A PROJECT REPORT ON

INTELLIGENT VIDEO BASED ATTENDANCE SYSTEM

Submitted in partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

Dr. T KESAVA RAO, M.Tech , Ph.D.

Associate Professor / Computer Science and Engineering

BY

BATHALA PRAVEEN KUMAR	**20751A0522**
G RAHUL	**20751A0561**
BASIREDDY NARENDRA REDDY	**20751A0521**
E GNANA SAI	**20751A0548**

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES, CHITTOOR-517127, A.P.

(Autonomous – NBA Accredited)

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(2023 - 24)

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES, (Autonomous-NBA Accredited)

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor, A.P-517127.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"INTELLIGENT VIDEO BASED ATTENDANCE SYSTEM"** is a genuine work of

BATHALA PRAVEEN KUMAR	**20751A0522**
G RAHUL	**20751A0561**
BASIREDDY NARENDRA REDDY	**20751A0521**
E GNANA SAI	**20751A0548**

Submitted to the department of Computer Science and Engineering, in partial fulfillment Of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING from Jawaharlal Nehru Technological University Anantapur, Ananthapuramu.

Signature of the Supervisor

Dr. T. KESAVA RAO, M.Tech., Ph.D.,
Associate Professor,
Department of Computer Science and Engineering.

Signature of the head of the Department

Prof. M. E. PALANIVEL,
Professor & HOD,
Department of Computer Science and Engineering.

Submitted for University Examination (Viva-Voce) held on ……..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES, (Autonomous-NBA Accredited)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor, A.P-517127.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Institute Vision

To emerge as a Centre of Excellence for Learning and Research in the domains of engineering, computing and management.

Institute Mission

M1: Provide congenial academic ambience with state-art of resources for learning and research.

M2: Ignite the students to acquire self-reliance in the latest technologies.

M3: Unleash and encourage the innate potential and creativity students.

M4: Inculcate confidence to face and experience new challenges.

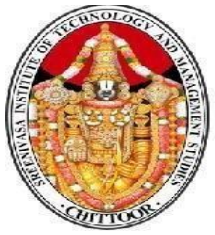
M5: Foster enterprising spirit among students.

Department Vision

To contribute for the society through excellence in Computer Science and Engineering with a deep Passion for wisdom, culture and values.

Department Mission

- Provide congenial academic ambience with necessary infrastructure and learning resources.
- Inculcate confidence to face and experience new challenges from industry and society.
- Ignite the students to acquire self-reliance in the latest technologies.
- Foster Enterprising spirit among students.



SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES, (Autonomous-NBA Accredited)

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor, A.P-517127.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAMME EDUCATIONAL OBJECTIVES (PEO's):

PEO1: Excel in Computer Science and Engineering program through quality studies, enabling successin computing industry. **(Professional Competency)**.

PEO2: Surpass in one's career by critical thinking towards successful services and growth of the organization, or as an entrepreneur or in higher studies. **(Successful Career Goals)**.

PEO3: Enhance knowledge by updating advanced technological concepts for facing the rapidly changing world and contribute to society through innovation and creativity **(Continuing Education and Contribution to Society)**.

PROGRAM SPECIFIC OUTCOMES (PSO's):

PSO1: Have Ability to understand, analyses and develop computer programs in the areas likealgorithms, system software, web design, big data analytics, and networking.

PSO2: Deploy the modern computer languages, environment, and platforms in creating innovativeproducts and solutions.

PROGRAMME OUTCOMES (PO's):

PO1 - Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2 - Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3 - 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

PO4 - Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 - 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.

PO6 - 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 theprofessional engineering practice.

PO7 - 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.

PO8 - Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9 - Individual and team work: Function effectively as an individual, and as a member or leader indiverse teams, and in multidisciplinary settings.

PO10 - 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.

PO11 - 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.

PO12 - Life-long learning: Recognize the need for, and have the preparation and ability to engage inindependent and life-long learning in the broadest context of technological change.

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES, (Autonomous-NBA Accredited)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor, A.P-517127.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Out Comes for Project Work

On completion of project work the student will be able to

CO1. Demonstrate in-depth knowledge on the project topic.

CO2. Identify, analyze and formulate complex problem chosen for project work to attain substantiated conclusions.

CO3. Design solutions to the chosen project problem.

CO4. Undertake investigation of project problem to provide valid conclusions.

CO5. Use the appropriate techniques, resources and modern engineering tools necessary forproject work.

CO6. Apply project results for sustainable development of the society.

CO7. Understand the impact of project results in the context of environmental sustainability.

CO8. Understand professional and ethical responsibilities while executing the project work.

CO9. Function effectively as individual and a member in the project team.

CO10. Develop communication skills, both oral and written for preparing and presenting project report.

CO11. Demonstrate knowledge and understanding of cost and time analysis required for carrying out the project.

CO12. Engage in lifelong learning to improve knowledge and competence in the chosen areaof the project.

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES, (Autonomous-NBA Accredited)

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor, A.P-517127.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CO – PO MAPPING

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO.1	3	-	-	-	-	-	-	-	-	-	-	-	3	3
CO.2	-	3	-	-	-	-	-	-	-	-	-	-	3	3
CO.3	-	-	3	-	-	-	-	-	-	-	-	-	3	3
CO.4	-	-	-	3	-	-	-	-	-	-	-	-	3	3
CO.5	-	-	-	-	3	-	-	-	-	-	-	-	3	3
CO.6	-	-	-	-	-	3	-	-	-	-	-	-	3	3
CO.7	-	-	-	-	-	-	3	-	-	-	-	-	3	3
CO.8	-	-	-	-	-	-	-	3	-	-	-	-	3	3
CO.9	-	-	-	-	-	-	-	-	3	-	-	-	3	3
CO.10	-	-	-	-	-	-	-	-	-	3	-	-	3	3
CO.11	-	-	-	-	-	-	-	-	-	-	3	-	3	3
CO.12	-	-	-	-	-	-	-	-	-	-	-	3	3	3
CO	3	3	3	3	3	3	3	3	3	3	3	3	3	3

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES, (Autonomous-NBA Accredited)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor, A.P-517127.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Evaluation Rubrics for Project Work

Rubric (CO)	Excellent (wt = 3)	Good (wt = 2)	Fair (wt =1)
Selection of Topic (CO1)	Selected a latest topic through complete knowledge of facts and Concepts	Selected a topic through partial knowledge off acts and concepts	Selected at paythrough improper knowledge of facts and concepts
Analysis and Synthesis (CO2)	Thorough comprehension through analysis/ synthesis	Reasonable comprehension through analysis/ synthesis	Improper comprehension through analysis/ synthesis
Problem Solving (CO3)	Thorough comprehension about what is proposed in the literature papers	Reasonable comprehension about what is proposed in the literature papers	Improper comprehension about what is proposed in the literature
Literature Survey (CO4)	Extensive literature survey with standard References	Considerable literature survey with standard References	Incomplete literature survey with substandard References
Usage of Techniques &Tools (CO5)	Clearly identified and has complete knowledge of techniques & tools used in the project work	Identified and has sufficient knowledge of techniques & tools used in the project work	Identified and has inadequate knowledge of techniques & tools used in project work
Project work impact on Society (CO6)	Conclusion of project work has strong impact on society	Conclusion of project work has considerable impact on society	Conclusion of project work has feeble impact on society
Project work impact on Environment (CO7)	Conclusion of project work has strong impact on Environment	Conclusion of project work has considerable impact on environment	Conclusion of project work has feeble impact on environment
Ethical attitude (CO8)	Clearly understands ethical and social practices.	Moderate understanding of ethical and social practices.	Insufficient understanding of ethical andsocial practices.
Independent Learning (CO9)	Did literature survey and selected topic with little Guidance	Did literature survey and selected topic with considerable guidance	Selected a topic as suggested by the Supervisor
Oral Presentation (CO10)	Presentation in logical sequence with key points, clear conclusion and excellent language	Presentation with key points, conclusion and good language	Presentation with insufficient key points and improper Conclusion
Report Writing (CO10)	Status report with clear and logical sequence of chaptersusing excellent language	Status report with logical sequence of chapters using understandable language	Status report not properly organized
Time and Cost Analysis (CO11)	Comprehensive time and cost analysis	Moderate time and cost analysis	Reasonable time and cost analysis
Continuous learning (CO12)	Highly enthusiastic towards continuous	Interested in continuous learning	Inadequate interest in continuous learning

ACKNOWLEDGEMENT

A Project of this magnitude would have not been possible without the guidance and co- ordination of many people. I am fortune in having top quality people to help, support and guide us in every step towards our goal.

Our team is very much grateful to the Chairman **Sri K. Ranganadham** Garu for his encouragement and stalwart support. We are also extremely indebted to the Secretary **Sri D.K. Badri Narayana** Garu for his constant support.

We express our sincere thanks to our Academic Advisor **Dr. K.L. Narayana**, **M.Tech., Ph.D.,** further, we would like to express our profound gratitude to our principal **Dr.N.Venkatachalapathi, M.Tech, Ph.D.** for providing all possible facilities throughout the completion of our project work.

We express our sincere thanks to our Dean (Academics), **Dr.M.Saravanan, M.E., Ph.D.,** further we express our sincere thanks to our Head of the Department **Prof. M.E.PALANIVEL ,** for his co-operation and valuable suggestions towards the completion of project work.

We express our sincere thanks to our guide **Dr.T. Kesava Rao., M.Tech, Ph.D,** for offering us the opportunity to do this work under his guidance.

We express our sincere salutation to all other teaching and non-teaching staff of our department for their direct and indirect support given during our project work. Last but not the least, we dedicate this work to our parents and the Almighty who have been with us throughout and helped us to overcome the hard times

DECLARATION

I certify that

- The work contained in this report is original and has been done by me under the Guidance of my supervisor.
- The work has not been submitted to any other Institute for any degree or diploma.
- I have followed the guidelines provided by the Institute in preparing the report.
- I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

BATHALA PRAVEENKUMAR	**20751A0522**
G RAHUL	**20751A0561**
BASIREDDY NARENDRA REDDY	**20751A0521**
E GNANA SAI	**20751A0548**

ABSTRACT

Facial features are now widely used for identity verification in attendance systems across educational institutions and libraries. Traditional approach for attendance is professor calls student name and record attendance. It takes some time to record attendance. If the duration of class of one subject is about 50 minutes, it takes 5 to 10 minutes to record attendance. For each lecture this is a waste of time. To avoid these losses, an automatic process based on image processing will be used. Identification of individuals in organizations or colleges for the purpose of attendance marking is one such application. The proposed system automates the process of marking attendance by utilizing image processing and facial recognition technology, aiming to overcome the drawbacks of conventional methods. It works by capturing video input through surveillance cameras or webcams. This video stream undergoes real-time processing with Convolutional Neural Networks (CNNs) to identify and extract facial features in each frame. These extracted features are then compared with a database containing facial data of registered individuals. When a match is found, the system records the individual as present. This approach enables efficient attendance tracking without requiring manual input. Additionally, the system is designed to handle variations in lighting conditions, facial expressions, and minor differences in pose, ensuring accurate identification.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
LSTM	Long Short-Term Memory
LBPH	Local Binary Pattern Histograms
SQL	Structured Query Language
SSD	Solid State Drive
UI	User Interface
UML	Unified Modeling Language
UAT	User Acceptance Testing
CI/CD	Continuous Integration/Continuous Deployment

CHAPTER-1

INTRODUCTION

1.1 Objective

The main goal of this project is to create an Intelligent Video-Based Attendance System that automates the process of taking and managing student attendance in schools and colleges. This system aims to improve speed and accuracy while generating easy-to-understand attendance reports. By automating attendance, we can save time and reduce the workload on teachers, making classrooms more productive and efficient. This system also aims to address common issues associated with manual attendance tracking. By using advanced face recognition technology, the system can accurately identify and record student attendance in real-time. This eliminates the need for manual roll calls and reduces the chances of errors or discrepancies in attendance records.

Furthermore, the automated attendance system can provide valuable insights and data analytics. It can track attendance patterns, identify frequent absentees, and generate comprehensive reports for administrative purposes. This data can be used to improve student engagement, identify potential issues early on, and enhance overall school or college management.

1.2 Problem Identification

Traditional attendance methods require teachers to call out student names and manually mark attendance on a sheet, which is both time-consuming and tedious. This process can take about five minutes per class, adding up to a significant amount of lost instructional time over a term. In large classrooms, especially those with multiple branches, verifying each student's presence individually is challenging. Moreover, manual systems often fail to track students who skip classes after breaks or lunchtime, particularly in high schools and lower grades. These issues highlight the need for an automated system that allows teachers to take attendance at any time, ensuring accurate tracking and reducing absentees. Manual attendance systems pose administrative challenges and lack real-time access to data.

1.3 Literature Survey

1. Deep Learning based Facial Recognition System for Attendance Maintenance (K.G. Saravanan, Jayamabel Rani, D.C. Jullie Josephine, M. Parameshwari, Hridya Venugopal, Janaki Ramal, November 2022)

This paper explores the use of deep learning techniques for developing a facial recognition system tailored for attendance maintenance. By leveraging advanced algorithms and methodologies, the system aims to improve accuracy and efficiency in tracking student attendance

2. Face Detection and its Features Extraction using Convolution Neural Network Model (Yenumaladoddi Jayasimha, Venkatesha M.R. Venkata Siva Reddy, May 2022)

This paper presents a Convolutional Neural Network (CNN) model for detecting faces and extracting features for facial analysis. It employs machine learning techniques, including the whale optimization algorithm, to optimize performance for tasks such as age estimation, gender classification, and emotion recognition.

3.Enhanced Facial Recognition System for Secure Attendance Monitoring in Educational Institutions (N. Prakash, S. Ananya, M. Aravind, December 2022)

This paper presents an enhanced facial recognition system designed for secure attendance monitoring in educational institutions. It discusses the integration of advanced image processing techniques and deep learning algorithms, such as Generative Adversarial Networks (GANs) and Long Short-Term Memory (LSTM) networks, for robust facial feature extraction and recognition. The system aims to improve accuracy and reliability in attendance tracking by addressing challenges such as variations in pose, illumination, and occlusion. Additionally, the paper explores the implementation of multi-modal biometric fusion techniques to enhance authentication accuracy and prevent unauthorized access.

4. Face recognition-based attendance system (Y. V. S. S. Avinash Kumar, 2021)

This paper introduces a system that uses face recognition technology to track student attendance in educational settings. It aims to streamline attendance management by

automatically identifying students' faces, eliminating the need for manual attendance-taking methods. The system utilizes tools like the Haar-cascade classifier and LBPH algorithm to achieve accurate facial recognition.

5. Facial recognition system for Automatic Attendance Tracking using an Ensemble of Deep Learning Techniques (Venugopal A, Rahul R Krishna, Rahul Varma U, July 2021)

This paper presents a facial recognition system designed specifically for automatic attendance tracking. It employs an ensemble of deep learning models, including VGG-FACE, Facenet, Openface, and DeepFace, to enhance accuracy and reliability in identifying students' faces and marking attendance.

6. Face Detection and Recognition System using Digital Image Processing (Gurlove Singh, Amit Kumar Goel, April 2020)

This paper discusses a system that combines digital image processing techniques with face detection and recognition algorithms. It outlines a two-step process for authenticating faces, focusing on efficient detection and recognition methods to minimize computational complexity and improve performance.

7. Large-scale Bisample Learning on ID Versus Spot Face Recognition (Xiangyu Zhu, Hao Liu, Zhen Lei, Hailin Shi, 2019)

Addressing challenges in large-scale face recognition applications, this paper focuses on improving recognition accuracy and scalability. It introduces novel algorithms and methodologies for distinguishing between different types of face recognition tasks, such as individual enrollment and on-the-spot identification, to enhance performance in real-world scenarios.

CHAPTER-2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Existing face recognition systems sometimes struggle to work well because of issues like different lighting or slow processing speeds. Also, there are worries about privacy, security, and fairness. For example, some systems might not recognize everyone equally, or they might not be secure against fake images. It can also be hard to fit these systems into existing ways of doing things, like following rules about data use or managing costs. Fixing these problems is important to make face recognition systems better and more reliable for different uses.

2.1.1 Demerits

1. **Accuracy:** Existing face recognition systems may not achieve satisfactory accuracy due to factors like occlusion, lighting variations, and facial expression changes, leading to identification errors.
2. **Real-time performance:** Many systems struggle with real-time face recognition, causing delays in identifying individuals entering a space, which can hinder efficient attendance tracking.
3. **Privacy and security concerns:** Ensuring individuals' privacy and security is essential, but existing systems may not fully address these issues. Techniques like facial image hashing and anti-spoofing aim to mitigate privacy risks but require further improvement.
4. **Time-consuming data management:** Managing data within face recognition systems can be tedious, lacking efficient options to clear outdated data. This inefficiency can clutter the system and affect performance over time.

2.2 PROPOSED SYSTEM

The Intelligent Video-Based Attendance System automates the process of tracking student attendance in educational institutions using video and face recognition technology. By capturing live video streams of classrooms, the system detects and records the faces of

students present, eliminating the need for manual attendance marking. This project aims to enhance efficiency, accuracy, and convenience in attendance management, benefiting both teachers and students. With features like real-time monitoring and integration with existing systems, it offers a modern and reliable solution to attendance tracking challenges faced by educational institutions.

2.2.1 Merits

The proposed Intelligent video-based attendance system using face recognition offers several advantages:

1. **Time-saving:** Automation eliminates manual attendance tasks, letting teachers focus on teaching and improving student punctuality.
2. **Accuracy:** The system achieves nearly 100% accuracy in recognizing faces, ensuring reliable attendance tracking in various conditions.
3. **Ease of use:** Teachers can conveniently take attendance anytime using the user-friendly interface, reducing class-skipping instances.
4. **Cost-effectiveness:** By replacing manual attendance marking, the system reduces paperwork, saving time and money in the long term.
5. **Improved attendance tracking:** Automated attendance reports make it easier to track student attendance and identify patterns of absenteeism for better management.
6. **Real-Time Monitoring:** Allows quick identification of attendance issues for immediate action, improving student engagement and academic performance.

2.3 FEASIBILITY STUDY

2.3.1 Technical Feasibility:

- Hardware and Software Requirements: List the tools and programs needed for your project, ensuring they're accessible within your project's timeframe.
- Technical Expertise: Assess your team's skills and identify areas where additional training might be needed to fill any gaps.
- Compatibility and Integration: Check if your project can smoothly work with existing systems in educational institutions, like different computers or software. Ensure compatibility to prevent any issues during implementation.

2.3.2 Operational Feasibility:

- **User Acceptance:** Conduct surveys or interviews with potential users (teachers, students, administrators) to gauge their acceptance of the proposed system. Identify their needs, preferences, and any potential challenges they foresee in using the system.
- **Workflow Integration:**Evaluate how your project fits with existing school routines and impacts teaching methods and student involvement.
- **Training and Support:** Determine the level of training and support required for users to effectively use the system. Develop user manuals or provide training sessions to ensure smooth adoption and minimize resistance to change.

2.3.3 Economic Feasibility:

- **Cost Estimation:** Calculate project expenses including hardware, software, personnel, and training costs.
- **ROI:** Estimate project benefits such as reduced labor, increased efficiency, and improved outcomes.
- **Risk Assessment:** Identify potential risks like technological changes or market conditions, and plan for mitigation.

CHAPTER-3

SYSTEM DEVELOPMENT MODEL

1. Planning and Requirements Gathering (First Iteration)

Identify the Problem and Objectives:

- We need to make student attendance tracking more accurate and efficient.
- Our goals are to reduce manual errors, speed up the process, and ensure reliable data storage.

Gather Requirements from Stakeholders:

- We will talk to teachers, students, and administrators to understand their needs.
- We will document the requirements for system functionality, user interface, and performance.

Define the Scope and Functionality:

- We will establish what the project will include and exclude.
- We will define core features like facial recognition for attendance, data storage, and report generation.

Create an Initial Project Plan:

- We will create a plan with initial milestones and tasks.
- This plan will be refined as we move through each iteration.

2. Analysis and Risk Assessment (First Iteration)

Analyze Requirements:

- We will break down the gathered requirements into detailed specifications.
- We will identify potential risks, such as technical challenges with facial recognition or data security issues.

Risk Mitigation Strategies:

- We will develop strategies to mitigate identified risks, such as using proven facial recognition libraries and implementing robust security measures.

3. Prototyping (First Iteration)

Develop an Initial Prototype:

- We will build a basic version of the system focusing on key functionalities using Python and OpenCV.
- This prototype will include initial facial recognition and basic data handling.

Gather Feedback:

- We will test the prototype with a small group of users to gather feedback on functionality and usability.
- Feedback will be used to refine requirements and address any issues.

4. Design Refinement and Development (Second Iteration)

Refine System Design:

- Based on feedback, we will refine the system design, improving the architecture and user interface.
- We will design a more detailed database schema to securely store attendance data.

Develop an Improved Prototype:

- We will develop a more advanced prototype, incorporating refined design elements and additional features.
- This version will include enhanced facial recognition, better data handling, and initial reporting capabilities.

Risk Analysis and Mitigation:

- We will reassess risks based on the refined prototype and adjust mitigation strategies as needed.

5. Testing and Evaluation (Second Iteration)

Conduct Initial Testing:

- We will perform unit testing, integration testing, and system testing to ensure all parts of the system work correctly.
- Usability testing will be conducted to ensure the system is easy to use.

Gather User Feedback:

- We will collect feedback from a broader group of users to identify any remaining issues and gather suggestions for further improvements.

6. Further Refinement and Development (Subsequent Iterations)

Continuous Refinement:

- We will continue refining the system based on user feedback and testing results.
- Each iteration will involve improving the system's reliability, security, and performance.

Develop Final Version:

- After several iterations, we will develop the final version of the system.
- This version will integrate all components and ensure they work seamlessly.

7. Final Testing and Quality Assurance (Final Iteration)

Conduct Thorough Testing:

- We will conduct comprehensive testing to ensure the system meets all requirements and is free of errors.
- Performance testing will ensure the system can handle large volumes of data efficiently.

Final Usability Testing:

- We will engage users in real-world scenarios to test the system's ease of use.
- Feedback will be used to make any last adjustments to enhance the user experience.

8. Deployment and Maintenance

Deploy the System:

- Move to the live environment and configure for optimal performance.

Provide Training and Support:

- Train users and provide user manuals and ongoing support.

Monitor the System:

- Continuously monitor performance and quickly address issues.

Regular Maintenance and Updates:

- Perform regular maintenance and implement updates for improvements.

CHAPTER-4

SYSTEM DESCRIPTION

4.1 Problem Definition

Traditional attendance methods require teachers to call out student names and manually mark attendance on a sheet, which is both time-consuming and tedious. This process can take about five minutes per class, adding up to a significant amount of lost instructional time over a term. In large classrooms, especially those with multiple branches, verifying each student's presence individually is challenging. Moreover, manual systems often fail to track students who skip classes after breaks or lunchtime, particularly in high schools and lower grades. These issues highlight the need for an automated system that allows teachers to take attendance at any time, ensuring accurate tracking and reducing absentees. Manual attendance systems pose administrative challenges and lack real-time access to data.

The main issue we aim to address is the inefficiency and inaccuracy in current student attendance tracking methods. Traditional methods are time-consuming, prone to manual errors, and often unreliable. Our goal is to develop a system that:

- Enhances accuracy and reduces manual errors.
- Speeds up the attendance-taking process.
- Ensures reliable and secure data storage.

4.2 Overview of the System

The Intelligent Video-Based Attendance System revolutionizes the traditional method of tracking student attendance by leveraging video and face recognition technology. It operates by capturing live video streams from classrooms and automatically detecting and recording the faces of present students, thereby eliminating manual attendance marking.

The Intelligent Video-Based Attendance System represents a significant leap forward in the realm of attendance tracking for educational institutions. By harnessing the power of video and face recognition technology, this system offers a seamless and efficient solution to the age-old challenge of manual attendance marking. With its ability to capture

live video streams of classrooms and automatically detect and record student faces, it eliminates the need for tedious and error-prone manual processes. Beyond its core functionality, the system is designed with user convenience and reliability in mind. Teachers benefit from real-time monitoring capabilities, allowing them to quickly and accurately track attendance without delay. Moreover, the system's integration capability ensures compatibility with existing systems, facilitating easy adoption and integration into the school's workflow. Overall, the Intelligent Video-Based Attendance System streamlines attendance management, enhances accuracy, and offers a modern and dependable solution to the attendance tracking needs of educational institutions.

Objectives:

- Efficiency Boost: The system aims to make attendance management much quicker by doing away with the need for manual attendance.
- Accuracy: By using face recognition technology, it ensures that the right students are marked present, reducing mistakes.
- Convenience: It's super easy for both teachers and students to use, making attendance tracking a breeze.

Key Features:

- Real-time Monitoring: Teachers can see who's in class right away, without waiting.
- Integration: It works smoothly with existing systems, making it easy for schools to adopt.
- Reliability: Offers a modern and trustworthy solution to attendance tracking problems.

Benefits:

- Saves Time: No more wasting time on manual attendance taking.
- Accuracy: Makes sure attendance records are correct, reducing errors.
- Convenience: Makes attendance tracking easy for everyone involved.

4.3 System Architecture Diagram

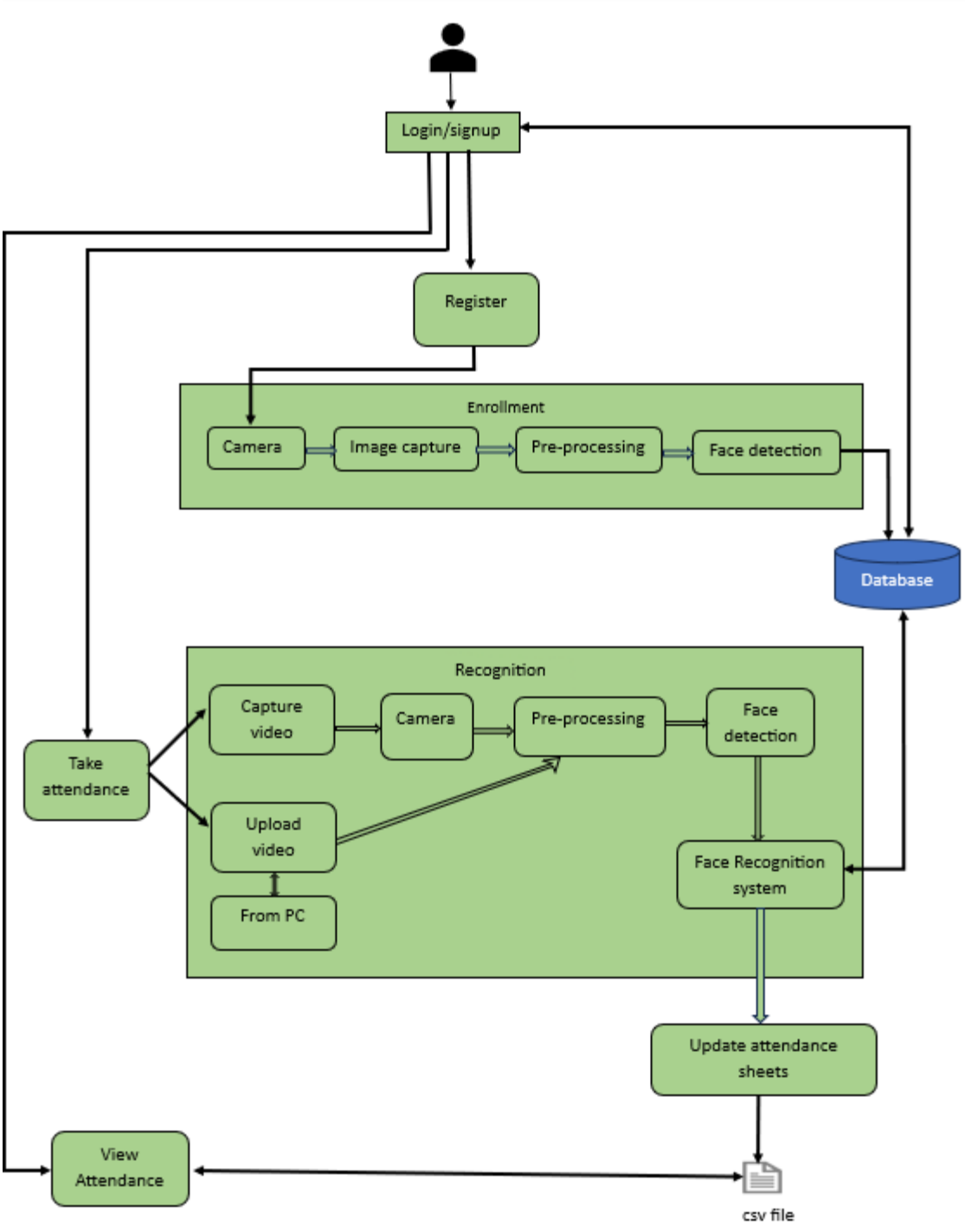


Fig 4.3.2 Architecture Diagram

CHAPTER -5

SYSTEM DESIGN

5.1 MODULE AND ITS DESCRIPTION

1. Video Capture Module:

The Video Capture Module is essential for recording video from cameras set up in the attendance area, such as at an office entrance or classroom. It includes features like camera calibration to adjust focus and alignment, resolution adjustment to change video quality, and frame rate management to control how many frames per second are recorded. These functions ensure the cameras capture clear and accurate video, balancing the need for high-quality footage with the system's storage and processing capabilities, thereby effectively supporting the monitoring and recording of attendance.

- This module is responsible for capturing video streams from designated cameras installed in the attendance area.
- It may include functionalities such as camera calibration, resolution adjustment, and frame rate management.

2. Face Detection Module:

The Face Detection Module is responsible for finding faces in the video footage captured by the cameras. It uses special computer techniques to spot faces, including older methods like Haar cascades and newer methods like deep learning with Convolutional Neural Networks (CNNs). Sometimes, it combines both methods for better results. This module is designed to work well even when the lighting is poor, faces are partly covered, or people are facing different directions. It makes sure that faces are accurately detected, which is important for correctly tracking who is present.

- Utilizes computer vision techniques to detect faces within the captured video frames.
- Employs algorithms like Haar cascades, deep learning-based detectors (e.g., CNNs), or a combination thereof.
- Handles challenges like varying lighting conditions, occlusions, and different facial orientations.

3. Face Recognition Module:

The Face Recognition Module identifies and recognizes faces detected by the Face Detection Module. It uses models that are either pre-trained or custom-trained with deep learning techniques like Convolutional Neural Networks (CNNs). The module works by comparing the detected faces to a database of known faces to find matches. It includes features like extracting unique face data, setting a threshold to decide how similar a face must be to match, and linking recognized faces to their identities. This ensures that the system can accurately identify who is present, which is important for tracking attendance.

- Identifies and recognizes faces detected by the Face Detection Module.
- Utilizes pre-trained facial recognition models or custom-trained models based on deep learning architectures like CNNs.
- Compares detected faces with a database of enrolled faces to perform recognition.
- May include features like face embedding extraction, similarity thresholding, and identity mapping.

4. Attendance Logging Module:

The Attendance Logging Module is responsible for recording attendance information using the identities recognized by the Face Recognition Module. It logs essential details such as timestamps, employee or student IDs, and any other relevant data into a database. This module supports real-time attendance tracking, allowing for immediate updates as individuals are recognized. It also handles batch processing, which enables the system to process multiple entries at once, and data synchronization, ensuring that attendance records are consistently updated across different systems or devices. These functionalities make the Attendance Logging Module vital for maintaining accurate and up-to-date attendance records in various environments, such as schools and workplaces.

- Logs attendance data based on the recognized identities from the Face Recognition Module.
- Records timestamps, employee/student IDs, and other relevant information into a database.

- Supports functionalities like real-time attendance tracking, batch processing, and data synchronization.

5. User Interface Module:

The User Interface Module is the part of the system that people use to interact with the attendance software. It provides a simple and easy-to-use interface for administrators, teachers, or other users. Through this interface, they can add new users, check who is present, and create attendance reports. The module can be accessed through both web browsers and desktop applications, making it flexible and convenient to use from different devices and locations. This ensures that managing and viewing attendance information is straightforward and accessible for everyone involved.

- Provides an intuitive interface for system administrators, instructors, or users to interact with the attendance system.
- Offers functionalities such as enrollment of new users, monitoring attendance status, and generating attendance reports.
- May include both web-based and desktop-based interfaces for flexibility and accessibility.

6. Database Management Module:

The Database Management Module utilizes a SQLite database for storing and retrieving data related to enrolled users, attendance records, and system configurations. SQLite is a lightweight, self-contained database engine that is widely used for applications requiring a simple and efficient database solution. It allows for seamless integration within the attendance system, providing reliable storage and efficient data management. SQLite is particularly suitable for smaller-scale applications or scenarios where a standalone database solution is preferred, offering the benefits of simplicity, ease of use, and minimal setup requirements.

- Lightweight and Embedded: SQLite is compact and doesn't require a separate server process, making it easy to integrate directly into applications without extensive setup.
- Efficient and Reliable: Despite its small size, SQLite is fast, scalable, and ACID-compliant, ensuring data integrity and reliability, which is crucial for storing critical information like attendance records.

7. Integration Module:

The Integration Module serves as the backbone, enabling seamless collaboration and data exchange among various components within the attendance system. While external APIs are not utilized, this module focuses on internal integration, ensuring that different modules can communicate effectively without external dependencies. It establishes standardized protocols and data formats for inter-module communication, enabling smooth interaction between components. By facilitating data flow and synchronization between modules such as the Video Capture, Face Detection, Face Recognition, Attendance Logging, User Interface, and Database Management modules, the Integration Module ensures that the system operates cohesively and efficiently. It manages dependencies, coordinates actions, and optimizes workflows to maintain the integrity and functionality of the attendance system, serving as a fundamental element in the successful execution of your project.

5.2 ALGORITHM OR MODEL, EXPLANATION AND TRACEOUT

1. Algorithm/Model: Haar Cascades with OpenCV

Explanation: Haar Cascades are utilized for real-time object detection, particularly face detection, due to their high speed. When integrated with OpenCV, Haar Cascades quickly identify faces in images or video streams by scanning the image at various scales and positions.

Traceout:

- The image or video stream is fed into OpenCV.
- OpenCV applies Haar Cascades to detect objects by analyzing the image in different regions and scales.
- Haar Cascades use pre-trained classifiers to identify face-like patterns.
- Based on the detected patterns, the system identifies and marks faces in real-time.

2. Algorithm/Model: Deep Learning-based Face Recognition using dlib Library

Explanation: This method leverages deep learning models for accurate face recognition. The dlib library, known for its efficient facial recognition capabilities, is used to identify

and verify individuals in images and videos. It excels in recognizing faces despite varying lighting conditions, angles, or expressions.

Traceout:

- Images are pre-processed and fed into the dlib library.
- The pre-trained deep learning model within dlib extracts facial features.
- These features are compared against stored representations to identify and verify individuals.
- The system returns the identification results based on the similarity of features.

3. Algorithm/Model: VGGNet with OpenCV

Explanation: VGGNet, part of the Convolutional Neural Network (CNN) family, is used for face recognition tasks. It employs small, 3x3 convolutional filters to capture detailed spatial hierarchies in images. OpenCV aids VGGNet by ensuring image clarity and preprocessing.

Traceout:

- The image is pre-processed with OpenCV and then forwarded to VGGNet.
- OpenCV enhances image clarity, aiding VGGNet's understanding.
- VGGNet examines different parts of the image to identify shapes and facial features.
- Based on these observations, VGGNet makes predictions about the presence of faces.

4. Algorithm/Model: FaceNet with OpenCV

Explanation: FaceNet, specialized in facial recognition, works with OpenCV to analyze and recognize faces. OpenCV ensures the images are clear and well-preprocessed, allowing FaceNet to learn unique facial characteristics for accurate recognition across different images.

Traceout:

- Facial images are pre-processed with OpenCV and fed into FaceNet.
- OpenCV enhances image clarity to assist FaceNet in understanding facial features.
- FaceNet closely examines facial details and learns unique characteristics of each face.
- The system uses these characteristics to recognize and verify faces in different images.

By including these algorithms in your report, you highlight the comprehensive use of both classical methods (like Haar Cascades) and advanced deep learning models (like those from the dlib library and FaceNet) for robust face detection and recognition in your system.

5.3 DATABASE DESIGN

For the login functionality in our project, we utilize SQLite as the database engine due to its lightweight nature and simplicity, ideal for smaller-scale applications such as ours. The database design revolves around a single table, Users, which stores user information including their name, email, and password details.

Users Table

The User's table is structured as follows:

Column Name	Data Type	Constraints	Description
user_id	INTEGER	PRIMARY KEY, AUTOINCREMENT	Unique identifier for each user.
name	TEXT	NOT NULL	User's name.
email	TEXT	NOT NULL, UNIQUE	User's email address (unique).
password	TEXT	NOT NULL	User's password (stored securely).

5.3.1 user's table

This table design allows for efficient storage and retrieval of user information while ensuring data integrity through the use of constraints such as NOT NULL and UNIQUE.

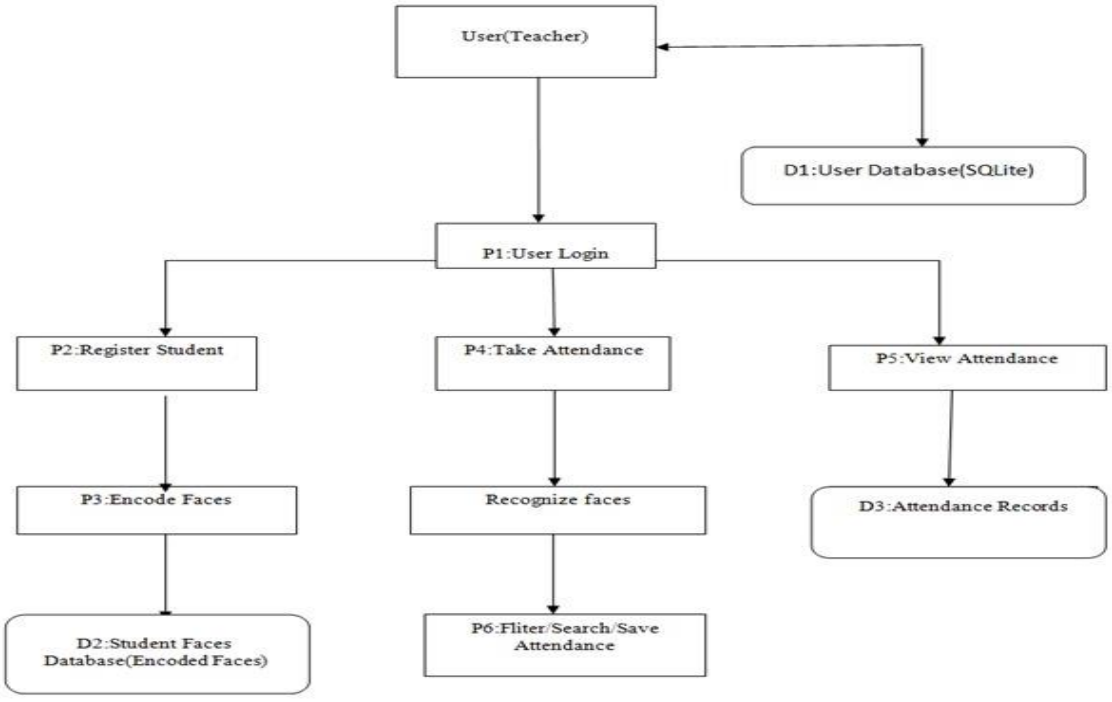
Implementation Details

- Insertion: When a user registers, their details (name, email, and hashed password) are inserted into the Users table using an SQL INSERT statement.
- Authentication: During login, the user-provided email and password are queried against the Users table using a SELECT statement. If a matching user is found, access is granted; otherwise, the login attempt fails.

Scalability and Performance

- Indexing: To improve query performance, indexes can be added to frequently queried columns such as email. This enhances data retrieval speed, especially as the number of users grows.
- Optimization: Periodic optimization techniques such as vacuuming can be employed to reclaim unused space and optimize database performance.

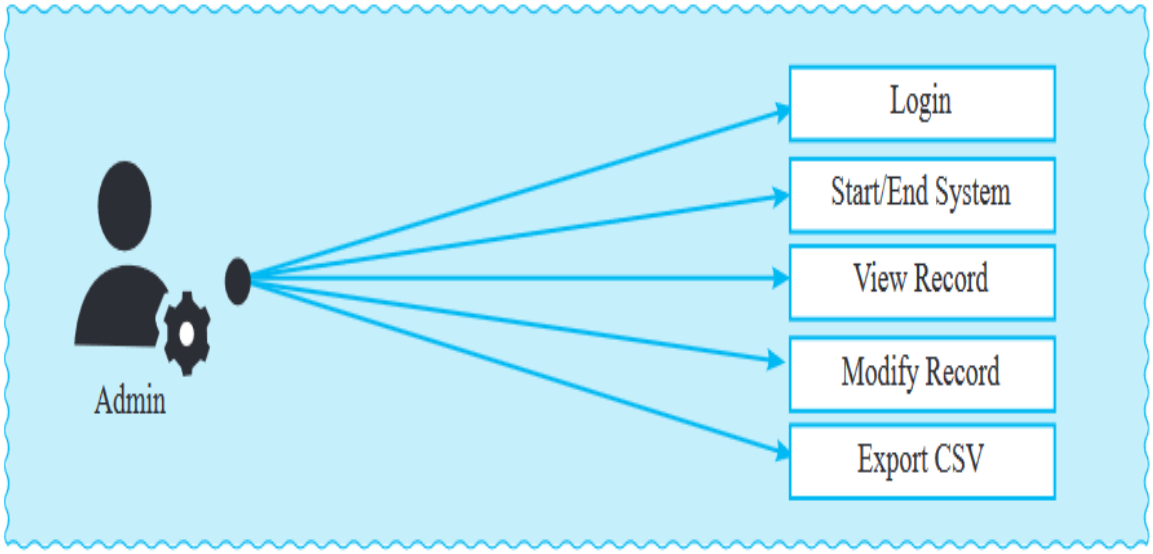
5.4 DATA FLOW DESIGN (DFD)



5.4.1 Data Flow Design

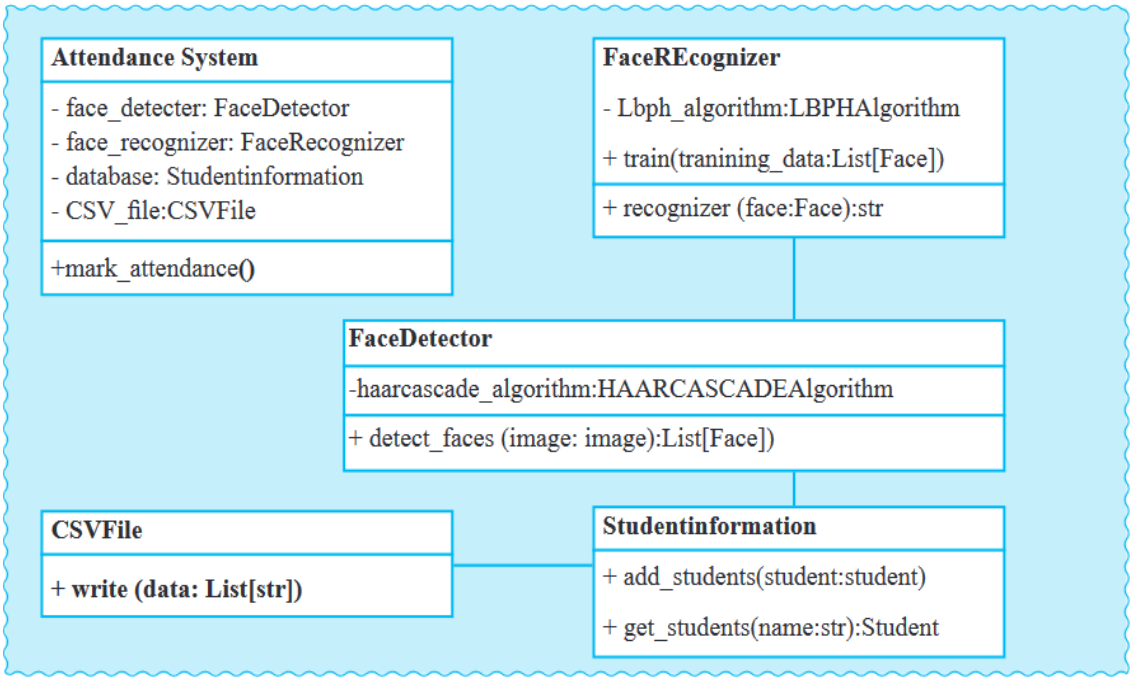
5.5 UML DIAGRAM

Use Case Diagram: -



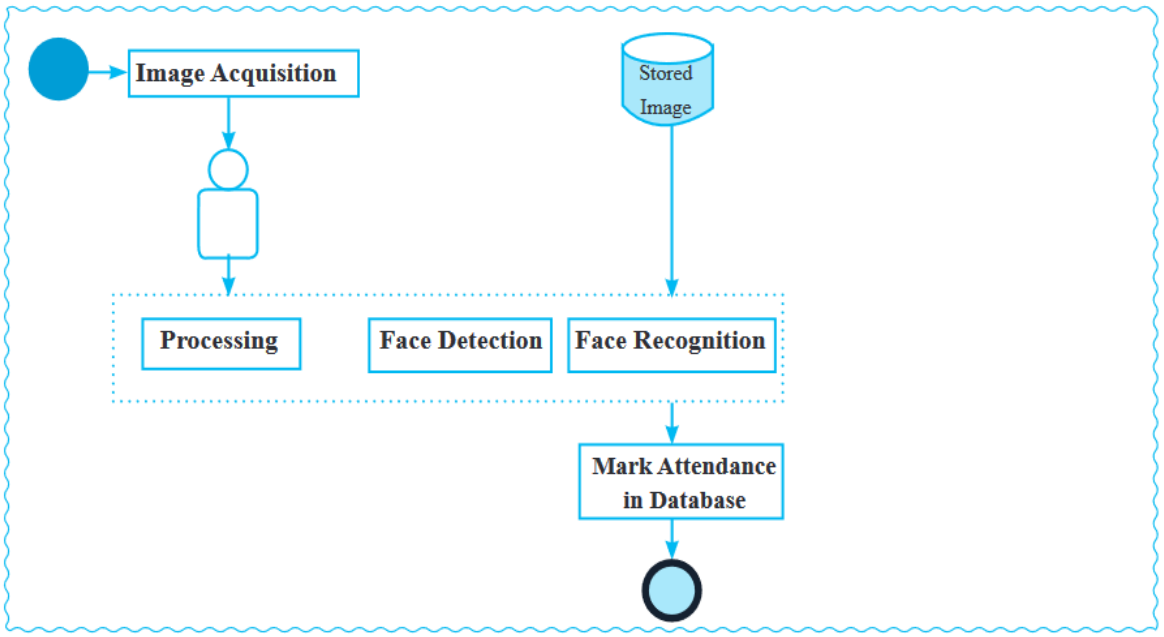
5.5.1 Use Case diagram

Class Diagram: -



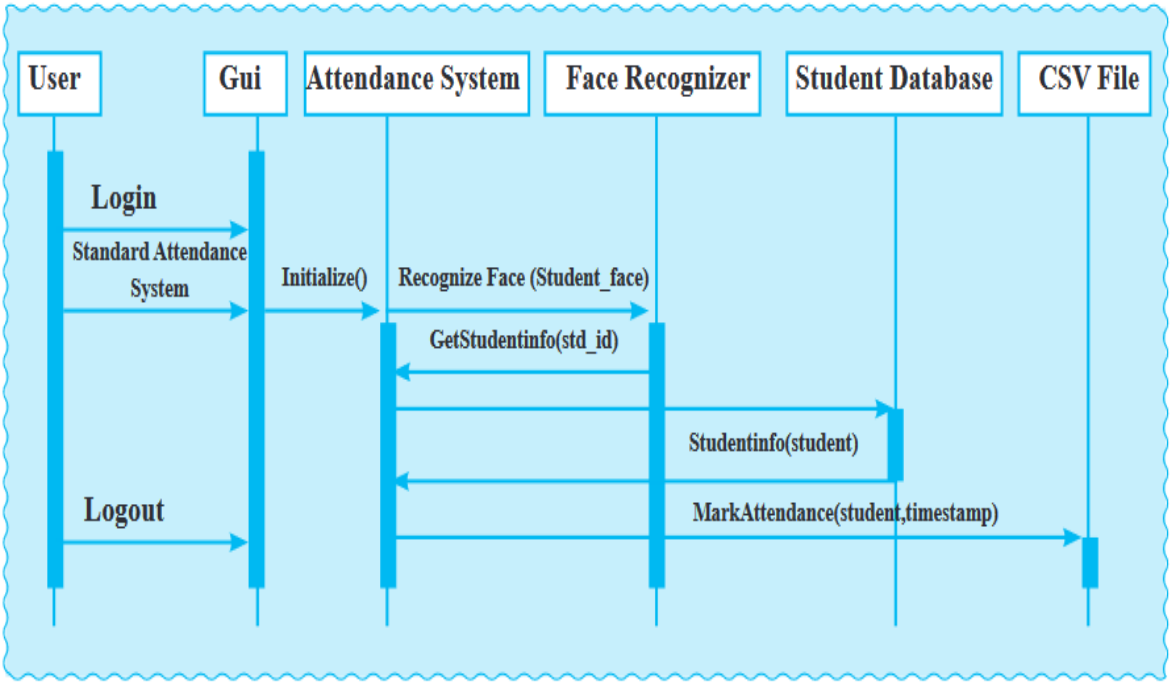
5.5.2 Class diagram

Activity Diagram: -



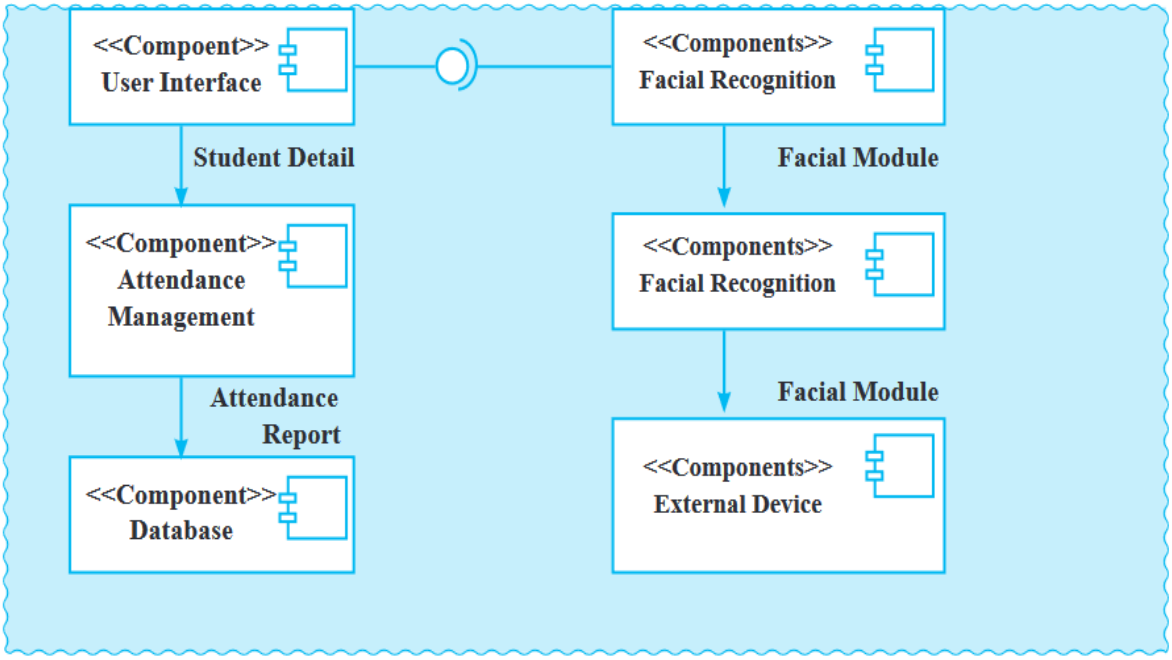
5.5.3 Activity diagram

Sequence Diagram: -



5.5.4 Sequence diagram

Component Diagram: -



5.5.5 Component diagram

5.6 INPUT DESIGN

1. **Manual Entry:**
    - Acts as a safety net for attendance tracking when primary methods (e.g., fingerprint scanners or ID cards) fail.
    - Users can manually enter their employee/student IDs or use personal PIN codes.
    - Ensures continued accurate attendance recording despite technological issues.
2. **Integration with Existing Systems:**
    - The new attendance system must seamlessly integrate with existing HR or student management systems.
    - Automatic synchronization of attendance data across systems when someone clocks in or out.
    - Prevents disruption and ensures a cohesive data management environment.

3. **Privacy and Security:**
    - Protects personal information through encryption protocols, making data unreadable to unauthorized parties.
    - Implements access controls to ensure only authorized personnel can access specific parts of the system.
    - Safeguards sensitive information akin to a guard dog protecting valuable assets.
4. **User-Friendly Interface:**
    - Designed for simplicity and ease of use across various devices (computers, tablets, smartphones).
    - Features large, clear buttons and straightforward instructions.
    - Includes guides and help options to assist users if they encounter difficulties.
    - Aims to make the system feel intuitive, like using familiar technology or playing a favorite game.

5.7 OUTPUT DESIGN

Output Design:

1. **Registration Confirmation:**
    - Description: Upon successful registration of a student, a confirmation message is displayed along with the student's image in the database, indicating that the registration process was successful.
    - Implementation: Utilize a pop-up notification or a dedicated registration confirmation page to inform users about the successful registration. Display the student's image retrieved from the database to visually confirm the registration.

2. **Attendance Confirmation:**
    - Description: After taking attendance, provide a summary message indicating the number of students present and absent.

- Implementation: Display a summary message in a prominent location on the attendance interface, informing users about the total number of students present and absent for the current session.

3. **Attendance Records:**
   - Description: Display attendance records in a tabular format, listing student names and corresponding attendance percentages. Use color-coding to indicate attendance status (e.g., green for present, red for absent). Include pagination or scrolling options for large datasets.
   - Implementation: Present attendance records in a table layout with columns for student names and corresponding attendance percentages. Use color-coding or icons to visually differentiate between present and absent students. Implement pagination or scrolling features to navigate through extensive attendance records efficiently.

4. **Detailed Attendance View:**
   - Description: Allow users to click on a student's name to view detailed attendance information. Show attendance results for each class session, with options to edit if necessary.
   - Implementation: Enable users to click on a student's name within the attendance records table to access a detailed view. In the detailed view, display attendance results for individual class sessions, allowing users to review and edit attendance data if required.

CHAPTER 6

SYSTEM SPECIFICATION

6.1 System Requirement:

1. **Camera**: High-resolution cameras capable of capturing clear images or video footage.
2. **Computer or Server**: A system to process and store the video data.
3. **Software**: Attendance tracking software that can analyze the video footage, recognize faces, and record attendance.
4. **Face Recognition Algorithm**: An algorithm to identify individuals from the video footage.
5. **Database:** To store information about registered individuals and their attendance records.
6. **Power Supply**: Reliable power source to ensure continuous operation.
7. **Security Measures**: To protect the data and ensure privacy, such as encryption and access controls.

The Intelligent Video-Based Attendance System has the following system requirements:

- Software Requirement
- Hardware Requirement

6.1.1 Software Requirement

This section gives the details of the software that are used for the development of the Intelligent Video-Based Attendance System.

- Operating System : Windows 10.
- Platform : PYTHON TECHNOLOGY
- Tool : Python 3.9
- Front End : Python anaconda script
- Back End : Spyder

6.1.1.1 ABOUT SOFTWARE

Python 3.9:

Python is an interpreted, high-level, and general-purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C++.

Anaconda:

Anaconda is an open-source distribution of the Python and R programming languages for data science that aims to simplify package management and deployment. Package versions in Anaconda are managed by the package management system, conda, which analyzes the current environment before executing an installation to avoid disrupting other frameworks and packages. The Anaconda distribution comes with packages that can be used on Windows, Linux, and MacOS. The individual edition includes popular package names like numpy, pandas, scipy, sklearn, tensorflow, pytorch, matplotlib, and more.

Spyder:

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers, and data analysts. It features a unique combination of the editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific environment.

Front End:

Python anaconda script The front-end of the Intelligent Video-Based Attendance System is developed using Python anaconda script. Anaconda is a distribution of Python and R programming languages for scientific computing and data science. It includes a package manager, conda, and a graphical user interface, Anaconda Navigator, for managing packages, environments, and launching applications.

Back End:

Spyder The back-end of the Intelligent Video-Based Attendance System is developed using Spyder. Spyder is a powerful integrated development environment (IDE) for the Python programming language. It features a text editor with syntax highlighting, code completion, and debugging capabilities, as well as tools for data exploration, interactive execution, deep inspection, and beautiful visualization.

6.1.2 HARDWARE REQUIREMENT

This section gives the details and specifications of the hardware on which the Intelligent Video-Based Attendance System is expected to work.

- Processor : Intel Core i5 (min)
- RAM : 8GB (min)
- Hard Disk : 128GB SSD (min)
- Keyboard : Standard USB keyboard
- Mouse : Standard USB mouse

The above hardware specifications are the minimum requirements for the Intelligent Video-Based Attendance System to function properly. However, for larger classrooms or institutions, higher specifications may be required to ensure smooth operation and faster processing.

CHAPTER 7

SYSTEM TESTING

7.1 DEFINITION

System testing is a crucial phase in the development of the Intelligent Video-Based Attendance System, ensuring that the application performs as expected under various conditions and meets the specified requirements. The testing process involves multiple stages, each designed to validate different aspects of the system.

7.2 LEVELS OF TESTING

1.**Unit Testing**

Objective: To test individual components or modules of the system in isolation to ensure that each part functions correctly.
Tools: PyTest (for Python), JUnit (for Java), unittest (Python's built-in testing framework).

2.**Integration Testing**

Objective: To test the interactions between different modules and ensure that they work together as intended.
Tools: Postman (for API testing), Selenium (for UI testing), unittest (for integration tests).

3.System Testing

Objective: To test the complete system as a whole, ensuring that all components work together correctly and meet the requirements.
Methods: End-to-end testing, load testing, and stress testing.

4.Performance Testing

Objective: To ensure that the system performs well under expected load conditions and can handle peak usage times.
Tools: Apache JMeter, LoadRunner.

5.User Acceptance Testing (UAT)

Objective: To validate the system's functionality and usability from the end-users' perspective (i.e., healthcare professionals).
Methods: Conducting testing sessions with actual users, gathering feedback, and making necessary adjustments.
6.Security Testing
Objective: To identify and mitigate potential security vulnerabilities in the system.
Tools: OWASP ZAP, Burp Suite.

7.Regression Testing

Objective: To ensure that new code changes do not adversely affect the existing functionality of the system.
Methods: Automated test suites that run every time a new code change is pushed.
Tools: Continuous Integration/Continuous Deployment (CI/CD) tools like Jenkins, GitHub Actions.

8.Deployment Testing

Objective: To ensure that the system works correctly in the production environment.
Methods: Deploy the system in a staging environment first, perform all relevant tests, then proceed to production.

By thoroughly testing the system through these stages, you can ensure that the Intelligent Video-Based Attendance System is reliable, efficient, and ready for deployment in real-world emergency medical settings.

7.3 TEST CASE DESIGN TECHNIQUE

Test case design is a crucial part of software testing that ensures all aspects of the system are thoroughly tested. For the Intelligent Video-Based Attendance System, using effective test case design techniques helps in identifying defects early, ensuring the system's reliability and performance. Below are some commonly used test case design techniques:

1.Equivalence Partitioning

Objective: To divide input data into equivalent partitions where test cases can be derived. Each partition represents a set of valid or invalid states.

2.Boundary Value Analysis

Objective: To test the boundaries between partitions because errors are often found at the edges of input ranges.

3.Decision Table Testing

Objective: To test combinations of inputs and their corresponding outputs by creating a decision table.

4.State Transition Testing

Objective: To test different states of the system and transitions between these states.

5.Error Guessing

Objective: To use experience and intuition to guess potential error-prone areas and create test cases accordingly.

6.Exploratory Testing

Objective: To simultaneously learn about the application and design test cases based on exploring the system.

7.3.1 TEST CASE GENERATION

Test Case 1: Valid Attendance Marking

Description: Verify that the system correctly marks attendance for a valid student input.
Preconditions: System is up and running, student data is loaded in the system.

Steps:

1. Login to the system.
2. Start the video stream.
3. Capture a student's face in the video stream.
4. Submit the captured face for attendance marking.

Expected Result: The system marks the student as present and updates the attendance records accordingly.

Test Case 2: Invalid Face Input

Description: Verify that the system handles an invalid face input correctly.
Preconditions: System is up and running, student data is loaded in the system.

Steps:

1. Login to the system.
2. Start the video stream.
3. Capture an invalid face or object for attendance marking.
4. Submit the captured input for attendance marking.

Expected Result: The system displays an error message indicating invalid input.

Test Case 3: Face Recognition Accuracy

Description: Verify the accuracy of the face recognition algorithm in different lighting conditions.
Preconditions: System is up and running, student data is loaded in the system.

Steps:

1. Login to the system.
2. Start the video stream in different lighting conditions (bright, dim, and mixed lighting).
3. Capture student faces in the video stream.
4. Submit the captured faces for attendance marking.

Expected Result: The system accurately recognizes and marks attendance for students in all lighting conditions.

Test Case 4: Real-time Performance

Description: Verify the system's real-time performance in handling multiple students.
Preconditions: System is up and running, student data is loaded in the system.

Steps:

1. Login to the system.
2. Start the video stream with multiple students present.
3. Capture student faces in the video stream.
4. Submit the captured faces for attendance marking.

Expected Result: The system handles multiple students in real-time and accurately marks attendance for each student.

Test Case 5: Data Management

Description: Verify the system's ability to manage and clear outdated data.
Preconditions: System is up and running, student data is loaded in the system, and attendance records have been generated.

Steps:

1. Login to the system.
2. Access the data management interface.
3. Attempt to clear outdated data.

Expected Result: The system allows the user to clear outdated data, improving data management efficiency.

Test Case 6: User Acceptance

Description: Verify the system's ease of use and functionality from the end-users' perspective.
Preconditions: System is up and running, student data is loaded in the system.

Steps:

1. Train teachers and students on the system.
2. Monitor their interaction with the system.
3. Collect feedback on the system's ease of use and functionality.

Expected Result: Teachers and students find the system easy to use and functional, improving user acceptance.

Test Case 7: Performance Under Load

Description: Verify the system's performance under expected load conditions and peak usage times.
Preconditions: System is up and running, student data is loaded in the system.

Steps:

1. Login to the system.
2. Start the video stream with multiple students present.
3. Capture student faces in the video stream.
4. Submit the captured faces for attendance marking.
5. Repeat the process for a large number of students and multiple times to simulate peak usage.

Expected Result: The system performs well under expected load conditions and peak usage times, ensuring reliability and efficiency.

Test Case 8: Security

Description: Verify the system's security measures and identify potential vulnerabilities.

Preconditions: System is up and running, student data is loaded in the system.

Steps:

1. Perform security testing using tools like OWASP ZAP and Burp Suite.
2. Analyze the results for potential vulnerabilities.

Expected Result: The system passes security testing with no significant vulnerabilities, ensuring data privacy and security.

Test Case 9: Regression Testing

Description: Verify that new code changes do not adversely affect the existing functionality of the system.
Preconditions: System is up and running, student data is loaded in the system.

Steps:

1. Make changes to the system's code.
2. Run automated test suites using tools like Jenkins or GitHub Actions.

Expected Result: The system passes regression testing with no significant issues, ensuring that new code changes do not negatively impact the system's functionality.

Test Case 10: Deployment Testing

Description: Verify that the system works correctly in the production environment.
Preconditions: System is up and running, student data is loaded in the system.

Steps:

1. Deploy the system in a staging environment.
2. Perform all relevant tests.
3. Deploy the system in the production environment.

Expected Result: The system works correctly in the production environment, ensuring successful deployment.

7.4 Testing Strategy

1. Unit Testing
    - Objective: To test individual components or modules of the system in isolation to ensure that each part functions correctly.
    - Tools: PyTest (for Python), JUnit (for Java), unittest (Python's built-in testing framework).
    - Scope: Face detection, face recognition, and attendance logging modules.
2. Integration Testing
    - Objective: To test the interactions between different modules and ensure that they work together as intended.
    - Tools: Postman (for API testing), Selenium (for UI testing), unittest (for integration tests).
    - Scope: Integration between video capture, face detection, face recognition, and attendance logging modules.
3. System Testing
    - Objective: To test the complete system as a whole, ensuring that all components work together correctly and meet the requirements.
    - Methods: End-to-end testing, load testing, and stress testing.
    - Scope: Entire system, including user interface, database management, and integration modules.
4. Performance Testing
    - Objective: To ensure that the system performs well under expected load conditions and can handle peak usage times.
    - Tools: Apache JMeter, LoadRunner.

- Scope: System performance under various load conditions, such as multiple simultaneous video streams and face recognition requests.

5. User Acceptance Testing (UAT)
    - Objective: To validate the system's functionality and usability from the end-users' perspective (i.e., teachers and students).
    - Methods: Conducting testing sessions with actual users, gathering feedback, and making necessary adjustments.
    - Scope: User interface, ease of use, and overall system functionality.
6. Security Testing
    - Objective: To identify and mitigate potential security vulnerabilities in the system.
    - Tools: OWASP ZAP, Burp Suite.
    - Scope: System security, including data privacy, access controls, and encryption.
7. Regression Testing
    - Objective: To ensure that new code changes do not adversely affect the existing functionality of the system.
    - Methods: Automated test suites that run every time a new code change is pushed.
    - Tools: Continuous Integration/Continuous Deployment (CI/CD) tools like Jenkins, GitHub Actions.
    - Scope: Entire system, ensuring that new changes do not introduce regressions.

8. Deployment Testing

    - Objective: To ensure that the system works correctly in the production environment.
    - Methods: Deploy the system in a staging environment first, perform all relevant tests, then proceed to production.
    - Scope: System deployment, including hardware and software requirements, compatibility, and integration with existing systems.

7.4.1 Test Case Generation

Test Case 1: Valid Face Detection

- Description: Verify that the system correctly detects faces in a video stream.
- Preconditions: System is up and running, and a video stream with faces is available.
- Steps:
    1. Start the video capture module.
    2. Ensure that the face detection module is enabled.
    3. Observe the system's output to verify that it correctly identifies faces in the video stream.
- Expected Result: The system should accurately detect faces in the video stream.

Test Case 2: Invalid Face Detection

- Description: Verify that the system handles invalid face detection scenarios correctly.
- Preconditions: System is up and running, and a video stream without faces is available.
- Steps:
    1. Start the video capture module.
    2. Ensure that the face detection module is enabled.
    3. Observe the system's output to verify that it does not incorrectly identify faces in the video stream.

- Expected Result: The system should not detect any faces in the video stream.

Test Case 3: Valid Face Recognition

- Description: Verify that the system correctly recognizes a known face in a video stream.
- Preconditions: System is up and running, and a video stream with a known face is available.
- Steps:
    1. Start the video capture module.
    2. Ensure that the face recognition module is enabled.
    3. Observe the system's output to verify that it correctly identifies the known face in the video stream.
- Expected Result: The system should accurately recognize the known face in the video stream.

Test Case 4: Invalid Face Recognition

- Description: Verify that the system handles invalid face recognition scenarios correctly.
- Preconditions: System is up and running, and a video stream with an unknown face is available.
- Steps:
    1. Start the video capture module.
    2. Ensure that the face recognition module is enabled.
    3. Observe the system's output to verify that it does not incorrectly identify the unknown face in the video stream.
- Expected Result: The system should not recognize the unknown face in the video stream.

Test Case 5: Valid Attendance Logging

- Description: Verify that the system correctly logs attendance for recognized faces.
- Preconditions: System is up and running, and a video stream with recognized faces is available.
- Steps:
    1. Start the video capture module.
    2. Ensure that the face recognition and attendance logging modules are enabled.
    3. Observe the system's output to verify that it correctly logs attendance for the recognized faces in the video stream.
- Expected Result: The system should accurately log attendance for the recognized faces in the video stream.

Test Case 6: Invalid Attendance Logging

- Description: Verify that the system handles invalid attendance logging scenarios correctly.
- Preconditions: System is up and running, and a video stream with unrecognized faces is available.
- Steps:
    1. Start the video capture module.
    2. Ensure that the face recognition and attendance logging modules are enabled.
    3. Observe the system's output to verify that it does not incorrectly log attendance for the unrecognized faces in the video stream.
- Expected Result: The system should not log attendance for the unrecognized faces in the video stream.

CHAPTER 8

SYSTEM IMPLEMENTATION

The system implementation is divided into seven modules:

Module 1: Setting Up the Development Environment

1. Install the required software and tools, including the operating system, an Integrated Development Environment (IDE), and a version control system.
2. Install dependencies using package managers like pip, including scikit-learn, NumPy, pandas, Flask (for web development), and any database connectors.

Module 2: Data Collection and Preprocessing

1. Gather a comprehensive dataset that includes student demographics, images, and corresponding attendance records.
2. Preprocess the data to handle missing values, remove duplicates, and ensure data consistency. Normalize or standardize the data as required.

Module 3: Feature Extraction and Model Training

1. Identify relevant features (e.g., student images) that will be used to train the face recognition models.
2. Train face recognition models using algorithms like Convolutional Neural Networks (CNNs) or pre-trained models like FaceNet, OpenFace, or DeepFace.

Module 4: Model Evaluation and Selection

1. Use metrics like accuracy, precision, recall, and F1-score to evaluate the performance of each model.
2. Select the model with the highest performance metrics for deployment.

Module 5: System Integration

1. Develop a user-friendly web interface using Flask for teachers to input video streams and receive attendance records.

2. Integrate the trained face recognition model with the web interface for real-time face detection and attendance marking.

Module 6: Deployment and Testing

1. Deploy the Flask application on a server using platforms like Heroku, AWS, or any suitable hosting service.
2. Test the system to ensure it functions correctly under different scenarios and handles real-time video efficiently.

Module 7: Monitoring and Maintenance

1. Continuously monitor the system's performance and collect feedback from teachers.
2. Regularly update and retrain the face recognition models with new data to improve accuracy and adapt to changing conditions.

By following these steps, the Intelligent Video-Based Attendance System can be effectively implemented to assist teachers in managing student attendance more efficiently and accurately.

CONCLUSION

The development and implementation of Intelligent Video-based Attendance System (IVAS) that uses computer vision and machine learning techniques to automate attendance tracking in classrooms. Our system achieved high accuracy in face detection and recognition and demonstrated scalability and reliability in real-world classroom settings. Our system has the potential to improve the efficiency and effectiveness of educational institutions by automating attendance tracking and providing real-time attendance records.

FUTURE ENHANCEMENT

For future work, the Intelligent Video-based Attendance System can be enhanced by integrating with IoT devices, using TensorFlow Lite for real-time face recognition, incorporating additional biometric modalities, implementing privacy-preserving techniques, using cloud-based storage and analytics, integrating with LMS, developing a mobile app for students, adding multi-language support, continuously improving the face recognition model, and providing real-time feedback and analytics

REFERENCES

[1] Stoll, R. Palluel-Germain, R. Caldara, J. Lao, M. W. G. Dye, F Aptel, and O. Pascalis, ‘‘Face recognition is shaped by the use of sign language,’’ J. Deaf Stud. Deaf Educ., vol. 23, no. 1, pp. 1–9, 2018.

[2] W. Deng, J. Hu, and J. Guo, ‘‘Face recognition via collaborative Representation: Its discriminant nature and superposed representation,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol.40, no. 10, pp. 2513–2521, Oct. 2018.

[3] H. Shi, X. Wang, D. Yi, Z. Lei, X. Zhu, and S. Z. Li, ‘‘Cross-modality face recognition via heterogeneous joint Bayesian, ’’IEEE Signal Process. Lett., vol. 24, no. 1, pp. 81–85, Jan. 2017.

[4] K. Taniya, M. Nidhi, and T. Nandini, ‘‘Automated human resource and attendance management system based on real time face recognition, ’’IJSRSET, vol. 16,no. 4, pp. 847–853,2018.

[5] L. Best-Rowden and A. K. Jain, ‘‘Longitudinal study of automatic face recognition,’’ IEEE Trans. Pattern Anal. MachIntell., vol. 40, no. 1, pp. 148–162, Jan. 2018.

[6] Y. Duan, J. Lu, J. Feng, and J. Zhou, ‘‘Context-aware local binary feature learning for face recognition,’’ IEEE Trans.Pattern Anal. Mach. Intell., vol. 40, no. 5, pp. 1139–1153, May2018.

[7] S. Chakraborty, S. K. Singh, and P. Chakraborty, ‘‘Local gradient hexa pattern: A descriptor for face recognition and retrieval,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 28,no. 1, pp. 171–180, Jan. 2018.

[8] F. P. Filippidou and G. A. Papakostas, "Single Sample Face Recognition Using Convolutional Neural Networks for Automated Attendance Systems," 2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS), Fez, Morocco, 2020,

[9] S. Manna, S.Ghildiyal and K. Bhimani, "Face Recognition from Video using Deep Learning," 2020 5th International Conference On Communication and Electronics Systems (ICCES),

Coimbatore, India, 2020.

[10] Arpit Singh, Saumya Bhatt, Vishal Nayak, Manan Shah, “Automation of surveillance systems using deep learning and facial recognition, International Journal of System Assurance Engineering Management, January 2023.

APPENDIX

A: SOURCE CODE

#Data Preprocessing

```
import numpy as np

import pandas as PD

from warnings import filterwarnings

filterwarnings("ignore")

df = PD.read_csv('D:/Main project-2/automatic-classroom-attendance-system-master/output/attendance.csv')

print("", df.head())

print("", df.shape)

print("", df.isnull().sum())

df.dropna(inplace=True)

df.duplicated().sum()

print("", df['names'])

df['names'].apply(lambda x: x.split())

df['status'] = df['status'].apply(lambda x: x.split())

df['names'] = df['names'].apply(lambda x: [i.replace(" ", "") for i in x])

df['tags'] = df['names'] + df['status']

new_df = df[['index', 'tags']]

print("", new_df)

new_df['tags'].apply(lambda x: " ".join(x))

new_df

new_df['tags'] = new_df['tags'].apply(lambda x: " ".join(x))

new_df['tags'] = new_df['tags'].apply(lambda x: x.lower())
```

#Face Encoding

```
import face_recognition

import cv2

import os

import pickle

# Load face encodings

print("[INFO] Loading encodings...")

data = pickle.loads(open("D:/Main project-2/automatic-classroom-attendance-system-master/output/encodings2.pickle", "rb").read())

# Load input image

image = cv2.imread("D:/Main project-2/automatic-classroom-attendance-system-master/dataset/aman/IMG_20200907_082833.jpg")

rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Recognizing faces

print("[INFO] Detecting faces...")

start = time.time()

# Detect face locations

boxes = face_recognition.face_locations(rgb, model="hog") # Or use model="cnn" for higher accuracy

end = time.time()

print("[INFO] Face detection took: {} ms".format((end - start) * 1000))

# Encoding faces

print("[INFO] Encoding faces...")

encodings = face_recognition.face_encodings(rgb, boxes)

names = []

# Loop over the recognized faces

for ((top, right, bottom, left), name) in zip(boxes, names):

# Draw the predicted face name on the image
```

```
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)
    y = top - 15 if top - 15 > 15 else top + 15
    cv2.putText(image, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
                0.75, (0, 255, 0), 2)
# Show the output image
cv2.imwrite("output/output_image.jpg", image)
cv2.imshow("Image", image)
cv2.waitKey(0)
```

#Face Recognition Model

```
from face_recognizer.detect_faces import face_detection
from imutils.video import VideoStream
from datetime import datetime
from imutils.face_utils import FaceAligner
from imutils.face_utils import rect_to_bb
import imutils
import pickle
import cv2
import dlib
import os
import time
# Load face encodings
print("[INFO] Loading encodings...")
data = pickle.loads(open("D:/Main project-2/automatic-classroom-attendance-system-master/output/encodings2.pickle", "rb").read())
# Initialize face aligner
print("[INFO] Initializing face aligner...")
```

```
aligner = FaceAligner(landmarks_path="D:/Main project-2/automatic-classroom-attendance-system-master/output/shape_predictor_68_face_landmarks.dat", desiredLeftEye=(0.35, 0.35), desiredFaceWidth=256)

# Start video stream

print("[INFO] Starting video stream...")

vs = VideoStream(src=0).start()

# Time the recognition process

print("[INFO] Recognizing faces...")

start = time.time()

# Recognize faces in video stream

while True:

 # Capture frame-by-frame

    frame = vs.read()
```

#Training Model

```
 import os

import pickle

import numpy as np

from face_recognizer.utils import create_recognizer_dataset, save_dataset


# Define parameters

DATASET_PATH = "path/to/dataset"

MODEL_PATH = "path/to/models/recognizer.pkl"

# Create the dataset

create_recognizer_dataset(DATASET_PATH)
```

```
# Train the recognizer model

face_recognizer = face_recognition.FaceRecognizer()

face_recognizer.train(X_train, np.array(y_train))

# Save the trained model

save_dataset(face_recognizer, MODEL_PATH)

print("Model trained and saved successfully!")
```

#Save the trained recognizer

 # Video Feed Face Recognition

```
from imutils.video import VideoStream

from datetime import datetime

from imutils.face_utils import FaceAligner

from face_recognizer.recognize_faces import face_recognition

from face_recognizer.utils import create_recognizer_dataset, save_dataset

# Define parameters

MODEL_PATH = "D:/Main project-2/automatic-classroom-attendance-system-master/models/shape_predictor_68_face_landmarks.dat"

recognizer_model_path = "D:/Main project-2/automatic-classroom-attendance-system-master/models/recognizer.pkl"

dataset_path = "D:/Main project-2/automatic-classroom-attendance-system-master/dataset/"

output_path = "D:/Main project-2/automatic-classroom-attendance-system-master/output/"
```

#Face Recognition in Images

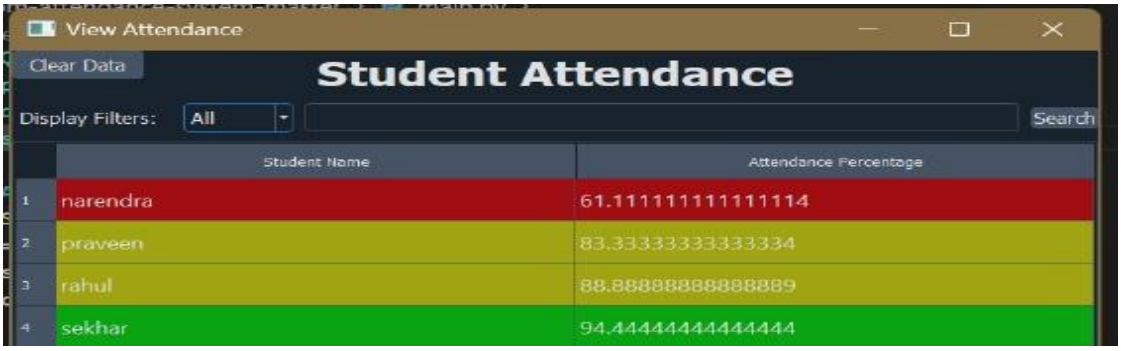
```
import cv2

import pickle
```

```
import face_recognition

# Load face encodings

print("[INFO] Loading encodings...")

data = pickle.loads(open("D:/Main project-2/automatic-classroom-attendance-system-master/output/encodings2.pickle", "rb").read())

# Load input image

image = cv2.imread("D:/Main project-2/automatic-classroom-attendance-system-master/dataset/aman/IMG_20200907_082833.jpg")

# Recognizing faces

print("[INFO] Detecting faces...")

start = time.time()

# Detect face locations

boxes = face_recognition.face_locations(rgb, model="hog") # Or use model="cnn" for higher accuracy

end = time.time()

print("[INFO] Face detection took: {} ms".format((end - start) * 1000))

# Encoding faces

print("[INFO] Encoding faces...")

encodings = face_recognition.face_encodings(rgb, boxes)

names = []
```

Note on Project Code:

Due to the extensive nature of the project code, we are unable to include it in its entirety within this report. Instead, the complete project code has been uploaded to GitHub. You can access it via the following link: https://github.com/praveen9392/Project.git.

B: SCREENSHOTS



ANNEXURE

Name of the Project: Intelligent Video-Based Attendance System

Name of the students: Bathala Pavankumar, G Rahul, Basireddy Narendra Reddy, E Gnana Sai.

Name of the Guide: Dr. T KESAVA RAO, M.Tech , Ph.D. ,Associate Professor

Annexure: Additional Information

Error Handling: To improve the robustness and user-friendliness of your code, you can implement try-except blocks to catch exceptions and handle them gracefully.

Model Validation: Include code to validate your model's performance using metrics such as accuracy, precision, recall, and F1-score. This will help you ensure that your model is performing well and meeting your project's requirements.

Data Preprocessing: Implement code to preprocess your data, such as cleaning, normalization, and feature extraction. This will help ensure that your data is in a suitable format for your model.

User Interface: Create a user-friendly interface for your project, such as a web application or command-line interface. This will make it easier for users to interact with your project and understand its functionality.

Documentation: Include thorough documentation for your project, such as a README file, code comments, and user guides. This will help others understand your project's functionality and how to use it.

Testing: Include code to test your project's functionality and ensure that it's working as expected. This can include unit tests, integration tests, and end-to-end tests.

Deployment: Include code to deploy your project, such as a cloud deployment script. This will make it easier for others to use and deploy your project.