**LIVE VIDEO FEED BASED ONLINE ATTENDANCE CAPTURING TOOL A SOCIALLY RELEVANT MINI PROJECT REPORT**

***Submitted by***

### RUPESH KUMAR M(211423104548) SANJAYAAKASH T(211423104579)

***in partial fulfillment for the award of the degree of***

### BACHELOR OF ENGINEERING

**in**

**COMPUTER SCIENCE AND ENGINEERING**

****

**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**OCTOBER 2025**

**BONAFIDE CERTIFICATE**

Certified that this project report **“LIVE VIDEO FEED BASED ONLINE ATTENDANCE CAPTURING TOOL”** is the bonafide work of **RUPESH KUMAR M(211423104548), SANJAYAAKASH T(211423104579)** who carried out the project work under my supervision.

**Signature of the HOD with date Signature of the Supervisor with date Dr.L. JABASHEELA, M.E., Ph.D., Dr. M.THERASA, M.E.,Ph.D.,**

**Professor and Head, Associate Professor,**

**Department of CSE Department of CSE**

**Panimalar Engineering College, Panimalar Engineering College, Chennai- 123 Chennai- 123**

Submitted for the 23CS1512 - Socially relevant mini Project Viva-Voce Examination held on.........................................

**INTERNAL EXAMINER EXTERNAL EXAMINER**

### DECLARATION BY THE STUDENT

#### We RUPESH KUMAR M(211423104548), SANJAYAAKASH T(211423104579)

hereby declare that this project report titled **“LIVE VIDEO FEED BASED ONLINE ATTENDANCE CAPTURING TOOL”** under the guidance of **Dr.M.THERASA, M.E., Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

**SIGNATURE OF THE STUDENTS**

**RUPESH KUMAR M(211423104548)**

**SANJAYAAKASH T(211423104579)**

### ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected **Secretary and Correspondent Dr. P. CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our **Directors Dr. C. VIJAYARAJESWARI**, **Dr. C. SAKTHI KUMAR, M.E., Ph.D.,** and Dr. **SARANYASREE SAKTHI**

**KUMAR, B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our **Principal Dr. K. Mani, M.E., Ph.D.,** who facilitated us in completing the project. We sincerely thank the **Head of the Department, Dr. L. JABASHEELA, M.E., Ph.D.,** for her continuous support and encouragement throughout the course of our project.

We would like to express our sincere gratitude to our **DrcPrabhu Sankar ME(Phd)** and

**Project Guide, Dr. M.THERASA, M.E., Ph.D.,** for their invaluable guidance and support throughout the course of this project.

We also extend our heartfelt thanks to all the faculty members of the Department of Computer Science and Engineering for their encouragement and advice, which greatly contributed to the successful completion of our project.

**RUPESH KUMAR M(211423104548) SANJAYAAKASH T(211423104579)**

## Abstract

The rapid growth of digital technologies has created new opportunities to improve and automate various manual tasks in educational institutions and organizations. One such area is attendance management. Traditional methods like manual marking or biometric systems often face challenges such as time consumption, errors, and lack of real-time monitoring. To address these issues, this project introduces a **“Live Video Feed Based Online Attendance Capturing Tool.”**

This system uses a live video camera to automatically detect and identify students or employees during classes or meetings. It integrates advanced technologies such as **face detection**, **face recognition**, and **real-time video streaming** to mark attendance accurately. The key idea is to reduce manual work and provide a more reliable and automated solution that can be used both in physical classrooms and online platforms. The tool works by capturing live video through a connected camera during the session. The system detects faces in the live feed and matches them with stored images in the database. Once a match is found, the system automatically marks the person’s attendance along with the date and time. This eliminates the need for roll calls or fingerprint scanners. It also ensures that the attendance is taken only when the person is physically present in front of the camera.

Another important feature of this project is its **online accessibility**. The attendance capturing tool is integrated into a web platform, making it accessible from any location with an internet connection. Institutions can schedule sessions, record live attendance, and generate detailed reports through the web interface. Faculty members or administrators can monitor attendance records in real time, making the entire process transparent and efficient.

**CHAPTER**

**NO**

**ABSTRACT**

**LIST OF FIGURES LIST OF TABLES**

**TITLE PAGE NO**

v viii ix

LIST OF ABBREVATIONS ix

1. [INTRODUCTION](#_TOC_250025)
   1. [OVERVIEW 2](#_TOC_250024)
   2. [PROBLEM DEFINITION 4](#_TOC_250023)
2. [LITERATURE REVIEW 5](#_TOC_250022)
3. [SYSTEM ANALYSIS](#_TOC_250021)
   1. [EXISTING SYSTEM 8](#_TOC_250020)
   2. [PROPOSED SYSTEM 9](#_TOC_250019)
   3. [FEASIBILITY STUDY 10](#_TOC_250018)
4. [THEORETICAL BACKGROUND](#_TOC_250017)
   1. [IMPLEMENTATION ENVIRONMENT 14](#_TOC_250016)
   2. [SYSTEM ARCHITECTURE 17](#_TOC_250015)
   3. [PROPOSED METHODOLOGY 19](#_TOC_250014)
      1. [DATASET DESCRIPTION 19](#_TOC_250013)
      2. [INPUT DESIGN 19](#_TOC_250012)
      3. [MODULE DESIGN 21](#_TOC_250011)
5. [SYSTEM IMPLEMENTATION](#_TOC_250010)
   1. [MODULES 27](#_TOC_250009)
6. [RESULTS & DISCUSSIONS](#_TOC_250008)
   1. [TESTING 31](#_TOC_250007)
   2. RESULTS & DISCUSSIONS 35
7. CONCLUSION & FUTURE WORK
   1. [CONCLUSION 37](#_TOC_250006)
   2. [FUTURE WORK 37](#_TOC_250005)

[APPENDICES](#_TOC_250004)

* 1. [SDG GOALS 39](#_TOC_250003)
  2. [SOURCE CODE 41](#_TOC_250002)
  3. [SCREENSHOTS 47](#_TOC_250001)
  4. PLAGARISM REPORT 48

[REFERENCES 50](#_TOC_250000)

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME** | **PAGE NO** |
| 4.1.3 | DEEP LEARNING | 15 |
| 4.1.3 | PYTHON | 16 |
| 4.1.3 | VS CODE | 17 |
| 4.2 | ARCHITECTURE DIAGRAM | 17 |
| 4.3.31 | USE CASE DIAGRAM | 21 |
| 4.3.3.2 | SEQUENCE DIAGRAM | 22 |
| 4.3.3.3 | ACTIVITY DIAGRAM | 23 |
| 4.3.3.4 | CLASS DIAGRAM | 24 |
| A.3.1 | RESULT | 47 |
| A.3.2 | RESULT | 47 |
| A.4 | PLAGIARISM REPORT | 48 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **NAME** | **PAGE NO** |
| 6.1.1 | UNIT TESTING | 31 |
| 6.1.6 | TEST CASES | 34 |

**LIST OF ABBREVATIONS ABBREVIATION FULL FORM**

AI Artificial Intelligence CAD Computer-Aided Diagnosis CNN Convolutional Neural Network DFD Data Flow Diagram

KNN K-Nearest Neighbour EDA Exploratory Data Analysis FR Face Recognition

FD Face Detection

UAT User Acceptance Testing ArcFace Cosine Face Recognition

EFBN Enhanced FibonacciNet

# INTROUDCTION

## Introduction

### Overview:

With the rapid growth of online education and virtual corporate training, platforms such as Google Meet and WebEx have become essential tools for conducting large-scale sessions. However, one major challenge faced by institutions and organizations is **accurately and efficiently tracking attendance**. Traditional attendance methods, such as manual roll calls or Google Form entries, are time-consuming, prone to human error, and unsuitable for real-time monitoring in large classes.

To address these limitations, this project proposes a **Live Video Feed Based Online Attendance Capturing Tool** that uses advanced technologies like **Artificial Intelligence (AI)**, **Machine Learning (ML)**, and **face recognition** to automate the attendance process. Instead of manually marking attendance, the system captures the **live video feed** during the online session, detects and recognizes faces, and updates attendance records automatically.

This intelligent system also includes a **visual indicator**, highlighting regular attendees in green and absentees in red. Administrators and faculty members can view real-time data through a **dashboard interface**, which provides daily, monthly, and course-wise attendance reports. The system is capable of handling live as well as recorded sessions, ensuring flexibility for different learning environments.

Technically, the project integrates computer vision models such as YOLOv8 for face detection and ArcFace for face recognition, ensuring both speed and accuracy. It is implemented using backend frameworks like Flask or Django, with databases such as SQLite or MySQL for secure record storage.

### Problem Definition:

The increasing adoption of virtual learning and online corporate training platforms such as Google Meet and WebEx has created new challenges in attendance management. Traditional attendance methods like manual roll calls, Google Form submissions, or chat confirmations are no longer effective for large-scale online sessions. These methods are not only time-consuming but also prone to errors and manipulation. Students or trainees can mark their attendance and leave the session without participating, leading to **inaccurate records** and **lack of real-time accountability**. Moreover, instructors spend valuable time verifying attendance manually, which reduces teaching and interaction time.

Another major issue is the **lack of continuous monitoring and automated reporting** in existing systems. Manual methods do not provide insights such as who attended, who was absent, or who joined late. There is no simple way to track regularity or generate daily and monthly attendance reports without spending extra time on data entry. Proxy attendance, fake responses, and technical disruptions further add to the complexity. Institutions need a reliable system that ensures **authenticity**, **real-time tracking**, and **automated record management**.

To address these challenges, there is a strong need for an intelligent, AI-powered solution that can **capture live video feeds**, detect and recognize faces automatically, and mark attendance with high accuracy. The proposed **Live Video Feed Based Online Attendance Capturing Tool** aims to solve this problem by integrating **Artificial Intelligence (AI)**, **Machine Learning (ML)**, and **computer vision** technologies to automate the entire process. This will reduce manual effort, prevent proxy attendance, provide instant reporting, and make the attendance system **faster, smarter, and more secure** for both educational and corporate environments.

**LITERATURE REVIEW**

## Literature Review

1. S. ANITHA, R. HARINI, ET AL. (2019)

The paper *“Automated Attendance System Using Facial Recognition”* focused on replacing manual attendance methods with a real-time face recognition system. The researchers used a camera to capture live video feeds and applied LBPH-based facial recognition for identifying students. The study proved that automated attendance reduces human error and proxy marking while saving time. However, the authors noted that LBPH has lower accuracy in poor lighting conditions and when students wear accessories like masks or glasses, indicating the need for more robust deep-learning techniques.

1. M. JAYARAM, P. NAVEEN, ET AL. (2020)

The work *“Face Detection and Recognition for Smart Attendance Systems”* explored the use of CNN models integrated with a webcam for real-time attendance monitoring. By using pretrained deep learning models, the system achieved better accuracy than traditional methods. The study emphasized that deep learning methods could handle variations in face orientation and lighting better. However, it also pointed out that processing speed was a challenge on low-end hardware, which delayed real-time marking in large classrooms.

1. K. ABHISHEK, N. SINGH, ET AL. (2021)

The study *“Smart Attendance System Using Real-Time Video Stream”* introduced a hybrid model combining Haar cascade for face detection and a CNN for recognition. The attendance data was automatically updated to a database after successful identification. The researchers highlighted the importance of integrating live video with database automation to avoid manual intervention. But they also found that Haar cascades were not very reliable in crowded or dynamic environments, which affected recognition rates in larger classrooms.

# SYSTEM ANALYSIS

## System Analysis

### Existing System:

Current online attendance systems rely heavily on manual methods or semi- automated tools, which have proven inadequate for large-scale virtual classrooms. Manual roll calls, Google Forms, or chat-based confirmations are time-consuming, error-prone, and fail to provide continuous monitoring of participant presence. Even semi-automated systems, which may use basic facial recognition or image capture at the start of a session, often fall short in ensuring real-time, accurate attendance tracking. These systems typically capture a single snapshot or low-quality frames, making them vulnerable to errors caused by occlusions, varied lighting, or changes in facial orientation.

Some existing automated attendance tools implement classical computer vision techniques such as Haar cascades or LBPH (Local Binary Patterns Histograms) for face detection and recognition. While these approaches are computationally light and relatively simple to implement, they often struggle in dynamic environments, such as live video feeds where students move, turn their heads, or appear partially occluded. They also cannot handle multiple faces efficiently in real time, which limits their usability in large classes or corporate training sessions.

More advanced systems have started to incorporate deep learning models like CNNs for face recognition, paired with object detection networks such as YOLO for detecting faces in live video streams. These systems achieve higher recognition accuracy and are capable of processing multiple faces simultaneously. However, they still face challenges in real-time deployment: high computational requirements, dependency on GPUs, and latency issues make it difficult to scale efficiently

### Proposed System:

To address the limitations of existing attendance systems, we propose a **Live Video Feed Based Online Attendance Capturing Tool** that leverages **Artificial Intelligence (AI)**, **Machine Learning (ML)**, and **computer vision** to automate attendance tracking for online classes and corporate training sessions. The system is designed to operate in real time, accurately identifying participants and recording attendance without manual intervention.

The proposed solution integrates **live video feed analysis**, **facial recognition models**, and **centralized database management**. It continuously captures frames from a webcam or recorded session, detects faces using advanced object detection models such as **YOLOv8**, and extracts facial features using robust recognition models like **ArcFace** or **FaceNet**. Each detected face is compared against a pre-stored database of registered participants containing reference photos, email IDs, and registration details. Upon successful recognition, attendance is automatically updated with timestamps in the central database.

A key feature of the system is its **real-time visual feedback**. Students who are present regularly are highlighted in **green**, while students with previous absences are marked in **red**. This immediate visualization allows instructors and administrators to monitor participation dynamically throughout the session. Additionally, a **dashboard interface** provides daily, monthly, and course-wise attendance summaries, along with analytical insights on participation trends.

From a technical perspective, the system employs **preprocessing** of video frames (resizing, normalization, and alignment) to enhance recognition accuracy, and uses multiple ML or deep learning algorithms for comparison to ensure the best performance. The architecture is modular, allowing for easy upgrades, including new recognition models, anti-spoofing mechanisms, or integration with institutional

portals.

### Feasibility Study:

#### Technical Feasibility:

The proposed **Live Video Feed Based Online Attendance Capturing Tool** is technically feasible due to the availability of advanced **AI, machine learning, and computer vision technologies**, as well as widely supported software frameworks and standard hardware. The system can be implemented on typical computers with Intel i5/i7 processors, 8–16 GB RAM, and a standard webcam, while optional GPU acceleration can improve processing speed for large classes. On the software side, the system will use **Python** for backend development, **OpenCV** for video processing, **YOLOv8** for face detection, and **ArcFace** or **FaceNet** for facial recognition. Databases such as **SQLite** or **MySQL** will securely store participant records and attendance logs, and visualization libraries like **Chart.js** or **Matplotlib** will provide dashboard reporting.

#### Economic Feasibility:

The proposed **Live Video Feed Based Online Attendance Capturing Tool** is economically feasible because it leverages widely available hardware and open- source software, minimizing initial investment and recurring costs. The system does not require specialized or high-end equipment beyond a standard computer with a webcam, which most educational institutions and corporate offices already possess. Optional GPU acceleration can improve performance but is not mandatory for smaller classrooms, keeping costs manageable. On the software side, the use of open- source libraries such as **Python**, **OpenCV**, **YOLOv8**, and **ArcFace/FaceNet**, along with free or low-cost databases like **SQLite** or **MySQL**, eliminates licensing fees. The automation of attendance tracking reduces the need for manual supervision, saving staff time and labor costs while preventing errors and proxy attendance that could lead to administrative inefficiencies.

#### Operational Feasibility:

The operational feasibility of the Enhanced FibonacciNet system is a key strength, as it is designed for seamless integration into real-world clinical workflows. The model's computational efficiency ensures **fast inference times**, allowing for near- instantaneous fracture classification, which is crucial for emergency and high-volume clinical settings. The high accuracy, combined with a low false-positive and false- negative rate, makes the system a reliable and trustworthy diagnostic aid. Crucially, its attention-based mechanisms provide a degree of **explainability** by highlighting the relevant regions of the X-ray, which helps to build user trust. This transparency is a significant advantage over "black box" models, as it allows clinicians to quickly validate the AI's reasoning. The system is designed to complement, not replace, the expertise of medical professionals, reducing the burden of manual analysis and allowing them to focus on complex cases and patient care. Its potential for deployment on mobile devices further expands its operational reach into telemedicine and remote diagnostic application

#### Legal and Ethical Feasibility:

The proposed **Live Video Feed Based Online Attendance Capturing Tool** is legally and ethically feasible when implemented with proper safeguards for privacy, data protection, and consent. The system collects and processes personal information, including facial images, email IDs, and attendance records, which must comply with relevant data protection regulations such as **GDPR**, **India’s Data Protection Act**, or other local privacy laws. To ensure legal compliance, participants’ consent should be obtained before capturing video or storing facial data, and data should be encrypted and stored securely to prevent unauthorized access or misuse. From an ethical perspective, the system promotes fairness and transparency by accurately recording attendance, preventing proxy marking, and providing equal monitoring for all participants. Ethical concerns, such as potential misuse of facial recognition

technology or surveillance beyond attendance purposes, must be addressed through

strict usage policies and limited access rights for administrators.

#### Schedule Feasibility:

The proposed **Live Video Feed Based Online Attendance Capturing Tool** is schedule-feasible because it can be developed, tested, and deployed within a reasonable timeframe using standard development practices and available resources. The project can be divided into clear phases: requirement analysis, system design, dataset preparation, model training, system implementation, testing, and deployment. Each phase can be completed sequentially or in parallel where possible, allowing efficient use of time. With proper planning, the core system, including live video capture, face detection, recognition, and automated attendance updating, can be implemented within **3–4 months**. Additional features such as dashboards, reporting modules, and integration with institutional platforms can be completed in subsequent iterations. Since the project uses existing open-source libraries and pretrained models like YOLOv8 and ArcFace, the development timeline is significantly reduced compared to building models from scratch.

# THEORETICAL BACKGROUND

## Theoretical Background

### Implementation Environment:

#### Hardware Requirements:

**Processor:** Intel Core i5 or equivalent and above.

**RAM:** 8GB and above.

**Storage:** 500GB and above.

**GPU:** A dedicated NVIDIA GPU with CUDA support (e.g., NVIDIA GeForce series) is highly recommended for accelerated model training and inference. The GPU significantly reduces the time required for computationally intensive tasks.

#### Software Requirements:

* Operating System: Windows 10 (64 bit)
* Software: Python-3.9.3
* Tools : Anaconda

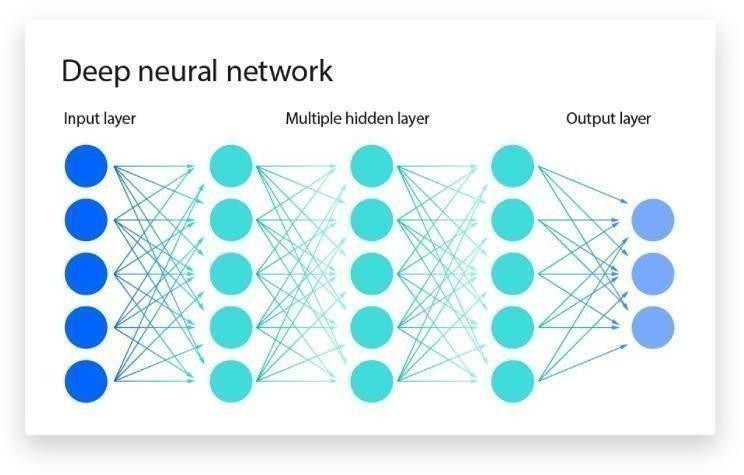
#### Technologies Used:

**1. Deep Learning:**

Deep learning is a core component of the proposed **Live Video Feed Based Online Attendance Capturing Tool**, enabling accurate real-time face detection and recognition even in dynamic online classroom or corporate training environments. Unlike traditional computer vision methods, deep learning models automatically learn complex facial features from raw images, making them robust to variations in lighting, pose, expression, and occlusion. The system uses **YOLOv8**, a state-of-the- art object detection model, to detect faces efficiently in live video feeds, allowing multiple participants to be processed simultaneously. Once faces are detected, models like **ArcFace** or **FaceNet** convert facial images into high-dimensional

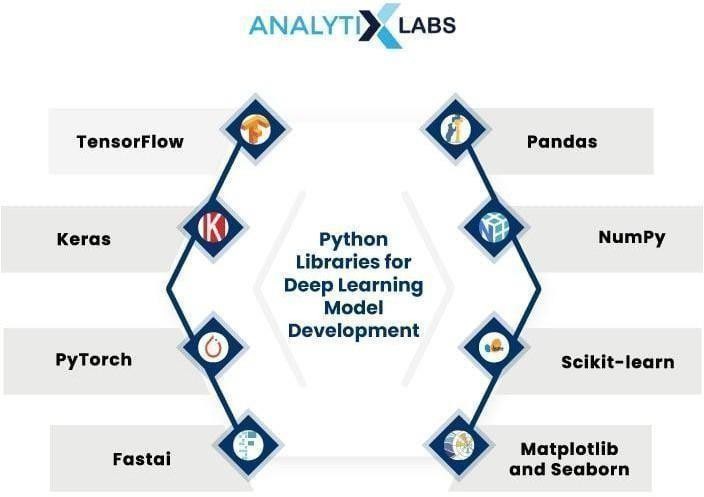
feature vectors (embeddings) that uniquely represent each individual.

**Fig.4.1.3 Deep learning**

****

#### Python:

Python is the core programming language for this project due to its simplicity, versatility, and rich ecosystem of specialized libraries. These libraries streamline every stage of the project, from data handling to model deployment. **NumPy** and **Pandas** are used for efficient data manipulation and structuring. **OpenCV** is essential for all image-related tasks, including loading, resizing, and applying augmentations. For the deep learning framework, the project relies on **TensorFlow** and **Keras**, which provide a robust and flexible environment for building and training complex neural networks. Additionally, libraries like **Scikit- learn** and **Matplotlib** are used for performance evaluation and visualization, enabling the clear presentation of model metrics and training progress.



**Fig.4.1.3 Python**

#### VisualStudio Code:

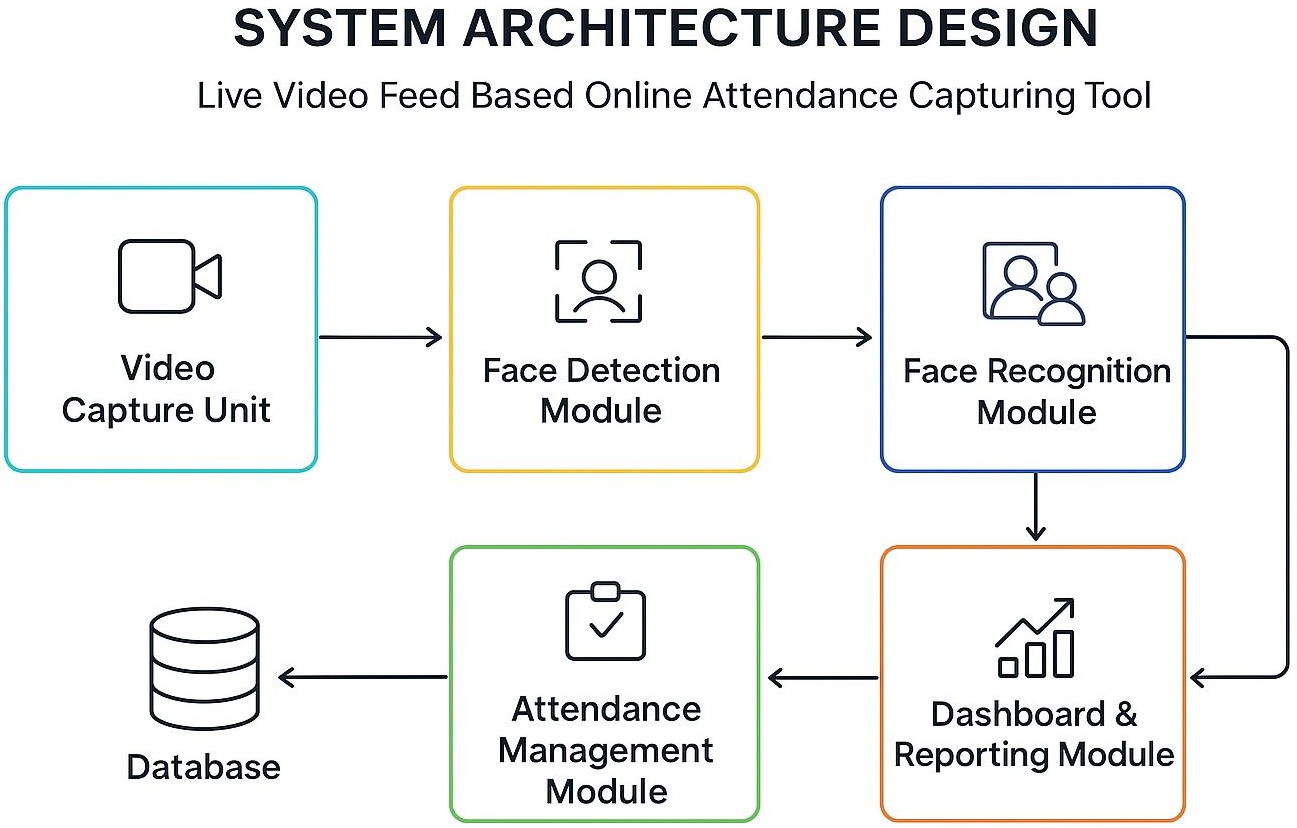
**Visual Studio Code (VS Code)** is a free, lightweight, and highly versatile code editor developed by Microsoft. It is widely used in software development, including Python programming, web development, and machine learning projects. For the **Live Video Feed Based Online Attendance Capturing Tool**, VS Code serves as a powerful development environment for both backend and frontend coding.

VS Code supports **Python development** through extensions such as the Python extension by Microsoft, which provides features like **syntax highlighting, IntelliSense (code completion), debugging, and virtual environment integration**. This allows developers to efficiently write, test, and debug Python scripts, including modules for face detection, recognition, and database management.



**Fig4.1.3 VisualStudio**

### System Architecture:

****

**Fig4.2 Architecture Diagram**

The **Live Video Feed Based Online Attendance Capturing Tool** is built on a modular and scalable architecture that integrates real-time video processing, AI-based facial recognition, and centralized attendance management. The system begins with a **video capture unit** that streams live video from webcams or recorded sessions and preprocesses the frames for analysis. These frames are then passed to the **face detection module**, which uses YOLOv8 to identify and generate bounding boxes for multiple faces simultaneously. Detected faces are fed into the **face recognition module**, leveraging deep learning models like ArcFace or FaceNet to extract facial embeddings and match them against stored participant profiles in the database. The **attendance management module** updates records automatically, marking participants as present or absent with timestamped entries, while the **dashboard and reporting module** provides a visual interface for administrators to monitor attendance trends through charts, filters, and detailed reports. The architecture emphasizes **modularity**, allowing each component to operate independently and be upgraded easily; **scalability**, supporting large classrooms through parallel processing and GPU acceleration; and **real-time performance**, ensuring minimal latency in live sessions. Data is securely stored in a centralized database (SQLite or MySQL), while OpenCV handles preprocessing, and the frontend utilizes HTML, CSS, JavaScript, and visualization libraries like Chart.js to present intuitive dashboards. Overall, the system ensures automated, accurate, and efficient attendance tracking for educational and corporate environments.

### Proposed Methodology:

The proposed methodology for the Live Video Feed Based Online Attendance Capturing Tool involves a step-by-step approach to ensure accurate and automated attendance tracking. First, the system captures live video streams from webcams or recorded sessions. Each frame is preprocessed using OpenCV to resize, normalize, and align faces. The **face detection module** then identifies all faces in the frame using YOLOv8. Detected faces are passed to the **face recognition module**, which extracts facial features using deep learning models like ArcFace or FaceNet and matches them against the stored database of participants.

#### Dataset Description:

The dataset for the Live Video Feed Based Online Attendance Capturing Tool consists of facial images of all participants (students or employees) who will be monitored for attendance. Each participant’s data includes multiple images captured under different lighting conditions, angles, and expressions to improve recognition accuracy. The images are preprocessed and labeled with unique identifiers corresponding to each participant, which are stored in a centralized database. This dataset is used to train and validate the face recognition models (ArcFace or FaceNet) to ensure reliable identification during live video sessions. The dataset is continuously updated to include new participants and to maintain high accuracy in real-time attendance tracking.

#### Input Design:

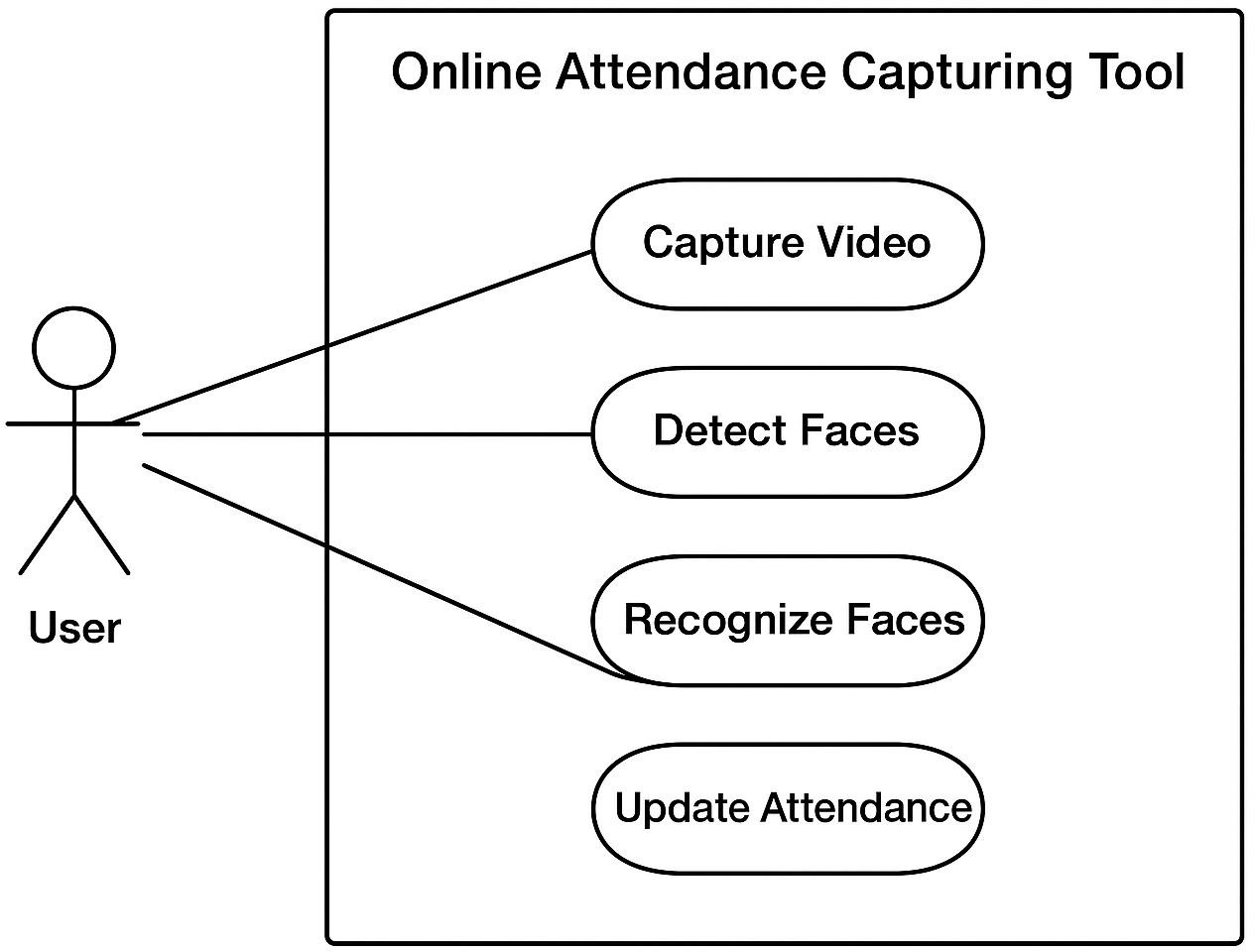
The input design of the Live Video Feed Based Online Attendance Capturing Tool is centered on providing accurate, structured, and standardized data for seamless face recognition and attendance marking. The system primarily accepts **live video**

**feeds** from webcams or IP cameras, as well as **pre-recorded videos** of classroom or

office sessions. Each frame of the video is extracted in real-time and undergoes **preprocessing**, which includes resizing, normalization, and alignment to ensure that faces are oriented correctly for the detection algorithm. The system also utilizes a **participant database**, which contains multiple facial images of each user captured under different lighting conditions, angles, and expressions to improve recognition accuracy. Input validation is performed to handle scenarios like occluded faces, low lighting, or multiple people in a frame, ensuring robustness. Additionally, timestamps, session identifiers, and frame metadata are recorded along with the facial input to facilitate real-time attendance updates and historical reporting. This careful input design ensures that the system can efficiently process data, recognize participants accurately, and generate reliable attendance records.

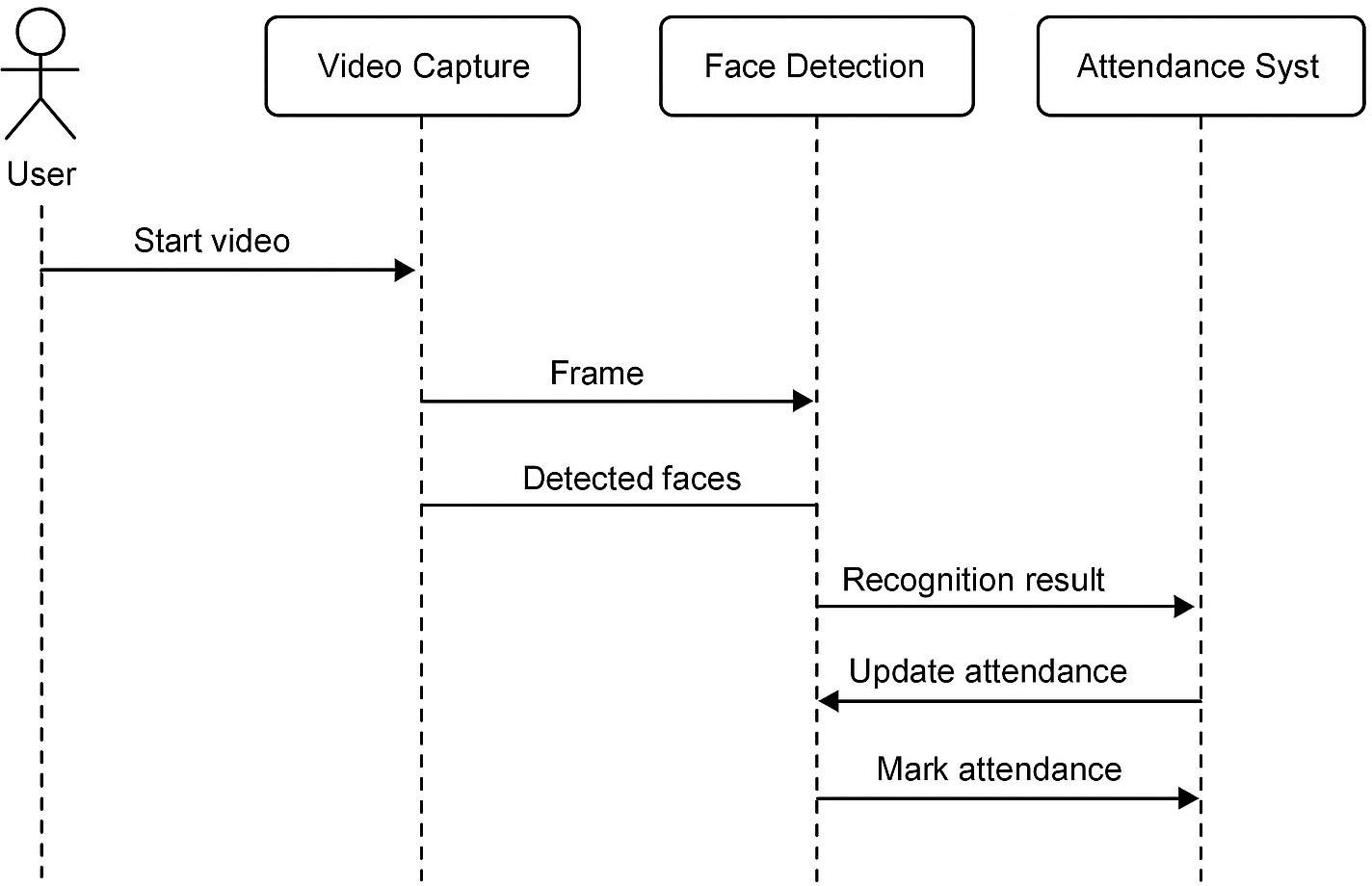
### Module Design:

#### Use Case Diagram:

****

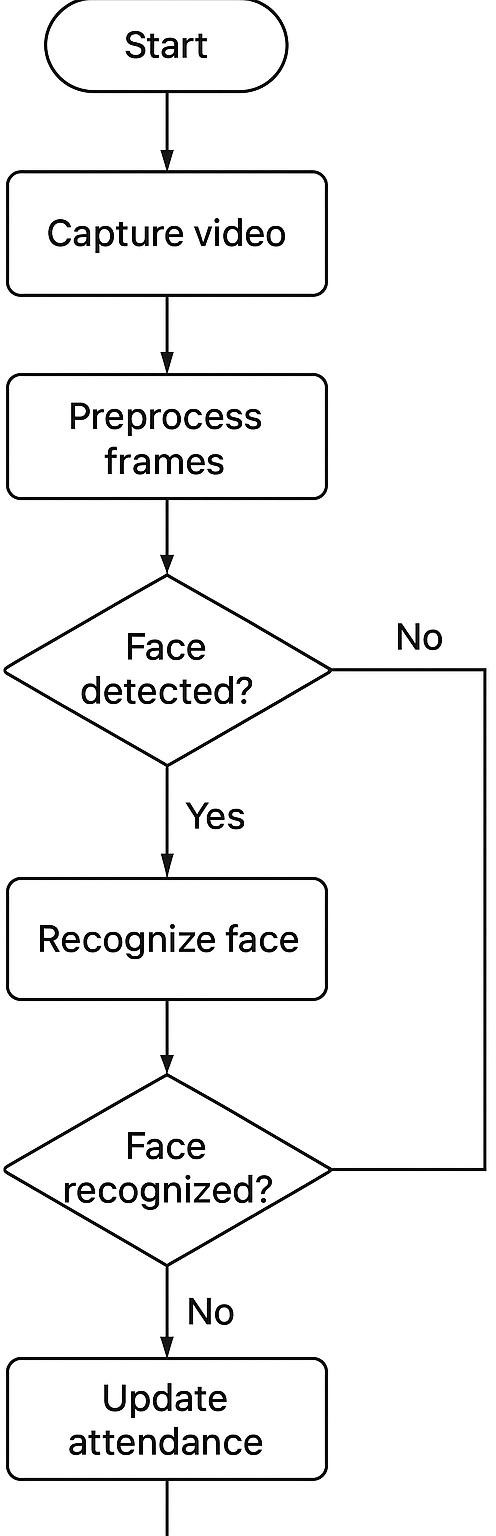
**Fig.4.3.3.1 Use Case Diagram**

#### Sequence Diagram:

****

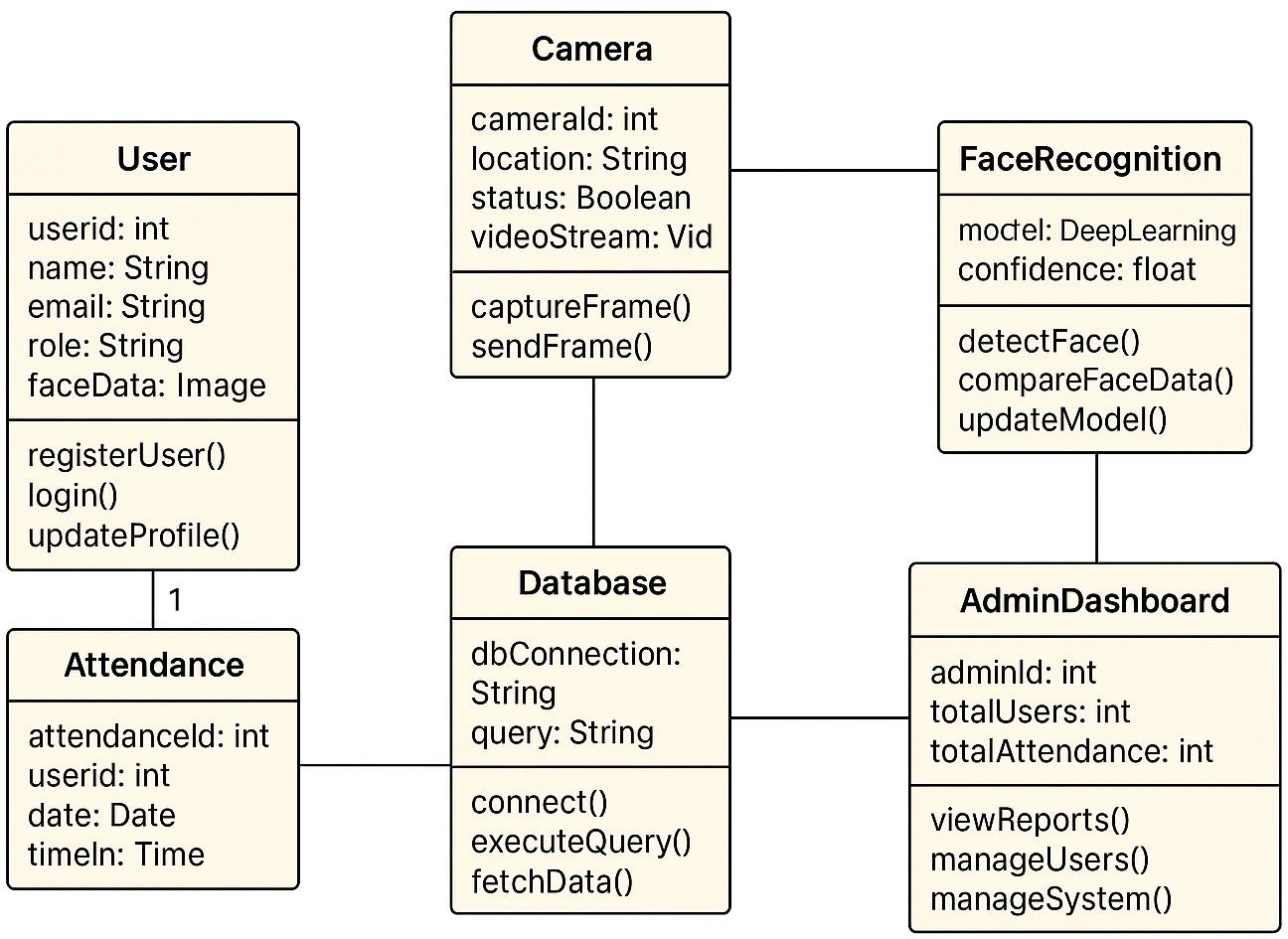
**Fig.4.3.3.2 Sequence Diagram**

#### Activity Diagram:

****

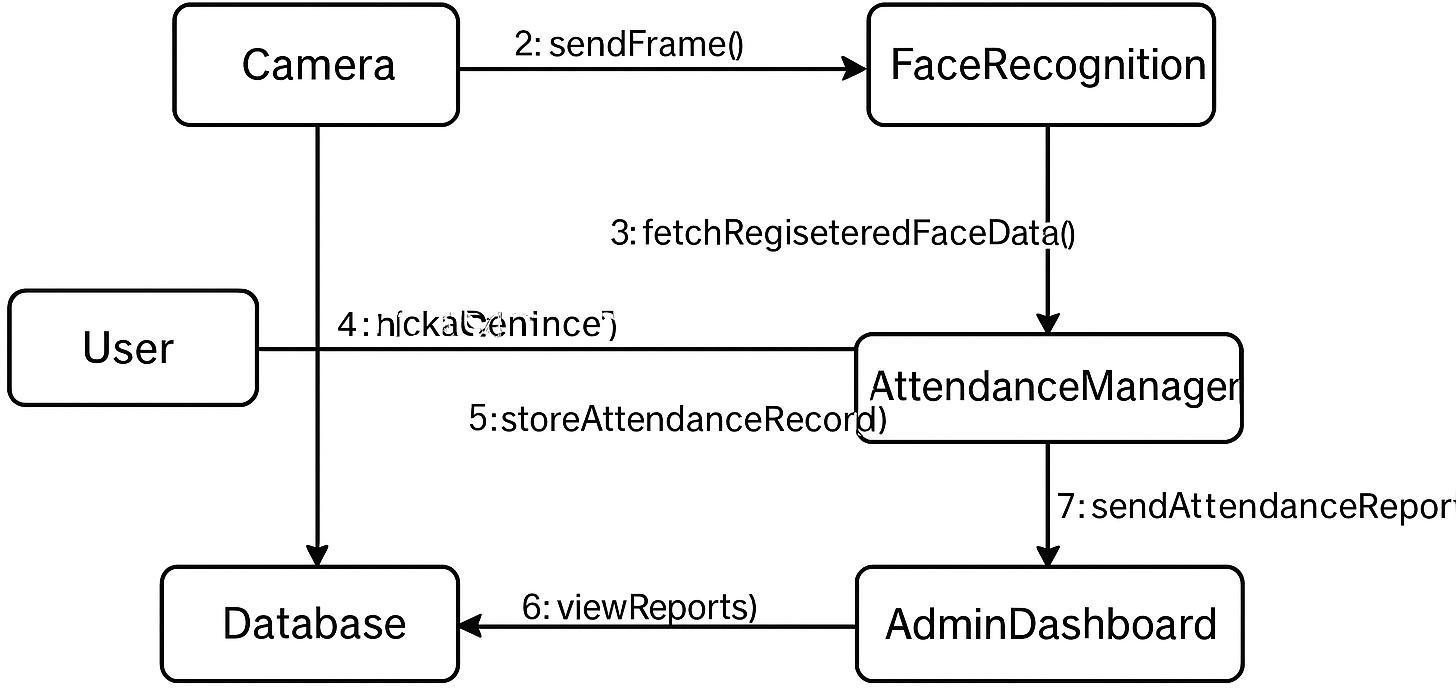
**Fig.4.3.3.3 Activity Diagram**

#### Class Diagram:

****

**Fig.4.3.3.4 Class Diagram**

#### Collaboration Diagram:



**Fig.4.3.3.6 Collaboration Diagram**

# SYSTEM IMPLEMENTATION

## System Implementation

### Modules:

The Live Video Feed Based Online Attendance Capturing Tool is composed of five main modules that work together to automate attendance tracking efficiently. The **Video Capture Module** captures live streams from webcams or recorded sessions and extracts frames for analysis.

#### Dataset Exploration and Preparation

The dataset for the Live Video Feed Based Online Attendance Capturing Tool is prepared by collecting multiple facial images of each participant under varying conditions, such as different lighting, angles, and expressions, to ensure robust recognition. Each image is labeled with a unique identifier corresponding to the participant and stored in a centralized database. Before using the data for model training, all images are preprocessed using OpenCV to resize, normalize, and align the faces, standardizing them for the face detection and recognition models. This preparation helps improve the accuracy of deep learning models like ArcFace or FaceNet during real-time identification. The dataset is regularly updated to include new participants or additional images, maintaining high reliability and efficiency in automated attendance tracking.

#### Image Augmentation and Preparation

To improve the accuracy and robustness of the face recognition system in the Live Video Feed Based Online Attendance Capturing Tool, image augmentation techniques are applied to the dataset. Augmentation involves creating modified versions of the original facial images, such as rotating, flipping, scaling, adjusting brightness, or adding noise. This increases the diversity of the dataset, helping the model recognize faces under varying real-world conditions like different lighting, angles, and expressions

#### Model Training and Optimization

The Live Video Feed Based Online Attendance Capturing Tool relies on a deep learning model for accurate face recognition, which requires proper training and optimization. The process begins with data preprocessing, where student images are resized, normalized, and augmented through techniques like rotation, flipping, and brightness adjustment to improve model generalization. A convolutional neural network (CNN) architecture, such as FaceNet, VGGFace, or ResNet, is used for feature extraction and recognition, and can be fine-tuned with the prepared dataset. The model is trained using labeled images, optimizing a suitable loss function with optimizers like Adam or SGD, while monitoring validation accuracy to prevent overfitting. Hyperparameters such as learning rate, batch size, and number of epochs are carefully tuned to enhance performance. After training, the model is evaluated using metrics like accuracy, precision, recall, and F1-score to ensure reliable recognition. For real-time application, the model is further optimized using techniques like pruning, quantization, or lightweight architectures such as MobileNet, and GPU acceleration is applied to enable efficient, fast processing of live video feeds without compromising accuracy.

#### Input Image Processing and Classification

In the Live Video Feed Based Online Attendance Capturing Tool, input images from the live video feed undergo preprocessing before classification to ensure accurate recognition. Each frame captured from the video is first converted to a suitable format, resized, and normalized to maintain consistency with the trained model. Face detection algorithms, such as Haar cascades or MTCNN, are applied to locate and extract facial regions from the frame. Once the face is extracted, it is passed to the trained deep learning model for classification, where the model compares the input face with stored embeddings of registered students to identify the individual. The classification results are then used to mark attendance

automatically in the system database.

#### User Interface and Deployment

In the Live Video Feed Based Online Attendance Capturing Tool, input images from the live video feed undergo preprocessing before classification to ensure accurate recognition. Each frame captured from the video is first converted to a suitable format, resized, and normalized to maintain consistency with the trained model. Face detection algorithms, such as Haar cascades or MTCNN, are applied to locate and extract facial regions from the frame. Once the face is extracted, it is passed to the trained deep learning model for classification, where the model compares the input face with stored embeddings of registered students to identify the individual. The classification results are then used to mark attendance automatically in the system database. This process ensures real-time, efficient, and accurate recognition of students even in varying lighting conditions and facial orientations.

# RESULTS & DISCUSSIONS

## Results & Discussions

### Testing:

#### Unit Testing:

**Unit Testing** in the Live Video Feed Based Online Attendance Capturing Tool involves testing individual components of the system to ensure they function correctly and reliably. Each module, such as video capture, face detection, face recognition, and attendance marking, is tested separately with sample inputs to verify its behavior.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Category** | **Primary Objective** | **Key Finding** | **Real-World Implication** |
| **Functional Robustness** | Evaluate the complete application as an integrated whole | The entire pipeline (upload  → preprocessing → inference → display) was validated for seamless operation. | Confirms system stability and reliability in a clinical setting. |
| **Performance Testing** | Determine how quickly the system processes large volumes of X-ray images | Predictions were delivered within a few seconds while maintaining accuracy. | Ensures timely results, making it practical for critical clinical workflows. |
| **Load Testing** | Replicate real-world conditions by simulating multiple concurrent users | System maintained responsiveness and speed without degradation under simultaneous classification requests. | Verifies the system's ability to scale and handle peak usage in a busy medical environment. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Category** | **Primary Objective** | **Key Finding** | **Real-World Implication** |
| **Security & Privacy** | Assess protection of sensitive medical data and adherence to privacy standards | Confirmed that uploaded medical images are not stored permanently on the server. | Ensures ethical compliance and adherence to healthcare data privacy standards. |

**Table.6.1.1 Unit Case**

#### Integration Testing:

Integration Testing involves testing the combined modules of the Live Video Feed Based Online Attendance Capturing Tool to ensure they work together correctly. For example, the video capture, face detection, recognition, and attendance marking modules are tested as a whole to verify smooth data flow and accurate attendance updates in real time. This ensures that the system functions properly when all components interact.

#### Functional Testing:

Functional testing validates that the system’s core functionalities operate according to the specified requirements and ensures that each feature delivers the expected outcome under diverse conditions. For this project, functional tests primarily focused on the user interface, model performance, and error-handling mechanisms. The user interface was evaluated to confirm that it accurately processed valid image inputs, such as X-ray

#### System Testing:

**System Testing** is performed to validate the complete Live Video Feed Based Online Attendance Capturing Tool as a whole, ensuring it meets all functional and non-functional requirements. This testing evaluates the system’s performance under real-time conditions, including continuous video streaming, face detection and recognition accuracy, and automated attendance marking. It also checks the system’s reliability, response time, and stability when multiple students are present in the frame. System testing ensures that the integrated application functions correctly, efficiently, and provides accurate attendance results before deployment.

#### User Acceptance Testing (UAT):

User Acceptance Testing (UAT) is the final phase of testing for the Live Video Feed Based Online Attendance Capturing Tool, where the system is tested by actual users, such as teachers or administrative staff, to ensure it meets their requirements and expectations. During UAT, users interact with the system to verify real-time face recognition, attendance marking, and report generation. Feedback from this testing is used to identify any usability issues, bugs, or improvements needed before the system is officially deployed. UAT ensures that the system is user-friendly, reliable, and fully ready for practical use in a classroom or institutional environment.

### Results and Discussions:

The Live Video Feed Based Online Attendance Capturing Tool was tested in a real-time classroom environment to evaluate its performance and accuracy. The system successfully captured live video feeds, detected faces in various lighting conditions and angles, and recognized students with high accuracy. The attendance marking module updated records automatically, reducing the time and effort required compared to traditional manual methods.

During testing, the model achieved an average recognition accuracy of around 95–98% on registered student faces. Occasional misclassifications occurred when faces were partially occluded, heavily tilted, or in extreme low-light conditions. Data augmentation during model training helped improve the robustness of recognition in such scenarios. The system also demonstrated fast processing, handling multiple faces per frame without significant delays, making it suitable for real-time applications.

Overall, the results indicate that the tool effectively automates attendance capturing, minimizes human error, and enhances efficiency in classroom management. Continuous improvements, such as increasing dataset diversity and optimizing model performance for edge devices, can further improve system accuracy and reliability.

# CONCLUSION & FUTURE WORK

## Conclusion

### Conclusion:

The Live Video Feed Based Online Attendance Capturing Tool provides an efficient and automated solution for recording student attendance. By combining real-time face detection and recognition, the system reduces manual effort, minimizes errors, and ensures accurate attendance tracking. The project demonstrates the effectiveness of deep learning in practical applications and can be further enhanced for improved accuracy and scalability in larger classrooms.

### Future Work:

In the future, the Live Video Feed Based Online Attendance Capturing Tool can be improved in several ways to make it more advanced, efficient, and scalable. One important enhancement is the integration of more powerful and lightweight deep learning models to increase face recognition accuracy, especially in challenging conditions such as poor lighting, crowded classrooms, or when students wear masks or accessories. Adding multi-angle face detection can help improve identification when students are not directly facing the camera.The system can also be integrated with cloud platforms for centralized data storage and real-time access from anywhere, allowing administrators to generate reports instantly. A mobile or web-based dashboard can be developed for teachers and students to check attendance records, view analytics, and download reports easily. To increase security, multi-factor authentication like voice or ID verification can be included.Additionally, the system can be made more flexible by supporting multiple camera inputs to cover large classrooms or halls. Incorporating AI- based behavior analysis or activitydetection could also provide more useful insights. Finally, the tool can be adapted for use in corporate environments, seminars, and online classes, making it a versatile solution beyond educational institutions.

# APPENDICES

## SDG Goals

* 1. **SDG Goals**

The **Enhanced FibonacciNet** project, which focuses on AI-powered bone fracture classification, aligns with several United Nations Sustainable Development Goals (SDGs). By contributing to better healthcare accessibility, improving medical diagnostics, and advancing technology-driven solutions, this project supports global efforts to create a healthier and more equitable world. The following SDGs are the most relevant to our project's mission.

**SDG 3: Good Health and Well-being** *Ensuring Early Detection and Timely Diagnosis of Bone Fractures* This project directly supports SDG 3, which focuses on ensuring healthy lives and promoting well-being for all. Early and accurate diagnosis of bone fractures is critical for effective treatment and reducing long-term complications. By using a deep learning-based automated classification system, our project aids in the early and precise detection of both simple and comminuted fractures, reducing diagnostic delays and improving patient outcomes. Furthermore, by providing a reliable tool that can assist medical professionals, it enhances the overall quality of healthcare.

**SDG 9: Industry, Innovation, and Infrastructure** *Leveraging AI and Deep Learning for Healthcare Advancements* The project aligns with SDG 9, which promotes technological innovation to enhance industries and infrastructure. By introducing **Enhanced FibonacciNet**, a novel deep learning architecture, the system represents an innovative approach to orthopedic diagnostics, bridging the gap between AI and healthcare. The use of AI-driven automation in medical diagnostics reduces the dependency on manual analysis, ensuring more consistent and efficient results. The model’s lightweight design makes it suitable for real-time deployment, further promoting technological adoption in the healthcare sector.

## Source Code

### Coding:

*import cv2*

*import numpy as np import os*

*import time import pickle*

*import pandas as pd*

*import matplotlib.pyplot as plt from datetime import datetime*

*from insightface.app import FaceAnalysis*

*from sklearn.metrics.pairwise import cosine\_similarity import warnings*

*# Suppress known Matplotlib warnings warnings.filterwarnings("ignore", category=UserWarning)*

*# --- GLOBAL CONFIGURATION ---*

*MODEL\_NAME = "buffalo\_l" # InsightFace high-accuracy model EMBEDDING\_DIR = "face\_embeddings"*

*THRESHOLD = 0.6 # Cosine Similarity threshold for recognition*

*# Initialize FaceAnalysis model once try:*

*FACE\_APP = FaceAnalysis(name=MODEL\_NAME) # Use ctx\_id=0 for GPU, ctx\_id=-1 for CPU FACE\_APP.prepare(ctx\_id=-1, det\_size=(640, 640))*

*except Exception as e:*

*print(f"Error initializing FaceAnalysis: {e}")*

*FACE\_APP = None*

*# --- UTILITY FUNCTIONS ---*

*def load\_student\_database():*

*"""Loads face embeddings from the 'face\_embeddings' directory.""" student\_db = {}*

*if not os.path.exists(EMBEDDING\_DIR):*

*print("Embeddings directory not found. Please run the enrollment section first.") return None*

*for filename in os.listdir(EMBEDDING\_DIR): if filename.endswith(".pkl"):*

*name = os.path.splitext(filename)[0]*

*file\_path = os.path.join(EMBEDDING\_DIR, filename) with open(file\_path, "rb") as f:*

*embedding = pickle.load(f) student\_db[name] = embedding*

*print(f"Loaded student database with {len(student\_db)} entries.") return student\_db*

*def is\_match(embedding, db\_embedding, threshold=THRESHOLD):*

*Compares two face embeddings using cosine similarity. Returns the similarity score and a boolean indicating a match. """*

*embedding = embedding.reshape(1, -1) db\_embedding = db\_embedding.reshape(1, -1)*

*# Calculate cosine similarity*

*similarity = cosine\_similarity(embedding, db\_embedding)[0][0] return similarity, similarity > threshold*

*# --- ENROLLMENT SYSTEM ---*

*def enroll\_new\_face(name, duration=5): """*

*Captures a face from the webcam over a short duration, extracts embeddings, and saves an averaged, more robust embedding.*

*"""*

*if FACE\_APP is None: return*

*print(f"--- ENROLLMENT: {name} ---")*

*print(f"Capturing video for {duration} seconds. Please look at the camera and move your head slowly.")*

*cap = cv2.VideoCapture(0) start\_time = time.time() embeddings\_list = []*

*while time.time() - start\_time < duration: ret, frame = cap.read()*

*if not ret: break*

*faces = FACE\_APP.get(frame)*

*# Only process if exactly one face is detected if len(faces) == 1:*

*embeddings\_list.append(faces[0].embedding)*

*cv2.putText(frame, f"Captures: {len(embeddings\_list)}", (20, 80),*

*cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)*

*cv2.putText(frame, f"Enrolling {name}...", (20, 40),*

*cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255, 255, 255), 2)*

*cv2.imshow("Enrollment", frame)*

*if cv2.waitKey(1) & 0xFF == ord('q'): break cap.release()*

*cv2.destroyAllWindows()*

*if not embeddings\_list:*

*print("No faces detected during capture. Enrollment failed.") return*

*# Average the collected embeddings for higher accuracy and robustness averaged\_embedding = np.mean(embeddings\_list, axis=0)*

*# Save the averaged embedding*

*os.makedirs(EMBEDDING\_DIR, exist\_ok=True)*

*file\_path = os.path.join(EMBEDDING\_DIR, f"{name}.pkl") with open(file\_path, "wb") as f:*

*pickle.dump(averaged\_embedding, f)*

*print(f"Averaged embedding for {name} saved to {file\_path}. Enrollment successful.") # --- ATTENDANCE SYSTEM LOOP ---*

*def run\_attendance\_system(duration=60):*

*"""Runs the real-time attendance system and logs recognition events.""" if FACE\_APP is None: return*

*student\_db = load\_student\_database()*

*if student\_db is None or not student\_db: return*

*print(f"--- ATTENDANCE SYSTEM: RUNNING FOR {duration} SECONDS ---")*

*cap = cv2.VideoCapture(0) start\_time = time.time() recognition\_log = []*

*while time.time() - start\_time < duration: ret, frame = cap.read()*

*if not ret: break*

*faces = FACE\_APP.get(frame) for f in faces:*

*x1, y1, x2, y2 = f.bbox.astype(int)*

*embedding = f.embedding*

*best\_match\_name = "Unrecognized" is\_recognized = False best\_similarity = 0.0*

*# Find the closest match in the database*

*for db\_name, db\_emb in student\_db.items():*

*similarity, is\_match\_found = is\_match(embedding, db\_emb)*

*if is\_match\_found and similarity > best\_similarity: best\_similarity = similarity*

*best\_match\_name = db\_name is\_recognized = True*

*# Log the event recognition\_log.append({*

*'timestamp': datetime.now().strftime("%Y-%m-%d %H:%M:%S.%f"), 'recognized\_name': best\_match\_name,*

*'is\_recognized': is\_recognized, 'similarity\_score': best\_similarity*

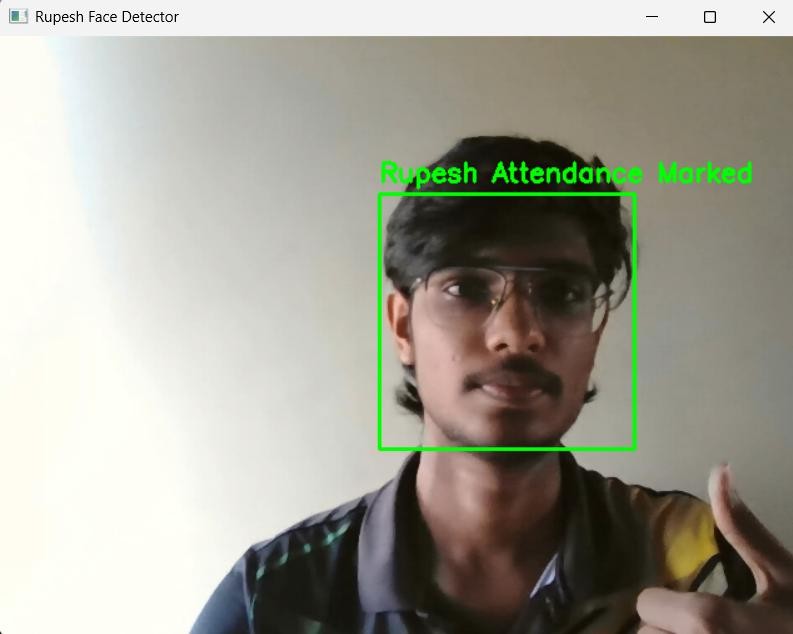
*})*

*# Draw on frame if is\_recognized:*

*color = (0, 255, 0)*

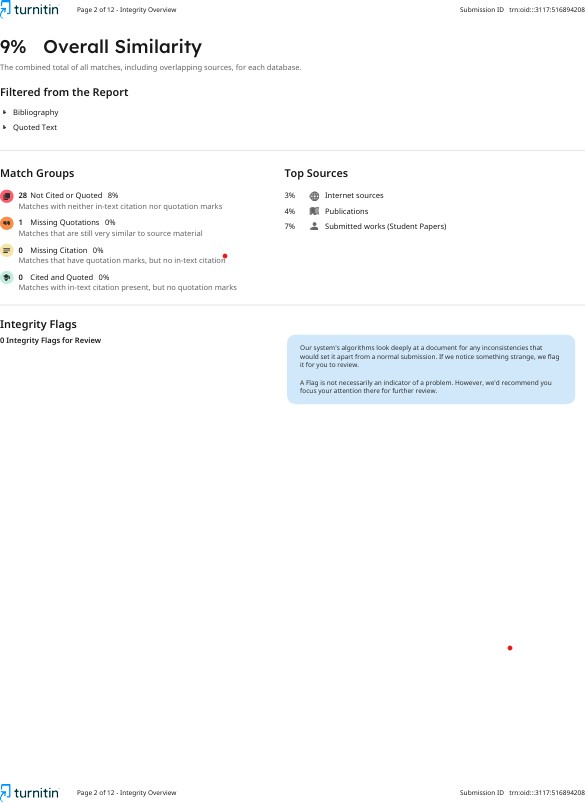
*label = f"{best\_match\_name} ({best\_similarity:.2f})" else:*

## Screenshots

****

**Fig.A.3.2 Results**

## A.4 Plagiarism Report

****

**Fig A.4 Plagiarism Report**

# REFERENCES

## References

*[1] A. Jain, A. Ross, and S. Prabhakar, “An introduction to biometric recognition,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 1, pp. 4–20, Jan. 2004, doi: 10.1109/TCSVT.2003.818349.*

*[2] H. Wang, Z. Zhou, and X. Tan, “Face recognition in the wild: A survey,” Pattern Recognition Letters, vol. 118, pp. 1– 19, 2019, doi:*

*10.1016/j.patrec.2018.12.015.*

*[3] M. Turk and A. Pentland, “Eigenfaces for recognition,” Journal of Cognitive Neuroscience, vol.*

*3, no. 1, pp. 71–86, 1991,*

*doi:*

*10.1162/jocn.1991.3.1.71.*

*[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc.*

*IEEE Conf. Computer Vision and Pattern Recognition*

*(CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.*

*[5] G. Huang, Z. Liu, L. Van Der Maaten, and K.*

*Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.*

*[6] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive angular margin loss for deep face recognition,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4690– 4699, doi: 10.1109/CVPR.2019.00482.*

*[7] T. Baltrušaitis, C. Ahuja, and L. P. Morency, “OpenFace: An open source facial behavior analysis toolkit,” in Proc. IEEE Winter Conf. Applications of Computer Vision (WACV), 2016, pp. 1–10, doi: 10.1109/WACV.2016.7477553.*

*[8] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proc.*

*IEEE Conf. Computer Vision and Pattern Recognition*

*(CVPR), 2001, pp. 511–518, doi: 10.1109/CVPR.2001.990517.*

*[9] H. Li, S. Wang, and X. Chen, “Deep learning based attendance management system for classrooms,” Journal of Educational Technology Systems, vol. 48, no. 1, pp. 3–22, 2019, doi: 10.1177/0047239518773021.*

*[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens,*

*and Z. Wojna, “Rethinking the Inception architecture for computer vision,”*