INTELLIGENT ATTENDANCE TRACKING THROUGH FACIAL RECOGNITION TECHNOLOGY USING CNN MODEL

A FINAL YEAR PROJECT REPORT

Presented in Partial Fulfilment of the Requirements for the Award of

Bachelor of Science (B.Sc.) in Computer Science

Department of Computer Science

Faculty of Computing and Information Technology

Godfrey Okoye University, Enugu State

By

IGBOKWE FRANCIS OBINNA

Registration Number:GOU/U22/CSC/925

Supervisor: MR.UGWU NNAEMEKA

JULY 2026

APPROVAL PAGE

This research, titled "**TITLE OF THE RESEARCH PROJECT**," has been assessed and approved by the Department of Computer Science and Mathematics Committees in the Faculty of Natural Sciences and Environmental Studies, Godfrey Okoye University, Enugu, Nigeria.

.....	.....	
Supervisor	Signature	Date
External Examiner	.....	
.....		
External Examiner	Signature	Date
Dr. Frank Okebanama	.....	
.....		
HOD, Department of Computer Science and Mathematics	Signature	Date
Dr .I.A. Ajah	.....	
.....		
Dean, Faculty of Computing And Information Technology.	Signature	Date

CERTIFICATION

I certify that I completed the research included therein and that it was primarily based on my observations and investigation. Consequently, I will fully accept the University's decision if I am found to have cheated on the assignment.

_____

IGBOKWE FRANCIS OBINNA
GOU/U22/CSC/925

DEDICATION

This work is dedicated to my parents, Mr. and Mrs. Igbokwe, whose unwavering support, resources and encouragement made this project possible. I am also profoundly grateful to God Almighty for granting me the strength and guidance to successfully complete this project.

Acknowledgement

All glory and honor to god almighty for granting me the wisdom, health and perseverance to finish this research. My faith in him has been my strength along the way.

The supervision by my supervisor, Mr. Ugwu Nnaemeka, from the inception of this research to its submission has been a great inspiration; his patience, expertise and dedication are much appreciated. His careful feedback and academic guidance were priceless, and I am lucky to have had a dedicated supervisor.

I must express my gratitude to all the teaching staff of the Department of Computer Science and Mathematics, Godfrey Okoye University, for the good education they rendered to me and the passion for computing they gave me.

To my parents, I can't say how grateful I am. Every page of this work reflects your prayers and sacrifices and your unconditional love.

I’m also particulary grateful for you a. I can't thank you enough for your encouragement, understanding and support during the toughest times of this research. You were there to support me through the times of doubt and made me feel confident to move forward. I am blessed to have you in my life.

I also want to thank all my friends for all the humour and encouragement over the course of this programme. This trip was an unforgettable one.

ABSTRACT

Attendance tracking is a necessary operation for every academic institution to keep check of student's participation, academic progress and adherence to institutional policies. Conventional attendance tracking methods using paper-based registers or manual roll call systems are not only time-consuming and error-prone but also easy to be impersonated and tough to handle in large class environments. This paper endeavors to mitigate the issues associated with inaccurate and inefficient attendance management using the "Intelligent Attendance Tracking System using Facial Recognition Technology". We developed and implemented the facial recognition-based attendance tracking system by utilizing React.js for the frontend, Flask for the backend services and SQLite for the database system whereas deep learning based face recognition mechanisms are employed for detecting faces and matching. YOLOV face detection module was implemented for detecting facial regions. Face recognition mechanism based on the Euclidean distance matching on facial embeddings was implemented for checking students and marking their attendance. Features supported by the system are registration of student, attendance using live camera, attendance using external camera, automatic attendance marking, blocking duplicate attendance, generation of attendance reports and notifications for absent students. In addition, the experiment shows the accuracy of 91.6% the precision 89.8% recall 88.9% and F1-Score 89.3% are achieved for attendances'automatic detection based on the collected few real data of real-time environment from originally face images of the student.This system represents an advancement towards smart academic management systems.

Table of Contents

CHAPTER ONE

INTRODUCTION

1.1 Background of Study

In the many real and practical real systems in life, the attendance of the student in a classroom determines his or her performance almost all automatically made in the human being identifier system by using some traditional methods, such as Ids, rollcalls or fingerprints. (Thai-Viet Dang.2023). The current reliance on paper logs mostly stems from low cost, but they pose challenges to attendance management due to the high chance of fakes, errors, slow check-in and verification processes. The Card scanners increased check-in speed, but another person could swipe on behalf of the legitimate person. Fingerprint scanning helps combat fraud, but also brings with it germ risks, queues become chaotic, and machines may break due to heavy daily usage. Facial recognition cameras offer another effective way to identifying people by looking at them through a lens.

Face recognition is arguably the most challenging biometric modality among others to realize in unconstrained environments since the variability in face images in the real-world is tremendously high. It may comprise of variations like head posture, illumination, facial expression, occlusion. With recent achievements in computer vision, CNN has demonstrated success in a wide array of computer vision tasks which may include optical character recognition, Segmentation, facial expression analysis, object detection & recognition, etc., and age estimation. (M. A. Ahmed et al.2022). As far back as (1998 LeCun et al) used gradient tuned algorithms and the results proved it could work for an array of pictures-the technology just had to improve further with messy backgrounds and a range of faces. These days, with a more advanced neural network to carry out face analysis it has increased rapidly in speed and efficiency. Two faces systems in fact rival human performance, but it should be stated one does so through good use of positions and features and the other through reducing identity information into a compact vector form to quantify difference with just a distance measurement.

So why are schools adopting facial recognition? It’s fast and it’s contactless. No cards to carry around, no need to remember passcodes, and no need to swipe. Faces are identified in milliseconds, time, date, student's matric number captured and along side; there are reports displayed and documents generated and exported easily. Analysis of check-ins demonstrate that facial recognition can outperformfake two-factor systems and tidies up processes (Kamil et al. 2023; Rao 2022), however performance suffered from a lack of ability to easily integrate with other technology, or in large scale systems, the management process was difficult (Kamil et al. 2023; Rao 2022).

Two reasons why the work was done: Technological progress and organizational processes. A facial recognition CNN's' efficiency under varying environments and human constraints have been proven, at the same time schools continue to face challenges such as reducing paperwork, preventing fraud andmoving on from more old-fashioned practices. At the intersection of these two driving forces face recognition came about.

1.2 Statement of Problem

Attendance management problems are common to most institutions.

I. Inaccuracy and inefficiency of the manual attendance system

Institutions which practice a manual attendance system, roll call, and signature check, the process is time consuming and results in inadequate and late records before feeding them into the main system. This becomes an inefficient process, and the quality of institutional attendance data is affected.

II. Impersonation and proxy attendance

Paper-based signatures and roll calls do not authenticate identity, and impersonation is a concern. Although the use of cards is an improvement, cards can still be shared. These problems affect the fairness, accountability, and legitimacy of any decision based on attendance (such as eligibility for exams).

III. Data management and integration problems

Data is often held in isolated formats such as paper, registers, and spreadsheets that don't communicate. This leads to errors in reporting, problem identification (such as serial

absenteeism), and quick access to records. This hampers the institution's capacity to do longer-term analysis or use data for decision-making. Ultimately, attendance data cannot inform academic planning and monitoring efforts.

1.3 Aim and Objectives of the Study

The purpose of this study is to design and document “Intelligent attendance tracking through facial recognition technology using CNN model”.

In an effort to achieve these aims, this study has the following objectives which are to:

1. Enhance effectiveness and accuracy in attendance tracking
2. Accurately identify and prevent proxy attendance
3. Create a unified and comprehensive data system
4. Model and assess an attendance system

1.4 Significance of the Study

Why This study matters for; In our work, we presented the development of smart and facial based intelligent attendance system for effective attendance monitoring which benefits the follows:

- **Students**

  Students will benefit from the system because it is a faster and more convenient attendance marking. The system eliminates the need for manual signing and reduces impersonation among students. It also ensures that attendance records are stored accurately and securely.
- **Lecturers**

  Lecturers could utilize this additional time more efficiently for the teaching of lessons.Automation could lead to increased accuracy of recorded attendance data.
- **School Administrators**

  School administrators benefit through improved monitoring and management of student attendance records. The system provides real-time attendance reports, attendance

statistics, and absent student notifications, making administrative supervision easier and more efficient.

- **Academic Institutions**
  Educational institutions benefit from improved attendance accuracy, reduced examination malpractice associated with impersonation, and better record management.
- **Researchers and Developers**
  The study contributes to the field of artificial intelligence, computer vision, and biometric systems. Future researchers and software developers can use the study as a reference for developing advanced facial recognition and automated management systems.

1.5 Scope of the Study

This paper addresses design & implementation of intelligent attendance tracking system with face recognition for academic institutions. The core purpose is for automate tracking and management of students' presence, using face as a biometric parameter. Our proposed web based application is equipped with a responsive and intuitive frontend framework i.e., React JS for UI rendering.The backend logic and functionality are developed using Flask.

Data Storage is managed efficiently through a relational database (SQLite) where facial biometric data are stored as vectors for recognition purposes. We have integrated open-sourced deep learning based face detection and recognition algorithms and models to perform feature extraction, facial matching & identification. For now, we have restricted the system for student attendance and managing their day to day attendance data in the premise of academic institutes, instead of a full fledged academic institution management suite including result management, finance management, accommodation management & etc..And the system is primarily implemented on a condition of controlled or near-controlled lighting environment and does not address extreme conditions, like low lighting or obstructed faces and large scaled enterprise system.

1.6 Limitations of the Study

I. Other Limits in Capturing Clear Images

Dark shadows could affect the performance of the system. Faces too far in profile are not the best. Low-resolution cameras are less accurate. You need constant distances from the subject to the camera in crowded areas.

II. Learning rates on quality and diversity of data

Face lighting is important, poor shadows lead to poor contrasts. Sure, cooperation is key, but users can't beat bad cameras. This can result in incomplete information being stored. Matching fails when templates aren't rich and formed by multiple samples.

III. Infrastructure and Deployment Limits

As the number of users increases, the system needs to perform faster. At this scale, this means greater computing power. Large systems need reliable connections to avoid stutters. Running many tasks stresses the components' speed.

1.7 Operational Definition of Terms

- **Attendance System**: A system used to record and manage the presence or absence of individuals in an institution.
- **Convolutional Neural Network (CNN):** is a deep learning model that is well suited for image recognition tasks and works effectively for analyzing images.
- **Dataset:** data that you train your ML model with (for use with real life scenarios)
- **Facial Recognition**: The technology is used to confirm the identity of or the existence of an individual through distinctive traits and features observed in digital or video images.
- **Image Processing**: The technique of enhancing or analyzing digital images to extract useful information.
- **Model Training**: Teaching a machine learning algorithm how to identify trends from data.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

For doing this project, The past work can be broadly categorized into three mutually connected research topics. One category is on the mechanism of training computers to recognize photos with the help of deep networks. Another field focuses on recognition of faces in a photograph and that of facial features with the help of trained models. The third is concerning with devices capable of keeping a record of whether a person is present at a certain place or not, with the help of scanning or biological properties like shape of a face. All the above are integrated into actual system where by the manipulating photographs effects the reliability and feasibility of the device to be used by humans. Review of previous work leads to the motivation for taking specific steps here and everything thereafter has been determined based on the understanding of the work done and not any assumptions.

2.2 Theoretical Background

2.2.0 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a deep learning architecture designed for processing grid-like data such as images. It automatically learns spatial features (edges, textures, shapes) through layers of filters rather than relying on manual feature engineering — which is why it's the standard choice for tasks like facial recognition.

Core layers:

- Convolutional layer – slides small filters (kernels) over the input to detect features like edges and patterns, producing feature maps.
- Activation (ReLU) – introduces non-linearity, allowing the network to learn complex patterns.
- Pooling layer – downsamples feature maps (e.g., max pooling), reducing size while keeping important information.
- Fully Connected (Dense) layer – flattens the features and combines them to make the final prediction.
- Output layer – produces class scores (e.g., via softmax) for classification.

2.2.1 Biometric Authentication Theory

Biometric identification theory discusses the application of unique human characteristics to authenticate a person's identity. Facial recognition is a biometric identification technique because everyone's facial structure is individual. This theory justifies using students' faces for identification when checking their attendance.

2.2.2 Facial Recognition Theory

The facial recognition theory states that all facial shapes, like the shape of the face and eyes, the mouth, the nose, could be numerically calculated and expressed as a matrix or numerical representations known as embeddings. These embeddings could then be compared with other facial data, where a match would result in an identification.

The presented system uses facial recognition by identifying the face and finding its facial features. It generates an embedding for the identified face and compares the embedding to the student embeddings in the database.

2.2.3 Artificial Intelligence Theory

Artificial intelligence theory primarily focuses on achieving computer systems capable of completing tasks typically done by humans, involving human intelligence. The application of artificial intelligence in the presented study covers: Face detection, Image analysis, Pattern recognition, Decision making, Verification of attendance. AI models are used to automatically detect faces, and then the detected face is checked against the registered student.

2.2.4 Machine Learning Theory

According to theory of machine learning, computers are capable of learning the relationships or pattern from data and predict future trends based on this learned pattern. Computer-based facial recognition systems utilize machine learning models that have been trained on face image databases for the task of recognizing a face.

The system uses machine learning techniques for:

- Face detection
- Feature extraction
- Embedding generation

- Similarity matching

2.2.5 Computer Vision Theory

It means creating computer software or hardware systems that can view images and videos as if humans see them and processing those visuals accordingly. This vision can be then used for functions like face identification and recognition using digital cameras and processing technologies.

The proposed system applies computer vision techniques to:

- Capture facial images
- Detect face regions
- Crop faces
- Preprocess images
- Analyze facial patterns

2.2.6 Pattern Recognition Theory

The pattern recognition theory explains the process how the systems can distinguish the matches or mismatches among the data patterns. In the case of face recognition, the system extracts facial embeddings and calculates the similarity scores of the features to check the identity match.

The proposed system uses the Euclidean distance matching method to compare the features and match the nearest face embedding.

2.3 Review of the related Literature

Many researches have done on automatic attendance systems and facial recognition technologies. A various number of biometric techniques, AI technologies and computer vision strategies have been developed for enhance monitoring attendance, security and automation.

The aforementioned Euclidean distance formula is applied for facial recognition to comparefacial embeddings, which compute the similarities between faces detected and stored faces.

However, the first significant development arose with Taigman and their team in 2014. Their DeepFace work demonstrated how deep neural networks could rival human accuracy when

identifying faces. They blended 3D facial alignment with deep neural networks and recorded high accuracy on the LFW dataset. The importance of this work lay in its demonstration that learned features outperformed hand-crafted features from earlier attempts. This approach, however, relies upon access to enormous datasets and high-performance hardware, which are typically unavailable in smaller organizations. This approach, although historically important, presents scalability problems for on-premise deployment even today.

In 2015, Schroff and their team introduced FaceNet, an approach which learns a single, unified representation for both identifying and clustering faces. Instead of defining classes beforehand, this approach aligns the relative distance between reference images and dissimilar or similar images via triplet loss. This results in tightly constrained feature vectors that are easily scalable and can be compared directly for identity. In the context of sign-in systems this means a clean separation of tasks whereby once a feature vector has been obtained, the only task left is to compare it to the feature vectors of known attendees. Test scores published back then resulted in the state-of-the-art on LFW and YouTube Faces and demonstrated clearly that an embedding approach works well for facial recognition in real life scenarios.

First location the face is in, then recognize, states Zhang and their team in 2016. Before actually identifying a person, the starting step must be correct positioning within the image. The proposed approach is built on top of MTCNN but improves facial localization accuracy prior to subsequent stages, which in turn boosts identification robustness. The preceding steps are not skipped because their inclusion is indeed critical for downstream accuracy and thus have their own place in the design process.

A new aspect of handling messy scenes came with Huang and team in 2007 with LFW. Although not specifically designed to detect presence, LFW has undoubtedly stimulated later research which explores how a face-recognition system can be robust in dynamic environments. Leaning on concepts that it embodied rather than mimicking them directly, the aim here was to create a system to handle an environment where lights are shifting, the backdrop might be complicated, the face angles can be far from perfect. Clearly such system needs to be resilient against such challenges in order to be applicable at school or an office.

The recognition of the attending subjects has prompted some researchers to explore the use of facial recognition systems in practical environments. Moving beyond traditional roll-call and

register systems, in 2021 Khairnar and team developed a facial recognition approach using the OpenCV libraries. This has been built upon; Kamil's system in 2023 combines facial identification and mask detection in a browser-based sign-in system. In another approach AttenFace was developed two years ago by Rao; this method bifurcates tasks, enabling separate facial detection/identification and person tracking stages thereby enabling lower overall computational load as work can be distributed. In this system the key difference to what has been presented is in that the emphasis is now on overall task optimization rather than merely facial identification accuracy and ease of web access and scalability are crucial design considerations.

Across recent papers the key takeaway seems to be the simultaneous presence of established visual recognition approaches along with pre-trained identification networks and a system that stores feature vectors in a database that handles the tracking. Take for example several recent applications: footage is typically captured by a camera system running OpenCV software, processed through a feature extraction stage, after which a deep neural network is employed to verify facial identity. The approach here is similar: recognition of facial identity is one component in a larger system for class attendance. The differentiating aspect of this work: a heightened focus on a systematic architecture and code implementation along with documentation throughout the entire design process.

Looking back through published papers there is clearly a theme that although very high performance is reported, most lack significant details regarding repeated entries, permissions, user management, data structure, and historical record-keeping. This omission is critical in an attendance system for offices, not research labs that are merely testing facial recognition. For a school this is even more critical with clear protocols and robust record keeping; the intent here is to provide a comprehensive system.

Every paper appears to note the fact that their performance is not to be considered definitive proof. While benchmarking a system determines its recognitional capability, such small-scale tests fail to reflect real-world conditions. In this system the problem has been addressed head on by presenting results only as proof of concept prior to intensive testing within target environments.

2.4 Summary of Literature Review

S/N	Author(s) and Year	Contribution / Work Done	Technique / Methodology	Limitation	Difference from the Present Study
1	Taigman et al. (2014)	DeepFace narrowed the accuracy gap between automated and human levels face verification, and demonstrated the power of deep representation learning.	Deep neural network with alignment and large-scale training.	Computationally expensive and not designed as a complete attendance workflow.	Lower accuracy and no real-time attendance
2	Schroff et al. (2015)	FaceNet obtained compact embeddings that could be exploited for recognition and clustering.	Triplet loss with deep CNN embedding learning.	Requires careful training, data design, and threshold calibration.	Older detection method
3	Zhang et al. (2016)	MTCNN improved face detection and alignment in unconstrained images.	Cascaded multitask convolutional networks.	Adds processing overhead and still depends on capture quality.	Review-based study
4	Huang et al. (2007)	LFW provided a benchmark for unconstrained face-recognition evaluation.	Large-scale benchmark dataset and pair-matching protocol.	Benchmark performance does not automatically transfer to institutional deployments.	Limited deep learning support
5	Khairnar et al. (2021)	Demonstrated a practical classroom attendance system using face-recognition ideas.	OpenCV-based image capture and recognition workflow.	Limited emphasis on formal system modeling and long-term scalability.	No attendance system
6	Kamil et al. (2023)	Proposed an online attendance	Browser-based interface with	Focused on application	Requires physical

		system with facial recognition and mask detection.	face-recognition pipeline.	context; limited discussion of software architecture depth.	contact
7	Rao (2022)	Presented AttenFace, a real-time attendance system that separates recognition from attendance logic.	Parallel attendance calculation architecture using snapshots.	Prototype orientation; requires broader institutional validation.	Vulnerable to impersonation
8	Parkhi et al. (2015)	VGG-Face is a deep convolutional neural network developed specifically for facial recognition tasks.	Deep CNN architecture containing multiple convolutional layers for extracting facial features automatically from images.	The model was computationally expensive because of its large architecture size.	General biometric applications
9	Sun et al. (2014)	DeepID introduced a facial recognition approach that focused on learning identity-specific facial features using deep neural networks.	The researchers used Convolutional Neural Networks (CNNs) to generate highly discriminative facial representations for identity classification.	The system required very large training datasets and extensive computational resources to perform effectively, making it data-intensive.	No attendance features
10	Amos et al. (2016)	OpenFace was developed as an open-source facial recognition framework	The system employed a lightweight CNN architecture for	Although lightweight, its recognition accuracy was generally lower	More advanced training architecture

		inspired by FaceNet.	generating facial embeddings used in recognition.	than that of more advanced commercial or large-scale deep learning systems.	
11	Turk & Pentland (1991)	Eigenfaces was one of the earliest facial recognition techniques and laid the foundation for many later biometric systems.	The study used Principal Component Analysis (PCA) to reduce facial images into smaller mathematical representations called eigenfaces.	The method performed poorly under varying lighting conditions, facial expressions, and pose changes, reducing practical reliability.	No attendance module
12	Ahonen et al. (2006)	The Local Binary Pattern Histogram (LBPH) method focused on analyzing facial texture patterns for recognition purposes.	It used texture-based feature extraction by comparing neighboring pixel intensities and converting them into binary patterns.	The algorithm was highly sensitive to image noise and environmental variations such as poor illumination and blur.	General object detection
13	Wagh et al. (2019)	Face-Based Automated Attendance management system was proposed in this paper that automatically recorded attendance without human being.	The researchers used OpenCV-based facial detection and recognition techniques for identifying students.	The system showed reduced recognition accuracy under poor lighting conditions and crowded environments.	Limited real-time features
14	Patil et al. (2020)	This research improved	The system applies deep	This was not as successful due to	Limited duplicate

		attendance automation by integrating CNN-based facial recognition into attendance tracking.	learning with the CNN model to capture and identify the faces of students automatically.	relatively low number of input data which restricted it to a specific scenario.	prevention
15	Serengil & Ozpinar (2020)	The DeepFace library combined several state-of-the-art facial recognition models into a unified framework for easier implementation.	The framework integrated hybrid deep learning models such as FaceNet, VGG-Face, OpenFace, and DeepID into one library.	The system heavily depended on pretrained models and external dependencies, reducing flexibility for custom optimization.	Limited frontend integration

Table 2.1: Summary of selected reviewed works relevant to CNN-based facial recognition attendance systems.

2.5 Research Gap

Prior to that, attendance tracking systems mostly used manual procedures, fingerprints, RFID cards or even naive facial detection methods, all of which involved problems with impersonation, poor accuracy rates, hardware dependence and unorganized attendance tracking. Most prior facial detection systems have the problems with low lighting conditions, angle issues and improper detection methods causing inaccurate recognition. There were also few prior systems lacking features like real time notification, no duplicate entry, integration of multiple cameras, multiple face detection and proper handling of unknown faces. Apart from this some systems also used restricted desktop based methods and unappealing UI's.

This paper proposes an Intelligent Attendance Tracking System by providing a system with a more efficient and reliable facial recognition,live andexternal camera functionality, multi angle registration for a person, real time tracking and monitoring, absence detection and notifications, no duplicate attendance, and a Web-based interface built with React.Js, Flask and SQLite.

CHAPTER THREE

SYSTEM ANALYSIS & DESIGN / RESEARCH METHODOLOGY

3.0 Introduction

This chapter provides a description of the system analysis, research design and methodology that has been followed in developing the Intelligent Attendance Tracking System using Facial Recognition Technology. The system has undergone several development stages where the process includes the acquisition of the system requirements and analysis of the current manual attendance process. The current system architecture design, and the system implementation has also been described in this chapter.

3.1 System Analysis

This proposed system was attempted to automate the attendance taking process using facial recognition and eliminate impersonation, errors, and the time wasted from manually tracking attendance. The analysis phase of this project analyzed the shortcomings of the manual attendance system, and outlined the required functional and non-functional requirements to develop a useful system.

The system was modeled using use case diagrams, activity diagrams and class diagrams in the design phase. In the proposed system, the React.js was used for the frontend interface and the Flask was used for the backend service, SQLite was used for the management of the database and the face recognition part of the system would utilize YOLO/CNN-based face recognition technology. The proposed system would provide for live camera feed, remote camera feed, mobile camera feed, automatic attendance tracking and reporting and notifications.

3.1.1 Analysis of Existing System

The typical attendance management system currently used in most academic institution is mostly manual and relies on paper. In a typical attendance management system, lecturers or course representative mark the attendance physically by requiring students to sign attendance sheet, or verbally.

In the manual attendance system, students have to write their names, matric number and signatures in the attendance sheet. In other manual attendance system, the lecturer will read out the names one by one and the students will reply accordingly. However simple it may be, the operation has some advantages and disadvantages.

3.1.2 Limitations of the Existing System

The existing system has various shortcomings:

- **Impersonation**: Students can sign attendance on behalf of absent colleagues, leading to inaccurate attendance records. This reduces the reliability and integrity of the attendance process.
- **Time consumption:** Manual attendance taking consumes valuable lecture time, especially in classes with large numbers of students. The process of distributing attendance sheets, verifying records, and compiling attendance data is often slow and inefficient.
- **Poor record management**: Attendance records stored on paper are vulnerable to loss, damage, duplication, and misplacement. Retrieving attendance information for report generation or student verification can also be difficult and time-consuming.

In order to overcome the shortcomings of the existing system an automatic attendance management system with facial recognition will be built with better accuracy, prevention against impersonation, save time and track the attendance real-time and secure. Thus the proposed system Intelligent Attendance Tracking System with Facial Recognition Technology is intended to resolve the problems of manual attendance management system. The manual system was replaced with a system that automate student verification and recording of their attendance using facial biometrics.

3.2 Analysis of the Proposed System

This proposed system is the intelligent attendance tracking system (IATS) used to capture facial print of the students to achieve the auto managing process. The aim of proposing such system were to resolve problem in manually managed system of attendance by introducing Biometrics Identification and auto recording attendanceThe proposed system comprises of the components used to capture face images of the students in real time during attendance. The captured images are compared against a dataset using facial recognition algorithm to detect students and record their attendance. Computer vision, machine learning, database system, and web application technologies are used to ensure efficient and reliable attendance management. The major difference between the proposed and manual attendance system is that, in proposed system, before recording attendance, it checks the identity of the student using facial recognition which overcomes the problem of students taking attendance of others.

The proposed system consists of several integrated components, including:

- Frontend user interface
- Backend server
- Facial recognition module
- Database management system
- Notification and reporting modules

Administrators can use the frontend to enroll students, recognize faces, track attendance history, and create reports. The backend server receives requests, performs the image processing and recognition functions, and interacts with the database. The facial recognition module performs:

- Face detection
- Face preprocessing
- Facial embedding generation
- Face matching and verification

The database stores student records, facial embeddings, and attendance information for future retrieval and analysis.

The proposed system supports:

- Live camera recognition

- External webcam integration
- Phone camera integration
- Multiple face detection
- Real-time attendance recording
- Duplicate attendance prevention
- Attendance notifications
- Attendance report generation

When the system is running, the student's face captured via the camera is sent to the backend server. It checks for the student's face, image is preprocessed, and a facial embedding is extracted and compared with other facial embeddings in the database via Euclidean distance matching. When the score reaches the limit of recognition, it recognizes the student and marks attendance.

It also checks whether the student has marked attendance on that particular day or not. If already marked attendance, it shows “already marked”.

If the face does not match with any of the registered students, it returns "unknown face".

It also marks and identifies absent students, in order to do so the system compares with registered students, students who were marked absent.

The system also has a notification system that checks the students who were absent.

The system that has been proposed is faster and more reliable than the manual attendance system. It provides a secure, fast, and automated system to manage attendance.

3.2.1 System Requirements Specification (SRS)

From the attendance issue area come the necessary functions of the system. Safe login by administrators should be implemented. Student details such as their names, matric numbers, departments, and years are stored in the database upon registration. The app receives images from a camera that can identify faces. After identification, these faces are cropped and then transformed into numerical representations. These unique numbers are then mapped to their corresponding student records. A user gains permission to start roll call. A picture is then taken. It then examines faces captured. If the faces are recorded in the database no other information

about the students is captured; it is a new record. The logs appear on screen. The logs can then be sorted either by date or name, and also be printed out.Reports can be produced from these images.

3.2.2 Functional Requirements

- The system shall authenticate administrative users before any registration or attendance activity is performed.
- The system shall register new students and store facial templates with biodata.
- The system shall capture a live face image, preprocess it, and compute an embedding.
- The system shall compare the live embedding with stored templates using a similarity measure.
- The system shall prevent duplicate attendance for the same student within the same attendance window.
- The system shall generate a searchable attendance report and allow export.

3.2.3 Non-Functional Requirements

- The system should provide a recognition response in a short and operationally acceptable time.
- The interface should remain usable on an ordinary desktop or laptop browser.
- The architecture should be modular enough for future improvement or replacement of the recognition engine.
- Credentials and access privileges should be protected through secure authentication.
- The data layer should preserve integrity and support backup and recovery

3.3 System Design Methodology

Object-Oriented Analysis and Design (OOAD) was used to model the system components and their interactions. OOAD focuses on identifying system objects, their attributes, methods, and relationships. This approach simplifies system development by breaking the system into manageable and reusable components. UML diagrams such as Use Case Diagrams, Activity Diagrams, Class Diagrams, and Entity Relationship Diagrams were used to represent the structure and behavior of the system.

3.3.1 Justification for Methodology

OOAD was appropriate for this study because it provided a clear development structure, improved system organization, simplified maintenance, and ensured that all system requirements were properly implemented.

A main factor that makes OOAD chosen is modularity. It seems that the proposed system has several individual modules as student registration, face detection, face recognition, attendance management, reporting and notification modules. With the application of OOAD, all of these modules can be implemented as independent classes and modules, it ease to design, test, maintain and upgrade the system.

The reusability is another benefit of using OOAD. Some components such as facial recognition class, database function class, camera link up class can be reused for new versions or other applications. It reduce the time of development and also enhance the quality of software.

Another benefits of using OOAD is system components and relationship transparency. With UML diagrams (e.g. Use Case Diagram, Activity Diagram, Class Diagram) which clearly define system structure and relations, it enhances understanding of system design for developers as well as any other related people.

3.3.2 Method of Data Collection

The data needed to design and implement the Intelligent Attendance Tracking system using the Facial Recognition technology was derived from collecting facial image data sets. Facial image data was collected from the student during the student registration process when the implementation and test of system. The image of the faces was taken from laptops, external web cameras and mobile phones, etc. Multiple face images were taken from different viewing angles to achieve greater recognition accuracy. This also enables reliable facial embedding to be produced.

The collected facial data included:

- Front-view facial images
- Left-side facial images
- Right-side facial images
- Slightly tilted facial images

These images were processed and converted into facial embeddings for storage and recognition purposes.

3.4 System Modeling

From a function view in Usecase diagram, there are two major Actors: The Administrator and the Student.

The Administrator would be performing Login, Register students, Manages students records, View students records, Export the attendance. And The Student only need to be in the range of camera so that the faces can be captured and recognized.

Viewing step by step will be able to provide you how checking the attendance really working from the activity diagram. As soon as the person opens the attendance screen, the process start. After that a photo would be taken and modified into a working picture. The process of comparing the live data and the database data only start from this point. There would be an authentication failure or asking the same question when there is no proper matching from the database. The duplicate checking happens when there is a high similarity detected before the data is saved. The purpose of the design here is to show you where exactly the rule interacts with the identification part.

Looking at the class diagram, we group classes by role, such as User, Student, Attendance, and RecognitionService. The User class would include the Authentication process, where the Student class includes the personal information and face data, session results stored in the Attendance class, while face check operations will be stored within the RecognitionService class. Through this, different components are kept separated to easily modify and reuse. And viewing through sequences, we could get the time between each operation and also the way they talk to each other. As shown from the diagram, the student will speak to Frontend, then Backend API, and then Recognition Service before talking to the Database.

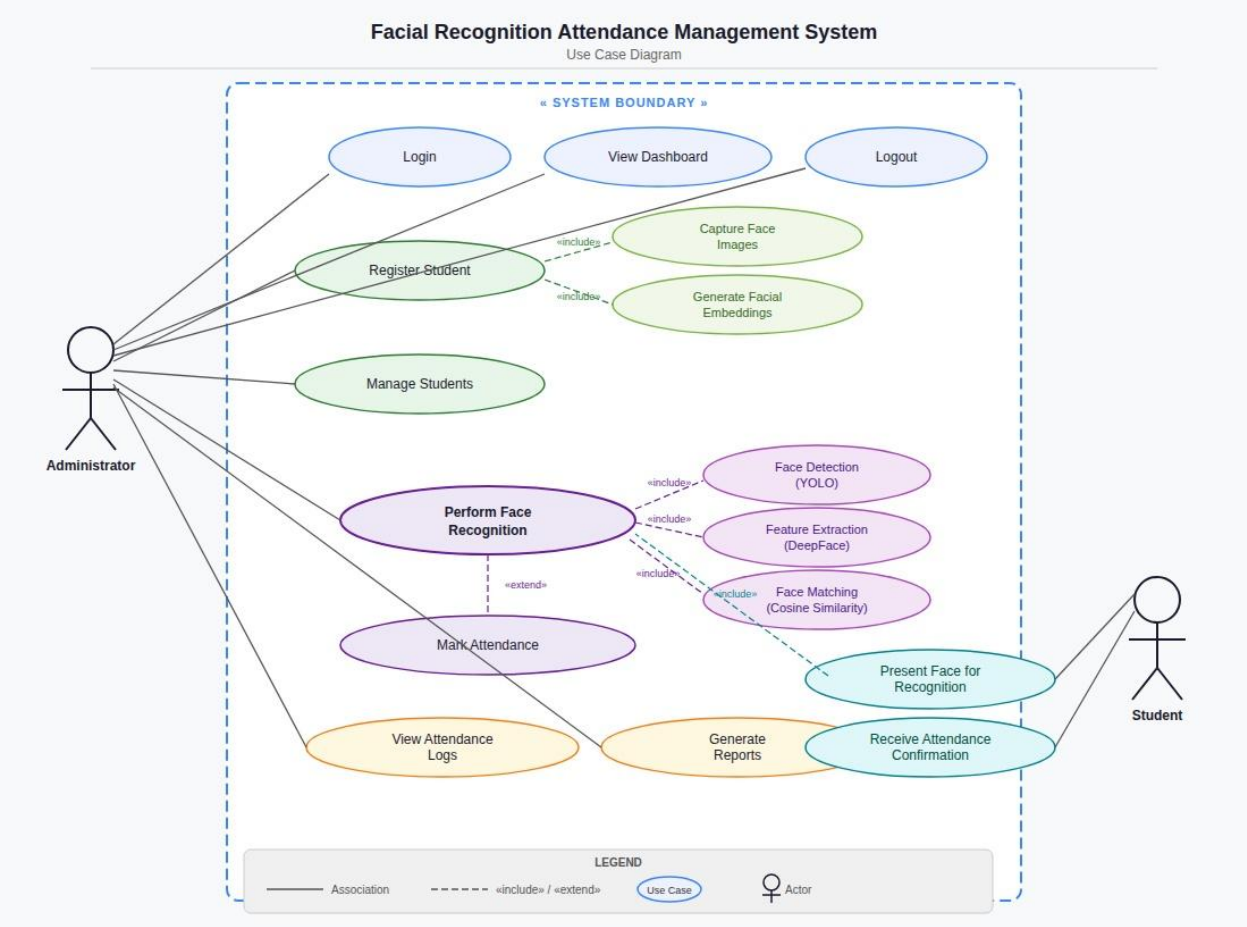


Figure 3.1: Use Case Diagram of the proposed system.

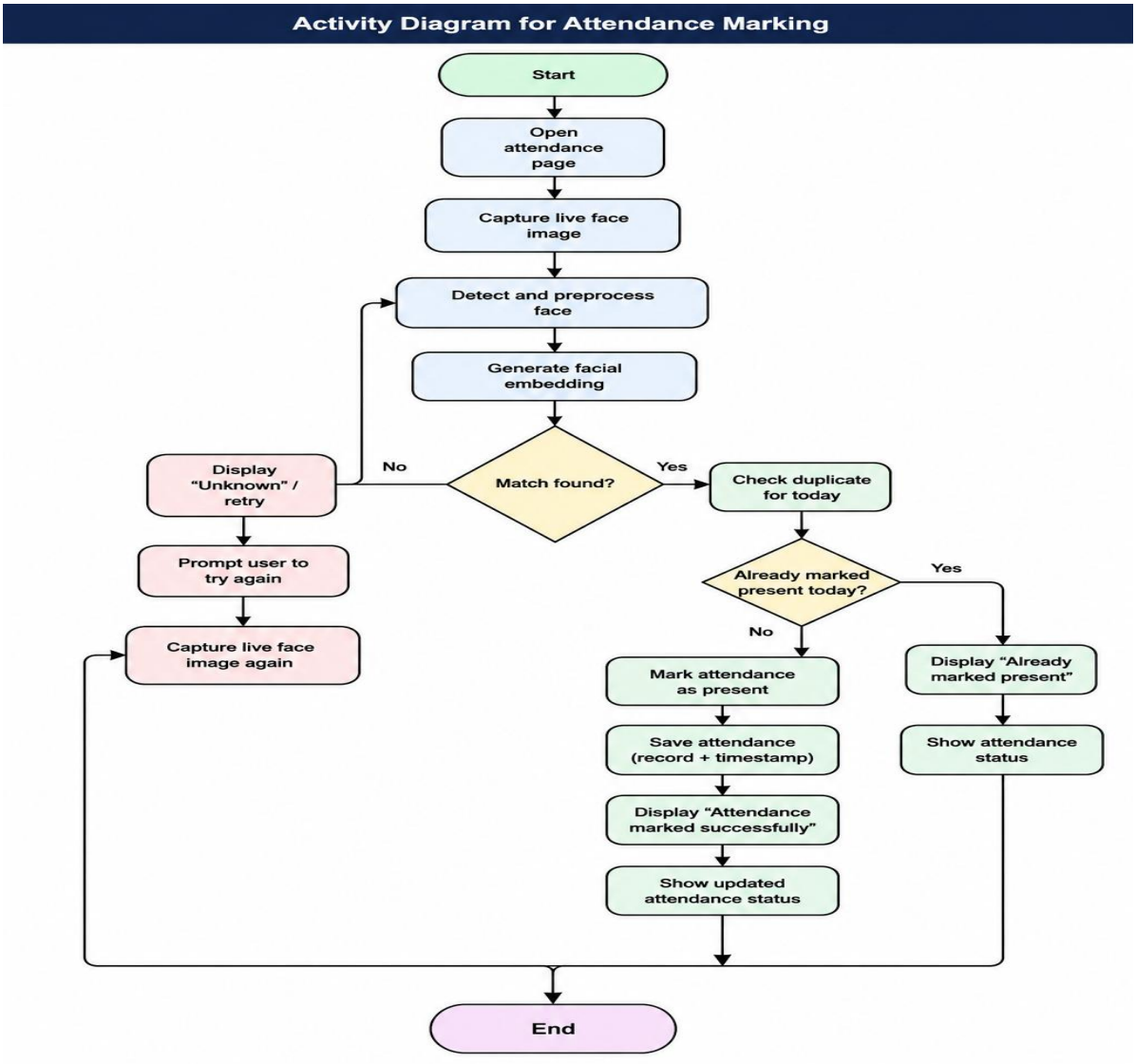


Figure 3.2: Activity Diagram for attendance marking.

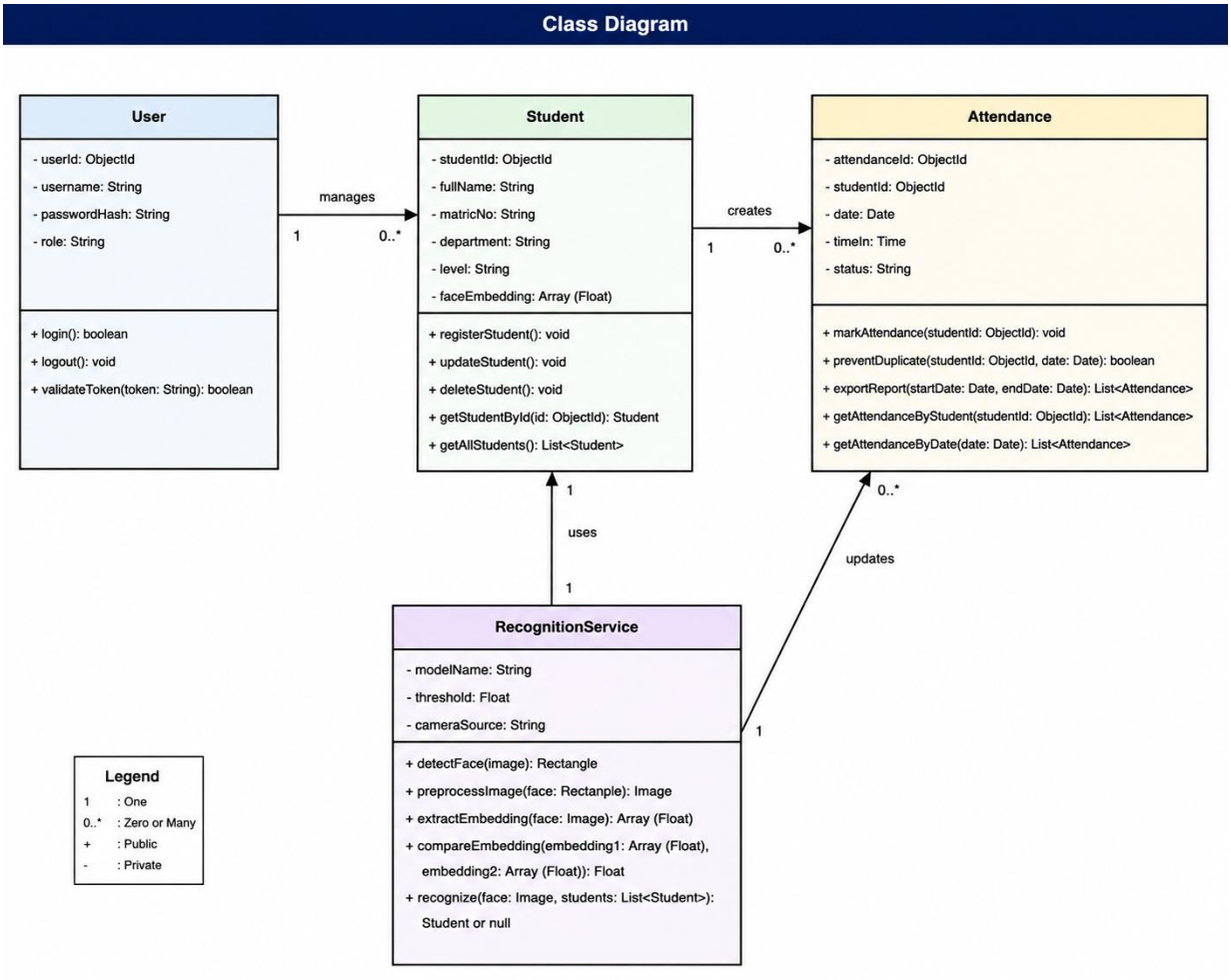


Figure 3.3: Class Diagram of the proposed system.

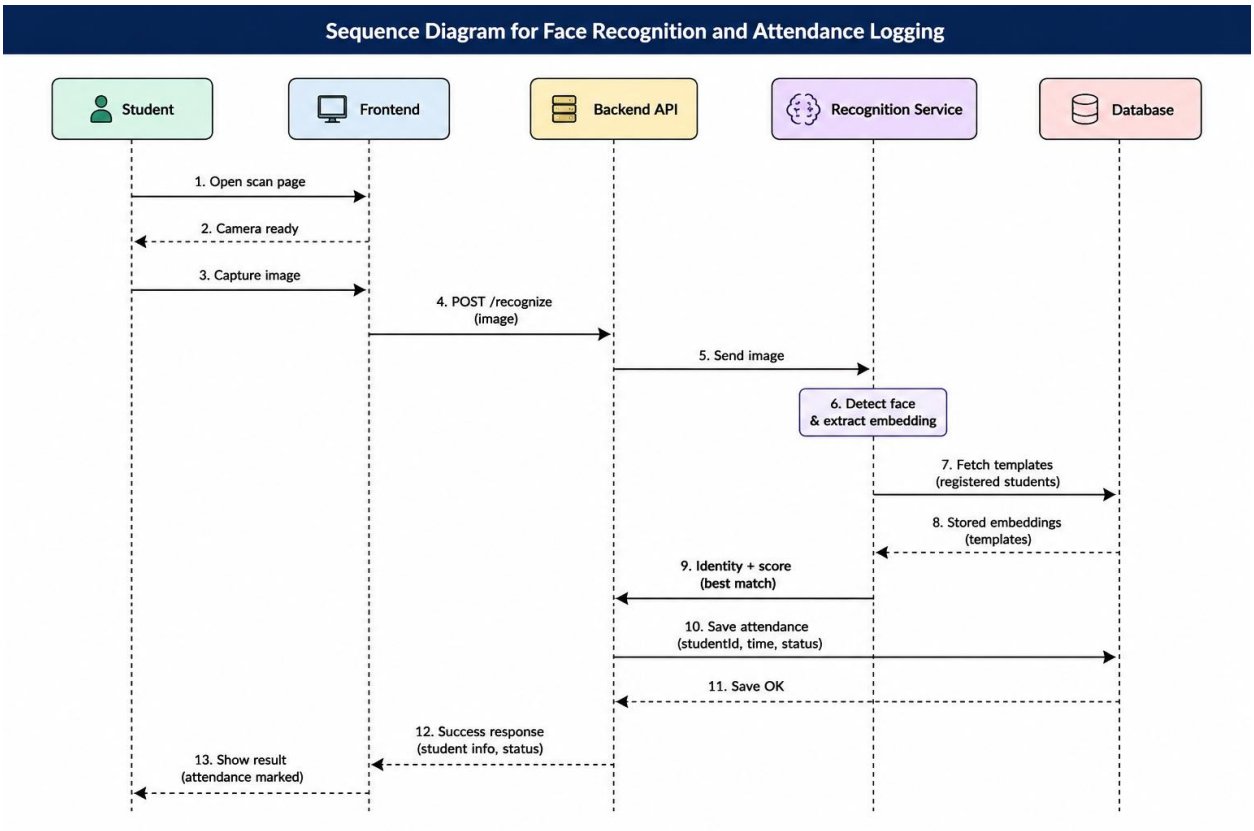


Figure 3.6: Sequence Diagram showing runtime attendance interaction.

3.5 System Design

It is planned to make the system as intelligent attendance monitoring system where facial recognition feature ensures perfect automation for the effective attendance record. And it is planned to automate existing traditional attendance record with biometric system with increase security, speed and reliable for attendance record and monitoring.

The system has to capture the face of a student using a camera; face detection model will be used for face detection of student face then facial embedding can be generated, this will be compared to student records, when the matching between the student record and the facial embedding is done a valid match will return and student's attendance is registered.

The system comprises of four main components. They include User Interface, Back-end server, Facial recognition module and the Database. The user interface is meant for student registration and also face capturing, facial recognition, checking attendance record, viewing reports, view

absentees and also notify absent student via alert or notification. The back-end server is to handle request of system, processing of images, interface with the facial recognition module and manage of the database system. Facial recognition module will perform face detection, facial preprocess, embedding and matching of faces. The database system will have a table for student detail, student facial embedding and the attendance log of each student.

The proposed system supports different ways of input, which are through the live camera, uploading from a file, through an external camera or through mobile camera. While registering a student, administrator will fill in the name, matric no, department and level, and then capture student's face, generate it facial embedding and save to database linked to student record. While recognizing, live camera feed will be captured and detection will be performed on detected faces, generated facial embeddings will be compared to the existing facial embeddings using Euclidean distance to find if it's a valid match andattendancewill be marked. When an existing match is found with accepted distance threshold, student is recognized and attendance is recorded.If student is already present, message of already marked will be displayed.If there is no valid match, it returns unknown faces without marking attendance. This notification system also has a mechanism to track which student failed to make it to class by comparing list of registered students and list of attended students for a particular day. Reporting system is available in the module showing attendance summary report of each student as well as a summary of all students, number of students present and absent and percentage of attended student.

3.5.1 Input Design

The Intelligent Attendance Tracking System provides several input screens that allow administrators to interact with the system. These interfaces include the Login Page, Student Registration Page, Face Recognition Page, Attendance Management Page, and Settings Page.

Student Registration Input Form

The student registration form is used to capture student information and facial images. The following data fields are required:

Field	**Type**	**Validation**
Student Name	Text	Cannot be empty

Matric Number	Text	Must be unique
Department	Dropdown	Required selection
Level	Dropdown	Required selection
Facial Image	Camera Capture	Face must be detected

During registration, multiple facial images are captured from different angles using a laptop camera, external webcam, or mobile phone camera. The system validates that at least one face is detected before registration is completed.

Validation Techniques

The following validation mechanisms were implemented:

- Required field validation
- Duplicate matric number prevention
- Face detection validation
- Recognition confidence validation
- Duplicate attendance validation
- Data type validation

These validations improve data accuracy and system reliability

3.5.2 Output Design

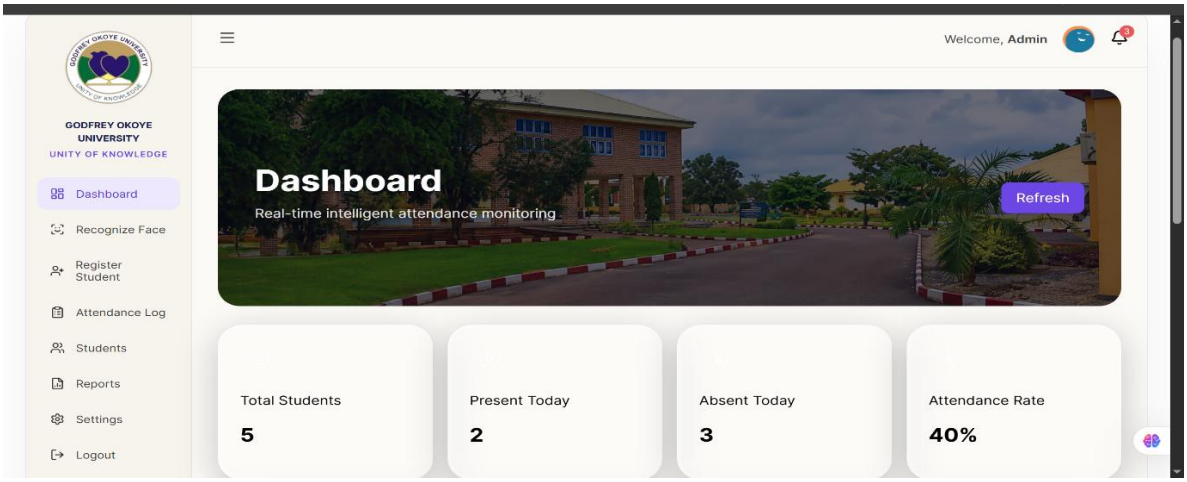
This system generates outputs through dashboards, attendance tables, notifications, and reports.

Dashboard Output

The dashboard provides a summary of attendance information, including:

- Total Registered Students
- Students Present
- Students Absent
- Attendance Percentage
- Recent Attendance Activities

The information is displayed using cards and summary panels for quick decision-making.



Recognition Result Output

After facial recognition, the system displays one of the following outputs:

Successful Recognition

- Student Name
- Matric Number
- Attendance Status (Present)
- Recognition Confidence Score
- Distance

Duplicate Attendance

- Student Name
- Message: "Attendance Already Marked"

Attendance Records Output

Attendance information is displayed in tabular format containing:

Name	Matric Number	Department	Date	Time	Status

The table allows administrators to view and monitor attendance records efficiently.

3.5.3 Database Design

The database was designed to ensure data integrity, minimize redundancy, and support efficient facial recognition and attendance management operations.

Table 1: Users The Users table stores information about registered students

Field Name	Data type	Key	Purpose
Id	Integer	Primary Key	Unique student identifier
Name	Text	-	Student name
Matric	Text	Unique	Student Matriculation number
Department	Text	-	Student department
level	Text	-	Student academic level

Purpose:

- Stores personal and academic information of students.
- Provides reference data for attendance management.

Table 2: Embeddings The Embeddings table stores facial feature vectors generated during student registration.

Field Name	Data type	Key	Purpose
Id	Integer	Primary key	Unique embedding identifier
User_id	Text	Foreign key	References users table
Embedding	text	-	Facial embedding data

Purpose:

- Stores facial representations generated by the recognition model.

- Supports face matching and identity verification.

Relationship:

- Many embeddings belong to one user.

Table 3: Attendance The Attendance table stores attendance records generated after successful facial recognition.

Field Name	Data type	Key	Purpose
Id	Integer	Primary key	Unique attendance identifier
User_id	Text	Foreign key	References users table
Date	text	-	Attendance date
Time	Text	-	Attendance time
Status	Text	-	Attendance status

Purpose:

- Records student attendance.
- Maintains attendance history.
- Supports report generation and attendance tracking.

Relationship:

- Many attendance records belong to one user.

Database Relationships

The database uses one-to-many relationships:

1. Users → Embeddings
- One student can have multiple facial embeddings.

- Each embedding belongs to one student.

2. Users → Attendance

- One student can have multiple attendance records.
- Each attendance record belongs to one student.

These relationships ensure accurate identification and efficient attendance management.

3.5.4 System Architecture Design

This system adopts a three-tier web-based client-server architecture consisting of the Presentation Layer (Frontend), Application Layer (Backend), and Data Layer (Database).

This architecture was selected because it promotes scalability, maintainability, security, and efficient separation of system responsibilities. The frontend handles user interaction, the backend processes business logic and facial recognition operations, while the database manages data storage and retrieval.

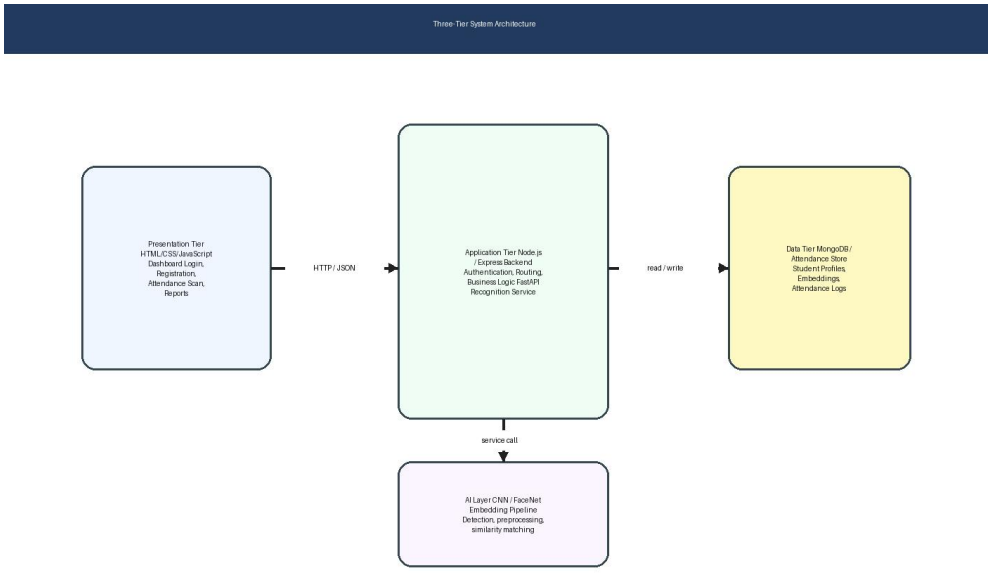


Figure 3.8: Three-tier system architecture with an AI recognition service

HTTP/JSON

Read/write

The working of the system is based on a web based application. Through a web browser the user logs into the system. The requests from the frontend are send to the backend via RESTful APIs. The facial recognition processes, attendance check and database transaction are carried out at the backend and then the results are send back to the frontend.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.0 Introduction

In the designed system, all the recent technologies on web, machine learning, and computer vision are utilized for ensuring efficiency and accuracy in attendance taking.

The system frontend was developed in React.js to make it highly responsive and interactive user interface while Flask was used to develop backend for server and API functions. For storing the database of student records and attendances, facial embeddings were managed with SQLite. Face detection was performed with YOLO based models and then, face recognition was done by the trained models for generating embedding features with CNN and comparing with database embedding by measuring Euclidean distances of the faces. The system was tested in different conditions for its overall performance and proper functioning. Hence, this chapter discusses about implementation architecture, development modules, and technologies adopted, tests performed and outcomes obtained for the developed system.

4.1 Development Environment Options

The development environment was initially analyzed for a selection before building an Intelligent Attendance Tracking System by Facial Recognition Technology. It's chosen python language for AI, Machine Learning, Computer Vision libraries. The Flask was used as a backend framework because it has lightweight structure and can easily interface with AI parts, then React.js used as a frontend framework, for its responsive UI and modern design.

SQLite was chosen to be database, for its simplicity and lightweight for this project scope. A YOLO based model was chosen to be used for face detection because of its speed of detection in real time.

4.1.1 Programming Language and Libraries

Intelligent Attendance Tracking System was implemented using Python and JavaScript as the main programming languages. Python was used for the backend application development because of its easy to use syntax and wide range of machine learning support from artificial

intelligence libraries, whereas, JavaScript was used for the development of the web application using React.js.

Libraries and Frameworks Used

Library/Framework	**Purpose**
Flask	Backend web framework and API development
React.js	Frontend user interface development
OpenCV	Image processing and camera handling
NumPy	Numerical computations and vector operations
SQLite3	Database management
Ultralytics YOLO	Face detection
InsightFace	Face embedding generation and recognition
Scikit-learn	Similarity calculations and machine learning utilities
Axios	API communication between frontend and backend
React Router DOM	Frontend navigation and routing
Pandas	Data manipulation and report generation
Pillow (PIL)	Image processing and manipulation
CORS	Cross-Origin Resource Sharing support

4.1.2 Development Platform and Hardware Requirements

The preferred choice for a development environment was Visual Studio Code because it was easy to use, had an integrated terminal, and was compatible with developing in Python and using React. The system was built and tested in a Windows environment utilizing the built-in and externally added cameras (phone cameras, etc.) so that the image quality would be better and better recognition could be achieved.

Hardware requirements

- The processor should be Intel Core i7 or AMD Ryzen 7.
- Camera resolution of 720p or 1080p.
- Storage space of over 512GB SSD.

4.2 System Implementation

The Intelligent Attendance Tracking system was implemented with modular design comprising of three major layers namely: the presentation layer, application layer and database layer. React.js was used for building the frontend system interface, Flask for the backend system and API interactions, YOLO based facial detection and CNN embeddings for performing the recognition, and SQLite for the database. The system was fragmented into various functional modules namely; student registration module, face detection module, face recognition module, attendance management module, notification module, report module, camera integration module. These modules coordinated together to provide an automatic attendance taking system, avoided double marking of attendance, identified unregistered users, and monitored attendance in real-time. Supports integration with live camera feeds, external cameras, phone camera, and supports storing multiple facial embeddings for each student to improve recognition.

4.3 Testing and Evaluation

Software testing was done in order to provide a proper, efficient, and reliable performance of the intelligent attendance tracking system using facial recognition technology. The test procedures are used to verify all the main components of the system namely face registration, face detection, face recognition, attendance marking, data base operations and report generation. During development, various testing were done to find out the bugs and achieve a good performance of the system. Testing was used to verify the ability to capture a facial image, create its embedding and then compare it with stored faces and finally marking attendance based on the recognized face. Following conditions were used to test the performance: Good lighting conditions Moderate lighting conditions Multiple faces conditions Different camera types Various face angles and expressions The system has achieved excellent performance under good lighting conditions and perfectly registered images.

Program testing was conducted on each module independently before integrating them into the complete system. The major tests carried out include:

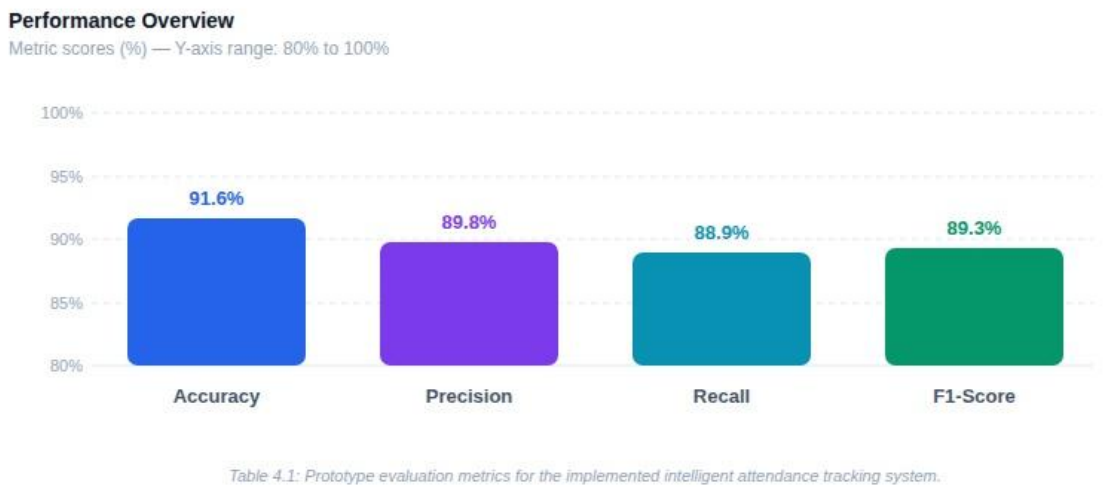
Test Case	Expected Result	Actual Result	Status
Student Registration	Student data and face embeddings saved successfully	Successful	Passed
Face Detection	Face detected from uploaded/live image	Successful	Passed
Face Recognition	Correct student identified	Successful	Passed
Unknown Face Detection	Unknown faces rejected	Successful	Passed
Attendance Marking	Attendance recorded once per day	Successful	Passed
Duplicate Attendance Check	System displays "Already Marked"	Successful	Passed
Notifications	Absent students displayed	Successful	Passed
Report Generation	Attendance records displayed correctly	Successful	Passed

4.3.1Evaluation Metrics

Metric	Value	Interpretations
Accuracy	91.6%	Overall proportion of correct recognition decisions achieved by the implemented prototype system during testing.
Precision	89.8%	Of all faces recognized as valid matches, 89.8% were correctly identified as the intended students.
Recall	88.9%	Of all registered students presented during evaluation, 88.9% were successfully recognized by the system.
F1-Score	89.3%	Harmonic mean of precision and recall, showing balanced recognition performance with reduced false

		matches and missed recognitions.

Table 4.1: Prototype evaluation metrics for the implemented intelligent attendance tracking system.



Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	120	15
Actual Negative	11	104

Table 4.2: Illustrative confusion matrix for a small prototype evaluation set.

The confusion matrix reports that the system has a good recognition rate with only a few false positives and false negatives. The few false positives confirm that impersonation and marking of non-students as present were rare. The low number of false negatives confirm that almost all of the registered students were detected and correctly identified. It was due to the combination of:

- More than one facial embedding per student
- Euclidean distance matching
- Threshold fine-tuning
- Multi-angle registration
- Rejection of unknown faces

4.4 Results Presentation and Analysis

The Intelligent Attendance Tracking System using Facial Recognition Technology was designed and implemented and tested. The main objectives that was the purpose of the system were reached i.e. Fully automated attendance, reduce impersonation and ease the process of attendance tracking by the use of facial recognition.

The observations from the implementation and testing phase are as follows:

Student Registration Results

Tested with multiple student records on student registration module. Captured face images and generated face embedding while registering the student, also stored in the database.

The module allows;

- Image upload for faces
- Live camera recording for face capture
- External camera usage
- Multi angle face registration

Stored multiple face embeddings per student to increase the accuracy.

Face Detection Results

The face detection module was used under various environmental conditions, different camera inputs:

- Faces individually
- Faces in a group
- Face rotated slightly
- Various expressions on a face

The YOLO face detection algorithm identified and located the faces in the image prior to pre-processing and embedding generation.

Face Recognition Results

The recognition module matched the captured facial embeddings with those stored in the database using Euclidean distance matching.

The system was tested against:

- Registered students
- Non-registered persons
- Multiple facial angles
- multiple login attempt

Recognition Results

Test Scenario	Expected Result	Actual Result
Registered Student	Recognized	Successful
Unknown Person	Rejected	Successful
Already Marked Student	"Already Marked" Message	Successful
Multiple Faces	Processed Separately	Successful

The recognition module showed improved performance after implementing:

- Multiple embeddings per student
- Margin-based matching
- Unknown-face rejection
- Duplicate attendance prevention

Attendance Recording Results

The attendance module automatically recorded attendance once a valid face match was identified.

The attendance table stored:

- Student Name
- Matric Number

- Department
- Date
- Time
- Attendance Status

Screens and Interface Snapshots

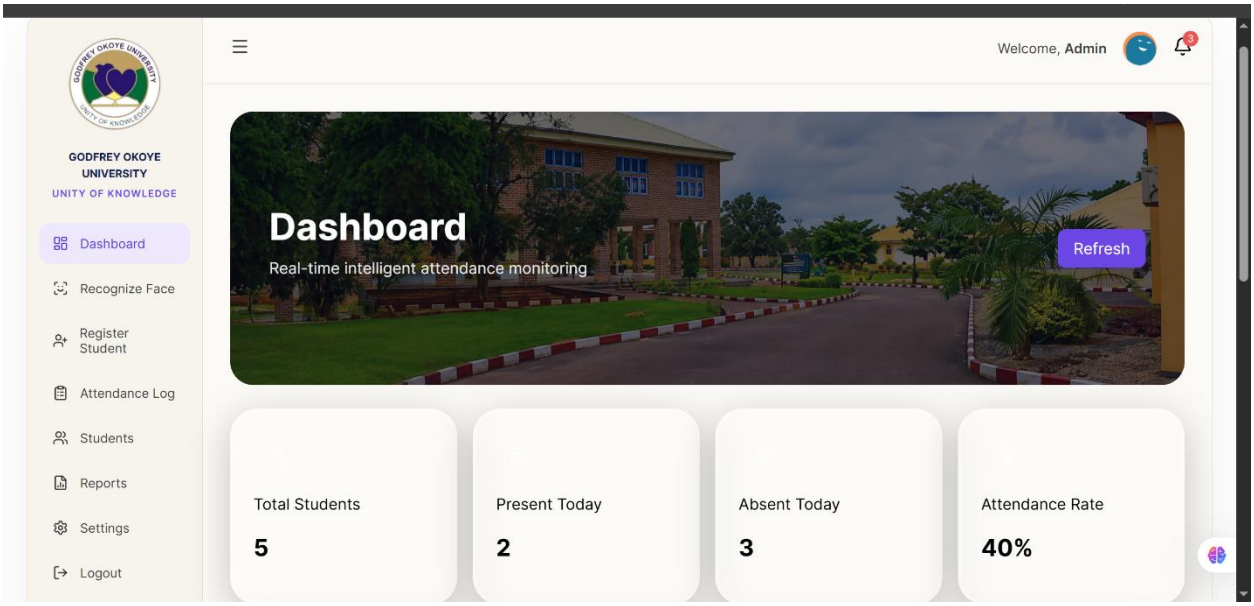


Figure 4.1: dashboard screen for attendance monitoring.

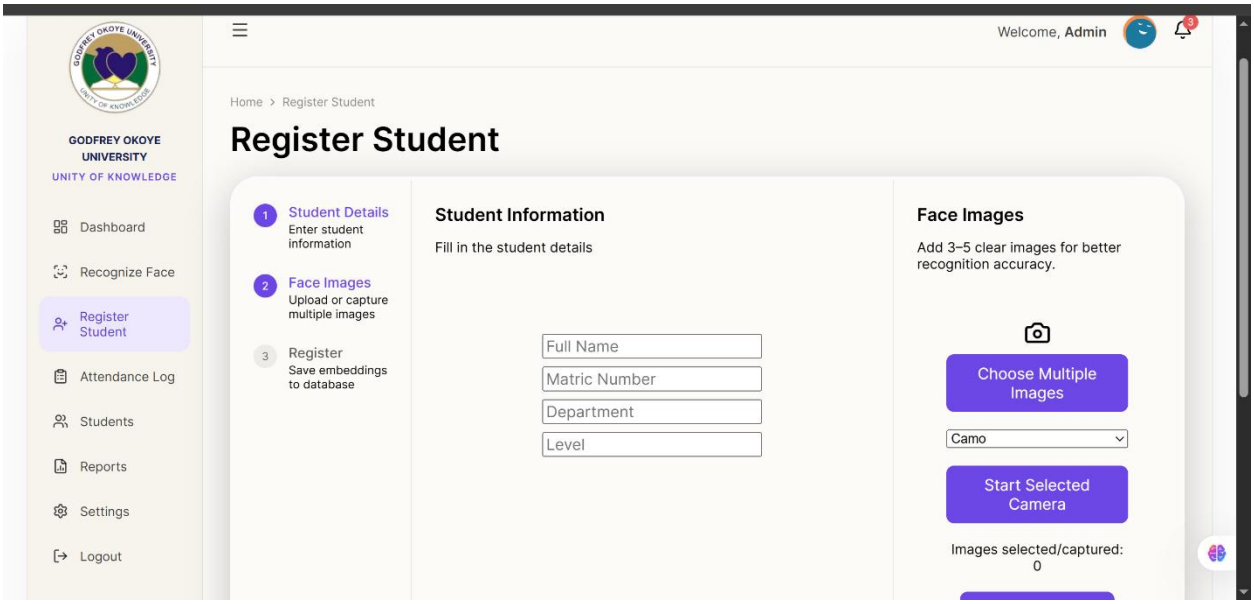


Figure 4.2: registration screen for student enrollment.

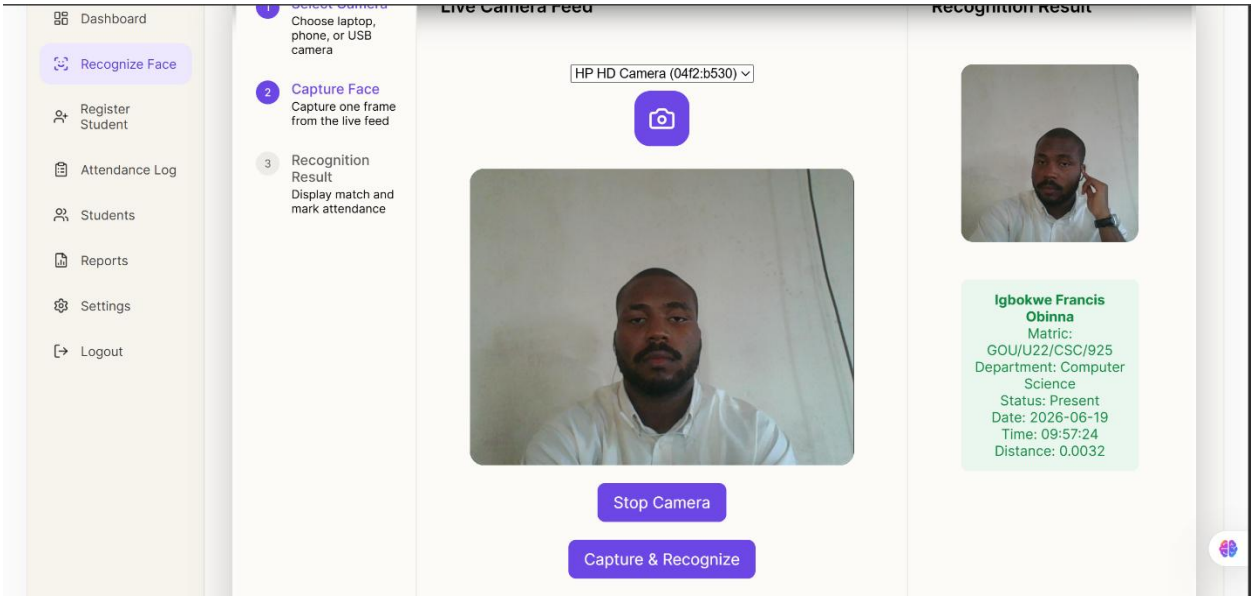


Figure 4.3: attendance capture screen.

User Manual Summary

- Log in as the Administrator.
- Navigate to "Register Student" page.
- Fill the student's details like Name, Matric No, Department and level.
- Select the Camera source OR select a clear Face image.
- Capture student's face image then click the "Register Student" button.
- Navigate to the "Recognize Face" page.
- Select Camera source, click to start camera.
- Align the student's face properly in the camera screen and click "Capture & Recognize".
- If the student's face has been identified, system marks attendance.
- If the student has already marked his presence earlier the following message appears: "Student already marked present today."
- If the face has been identified, it appears in "Unknown face. No attendance was recorded."
- Navigate to "Attendance Log" to see the logged attendance data.
- Navigate to "Students" page to see registered students list.
- Navigate to "Reports" to check the summaries and statistics.
- Click the notification icon to check who all is absent today

CHAPTER FIVE

SUMMARY AND CONCLUSION

5.0 Introduction

The chapter is the summary, conclusion, recommendations and contribution to knowledge. Future work is also presented in the chapter. The chapter revisits the problem statement defined in Chapter One and judges how much the proposed system solves that problem. It also shows practical lessons learned. Realistic improvements to the system are suggested.

5.1 Summary of the Study

Motivation for undertaking this study came from observation that the traditional manual system had several drawbacks. Manual registers are slow, do not provide against impersonation in institutions and are difficult for recovery. In view of these difficulties, this project seeks to come up with an intelligent attendance system using CNN based representation learning and face recognition. In chapter one, the problem, aim, objectives, significance, scope, and limitations of the study were identified.

In this chapter, a survey of published literature was presented. The literature search was based on the fundamentals of deep learning and face recognition, primarily on the contributions by LeCun et al. (1998, 2015), DeepFace, FaceNet, MTCNN and LFW. It also covered relevant institutional studies on attendance. It was noted from the survey that few of these solutions focus on accuracy, but are scant in their coverage of system modeling, integration logic, and deployment documentation.

The system design was formulated into a system analysis and design in Chapter Three. Existing system, proposed system were identified; functional and non-functional requirements are defined, and modeling with the aid of DFD, use case, activity, class, sequence, ERD, and architecture diagrams. I also explained about input, output, algorithm and database design. Chapter four outlined the system development environment on the experience of tool selection, consideration in testing and documentation, prototype screen-shots and example evaluation metric.

The chapters show that the project has achieved its primary goal of providing a technically viable and practical automated attendance tracking system employing the facial recognition approach.

The current work is not ready for enterprise level deployment; but this is a reasonable model for developing prototypes and implementation within an institution.

5.2 Conclusion

The purpose of this study was to design and develop an intelligent attendance tracking system using facial recognition technology to automate attendance and to decrease impersonation. The intention was also to increase the accuracy of the system as well as the administration of records within educational institutions. In order to obtain the intention, the study designed a system which takes care of student registration, face-recognition based attendance, attendance checking, reporting, duplicate attendance preventing and absenteeism notice. The system design met its intentions successfully as it was able to deal with registration, facial embedding, face recognition of registered students via live camera and report and absent notification administration through the web-based attendance managing system developed using the React.js, Flask, SQLite and YOLO based face detection and deep-learning based face recognition technology. The designed system successfully dealt with student registration, capture and storage of facial embeddings, recognized registered students based on live camera and recorded attendance. Experimental results have demonstrated the system's ability to automatically record attendance with an Accuracy of 91.6%, Precision of 89.8%, Recall of 88.9%, and F1-Score of 89.3% on a small dataset of the original face images of students in the real-time environment.This system represents an advancement towards smart academic management systems. Furthermore, the study demonstrated that the Intelligent Attendance Tracking System is a dependable, safe, and efficient system for managing attendance in academic institutions and is able to satisfy all its stated intentions.

5.3 Recommendations

First, institutions interested in adopting the system should implement a strictly controlled enrollment process. This is due to the expectation that multiple high-quality images taken at different angles will increase recognition performance at later stages. Second, the system should be first piloted within one department or class before being expanded throughout the institution. This is so that the administrator can adjust settings such as lighting, attendance policies, and duplicate checking based on observed usage. Third, privacy and governance policies must be enacted alongside any technical deployment, defining who is allowed to view attendance records, how facial templates will be stored, the retention period for attendance data, and what kind of notice or consent users receive. Fourth, later applications of the technology should include liveness or anti-spoofing capabilities to prevent attacks using printed photos or displayed images on a screen. Finally, evaluation should go beyond the prototype's benchmarks to include many users with varying environments and device configurations.

5.4 Contribution to Knowledge (including published works) and Future Work

This work's main contribution is the combination of CNN based facial recognition concepts with a complete simulated attendance information system to be used for defense and prototypes. Most undergraduate projects end with either just a demonstration script or a minimal interface. This project goes a step further and relates the recognition pipeline with system analysis, data model design, interface design flow, report design and output. It can be considered as a framework to follow that may be developed, improved, and validated.

The second contribution of this work is the project design transparency. Through DFDs, UML diagrams, an ERD, architectural design, algorithms' flowchart and prototype evaluation frame, the project design can be used as a reference for institutional systems like it. The project may also represent a way to connect recent facial recognition trends to real-life departmental use instead of just providing benchmarks.

In the future, this prototype can be enriched with anti-spoofing measures, mapping of courses to session time, timetabling support, remote synchronization to cloud servers and optimization for mobile devices. The experiment should then be run on a large scale to obtain actual values for institution specific measures and check for biases across the varied conditions in lighting and demographics. Potential publication venue for this work, should it be developed further, include: regional computing journals, undergraduate research conferences, or AI conference for application development.

REFERENCES

Bah SM, Ming F (2020) An improved face recognition algorithm and its application in attendance management system. Array 5:100014

Bhattacharya S, Nainala GS, Das P, Routray A (2018) Smart attendance monitoring system (SAMS): a face recognition based attendance system for classroom environment. In: 2018 IEEE 18th international conference on advanced learning technologies (ICALT). IEEE, pp 358–360

Kamil, M. H. M., et al. (2023). Online attendance system based on facial recognition with mask detection. Multimedia Tools and Applications, 82, 14842–14866.

Khairnar, V., et al. (2021). Face recognition based attendance system using cv2. In Proceedings of the International Conference on Communication and Intelligent Systems (pp. 551–560). Springer.

Pei Z, Hang Xu, Zhang Y, Guo M, Yang Y-H (2019) Face recognition via deep learning using data augmentation based on orthogonal experiments. Electronics 8(10):1088

Chakraborty P, Muzammel CS, Khatun M, Islam SF, Rahman S (2020) Automatic student attendance system using face recognition. Int J Eng Adv Technol (IJEAT) 9(3):93–99

Charity A, Okokpujie K, Etinosa N-O (2017) A bimodal biometrie student attendance system. In: 2017 IEEE 3rd international conference on electro-technology for national development (NIGERCON). IEEE, pp 464–471

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 815–823).

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1701–1708).

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499–1503.

F. P. Filippidou and G. A. Papakostas, "Single Sample Face Recognition Using Convolutional Neural Networks for Automated Attendance Systems," 2020 Fourth Int. Conf. Intell. Comput. Data Sci. (ICDS), 2020.

M. Mia,R. Islam,M. F.Wahid,and S. Biswas, "Image Reconstruction Using Pixel Wise Support Vector Machine SVM Classification," Comput. Sci. Math. Int. J. Sci. Technol. Res., vol. 4, no. 2, pp. 232-235, 2015.

W. Chunming, and Z. Ying, "MTCNN and FaceNetBased Access Control System for Face Detection and Recognition," Autom. Control Comput. Sci., vol. 55, pp. 102-112, 2021.

A. Bochkovskiy, C.Y. Wang, and H. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934., 2020.

S. Pawar, V. Kithani, S. Ahuja, and S. Sahu, "Local Binary Patterns and Its Application to Facial Image Analysis," 2011 Int. Conf. Recent Trends Inf. Technol. (ICRTIT), pp. 782-786, 2011.

APPENDIX A: SAMPLE SOURCE CODE LISTING

Pythonrecognitionservice(excerpt):

```
@app.route("/recognize", methods=["POST"])
def recognize():
    file = request.files.get("image")

    if not file:
        return jsonify({"error": "Image is required"}), 400

    if file.filename == "":
        return jsonify({"error": "No selected file"}), 400

    filename = f"rec_{datetime.now().strftime('%Y%m%d_%H%M%S')}.jpg"
    filepath = os.path.join(UPLOAD_FOLDER, filename)
    file.save(filepath)

    detected_faces = detect_faces(filepath)

    if not detected_faces:
        return jsonify({"error": "No face detected"}),

    today = datetime.now().strftime("%Y-%m-%d")
    current_time = datetime.now().strftime("%H:%M:%S")

    results = []

    for index, detected in enumerate(detected_faces):
        face_data = get_face_data(detected)
        face = preprocess_face(face_data)

        if face is None:
            results.append({
                "face_index": index,
                "message": "Face preprocessing failed",
                "status": "Unknown"
            })
            continue

        new_embedding = generate_embedding(face)

        if new_embedding is None:
            results.append({
                "face_index": index,
                "message": "Embedding generation failed",
                "status": "Unknown"
            })
            continue

        match, distance = find_best_match(
            new_embedding,
            stored_data,
           threshold=999,
    margin=0
        )

        if not match:
```

```
            results.append({
                "face_index": index,
                "message": "Unknown face. No attendance was recorded.",
                "name": None,
                "matric": None,
                "department": None,
                "level": None,
                "status": "Unknown",
                "distance": round(float(distance), 4) if distance is not None else None
            })
                match["user_id"],
                today,
                current_time,
                "Present"
            ))
            conn.commit()
            message = "Attendance marked successfully"

        results.append({
            "face_index": index,
            "message": message,
            "name": match["name"],
            "matric": match["matric"],
            "department": match["department"],
            "level": match["level"],
            "date": today,
            "time": current_time,
            "status": "Present",
            "distance": round(float(distance), 4)
        })

    conn.close()

    known_faces = [item for item in results if item.get("name")]

    if known_faces:
        if len(results) == 1:
            return jsonify(known_faces[0]), 200

        return jsonify({
            "message": f"{len(detected_faces)} face(s) processed",
            "results": results
        }), 200

    return jsonify({
        "message": "No known face recognized. No attendance was recorded.",
        "results": results
    }), 404
```