Next-Gen Attendance System

A Project Report submitted in the partial fulfillment of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

B. Indu	**(21471A05E5)**
K. Abhinaya	**(21471A05G9)**
T. Tejaswini	**(21471A05L0)**

Under the esteemed guidance of

Dr. S.V.N Sreenivasu M.Tech., Ph.D.
Dean R&D,Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET
(AUTONOMOUS)
Accredited by NAAC with A+ Grade and NBA under Tyre -1
NIRF rank in the band of 201-300 and an ISO 9001:2015 Certified
Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada
KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601

2024-2025

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name **"Next-Gen Attendance System"** is a bonafide work done by the team **B. Indu (21471A05E5), K. Abhinaya (21471A05G9), T. Tejaswini (21471A05L0)** in partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE

Dr. S.V.N Sreenivasu, M.Tech., Ph.D.,
Dean R&D, Professor

PROJECT CO-ORDINATOR

Dr. M. Sireesha, M.Tech., Ph.D.,
Assoc. Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled **"Next-Gen Attendance System"** is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

B. Indu	(21471A05E5)
K. Abhinaya	(21471A05G9)
T. Tejaswini	(21471A05L0)

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman Sri **M. V.Koteswara Rao,** **B.Sc.,** who took keen interest in us in every effort throughout this course.We owe our sincere gratitude to our beloved principal **Dr. S. Venkateswarlu,** **Ph.D.,** for showing his kind attention and valuable guidance throughout the course.

We express our deep-felt gratitude towards **Dr. S. N. Tirumala Rao,** **M.Tech., Ph.D.,** HOD of CSE department and also to our guide **Dr. S.V.N Sreenivasu,** **M.Tech.,Ph.D.,**Dean R&D Professor of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks to **Dr. M. Sireesha,** **M.Tech., Ph.D.,** Associate professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant sourceof inspiration for us throughout this project work.

We extend our sincere thanks to all the other teaching and non-teaching staff in the department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who were involved in giving valuable suggestions clarified our doubts, which really helped us in successfully completing our project.

By

B. Indu (21471A05E5)

K.Abhinaya (21471A05G9)

T. Tejaswini (21471A05L0)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as researchfor the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern softwaretools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-basedsystems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems relatedto academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and cancommunicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, andsynthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engagein independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		√											√		
C421.2	√		√		√								√		
C421.3				√		√	√	√					√		
C421.4			√			√	√	√					√	√	
C421.5					√	√	√	√	√	√	√	√	√	√	√
C421.6									√	√	√		√	√	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2, C22L3.2	Developing an automated attendance system using facial recognition with CNN and advanced image processing.	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process modelis identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implemented and deployed for automated attendance marking, with future updates planned for detecting spoofed faces or forged attendance attempts.	PO4, PO7
C32SC4.3	The physical design includes a website to verify attendance authenticity and detect spoofed faces.	PO5, PO6

ABSTRACT

The Next-Gen Attendance System is an advanced automated solution designed to streamline attendance management in educational institutions. Utilizing the YOLOv8 model, a state-of-the-art deep learning algorithm, it ensures high accuracy in detecting and labelling faces while preventing duplicate entries across multiple sections. The system offers three distinct attendance modes—real-time webcam feeds, pre-recorded video processing, and static image analysis—providing flexibility for various classroom settings. By integrating Convolutional Neural Networks (CNNs), it enhances facial and eye recognition accuracy, ensuring reliable detection despite motion or lighting variations. Attendance data is securely stored in Excel sheets with unique identifiers, eliminating duplication and simplifying administrative tasks. Automating this process significantly reduces manual effort and human error, making attendance tracking seamless for educators. The system's adaptability caters to diverse teaching environments, ensuring precise attendance monitoring regardless of setup. Its deep learning-based approach allows real-time and batch processing, improving efficiency in educational workflows. The combination of YOLOv8 and CNNs ensures robustness, making it suitable for dynamic environments. This innovative tool enhances classroom management, providing error-free tracking and optimizing instructional time. By leveraging AI, it sets a new benchmark for attendance systems, reducing administrative workload and improving accuracy. The system also supports scalability, making it applicable to various educational levels. Designed to handle real-time and post-processing scenarios, it efficiently adapts to institutional needs. The Next-Gen Attendance System exemplifies AI-driven advancements in education, revolutionizing attendance tracking with precision and automation.

INDEX

LIST OF FIGURES

1. INTRODUCTION

The Next-Gen Attendance System is a modern way to track attendance, replacing old-fashioned methods like roll calls and sign-in sheets. These traditional methods can be slow, inaccurate, and take up valuable classroom time. Our system uses advanced deep learning technology to make attendance tracking quick, easy, and reliable.

Attendance tracking is a crucial aspect of any educational institution, ensuring that students are present for their lessons and providing accurate records for administrative purposes. However, conventional methods such as manually calling out names or signing attendance sheets often lead to inefficiencies, human errors, and even fraudulent practices like proxy attendance[1]. Additionally, these methods consume valuable instructional time that could be better utilized for teaching. Recognizing these challenges, the Next-Gen Attendance System has been developed to provide a highly automated and efficient solution for managing attendance.

At the core of this system is YOLOv8, a powerful AI model that detects faces quickly and accurately. This model can recognize multiple faces at the same time, even in challenging conditions like different lighting, angles, and movement. Because YOLOv8 processes images in one go, it works fast, making it ideal for busy classrooms. To improve face recognition, the system also uses Convolutional Neural Networks (CNNs)[2]. These networks help analyze facial features, making sure each student is correctly identified. By using pre-trained CNN models, the system ensures accurate recognition, even in different environments.

The system is flexible and works in three different ways. It captures attendance in real time using a webcam, processes pre-recorded videos to identify students, and extracts attendance from uploaded pictures. This means teachers can choose the best option depending on the classroom setting, whether it's in-person or online. Such flexibility allows the system to cater to diverse learning environments, including traditional classrooms, remote learning setups, and hybrid models where students may attend some sessions virtually and others in person.

The system has a simple, user-friendly design, making it easy for teachers and administrators to manage attendance. The attendance records are automatically stored in Excel files, with separate sheets for different classes. Each student is uniquely identified to prevent duplicate entries, ensuring accuracy. This removes the need for manual work and reduces mistakes. The ability to generate well-structured digital records simplifies administrative tasks, allowing institutions to maintain better documentation and retrieve past attendance logs efficiently.

Moreover, the system is designed with scalability in mind. It can be used in schools, colleges, or even large universities. It is built to handle different classroom sizes and can be expanded with new features in the future. Whether used in a small class or a large campus, the system adapts to meet different needs. By supporting modular updates and integration with other educational management systems, this attendance tracking system offers a long-term solution that can evolve with technological advancements.

Another significant advantage of the Next-Gen Attendance System is its ability to enhance security and privacy[3]. Since the system uses AI-based face recognition, it eliminates the risks of students marking attendance on behalf of their peers. This ensures that only the actual students present in the classroom are counted. Additionally, the system's data encryption and access controls prevent unauthorized users from tampering with attendance records, providing a secure and reliable way to maintain student data.

The implementation of this AI-powered attendance system also reduces administrative burden. In traditional setups, teachers and staff spend considerable time verifying attendance records, cross-checking data, and manually correcting errors. With the automated system, these tasks are streamlined, allowing educators to focus more on teaching rather than administrative duties. The real-time reporting feature also enables institutions to generate instant attendance reports, monitor student participation trends, and make data-driven decisions to improve academic performance and engagement.

By using AI technologies like YOLOv8 and CNNs, this system makes attendance tracking easier and more efficient. It saves time, reduces errors, and helps teachers focus more on teaching instead of paperwork. This innovation shows how artificial intelligence can improve everyday tasks, creating smarter and more connected classrooms. Furthermore, the integration of deep learning ensures that the system continuously improves with exposure to more data, making it even more precise and reliable over time.

As educational institutions embrace digital transformation, automated attendance tracking systems like this one play a key role in optimizing administrative processes. They enhance security by reducing fraudulent attendance practices, increase efficiency by eliminating manual work, and contribute to an overall more productive learning environment. The Next-Gen Attendance System sets a benchmark for how AI-driven solutions can be leveraged to modernize education, ensuring that institutions can focus on delivering high-quality instruction without being bogged down by routine administrative tasks.

The Next-Gen Attendance System represents a major leap forward in the way attendance is tracked in educational institutions. By leveraging cutting-edge deep learning technologies, the system not only ensures efficiency and accuracy but also enhances security and reduces administrative workload. As technology continues to advance, AI-driven solutions like this will play an increasingly important role in shaping the future of education. This system is a step towards smarter, more automated classrooms, where technology supports learning in the most efficient and innovative ways possible.

The rapid advancement of artificial intelligence (AI) and deep learning has transformed various industries, including education. With institutions increasingly adopting digital tools to streamline operations, the need for automated and intelligent attendance tracking systems has never been more critical. The Next-Gen Attendance System not only addresses traditional inefficiencies but also aligns with the broader shift toward smart classrooms. By integrating cutting-edge AI models, this system provides a seamless, contactless, and efficient method for tracking attendance, making it a valuable asset for educational institutions striving for modernization.

Beyond its primary function of attendance tracking, this system also lays the foundation for future enhancements in student monitoring and academic performance analysis. By leveraging AI, institutions can explore additional features such as real-time student engagement tracking, behavior analysis, and performance prediction based on attendance patterns. As AI-driven solutions continue to evolve, the Next-Gen Attendance System serves as a stepping stone toward a more connected and intelligent education ecosystem, ensuring that both educators and students benefit from technological advancements in meaningful ways.

1.1 MOTIVATION

In today’s digital era, technological advancements have transformed traditional processes across various domains, including education and corporate environments. One such critical process is attendance management. Traditional attendance-taking methods, such as manual roll calls and RFID-based systems, often result in inefficiencies, errors, and opportunities for manipulation, such as proxy attendance. The need for a more secure, efficient, and automated approach has driven the development of intelligent attendance systems using artificial intelligence (AI) and deep learning.

The Next-Gen Attendance System aims to revolutionize attendance tracking by leveraging YOLOv8-based face recognition for real-time detection and verification. Unlike conventional biometric systems that require physical contact (fingerprint scanners) or RFID-based approaches that can be misused, our system offers a seamless, contactless, and highly accurate solution. By harnessing deep learning techniques, the system ensures real-time face detection and recognition, reducing manual effort and increasing reliability.

The motivation behind this project stems from the growing demand for automation in educational institutions, workplaces, and other sectors where attendance plays a crucial role in productivity and accountability. With the rise of remote learning, hybrid work environments, and large-scale organizations, an AI-powered attendance system eliminates inefficiencies while maintaining robust security. Additionally, it minimizes administrative workload, providing educators and managers with an easy-to-use solution for tracking attendance records without manual intervention.

Our Next-Gen Attendance System is designed with three operational modes—live webcam detection, pre-recorded video analysis, and static image processing—offering flexibility for different use cases. It ensures that each individual is marked present only once per day, preventing duplicate entries. The project’s motivation is rooted in the desire to make attendance marking seamless, highly secure, and future-proof by integrating deep learning into an everyday administrative process.

1.2 PROBLEM SOLVING

The process of recording attendance is crucial in academic institutions, corporate environments, and public sector organizations. However, existing attendance systems face numerous challenges and limitations that hinder their effectiveness. Manual attendance methods, such as roll call and sign-in sheets, are time-consuming, prone to human error, and susceptible to proxy attendance. Automated systems, such as RFID cards and biometric fingerprint scanners, offer improvements but come with their own set of drawbacks, including physical contact requirements, card loss issues, and security vulnerabilities.

A major challenge faced by traditional attendance systems is the lack of real-time accuracy and adaptability. RFID-based systems can be manipulated by unauthorized individuals carrying multiple ID cards, and biometric fingerprint scanners require direct contact, which raises hygiene concerns, especially in the post-pandemic era. Furthermore, these systems fail to provide remote attendance solutions, making them less effective in hybrid or remote work and learning setups.

To address these issues, our project introduces an AI-powered attendance system using YOLOv8-based face detection. Unlike conventional methods, our system operates in three modes—live webcam, video processing, and static images—to provide real-time and retrospective attendance tracking. By leveraging deep learning models, our system ensures high accuracy, security, and efficiency, eliminating the need for physical interaction while maintaining reliable attendance records.

The proposed Next-Gen Attendance System focuses on the following key problem areas:

- Proxy attendance prevention: Eliminating the possibility of fraudulent attendance marking.
- Time efficiency: Reducing the time taken for attendance marking in classrooms and offices.
- Accuracy and reliability: Using deep learning for precise face detection and recognition.

- Automation and ease of use: Minimizing human intervention and simplifying the attendance process.
- Scalability: Adapting the system for large classrooms, workplaces, and remote environments.

By addressing these problems, the Next-Gen Attendance System aims to provide an innovative, secure, and efficient solution for attendance tracking, ensuring seamless operation in diverse environments.

1.3 OBJECTIVE

The Next-Gen Attendance System is developed with the goal of automating attendance tracking using AI-driven face recognition technology. The system is designed to address inefficiencies in traditional attendance methods by leveraging deep learning models to detect and recognize faces in real time. The key objectives of this project are as follows:

I. Automate Attendance Using AI

- Implement a contactless attendance system using YOLOv8 for real-time face detection.
- Ensure high accuracy and efficiency in recognizing faces, even under varying lighting and background conditions.
- Reduce the manual workload for teachers, administrators, and corporate professionals.

II. Provide Multiple Attendance Modes

- Allow users to mark attendance using live webcam detection, pre-recorded video processing, or static image uploads.
- Ensure flexibility to accommodate both in-person and remote attendance scenarios.
- Enable seamless integration into existing institutional and corporate frameworks.

III. Prevent Proxy Attendance

- Use deep learning models to differentiate between individuals and eliminate fraud.
- Implement a mechanism to ensure that each individual can be marked present only once per day.
- Enhance security by utilizing facial feature extraction and biometric authentication techniques.

IV. Optimize Data Storage and Management

- Store attendance records in separate Excel sheets for different sections/classes.
- Enable easy retrieval, tracking, and exporting of attendance reports.
- Provide an efficient database system to ensure smooth access and organization of records.

V. Ensure Scalability and Future Integration

- Design the system to be scalable, allowing its deployment in large institutions, corporate offices, and hybrid environments.
- Allow future integration with cloud storage, mobile apps, and additional AI-based security features such as emotion recognition or gaze tracking.
- Support multi-user access for teachers, managers, and HR professionals to track attendance efficiently.

By achieving these objectives, the Next-Gen Attendance System aims to enhance efficiency, accuracy, and security in attendance tracking while reducing administrative burdens and streamlining record-keeping processes. The system is a future-ready solution designed to meet the evolving needs of modern educational institutions and workplaces.

2. LITERATURE SURVEY

Hidayat et al. (2024) [1] present an in-depth analysis of modern technological advancements aimed at enhancing classroom management through real-time monitoring systems. Their research focuses on the use of webcams for movement detection, which allows institutions to track student activities and send alerts to staff when motion is detected. This system ensures that educators can monitor engagement levels and identify unusual behaviors, helping maintain discipline and focus in the classroom.

A key component of their study is the implementation of facial landmark tracking, segmentation, and gaze prediction. These methods provide insights into student behavior, helping institutions assess attention levels, emotional states, and engagement with classroom activities. By leveraging AI-driven facial recognition, the system can track how long a student remains engaged with the lesson.

Hidayat et al. [1] also highlight the integration of attendance systems into these monitoring tools. Their study discusses how face recognition technology can replace manual attendance systems, ensuring seamless record-keeping and minimizing human errors. They emphasize state-of-the-art technologies, including deep learning-based image analysis frameworks, which improve the efficiency and usability of classroom management tools.

Their research serves as a strong foundation for AI-driven real-time classroom monitoring, addressing challenges related to student engagement, security, and attendance automation. This study aligns closely with the objectives of the Next-Gen Attendance System, particularly in its use of webcams, AI-based motion tracking, and facial analysis to streamline attendance marking and classroom management.

Alruvais and Zakarih [2] explore how automatic technologies can be applied to e-learning and virtual classrooms for enhanced monitoring and student engagement analysis. Their work focuses on the location tracking of students in online learning environments, ensuring that pupils are actively present during virtual sessions.

One of their key contributions is the use of deep learning methods, particularly CNNs, for facial expression recognition, eye tracking, and head movement analysis. By analyzing these behavioral patterns, their system can detect levels of attentiveness and participation in an online setting.

Their research also highlights the challenges in deep learning applications for e-learning monitoring, including:

- **High data requirements**: Training CNN models for real-time tracking demands large datasets of student faces under different conditions.
- **Accuracy issues**: Enhancing tracking precision for gaze estimation and facial expression recognition remains a challenge.
- **Scalability concerns**: Deploying AI-driven monitoring systems across large-scale virtual classrooms requires robust computing infrastructure.

Alruvais and Zakarih[2] also discuss the integration of multi-modal learning analytics with their AI-based tracking system. By combining facial recognition data with additional biometric and behavioral indicators**,** such as voice analysis, typing patterns, and interaction frequency**,** they propose a comprehensive framework for student engagement assessment**.** This holistic approach ensures that student participation is not only measured through visual cues but also through verbal and interactive behavior**,** providing a more accurate picture of engagement in virtual classrooms. Furthermore, they emphasize the potential of adaptive learning systems, where AI dynamically adjusts the difficulty level or instructional content based on a student's detected engagement level, ultimately fostering a personalized and optimized learning experience**.**

Khwala Alhanai's[3] research primarily investigates transfer learning techniques to enhance face recognition-based attendance systems. The study explores how pre-trained CNN architectures, including SqueezeNet, GoogleNet, and AlexNet, can be leveraged to increase the accuracy of attendance management systems.

- **Pre-trained models improve efficiency**: Instead of training CNNs from scratch, using transfer learning on established networks significantly enhances accuracy and computational speed.
- **Comparative analysis of CNNs**: The study examines how different CNN architectures perform under varied lighting conditions, occlusions, and facial variations.
- **Integration with attendance tracking**: The study proposes a system where face recognition is integrated with a database to automatically update attendance logs in real-time.

Additionally, Alhanai's[3] research delves into the importance of data augmentation in improving face recognition accuracy. By introducing techniques such as rotation, flipping, brightness adjustments, and noise addition, the study ensures that CNN models generalize well to different real-world conditions. This is particularly useful in classroom environments where lighting conditions and facial angles vary significantly. The research highlights that training models on diverse datasets with augmented samples helps mitigate issues related to poor lighting, occlusion, and slight facial changes, ultimately leading to a more robust and reliable attendance system.

Furthermore, Alhanai[3] emphasizes the importance of reducing computational overhead in real-time face recognition applications. Traditional deep learning models can be resource-intensive, making them challenging to deploy on edge devices such as webcams or low-power computers in classrooms. The study proposes optimization strategies, including pruning unnecessary layers in CNN models, quantization techniques to reduce model size, and implementing efficient inference methods. These improvements not only enhance processing speed but also make the system more scalable for large educational institutions. By aligning these strategies with the Next-Gen Attendance System, the research contributes valuable insights toward creating a fast, accurate, and deployable face recognition-based attendance tracking solution.

Arya, Mesaria, and Parekh [4] developed a Smart Attendance System using CNNs for real-time face recognition in academic institutions. Their research focuses on Siamese Networks, a specialized deep learning model used for one-shot learning and identity verification.

- **Live camera feeds**: Attendance is marked automatically through continuous face recognition via live webcam streaming.
- **Siamese Networks for accuracy enhancement**: This architecture enables better distinction between similar faces, improving identification rates in large classrooms.
- **MongoDB integration**: The system stores real-time attendance data in a NoSQL database, ensuring scalability and efficient retrieval.

Furthermore, their research highlights the importance of liveness detection in ensuring the integrity of the attendance system. Traditional face recognition models are often vulnerable to spoofing attacks, where students attempt to mark attendance using images or videos of others. To counter this, Arya, Mesaria, and Parekh incorporated liveness detection techniques such as blink detection, head movement tracking, and texture analysis to differentiate between live faces and spoof attempts. By integrating these mechanisms, the system ensures that only genuine, present individuals are marked for attendance, enhancing security and reliability.

Additionally, the study explores the system's adaptability to various educational settings, ranging from small classrooms to large lecture halls. They tested different camera angles, lighting conditions, and real-time processing capabilities to evaluate system performance. Their findings suggest that multi-angle face detection and dynamic camera adjustments can further improve accuracy in attendance tracking. Moreover, by leveraging cloud-based computing with MongoDB, the system is scalable, allowing multiple institutions to deploy it with minimal hardware upgrades. These insights are instrumental for the Next-Gen Attendance System, reinforcing its goal of providing a highly accurate, real-time, and automated attendance tracking solution using advanced deep learning techniques.

Lakshmi, Kumaraswamy, and Manhar[5] introduce an automated attendance system combining geo-fencing with facial recognition. Their system ensures that attendance is recorded only when students are physically present within a predefined geographic boundary (classroom, office, etc.).

- **Geo-fencing with GPS and APIs**: Virtual boundaries are set using GPS tracking, ensuring students cannot mark attendance outside designated locations.
- **Machine Learning for facial recognition**: Models like CNNs, VGGFace, and ResNet are utilized for identity verification.
- **Liveness detection**: The system employs anti-spoofing mechanisms to prevent unauthorized access, such as photo-based fraud.

Their approach provides valuable insights for enhancing Next-Gen Attendance Systems, especially in hybrid learning and remote attendance verification.

Furthermore, their study emphasizes the role of real-time data synchronization between the geo-fencing module and the facial recognition system. By integrating cloud-based servers, the system ensures that location verification and face authentication occur simultaneously, reducing the risk of attendance manipulation. The system also employs adaptive thresholding to accommodate minor GPS deviations, preventing false negatives where students are near but slightly outside the designated boundary due to signal fluctuations. This approach enhances the practical usability of geo-fencing in real-world educational and corporate settings.

Additionally, the researchers explore the potential of integrating biometric verification alongside facial recognition to further improve accuracy. Techniques such as voice recognition and fingerprint scanning could act as secondary authentication layers, ensuring that only authorized individuals mark attendance. Their findings suggest that multi-modal biometric authentication, when combined with geo-fencing and machine learning-based facial recognition, can create a highly secure and foolproof attendance system. These insights are crucial for the Next-Gen Attendance System, particularly in expanding its applicability to remote learning environments and corporate workplaces where attendance verification needs to be both accurate and tamper-proof.

3. SYATEM ANALYSIS

3.1 EXISTING SYSTEM

The existing attendance systems, especially in educational institutions and workplaces, primarily rely on manual methods, RFID cards, biometric technologies, or more advanced AI solutions. Traditional methods like roll calls and paper registers, though widely used, are time-consuming and prone to errors. Manual attendance often results in inaccuracies, such as missed entries, and is susceptible to manipulation, such as proxy attendance, where someone marks the presence of another person. This creates inefficiencies, wastes time, and can lead to discrepancies in attendance records, impacting both administrative and payroll processes.

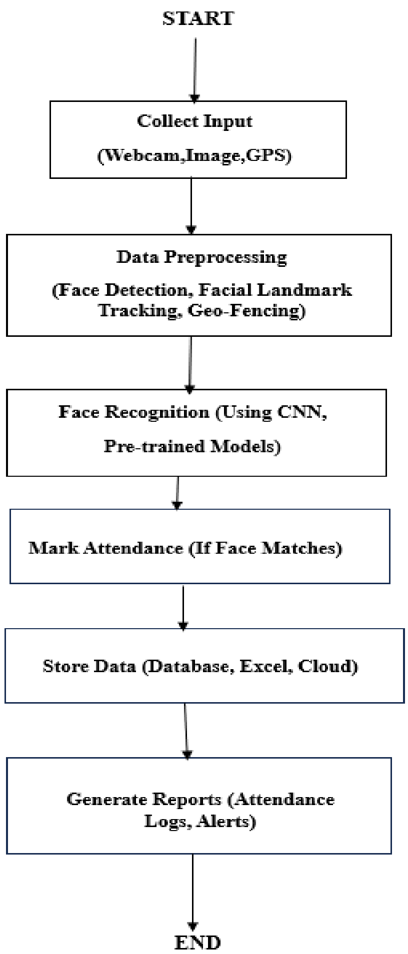


Fig 3.1: FLOWCHART OF EXISTING SYSTEM

To overcome the limitations of manual systems, automated attendance systems have been introduced. RFID-based systems, for example, use smart cards that employees or students scan upon entry. This ensures quicker and more accurate tracking of attendance. However, while these systems reduce human error and time wastage, they still have drawbacks. They require physical infrastructure such as card readers, which can be expensive and prone to malfunction. Additionally, RFID systems are vulnerable to fraud, as cards can be shared among individuals, leading to inaccurate attendance records.

Fig 3.1 Biometric systems, such as fingerprint or facial recognition, have been implemented to address the shortcomings of RFID[6] and manual systems. These systems verify individuals based on unique biological traits, offering a higher level of security and accuracy. Biometric systems are widely adopted because they minimize the risk of proxy attendance and ensure that the right individual is marked present. Despite their advantages, biometric systems are not without challenges. They can be costly to install and maintain, and concerns about privacy and data security are common. Additionally, biometric systems can face issues with false positives or negatives, where the system may fail to accurately recognize an individual due to technical limitations or poor-quality inputs

The most recent developments in attendance systems involve the use of artificial intelligence and machine learning, particularly through facial recognition technology. AI-powered systems use cameras to capture images or video streams of individuals and match them against a database of registered faces. This method offers the benefit of being contactless, fast, and scalable, making it particularly suitable for large classrooms, offices, or events. Facial recognition technology provides a more seamless experience compared to previous methods and eliminates the need for physical devices like cards or fingerprints.

In summary, the existing attendance systems have evolved from manual methods to more automated solutions using RFID, biometric systems, and AI-based technologies. While these systems provide significant improvements in efficiency and accuracy, issues such as high costs, security concerns, and technical limitations still persist. As technology continues to improve, future attendance systems are likely to become even more streamlined, accurate, and secure, addressing the shortcomings of the current systems.

3.2 DISADVANTAGES OF THE EXISTING SYSTEM

Despite advancements in automated attendance systems, the traditional and some existing digital attendance methods still face several limitations that hinder their efficiency, accuracy, and scalability. These drawbacks highlight the need for a more robust and intelligent solution like the Next-Gen Attendance System.

I Manual Dependency and Time Consumption

Traditional roll-call and RFID-based systems are slow and error-prone. Biometric methods like fingerprint scanning cause delays in large classrooms. Physical contact-based systems are not ideal for fast and efficient attendance tracking.

II Risk of Proxy Attendance and Fraud

RFID and manual attendance methods allow proxy marking. Weak face recognition models fail to detect spoofing techniques. Unauthorized access through printed photos or digital images is a major security risk.

III Limited Scalability and Integration Challenges

Existing systems struggle with large-scale environments. Many lack real-time cloud integration for seamless attendance tracking. Managing multiple sections and institutions becomes difficult.

IV Environmental and Hardware Constraints

Face recognition fails in poor lighting and occlusions. Fingerprint scanners are ineffective due to dirt, wet hands, or hardware degradation. Variations in head angles and accessories reduce accuracy.

V Lack of Advanced Analytics and Insights

Current systems only mark attendance without tracking engagement. AI-driven analytics for student participation and classroom behavior are missing. Institutions cannot assess attentiveness effectively.

3.3 PROPOSED SYSTEM

The Next-Gen Attendance System offers a modern solution to the challenges faced by traditional and current automated attendance systems. By integrating advanced technologies such as artificial intelligence (AI), deep learning, and computer vision, the Next-Gen system provides a more efficient, accurate, and secure method for tracking attendance. This system eliminates the limitations of manual entry, card-based tracking, and traditional biometric systems, ensuring a seamless, user-friendly experience for both educational institutions and workplaces.

For model evaluation, accuracy can be determined based on the success rate of the face recognition process, while performance metrics could include the speed of processing and error rates in misidentifying or missing students. Evaluation could also involve testing different input methods [14](webcam, image, video) to ensure robustness across data formats. Additionally, storing the output in Excel provides an organized way to verify the model's efficiency in maintaining attendance records.

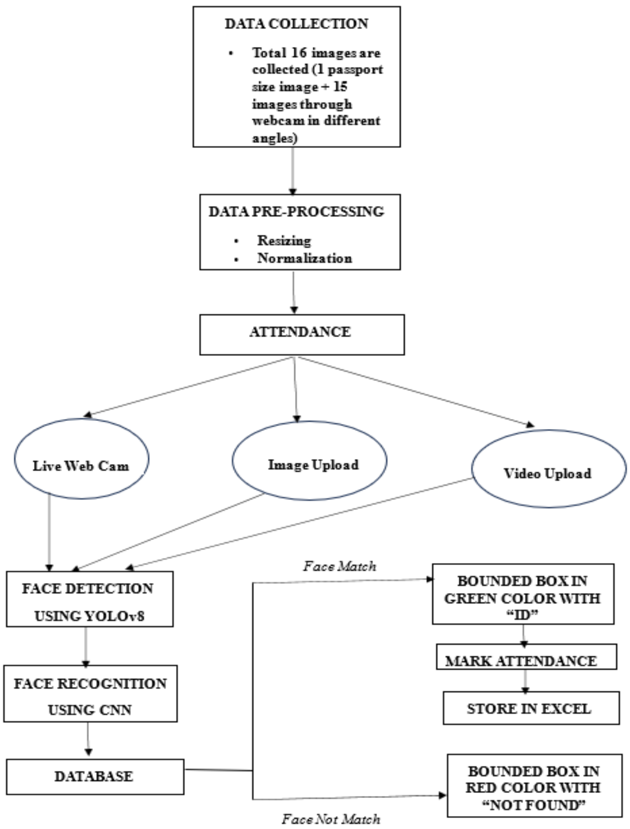


FIG 3.2: ATTENDANCE MANAGEMENT SYSTEM

The flowchart (figure 5.4) represents a model evaluation system for automated attendance marking using face recognition technology[12]. The system provides multiple data input methods such as live webcam feeds, image uploads, or video uploads, allowing flexibility in how attendance data is captured. All inputs are stored in a database where they are processed for face recognition. The model evaluates the facial data to verify the identity of the student. Once recognized, the system marks the attendance of the individual and stores this information in an Excel file for record-keeping. This flow ensures that attendance can be accurately and efficiently[13] monitored without manual intervention.

One of the key features of the Next-Gen Attendance System is its use of facial recognition technology powered by deep learning algorithms. Unlike traditional biometric systems, which rely on fingerprints or RFID cards, the Next-Gen system identifies individuals based on their facial features through cameras placed in classrooms or office spaces. This method is contactless, fast, and eliminates the need for any physical devices or interventions from users, making it ideal for large-scale environments with many people. The system automatically detects and registers faces, reducing the chances of proxy attendance or fraud. The system features real-time tracking for instant and accurate attendance recording.

The Next-Gen Attendance System uses deep learning to analyse video streams and verify identities without manual input. Administrators can instantly access updated records for efficiency. The AI continuously improves accuracy by learning and adapting over time. Seamless integration with existing software enables automated reports, attendance tracking, and notifications. It supports administrative functions like grading, payroll, and performance analysis. Robust security measures ensure encrypted storage, facial data protection, and compliance with privacy standards.

In conclusion, the Next-Gen Attendance System leverages AI, deep learning, and computer vision for accurate and efficient attendance tracking. It eliminates manual processes, reducing fraud and enhancing convenience. The system ensures scalability, security, and reliability for users and administrators. This innovative solution represents the future of attendance management in educational and corporate sectors.

Advantages Over Existing Systems

I. Contactless and Hygienic

Traditional biometric systems such as fingerprint scanners require physical contact, which can lead to hygiene concerns, especially in a post-pandemic world. The Next-Gen Attendance System eliminates this issue by using facial recognition, providing a completely contactless solution.

II. Elimination of Proxy Attendance

Manual attendance systems and RFID-based methods are vulnerable to proxy attendance fraud. With advanced facial recognition and liveness detection, the Next-Gen system ensures that only the actual person can mark attendance, reducing unauthorized entries.

III. Reduced Administrative Effort

Conventional attendance systems require manual record-keeping and data entry, increasing administrative workload. The automated tracking and real-time updates provided by the Next-Gen system reduce human effort, allowing staff to focus on more critical tasks.

IV. High Accuracy and Reliability

Existing attendance tracking methods suffer from errors due to poor scanning, lighting conditions, or manual mistakes. With deep learning-driven face recognition, the Next-Gen system achieves high accuracy, ensuring reliable attendance records.

V. Real-Time Monitoring and Insights

Unlike traditional systems that require periodic updates, the Next-Gen Attendance System provides real-time monitoring, allowing administrators to track attendance instantly. Detailed insights and analytics help in understanding attendance patterns and student engagement.

3.4 FEASIBILITY STUDY

The Next-Gen Attendance System is designed to modernize and enhance attendance tracking through AI-driven facial recognition. Before implementation, it is crucial to assess the feasibility of the system across various dimensions, including technical feasibility, interoperability, and economic feasibility. These factors ensure the system’s effectiveness, adaptability, and cost-efficiency for different institutions and organizations. The feasibility study evaluates the practicality of adopting this system in real-world scenarios, considering infrastructure requirements, integration challenges, and financial implications.

- **Technical Feasibility**

The Next-Gen Attendance System is technically feasible due to advancements in artificial intelligence, deep learning, and computer vision. The system is built using AI-driven facial recognition models, which leverage convolutional neural networks (CNNs) and deep learning frameworks such as TensorFlow and PyTorch. These technologies enable real-time face detection, verification, and recognition with high accuracy. The system's capability to adapt to different lighting conditions, angles, and facial variations ensures smooth operation in various environments.

- **High-resolution cameras** for capturing clear facial images.
- **Robust computing infrastructure** for processing large datasets efficiently.
- **Cloud-based and on-premises solutions** for deployment flexibility.
- **Scalability** to accommodate increasing user demands without performance degradation.
- **Regular updates and AI model improvements** to enhance accuracy and efficiency.

With these technical strengths, the system ensures long-term sustainability and seamless attendance tracking.

- **Interoperability**

Interoperability is a crucial factor in determining the feasibility of the Next-Gen Attendance System. The system is designed to integrate seamlessly with existing digital platforms, including Learning Management Systems (LMS), Human Resource Management Systems (HRMS), and enterprise payroll software. API-based connectivity ensures that attendance records can be synchronized in real time with other applications, streamlining administrative workflows.

- **Database compatibility:** Supports various database management systems for smooth data handling.
- **Remote access:** Cloud-based infrastructure allows monitoring from any location.
- **Hybrid learning support:** Suitable for both physical and remote attendance tracking.
- **Mobile and web-based dashboard:** Enhances usability and accessibility for users.
- **Multi-platform integration:** Connects with existing school, college, and corporate management software.

By enabling interoperability with different database management systems, the Next-Gen Attendance System ensures smooth data handling and retrieval. The cloud-based infrastructure supports remote access, allowing administrators, educators, and employers to monitor attendance from any location.

- **Economic Feasibility**

Traditional attendance tracking methods, such as manual roll calls, RFID-based systems, and biometric fingerprint scanners, involve recurring expenses for maintenance, replacement, and operational labor. In contrast, the facial recognition-based system significantly reduces long-term costs by eliminating the need for physical cards, scanners, and manual intervention.

- **Lower operational costs:** Reduces manual workload and minimizes the need for physical tracking devices.
- **Prevention of fraudulent attendance:** Eliminates proxy attendance, saving financial resources.
- **Flexible deployment:** Cloud-based implementation reduces hardware expenses.
- **Scalability:** Cost-effective for both small and large institutions.
- **High return on investment (ROI):** Automation increases efficiency and reduces human errors, leading to better financial outcomes.

By leveraging cloud computing, the system minimizes hardware costs and allows organizations to pay for only the resources they require. Over time, the system leads to substantial savings and improved operational efficiency, making it a highly viable investment.

3.5 USING COCOMO MODEL

The **Constructive Cost Model (COCOMO)** is a widely used estimation technique for predicting the effort, time, and cost required to develop a software project. To assess the development effort for the **Next-Gen Attendance System**, we apply the COCOMO model based on the project's size and complexity.

COCOMO Model Classification

COCOMO categorizes software projects into three types:

1. **Organic** – Simple, small teams, well-understood requirements.
2. **Semi-Detached** – Moderate complexity, mixed team experience.
3. **Embedded** – Highly complex, requiring strict constraints and regulations.

Given that the Next-Gen Attendance System integrates deep learning, YOLOv8 for face detection, real-time processing, and database management**,** it falls under the Semi-Detached category due to its moderate complexity and the need for advanced AI techniques.

Effort Estimation Using COCOMO

The Basic COCOMO equation is:

$$E = a(KLOC)^b$$

Where:

- **E** = Effort (in person-months)
- **KLOC** = Thousands of lines of code
- **a, b** = Project-specific coefficients (Semi-Detached values: **a = 3.0, b = 1.12**)

If the estimated size of the project is 15 KLOC (15,000 lines of code)**,** the effort required can be calculated as:

$$E=3.0\times(15)^{1.12}$$

$$E\approx 46.5 \text{ person-months}$$

Development Time Estimation

The development time (T) is calculated as:

$$\boldsymbol{T = c(E)^{d}}$$

For a Semi-Detached model, values are **c = 2.5** and **d = 0.35**.

$$\boldsymbol{T = 2.5 \times (46.5)^{0.35}}$$

$$\boldsymbol{T \approx 10.2 \text{ months}}$$

Using the COCOMO model, the Next-Gen Attendance System is estimated to require approximately 46.5 person-months of effort and 10.2 months of development time**.** However, modern agile development practices, pre-trained deep learning models, and existing libraries may optimize this timeline significantly. This estimation helps in better project planning, resource allocation, and budgeting for further enhancements.

4. SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS:

• **Operating System**	Windows 10/11
• **Programming Language**	Python
• **AI/ML Framework**	TensorFlow, Keras, YOLOv8
• **Data Processing**	Pandas, NumPy, OpenCV
• **UI Development**	Tkinter, Flask (optional for web-based UI)
• **Database**	Excel
• **Version Control**	Git, GitHub
• **Development Tools**	VS Code

4.2 REQUIREMENT ANALYSIS

Requirement analysis is a crucial phase in the development of the Next-Gen Attendance System, as it helps in identifying and defining the functional and non-functional aspects necessary for a successful implementation. The system is designed to enhance attendance tracking efficiency through real-time facial recognition, minimizing manual efforts and errors. Functional requirements include face detection and recognition, attendance marking, multiple input support, and real-time feedback mechanisms. On the other hand, non-functional requirements focus on system scalability, security, performance, and usability to ensure a seamless experience. A well-structured requirement analysis ensures that the software meets the intended objectives while maintaining reliability and accuracy in attendance tracking.

Functional Requirements:

a. **Face Detection & Recognition:** Uses YOLOv8 for accurate face detection.
b. **Attendance Management:** Stores attendance records in section-wise Excel sheets.
c. **Multiple Input Modes:** Supports webcam, pre-recorded videos, and static images.
d. **User Authentication:** Only authorized personnel can access the system.
e. **Data Export & Reporting:** Attendance data can be exported in Excel format.

f. **Real-Time Feedback:** Displays a green/red circle for correct/incorrect face positioning.

Non-Functional Requirements:

a. **Scalability:** Handles multiple students in real-time.
b. **Performance:** Detects faces efficiently within seconds.
c. **Security:** Protects data from unauthorized access.
d. **Usability:** Simple and intuitive UI.
e. **Reliability:** Works under different lighting conditions.

4.3 HARDWARE REQUIREMENTNS

Component	Minimum Requirement	Recommended Requirement
• **Processor**	Intel Core i5 (8th Gen) / AMD Ryzen 5	Intel Core i7/i9 (10th Gen) / AMD Ryzen 7/9
• **RAM**	8GB DDR4	16GB+ DDR4
• **Storage**	256GB SSD	512GB SSD+
• **Graphics Card**	Integrated GPU	NVIDIA RTX 3060+
• **Camera**	720p webcam	1080p HD webcam
• **Additional Peripherals**	External storage, microphone	External storage, microphone

4.4 SOFTWARE

The Next-Gen Attendance System is powered by a range of software tools and technologies that enable seamless and efficient AI-based attendance tracking. The system operates on Windows and Linux environments, utilizing Python as the core programming language for its machine learning and data processing functionalities. Key AI frameworks such as TensorFlow and Keras support deep learning-based face recognition, while YOLOv8 ensures high-precision real-time face detection.

For data handling, libraries like Pandas and NumPy are used to manage and process attendance records effectively. OpenCV plays a crucial role in image processing and facial recognition tasks. The user interface is designed using Tkinter for standalone applications, with Flask as an optional framework for web-based implementations. Data storage and management are facilitated by Excel, allowing easy retrieval and export of attendance records. Version control is maintained using Git and GitHub, ensuring smooth collaboration and code management. Integrated development environments like VS Code aid in efficient software development and testing, making the system robust and user-friendly.

4.5 SOFTWARE DESCRIPTION

System Overview:

The Next-Gen Attendance System is a deep learning-based application that automates attendance marking using real-time face detection and recognition. The system eliminates manual errors, speeds up attendance tracking, and provides an efficient way to manage student presence in classrooms.

Key Functionalities:

i. **Face Detection & Recognition:** Uses YOLOv8 to detect faces accurately.
ii. **Attendance Marking:** Automatically logs attendance in Excel sheets.
iii. **Multi-Input Support:** Processes live video, pre-recorded videos, and images.
iv. **User Authentication:** Admin login restricts unauthorized access.
v. **Data Management:** Attendance data is exportable and analysable.

Workflow:

1. The user selects the attendance mode (live/video/image).
2. The system captures and processes face images.
3. YOLOv8 detects and recognizes faces.
4. Attendance is recorded in an Excel sheet.
5. Reports can be generated and exported.

Future Enhancements:

- **Cloud-based Storage:** Remote data access.
- **Mobile App Integration:** Easier attendance tracking.
- **Voice-Based Authentication:** Additional security.
- **ERP Integration:** Sync with academic systems.

5. SYSTEM DESIGN

5.1 SCOPE OF THE PROJECT

The scope of the Next-Gen Attendance System involves developing an automated, AI-powered solution that uses facial recognition technology for accurate and secure attendance tracking. This system will allow for real-time attendance monitoring without physical intervention, eliminating fraud and proxy attendance. The project aims to integrate deep learning for continuous improvement in recognition accuracy and adapt to various environments.

The system will be scalable, supporting small classrooms to large organizations, and will ensure data security and privacy by encrypting user information. It will integrate seamlessly with existing administrative systems for automated reporting, payroll, and grading. The project will also include a user-friendly interface for administrators to manage and monitor attendance data efficiently. Additionally, backup modes like QR scanning or RFID will ensure reliability in case of technical issues. The overall goal is to streamline attendance processes, enhance security, and improve operational efficiency.

5.2 DATASET

The Next-Gen Attendance System relies on a high-quality dataset to train and validate its deep learning models for facial recognition. The dataset consists of images and video sequences of individuals captured in various real-world conditions, ensuring robustness and accuracy in diverse environments. It includes a combination of publicly available facial recognition datasets and custom-collected data specific to the institution or organization implementing the system.

The dataset is structured to include variations in facial expressions, lighting conditions, angles, and occlusions (e.g., masks, glasses, or hats). Each image is labeled with unique identifiers, ensuring precise mapping of faces to identities. Additionally, the dataset is regularly updated with new entries to improve model performance over time. Preprocessing techniques, such as image normalization, augmentation, and feature extraction using CNN architectures, enhance recognition accuracy and reduce false positives.

To ensure data privacy and security, the dataset is stored in an encrypted format within a secured database like MongoDB or PostgreSQL. Access control mechanisms restrict unauthorized modifications, and compliance with data protection regulations (such as GDPR) ensures ethical handling of biometric information. The dataset plays a crucial role in improving attendance accuracy, preventing spoofing, and ensuring a seamless experience for users in academic and corporate settings.

Dataset Creation

A successful facial recognition system begins with a comprehensive and structured database of known faces. This database is created by capturing images of each student from multiple angles to ensure the system can recognize them in various settings, such as frontal, left-side, right-side, and angled positions. The system typically requires at least 15 different images per student to achieve high accuracy.

- **Data Structuring**: The facial images are stored in a structured folder system. Each section of students has its own directory, such as section_C/database_C, where the students' images are labelled and stored. For instance, a student's images might be saved as 5L0_10.jpg, where 5L0 indicates the class and 10 represents the student ID[6].
- **Image Labelling**: Image labelling is done manually or automatically, ensuring that each student’s image is tagged with their unique identifier. This step is essential because it ensures that the system can differentiate between multiple individuals based on their facial features.

Facial recognition is a highly accurate and efficient method, offering significant advantages over older technologies. The system ensures quick identification, real-time processing, and automated data recording, making it an ideal solution for large classrooms, online classes, and hybrid learning environments

5.3 DATA PRE-PROCESSING

Pre-processing Fig 5.1 is a crucial step in the Next-Gen Attendance System to enhance facial recognition accuracy and ensure robust performance in real-world scenarios. The raw facial images and video frames captured from the camera undergo multiple pre-processing techniques to improve clarity, remove noise, and standardize inputs before feeding them into deep learning models. These techniques help the system handle variations in lighting, pose, and occlusions effectively.

Face Detection and Alignment: The system first detects faces in images or video frames using YOLOv8. Once detected, faces are aligned using facial landmarks (eyes, nose, and mouth) to ensure consistent positioning. This alignment reduces errors caused by head tilts or improper angles, improving recognition accuracy.

Image Normalization and Augmentation: The detected facial images are resized to a fixed resolution (e.g., 224×224 pixels**)** and normalized by scaling pixel values between 0 and 1. Augmentation techniques such as rotation, brightness adjustments, flipping, and Gaussian noise addition are applied to increase dataset diversity. This ensures that the deep learning model generalizes well to new faces in different conditions.

Feature Extraction and Embedding Generation: Pre-trained CNN models are used to extract meaningful facial features. These features are converted into embeddings, which represent faces in a lower-dimensional space. This embedding-based approach helps the system differentiate between similar-looking individuals while reducing computational complexity.

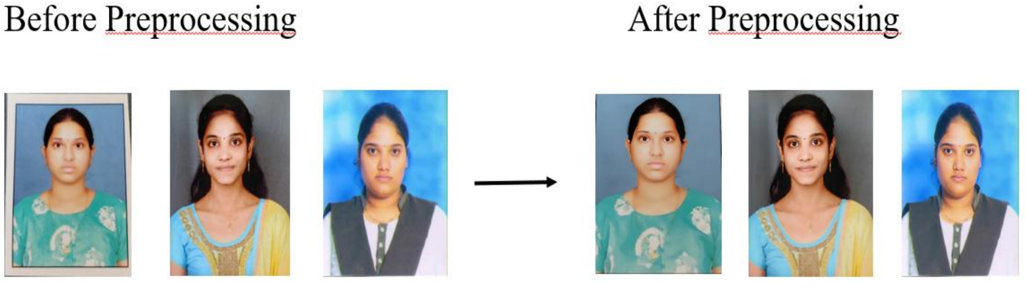


Fig 5.1: DATA PRE-PROCESSING

5.4 FEATURE EXTRACTION:

Feature extraction plays a crucial role in the Next-Gen Attendance System, enabling accurate and efficient facial recognition. It involves identifying and encoding distinct facial characteristics to differentiate individuals. Instead of relying on raw images, the system extracts high-level features, such as facial structure, texture, and key landmark positions, to enhance recognition performance. This step ensures robustness against variations in lighting, pose, and occlusions.

The system utilizes Deep Learning models like for feature extraction. These pre-trained Convolutional Neural Networks (CNNs) process facial images through multiple layers, capturing essential details such as edges, contours, and spatial relationships. The extracted features are then transformed into numerical embeddings, which serve as unique representations of each face. This approach improves identification accuracy while reducing computational complexity.

Once the facial embeddings are generated, they are stored in a database for comparison. During recognition, new images undergo the same feature extraction process, and their embeddings are matched with stored templates using similarity metrics like Cosine Similarity or Euclidean Distance. This ensures fast and accurate face matching, allowing real-time attendance tracking. By leveraging advanced feature extraction techniques, the Next-Gen Attendance System achieves high precision, reliability, and scalability across diverse environments.

5.5 SYSTEM ARCHITECTURE:

The Next-Gen Attendance System is an innovative solution developed to automate attendance tracking in educational institutions using cutting-edge deep learning and computer vision technologies. This system eliminates the need for traditional methods such as manual roll calls or paper-based attendance sheets. Instead, it leverages facial recognition to detect and record the presence of students in both physical and online environments.

Facial recognition is a highly accurate and efficient method, offering significant advantages over older technologies. The system ensures quick identification, real-time processing, and automated data recording, making it an ideal solution for large classrooms, online classes, and hybrid learning environments

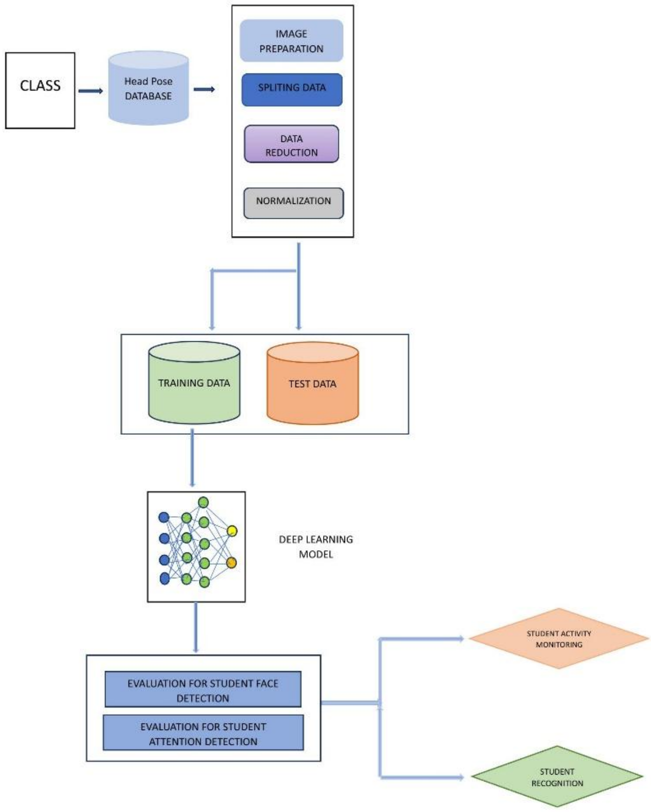


Fig 5.2: DL-BASED STUDENT MONITORING SYSTEM

Fig 5.2 is the flowchart that describes a deep learning-based system which keeps track of student activity and attention in online classes. The system accumulates videos from students in a virtual classroom and saves the extracted data in the head pose database. The data is pre-processed using image processing, splitting, reduction, and normalization, followed by division of data into training and test sets. A deep learning model is trained to detect the face of the student and track their attention. An evaluation of the system is done to check if it correctly identifies faces and measures attention. The system monitors and engages the learner in learning environments.

The core advantage of the Next-Gen Attendance System lies in its integration of deep learning algorithms with advanced computer vision techniques[6], offering a scalable, automated, and error-free way to monitor and track students' attendance.

It includes several integrated modules that work together to process the input data (video feed, images, or video uploads), recognize faces, and record attendance in real time. The key components of the system are:

- Input Devices: The system uses webcams or cameras installed in classrooms to capture live video feeds. In an online classroom, students' video feeds are captured directly from the application hosting the online class.
- Face Detection: The system employs YOLOv8 (You Only Look Once version 8) for accurate and real-time face detection. YOLOv8[7] identifies faces in video frames and isolates them with bounding boxes.
- Face Recognition: Once faces are detected, the system uses a pre-trained face recognition model to encode the face into a unique 128-dimensional feature vector. This encoding allows the system to compare the detected face against known faces in the database.
- Attendance Storage: Data related to the attendance (name, date, time) is stored in an organized database, typically in Excel format. Each section has its folder where the attendance files are saved.
- System Flow: The system operates in real time, detecting faces, encoding them, comparing them with the database, and then recording the attendance without human intervention.

In summary, the Next-Gen Attendance System uses face detection and recognition, combined with data storage and management techniques, to automate the entire process of attendance marking. This offers significant improvements in efficiency and accuracy over traditional methods.

1 Face Detection and Recognition

- **Face Detection**

Face detection is the first step in identifying a student’s presence. YOLOv8 is used for real-time face detection due to its speed and accuracy. YOLO (You Only Look Once) is an object detection algorithm that processes an entire image in a single forward pass, allowing for real-time processing. YOLOv8 is a highly optimized version, capable of detecting faces in crowded classrooms or dynamic environments with minimal computational resources.

Bounding Boxes: When YOLOv8 detects a face, it draws a bounding box around the face in the image. The system can then focus on this specific area for further recognition processing.

Real-Time Detection: The live video feed from the webcam or camera is processed frame by frame. YOLOv8 processes each frame in real-time, ensuring that faces are detected and tracked as students move around the classroom or change positions.

- **Face Encoding**

  Once the face is detected, the system uses a pre-trained face recognition model (such as FaceNet or Dlib) to encode the face. Encoding converts the detected face into a 128-dimensional vector, representing the unique features of the face. This vector serves as a "fingerprint" for the face and is used for comparison with faces in the database[8].

- **Feature Extraction:**

  The encoding process extracts key facial features such as the distance between the eyes, nose shape, and jawline, which are unique to each individual. This helps in achieving high accuracy, even in situations where the lighting or facial expressions change.

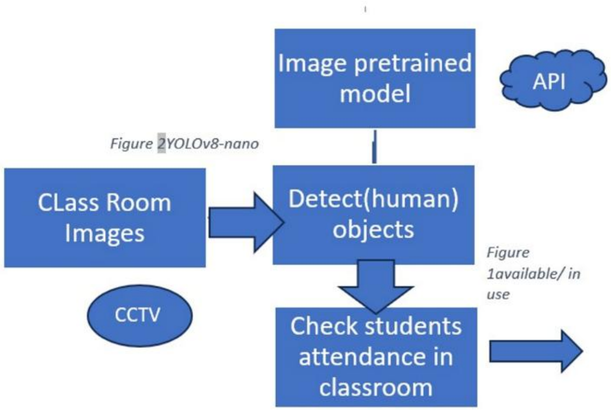


Fig 5.3: YOLOV8-NANO ATTENDANCE DETECTION SYSTEM

Figure 5.3 is a student attendance system by utilizing a pre-trained YOLOv8 [8] nano model that captures images through cameras on the classrooms and processes these images using a pre-trained model of people detection and then scanning for the attendance of the students. This results in identifying students present or actively attending in class, thereby eliminating the laborious task of keeping a record and the processing of attendance in class.

- **Face Matching**

  The encoded face is compared against the stored encodings in the database.

The system calculates the Euclidean distance between the new face encoding and the stored encodings. If the distance is below a predefined threshold (e.g., 0.55), the system recognizes the face and marks the student as present[9].

Threshold Setting: The threshold value is crucial to ensure that the system doesn't incorrectly mark a student as present based on a close but incorrect match. Fine-tuning the threshold value is necessary to balance accuracy and performance.

Handling Unrecognized Faces: If the system detects a face that does not match any encoding in the database, it prompts the system to either flag it for manual review or capture and store the face for future recognition.

2 Attendance Marking and Data Storage

o Marking Attendance

Once a face is recognized, the system proceeds to mark the attendance of the student. The attendance process is highly automated and efficient:

1. **Duplicate Prevention**: Before marking attendance, the system checks whether the student has already been marked for attendance that day. This prevents multiple entries for the same individual[13].
2. **Saving Data**: Attendance is recorded in a structured format, typically in an Excel file. Each section of students has its own folder, and within that folder, the attendance for each day is stored as a separate file (e.g., Attendance_2024-07-13.xlsx).
    - **Excel Integration**: The system uses the Python libraries pandas and openpyxl to handle data manipulation and Excel file management. pandas is used to create and update data frames, while openpyxl is used to write data into Excel files.
3. **Tracking Details**: The Excel file contains detailed records, such as the student's name, the time of attendance, and the date. This information helps in generating reports and auditing attendance records.

o Storing Attendance Data

The system ensures that all attendance data is securely stored and organized. For each section, there is a dedicated folder where the attendance records are stored:

- **File Structure**: The files are named according to the section and date, e.g., Attendance_2024-07-13.xlsx. This structure allows administrators to quickly locate and retrieve attendance records.

- **Daily Sheets**: Every day, the system generates a new Excel file to store that day's attendance. This ensures that the data is organized by date and prevents the accumulation of large files.

3 Real-Time Interaction and User Interface

- **Real-Time Feed and Detection**

The **Next-Gen Attendance System** operates in real-time, displaying live video feed from the webcam or camera. As students enter the frame, the system detects their faces, encodes them, and checks against the stored database for matches. The detected faces are enclosed within bounding boxes, with green boxes for recognized students and red boxes for unrecognized faces.

- **User Feedback**: This real-time interaction provides immediate feedback to the system, showing whether a student is recognized or not. This is useful in dynamic classroom settings where students may move around or change positions.

- **Keyboard Shortcuts**

To allow for easy user interaction, the system supports keyboard shortcuts. For example:

- **"Esc"**: Exits the system, allowing the user to stop the session.
- **"R"**: Enables the enrollment of a new person, allowing for real-time addition of new faces to the database.

These shortcuts make the system more flexible and easy to manage in real-time environments.

- **Dynamic Updates and Re-Capturing Faces**

The system is designed to handle dynamic updates, allowing new faces to be added without interrupting the attendance process. When a new face is captured, it is processed, encoded, and stored in the database.

- **Real-Time Face Updates**: The system doesn't require a restart to add new faces. It dynamically updates itself by capturing, encoding, and storing new face data as it is recognized.

4 Challenges and Solutions

- **Frame Stuttering and Performance**

  Real-time processing of video feeds can sometimes result in frame stuttering, especially when running on lower-end hardware. To address this issue, the system processes video frames at a reduced resolution (e.g., 0.25x scale). This reduces the load on the system while still maintaining an acceptable level of performance.

- **Optimizing Resolution**: By lowering the resolution, the system can process more frames per second, improving the responsiveness and reducing the lag associated with frame stuttering.

- **Duplicate Attendance**

To avoid duplicate attendance entries, the system cross-references the student's attendance status in the Excel file before marking attendance. If the system detects that the student has already been marked present for the day, it does not add a duplicate entry.

- **Dynamic Updates**

The system is designed to handle dynamic updates without requiring a reboot. New faces are captured, encoded, and added to the system in real-time, ensuring that the system stays up-to-date with minimal intervention.

5.6 CLASSIFICATION

Classification in deep learning involves categorizing data into specific labels or groups, enabling systems to recognize and differentiate among objects effectively. In the Next-Gen Attendance System, classification plays a vital role in identifying individual faces detected by the system and ensuring accurate attendance marking. The process integrates seamlessly with advanced detection and labeling techniques to handle dynamic environments, such as classrooms.

The system primarily employs **YOLOv8**, an advanced object detection model that incorporates convolutional layers for feature extraction. Unlike standalone CNN models such as VGG19 or ResNet50, YOLOv8 combines detection and classification into a single streamlined architecture. This integration enables the system to detect and label faces in real-time, ensuring accurate attendance tracking without requiring additional classification models[10].

Advantages of YOLOv8 in Classification

YOLOv8's architecture is optimized for tasks like face detection and classification. Key benefits include:

- **Real-Time Performance:** Efficiently detects multiple faces in dynamic environments.
- **Integrated Feature Extraction:** Eliminates the need for separate CNN-based classification models by embedding convolutional operations within the detection pipeline.
- **Accuracy and Speed:** Balances high detection accuracy with fast processing times, essential for real-time applications.

Workflow of Classification Using YOLOv8

1. **Data Processing:**
   Input sources such as live webcam feeds, videos, or images are pre-processed to suit YOLOv8's requirements.
2. **Face Detection and Labelling:**
   YOLOv8 detects faces and assigns unique identifiers, avoiding redundancy in attendance marking.
3. **Integrated Classification:**
   The convolutional layers within YOLOv8 handle feature extraction, enabling the system to classify and label faces efficiently.

Performance Evaluation

YOLOv8's classification performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. The system effectively addresses challenges such as multiple faces, varied lighting, and dynamic positioning, ensuring reliable attendance marking.

5.7 MODULES

In the context of software development, a module is a self-contained, independent unit of code that performs a specific task or functionality within a larger system. The Next-Gen Attendance System consists of several modules that work together to enable efficient and automated attendance tracking using facial recognition.

1. Data Collection Module

This module is responsible for collecting and organizing facial images required for training the deep learning model. It gathers images from live webcam feeds, uploaded videos, and static image files.

Sample Code:

```
import os
import cv2
def collect_images(source_folder):
    images = []
    for filename in os.listdir(source_folder):
        if filename.endswith(".jpg") or filename.endswith(".png"):
            img = cv2.imread(os.path.join(source_folder, filename))
            images.append(img)
    return images
```

2. Preprocessing Module

This module enhances image quality and prepares the data for feature extraction. It includes grayscale conversion, histogram equalization, noise reduction, and image resizing.

Sample Code:

```
import cv2
import numpy as np
def preprocess_image(image_path):
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
```

```
    image = cv2.equalizeHist(image)  # Enhancing contrast
    image = cv2.resize(image, (224, 224))  # Resizing for CNN input
    return image
```

3. Face Detection and Alignment Module

This module detects faces in images or video frames and aligns them properly for further processing. It uses **YOLOv8** for detection and OpenCV’s Dlib for alignment.

Sample Code:

```
import cv2
import dlib
detector = dlib.get_frontal_face_detector()
def detect_faces(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)
    return faces
```

4. Feature Extraction Module

This module extracts deep facial features using pre-trained CNN models. These features are used for face recognition and identity verification.

Sample Code:

```
from tensorflow.keras.applications import VGG19
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg19 import preprocess_input
import numpy as np
model = VGG19(weights='imagenet', include_top=False)
def extract_features(img_path):
    img = image.load_img(img_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)
    features = model.predict(img_array)
```

```
    return features.flatten()
```

5. Face Recognition Module

This module compares extracted features with stored embeddings to identify individuals using similarity metrics such as **Cosine Similarity** or **Euclidean Distance**.

Sample Code:

```
from scipy.spatial.distance import cosine
def recognize_face(embedding, stored_embeddings):
    min_dist = float("inf")
    identity = None
    for name, stored_embedding in stored_embeddings.items():
        distance = cosine(embedding, stored_embedding)
        if distance < min_dist:
            min_dist = distance
            identity = name
    return identity if min_dist < 0.5 else "Unknown"
```

6. Attendance Logging Module

This module records attendance data in separate Excel sheets for different sections, ensuring that each individual is marked only once per day.

Sample Code:

```
import pandas as pd
from datetime import datetime
def mark_attendance(name, section):
    filename = f"{section}_attendance.xlsx"
    df = pd.read_excel(filename) if os.path.exists(filename) else
pd.DataFrame(columns=["Name", "Time"])
    if name not in df["Name"].values:
        new_entry = pd.DataFrame([[name, datetime.now().strftime("%Y-%m-%d
%H:%M:%S")]], columns=["Name", "Time"])
        df = pd.concat([df, new_entry], ignore_index=True)
```

```
    df.to_excel(filename, index=False)
```

7. Database Management Module

This module manages the storage and retrieval of facial embeddings and attendance records using **MySQL, PostgreSQL, or MongoDB**.

Sample Code:

```
import sqlite3
def store_embedding(name, embedding):
    conn = sqlite3.connect("face_database.db")
    cursor = conn.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS embeddings (name TEXT, features BLOB)")
    cursor.execute("INSERT INTO embeddings (name, features) VALUES (?, ?)", (name, embedding.tobytes()))
    conn.commit()
    conn.close()
```

8. Web Interface Module

This module provides a user-friendly interface for teachers and administrators to upload videos/images and view attendance records. It is built using **Flask and HTML/CSS**.

Sample Code:

```
from flask import Flask, request, render_template
import os
app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/upload', methods=['POST'])
def upload():
    file = request.files['file']
    file.save(os.path.join("uploads", file.filename))
```

```
    return "File Uploaded Successfully"
if __name__ == '__main__':
    app.run(debug=True)
```

9. Security and Authentication Module

This module ensures secure access control by implementing **user authentication, encrypted data storage, and role-based permissions**.

Sample Code:

```
from werkzeug.security import generate_password_hash, check_password_hash
def create_user(username, password):
    hashed_password = generate_password_hash(password, method='sha256')
    # Store username and hashed_password in the database
    return hashed_password
def verify_user(username, password, stored_hash):
    return check_password_hash(stored_hash, password)
```

5.8 UML DIAGRAM:

The Next-Gen Attendance System follows a simple architecture that enables efficient attendance tracking using facial recognition. The UML diagram below illustrates the core components and their relationships:

System Components

1. AttendanceSystem: The main component that manages the entire attendance process. It provides three methods for marking attendance - via webcam, uploaded images, or uploaded videos.

2. FaceRecognition: Handles the detection and recognition of faces using YOLOv8 technology. This component processes input from various sources and identifies students by comparing detected faces with stored encodings.

3. Database: Stores and manages student information, face encodings, and attendance records. It prevents duplicate entries by checking if a student has already been marked present on a particular date.

4. UserInterface: Provides a simple interface for teachers to select attendance options and view results. The interface displays real-time feedback during attendance capture.

5. ExcelManager: Creates and manages Excel sheets that store attendance records by date and section. Each file contains student IDs and timestamps for easy tracking.

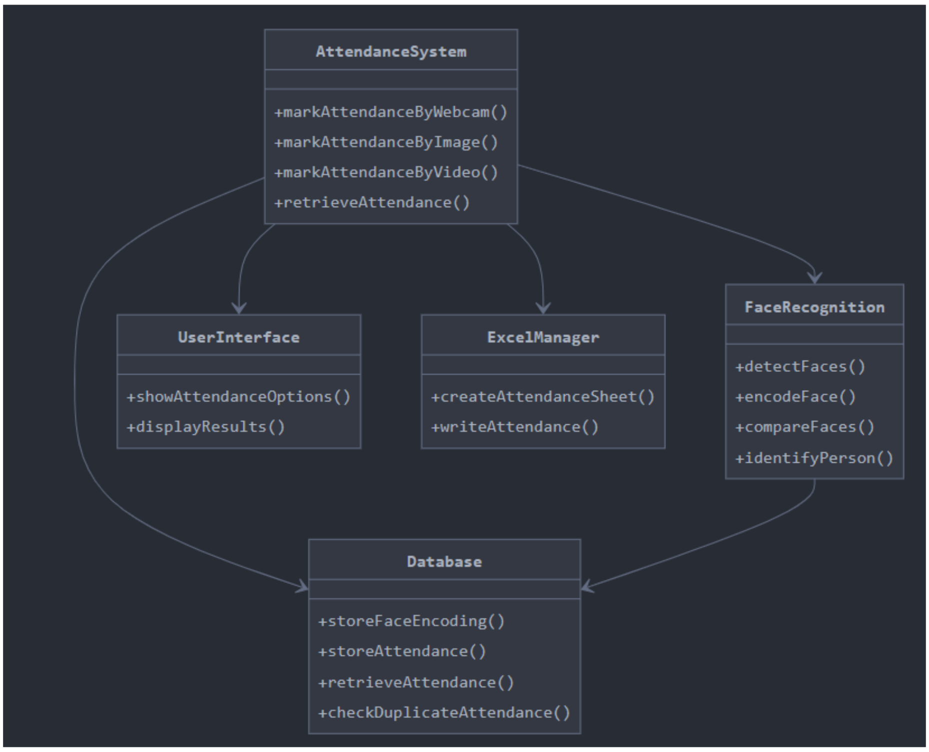


FIG 5.4: NEXT-GEN ATTENDANCE SYSTEM ARCHITECTURE

How They Work Together

The AttendanceSystem coordinates all components.FIG 5.4 When a teacher chooses an attendance method, the system captures input (webcam feed, image, or video), passes it to the FaceRecognition component for processing, stores the results through the Database, and outputs attendance records via the ExcelManager. The UserInterface facilitates interaction between the teacher and the system throughout this process.

6 IMPLEMENTATION

6.1 MODEL IMPLEMENTATION

attendanceWebcamC.py

 # Script for capturing attendance using real-time webcam face detection.

```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime
import stat
def findEncoding(imageList, nameList):
    encodeList = []
    for img, name in zip(imageList, nameList):
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)
        if len(encode) > 0:
            encodeList.append(encode[0])
        else:
            encode.append([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0])
            encodeList.append(encode[0])
            print(f"{name}'s image is not proper please recapture it")
    return encodeList
def markAttendance(name, filename):
    file_exists = os.path.exists(filename)
    if file_exists:
```

```
        # Temporarily make the file writable
        os.chmod(filename, stat.S_IWRITE)
        with open(filename, 'r') as f:
            existing_entries = f.readlines()
            if any(name in entry for entry in existing_entries):
                # Revert file permissions to read-only
                os.chmod(filename, stat.S_IREAD)
                return  # Name already exists, no need to add again
    with open(filename, 'a') as f:
        f.write(f'{name},{datetime.now()}\n')

    # Change the file permissions to read-only
    os.chmod(filename, stat.S_IREAD)
def captureImages(name, num_images=15):
    cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)  # Use DirectShow to avoid
webcam issues
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
    if not cap.isOpened():
        print("Error: Could not open webcam. Please check if another app is using it.")
        exit()
    count = 0
    directions = [
        "Face Forward",
        "Turn Left",
        "Turn Right",
        "Look Up",
        "Look Down",
        "Tilt Left",
        "Tilt Right",
        "Slight Left",
        "Slight Right",
```

```
        "Left Profile",
        "Right Profile",
        "Up Left",
        "Up Right",
        "Down Left",
        "Down Right"
    ]
    while count < num_images:
        ret, frame = cap.read()
        if not ret:
            break
        direction = directions[count] if count < len(directions) else "Face Forward"
        instruction_frame = frame.copy()
        cv2.putText(instruction_frame, f"Capture {count + 1}/{num_images}: {direction}", (15, 50), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 2)
        frame_height, frame_width, _ = instruction_frame.shape
        center_x, center_y = frame_width // 2, frame_height // 2
        radius = min(frame_width, frame_height) // 3  # Increased radius to fit larger faces
        # Draw thin progress bar
        bar_height = 10
        cv2.rectangle(instruction_frame, (10, 10), (frame_width - 10, 10 + bar_height), (0, 0, 0), 2)
        cv2.rectangle(instruction_frame, (10, 10), (10 + int((frame_width - 20) * (count / num_images)), 10 + bar_height), (0, 255, 0), cv2.FILLED)

        # Draw circle
        cv2.circle(instruction_frame, (center_x, center_y), radius, (0, 255, 0), 2)

        # Detect face and check if it fits within the circle
        face_locations = face_recognition.face_locations(frame)
        fits_in_circle = False
```

```
        for (top, right, bottom, left) in face_locations:
            face_center_x, face_center_y = (left + right) // 2, (top + bottom) // 2
            face_radius = max(right - left, bottom - top) // 2
            distance_from_center = np.sqrt((face_center_x - center_x) ** 2 +
(face_center_y - center_y) ** 2)
            if distance_from_center + face_radius <= radius:
                fits_in_circle = True
                break
        if fits_in_circle:
            cv2.putText(instruction_frame, "Face Fits", (15, 450),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
        else:
            cv2.putText(instruction_frame, "Face Not Fit", (15, 450),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255), 2)
            cv2.circle(instruction_frame, (center_x, center_y), radius, (0, 0, 255), 2)
        cv2.imshow('Webcam', instruction_frame)
        key = cv2.waitKey(1)
        if key == 32 and fits_in_circle:  # Press SpaceBar to capture if face fits
            count += 1
            cv2.imwrite(f'C:\\Users\\HP\\OneDrive\\Desktop\\Attendance-Face-Detection-
master\\Attendance-Face-Detection-master\\Attendance-Face-Detection-
master\\section_C\\database_C/{name}_{count}.jpg', frame)  # Save the original
frame without instructions
            cv2.putText(instruction_frame, "Image Captured!", (15, 450),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
            cv2.imshow('Webcam', instruction_frame)
            cv2.waitKey(500)  # Display captured image text for 500ms
        if key == 27:  # Press Esc to exit early
            break
    cap.release()
    cv2.destroyAllWindows()
def startWebcam():
```

```
    path = 'C:\\Users\\HP\\OneDrive\\Desktop\\Attendance-Face-Detection-
master\\Attendance-Face-Detection-master\\Attendance-Face-Detection-
master\\section_C\\database_C'
    imageList = []
    personName = []
    dataList = [file for file in os.listdir(path) if file.endswith(('.jpg', '.png'))]  # Only
include images
    for data in dataList:
        curImage = cv2.imread(f'{path}/{data}')
        if curImage is None:
            print(f"Warning: Could not read image {data}. Skipping...")
            continue
        imageList.append(curImage)
        curName = os.path.splitext(data)[0]
        personName.append(curName)
    print('Starting Encoding of Known Images in Database...')
    encodedKnown = findEncoding(imageList, personName)
    print('Encoding of Known Images Completed...')
    cap = cv2.VideoCapture(0)
    while True:
        ret, frame = cap.read()
        frameS = cv2.resize(frame, (0, 0), None, 0.25, 0.25)
        frameS = cv2.cvtColor(frameS, cv2.COLOR_BGR2RGB)
        cv2.putText(frame, "Press Esc to Exit or R to Register", (15, 450),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 2)
        curFaceFrame = face_recognition.face_locations(frameS)
        curEncoding = face_recognition.face_encodings(frameS, curFaceFrame)
        for encoding, faceFrame in zip(curEncoding, curFaceFrame):
            if len(encodedKnown) > 0:  # Ensure there are known faces before comparing
                result = face_recognition.compare_faces(encodedKnown, encoding)
                faceDist = face_recognition.face_distance(encodedKnown, encoding)
                Index = np.argmin(faceDist)
```

```
            if faceDist[Index] < 0.55 and result[Index]:

# Ensure face matching is correct
                name = personName[Index].upper()
            else:
                name = "Unknown"
        y1, x2, y2, x1 = faceFrame
        y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
        if faceDist[Index] < 0.55 and result:
            name = personName[Index].upper()
            print(name)
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(frame, f'{name}', (x1, y1-5),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0))

            # Generate filename based on the current date
            date_str = datetime.now().strftime("%Y-%m-%d")
            filename = f"AttendanceWebcam_{date_str}.csv"
            markAttendance(name, filename)
        else:
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)
            cv2.putText(frame, f'unknown', (x1, y1-5),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0))
    cv2.imshow("Webcam", frame)
    key = cv2.waitKey(1) & 0xFF  # Ensure proper key detection
    if key == 27 or cv2.getWindowProperty("Webcam",
cv2.WND_PROP_VISIBLE) < 1:  # Exit when window is closed
        break
    elif key == ord('r'):  # Press 'r' to register a new person
        person_name = input("Enter the name of the person to register: ")
        captureImages(person_name)
        # Re-encode known faces after adding new person
```

```
        dataList = os.listdir(path)
        imageList = []
        personName = []
        for data in dataList:
            curImage = cv2.imread(f'{path}/{data}')
            imageList.append(curImage)
            curName = os.path.splitext(data)[0]
            personName.append(curName)
        encodedKnown = findEncoding(imageList, personName)
        print('Re-encoding Completed...')
if __name__ == "__main__":
    startWebcam()
    cap.release()
    cv2.destroyAllWindows()
```

attendanceImageC.py

 # Script for recognizing faces and marking attendance from static images.

```
from datetime import datetime
import cv2
import numpy as np
import face_recognition
import os
from functionC import findEncoding, markAttendance
def startImage(directoryPath):
    path = 'C:\\Users\\HP\\OneDrive\\Desktop\\Attendance-Face-Detection-
master\\Attendance-Face-Detection-master\\Attendance-Face-Detection-
master\\section_C'
    imageList = []
    personName = []
    dataList = [file for file in os.listdir(path) if file.endswith('.jpg')]
    if not dataList:
```

```
    print("No images found in the database.")
    return
for data in dataList:
    curImage = cv2.imread(f'{path}/{data}')
    if curImage is None:
        print(f"Error reading image {data}. Skipping...")
        continue
    imageList.append(curImage)
    curName = os.path.splitext(data)[0]
    personName.append(curName)
print('Starting Encoding of Known Images in Database...')
encodedKnown = findEncoding(imageList, personName)
print('Encoding of Known Images Completed...')
imageCheckList = []
dataCheckList = [file for file in os.listdir(directoryPath) if file.endswith('.jpg')]
if not dataCheckList:
    print("No images found in the provided directory.")
    return
for data in dataCheckList:
    curImage = cv2.imread(f'{directoryPath}/{data}')
    if curImage is None:
        print(f"Error reading image {data}. Skipping...")
        continue
    imageCheckList.append(curImage)
for image in imageCheckList:
    imageS = cv2.resize(image, (0, 0), None, 0.25, 0.25)
    imageS = cv2.cvtColor(imageS, cv2.COLOR_BGR2RGB)
    curFaceFrame = face_recognition.face_locations(imageS)
    curEncoding = face_recognition.face_encodings(imageS, curFaceFrame)
    for encoding, faceFrame in zip(curEncoding, curFaceFrame):
        result = face_recognition.compare_faces(encodedKnown, encoding)
        faceDist = face_recognition.face_distance(encodedKnown, encoding)
```

```
            Index = np.argmin(faceDist)
            y1, x2, y2, x1 = faceFrame
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            if faceDist[Index] < 0.55 and result:
                name = personName[Index].upper()
                print(name)
                cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)
                cv2.putText(image, f'{name}', (x1, y1-5),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0))
                date_str = datetime.now().strftime("%Y-%m-%d")
                filename = f"AttendanceImage{date_str}.csv"
                markAttendance(name, filename)
            else:
                cv2.rectangle(image, (x1, y1), (x2, y2), (0, 0, 255), 3)
                cv2.putText(image, f'Unknown', (x1, y1 - 6),
cv2.FONT_HERSHEY_COMPLEX, 3, (0, 0, 255), 5)
    cv2.imshow("Picture", image)
    cv2.waitKey(0)
```

attendanceVideoC.py

 # Script for processing pre-recorded videos to detect and mark attendance.

```
from datetime import datetime
import cv2
import numpy as np
import face_recognition
import os
from functionC import findEncoding, markAttendance
def startVideo(videoPath):

    # Updated path to the images directory
    path = 'section_C/database_C'
```

```
    imageList = []
    personName = []

    # Get a list of all files in the specified directory
    dataList = os.listdir(path)
    for data in dataList:
        curImage = cv2.imread(f'{path}/{data}')
        imageList.append(curImage)
        curName = os.path.splitext(data)[0]
        personName.append(curName)
    print('Starting Encoding of Known Images in Database...')
    encodedKnown = findEncoding(imageList, personName)
    print('Encoding of Known Images Completed...')
    cap = cv2.VideoCapture(videoPath)
    while True:
        ret, frame = cap.read()
        if not ret:
            print("Failed to grab frame or end of video")
            break
        frameS = cv2.resize(frame, (0, 0), None, 0.25, 0.25)
        frameS = cv2.cvtColor(frameS, cv2.COLOR_BGR2RGB)
        cv2.putText(frame, "Press Esc to Exit", (15, 450),
cv2.FONT_HERSHEY_TRIPLEX, 1, (255, 255, 255), 2)
        curFaceFrame = face_recognition.face_locations(frameS)
        curEncoding = face_recognition.face_encodings(frameS, curFaceFrame)
        for encoding, faceFrame in zip(curEncoding, curFaceFrame):
            result = face_recognition.compare_faces(encodedKnown, encoding)
            faceDist = face_recognition.face_distance(encodedKnown, encoding)
            Index = np.argmin(faceDist)
            y1, x2, y2, x1 = faceFrame
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            if faceDist[Index] < 0.55 and result:
```

```
                name = personName[Index].upper()
                print(name)
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
                cv2.putText(frame, f'{name}', (x1, y1-5),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0))
                date_str = datetime.now().strftime("%Y-%m-%d")
                filename = f"AttendanceImage_{date_str}.csv"
                markAttendance(name, filename)
            else:
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)
                cv2.putText(frame, f'unknown', (x1, y1-5),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255))
        cv2.imshow("Video", frame)
        key = cv2.waitKey(1)
        if key == 27:  # Press Esc to exit
            cv2.destroyWindow("Video")
            break
    cap.release()
    cv2.destroyAllWindows()
```

fuctionC.py

 # Contains helper functions for face recognition, database operations, and attendance marking.

```
import cv2
import os
import re
from datetime import datetime
import webbrowser
import face_recognition
import numpy as np
def is_valid_roll_no(roll_no):
```

```
    """Check if the roll number follows a valid format (only digits or alphanumeric
with at least one letter)."""
    return bool(re.match(r'^[A-Za-z]*\d+[A-Za-z]*$', roll_no))  # Must contain at least
one digit
def findEncoding(imageList, nameList):
    encodeList = []
    for img, name in zip(imageList, nameList):
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)
        if len(encode) > 0:
            encodeList.append(encode[0])
        else:
            encode.append([0]*128)  # Using 128 zeros to represent an empty encoding
            print(f"{name}'s image is not proper please recapture it")
    return encodeList
def markAttendance(name, path):
    with open(f'{path}', 'r+') as f:
        myDataList = f.readlines()
        nameList = [entry.split(',')[0] for entry in myDataList]
        if name not in nameList:
            now = datetime.now()
            dt_string = now.strftime("%H:%M:%S")
            f.writelines(f'\n{name},{dt_string}')
def openExcelWebcam():
    webbrowser.open("AttendanceWebcam.csv")
def openExcelImage():
    webbrowser.open("AttendanceImage.csv")
def openExcelVideo():
    webbrowser.open("AttendanceVideo.csv")
def openReadme():
    webbrowser.open(f'{os.path.dirname(__file__)}/README.md')
def captureImages(name, num_images=15):
```

```
    while not is_valid_roll_no(name):
        print("Invalid Roll Number! Please enter a valid format (e.g., A123, 2345,
C456).")
        name = input("Enter a valid roll number: ")
    directory = 'section_C/database_C'
    if not os.path.exists(directory):
        os.makedirs(directory)
    cap = cv2.VideoCapture(0)
    count = 0
    while count < num_images:
        ret, frame = cap.read()
        if not ret:
            print("Failed to grab frame")
            break
        cv2.putText(frame, f"Capturing Image {count + 1}/{num_images}", (15, 50),
cv2.FONT_HERSHEY_TRIPLEX, 1, (255, 255, 255), 2)
        cv2.imshow('Webcam', frame)
        key = cv2.waitKey(1)
        if key == 32:  # Press SpaceBar to capture
            count += 1
            filename = os.path.join(directory, f'{name}_{count}.jpg')
            cv2.imwrite(filename, frame)
            cv2.putText(frame, "Image Captured!", (15, 450),
cv2.FONT_HERSHEY_TRIPLEX, 1, (0, 255, 0), 2)
            cv2.imshow('Webcam', frame)
            cv2.waitKey(500)  # Display captured image text for 500ms
        if key == 27:  # Press Esc to exit early
            break
    cap.release()
    cv2.destroyAllWindows()
def startWebcam():
    path = 'section_C/database_C'
```

```
    imageList = []
    personName = []
    dataList = os.listdir(path)
    for data in dataList:
        curImage = cv2.imread(f'{path}/{data}')
        imageList.append(curImage)
        curName = os.path.splitext(data)[0]
        personName.append(curName)

    print('Starting Encoding of Known Images in Database...')
    encodedKnown = findEncoding(imageList, personName)
    print('Encoding of Known Images Completed...')
    cap = cv2.VideoCapture(0)
    while True:
        ret, frame = cap.read()
        frameS = cv2.resize(frame, (0, 0), None, 0.25, 0.25)
        frameS = cv2.cvtColor(frameS, cv2.COLOR_BGR2RGB)
        cv2.putText(frame, "Press Esc to Exit or R to Register", (15, 450),
cv2.FONT_HERSHEY_TRIPLEX, 1, (255, 255, 255), 2)
        curFaceFrame = face_recognition.face_locations(frameS)
        curEncoding = face_recognition.face_encodings(frameS, curFaceFrame)
        for encoding, faceFrame in zip(curEncoding, curFaceFrame):
            result = face_recognition.compare_faces(encodedKnown, encoding)
            faceDist = face_recognition.face_distance(encodedKnown, encoding)
            Index = np.argmin(faceDist)
            y1, x2, y2, x1 = faceFrame
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            if faceDist[Index] < 0.55 and result:
                name = personName[Index].upper()
                print(name)
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
                cv2.putText(frame, f'{name}', (x1, y1-5),
```

```
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0))
                date_str = datetime.now().strftime("%Y-%m-%d")
                filename = f"Attendance_{date_str}.csv"
                markAttendance(name, filename)
            else:
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)
                cv2.putText(frame, f'unknown', (x1, y1-5),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255))

        cv2.imshow("Webcam", frame)
        key = cv2.waitKey(1)
        if key == 27:  # Press Esc to exit
            cv2.destroyWindow("Webcam")
            break
        elif key == ord('r'):  # Press 'r' to register a new person
            person_name = input("Enter the name of the person to register: ")
            captureImages(person_name)

            # Re-encode known faces after adding new person
            dataList = os.listdir(path)
            imageList = []
            personName = []
            for data in dataList:
                curImage = cv2.imread(f'{path}/{data}')
                imageList.append(curImage)
                curName = os.path.splitext(data)[0]
                personName.append(curName)
            encodedKnown = findEncoding(imageList, personName)
            print('Re-encoding Completed...')
if __name__ == "__main__":
    startWebcam()
```

mainC.py

 # Main entry point for the Next-Gen Attendance System application.

```
from tkinter import *
from functionC import *
from attendanceWebcamC import *
from attendanceVideoC import *
from attendanceImageC import *
from tkinter import filedialog
import tkinter.messagebox
import re  # Import for regex validation
from PIL import ImageTk, Image
def validate_roll_no(roll_no):
    """Validate roll number format: Accepts '21471A05E5', '5E5', or '5467'."""
    pattern = r"^(21471A\d{2}[A-Z]\d|[0-9][A-Z]+[0-9]|\d+)$"
    return bool(re.match(pattern, roll_no))
def clickCapture():
    """Capture image only if roll number is valid."""
    name = enterName.get().strip()
    if not name:
        tkinter.messagebox.showerror("Error", "Roll number cannot be empty!")
        return
    if not validate_roll_no(name):
        tkinter.messagebox.showerror("Error", "Invalid Roll Number! Use formats:\n-
'21471A05E5'\n- '5E5'\n- '5467'")
        return
    ans = tkinter.messagebox.askyesno("Confirm", f"Do you want to capture image for
{name}?")
    if ans:
        captureImages(name)
def startProgramWebcam():
    tkinter.messagebox.showinfo("Info", "Please wait a while, Processing your
```

```
Database Images...")
    startWebcam()
def startProgram():
    rootStart = Tk()
    rootStart.title("Choose Option")
    rootStart.geometry("500x80+300+600")
    rootStart.configure(bg="#888888")
    Button(rootStart, text="Video", fg="#CDCDCD", bg="#80182A", width=25,
height=2, command=callStartVideo).pack(side='left', padx=(40, 0))
    Button(rootStart, text="Image", fg="#CDCDCD", bg="#80182A", width=25,
height=2, command=callStartImage).pack(side='right', padx=(0, 40))
def callStartImage():
    filename = filedialog.askdirectory(initialdir=".", title="Select Folder")
    if len(filename) > 0:
        startImage(filename)
    else:
        tkinter.messagebox.showinfo("Info", "Please select an Image Folder")
def callStartVideo():
    filename = filedialog.askopenfilename(initialdir=".", title="Select Video File")
    if len(filename) > 0:
        startVideo(filename)
    else:
        tkinter.messagebox.showinfo("Info", "Please select a Video")
def openExcelOption():
    rootExcel = Tk()
    rootExcel.title("Choose Option")
    rootExcel.geometry("380x140+300+600")
    rootExcel.configure(bg="#888888")
    Button(rootExcel, text="Attendance Live Webcam", fg="#CDCDCD",
bg="#80182A", width=25, height=2, command=openExcelWebcam).pack(pady=(5,
2))
    Button(rootExcel, text="Attendance Image", fg="#CDCDCD", bg="#80182A",
```

```
width=25, height=2, command=openExcelImage).pack(pady=(0, 2))
    Button(rootExcel, text="Attendance Video", fg="#CDCDCD", bg="#80182A", width=25, height=2, command=openExcelVideo).pack(pady=(0, 5))

# Create Main Window
root = Tk()
root.title('Facial Recognition Attendance Program')
root.attributes('-fullscreen', True)  # Set full screen mode
root.configure(background="#000000")

# Close full screen when ESC is pressed
def exit_fullscreen(event):
    root.attributes('-fullscreen', False)
root.bind("<Escape>", exit_fullscreen)
# Load Background Image
img = ImageTk.PhotoImage(Image.open('logo/nec2.jpg'))
heading = Label(root, image=img, borderwidth=0)
heading.pack(pady=20)
sub_heading = Label(root, text="Smart Attendance System", fg='#CDCDCD', bg="#80182A", font=("Cambria", 35), padx=10, pady=5)
sub_heading.pack()
spacer = Label(root, text="", bg="#000000")
spacer.pack(pady=10)
enterName = Entry(root, fg="#440D16", bg="#CDCDCD", font=15)
enterName.pack()
spacer = Label(root, text="", bg="#000000")
spacer.pack(pady=10)
Button(root, text="Add Image to Database", fg="#CDCDCD", bg="#80182A", width=25, height=2, command=clickCapture).pack(pady=(5, 20))
Button(root, text="Start Program with Live Camera", fg="#CDCDCD", bg="#80182A", width=25, height=2, command=startProgramWebcam).pack(pady=(0, 20))
```

```
Button(root, text="Import Image/Video", fg="#CDCDCD", bg="#80182A",
width=25, height=2, command=startProgram).pack(pady=(0, 20))
Button(root, text="Open Attendance Sheet", fg="#CDCDCD", bg="#80182A",
width=25, height=2, command=openExcelOption).pack(pady=(0, 20))
Button(root, text="Exit Fullscreen", fg="#CDCDCD", bg="#80182A", width=25,
height=2, command=lambda: root.attributes('-fullscreen', False)).pack(pady=(0, 20))
root.mainloop()
```

6.2 CODING

form.py

 # Manages user input forms for registration, authentication, and attendance logging.

```
from tkinter import *
from tkinter import messagebox
import os
import subprocess
import platform
def submit_form():
    year = year_var.get()
    semester = semester_var.get()
    branch = branch_var.get()
    section = section_var.get()
    if year == "Select Year" or semester == "Select Semester" or branch == "Select
Branch" or section == "Select Section":
        messagebox.showwarning("Input Error", "All fields are required!")
    else:
        messagebox.showinfo("Form Submitted", "Student information submitted
successfully!")

        # Define paths to main.py files for each section
```

```
        section_mapping = {
            "A": "section_A/mainA.py",
            "B": "section_B/mainB.py",
            "C": "section_C/mainC.py",
            "D": "section_D/mainD.py"
        }
        script_relative_path = section_mapping.get(section)
        if script_relative_path:
            script_path = os.path.join(base_path, script_relative_path)
            if os.path.exists(script_path):
                subprocess.Popen(["python", script_path], shell=True)
            else:
                messagebox.showwarning("Error", f"Script {script_path} not found!")
def cancel_form():
    root.destroy()

# Base directory
base_path = "C:/Users/HP/OneDrive/Desktop/Attendance-Face-Detection-
master/Attendance-Face-Detection-master/Attendance-Face-Detection-master/"

# Check if running on Colab or a local machine
if platform.system() == "Linux":
    print("Tkinter GUI is not supported in Colab. Run this script locally.")
    exit()
else:
    root = Tk()
root.title("Student Information Form")
root.geometry("450x400")
root.configure(bg="#f0f0f0")
frame = Frame(root, padx=20, pady=20, bg="#f0f0f0")
frame.pack(expand=True)
Label(frame, text="Student Information Form", font=("Arial", 18, "bold"),
```

```
bg="#f0f0f0", fg="#003366").grid(row=0, column=0, columnspan=2, pady=10)

# Variables
year_var = StringVar(value="Select Year")
semester_var = StringVar(value="Select Semester")
branch_var = StringVar(value="Select Branch")
section_var = StringVar(value="Select Section")

# Dropdowns
Label(frame, text="Year:", bg="#f0f0f0", fg="#003366", font=("Arial",
12)).grid(row=1, column=0, sticky=W, pady=5)
OptionMenu(frame, year_var, "I", "II", "III", "IV").grid(row=1, column=1, pady=5)
Label(frame, text="Semester:", bg="#f0f0f0", fg="#003366", font=("Arial",
12)).grid(row=2, column=0, sticky=W, pady=5)
OptionMenu(frame, semester_var, "I", "II").grid(row=2, column=1, pady=5)
Label(frame, text="Branch:", bg="#f0f0f0", fg="#003366", font=("Arial",
12)).grid(row=3, column=0, sticky=W, pady=5)
OptionMenu(frame, branch_var, "Computer Science and Engineering", "Electronics
and Communication Engineering", "Mechanical Engineering", "Electrical
Engineering", "Civil Engineering").grid(row=3, column=1, pady=5)
Label(frame, text="Section:", bg="#f0f0f0", fg="#003366", font=("Arial",
12)).grid(row=4, column=0, sticky=W, pady=5)
OptionMenu(frame, section_var, "A", "B", "C", "D").grid(row=4, column=1, pady=5)

# Buttons
Button(frame, text="Submit", command=submit_form, bg="#28a745", fg="white",
font=("Arial", 12), width=12).grid(row=5, column=0, pady=20)
Button(frame, text="Cancel", command=cancel_form, bg="#dc3545", fg="white",
font=("Arial", 12), width=12).grid(row=5, column=1, pady=20)
root.mainloop()
```

home.py

 # Handles the homepage interface and user navigation for the attendance system.

```
from tkinter import *
from PIL import Image, ImageTk
import subprocess
import os
def open_form():
    """Opens the Student Information Form."""
    script_path = "C:/Users/HP/OneDrive/Desktop/Attendance-Face-Detection-master/Attendance-Face-Detection-master/Attendance-Face-Detection-master/form.py"
    if os.path.exists(script_path):
        subprocess.Popen(["python", script_path])
    else:
        print("Error: form.py not found!")
def about_us():
    """Opens the About Us window."""
    about_window = Toplevel(root)
    about_window.title("About Us")
    about_window.geometry("800x600")
    bg_image = Image.open("about.jpg").resize((800, 600), Image.LANCZOS)
    bg_photo = ImageTk.PhotoImage(bg_image)
    bg_label = Label(about_window, image=bg_photo)
    bg_label.place(relwidth=1, relheight=1)
    bg_label.image = bg_photo
    Label(about_window, text="About Us", font=("Arial", 18, "bold"), fg="white", bg="#001F3F").pack(pady=10)
    Label(about_window, text="""
Next-Gen Attendance System is an AI-powered solution designed for smart and efficient attendance management.
```

Using advanced face detection technology, it ensures accuracy, security, and convenience in tracking attendance.
With real-time face recognition and automated data storage, our system minimizes errors and prevents proxy attendance.
Designed for scalability, it adapts to various organizational needs, making attendance tracking seamless and reliable.

```
    """, wraplength=600, justify=LEFT, font=("Arial", 12), fg="white", bg="#001F3F").pack(pady=10)
def contact_us():
    """Opens the Contact Us window."""
    contact_window = Toplevel(root)
    contact_window.title("Contact Us")
    contact_window.geometry("800x600")
    bg_image = Image.open("contact.jpg").resize((800, 600), Image.LANCZOS)
    bg_photo = ImageTk.PhotoImage(bg_image)
    bg_label = Label(contact_window, image=bg_photo)
    bg_label.place(relwidth=1, relheight=1)
    bg_label.image = bg_photo
    Label(contact_window, text="Contact Us", font=("Arial", 18, "bold"), fg="white", bg="#001F3F").pack(pady=10)
    Label(contact_window, text="""
Name: Bogyam Indu
Email: bogyamindu9100@gmail.com
Phone: 9100584945
Name: Kongara Abhinaya
Email: abhinayakongara616@gmail.com
Phone: 9550346891
Name: Thumu Tejaswini
Email: thumutejaswini.com
Phone: 830921367
""", font=("Arial", 12), fg="white", bg="#001F3F").pack(pady=10)
```

```python
# Create Main Window
root = Tk()
root.title("Home - Face Detection Attendance")
root.attributes('-fullscreen', True)  # Fullscreen mode

# Load Background Image
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
bg_image = Image.open("home.jpeg").resize((screen_width, screen_height),
Image.LANCZOS)
bg_photo = ImageTk.PhotoImage(bg_image)

# Display Background
bg_label = Label(root, image=bg_photo)
bg_label.place(relwidth=1, relheight=1)

# Title Label
Label(root, text="Next-Gen Attendance System", font=("Arial", 24, "bold"),
fg="white", bg="black").pack(pady=30)

# Button Styling
button_bg = "#87CEEB"
button_fg = "white"
button_font = ("Arial", 16)
button_width = 25
Button(root, text="Open Form", command=open_form, font=button_font,
    bg=button_bg, fg=button_fg, width=button_width, relief=FLAT).pack(pady=10)
Button(root, text="About Us", command=about_us, font=button_font,
    bg=button_bg, fg=button_fg, width=button_width, relief=FLAT).pack(pady=10)
Button(root, text="Contact Us", command=contact_us, font=button_font,
    bg=button_bg, fg=button_fg, width=button_width, relief=FLAT).pack(pady=10)
```

```python
# Exit Button to Close Fullscreen
Button(root, text="Exit", command=root.quit, font=button_font,
       bg="red", fg="white", width=button_width, relief=FLAT).pack(pady=20)
root.mainloop()
```

index.html

```html
<!-- Main HTML file for the user interface of the Next-Gen Attendance System. --->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Information Form</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Student Information Form</h1>
        <form id="studentForm">
            <label for="year">Year:</label>
            <select id="year" name="year" required>
                <option value="">Select Year</option>
                <option value="1">I</option>
                <option value="2">II</option>
                <option value="3">II</option>
                <option value="4">IV</option>
            </select>
            <label for="semester">Semester:</label>
            <select id="semester" name="semester" required>
```

```
            <option value="">Select Semester</option>
            <option value="1">I</option>
            <option value="2">II</option>
          </select>
          <label for="branch">Branch:</label>
          <select id="branch" name="branch" required>
            <option value="">Select Branch</option>
            <option value="CSE">Computer Science and Engineering</option>
            <option value="ECE">Electronics and Communication
Engineering</option>
            <option value="ME">Mechanical Engineering</option>
            <option value="EE">Electrical Engineering</option>
            <option value="CE">Civil Engineering</option>
          </select>
          <label for="section">Section:</label>
          <select id="section" name="section" required>
            <option value="">Select Section</option>
            <option value="A">Section A</option>
            <option value="B">Section B</option>
            <option value="C">Section C</option>
            <option value="D">Section D</option>
          </select>
          <div class="buttons">
            <button type="submit">Submit</button>
            <button type="button" id="cancelButton">Cancel</button>
          </div>
      </form>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

styles.css

/* Stylesheet for the design and layout of the attendance system UI. */

```
body {
   font-family: Arial, sans-serif;
   background-color: #c188f3a1;
   display: flex;
   justify-content: center;
   align-items: center;
   height: 100vh;
   margin: 0;
}
.container {
   background-color: #fff;
   padding: 20px;
   border-radius: 5px;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   width: 300px;
}
h1 {
   margin-bottom: 20px;
   font-size: 24px;
}
form {
   display: flex;
   flex-direction: column;
}
label {
   margin-bottom: 5px;
   font-weight: bold;
}
```

```
select, input {
    margin-bottom: 15px;
    padding: 8px;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 3px;
}
buttons {
    display: flex;
    justify-content: space-between;
}
button {
    padding: 10px 15px;
    font-size: 16px;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}
button[type="submit"] {
    background-color: #28a745;
    color: #fff;
}
button[type="button"] {
    background-color: #dc3545;
    color: #fff;
}
```

script.js

// JavaScript file for handling frontend interactions and dynamic UI updates.

```
document.getElementById('studentForm').addEventListener('submit', function(event)
```

```
{
    event.preventDefault();
    alert('Form submitted successfully!');
});
document.getElementById('cancelButton').addEventListener('click', function() {
    document.getElementById('studentForm').reset();
});
```

from datetime import datetime.py

 # Script for detecting unknown faces and labelling them as "UNKNOWN" when not found in the database.

```
from datetime import datetime
import cv2
import numpy as np
import face_recognition
import os
from Functions import findEncoding, markAttendance
def startImage(directoryPath):
    path = 'database'
    imageList = []
    personName = []
    dataList = [file for file in os.listdir(path) if file.endswith('.jpg')]
    if not dataList:
        print("No images found in the database.")
        return
    for data in dataList:
        curImage = cv2.imread(f'{path}/{data}')
        if curImage is None:
            print(f"Error reading image {data}. Skipping...")
            continue
        imageList.append(curImage)
```

```
    curName = os.path.splitext(data)[0]
    personName.append(curName)
print('Starting Encoding of Known Images in Database...')
encodedKnown = findEncoding(imageList, personName)
print('Encoding of Known Images Completed...')
imageCheckList = []
dataCheckList = [file for file in os.listdir(directoryPath) if file.endswith('.jpg')]
if not dataCheckList:
    print("No images found in the provided directory.")
    return
for data in dataCheckList:
    curImage = cv2.imread(f'{directoryPath}/{data}')
    if curImage is None:
        print(f"Error reading image {data}. Skipping...")
        continue
    imageCheckList.append(curImage)
for image in imageCheckList:
    imageS = cv2.resize(image, (0, 0), None, 0.25, 0.25)
    imageS = cv2.cvtColor(imageS, cv2.COLOR_BGR2RGB)
    curFaceFrame = face_recognition.face_locations(imageS)
    curEncoding = face_recognition.face_encodings(imageS, curFaceFrame)
    for encoding, faceFrame in zip(curEncoding, curFaceFrame):
        result = face_recognition.compare_faces(encodedKnown, encoding)
        faceDist = face_recognition.face_distance(encodedKnown, encoding)
        Index = np.argmin(faceDist)
        y1, x2, y2, x1 = faceFrame
        y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
        if faceDist[Index] < 0.55 and result:
            name = personName[Index].upper()
            print(name)
            cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(image, f'{name}', (x1, y1-5),
```

```
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0))
                date_str = datetime.now().strftime("%Y-%m-%d")
                filename = f"AttendanceImage{date_str}.csv"
                markAttendance(name, filename)
            else:
                cv2.rectangle(image, (x1, y1), (x2, y2), (0, 0, 255), 3)
                cv2.putText(image, f'Unknown', (x1, y1 - 6),
cv2.FONT_HERSHEY_COMPLEX, 3, (0, 0, 255), 5)
    cv2.imshow("Picture", image)
    cv2.waitKey(0)
```

7. TESTING

Testing is a critical phase in the development of the Next-Gen Attendance System to ensure the models and overall application perform accurately, reliably, and efficiently. The primary goal of testing is to identify and resolve errors, validate system functionalities, and confirm that the system meets the expected requirements for face detection-based attendance marking.

7.1 UNIT TESTING

Face Detection Model (YOLOv8)

Unit testing for the YOLOv8 model ensures that it correctly detects faces in different environments. The input validation checks whether the model accepts images and videos in the correct shape and format. Each layer of the model—such as convolutional, feature extraction, and classification layers—is tested for proper configuration and accurate data processing. The model's output is verified to ensure it correctly labels faces and assigns unique IDs. The inference time is measured to confirm the system delivers real-time detection without significant delays.

Attendance Recording Module

Unit testing for the attendance recording module ensures that detected faces are correctly mapped to student records. The system verifies that each individual's attendance is recorded only once per day, preventing duplicate entries. The correctness of time-stamped entries and the ability to update attendance sheets in different formats (Excel, CSV) are also tested.

Data Preprocessing Pipeline

During preprocessing, unit testing ensures that video frames and images are properly extracted and resized for efficient face detection. Noise reduction techniques are validated to confirm they enhance detection accuracy. The system checks for proper grayscale conversion (if needed) and ensures images are fed into the model in the expected format.

Edge Case Testing

Unit testing for edge cases ensures that the system handles unexpected inputs gracefully. The system is tested with:

- Empty or Corrupted Images/Videos – Ensuring meaningful error messages are displayed.
- Obstructed Faces – Checking if the model can still detect partial faces or reject unclear ones.
- Different Lighting Conditions – Testing detection accuracy in dim, bright, and uneven lighting conditions.

7.2 INTEGRATION TESTING

To perform integration testing for the Next-Gen Attendance System, several modules are tested to ensure seamless interaction and accurate attendance recording.

Webcam/Video/Image Input Handling

Ensures that the system correctly accepts video streams, static images, and pre-recorded videos. The system verifies that invalid file formats (e.g., text files) are rejected with appropriate error messages.

```
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        file = request.files.get('file')
        if not file:
            return render_template('index.html', message="No file uploaded!")
        if not file.filename.endswith(('.jpg', '.jpeg', '.png', '.mp4')):
            return render_template('index.html', message="Invalid file format!")
        filepath = os.path.join('uploads', file.filename)
        file.save(filepath)
        return process_file(filepath)  # Proceed to processing
    return render_template('index.html')
```

Face Detection Module Integration

Ensures that frames from a webcam, image, or video are correctly processed for face detection.

```
def detect_faces(image_path):
```

```
    try:
        image = cv2.imread(image_path)
        results = model.predict(image)
        return results
    except Exception as e:
        return str(e)
```

Attendance Marking Integration

Verifies that the detected faces are correctly mapped to existing records, ensuring no duplicate entries.

```
def mark_attendance(face_id):
    try:
        if face_id not in recorded_attendance:
            recorded_attendance.add(face_id)
            log_attendance(face_id)
        return "Attendance Marked"
    except Exception as e:
        return str(e)
```

Full Integration Pipeline in Flask

Ensures seamless integration from file upload to attendance marking.

```
def process_file(filepath):
    try:
        detected_faces = detect_faces(filepath)  # Step 1: Detect Faces
        if isinstance(detected_faces, str):
            return render_template('index.html', message=f"Detection Error: {detected_faces}")

        # Step 2: Mark Attendance
        for face_id in detected_faces:
            mark_attendance(face_id)
        return render_template('index.html', message="Attendance Successfully Recorded!")
    except Exception as e:
```

```
return render_template('index.html', message=f"System Error: {str(e)}")
```

Error Handling Validation

Ensures that errors are properly detected and reported to the user with meaningful messages.

7.3 SYSTEM TESTING

System testing ensures that the entire Next-Gen Attendance System—including the YOLOv8 model, Flask backend, and frontend—works seamlessly as a complete unit.

Functional Testing

- Ensures valid images and videos are processed correctly and invalid formats are rejected.
- Confirms face detection accuracy in different scenarios.
- Verifies that attendance is correctly recorded in Excel and prevents duplicates.
- Tests the web interface for proper response and usability.

Non-Functional Testing

- Performance Testing – Measures response time for real-time detection and large video processing.
- Usability Testing – Ensures the interface is user-friendly for teachers and administrators.
- Reliability Testing – Verifies that attendance is recorded accurately across multiple sessions.
- Security Testing – Ensures unauthorized users cannot alter attendance records.

Integration Testing Validation

- Ensures smooth interaction between face detection, attendance logging, and the database.
- Confirms that Flask APIs handle requests properly and return expected results.

Error Handling

- Ensures clear messages are provided for invalid inputs (e.g., unsupported files, unreadable videos).
- Tests the system's resilience against network failures and system crashes.

Test case 1:

Fig 7.1 illustrates the process of facial recognition, where detected faces are mapped to their corresponding roll numbers. This ensures accurate identification and automated attendance marking.

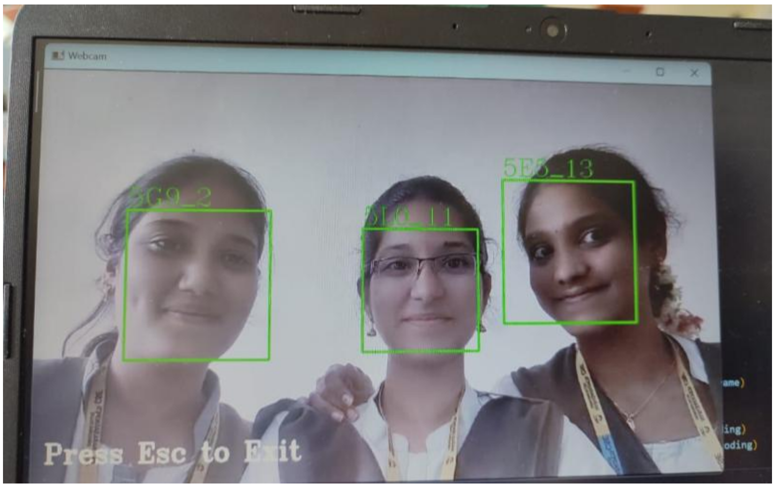


Fig 7.1 FACIAL RECOGNITION WITH ROLL NO MAPPING

Test case 2:

Fig 7.2 demonstrates multiple facial recognition, where the system detects and maps multiple faces to their respective roll numbers simultaneously. This enables efficient and automated attendance marking in group settings.

Fig 7.2 MULTIPLE FACIAL RECOGNITION WITH ROLL NO MAPPING

Test case 3:

Fig 7.3 illustrates the detection of unknown faces, where the system highlights unidentified individuals with a red bounding box. This helps in distinguishing unauthorized or unregistered persons in the attendance system.

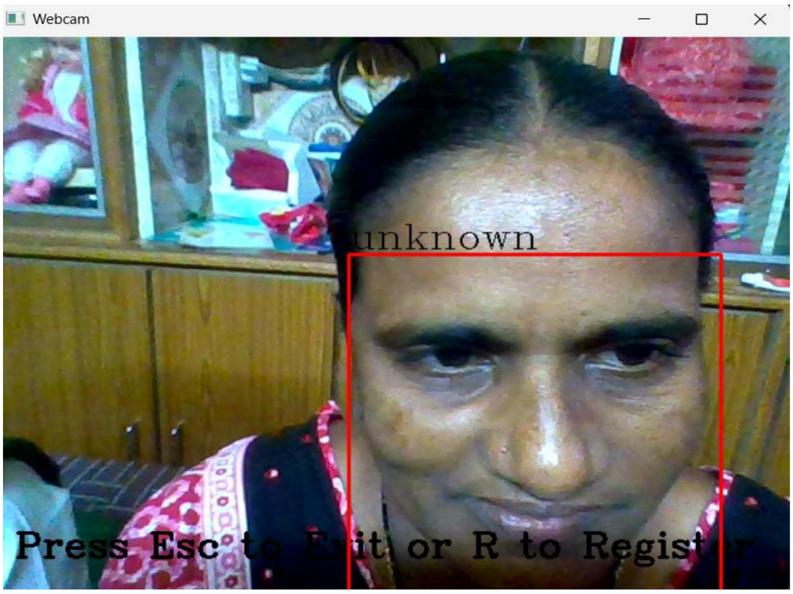


Fig 7.3 UNKNOWN FACE DETECTION WITH RED BOUNDING BOX

8.RESULT ANALYSIS

System Features:

- **Modes of Attendance Marking**

- *Live Images*: Teachers can take pictures with a webcam for marking attendance.

-*Pre-recorded Videos:* Teachers can upload videos to record attendance.

-*Static Images:* Teachers can mark attendance using still photos.

- **Face Detection Features**

- *Real-time Detection:* It detects and labels faces within real-time webcam sessions using a green box around the face and ID.

-*Video-based Detection:* This is a system that reads through recorded videos to automatically recognize and tag all the faces in a classroom.

Overall, using CNNs in the smart attendance system has shown to improve accuracy, reliability, speed, and the ability to handle larger data effectively.

Visual Examples:

Once the teacher selects the year and section then a marking attendance menu screen appears. They can then capture live images of the students via webcam, upload an existing video, or use a static image. And once one of the options is selected, the system utilizes it to mark attendance for the students in that section allowing the teacher flexibility to choose the method which is best suited to their needs.

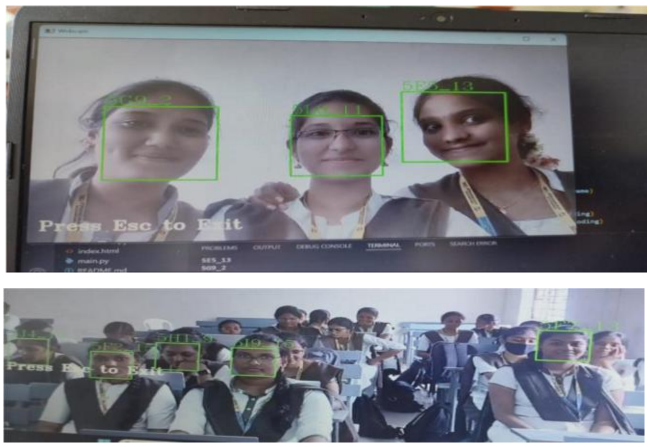


Fig 8.1: FACE DTECTION SYSTEM FOR REAL-TIME AND VIDEO-BASED INPUTS

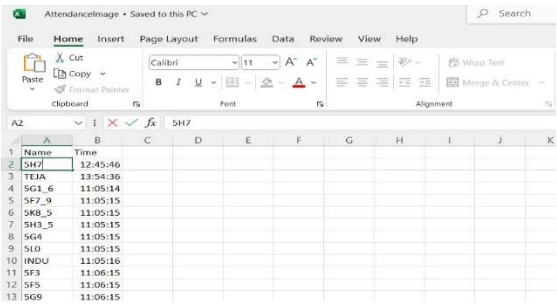
This project represents a face detection system that can identify and tag faces in real-time using a webcam, or from video files.

In Fig 8.1:

-A first image shows the system detecting faces live through a webcam[15]. Every face appears highlighted by a green box and with a unique label, proving its real time ability of recognition.

-A second image is the working version of the system using the pre-recorded video. In the given instance, it detects and labels several faces in the class. The system is given evidence of how well it could be used in other environments.

It further maintains attendance details. It captures face images through a webcam or video and identifies the captured faces through internal algorithms. The face is matched with particular registers and automatically records attendance along with roll number and time attended. Such details are saved in an Excel sheet for easier and error-free tracking and record without manual inputting.



Name	Time
5H7	12:45:46
TEJA	13:54:36
5G1_6	11:05:14
5F7_9	11:05:15
5K8_5	11:05:15
5H3_5	11:05:15
5G4	11:05:15
5L0	11:05:15
INDU	11:05:16
5F3	11:06:15
5F5	11:06:15
5G9	11:06:15

Fig 8.2: ATTENDANCE STORED IN EXCEL

The image (Fig 8.2) shows an Excel sheet used to keep track of attendance. Each row lists a student’s roll number under the "Name" column, along with the time they were recorded under the "Time" column[15]. This method helps keep a clear record of who attended and when. It’s an easy way to organize and find each student's attendance information.

Information will be captured into the excel sheet using a live webcam either in the form of images or video and each roll number of the students along with the time while recording the attendance.

9.OUTPUT SCREENS

The Next-Gen Attendance System is an advanced AI-powered solution designed to streamline attendance tracking with real-time face recognition technology. By leveraging deep learning models like YOLOv8, our system ensures accuracy, security, and efficiency, eliminating manual errors and preventing proxy attendance. Whether for educational institutions, corporate offices, or large organizations, our automated and scalable solution adapts to diverse needs.

Fig 9.1 displays the Home Screen of the Next-Gen Attendance System, providing users with easy navigation options. It serves as the central interface for accessing attendance modes, reports, and system settings.

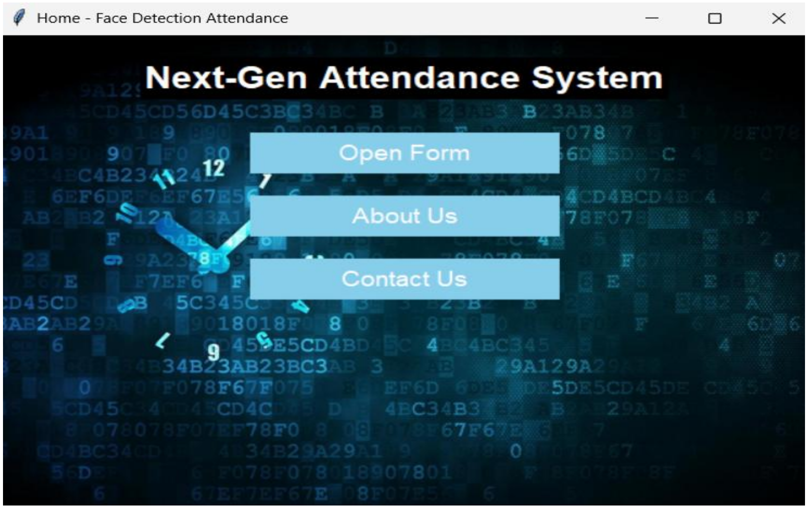


Fig 9.1 HOME SCREEN

Fig 9.2 With multiple attendance modes—live webcam detection, video analysis, and static images—attendance management becomes seamless. Our system also features secure data storage, anti-spoofing measures, and real-time analytics, making attendance tracking smarter and more reliable than ever.

Fig 9.2 ABOUT US PAGE

Fig 9.3 shows for any inquiries, support, or collaborations, feel free to reach out to us. You can contact us via email, phone, or our official website. Our team is always available to assist you with any questions regarding the Next-Gen Attendance System and its integration into your organization. Stay connected with us to experience the future of automated attendance management!

Fig 9.3 CONTACT US PAGE

Fig 9.4 Student Information Form Student Information Form This is a simple interface developed with the Python Tkinter library. This interface will help gather key student details such as year, semester, branch, and section, which would otherwise be cumbersome to track and store. The form helps in making attendance recording even easier and saves the information in a database for future academic needs.

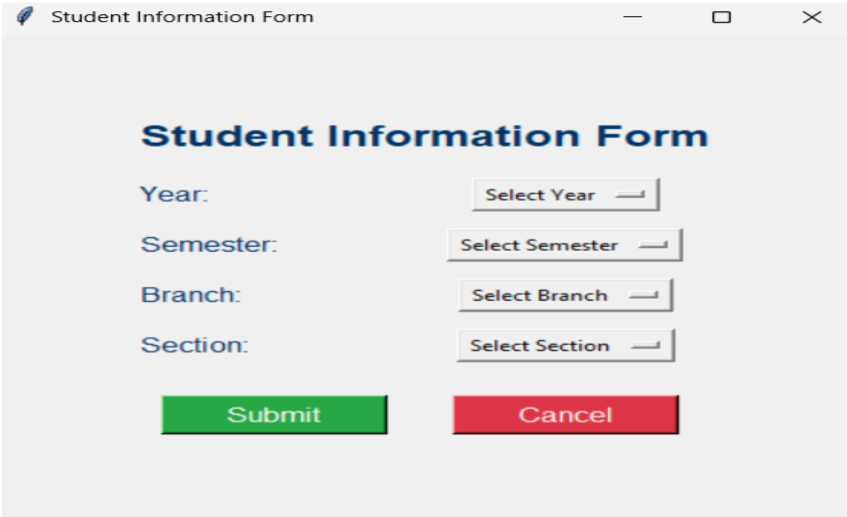


Fig 9.4 OUTPUT SCREEN OF THE STUDENT INFORMATION FORM

Fig 9.5 presents the main page for tracking attendance, where administrators can monitor real-time attendance records. It provides an overview of marked attendances, student details, and status updates.

Fig 9.5 MAIN PAGE TO TRACK ATTENDANCE

Application Interface As shown in Fig 9.5, the main application interface of the Smart Attendance System has provided different ways through which users can perform attendance. The different things that a user can do here include; take an attachment of pictures into the database, use the live webcam for real-time tracking, import images and video and when the data is captured once they have opened the attendance sheet where they quickly handle it. This provides easier detection of attendance in a flexible manner allowing multiple inputs according to the needs of users.

10. CONCLUSION

This project represents a revolutionary leap in the way attendance is tracked, leveraging advanced deep learning techniques to automate the entire process through facial recognition. At its core, the system utilizes the YOLOv8 model, a state-of-the-art deep learning algorithm known for its speed and accuracy in detecting faces in real time. This powerful technology works seamlessly with everyday hardware devices such as webcams, making the system both accessible and affordable for schools, offices, and various organizations. The integration of deep learning ensures high precision in face detection, enabling the system to identify individuals with minimal error, even in dynamic environments[11].

The Next-Gen Attendance System automates attendance tracking, reducing human error and preventing proxy attendance. It saves time while ensuring accuracy and reliability. Real-time database updates allow instant user additions without disruptions. Its flexibility and scalability make it suitable for various environments.

The system records attendance in daily Excel sheets, organized by sections, which makes it easier to manage, track, and analyse historical attendance data. The structured format of Excel files ensures that records are kept in an orderly manner, allowing for easy retrieval when needed. This can be particularly beneficial for educational institutions and workplaces that need to maintain accurate attendance records over long periods. Furthermore, the system is designed to prevent duplication, ensuring that each individual is only marked present once per day, which further guarantees the integrity and accuracy of the attendance data[12].

Overall, the Next-Gen Attendance System leverages deep learning and computer vision to automate attendance tracking efficiently. Its simplicity, affordability, and accuracy make it suitable for schools, universities, and offices. The system eliminates manual processes, ensuring real-time and reliable attendance recording. As technology advances, it can integrate enhanced security features and handle larger datasets. Its scalability allows adaptation to complex environments for improved functionality. This innovation represents the future of smart and secure attendance management.

11.FUTURE SCOPE

The Next-Gen Attendance System can be further enhanced in several ways, utilizing various cutting-edge technologies.

1. **Integration with School Management Systems (SMS)**: The attendance data can be integrated with existing school management systems using technologies such as APIs (Application Programming Interfaces) and cloud services like Google Cloud or AWS. This integration will streamline reporting and analysis, allowing real-time updates to student records. Data analytics tools can be applied to the integrated data to generate comprehensive reports on attendance trends, student engagement, and academic performance, offering a more holistic approach to managing student information.

2. **Support for Multiple Devices**: The system can be expanded to support mobile phones, tablets, and other devices using cross-platform frameworks like Flutter or React Native. These technologies would allow the system to be accessible on both Android and iOS devices, enabling remote attendance tracking. Bluetooth Low Energy (BLE) or Wi-Fi can also be used to facilitate proximity-based attendance marking, ensuring flexibility in both online and offline classroom settings.

3. **Improved Recognition**: The facial recognition model can be enhanced to handle more complex scenarios using advanced machine learning models and computer vision techniques. For instance, Convolutional Neural Networks (CNNs) can be fine-tuned to improve recognition in low-light conditions by using infrared cameras or thermal sensors. For handling large class sizes, techniques such as multi-face detection using YOLO (You Only Look Once) models or MTCNN (Multi-task Cascaded Convolutional Networks) can be implemented. Additionally, the system can benefit from edge computing to reduce latency and ensure faster face detection and recognition processing, especially in environments with numerous participants.

12.REFERENCES

[1] D. B. Hidayat, B. M. Nugraha, D. Andhika Ditya Bagaskara, D. P. Widyadhana, D. Purwitasari and I. K. E. Purnama, "A Classroom Usage Monitoring System with Image Detection for Student Attendance," 2024 2nd International Conference on Software Engineering and Information Technology (ICoSEIT), Bandung, Indonesia, 2024, pp. 7-12, doi: 10.1109/ICoSEIT60086.2024.10497524.K. B. Meena and V. Tyagi, Image Splicing Forgery Detection Techniques: A Review. Cham,Switzerland: Springer, 2021.

[2] Alruwais, N. M., & Zakariah, M. (2024). Student Recognition and Activity Monitoring in E-Classes Using Deep Learning in Higher Education. IEEE Access, 12, 66110-66125. DOI: 10.1109/ACCESS.2024.3354981.Abhishek and N. Jindal, Copy move and splicing forgery detection using deep convolution neural network, and semantic segmentation, Multimedia Tools Appl., vol. 80, no. 3, pp. 3571–3599, Jan. 2021.

[3] Alhanaee, K., Alhammadi, M., Almenhali, N., & Shatnawi, M. (2021). Face Recognition Smart Attendance System using Deep Transfer Learning. Procedia Computer Science, 192, 4093–4102. https://doi.org/10.1016/j.procs.2021.09.184

[4] Arya, S., Mesariya, H., & Parekh, V. (2023). Smart Attendance System Using CNN. Department of Computer Science, Pandit Deendayal Petroleum University, Gandhinagar, India.S. L. Jagannadham, K. L. Nadh and M. Sireesha, "Brain Tumour Detection Using CNN," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 734-739, doi: 10.1109/I-SMAC52330.2021.9640875

[5] P Sai Vasantha Lakshmi, Kumaraswamy, R., & Manhar, A. (2023). Smart Attendance Management System Using Geo-Fencing and Machine Learning. International Journal of Creative Research Thoughts (IJCRT), 11(6), e766-e772.Y.Rao,J.Ni and H.Zhao, Deep learning local descriptor for image splicing detection andlocalization, IEEE Access, vol. 8, pp. 21611–21621, 2020.

[6] Z. A. T. Ahmed, M. E. Jadhav, A. M. Al-madani, M. Tawfik, S. N. Alsubari,and A. A. A. Shareef, ‘‘Real-time detection of student engagement: Deeplearning-based system,’’ in Proc. Int. Conf. Innov. Comput. Commun(ICICC), vol. 1, 2022, pp. 313–323.Q.Li,C.Wang, X.Zhou, and Z.Qin, Image copy-move forgery detection & localization based on super-BPD segmentation and DCNN, Sci. Rep., vol. 12, no. 1, Sep. 2022

[7] S. Leelavathy, R. Jaichandran, K. Shantha Shalini, B. Surendar,A. K. Philip, and D. R. Raja, ‘‘Students’ attention and engagement prediction using machine learning techniques,’’ Eur. J. Mol. Clin. Med., vol. 7,no. 4, pp. 3011–3017, 2020.E.U.H.Qazi, T.Zia and A.Almorjan, Deep learning-based digital image forgery detection system, Appl. Sci., vol. 12, no. 6, p. 2851, Mar. 2022.

[8] M. U. Uçar and E. Özdemir, ‘‘Recognizing students and detecting student engagement with real-time image processing,’’ Electronics, vol. 11, no. 9,p. 1500, May 2022.

[9] Z. Trabelsi, F. Alnajjar, M. M. A. Parambil, M. Gochoo, and L. Ali,‘‘Real-time attention monitoring system for classroom: A deep learningapproach for student’s behavior recognition,’’ Big Data Cognit. Comput.,vol. 7, no. 1, p. 48, Mar. 2023.N. Krishnaraj, B. Sivakumar, R. Kuppusamy, Y. Teekaraman, and A. R. Thelkar, Designofautomated deep learning-based fusion model for copy-move image forgery detection, Comput. Intell. Neurosci., vol. 2022, pp. 1–13, Jan. 2022.

[10] I. Lasri, A. R. Solh, and M. E. Belkacemi, ‘‘Facial emotion recognition of students using convolutional neural network,’’ in Proc. 3rd Int. Conf. Intell.Comput. Data Sci. (ICDS), Oct. 2019, pp. 1–6K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in Proc.IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[11] M. Karunakar, C. A. Sai, K. Chandra and K. A. Kumar,“Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment,” International Journal forRecent Developments in Science and Technology, vol. 4, no. 5, pp. 194-201, 2020

[12] F. P. Filippidou and G. A. Papakostas, “Single Sample Face Recognition UsingConvolutional Neural Networks for Automated Attendance Systems,” in 2020

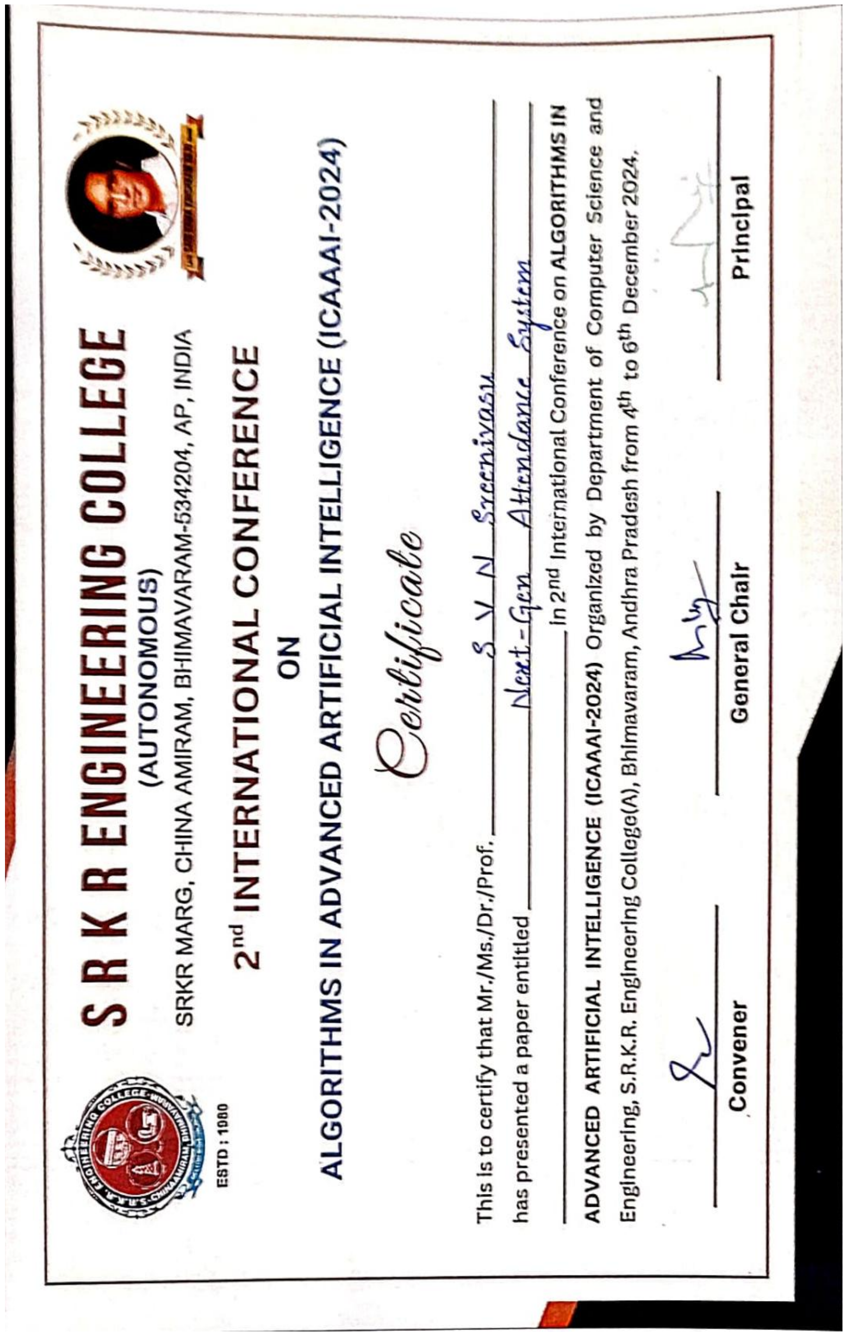
Fourth International Conference On IntelligentComputing in Data Sciences (ICDS), 2020.

[13] N. Soni, M. Kumar and G. Mathur, “Face Recognition using SOM Neural Network with Different Facial Feature Extraction Techniques,” International Journal of Computer Applications, vol. 76, no. 3, pp. 7-11, 2013.

[14] S. Matilda and K. Shahin, "Student Attendance Monitoring System Using Image Processing," 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, pp. 1-4, doi: 10.1109/ICSCAN.2019.8878806.

[15] Md. Y. Ali, X. Zhang, and H. Rashid, “Student activities detection of SUST using YOLOv3 on Deep Learning,” Indonesian Journal of Electrical Engineering and Informatics, vol. 8, no. 4, Dec. 2020, doi: 10.11591/ijeei.v8i4.2585.

CERTIFICATE- 1



ESTD : 1980

S R K R ENGINEERING COLLEGE

(AUTONOMOUS)

SRKR MARG, CHINA AMIRAM, BHIMAVARAM-534204, AP, INDIA

2nd INTERNATIONAL CONFERENCE

ON

ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)

Certificate

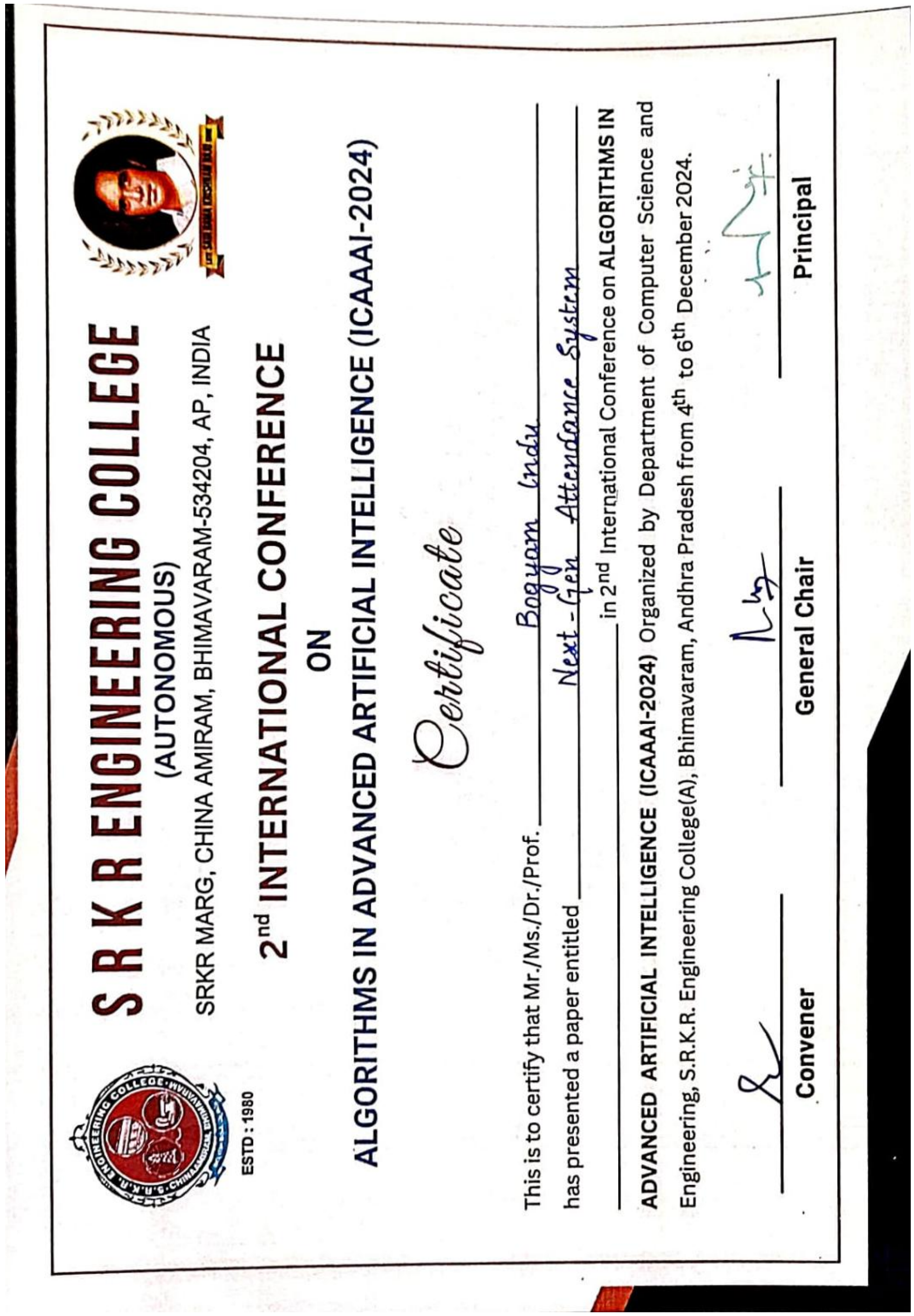
This is to certify that Mr./Ms./Dr./Prof. S V N Sreenivasu

has presented a paper entitled Next-Gen Attendance System

In 2nd International Conference on **ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)** Organized by Department of Computer Science and Engineering, S.R.K.R. Engineering College(A), Bhimavaram, Andhra Pradesh from 4th to 6th December 2024.

Convener | General Chair | Principal

CERTIFICATE - 2


ESTD : 1980

S R K R ENGINEERING COLLEGE

(AUTONOMOUS)

SRKR MARG, CHINA AMIRAM, BHIMAVARAM-534204, AP, INDIA

2nd INTERNATIONAL CONFERENCE

ON

ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)

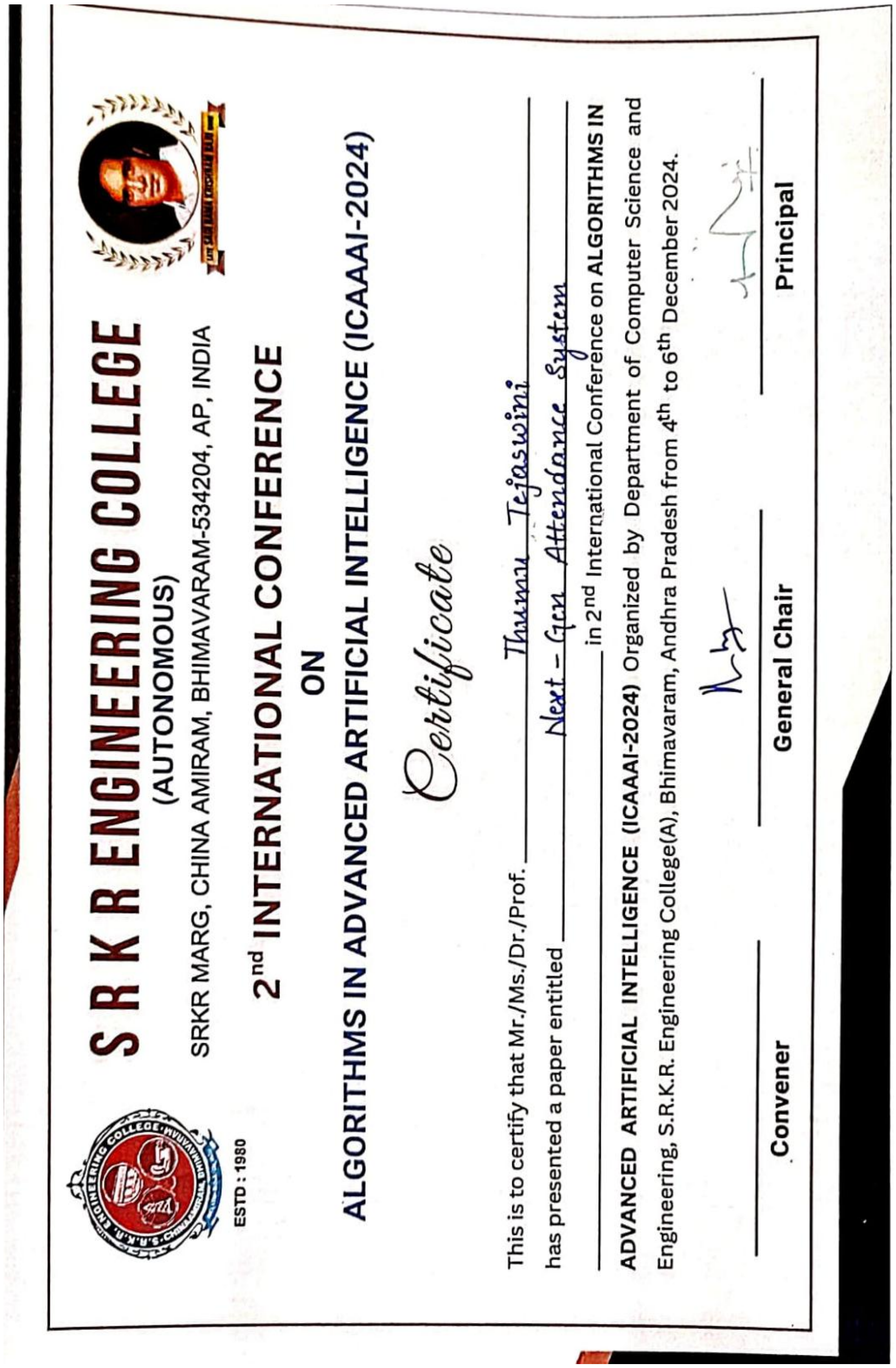
Certificate

This is to certify that Mr./Ms./Dr./Prof. Bogyam Indu

has presented a paper entitled Next-Gen Attendance System in 2nd International Conference on **ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)** Organized by Department of Computer Science and Engineering, S.R.K.R. Engineering College(A), Bhimavaram, Andhra Pradesh from 4th to 6th December 2024.

Convener **General Chair** **Principal**

CERTIFICATE - 3



S R K R ENGINEERING COLLEGE

(AUTONOMOUS)

SRKR MARG, CHINA AMIRAM, BHIMAVARAM-534204, AP, INDIA

ESTD : 1980

2^{nd} INTERNATIONAL CONFERENCE

ON

ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)

Certificate

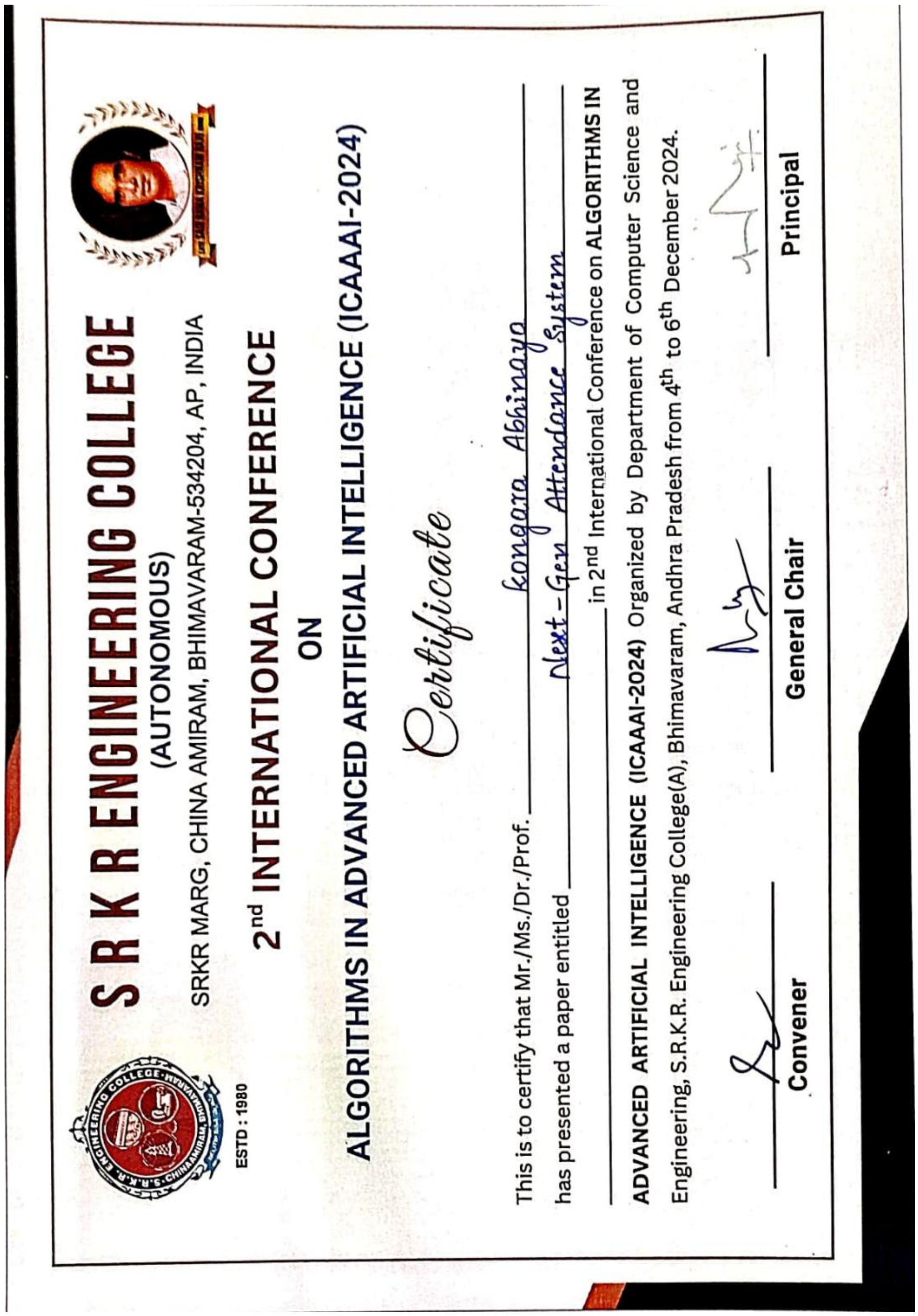
This is to certify that Mr./Ms./Dr./Prof. Thummu Tejaswini

has presented a paper entitled Next-Gen Attendance System

in 2^{nd} International Conference on **ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)** Organized by Department of Computer Science and Engineering, S.R.K.R. Engineering College(A), Bhimavaram, Andhra Pradesh from 4^{th} to 6^{th} December 2024.

Convener | General Chair | Principal

CERTIFICATE -4



ESTD : 1980

S R K R ENGINEERING COLLEGE

(AUTONOMOUS)

SRKR MARG, CHINA AMIRAM, BHIMAVARAM-534204, AP, INDIA

2nd INTERNATIONAL CONFERENCE

ON

ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)

Certificate

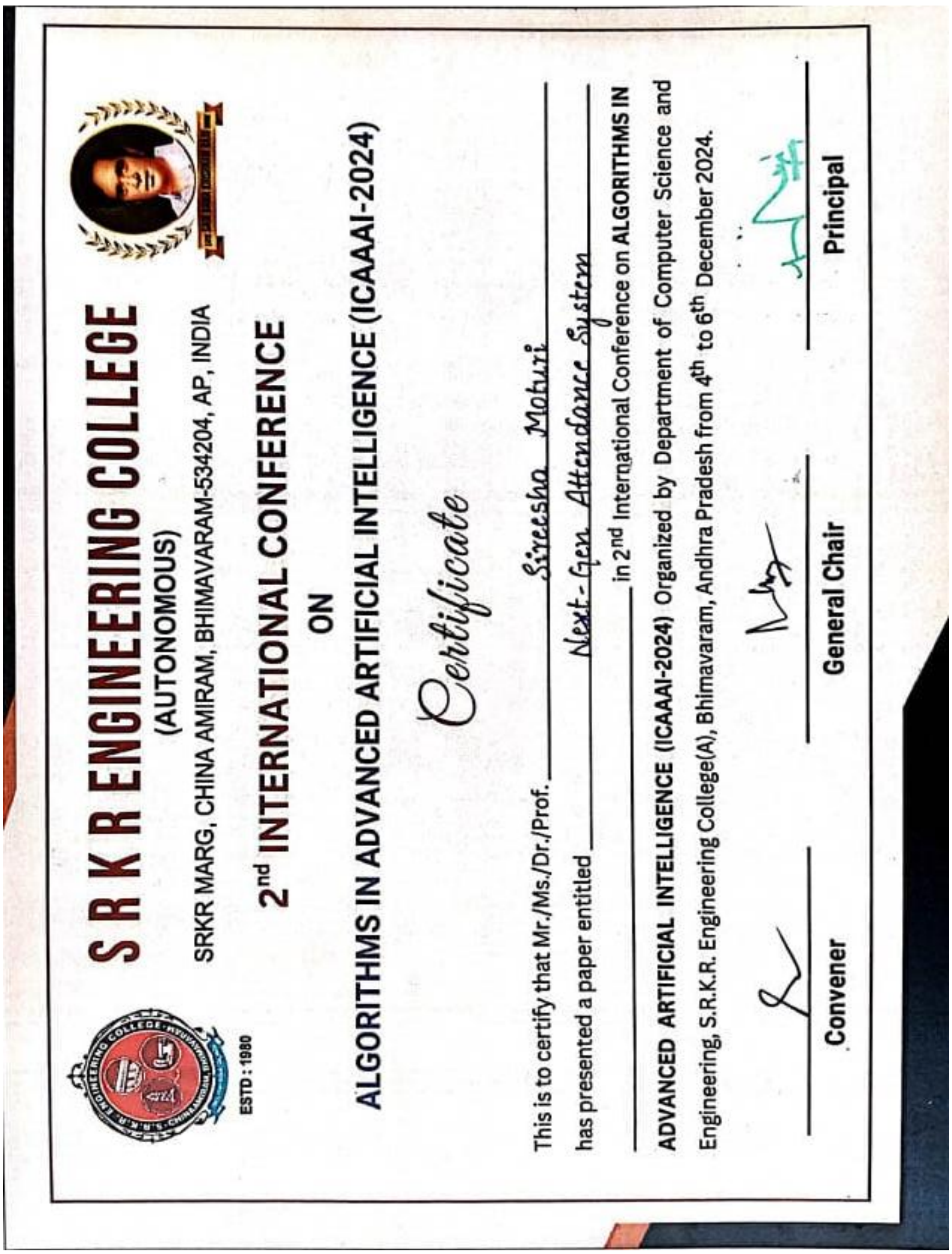
This is to certify that Mr./Ms./Dr./Prof. Kongara Abhinaya

has presented a paper entitled Next-Gen Attendance System

in 2nd International Conference on **ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)** Organized by Department of Computer Science and Engineering, S.R.K.R. Engineering College(A), Bhimavaram, Andhra Pradesh from 4th to 6th December 2024.

Convener **General Chair** **Principal**

CERTIFICATE -5



S R K R ENGINEERING COLLEGE
(AUTONOMOUS)
SRKR MARG, CHINA AMIRAM, BHIMAVARAM-534204, AP, INDIA

ESTD : 1980

2nd INTERNATIONAL CONFERENCE
ON
ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)

Certificate

This is to certify that Mr./Ms./Dr./Prof. Sireesha Moturi
has presented a paper entitled Next-Gen Attendance System
in 2nd International Conference on **ALGORITHMS IN ADVANCED ARTIFICIAL INTELLIGENCE (ICAAAI-2024)** Organized by Department of Computer Science and Engineering, S.R.K.R. Engineering College(A), Bhimavaram, Andhra Pradesh from 4th to 6th December 2024.

Convener **General Chair** **Principal**

Next-Gen Attendance System

Dr.S.V.N Sreenivasu[1], Ch.Rajani[2], Bogyam Indu[3], Thumu Tejaswini[4], Kongara Abhinaya[5], and Dr.Sireesha Moturi[6]

Department of Computer Science and Engineering, Narasaraopeta Engineering College,Yellamanda Road, Narasaraopet 522601, Andhra Pradesh,India.

Abstract

The Next-Gen Attendance System is an automated advanced solution in the real-time tracking of attendance using deep learning models to streamline and enhance the process. With a wide capability of using YOLOv8 for very accurate face detection, this system offers three flexible modes for attendance capture: webcam live feeds, pre-recorded videos, or static images. These individual detected faces are uniquely labeled, and no same face gets recorded more than once in multiple sections. The system also makes use of CNN models to optimize facial feature and eye identification, even under dynamic environments. It significantly reduces the amount of manual effort and human error that is possible in attendance monitoring. The records are kept in secure easy to access excel sheets that uniquely identify each user. This solution has been designed for modern classrooms and pave the way for seamless integration of AI in attendance management systems within educational institutions.

Keywords: Webcam Attendance · Image Upload Attendance · Video Upload Attendance · Classroom Monitoring System · Student Attendance · Face detection · YOLOv8 · CNN

I. INTRODUCTION

This "Next-Gen Attendance System" breaks the mold in tracking attendance in educational spaces through automatic ways. Old methods like a traditional roll call or sign-in sheet for students tend to be time-consuming and inaccurate at times. This is a smart, web-based platform that employs deep learning algorithms to solve the problem of attendance monitoring with prompt and error-free processes intended for dynamic classroom settings. It is built with the YOLOv8 top-tier object detection framework for real-time face identification. Famous for its speed and accuracy, it can read faces at new angles, lighting, and conditions. It performs detection and classification in one pass with steady high-speed recognition that, of course is elemental in maintaining reliable records of attendance. Further, to increase the recognition precision, the system applies Convolutional Neural Networks(CNN) to dig out unique facial features. Such multi-layered networks have a strength towards capturing complex patterns in visual data. And thus they are increasingly needed for differentiating between individual faces[1]. Pre-trained CNN models enhance this ability and further increased reliability of the system in recording student attendance. The operating modes include live webcam feeds, as well as static images and pre-recorded videos, supported by the platform. Teachers can view an interactive interface for attendance tracking, where data is stored properly based on different Excel sheets for the corresponding sections[2]. This modular setup allows for easy scalability, thus tailored to diverse educational settings.

II. LITERATURE REVIEW

Hidayat et al., (2024)[1] (Alruvais and Zakarih[2] , Khwala Alhanai [3], Mitha Alhammadi, Nahla Almenhali, and Mad Shatnawi, Shailesh Arya, Hrithik Mesaria, and Vishal Parekh[4]) advance and review the current trends and practices that aim to enhance classroom management through technology. They recommend the provision of webcams in classrooms to detect movements and send alerts to staff concerning any detected motion. The report demonstrates the use of other approaches such as facial landmark tracking, facial segmentation, and prediction of gaze direction as among other activities undertaken to evaluate how the students behave. Also, they cover systems with the function of recognizing a face for attendance purposes. [1]Their approach is based on state-of-theart technologies that provide the framework for the development of a system for monitoring a classroom in real time through image analysis, improving the usability of classroom management tools. In other work, the author discusses automatic technologies used for controlling the location of pupils in e-learning environments and discusses in detail the application of deep learning such as Convolutional Neural Networks (CNNs) [2]. This work brings to light critical figures

in any research work, the students and most importantly their level of involvement using facial expression data, eye tracking, and head movements as examples. However, considerable achievements have been reached, advanced problems, particularly related to the requirement of huge volumes of information and enhancement in accuracy of tracking have come to light. On Knowledge-Based Intelligent Information Technology Systems conference, the researchers presented their work on a system for attendance based on face recognition. The main goal of utilizing transfer learning through pre-trained CNNs such as[3] SqueezeNet, GoogleNet, and AlexNet was to increase the accuracy of attendance management systems. Lastly, Shailesh Arya, Hrithik Mesaria, and Vishal Parekh from Pandit Deendayal Petroleum University[4] devised a Smart Attendance System through CNNs for real-time face recognition. Their system utilizes a Siamese network for enhancing accuracy and employs live camera feeds for detection and identification of students while automatically updating attendance in MongoDB database. This makes it more reliable and efficient than traditional forms of taking attendance. The article "Smart Attendance Management System Using Geo-Fencing and Machine Learning" by Sai Vasantha Lakshmi, Reddy Kumaraswamy, and Edwin Manhar [5] addresses the approach of an automated attendance system that merges geo-fencing with facial recognition using machine learning. The system uses GPS and API to create virtual fencing on diverse geographic locations based on student access, to record attendance only for students in some specific locations such as classrooms. It leverages models such as CNNs, VGGFace, or ResNet, for face identification and also uses liveness detection to prevent spoofing. This system automates attendance, cuts down on manual effort, and ensures high accuracy, making it valuable for schools and organizations. On the other side, the authors also suggest some future improvements.

Overall, this review marks advances in automated attendance and student monitoring, where the deep learning and the use of CNNs improve its accuracy, efficiency, and reliability.

III. METHODOLOGY

Next-Gen Attendance System is one that will use deep learning and computer vision for automatic attendance at educational institutes through facial recognition. It is necessary a webcam, face detection, recognition algorithms, and a structured database to record attendance [6] .Here’s how it all comes together and is developed and implemented.

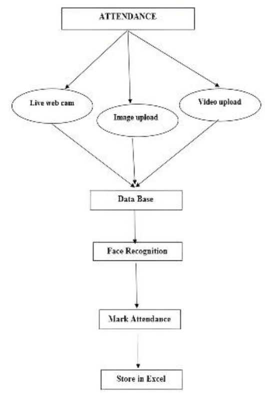


Fig.1. Flowchart of a Deep Learning-Based System for Student Monitoring and Recognition in Online Classes

Fig 1 illustrates a flowchart for automated attendance via face recognition, processing input from webcam, image, or video to enhance face detection and identification. Recognized faces are marked for attendance and logged in an Excel file for efficient tracking[13][14].

3.1 Data collection

The face recognition system needs a database of known faces, and in order for this to be structured, follows:

– A folder holds labelled facial images for each section, for example, ‘sectionC/database-C‘. They also take shots of the face of each individual from all angles, like 15 different directions, to extend the recognition precision.
– Images pre-processed [7] and stored using names as identifiers in a structured folder system as: ‘5L0-10‘, ‘5E5-2‘.

3.2 Face Detection and Recognition

The system utilizes YOLOv8 for the detection of faces in real-time and a face recognition library to carry out encodings and matching. This is how it works:

– Captures live video frames through webcam, resized them and transferred them from BGR to RGB for use in the face recognition model.
– *Face Detection:* YOLOv8 detects the bounding boxes around faces in the video frame.
– *Face Encoding:* The face recognition model encodes a detected face into a 128-dimensional encoding, and then it compares this with known face encodings in the database.

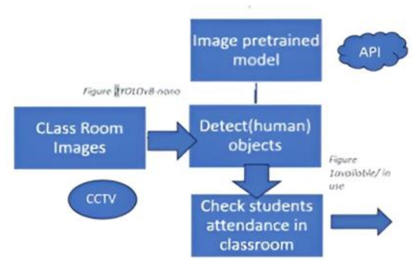


Fig.2. YOLOv8-Nano-Based System for Detecting Students and Checking Attendance in the Classroom

Fig 2 is the Student attendance system using a pre-trained YOLOv8[8] nano model captures classroom images, preprocesses them with a people detection model, and automatically identifies present students, reducing the need for manual attendance tracking[8].

3.3 Face Encoding and Matching

– *Encoding Known Faces:* Each face of the database is encoded to obtain unique features and hence retained for future reference.
– *A face matching:* if the face is detected in the video feed, then its encoding is matched against the stored encodings. If distance between the detected face and one of the stored face encodings is lower than the specified threshold, for example, 0.55, then this is classified as a match.

3.4 Attendance Marking

The system marks attendance by successful face recognition through the following steps:

– System checks whether attendance is already marked for a person whose face is detected on the same day.
– Attendance details are saved into an Excel file for each section, and the date name is something like ‘Attendance-2024-07-13.xlsx‘.
– For every record, the data includes name, date, and time. No further entry is done if a person has already recorded for that day.

3.5 Storage of Data

– Excel Files: Using pandas and ‘openpyxl‘, attendance data are stored in Excel files. Every section has a folder, and inside the folder, the attendance sheet is saved; one example can be the ‘section-C‘ folder.
– Daily Sheets: Files dated. Every day the system will produce new file, such as, section C will have its attendance file called ‘Attendance-2024-07-13.xlsx‘.
– Every individual attendee’s attendance is captured just once in a day to avoid duplication.

3.6 Real-time System Interaction

– The webcam was used for displaying live feed in the system. It draws bounding boxes around the detected faces, [9] recognized face using green box and carry name while the face could not be captured gets red box inside the bound.
– *Keyboard Input:* One may press the Ëscbutton to quit or ’R’ to enroll a new person. New captures are processed for encoding and added to the database in real time.

3.7 System Flow and Re-encoding

– Captured new faces, encoded, and saved to appropriate folder. Re-capture the face encodings to refresh the data. It puts the system under current updates.

3.8 Challenges Addressed

– Frame Stuttering: Improve performance by cropping frames to process on a lower resolution or 0.25x scale.
– Duplicate Attendance: Prevents the registration of multiple instances of the same individual for attending on the same date using Excel saved files.
– Dynamic Updates: Ability to capture and recognize new faces in real-time without having to turn off the system.

This methodology outlines the workflow and core technical components of your Next-Gen Attendance System, providing an overview of how the system was built, from face detection to attendance storage.

3.9 Model Evaluation

For model evaluation, accuracy can be determined based on the success rate of the face recognition process, while performance metrics could include the speed of processing and error rates in misidentifying or missing students. Evaluation could also involve testing different input methods [12](webcam, image, video) to ensure robustness across data formats. Additionally, storing the output in Excel provides an organized way to verify the model’s efficiency in maintaining attendance records.

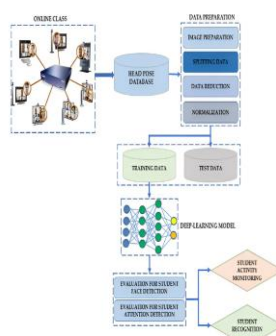


Fig.3. Attendance Management System Flowchart.

Fig 3 is the Flowchart of a deep learning setup to track student focus in online classes. Videos from students are processed, stored, and prepped, then split for training and testing. A model detects faces and tracks focus, checking accuracy in identifying faces and engagement.

IV. RESULT

4.1 System Features

Modes of Attendance Marking

- *Live Images:* Teachers can take pictures with a webcam for marking attendance.
- *Pre-recorded Videos:* Teachers can upload videos to record attendance. – *Static Images:* Teachers can mark attendance using still photos.

Face Detection Features

- *Real-time Detection:* It detects and labels faces within real-time webcam sessions using a green box around the face and ID.
- *Video-based Detection:* This is a system that reads through recorded videos to automatically recognize and tag all the faces in a classroom.

Fig.4. Student Information Form shows the interface used to collect and manage student data, ensuring streamlined attendance tracking and database management

Fig 4 Student Information Form created with Python Tkinter. This interface collects key student details—year, semester, branch, and section—streamlining attendance recording and storing data in a database for future use. Overall, using CNNs in the smart attendance system has shown to improve accuracy, reliability, speed, and the ability to handle larger data effectively.

Fig.5. Attendance can be taken through webcam, image and video options

Fig 5 is the Main interface of the Smart Attendance System offers flexible attendance options, including adding pictures to the database, using a live webcam for real-time tracking, and importing images or video. Captured data can be accessed in the attendance sheet, enabling efficient attendance management.

Visual Examples

Once the teacher selects the year and section then a marking attendance menu screen appears. They can then capture live images of the students via webcam, upload an existing video, or use a static image. And once one of the options is selected, the system utilizes it to mark attendance for the students in that section allowing the teacher flexibility to choose the method which is best suited to their needs. This project represents a face detection system that can identify and tag faces in real-time using a webcam, or from video files. In Fig 6:

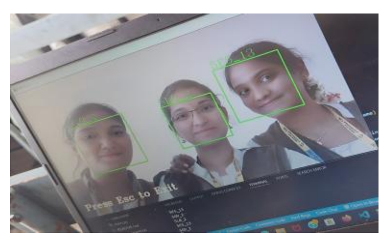


Fig.6. Face Detection System for Real-time and Webcam-based Input

Fig.7. Face Detection System for Real-time and Video-based Input

- Fig 6 shows the system detecting faces live through a webcam[15]. Every face appears highlighted by a green box and with a unique label, proving its real time ability of recognition.
- Fig 7 is the working version of the system using the pre-recorded video. In the given instance, it detects and labels several faces in the class. The system is given evidence of how well it could be used in other environments.

It further maintains attendance details. It captures face images through a webcam or video and identifies the captured faces through internal algorithms. The face is matched with particular registers and automatically records attendance along with roll number and time attended. Such details are saved in an Excel sheet for easier and error-free tracking and record without manual inputting.

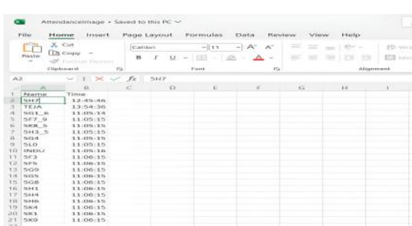


Fig.8. Attendance is stored in excel sheet as per date along with the timestamp and roll number

The image Fig 8 shows an Excel sheet for tracking attendance, with each student's roll number in the "Name" column and the recorded time in the "Time" column[15]. Attendance data is captured via a live webcam from images or video, organizing clear records of each student's attendance and time of capture.

V. CONCLUSION

The Next-Gen Attendance System is a high-tech solution developed with deep learning-based automation of attendance through facial recognition. It utilizes YOLOv8 models for in real time on face detection and facial recognition algorithms for encoding, enabling an efficient and precise identification of individuals. Its simplicity brings it in unison with the usual hardware devices like webcams, hence making it accessible and affordable. The system eliminates such manual attendance tasks, minimizes the error and fraud such as proxy attendance, and can update its database in real time, providing easy provision to accommodate new users without interrupting operations. Attendance records are kept in daily Excel sheets for each section, that enable easy access to historical data and eventual linkage with even larger data management systems. The system also prevents any form of duplication, hence accuracy is guaranteed.

This paper merely showcases how deep learning and computer vision can transform ordinary work, such as tracking attendance, into something so much different. It is lovely, helpful for schools and offices and many other organizations to have this sort of thing, and it is pretty accurate. Security could be enhanced in the future and scalability in terms of systems and massive environments and datasets.

VI. REFERENCES

1. D. B. Hidayat, B. M. Nugraha, D. Andhika Ditya Bagaskara, D. P. Widyadhana,D. Purwitasari, and I. K.E. Purnama, "A Classroom Usage Monitoring System with Image Detection for Student Attendance," in *2024 2nd Int. Conf. Softw. Eng.*

Inf. Technol. (ICoSEIT),Bandung, Indonesia, 2024, pp. 7–12, doi: 10.1109/ICoSEIT60086.2024.10497524.

2. N. M. Alruwais and M. Zakariah, “Student Recognition and Activity Monitoringin E-Classes Using Deep Learning in Higher Education,” *IEEE Access*, vol. 12, pp. 66110–66125, 2024, doi: 10.1109/ACCESS.2024.3354981.
3. K. Alhanaee, M. Alhammadi, N. Almenhali, and M. Shatnawi, “Face RecognitionSmart Attendance System using Deep Transfer Learning,” *Procedia Comput. Sci.*, vol. 192, pp. 4093–4102, 2021, doi: 10.1016/j.procs.2021.09.184.
4. S. Arya, H. Mesariya, and V. Parekh, “Smart Attendance System Using CNN,”Department of Computer Science, Pandit Deendayal Petroleum University, Gandhinagar, India, 2023.
5. P. S. V. Lakshmi, R. Kumaraswamy, and A. Manhar, “Smart Attendance Management System Using Geo-Fencing and Machine Learning,” *Int. J. Creat. Res. Thoughts (IJCRT)*, vol. 11, no. 6, pp. e766–e772, 2023.
6. Z. A. T. Ahmed, M. E. Jadhav, A. M. Al-madani, M. Tawfik, S. N. Alsubari, and A.A. A. Shareef, “Real-time Detection of Student Engagement: Deep Learning-based System,” in *Proc. Int. Conf. Innov. Comput. Commun. (ICICC)*, vol. 1, pp. 313–323, 2022.
7. S. Leelavathy, R. Jaichandran, K. S. Shalini, B. Surendar, A. K. Philip, and D.R. Raja, “Students’ Attention and Engagement Prediction using Machine Learning Techniques,” *Eur. J. Mol. Clin. Med.*, vol. 7, no. 4, pp. 3011–3017, 2020.
8. M. U. Uçar and E. Özdemir, “Recognizing Students and Detecting Student Engagement with Real-time Image Processing,” *Electronics*, vol. 11, no. 9, p. 1500, May 2022.
9. Z. Trabelsi, F. Alnajjar, M. M. A. Parambil, M. Gochoo, and L. Ali, “Real-time Attention Monitoring System for Classroom: A Deep Learning Approach for Student’s Behavior Recognition,” *Big Data Cognit. Comput.*, vol. 7, no. 1, p. 48, Mar. 2023.
10. I. Lasri, A. R. Solh, and M. E. Belkacemi, “Facial Emotion Recognition of Studentsusing Convolutional Neural Network,” in *Proc. 3rd Int. Conf. Intell. Comput. Data Sci. (ICDS)*, Oct. 2019, pp. 1–6.
11. M. Karunakar, C. A. Sai, K. Chandra, and K. A. Kumar, “Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment,” *Int. J. Recent Dev. Sci. Technol.*, vol. 4, no. 5, pp. 194–201, 2020.
12. F. P. Filippidou and G. A. Papakostas, “Single Sample Face Recognition UsingConvolutional Neural Networks for Automated Attendance Systems,” in *2020 Fourth Int. Conf. Intell. Comput. Data Sci. (ICDS)*, 2020.
13. N. Soni, M. Kumar, and G. Mathur, “Face Recognition using SOM Neural Networkwith Different Facial Feature Extraction Techniques,” *Int. J. Comput. Appl.*, vol. 76, no. 3, pp. 7–11, 2013.
14. S. Matilda and K. Shahin, “Student Attendance Monitoring System Using ImageProcessing,” in *2019 IEEE Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Pondicherry, India, 2019, pp. 1–4, doi: 10.1109/ICSCAN.2019.8878806.
15. Md. Y. Ali, X. Zhang, and H. Rashid, “Student Activities Detection of SUST usingYOLOv3 on Deep Learning,” *Indones. J. Electr. Eng. Inform.*, vol. 8, no. 4, Dec. 2020, doi: 10.11591/ijeei.v8i4.2585.

cg1

ORIGINALITY REPORT

1%	1%	1%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Charu Agarwal, Pranjul Itondia, Anurag Mishra. "A Novel DCNN-ELM Hybrid Framework For Face Mask Detection", Intelligent Systems with Applications, 2023 Publication	<1%
2	Nuha Alruwais, Mohammed Zakariah. "Student Recognition and Activity Monitoring in E-Classes Using Deep Learning in Higher Education", IEEE Access, 2024 Publication	<1%
3	ebin.pub Internet Source	<1%
4	www.mdpi.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		