera

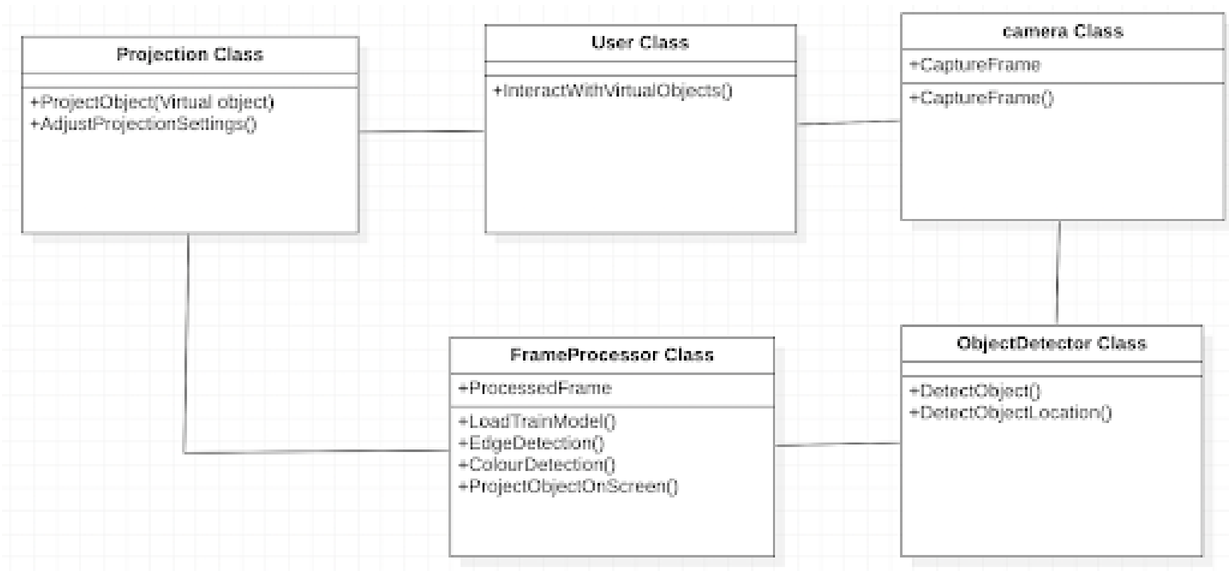

Figure 5.3: Class Diagram

Class represents the physical camera hardware and handles capturing frames. It is closely associated with the FrameProcessor Class, which aggregates the components responsible for image processing, object detection, and object projection.The Frame-Processor Class is the heart of the system. It processes captured frames by loading a trained model for object detection, performing edge and color detection, identifying objects, projecting them onto the screen, and detecting their locations.The ObjectDetector Class specializes in detecting objects, while the ObjectProjector Class focuses on projecting objects onto the screen. These classes work together within the FrameProcessor to provide real-time object interaction.The User Class represents the end user, responsible for interacting with the virtual objects projected on the screen. It utilizes the Camera, ObjectDetector, and ObjectProjector components to enable user interaction with the detected physical objects.

5.1.4 DFD Diagram

A Data Flow Diagram (DFD) is a visual representation of the flow of data within a system, illustrating how information is input, processed, stored, and output. It employs various symbols to represent processes, data stores, data flow, and external entities.

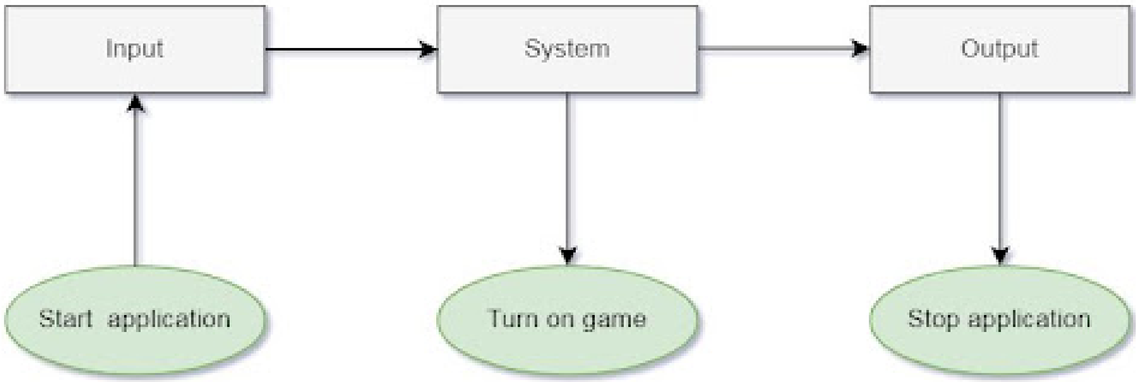

Figure 5.4: DFD-0

DFD-0 Diagram

A DFD-0, or Level 0 Data Flow Diagram, is the highest level of abstraction in the Data Flow Diagram hierarchy. It provides a broad overview of the entire system, focusing on major processes, data flows, data stores, and external entities without delving into the internal details of individual processes. The DFD-0 is often referred to as the context diagram.

5.1.5 ER-Diagram

An Entity-Relationship (ER) diagram is a visual representation of the entities, relationships, attributes, and constraints within a database. It is a widely used tool in database design to model and illustrate the logical structure of a database system. ER diagrams are particularly helpful for understanding how different entities in a system relate to each other and how data is organized.

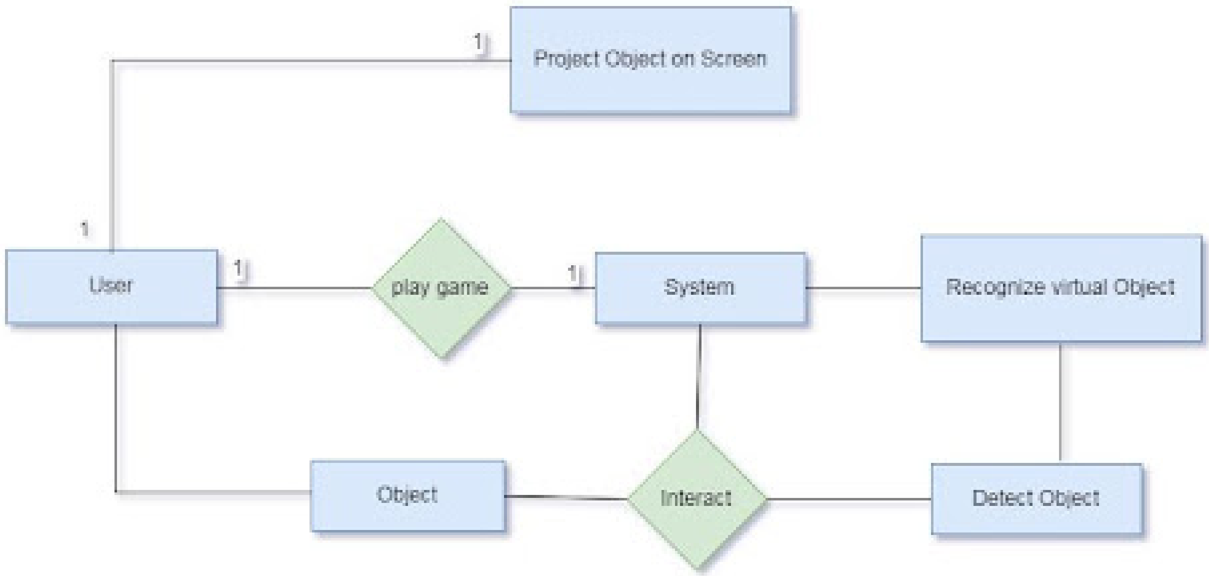

Figure 5.5: ER-Diagram

Summary

This chapter provides a detailed description of the High level design of the project, UML Diagrams, class diagram Activity, diagram Sequence Diagram.In next chapter, Experimental setup with performance parameters are discussed in detail.

Chapter 6

Experimental Setup / Simulation

In the prvious chapter, Detailed design of project is discussed.
In this chapter, information about experimental setup of project with the help of suitable diagram is discussed.

6.1 Diagram/Setup

Step 1: Setup

- Ensure that a camera is embedded in the laptop for capturing real-time images.
- Connect a projector to the laptop to project the application on the wall.

Step 2: Train the Model

- Collect a diverse dataset of images containing the objects you want to track.
- Preprocess the dataset to enhance image quality and remove noise.
- Train a machine learning model using a suitable algorithm (e.g., deep learning with a convolutional neural network) on the prepared dataset.
- Fine-tune the model to improve accuracy, considering factors such as lighting conditions and potential object orientations.
- Save the trained model and embed it in the laptop for real-time processing.

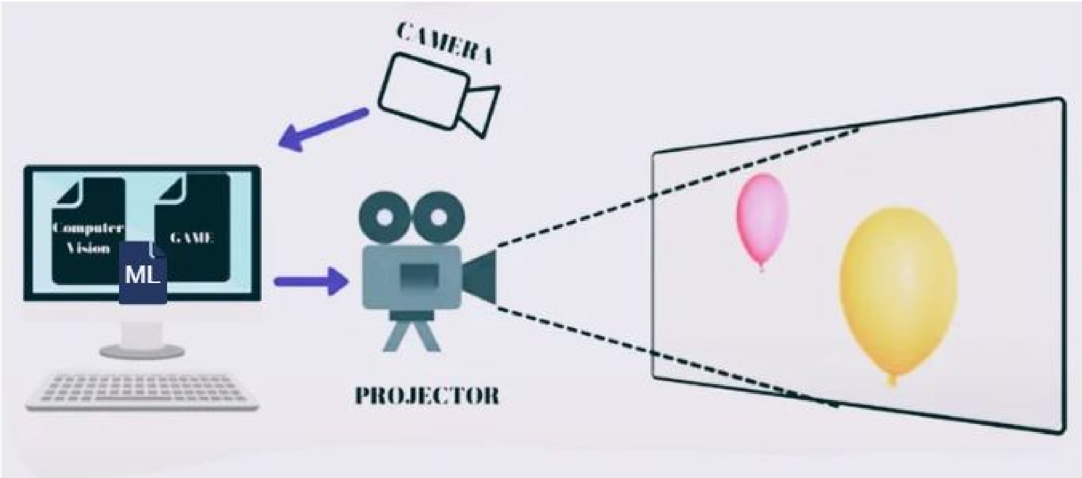

Figure 6.1: Setup Diagram

Step 3: Application Projection

- Develop or install an application that interacts with the projected content on the wall.
- Ensure the application is synchronized with the camera feed and capable of receiving real-time updates.
- Project the application onto the wall using the connected projector.

Step 4: Real-Time Object Tracking

- Initiate the camera to capture images at regular intervals (e.g., every second).
- Implement real-time image recognition using the trained model to identify objects in each captured frame.
- Process the identified objects to check if they interact with the application on the wall.

Step 5: Object-Application Interaction

- When the image recognition identifies an object interacting with the projected application, trigger a reaction from the application.
- Define specific interactions and reactions based on the type of object detected.

Step 6: Testing and Optimization

- Conduct extensive testing to ensure the system's accuracy and responsiveness.
- Fine-tune the model and application based on testing feedback, optimizing for various scenarios.
- Address any issues related to false positives/negatives, lighting conditions, or other environmental factors.

Step 7: Deployment

- Once satisfied with the performance, deploy the real-time object tracking system for practical use.
- Ensure the system is configured for continuous monitoring and can handle various real-world conditions.

Step 8: Maintenance and Updates

- Regularly monitor and maintain the system to address any issues that may arise.
- Implement updates and improvements as needed, considering changes in the environment or user requirements.

6.2 Performance Parameter

6.2.1 Object Detection Accuracy:

- Objective: Evaluate the system's ability to accurately identify and localize objects in real-time.
- Measurement: Calculate precision and recall to assess the accuracy of object detection. Precision is the ratio of true positive detections to the total positive detections, while recall is the ratio of true positive detections to the total number of actual positive instances.

- Target: Aim for high precision and recall values to minimize false positives and false negatives, ensuring reliable object detection.

6.2.2 Real-time Processing:

- Objective: Assess the system's capability to process video feeds and perform object detection without significant delays.
- Measurement: Measure the time taken to process each frame and ensure that the system operates within a defined real-time constraint (e.g., processing each frame in less than 100 milliseconds).
- Target: Achieve consistent and low processing times to enable real-time interaction with minimal lag.

6.2.3 Robustness to Lighting Conditions:

- Objective: Evaluate the system's performance under various lighting conditions, including low light and bright light scenarios.
- Measurement: Test the system's object detection accuracy in environments with different lighting levels, and analyze how well it adapts to variations in illumination.
- Target: Ensure that the system maintains reliable object detection across a range of lighting conditions.

6.2.4 Area Coverage:

- Objective: Evaluate the scalability of the system concerning the size of the tracking area.
- Measurement: Determine the maximum area the system can cover while maintaining accurate object detection.

- Target: Strive for scalability to accommodate tracking in both small and large physical or virtual spaces.

6.2.5 Processing Latency:

- Objective: Measure the time taken from capturing a frame to detecting and tracking objects, ensuring low latency for real-time interaction.
- Measurement: Calculate the processing latency by measuring the time difference between capturing a frame and completing object detection.
- Target: Minimize processing latency to provide a responsive and interactive user experience.

6.3 Efficiency Issues

6.3.1 Computational Resource Usage:

- Objective: Evaluate how efficiently the system utilizes computational resources (CPU, GPU, memory) during real-time object tracking.
- Measurement: Monitor resource consumption, such as CPU and GPU usage, memory usage, and ensure that the system operates within acceptable limits.
- Target: Aim for efficient resource utilization to prevent performance degradation and ensure the system can run on standard hardware.

6.3.2 Model Inference Speed:

- Objective: Assess the speed at which the trained machine learning model performs inference on each frame for object detection
- Measurement: Measure the time taken by the model to analyze a frame and identify objects.

- Target: Strive for fast model inference times to achieve real-time processing without significant delays.

6.3.3 Application Responsiveness:

- Objective: Evaluate how quickly the application responds to detected objects and triggers interactions.
- Measurement: Measure the time taken from object detection to the initiation of the corresponding application response.
- Target: Minimize delays between object detection and application response to enhance real-time interactivity.

6.3.4 System Throughput:

- Objective: Assess the overall throughput of the system, measuring the number of frames processed per unit of time
- Measurement: Calculate the frames per second (FPS) or throughput to ensure the system can handle a continuous stream of input frames.
- Target: Aim for a high throughput to maintain real-time performance across various scenarios.

6.3.5 Algorithmic Efficiency:

- Objective: Evaluate the efficiency of the object detection algorithm in terms of both accuracy and computational complexity.
- Measurement: Consider the trade-off between accuracy and computational efficiency. Evaluate the algorithm's performance in different scenarios.
- Target: Optimize the algorithm for a balance between accuracy and computational efficiency, ensuring effective real-time object tracking.

6.3.6 Energy Consumption:

- Objective: Assess the energy efficiency of the system, especially if it is running on battery-powered devices.
- Measurement: Monitor the system's energy consumption during operation and identify opportunities for optimization.
- Target: Strive for energy-efficient processing to extend the system's usability on battery-powered devices.

Summary

This chapter provides a detailed description of the Experimental setup with performance parameters and efficiency issues in this project. In the next chapter , Project Planning is discussed.

Chapter 7

Result and Evaluation

In the previous chapter, Experimental setup of system is discussed.
In this chapter, Result, evaluation and cost estimation is discussed.

7.1 Working Modules (GUI)

7.1.1 Results

In the figure 7.1, the initial window is displayed which contains the application which is projected on the wall. The window displays three functionalities namely

- Play: Used to start the game.
- Calibrate: Here four points are identified for further calibration of the window.
- Quit: Used to end the application.

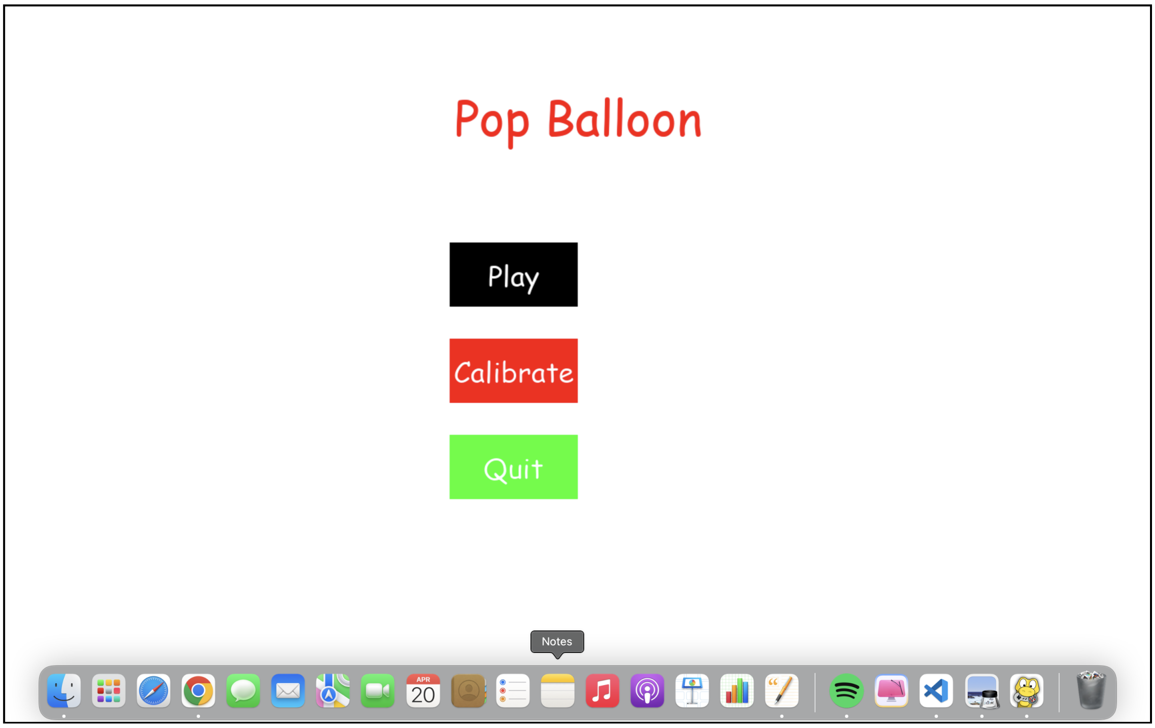

Figure 7.1: Initial Window

In the figure 7.2, we calibrated the window that is projected on the wall to ensure that the projected image accurately align with the physical boundaries of the screen, without proper calibration, the image maybe be skewed, stretched, or cutoff at the edges leading to a distorted viewing experience

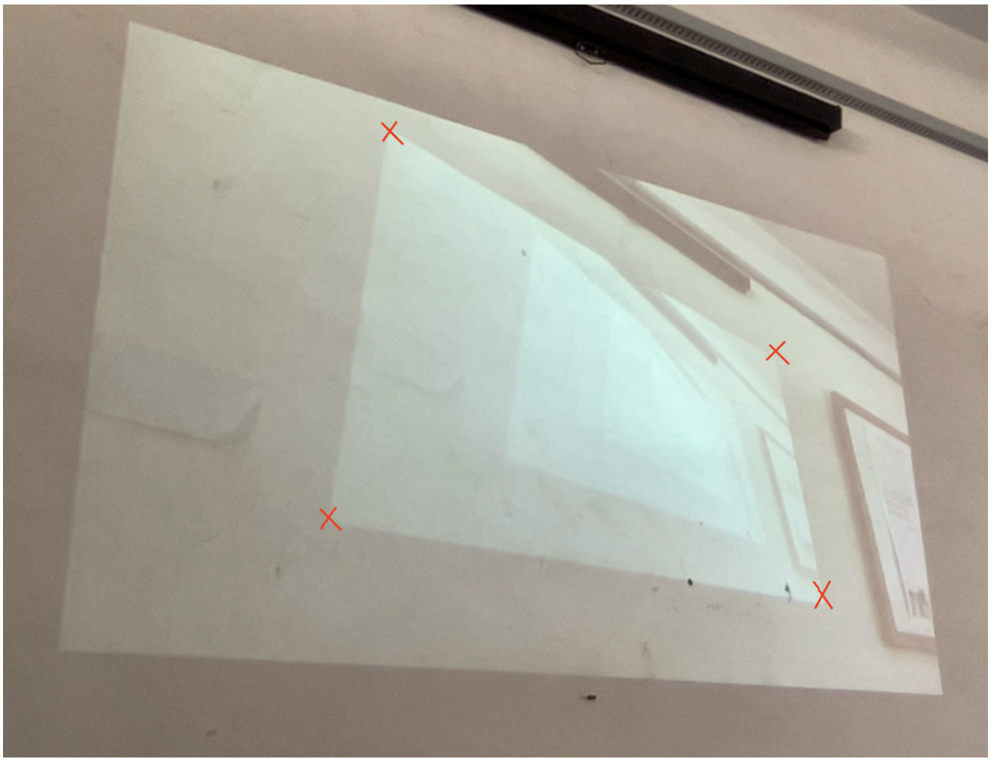

Figure 7.2: Calibration Window

In the figure 7.3 illustrates the projection of the balloon Application onto the wall using a projector. It demonstrates how the application, in which the balloons are floating, is portrayed in this manner and also it shows the score and the timing limit. The balloon serves as the focal point, casting its image onto the wall through the projector. This method effectively showcases the floating balloons within the application, providing a visually engaging experience for players.

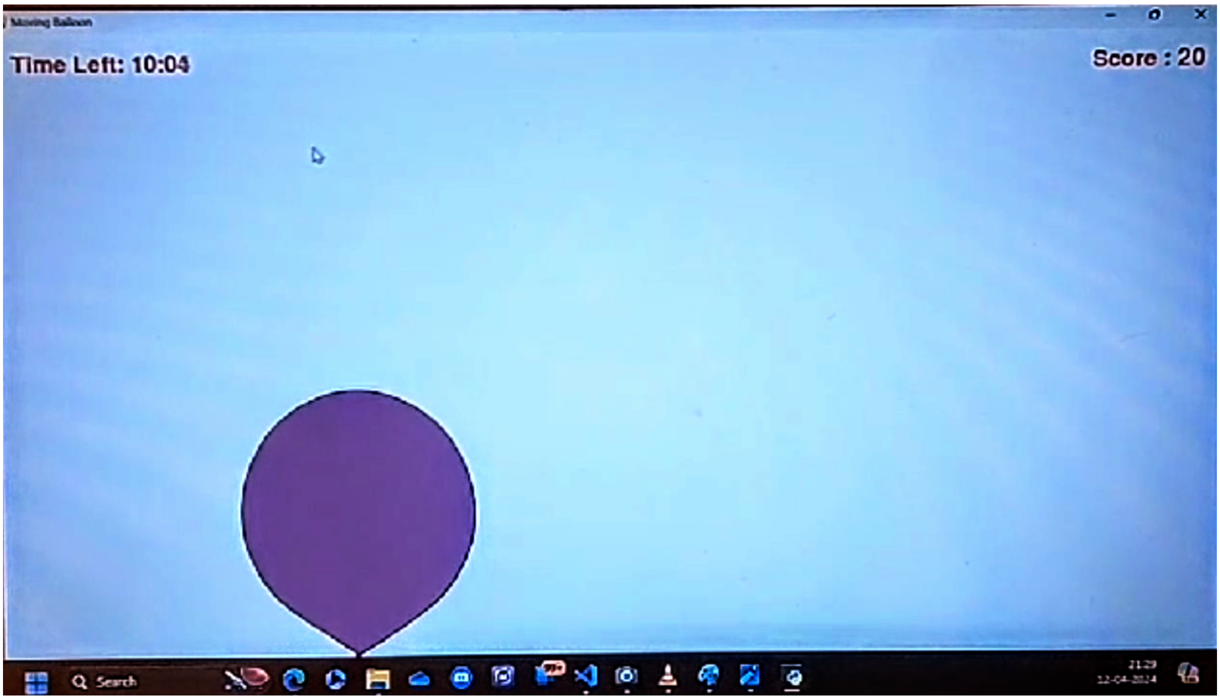

Figure 7.3: Application projected on Wall

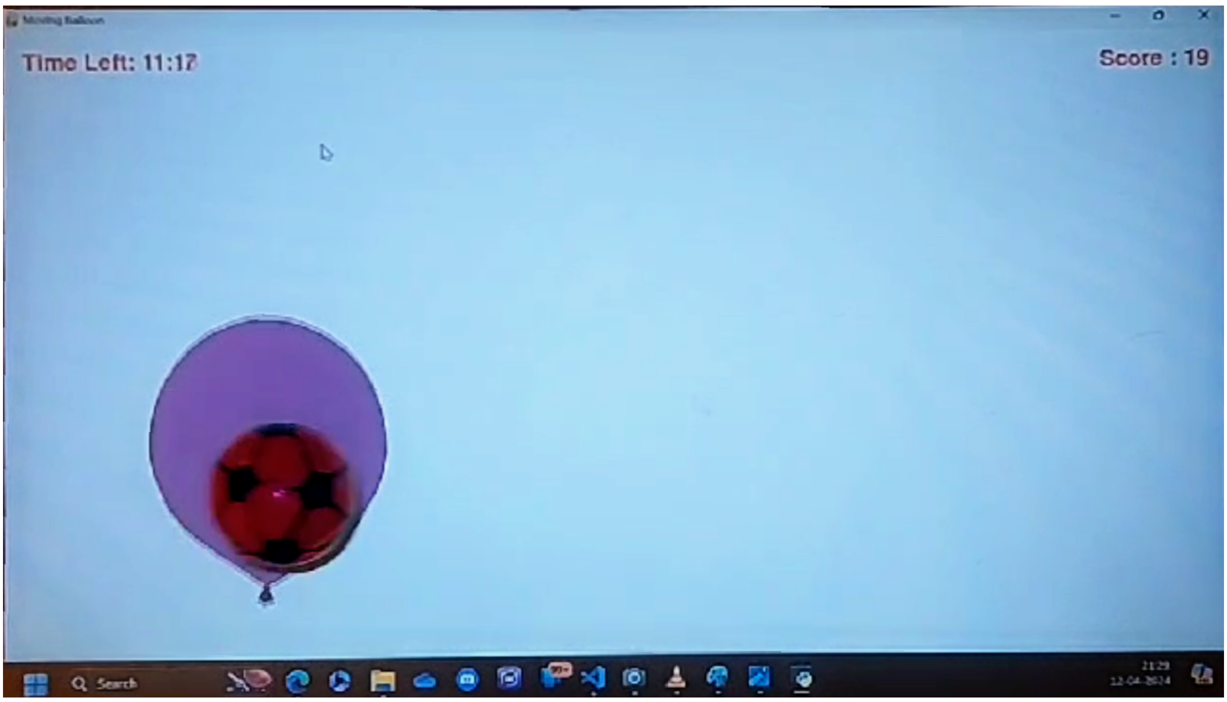

Figure 7.4: Ball targeted towards the Balloon

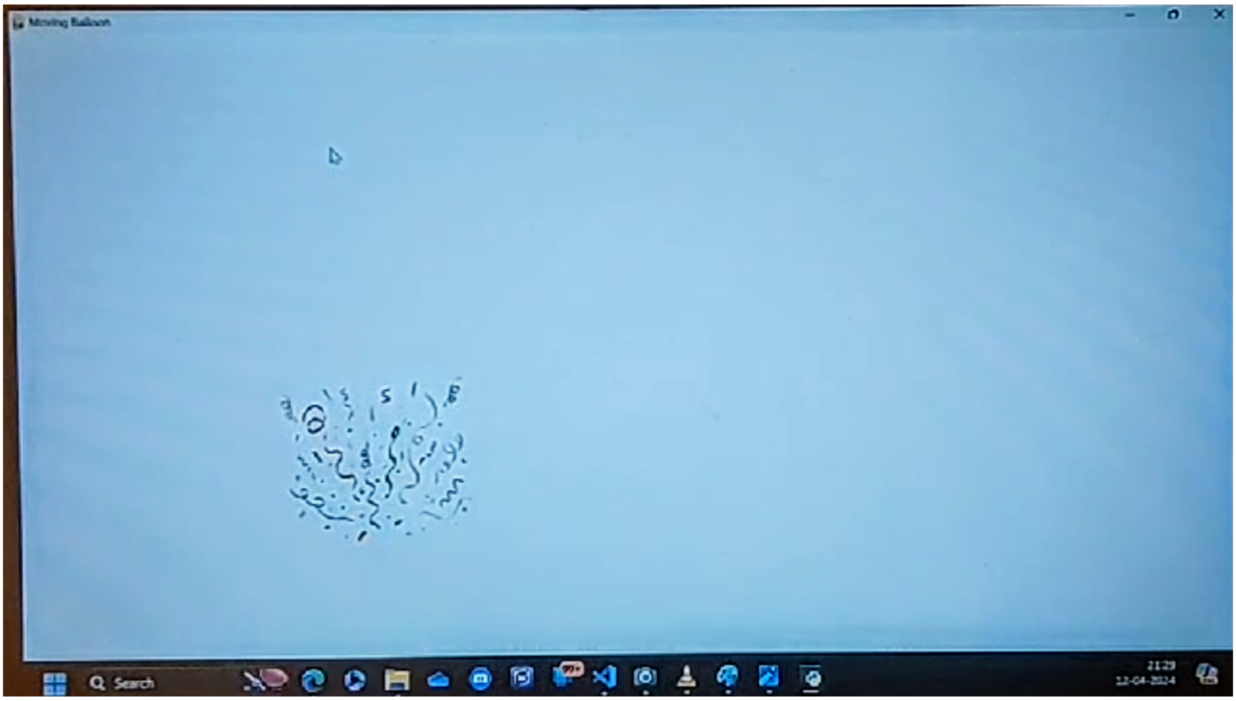

Figure 7.5: Balloon is burst by the ball

In the figure 7.4, the physical ball is thrown by the players towards the wall where the application is projected. Within this application, balloons are floating, and the players need to target these balloons.
In the figure 7.5, the target is hit at that point, causing the balloon to burst, and the score is counted.

7.1.2 Experimental Platforms

OpenCV plays a pivotal role in our project by providing robust tools and functionalities for image processing and computer vision tasks. In particular, we leverage several key components of OpenCV, including contours, Hough circles, NumPy, and warpPerspective, to achieve various objectives in our image processing pipeline.

- Contours: Contours are instrumental in our project for object detection and shape analysis. By identifying and extracting contours from images, we can delineate the boundaries of objects and subsequently analyze their shapes, sizes, and spatial relationships. This facilitates tasks such as object recognition, classification, and tracking within our application.

- Hough Circles: The Hough circles transform is a critical component in our project for detecting circular objects within images. By employing this technique, we can robustly identify and localize circular features, even in the presence of noise or varying lighting conditions. This capability is particularly valuable for applications requiring the detection of circular objects such as coins, lenses, or wheels.
- NumPy: NumPy serves as the backbone of our image processing pipeline, providing powerful array manipulation capabilities and mathematical operations. Leveraging NumPy arrays, we can efficiently store, manipulate, and process image data, enabling tasks such as filtering, enhancement, and feature extraction. NumPy's versatility and efficiency are indispensable for handling large volumes of image data in real-time or batch processing scenarios.
- Wrap Perspective: Application of perspective transformation using the wrap perspective technique allows us to correct geometric distortions within images. This becomes crucial when dealing with images captured from non-linear perspectives, such as those obtained from surveillance cameras or aerial drones. By rectifying perspective distortions, we ensure accurate feature extraction and defect localization.

7.2 Cost Analysis

This section briefly describes about the Total Cost of the Project. As displayed in Table 7.1, cost of the Projector is 6999. Cost of Ball is 100. Thus, total cost of Hardware Components in this project 7199. Also, COCOMO (Constructive Cost Model) model has been used to estimate the software development cost of the project. Details about the total number of Lines of Source code delivered, Development time and total manpower or the number of people required for the completion of this project has been given.

.

Sr no.	Hardware Components	Cost
1.	Projector	**6999**
2.	Ball	**100**
Total Cost : 6999+100 = 7199/-		
Software Cost Estimation		
1.	Approx. delivered line of source code(E=KLOC)	**572**
2.	Development Time(day in months)	**10 months**
3.	Manpower Required(P)	**4 people**

Table 7.1: Overall Cost Analysis of this Project

Chapter 8

Project Planning

8.1 Project Plan for Semester - I

SEMESTER - I		
1.	Searching for Project Topic and Reading Journal (IEEE) Papers	3rd and 4th Week -July 2023
2.	Finalization of project topic and scope	1st and 2nd Week -August 2023
3.	Project scope finalization	3rd Week -August 2023
4.	Project approval presentation	4th Week -August 2023
5.	Abstract preparation	1st Week-September 2023
6.	Working on literature review and architecture	2nd and 3rd Week - September 2023
7.	Project review I presentation	4th Week-September 2023
8.	Working on requirement specification and design	1st Week-October 2023
9.	Working on the experimental setup and performance parameter	2nd and 3rd Week -October 2023
10.	Project review II presentation	4th Week-October 2023
11.	Compilation of project report stage I	1st and 2nd Week - November 2023
12.	Preparation for project stage 1 examination	3rd Week-November 2023
13.	Project stage I examination	4th Week-November 2023

Table 8.1: Project Plan: Semester I

8.2 Project Plan for Semester - II

SEMESTER - II		
14.	Working on tools techniques and project implementation	4th Week -December 2023
15.	Working on partial results	1st to 4th Week -January 2024
16.	Project review III presentation	1st Week -February 2024
17.	Complete implementation and modular testing	2nd to 4th Week - September 2024
18.	Preparation for project participation	1st and 2nd Week -March 2024
19.	Project review IVth presentation	3rd and 4th Week -March 2024
20.	Compilation of project report stage to	1st and 2nd Week -April 2024
21.	Project participation	3rd and 4th Week -April 2024
22.	Preparation for project stage II examination	1st and 2nd Week -May 2024

Table 8.2: Project Plan: Semester II

Chapter 9

Conclusion

In conclusion, the real-time object tracking project encompasses critical considerations for optimal performance. Achieving high object detection accuracy through continuous model refinement is paramount, minimizing false positives and negatives. Real-time processing efficiency is crucial, necessitating low-latency frame analysis to ensure a seamless user experience. Robustness to varying lighting conditions is addressed through adaptive strategies, while scalability considerations ensure effective coverage across different tracking areas. Striking a balance between algorithmic efficiency and accuracy is vital for sustained system performance. Additionally, the project places emphasis on computational resource optimization, fast model inference, application responsiveness, and energy efficiency, all contributing to an overall well-optimized and high-performing system. Regular testing, monitoring, and updates will be key in maintaining effectiveness in diverse real-world scenarios. This project, with its focus on accuracy, efficiency, and adaptability, aims to deliver a robust real-time object tracking solution for interactive applications.

Bibliography

[1] William T. Freeman, David B. Anderson, Paul A. Beardsley, Chris N. Dodge, Michal Roth, Craig D. Weissman, and William S. Yerazunis Computer Vision for Interactive Computer Graphics Article in IEEE Computer Graphics and Applications DOI: 10.1109/38.674971

[2] VirtualTable: a projection augmented reality game, A. Dal Corso M. Olsen K. H. Steenstrup J. Wilm S. Jensen R. Paulsen E. Eiriksson,J. Nielsen J. R. Frisvad G. Einarsson H. M. Kjer.Department of Applied Mathematics and Computer Science, Technical University of Denmark.

[3] Markus Löchtefeld,Johannes Schöning, Michael Rohs,Antonio Krüger, LittleProjectedPlanet: An Augmented Reality Game for Camera Projector Phones.

[4] A Dice Game in Third-Person Augmented Reality, Richard Colvin, Ted Hung, David Jimison, Benjamin Johnson, Eben Myers, Tina Blaine Entertainment Technology Center Carnegie Mellon University 700 Technology Drive Pittsburgh, PA I521 9 USA

[5] Phys2D Java based open source physic engine , http://www.cokeandcode.com/phys2d/, 2006.

[6] LittleBigPlanet , http://www.littlebigplanet.com/, 2008. (Online; accessed 15-April-2009).

[7] WiiRemoteJ, http://www.world-of-cha0s. hostrocket.com/WiiRemoteJ/, 2008. (Online; accessed 15-April-2009).

[8] C. Alvarado and R. Davis. Resolving ambiguities to create a natural sketch based interface. In Proceedings of IJCAI-2001, 2001.

[9] P. Beardsley, J. Van Baar, R. Raskar, and C. Forlines. Interaction using a handheld projector. IEEE Computer Graphics and Applications, 25(1):39–43, 2005.

[10] A. D. Wilson. Playanywhere: a compact interactive tabletop projection-vision system. In UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology, pages 83–92, New York, NY, USA, 2005. ACM