USMAN INSTITUTE OF TECHNOLOGY

Affiliated with NED University of Engineering & Technology, Karachi

Department of Computer Science

B.S Software Engineering

FINAL YEAR PROJECT REPORT

Batch-2021

AIR TRACKER

By

HADIQA MEHDI	21B-007-SE
MUSFIRA KHALID BUTT	21B-027-SE
SAAD ATHER ALI	21B-066-SE
DUA FAZIL	21B-140-SE

Supervised by

Ms. MARVI JOKHIO

ST-13, Block 7, Gulshan-e-Iqbal, Abul Hasan Ispahani Road, Opposite Safari Park, P.O. Box 75300,
Karachi, Pakistan. Phone: 34978274-5; 34994305; 34982476; http://www.uit.edu

Submission Performa

Name (1) HADIQA MEHDI
(2) MUSFIRA KHALID BUTT
(3) SAAD ATHER ALI
(4) DUA FAZIL

Address (1) FB Area
(2) KAECHS Society
(3) Malir Cantt
(4) Shabbirabad

Title of Report: **Air Tracker**

Project Supervisor's Name: **Ms. Marvi Jokhio.**

Project Supervisor's Signature: ……………………

This report is submitted as required for the Project in accordance with the rules laid down by the Usman Institute of technology as part of the requirements for the award of the degree of Bachelor **Software Engineering**. I declare that the work presented in this report is my own except where due reference or acknowledgement is given to the work of others.

Signatures of students	Date
(1) ……………………………..	……………………..
(2) …………………………….	…………………….
(3) ……………………………..	…………………….
(4) ……………………………..	……………………

Acknowledgments

First and foremost, all praise is due to **Allah**, the Almighty, on whom we ultimately rely for sustenance and guidance.

We would like to express our sincerest gratitude to our **Vice Chancellor**, Dr. Vali Uddin, and **Head of Department**, Dr. Muhammad Wasim, at Usman Institute of Technology, for their guidance and support throughout this project.

Our deepest thanks go to **Ms. Marvi Jokhio,** our project supervisor, for her invaluable guidance, encouragement, and insightful feedback during the development of the Air-Tracker project. Her expertise and continuous support were crucial in overcoming challenges and enhancing the overall quality of the project.

We also extend our heartfelt thanks to all the **team members** for their collaboration, dedication, and relentless efforts in bringing the Air-Tracker project to life. The teamwork and collective ideas made this project a reality.

Special thanks to **Usman Institute of Technology** and its faculty for providing a nurturing learning environment that greatly contributed to our academic and project success.

Lastly, we extend our heartfelt appreciation to our **families** for their unwavering support, patience, and encouragement throughout this journey. Their belief in our abilities kept us motivated and focused.

This project is dedicated to everyone who contributed to its successful completion.

Abstract

The Air-Tracker project presents an innovative approach to enhancing presentation experiences through advanced gesture recognition technology. As a hands-free tool, Air-Tracker allows presenters to interact with their content without the need for traditional input devices like remotes, keyboards, or mice. By utilizing OpenCV for real-time image processing, Media Pipe for accurate hand tracking, and TensorFlow models (including CNNs, RNNs, and LSTMs) for gesture recognition and prediction, the system enables seamless slide control, real-time annotation, and smooth navigation during presentations. This project addresses the growing need for intuitive, non-intrusive interaction methods, particularly in academic and professional settings where effective engagement is key. Air-Tracker's ability to recognize gestures in real-time ensures that users can focus on delivering their content dynamically, without technical disruptions. By employing machine learning and computer vision techniques, Air-Tracker offers a hardware-independent, cost-effective solution adaptable to diverse environments, including classrooms, business meetings, and online conferences. The system's ease of use, coupled with its powerful technology, not only improves the presentation experience but also introduces a more natural way to control digital content. The Air-Tracker project represents a significant step forward in the development of interactive tools, providing an innovative solution that enhances both the functionality and accessibility of presentations. The following report details the research, design, implementation, and evaluation of the Air-Tracker system, underscoring its potential to revolutionize the way presentations are conducted across various industries.

Table of contents

Contents

List of tables

List of Figures

List of symbols and Units

1- Actor: Actor can be a person or stakeholder who initiates the use case.

2- Use case: activity, action or task performed by actor.

3- System Boundary: it is a conceptual boundary, which represents system’s environment.

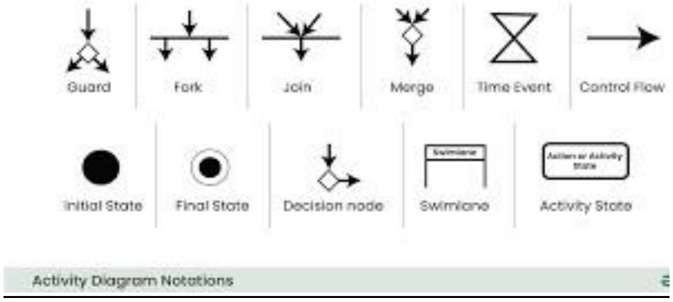
4- External Entity: represents actor that has predefine behavior.

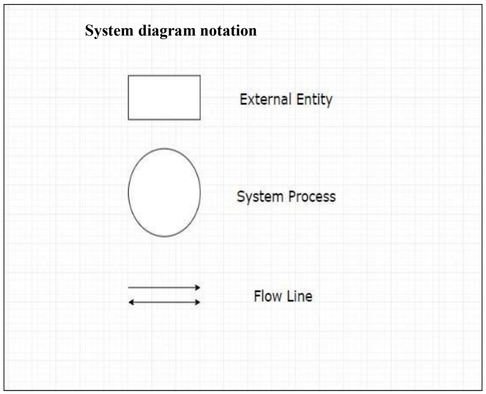
5- Relationship: The common cause or the connection between two entities.

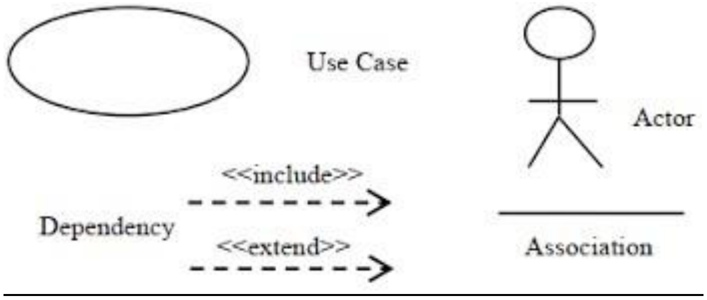
6- Action state: activities and action transmit.

7- Fork: control node splits in to two more flows.

8- Decision node: represents where decisions are taken.







CHAPTER I

INTRODUCTION

1. INTRODUCTION

In the evolving landscape of technology, the way we interact with digital content has significantly changed. Traditional presentation tools, such as remote clickers or keyboards, often limit a presenter's mobility, reduce engagement, and hinder the dynamic flow of information. The **Air-Tracker** project is designed to address these limitations by introducing a gesture-based control system for presentations, making presentations more intuitive, interactive, and engaging.

Air-Tracker is a cutting-edge presentation tool that allows users to control slides through hand gestures, enabling presenters to move between slides or interact with the content without the need for physical clickers or buttons. It provides a natural, hands-free way of controlling presentations, giving the presenter more freedom and flexibility. The system goes beyond simple slide navigation, incorporating **real-time annotation capabilities** to further enhance audience engagement. Presenters can use specific hand gestures to annotate slides on-the-fly, facilitating real-time customization of content to suit audience needs.

The core of **Air-Tracker** lies in its **gesture recognition** and **machine learning** systems, which use advanced algorithms to interpret hand movements and convert them into actions. To achieve this, Air-Tracker integrates a combination of technologies: **OpenCV** for real-time video processing, **Media-Pipe** for accurate hand tracking, and **TensorFlow** for machine learning. These tools work together to create a robust and reliable system capable of understanding complex hand gestures.

A notable strength of Air-Tracker is its use of **Convolutional Neural Networks (CNN)**, **Recurrent Neural Networks (RNN)**, and **Long Short-Term Memory (LSTM)** models. Each of these plays a specific role in enabling sophisticated gesture recognition.

CNN (Convolutional Neural Networks): CNNs are critical for extracting features from the video frames captured by the system's camera. They analyze each frame to detect hand shapes and gestures, identifying key points like the position and movement of fingers. CNNs are highly efficient in processing visual data, making them an ideal choice for identifying gestures from real-time video feeds.

RNN (Recurrent Neural Networks): RNNs are designed to handle sequential data, making them suitable for interpreting gestures that evolve over time, such as a swipe or a point. RNNs allow the system to understand not just individual hand positions but how these positions change over a sequence of frames, leading to more accurate gesture detection.

LSTM (Long Short-Term Memory): LSTMs, a type of RNN, are particularly useful in tracking longer sequences of movements, such as complex gestures that require multiple steps or continuous hand motion. LSTM networks remember previous frames in the sequence, allowing them to predict the full gesture based on the initial movements, which is vital for detecting more advanced gestures like drawing on a slide or making complex annotations.

In Air-Tracker, **Media-Pipe** handles the **hand tracking**, providing the system with precise data about the location and orientation of the user's hands. **TensorFlow**, on the other hand, processes this data to **predict the gestures** using CNN, RNN, and LSTM models. For example, if the user swipes their hand to the right, TensorFlow will identify this as a gesture to move to the next slide. If the user holds their hand in a specific position, TensorFlow may interpret this as a command to enter annotation mode, allowing the user to draw directly on the slide.

The system is designed to provide real-time feedback, meaning gestures are recognized and acted upon instantly, allowing for smooth and seamless control during presentations.

This ability to process hand gestures with high accuracy, speed, and flexibility makes Air-Tracker a gamechanger for presentations.

Air-Tracker's **real-time annotation** feature is also a major innovation. Using hand gestures, presenters can draw, highlight, or mark specific points on the slides as they speak, offering a highly interactive and engaging presentation style. The system can detect gestures associated with drawing (such as pointing or circling), and TensorFlow's machine learning models interpret these gestures and enable the real-time drawing on the slides.

Additionally, Air-Tracker enhances accessibility by providing an intuitive and easy-to-use interface for individuals with disabilities or those who have difficulty using traditional input devices like keyboards or clickers. This makes it a highly inclusive tool for a wide range of users in educational, corporate, and other professional settings.

In conclusion, the **Air-Tracker** project is a significant advancement in presentation technology, blending **gesture recognition** with **machine learning** to offer a seamless and interactive user experience. By incorporating **CNN**, **RNN**, and **LSTM** models, the system can handle both simple and complex gestures, ensuring precise and responsive control over presentation slides. The project's reliance on state-of-the-art technologies like **OpenCV**, **Media-Pipe**, and **TensorFlow** ensures high accuracy and scalability, positioning Air-Tracker as a future-proof solution for next generation presentations.

1.1 BRIEF SUMMARY:

Air-Tracker is an innovative presentation tool developed to enhance user interaction and engagement through **gesture-based control** and **real-time annotation** capabilities. Unlike traditional presentation tools that rely on physical remotes or clickers, **Air-Tracker** empowers presenters to seamlessly control their slides and annotate content directly using **hand gestures**.

This system is designed to eliminate the need for physical interaction with devices, allowing presenters to maintain a fluid connection with their audience while focusing on their content delivery.

At its core, **Air-Tracker** leverages cutting-edge technologies to ensure precise and responsive gesture recognition. The system utilizes **OpenCV** for real-time image processing, capturing the presenter’s gestures through a standard camera. The gestures are then tracked using **Media Pipe**, an advanced hand-tracking library that identifies and tracks the position of hands and fingers with high accuracy. This ensures that even subtle movements are detected, allowing for refined control over presentations.

To recognize and predict the meaning of gestures, the project integrates powerful machine learning models, including **Convolutional Neural Networks (CNNs)**, **Recurrent Neural Networks (RNNs)**, and **Long Short-Term Memory (LSTM)** networks. CNNs handle the spatial features of the gestures, enabling accurate recognition of hand shapes and positions. Meanwhile, RNNs and LSTMs focus on analyzing the temporal sequence of movements, making it possible to interpret dynamic gestures that occur over time, such as swipes or finger pointing. These models work in tandem to deliver smooth, responsive, and intuitive control, ensuring that the user experience is both flexible and reliable.

The combination of these advanced technologies results in a dynamic, gesture-controlled presentation system that significantly enhances the overall experience. **Air-Tracker** makes presentations more **intuitive**, **interactive**, and **accessible**, freeing presenters from the constraints of traditional input devices and allowing them to focus fully on engaging their audience. Whether for professional conferences, academic lectures, or remote presentations, Air-Tracker offers a futuristic, user-friendly solution that revolutionizes the way presentations are delivered.

CHAPTER II

2.BACKGROUND AND PROBLEM

2.1 BACKGROUND AND LITERATURE REVIEW:

The background section introduces the growing demand for intuitive, non-intrusive presentation tools in both professional and academic settings, particularly as the modern work and learning environments evolve. Traditional input devices like keyboards, remotes, and mice are increasingly seen as cumbersome and restrictive, especially in dynamic, interactive presentation settings. These devices can hinder a presenter's ability to engage freely with their audience and move naturally around the stage or classroom. The necessity for more fluid, seamless control mechanisms is evident as presentations become more complex, incorporating multimedia elements and requiring swift, precise interaction with digital content. This section emphasizes how emerging technologies, particularly those that leverage natural user interfaces like gesture-based controls, are transforming the way individuals interact with presentation software, making these experiences more engaging, efficient, and accessible.

Moreover, the rise of remote work, online lectures, and virtual meetings has further highlighted the shortcomings of traditional input devices. As presenters increasingly operate in digital environments where face-to-face interaction is limited, the need for tools that allow seamless, non-disruptive control over presentations has become paramount. Virtual and hybrid workspaces demand technologies that enable presenters to maintain audience attention and smoothly transition between slides or emphasize key points without breaking the flow of the presentation.

Gesture-based systems provide an intuitive solution, offering a hands-free method for controlling content in real-time, which aligns with the evolving needs of professionals, educators, and public speakers in the post-pandemic digital world. These technologies not only improve user experience but also offer potential for enhancing audience engagement, as presenters are able to interact with their materials in a more dynamic and expressive manner.

In the literature review, existing technologies and research in the domain of gesture recognition are explored in-depth, particularly within the field of computer vision.

The review covers a range of gesture-based interaction methodologies, with a primary focus on **hand-tracking techniques** that utilize advanced machine learning models such as **Convolutional Neural Networks (CNNs)**, **Long Short-Term Memory (LSTMs)**, and **Recurrent Neural Networks (RNNs)**. These models are widely regarded for their capacity to accurately recognize and interpret gestures in real-time, making them ideal for applications in gesture-controlled systems. The review delves into how CNNs have been effectively used in image classification tasks, including recognizing static hand gestures by extracting key features from input frames. Meanwhile, RNNs and LSTMs are noted for their ability to analyze sequential data, making them especially useful for tracking dynamic gestures over time.

The ability of these models to learn and interpret patterns in movement positions them as foundational technologies for projects like Air-Tracker, where both static and dynamic gestures must be recognized with high precision.

The literature review further examines the application of these techniques in similar projects, offering insights into their success rates, challenges, and limitations. For instance, CNNs have been highly successful in achieving high accuracy in image classification tasks, but their performance can degrade in real-world conditions with varying lighting or backgrounds. LSTMs and RNNs, while effective in sequence recognition, can encounter challenges with computational complexity and require substantial training data to ensure robustness. Additionally, the review discusses the importance of **real-time gesture recognition**, which is a key feature of systems like Air-Tracker. Unlike sensor-based approaches that often require the user to wear specialized hardware (e.g., gloves with sensors), camera-based recognition systems provide a more **cost-effective and user-friendly solution**, relying solely on readily available hardware such as a laptop or desktop camera. This allows for greater accessibility and ease of use, as users can interact with the system using natural hand movements without the need for additional devices.

Moreover, the literature identifies key gaps in current research and solutions, emphasizing the lack of systems that are both hardware-independent and accessible to a wide range of users. Many existing gesture-based systems either rely on expensive, specialized hardware or are limited in terms of their user interface flexibility. These limitations create barriers to widespread adoption, particularly in education and business environments where accessibility and ease of integration are critical.

The review highlights the need for a solution that offers **high recognition accuracy**, real-time response, and compatibility with a variety of hardware setups, all while being intuitive and customizable.

The **Air-Tracker** project addresses these gaps by aiming to provide a **hardware-independent, user-friendly system** that enhances the overall presentation experience through intuitive controls and real-time feedback. The literature further underscores the importance of user engagement in presentation settings, noting that systems which allow presenters to move freely and interact with content through natural gestures can significantly improve audience connection and attention, making presentations more effective and memorable.

2.2 BACKGROUND AND PROBLEM

In professional and academic environments, delivering engaging presentations is crucial for effective communication. However, traditional input devices like keyboards, remotes, and mouse often limit the presenter's ability to interact seamlessly with digital content. Presenters are typically tethered to a device, requiring physical movement back and forth, which disrupts the flow of the presentation and engagement with the audience. For instance, when a presenter needs to annotate or navigate through slides, the reliance on remotes or clickers limits the freedom of movement. This limitation is particularly challenging when addressing large audiences, as fluid mobility and spontaneous interactions are key to maintaining attention and enhancing the presentation's impact.

Air-Tracker addresses these limitations by enabling gesture-based controls and real-time annotations directly from the presenter's standing position. By leveraging advanced hand-gesture recognition technology, presenters can control slides, annotate, and engage without returning to their devices, thereby allowing a more natural interaction with the content.

2.3 APPROACHES

2.3.1 Computer Vision Approach

The computer vision approach is fundamental to the "Air-Tracker" project, focusing on the analysis of visual data captured by cameras. By employing advanced image processing techniques, the system interprets hand gestures in real time, enabling intuitive interaction. This method minimizes the reliance on traditional input devices, facilitating a more natural user experience. The camera captures continuous video frames, which are analyzed to extract meaningful features related to hand positions and movements.[4]

2.3.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) play a crucial role in the gesture recognition process within the project. By processing video frames through multiple layers, CNNs can identify and learn patterns associated with different hand gestures. This deep learning technique enhances the system's ability to recognize complex gestures with high accuracy.

The trained CNN model analyzes the features extracted from the video frames, providing real-time feedback on hand movements. Utilizing CNNs allows for effective generalization across different users and environments, further enhancing the tool's usability.[7]

2.3.3 Real-Time Hand Tracking

Real-time hand tracking is essential for ensuring seamless interaction during presentations. By using libraries like Media-Pipe, the system continuously monitors hand gestures without the need for additional hardware or sensors. This tracking capability enables the "Air-Tracker" to respond immediately to user inputs, providing an interactive experience. The algorithm detects key points on the hands and calculates their positions in real time, allowing for precise gesture interpretation. This feature enhances the overall effectiveness of the presentation tool, making it more engaging for both the presenter and the audience.[3]

2.3.4 Gesture Recognition Techniques

Gesture recognition techniques are pivotal to the functionality of the "Air-Tracker" system, as they facilitate the interpretation of hand movements. The project employs machine learning algorithms trained on a diverse dataset of hand gestures, enabling the classification of various gestures such as swipes and drawing actions. This classification process is essential for translating user movements into actionable commands within the presentation software. By using a combination of CNNs and RNNs, the system can achieve higher accuracy in gesture detection, even in dynamic environments. Implementing these techniques allows for a fluid interaction model that enhances the presenter's ability to engage with their audience.[2]

2.3.5 Gesture Classification

Gesture classification is a critical component of the recognition process, where the system identifies specific hand movements and assigns them to predefined categories. The project utilizes supervised learning methods to train models that can accurately classify gestures based on the features extracted from video frames. This classification process ensures that the system can differentiate between similar gestures, improving usability and reducing errors during presentations. By providing a clear mapping between gestures and actions, the "Air-Tracker" enhances user confidence in interacting with the presentation tool. This precision is particularly important in high-stakes environments where clear communication is essential. [1]

2.3.6 Temporal Sequence Analysis

Incorporating Recurrent Neural Networks (RNNs) into the gesture recognition framework allows the system to analyze the temporal aspects of hand movements. This capability is essential for understanding the context and sequence of gestures, which enhances the accuracy of recognition. RNNs consider the order and timing of gestures, enabling the system to differentiate between similar movements based on their sequence. By leveraging this temporal analysis, the "Air

Tracker" can improve its response to user commands, making interactions more fluid and intuitive. This approach addresses the limitations of static gesture recognition systems, providing a more dynamic user experience.[8]

2.3.7 Real-Time Annotation

Real-time annotation allows presenters to draw or highlight elements on their slides as they speak, facilitating a more interactive dialogue. The system interprets specific hand gestures as drawing commands, enabling users to create visual aids instantly. This capability helps in clarifying complex concepts and allows for spontaneous adjustments based on audience feedback. By integrating this feature into the "Air-Tracker," presenters can maintain eye contact with their audience while interacting with the content, which is crucial for effective communication. This innovation enhances the overall impact of presentations, making them more memorable and engaging.[3]

2.3.8 Slide Navigation

Gesture-based slide navigation provides a seamless way for presenters to control their presentations without physical devices. The system translates specific hand gestures into commands for advancing or reversing slides, enabling fluid movement through the presentation. This functionality allows presenters to focus on their delivery rather than managing equipment, enhancing the overall user experience. By supporting various gestures for navigation, the "Air Tracker" promotes flexibility and adaptability during presentations. This intuitive control mechanism empowers presenters to engage more dynamically with their audience, fostering a more interactive environment.

2.3.9 Software and Tools

The success of the "Air-Tracker" project relies on the integration of various software tools and libraries that enhance its functionality. OpenCV serves as the backbone for real-time image processing, enabling accurate gesture detection and hand tracking.

TensorFlow is utilized for developing machine learning models that facilitate gesture recognition and classification. Media Pipe complements these technologies by providing efficient hand tracking capabilities, allowing for real-time analysis without the need for specialized sensors. By leveraging these tools, the "Air Tracker" ensures that its functionalities are robust and user-friendly, making the technology accessible to a broader audience

2.3.9.1 OpenCV

OpenCV is a powerful library for computer vision tasks, enabling the "Air-Tracker" to process video streams and extract relevant features in real time. Its capabilities include image filtering, edge detection, and contour analysis, which are essential for accurate hand gesture recognition. By employing OpenCV, the project can efficiently analyze visual data and make real-time decisions based on hand movements. The library's versatility and efficiency make it a suitable choice for the dynamic requirements of the presentation tool. This integration ensures that the system remains responsive and capable of adapting to various presentation environments.

2.3.9.2 TensorFlow

TensorFlow is a widely-used framework for machine learning and deep learning applications, and it plays a crucial role in training the gesture recognition models for the "Air-Tracker." The library provides powerful tools for building and optimizing CNNs and RNNs, allowing the project to achieve high accuracy in gesture classification. By leveraging TensorFlow's capabilities, the project can develop models that learn from large datasets and generalize well across different users. This framework not only enhances the performance of the gesture recognition system but also simplifies the process of model training and evaluation. The use of TensorFlow ensures that the "Air-Tracker" remains at the forefront of technological advancements in machine learning.

2.3.9.3 Media-Pipe

Media-Pipe is an open-source framework designed for building multimodal applied machine learning pipelines, and it significantly enhances the capabilities of the "Air-Tracker." By providing efficient hand tracking and gesture recognition features, Media-Pipe allows the project to implement real-time analysis with minimal latency. Its pre-built models and easy-to-use APIs simplify the integration of complex functionalities, making it accessible to developers. This framework ensures that the system can accurately track hand movements and interpret gestures effectively, contributing to a seamless user experience. Media-Pipe's robustness and efficiency make it an ideal choice for enhancing the interactive features of the presentation tool.[2]

2.4 METHODOLOGY

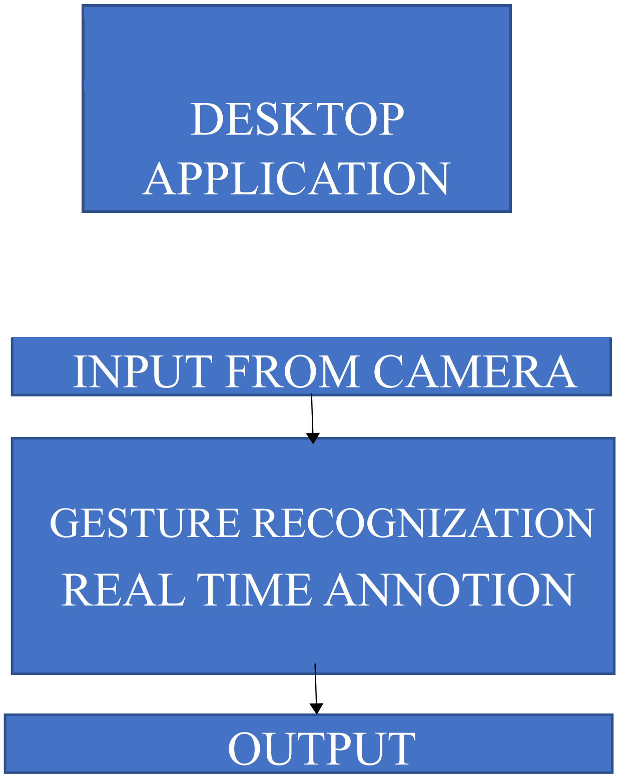
The primary goal of the "Air-Tracker" project is to develop an intuitive gesture-based application that enhances user interaction during presentations. The application will utilize machine learning prediction models to recognize hand gestures and convert them into actionable commands for slide control and real-time annotations.

2.4.1 Data Extraction and Machine Learning:

We will employ a supervised learning approach to machine learning. The process begins with data collection, where video recordings of hand gestures are captured using standard cameras. This data will then be divided into training and testing sets for model training and evaluation.

The collected data will undergo preprocessing, including normalization and augmentation, to ensure robustness. We will implement various supervised learning models, such as Convolutional Neural Networks (CNNs) for gesture recognition and Recurrent Neural Networks (RNNs) for analyzing temporal sequences of gestures**)**, specifically Long Short-Term Memory (LSTM) networks, to capture the temporal dimension of the gestures. RNNs are designed to process sequential data, allowing the model to understand the dynamics of gestures over time, such as distinguishing between similar hand shapes but different motions The entire process will be fine-tuned to ensure the models generalize well to unseen gestures, ensuring reliable hand gesture recognition in real-world scenarios.

System Flow:



2.4.2 Application Development

The desktop application will be developed using Python and various libraries, including OpenCV for image processing and user interface design. The integration of gesture recognition with popular presentation software will be a key focus, allowing users to control slides effortlessly. The application will also incorporate features for real-time drawing and annotations on presentation slides, providing an interactive platform for presenters. [6]

2.4.3 Similar Projects Comparison:

Features	Air-Tracker (Our project)	Leap Motion	Myo-Armband
Words Covered	(presentation commands)	(general motion control)	(gesture-based commands)
Accuracy	85-90% (using CNN, RNN, LSTM)	70-80% (hand tracking)	60-70% (sensor-based)
Limitations	Requires proper lighting for hand gesture recognition	Limited to short-range hand motions	Restricted to fixed arm positions
Reference	-	[4]	[6]

Table 1 comparison table

CHAPTER III

3.AIM AND STATEMENT

3.1 AIM AND STATEMENT OF PROBLEM

In the realm of modern presentations, there is a growing need for innovative solutions that facilitate seamless user interaction and engagement. Presenters often face significant challenges due to the limitations of traditional input devices, such as remotes, keyboards, and mouse. These devices can disrupt the natural flow of presentations, particularly in large or formal settings, where mobility and audience engagement are essential.

3.2 AIM OF THE PROJECT

The aim of the **Air-Tracker** project is to revolutionize the way presentations are controlled by eliminating the need for physical input devices. The project seeks to create a gesture-based system that allows presenters to navigate slides, make real-time annotations, and interact with their content from a distance using only hand gestures. The solution is particularly tailored for large audiences and formal presentation environments, where freedom of movement and audience engagement are crucial. By integrating advanced machine learning and computer vision technologies, the **Air Tracker** enhances user convenience and interaction during presentations, providing a seamless experience for both presenters and their audiences.

3.3 STATEMENT OF PROBLEM

In traditional presentation setups, presenters face several challenges due to the reliance on physical control devices such as remotes, keyboards, and mice. These limitations can disrupt the flow of presentations, especially in large or professional settings. The primary problems include:

3.3.1 Limited Mobility

Presenters often have to move back and forth to control slides or annotate, which disrupts their engagement with the audience and breaks the continuity of the presentation.

3.3.2 Physical Device Dependency

The requirement to hold or use a physical input device limits the natural flow of movement, which can be awkward, particularly when handling complex presentations.

3.3.3 Interruption of Audience Engagement

Frequent interruptions caused by the need to operate devices may reduce audience engagement, affecting the overall impact of the presentation.

3.3.4 Annotation Constraints

In traditional setups, annotating content requires proximity to the device, reducing the presenter's ability to dynamically interact with slides from a distance. The **Air-Tracker** project seeks to resolve these issues by offering an innovative, hands-free method for controlling presentations, thus significantly improving the user experience.

3.4 SCOPE OF THE PROJECT

3.4.1 In-Scope

The **Air-Tracker** project focuses on creating a robust, gesture-based control system for presentations, aiming to provide seamless interaction and real-time annotation using hand gestures.

The features and components that are part of the project's scope include:

3.4.1.1 Real-Time Hand Gesture Recognition

Implementing computer vision algorithms using **OpenCV** and **TensorFlow** to track hand movements and recognize gestures in real-time, allowing for slide navigation and annotation.

3.4.1.2 Presentation Control

Gestures for navigating through slides (e.g., swipe gestures for moving forward and backward), enabling presenters to maintain focus on their audience while controlling the presentation.

3.4.1.3 Annotation Capability

Presenters can use gestures to draw, highlight, or annotate content directly on the slides, making real-time interaction more dynamic.

3.4.1.4 User-Friendly Interface

A simple, intuitive interface for configuring gesture controls and setting up the application, making it accessible for presenters with varying levels of technical expertise.

3.4.1.5 Hardware Requirements

The system will work with commonly available hardware, including standard webcams and computers, without the need for specialized equipment.

3.4.2 Out of Scope

To ensure focused development and timely completion, certain features and functionalities are excluded from the current version of Air-Tracker.

3.4.2.1 Voice Recognition

While initially considered, the integration of voice commands for slide control has been excluded from this project phase due to time constraints and the complexity of combining both gesture and voice recognition.

3.4.2.2 Multi-Platform Support

Although there is potential to extend compatibility to platforms like **Google Slides** or **Apple Keynote**, the current scope is limited to Microsoft PowerPoint to ensure full functionality and reliability on a single platform.

3.4.2.3 Mobile Application

The project is focused solely on a **desktop application**; development for mobile devices or tablets is not within the current scope.

3.4.2.4 Advanced Gesture Customization

Customizing complex gestures (beyond basic swipes and annotations) is not included in this version, though it may be explored in future iterations of the project.

3.4.2.5 3D Gesture Tracking or Specialized Sensors

The system will rely on 2D gesture tracking using standard webcams, with no support for specialized hardware like **Leap Motion** sensors or other 3D tracking devices.

3.5 PROBLEMS WITH STAKEHOLDERS

3.5.1 Presenters

For presenters, the main challenge is the **lack of mobility** while engaging with their audience. Presenters are often confined to specific spots near their devices in order to control their slides or annotate content. This limitation hinders their ability to move freely and maintain a strong connection with their audience, reducing the overall effectiveness of their presentation.

3.5.2 Event Organizers

Event organizers face logistical challenges in ensuring that the presentation flow remains smooth and uninterrupted. The dependence on physical input devices like remotes or keyboards can lead to **technical difficulties** or **malfunctions**, especially in larger venues, where signal strength and device range become problematic.

3.5.3 Audience

From the audience's perspective, presentations often feel less engaging when the presenter is distracted by technical operations, such as using a remote or returning to their computer to change slides or annotate. The **disruptions** caused by these actions reduce the overall impact of the presentation and may lead to a loss of focus among the audience.

3.6 TECHNOLOGY

3.6.1 Computer Vision

The **Air-Tracker** uses computer vision technology to recognize and interpret hand gestures. **OpenCV** and **Media Pipe** are key libraries that provide real-time hand tracking and gesture detection capabilities. These technologies enable the system to detect movements accurately, ensuring smooth interaction during presentations.

3.6.2 Machine Learning

The system relies on **TensorFlow** to implement machine learning models for gesture recognition. These models are trained using large datasets to recognize specific hand gestures. Machine learning enhances the system's ability to adapt to various hand shapes and movements, making it more efficient in real-world environments.

3.6.3 Hardware

The technology runs on standard computers and uses regular webcams to capture hand gestures. While most users can rely on built-in webcams, the system is designed to be compatible with external, higher-quality cameras for enhanced performance.

3.6.4 Real-Time Processing

To ensure a smooth user experience, real-time processing is crucial. The combination of computer vision, machine learning, and gesture detection technologies allows the system to interpret gestures instantly, providing immediate feedback and control during live presentations.

3.7 PROJECT RISK

3.7.1 Accuracy of Gesture Recognition

The effectiveness of the Air-Tracker heavily depends on accurately recognizing hand gestures in real-time. Factors like poor lighting, different hand shapes, backgrounds, or camera angles may cause the system to misinterpret gestures or produce unreliable results, leading to user frustration and decreased adoption.

3.7.2 Hardware and Performance Limitations

The Air-Tracker system relies on the performance of standard webcams and the processing power of computers. Users with low-end devices may experience performance issues such as lag, gesture recognition delays, or incorrect responses, which could negatively impact user experience.

3.7.3 User Adoption and Learning Curve

For presenters unfamiliar with gesture-based systems, there may be a steep learning curve. If the interface is too complex or difficult to understand, users might find it challenging to adopt the system, leading to dissatisfaction and low engagement.

3.7.4 Time Constraints and Feature Creep

With limited development time, the team may struggle to complete all the desired features, especially as the complexity of the project increases. The risk of "feature creep," or adding extra functionalities beyond the original scope, could further delay the project's completion.

3.8 OVERCOME PROBLEM

3.8.1 Problem 1: Low Accuracy in Gesture Recognition

To improve accuracy, diverse training data and data augmentation will be used to help the model adapt to different environments. Real-time feedback will allow users to correct their gestures as needed. Continuous model learning will further enhance accuracy by adjusting based on user interaction.

3.8.2 Problem 2: Hardware Limitations

Optimizing algorithms for efficiency will ensure the system works on lower-end devices. Clear hardware specifications will guide users on expected performance, and compatibility with external webcams will provide better gesture recognition. This will help users with varying device capabilities.

3.8.3 Problem 3: Complexity of User Interface and Learning Curve

A simplified, user-friendly interface with clear controls will make the system easy to use. In-app tutorials and pre-configured gesture settings will help users get started quickly. This approach minimizes complexity and ensures a smooth user experience.

3.8.4 Problem 4: Time Constraints

By prioritizing essential features and using an agile development approach, the team will manage time effectively. Phased releases will allow core functionality to be delivered on time while additional features are added later. This ensures timely project completion without compromising quality.

CHAPTER IV

4.HARDWARE, SOFTWARE, ANALYSIS AND REQUIREMENT

4.1 HARDWARE, SOFTWARE, ANALYSIS AND REQUIREMENT

This section offers an in-depth exploration of the hardware and software requirements essential for the successful development and deployment of the **Air-Tracker** project. The Air-Tracker project seeks to revolutionize how users interact with presentation software by providing a **gesture-based control system** that enables hands-free navigation, real-time annotation, and slide control. This document covers the system's key requirements across multiple domains—fact-finding, hardware specifications, software infrastructure, machine learning models, and application needs. By addressing these elements, the project aims to deliver an intuitive and robust system that significantly improves user engagement and interaction, eliminating reliance on traditional input devices like remotes, keyboards, or mice. Below is a detailed breakdown of the requirements needed to ensure seamless real-time gesture recognition, which is at the core of Air-Tracker's functionality

4.2 FACT FINDING REQUIREMENT

The scope of the **Air-Tracker** project aims to develop a real-time gesture-based presentation control system. The system allows presenters to control Microsoft PowerPoint slides through hand gestures, eliminating the need for traditional input devices like keyboards and mice. Fact-finding involved understanding user needs and the technical requirements for gesture recognition, ensuring that the application provides seamless interaction during live presentations.

4.3 HARDWARE REQUIREMENT

4.3.1 Computer System

A laptop or desktop with a built-in or external camera capable of real-time video capture. Minimum specifications include:

- Processor: Intel i5 or above
- RAM: 8GB or more
- Camera: A standard webcam for gesture detection

4.3.2 Camera Requirements

The **Air-Tracker** project relies heavily on a stable and continuous video feed for effective gesture recognition. Therefore, the camera should meet the following specifications:

D resolution (720p or higher) to ensure accurate gesture tracking.

Frame Rate: A minimum frame rate of 30 FPS (frames per second) for smooth video capture and real-time processing.

4.4 SOFTWARE REQUIREMENT

4.4.1 Operating Systems

The Air Tracker system should be compatible with multiple operating systems, including Windows, macOS, and Linux.

4.4.2 Programming Language:

- **Python:** Used for backend development, particularly for implementing the machine learning models and data processing tasks.

4.4.3 Frameworks & Libraries:

- **OpenCV:** Employed for real-time image and video processing, allowing the system to detect and interpret hand gestures from the camera feed.

- **TensorFlow:** A machine learning framework used to train Convolutional Neural Networks (CNNs) for accurate gesture recognition.

- **Media Pipe:** A Google framework that provides real-time hand tracking and gesture detection capabilities, simplifying gesture recognition implementation.

- **NumPy:** A library for numerical and matrix operations essential for processing data used in machine learning models.

4.5 USER REQUIREMENT

4.5.1 Presenter Requirements

- As a presenter, I want to control my PowerPoint slides using hand gestures in real-time so that I can engage my audience without needing a keyboard or mouse.
- As a presenter, I want the application to provide immediate feedback on my gestures so that I can confirm my actions during the presentation without delays.
- As a presenter, I want the ability to customize gesture controls so that I can tailor the application to fit my unique presentation style.
- As a presenter, I want to use a hand gesture to control a pointer on my slides so that I can draw attention to specific parts of my presentation.
- As a presenter, I want to use gestures to highlight important sections of my slide so that I can emphasize key points during the presentation.
- As a presenter, I want to move between slides using left- and right-hand gestures so that I can seamlessly transition without manual intervention.

- As a presenter, I want to pause a video within my presentation using a specific gesture so that I can discuss the video content with my audience.
- As a presenter, I want to play a paused video with a different hand gesture so that I can resume my presentation when needed.

- As a presenter, I want the ability to use a gesture to zoom in on a section of my slide so that I can show details more clearly to my audience.
- As a presenter, I want to control volume levels with a hand gesture so that I can adjust sound without disrupting the flow of the presentation.
- As a presenter, I want to toggle the laser pointer on and off using gestures so that I can control it without the need for extra hardware.
- As a presenter, I want the ability to undo a gesture action (like going back a slide) so that I can correct mistakes quickly.

4.5.2 Developer Requirements

- As a developer, I want access to diverse gesture datasets to effectively train the machine learning models for improved recognition accuracy.
- As a developer, I want the application to be compatible with multiple presentation software options (e.g., PowerPoint, Google Slides) so that it can cater to various users.
- As a developer, I want to integrate gesture recognition technologies (e.g., OpenCV, TensorFlow, Media Pipe) to ensure the system is accurate and responsive.
- As a developer, I want to provide a simple, intuitive user interface that allows presenters to calibrate and test their gestures before the presentation starts.
- As a developer, I want to ensure the system works on different operating systems (Windows, macOS, Linux) so that it reaches a wider audience.

- As a developer, I want to offer customization options for gesture sensitivity so that users can adjust the system based on their preferences.
- As a developer, I want the system to run efficiently on computers with average specifications so that it is accessible to most users.
- As a developer, I want to log gestures and system usage to provide insights for performance improvements and updates.
- As a developer, I want to ensure that the gesture detection runs in real-time with minimal latency to improve user experience during live presentations.

4.5.3 Audience Requirements

- As an audience member, I want the presenter's gestures to be smooth and responsive so that I can follow along with the presentation without distractions.
- As an audience member, I want the highlighted content to be clearly visible so that I can understand the emphasis made by the presenter.
- As an audience member, I want the slides to move fluidly without pauses or delays so that the presentation is engaging and professional.
- As an audience member, I want the presenter to easily zoom in on content so that I can see details better from a distance.
- As an audience member, I want the video or audio elements of the presentation to play or pause without interruption so that my attention stays on the content.

Use Case Diagram

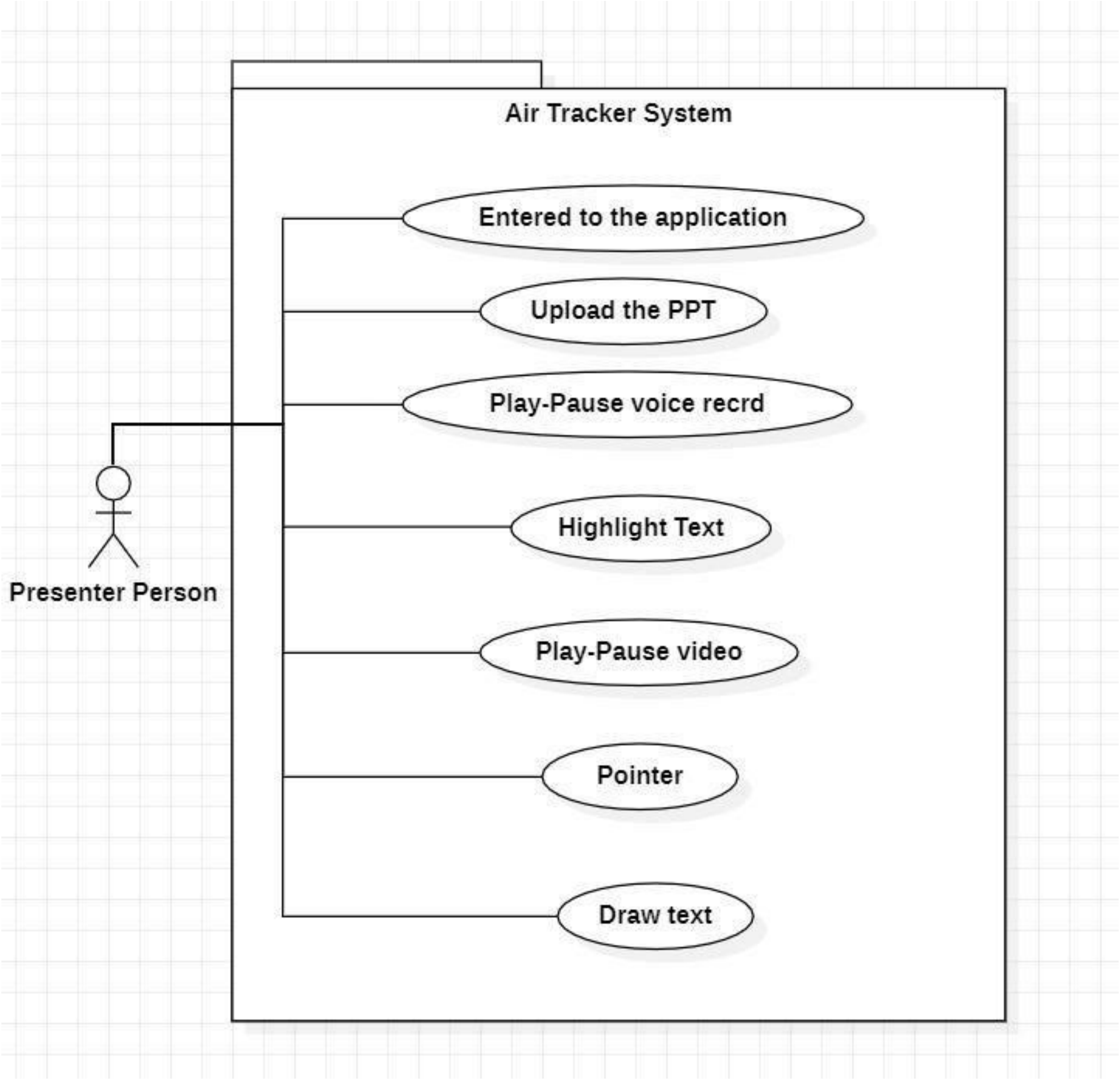


Figure 1 Use Case Diagram

System Diagram

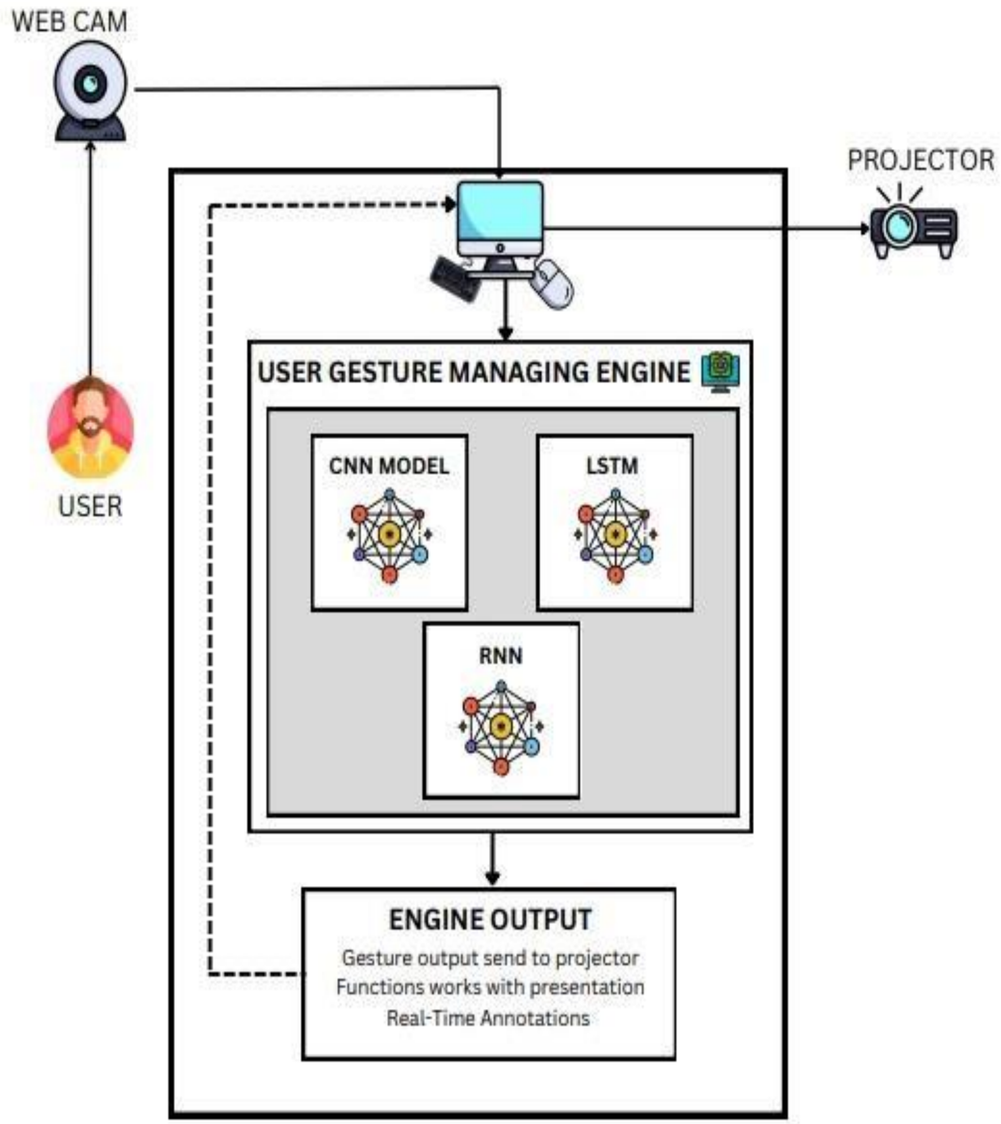

WEB CAM
PROJECTOR
USER
USER GESTURE MANAGING ENGINE
CNN MODEL
LSTM
RNN
ENGINE OUTPUT
Gesture output send to projector
Functions works with presentation
Real-Time Annotations

Figure 2 System Diagram

Activity Diagram

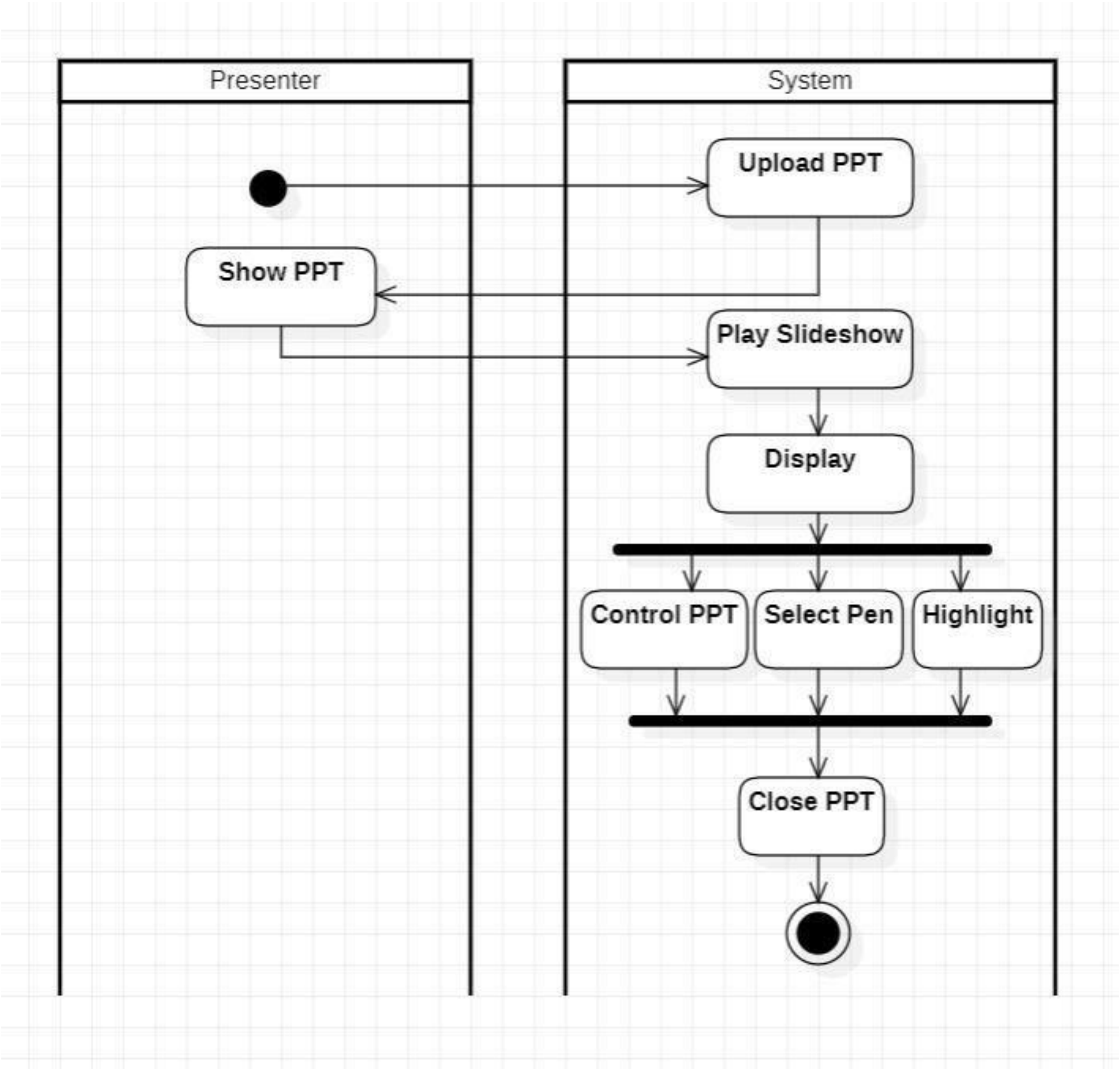


Figure 3 Activity Diagram

REFERENCES

[1] T. R. Trigo and S. R. M. Pellegrino, "An analysis of features for hand-gesture classification," in *Proc. Conf. Systems, Signals and Image Processing*, 2010. [Online]. Available: [URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4dc5906bbd6b56e658a3a96c c52553281caa052b]

[2] M. Harris and A. S. Agoes, "Applying hand gesture recognition for user guide application using MediaPipe," in *Proc. 2nd Int. Seminar Science Technol.*, Atlantis Press, 2021. [Online]. Available: [URL: https://www.atlantis-press.com/proceedings/issat-21/125963795]

[3] T. Sharp, C. Keskin, D. Robertson, and J. Taylor, "Accurate, robust, and flexible realtime hand tracking," in *Proc. 33rd Annu. ACM Conf. Human Factors Comput. Syst. (CHI '15)*, ACM,2015[Online]Available:[URL:https://dl.acm.org/doi/abs/10.1145/2702123.2702179]

[4] Author(s). "Leap Motion Technology In Learning," *Conference/Journal Name*, vol. X, no. Y, pp. Z-Z, Year. [Online]. Available: https://www.researchgate.net/publication/317153654_Leap_Motion_Technology_In_Learning

[5] P. Trigueiros, F. Ribeiro, and L. P. Reis, "Hand gesture recognition system based on computer vision and machine learning," in *Image Processing and Computational Vision*, A. Pinho and A. Tavares, Eds. Springer, 2015, pp. 277-285. [Online].Available: [URL: https://link.springer.com/chapter/10.1007/978-3-319-13407-9_21]

[6] Author(s). "Technical Features and Functionalities of Myo Armband: An Overview on Related Literature and Advanced Applications of Myoelectric Armbands Mainly Focused on Arm Prostheses," Conference/Journal Name, vol. X, no. Y, pp. Z-Z, Year. [Online]. Available: https://www.researchgate.net/publication/324889539_Technical_Features_and_Functionalities_of_Myo_Armband_An_Overview_on_Related_Literature_and_Advanced_Applications_of_Myoelectric_Armbands_Mainly_Focused_on_Arm_Prostheses

[7] C. G. Wolf, "A comparative study of gestural, keyboard, and mouse interfaces," *Behaviour & Information Technology*, vol. 11, no. 4, pp. 211-220, 1992. [Online]. Available: [URL: https://www.tandfonline.com/doi/abs/10.1080/01449299208924315]

[8] N. Mohamed, M. B. Mustafa, and N. Jomhari, "A review of the hand gesture recognition system: Current progress and future directions," *IEEE Access*, 2021. [Online]. Available: [URL: https://ieeexplore.ieee.org/document/9622242]

[9] "A convolutional neural network with feature fusion for real-time hand posture recognition," *Appl. Soft Comput. J.*, vol. 67, pp. 685-694, 2018. [Online]. Available: [URL: https://www.researchgate.net/publication/327605744_A_Convolutional_Neural_Network_with_ Feature_Fusion_for_Real-Time_Hand_Posture_Recognition]

[10] C. W. Ng and S. Ranganath, "Real-time gesture recognition system and application," *Image Vis. Comput.*, vol. 20, no. 12, pp. 993-1007, 2002. [Online]. Available: [URL:https://www.sciencedirect.com/science/article/abs/pii/S0262885602001130]

[11] "A Survey on Vision-Based Hand Gesture Recognition," in *Proc. Intell. Commun., Control Devices*, P. Sharma, G. S. Tomar, and R. Goyal, Eds. Springer, 2019, pp. 171-184. [Online].Available:[URL:https://link.springer.com/chapter/10.1007/978-3030043759_19?fromPaywallRec=true]

[12] Z. Ren, J. Meng, and J. Yuan, "Depth camera based hand gesture recognition and its applications in Human-Computer-Interaction," *IEEE Xplore*, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6173545.

[13] M. U. Rehman, F. Ahmed, M. A. Khan, U. Tariq, F. A. Alfouzan, N. M. Alzahrani, and J. Ahmad, "Dynamic Hand Gesture Recognition Using 3D-CNN and LSTM Networks," *CMC-Computers, Materials & Continua*, vol. 70, no. 3, pp. 5691–5711, 2022. [Online]. Available:[https://www.techscience.com/cmc/v70n3/44942/html.]

[14] S. Kodali, G. Divya, B. Dharani, A. V. Bhanu Prasad, and G. V. S. R. Raju, "Touch less computer interaction and user identification," *International Journal for Research in Technological Innovation*, vol. 3, no. 4, pp. 1-9, 2023. [Online]. Available: [https://www.ijrti.org/papers/IJRTI2304038.pdf.]

[15] J.-H. Sun, T.-T. Ji, S.-B. Zhang, J.-K. Yang, and G.-R. Ji, "Research on the Hand Gesture Recognition Based on Deep Learning," *IEEE Xplore*, [Online]. Available: https://ieeexplore.ieee.org/document/8634348/figures.

[16] R. Feranec, "ML Explained + Example: Arduino and Hand Gesture Recognition," *YouTube*, Nov. 12, 2020. [Online]. Available: https://www.youtube.com/watch?v=ISMcaD2RxiI.

[17] A. Eid and F. Schwenker, "Visual Static Hand Gesture Recognition Using Convolutional Neural Network," *MDPI Algorithms*, vol. 16, no. 8, p. 361, 2023. [Online]. Available: https://www.mdpi.com/1999-4893/16/8/361.

[18] P. T. Campos, "GestIA: Control your computer with your hands," *Medium*, [Online]. Available: https://medium.com/saturdays-ai/gestia-control-your-computer-with-your-hands-6bd65dba09b6.

[19] Hand Gesture Recognition Based Presentation System," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 4, 2020. [Online]. Available: https://www.ijert.org/hand-gesture-recognition-based-presentation-system.

[20] "Hand Gesture Recognition for Presentation Control," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 6, no. 4, 2024. [Online].Available: https://www.irjmets.com/uploadedfiles/paper/issue_4_april_2024/52308/final/fin_irjmets1712654490.pdf

[21] Touch-less Human-Computer Interaction using Hand Gestures," *International Journal for Research in Technological Innovation (IJRTI)*, vol. 1, no. 3, 2017. [Online]. Available: https://www.ijrti.org/papers/IJRTI1703013.pdf.

[22] The Use of Hand Gestures as a Tool for Presentation," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 11, 2023. [Online]. Available: https://thesai.org/Downloads/Volume14No11/Paper_59-The_Use_of_Hand_Gestures_as_a_Tool_for_Presentation.pdf.

[23] Y. Meng, H. Jiang, N. Duan, and H. Wen, "Real-Time Hand Gesture Monitoring Model Based on MediaPipe's Registerable System (Modelling process)," Sensors, vol. 24, no. 19, p. 6262, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/19/6262