“FACE RECOGNITION BASED ATTENDANCE SYSTEM USING PYTHON”

A Dissertation
Submitted in Partial fulfillment for the award of

Bachelor of Technology in Computer Science and Engineering Submitted to:

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYABHOPAL (M.P.)

Submitted By:

NIKHIL SHRIVASTAVA
ENROLL. NO. 0905CS201116
NITIN GANGWANI
ENROLL. NO. 0905CS201118
PRIYANSHIKA SINGH
ENROLL. NO. 0905CS201131
SAKSHI GALGALE
ENROLL. NO. 0905CS201151

Under the Guidance of:
Mr. ARUN AGARWAL
Assistant Professor

Department of Computer Science and Engineering

INSTITUTE OF TECHNOLOGY &MANAGEMENTGWALIOR (M.P.) – 475001

CANDIDATE DECLARATION

We **Nikhil Shrivastava, Nitin Gangwani, Priyanshika Singh, and Sakshi Galgale** students of B.Tech. (Computer Science) VII/VIII semester **Roll No 0905CS201116, 0905CS201118, 0905CS201131 and 0905CS201151** hereby declare that the Project entitled **"FACE RECOGNITION BASED ATTENDANCE SYSTEM USING PYTHON"** which is being submitted to Department of computer science & Engineering in ITM, Gwalior is our authentic work carried out in our VII/VIII semesters. We declare that our work has not been submitted in part or in full to any other university or institution for the award of any degree or diploma.

NIKHIL SHRIVASTAVA (0905CS201116)

NITIN GANGWANI (0905CS201118)

PRIYANSHIKA SINGH (0905CS201131)

SAKSHI GALGALE (0905CS201151)

CERTIFICATE

This is to certify that the project entitled **“FACE RECOGNITION BASED ATTENDANCE SYSTEM USING PYTHON”** being
submitted by **Nikhil Shrivastava (0905CS201116), Nitin Gangwani (0905CS201118), Priyanshika Singh(0905CS201131), and Sakshi Galgale (0905CS201151)** in partial fulfillment of the requirement for the award of B. Tech. degree in Computer Science & Engineering to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) is a record of bonafide work done by them, under my guidance.

HEAD OF DEPARTMENT:	**PROJECT GUIDE:**
Dr. Rishi Soni	**Mr. Arun Agarwal**
Department of CSE	**Assistant Professor**
ITM, Gwalior (M.P)	**Department of CSE**
	ITM, Gwalior (M.P)

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The most awaited moment of successful completion of an endeavor is always a result of persons involved implicitly or explicitly there in and it is impossible without the help and guidance of the people around us. The successful completion of this thesis report is a result of dedicated efforts put in by many people and this report would have been incomplete without giving due credit to them. This acknowledgement is indeed a small token of gratitude in recognition of their help.

We express our gratitude to **Dr. Meenakshi Mazumdar,** Director, Institute of Technology and Management, Gwalior for her valuable suggestions. We feel highly privileged to express our sincere thanks and deepest sense of gratitude to our guide **Mr. Arun Agarwal** who has spared her precious time & guided us to present this dissertation. We sincerely acknowledge her for extending her valuable guidance, support for literature, critical review of report & above all the moral support she has provided us at all stages of this dissertation. We wish to thank our Head of the department, **Dr. Rishi Soni**, for his support and encouragement. Our thanks are extended to all staff members and people who directly or indirectly helped us in achieving our goals. We could not have completed this work without the guidance, confidence and encouragement of our parents. They deserve more credit than We can give for instilling in us a good work ethics and a desire to always learn more. Our thanks are also extended to all our friends who have kept our spirits high throughout this work.

NIKHIL SHRIVASTAVA (0905CS201116)

NITIN GANGWANI (0905CS201118)

PRIYANSHIKA SINGH (0905CS201131)

SAKSHI GALGALE (0905CS201151)

ABSTRACT

Facial Recognition-Based Attendance System

In contemporary educational institutions, managing student attendance efficiently is crucial for ensuring academic integrity, compliance with regulations, and effective monitoring of student engagement. Traditional methods of attendance tracking, such as manual roll-calls or barcode scanning, are often labor-intensive, time-consuming, and prone to errors. To address these challenges, this project proposes the development of a comprehensive Facial Recognition Based Attendance System (FRBAS).

The FRBAS is an innovative solution that leverages advancements in computer vision and facial recognition technology to automate the attendance management process. The system consists of several interconnected modules, each designed to fulfill specific functions seamlessly.

Firstly, the registration module enables administrators to onboard new students into the system. Through a user-friendly interface, administrators can capture students' facial images and associate them with their personal information, such as name and enrollment number. These images are then processed using facial recognition algorithms to generate unique facial encodings, which serve as the basis for future identification.

During attendance sessions, the system employs live video feed from designated cameras to capture real-time images of students present in the classroom or designated area. Through face detection algorithms, the system identifies individuals within the captured frames and matches their facial features against the stored encodings. Upon successful recognition, the attendance of the corresponding student is automatically marked, along with relevant metadata such as timestamp and subject.

The FRBAS also incorporates robust security measures to ensure the integrity and privacy of student data. Facial encodings are securely stored and encrypted, mitigating potential risks associated with data breaches or unauthorized access. Additionally, the system adheres to established data protection regulations, such as GDPR and CCPA, to safeguard students' privacy rights.

Furthermore, the system offers extensive reporting and analytics capabilities through its viewing module. Administrators can access comprehensive attendance records, including detailed reports on student attendance patterns, late arrivals, and absences. Additionally, the

system provides insights into unique entry times, enabling administrators to identify trends and address potential issues proactively.

Overall, the Facial Recognition Based Attendance System represents a paradigm shift in attendance management within educational institutions. By harnessing the power of facial recognition technology, the system streamlines administrative processes, enhances accuracy, and promotes accountability. Furthermore, its versatility and scalability make it suitable for implementation across various educational settings, ranging from schools and colleges to training centers and corporate campuses. As educational institutions continue to embrace digital transformation, the FRBAS emerges as a pioneering solution to meet the evolving demands of modern attendance tracking.

TABLE OF CONTENTS

Contents

CHAPTER 1. INTRODUCTION

1.1 INTRODUCTION

In recent years, technology has profoundly transformed various aspects of our lives, including how organizations manage attendance. Traditional methods, such as manual sign-ins or punch cards, are often inefficient, error-prone, and susceptible to manipulation. To address these shortcomings, our project aims to develop a Face Recognition Based Attendance System using Python. This system leverages the advancements in artificial intelligence and computer vision to offer a more reliable, accurate, and user-friendly alternative to traditional attendance tracking methods. The motivation to develop a Face Recognition Based Attendance System from the necessity to overcome the limitations of traditional methods. Manual systems are not only labor-intensive but also susceptible to various forms of manipulation, such as buddy punching, where one individual clocks in on behalf of another. Additionally, the maintenance and verification of physical attendance records can be cumbersome and prone to errors. These challenges highlight the need for a more sophisticated approach that can provide higher accuracy and reliability.

Face recognition technology offers a promising solution by providing a contactless, secure, and efficient method for identifying individuals. Unlike fingerprint or card-based systems, face recognition does not require physical interaction, thereby enhancing hygiene and reducing the risk of fraudulent activities. With advancements in machine learning and the availability of powerful computing resources, implementing a face recognition system has become more feasible and practical. Python, as a programming language, has gained popularity due to its simplicity and the extensive libraries available for machine learning and computer vision. Libraries such as OpenCV, dlib, and face recognition provide robust tools for developing face recognition applications. OpenCV offers a comprehensive suite of computer vision functions, while dlib provides state-of-the-art machine learning algorithms for face detection and feature extraction. The face recognition library simplifies the implementation of face recognition by building on top of these powerful libraries. Integrating face recognition into an attendance system involves several steps. Initially, the system captures an image or video of the individual. The face detection algorithm then identifies the presence of a face in the frame and extracts relevant features. These features are compared against a pre-registered database of facial images to determine the individual's identity. Upon successful identification, the system records the attendance in real-time.

One of the key advantages of using a Face Recognition Based Attendance System is its ability to operate in real-time, ensuring quick and efficient attendance marking without the need for manual intervention. This not only saves time but also minimizes the possibility of errors and fraud. Additionally, the system can handle large volumes of data and scale to accommodate growing organizations. The benefits of a Face Recognition Based Attendance System are manifold. It enhances the accuracy and reliability of attendance records, reduces administrative workload, and provides a secure and hygienic method of tracking attendance. Furthermore, it can be integrated with existing organizational systems, such as payroll and human resource management software, to streamline operations. In educational institutions, this technology can ensure that students' attendance is recorded accurately, contributing to better monitoring of their presence and participation. In corporate settings, it can improve employee attendance tracking, helping to enforce punctuality and accountability. Additionally, the system can be employed in various other contexts, such as event management and security, where accurate identification and tracking are essential.

The integration of facial recognition technology into attendance tracking represents a paradigm shift. By relying on distinctive biometric features, this system not only ensures accurate identification but also provides a seamless and contactless experience for users. This introduction delves into the rationale behind adopting facial recognition, highlighting its potential to redefine attendance management in the digital age.

1.2 OBJECTIVE

The primary objectives of this project are as follows:

1. **Automate Attendance Tracking:**
    - Develop a system to automatically track and record attendance using face recognition, eliminating the need for manual sign-ins.
2. **Enhance Accuracy:**
    - Ensure high accuracy in identifying individuals to reduce errors in attendance records.
3. **Real-Time Processing:**
    - Implement real-time face recognition to instantly mark attendance as individuals arrive.
4. **User-Friendly Interface:**
    - Design an intuitive and easy-to-use interface for users to interact with the system.
5. **Reduce Fraud:**
    - Minimize the possibility of fraudulent attendance practices, such as buddy punching, by using unique facial features for identification
6. **Integration Capability:**
    - Ensure the system can be seamlessly integrated with existing attendance management and payroll systems.
7. **Scalability:**
    - Develop the system to handle large volumes of data, accommodating organizations of various sizes.
8. **Security and Privacy**:
    - Implement robust security measures to protect the facial data and attendance records, ensuring compliance with privacy regulations.
9. **Cost-Effectiveness:**
    - Provide a cost-effective solution compared to traditional biometric systems, reducing the need for additional hardware like fingerprint scanners.
10. **Hygienic and Contactless:**
    - Offer a contactless method for attendance tracking, promoting hygiene and reducing the spread of germs, especially relevant in the context of health crises.
11. **Comprehensive Reporting:**
    - Enable detailed reporting and analytics on attendance patterns, aiding in better management and decision-making.
12. **Adaptability to Various Environments:**
    - Design the system to be adaptable for different environments, such as schools, offices, and events, ensuring wide applicability and flexibility.

1.3 PROBLEM SELECTION

Traditional attendance management systems, widely used in educational institutions, corporate offices, and various organizational settings, suffer from numerous inefficiencies and vulnerabilities. These systems typically rely on manual sign-ins, punch cards, or RFID tags, all of which are susceptible to several critical issues. Manual attendance methods require significant time to administer and verify, leading to productivity losses, as teachers or administrators must spend valuable time recording attendance instead of focusing on their primary responsibilities. Moreover, manual entry systems are prone to human error, resulting in mistakes in recording attendance, such as missed entries or incorrect data, which can lead to inaccurate records affecting payroll, academic records, and organizational planning.

Additionally, traditional systems are easily manipulated. Buddy punching, where one employee clocks in for another, is a common problem that undermines the integrity of attendance records, leading to financial losses for organizations due to inaccurate payroll processing. The administrative burden of maintaining and managing paper-based attendance records is labor-intensive and cumbersome, requiring substantial effort to organize, store, and retrieve records, especially in large organizations with many employees or students. Furthermore, systems that rely on physical tokens, such as ID cards or RFID tags, are inconvenient as these tokens can be easily lost, forgotten, or damaged, causing further delays and administrative overhead.

Hygiene concerns also arise with traditional biometric systems, such as fingerprint scanners, which require physical contact and raise issues, particularly relevant during health crises like the COVID-19 pandemic. These concerns necessitate a shift towards contactless solutions. Given these challenges, there is a clear need for a more efficient, accurate, and secure attendance management system. A Face Recognition Based Attendance System using Python addresses these issues by providing a contactless, automated solution that enhances accuracy, reduces administrative burden, and mitigates the risk of fraud. By leveraging advanced face recognition technology, this system can significantly improve the reliability and efficiency of attendance tracking across various settings.

1.4 MACHINE LEARNING

To use machine learning in multiple diseases prediction:

1. **Face Detection:**

Explanation: The first step in any face recognition system is to detect faces within an image or video frame. Machine learning algorithms, particularly those involving convolutional neural networks (CNNs), are used to identify and localize faces. Tools like OpenCV and dlib are commonly employed for this purpose. These algorithms scan the input frame for patterns that match human facial features, such as eyes, nose, and mouth, and generate bounding boxes around detected faces.

Importance: Accurate face detection is crucial as it ensures that only relevant parts of the image are processed for recognition, improving both the efficiency and accuracy of the system.

2. **Feature Extraction:**

Explanation: Once faces are detected, the next step is to extract unique features from each face. This involves identifying key points and characteristics that distinguish one face from another. Deep learning models, such as those based on CNNs, are trained to recognize these features by analyzing large datasets of facial images.

Importance: Effective feature extraction is vital for differentiating between individuals. It ensures that the recognition system can accurately identify and distinguish between different faces, even if they have similar appearances.

3. **Face Recognition:**

Explanation: In the face recognition phase, the system compares the extracted features of the detected face with a database of known faces. This comparison is often done using machine learning techniques like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), or more advanced deep learning models like FaceNet or VGGFace. These models generate embeddings (numerical representations) of faces that can be compared for similarity.

Importance: The accuracy of the face recognition process determines the overall effectiveness of the attendance system. High precision in matching faces ensures reliable attendance records and minimizes false positives and negatives.

4. **Model Training and Dataset Management:**

Explanation: Developing an accurate face recognition system requires a robust training process. This involves collecting a large dataset of facial images with diverse lighting conditions, angles, and

expressions. The dataset is then used to train machine learning models to improve their ability to generalize and perform well on unseen data. Data augmentation techniques, such as rotation, scaling, and cropping, are often used to enhance the dataset.
Importance: Proper training and dataset management are critical for building a model that performs well in real-world scenarios. A well-trained model can handle variations in lighting, pose, and expression, ensuring reliable face recognition in different environments.

5. Real-Time Processing and Optimization:

Explanation: For an attendance system to be practical, it must process facial recognition in real-time. This requires optimizing the machine learning models to run efficiently on available hardware, whether it be on local servers, edge devices, or cloud platforms. Techniques such as model pruning, quantization, and the use of efficient inference engines (like TensorRT or OpenVINO) are employed to reduce latency and computational load.
Importance: Real-time processing is essential for practical usability. Optimizing the system ensures that it can quickly and accurately recognize faces and mark attendance without causing delays, making it suitable for high-traffic environments like schools and offices.

1.5 DATASETS

The successful implementation and training of a Facial Recognition Based Attendance System (FRBAS) rely heavily on the availability of high-quality datasets for facial recognition and attendance records. The dataset used in this project serves as the foundation for training the facial recognition algorithms and validating the system's performance in real-world scenarios.

- **Facial Images Dataset**: The primary component of the dataset consists of facial images captured from students during the registration process. These images represent a diverse range of facial expressions, poses, and lighting conditions to ensure robustness and generalization of the facial recognition model. The dataset may include images collected from various sources, such as enrollment sessions, student ID photos, or publicly available face image databases.
- **Annotations and Labels**: Each facial image in the dataset is accompanied by corresponding annotations or labels, indicating the identity of the individual depicted in the image. These annotations are crucial for supervised learning algorithms to learn the associations between facial features and individual identities accurately. Additionally, annotations may include metadata such as student enrollment numbers, names, and other relevant information.
- **Attendance Records**: In addition to facial images, the dataset incorporates attendance records associated with each student. These records provide ground truth labels for training and evaluation purposes, indicating whether a student was present or absent during specific attendance sessions. Attendance records may include timestamps, session identifiers, and other contextual information to facilitate comprehensive analysis and validation of the FRBAS.
- **Data Augmentation and Preprocessing**: To enhance the diversity and richness of the dataset, various data augmentation techniques may be applied. These techniques include geometric transformations (e.g., rotation, scaling, cropping), color adjustments, and noise addition. Additionally, preprocessing steps such as face alignment, normalization, and feature extraction are performed to ensure consistency and quality in the dataset.
- **Privacy and Ethical Considerations**: Privacy and ethical considerations play a critical role in the collection and usage of facial recognition datasets, particularly in educational settings. Measures must be implemented to protect the privacy and rights of students, including obtaining informed consent, anonymizing sensitive information, and adhering to relevant data protection regulations (e.g., GDPR, FERPA).

Overall, the dataset used in the FRBAS project serves as a fundamental resource for training, testing, and validating the facial recognition algorithms and attendance management functionalities. By leveraging a diverse and well-curated dataset, the FRBAS aims to achieve high accuracy, reliability, and fairness in student identification and attendance tracking, while upholding principles of privacy, security, and ethical conduct.

1.6 RESEARCH GAPS

While there has been significant progress in this field using machine learning, there are still some research gaps that need to be addressed :

1. **Handling Occlusions and Variations**: Current face recognition models often struggle with occlusions, such as hats, glasses, or masks, and variations in facial expressions, lighting, and angles. While datasets like LFW and VGGFace2 provide a diverse range of images, there is still a need for more specialized datasets and algorithms that can effectively handle these challenges. Research is needed to develop models that can accurately recognize faces under such conditions without significant drops in performance.
2. **Bias and Fairness:** Face recognition systems have been shown to exhibit biases based on race, gender, and age, leading to disparities in accuracy across different demographic groups. This issue arises due to the lack of diverse representation in training datasets. Research is required to explore methods for mitigating these biases, such as developing more balanced datasets and implementing fairness-aware algorithms. Ensuring that the systems perform equitably across all user groups is critical for their widespread adoption.
3. **Data Privacy and Security:** The use of facial data raises significant privacy and security concerns. Current research often overlooks the development of secure data storage and transmission methods. There is a need for research into techniques such as differential privacy, homomorphic encryption, and secure multi-party computation to protect users' facial data from unauthorized access and misuse, while still enabling the effective functioning of face recognition systems.
4. **Scalability and Real-Time Performance:** While many face recognition systems demonstrate high accuracy in controlled environments, their performance can degrade when scaled to real-world applications involving large user bases and high traffic. Research is needed to improve the scalability of these systems, ensuring they can handle large datasets and real-time processing demands efficiently. This includes optimizing algorithms and hardware acceleration techniques to maintain performance without sacrificing accuracy.
5. **Integration with Multimodal Systems:** Face recognition systems are often deployed as standalone solutions, but there is a growing interest in integrating them with other biometric and non-biometric identification methods, such as voice recognition, fingerprint scanning, and behavioral biometrics. Research is needed to explore the synergistic benefits of multimodal systems, including how they can improve accuracy, reliability, and user experience. Developing frameworks for seamless integration and interoperability among different identification technologies is a crucial area for further study.

1.7 PRECISION AND ACCURACY

The precision and accuracy of a face recognition-based attendance system can be evaluated using various metrics such as accuracy, precision, recall, F1 score, and area under the curve (AUC) of the receiver operating characteristic (ROC) curve. Here is a brief explanation of these metrics:

1. **Accuracy:** Accuracy is the proportion of correctly identified faces over the total number of faces. It measures how well the system predicts the identity of individuals. High accuracy indicates that the system performs well in recognizing faces and marking attendance correctly.
2. **Precision:** Precision is the proportion of true positive identifications over the total number of positive identifications (true positive + false positive). It measures the proportion of correctly identified faces among all identified faces. High precision ensures that the system has a low false positive rate, meaning that when it identifies a face, it is often correct.
3. **Recall:** Recall is the proportion of true positive identifications over the total number of

actual faces (true positive + false negative). It measures the proportion of actual faces correctly identified by the system. High recall ensures that the system has a low false negative rate, meaning it successfully identifies most of the actual faces.

4. **F1 Score:** The F1 score is the harmonic mean of precision and recall. It combines precision and recall into a single metric that balances both measures. The F1 score provides a single measure of the system's performance, especially useful when dealing with imbalanced datasets or when both precision and recall are equally important.
5. **AUC of the ROC Curve**: The AUC of the ROC curve measures the system's ability to discriminate between positive (correctly identified) and negative (incorrectly identified) faces. It represents the probability that a positive face will be ranked higher than a negative face by the system. A high AUC indicates a good measure of separability, meaning the system is effective at distinguishing between correct and incorrect face identifications across various threshold levels.

CHAPTER 2. LITERATURE SURVEY

In the landscape of Facial Recognition-based Attendance Systems (FRAS), a rich tapestry of previous work lays the foundation for the advancements and innovations that shape contemporary solutions. This section provides an overview of the key themes, methodologies, and contributions from prior research, offering insights into the evolution of FRAS.

2.1 PREVIOUS WORK

Early Explorations:

Early works in FRAS focused on fundamental aspects of facial recognition, exploring techniques such as eigenface-based recognition and template matching. These seminal studies laid the groundwork for subsequent developments, highlighting the potential of facial features as unique identifiers.

Feature Extraction Techniques:

1. **Local Binary Pattern (LBP):**
   Researchers delved into the efficacy of LBP for facial feature extraction. LBP's ability to capture local texture patterns emerged as a pivotal advancement, contributing to enhanced accuracy in identifying facial features.
2. **Histogram of Oriented Gradients (HOG):**
   Previous research examined the application of HOG in FRAS, particularly in addressing challenges posed by variations in lighting conditions. HOG's focus on gradient information added a robust dimension to facial feature representation.
3. **Principal Component Analysis (PCA) and Eigenfaces:**
   Early FRAS solutions often utilized PCA and eigenface techniques for dimensionality reduction in facial images. These approaches, while foundational, faced limitations in handling variations in pose and illumination.

Local Feature-based Approaches:

1. **Scale-Invariant Feature Transform (SIFT):**
   SIFT gained attention for its robustness in detecting local key points and descriptors. Researchers explored its application in FRAS, emphasizing its suitability for diverse facial recognition scenarios.
2. **Speeded-Up Robust Features (SURF):**
   Previous work investigated the efficiency and robustness of SURF in local feature detection and description within FRAS. Its computational advantages made it well-suited for real-time applications.

Deep Learning Paradigms:

1. **Convolutional Neural Networks (CNNs):**
   The advent of deep learning marked a paradigm shift in FRAS. Early implementations explored the application of CNNs, demonstrating their prowess in automatically learning hierarchical features for accurate facial recognition.
2. **Siamese Networks:**
   Researchers investigated Siamese networks for one-shot learning scenarios in FRAS, showcasing their adaptability to situations with limited labeled training data.
3. **Generative Adversarial Networks (GANs):**
   GANs were explored for generating synthetic facial images, contributing to the augmentation of training datasets. This approach bolstered the robustness of FRAS models, particularly in

scenarios with limited real-world data.

Real-World Implementations:
Numerous studies delved into the practical implementation of FRAS various settings. Previous work showcased the adaptability of FRAS in educational institutions, workplaces, and public spaces, providing insights into system architectures, user experiences, and challenges encountered during deployment.

Ethical and Privacy Considerations:
Contemporary research underscored the ethical implications and privacy concerns associated with FRAS. Scholars delved into issues such as consent, data storage practices, and the potential misuse of facial data. This body of work contributes to the ongoing dialogue surrounding the responsible deployment of FRAS.
In conclusion, the panorama of previous work in FRAS reflects a continuous evolution from foundational studies to the embrace of advanced methodologies, such as deep learning paradigms. The collective insights from early explorations, feature extraction techniques, and real-world implementations serve as a springboard for the ongoing refinement and innovation in Facial Recognition-based Attendance Systems. Understanding this historical context is pivotal for shaping the requirements and expectations of contemporary FRAS solutions.

2.2 ALGORITHMS USED

1. **Facial Recognition Algorithm**:
    - The facial recognition algorithm used in the Facial Recognition Based Attendance System (FRBAS) is primarily based on the principles of deep learning and computer vision.
    - Specifically, the system employs Convolutional Neural Networks (CNNs) for facial feature extraction and recognition. CNNs are well-suited for image classification tasks, including facial recognition, due to their ability to learn hierarchical representations of visual features.
    - During the registration process, facial images of students are captured and preprocessed to extract facial features. These features are then encoded into high-dimensional vectors using CNN-based feature extraction networks, such as VGG, ResNet, or MobileNet.
    - The encoded facial features serve as embeddings that uniquely represent each individual's facial characteristics. These embeddings are then stored securely in a database for later retrieval and comparison during attendance marking.
    - During attendance sessions, live video feed from cameras is processed in real-time using the same facial recognition algorithm. Faces detected in the video frames are passed through the CNN-based feature extraction network to generate embeddings.
    - These embeddings are compared against the stored embeddings of registered students using similarity metrics such as cosine similarity or Euclidean distance. If a match is found above a predefined threshold, the student's attendance is marked accordingly.
    - The facial recognition algorithm continually learns and adapts to variations in facial appearance, illumination conditions, and occlusions, ensuring robust performance in diverse environments.
2. **Face Detection Algorithm**:
    - Face detection is a crucial component of the facial recognition system, responsible for locating and extracting facial regions from input images or video frames.
    - The FRBAS utilizes state-of-the-art face detection algorithms, such as Haar cascades,

    - Histogram of Oriented Gradients (HOG), or more advanced deep learning-based methods like Single Shot MultiBox Detector (SSD) or Faster R-CNN.
    - These algorithms analyze input images or video frames to identify regions containing human faces, typically represented as bounding boxes.
    - Once faces are detected, they are passed to the facial recognition algorithm for further processing, including feature extraction and recognition.
3. **Attendance Marking Algorithm**:
    - The attendance marking algorithm determines whether a student's presence is detected during an attendance session based on the results of facial recognition.
    - Upon successful recognition of a student's face, the algorithm marks the student as present in the attendance records, along with relevant metadata such as timestamp and subject.
    - Attendance marking may involve additional logic to handle edge cases, such as multiple detections of the same student within a short time frame or false positives/negatives in recognition results.
    - The algorithm ensures the accuracy and reliability of attendance records, minimizing errors and inconsistencies in marking attendance.
4. **Data Augmentation and Preprocessing**:
    - Data augmentation and preprocessing techniques are applied to enhance the robustness and generalization of the facial recognition algorithm.
    - Augmentation techniques include geometric transformations (e.g., rotation, scaling, translation), color adjustments, and noise addition. These techniques increase the diversity of training data and improve the algorithm's ability to handle variations in facial appearance.
    - Preprocessing steps such as face alignment, normalization, and histogram equalization are performed to standardize input images, remove noise, and enhance feature visibility.
    - Additionally, data augmentation and preprocessing help mitigate overfitting and improve the algorithm's performance on unseen data.
5. **Real-Time Processing and Multithreading**:
    - To ensure real-time performance and responsiveness, the FRBAS implements multithreading and parallel processing techniques.
    - Live video feed from cameras is processed concurrently using multiple threads, allowing for efficient utilization of computational resources and minimizing latency.
    - Multithreading enables the system to handle simultaneous tasks such as face detection, feature extraction, and attendance marking in parallel, optimizing throughput and responsiveness.

Overall, the Facial Recognition Based Attendance System leverages a combination of advanced algorithms and techniques to achieve accurate, reliable, and efficient attendance management in educational institutions. By integrating state-of-the-art facial recognition technology with robust data processing pipelines, the system offers a scalable and adaptable solution for automating attendance tracking while ensuring fairness, transparency, and privacy compliance.

CHAPTER 3. SYSTEM STUDY

3.1 EXISTING SYSTEM ALONG WITH LIMITATION

1. **Manual Attendance Systems**
   - **Description**: This traditional method involves manually marking attendance in a register or a spreadsheet by calling out names or having individuals sign in.
   - **Limitations**:
     - **Time-Consuming**: Takes a significant amount of time, especially for large groups.
     - **Human Error**: Prone to mistakes such as incorrect marking or missed entries.
     - **Fraud**: Susceptible to proxy attendance where one person marks attendance for another.
2. **RFID-Based Attendance Systems**
   - **Description**: Utilizes Radio Frequency Identification (RFID) cards that individuals swipe to mark their attendance.
   - **Limitations**:
     - **Cost**: Requires the purchase and maintenance of RFID cards and readers.
     - **Card Misplacement**: Cards can be easily lost or forgotten.
     - **Proxy Attendance**: Individuals can swipe cards on behalf of others.
3. **Biometric Attendance Systems (Fingerprint/Hand Geometry)**
   - **Description**: Uses biometric data such as fingerprints or hand geometry to record attendance.
   - **Limitations**:
     - **Hygiene Issues**: Direct contact with scanners raises concerns, especially in the context of health (e.g., during a pandemic).
     - **Sensor Quality**: Low-quality sensors can lead to false negatives/positives.
     - **Environment Dependent**: Fingerprint scanners may not work well if fingers are dirty or wet.
4. **QR Code-Based Attendance Systems**
   - **Description**: Individuals scan a QR code using their mobile devices to register their attendance.
   - **Limitations**:
     - **Technical Requirements**: Requires individuals to have smartphones with a QR code scanner.
     - **Network Dependency**: Requires a stable internet connection to sync attendance data.
     - **Security**: QR codes can be easily replicated or shared.

3.2 PROPOSED SYSTEM ALONG WITH INTENDED OBJECTIVES

The proposed system, the Facial Recognition Based Attendance System (FRBAS), aims to revolutionize the process of attendance tracking in educational institutions through the integration of advanced facial recognition technology. The system is designed to automate attendance management, enhance accuracy and reliability, improve administrative efficiency, ensure compliance and accountability, facilitate real-time monitoring, enhance security and privacy, and promote technological innovation. These objectives align with the overarching goal of optimizing attendance management processes and supporting the educational mission of institutions.

1. **Automate Attendance Management**:
   - Objective: The primary objective of the FRBAS is to automate the process of attendance tracking, eliminating the need for manual methods such as roll-calls or barcode scanning. By leveraging facial recognition technology, the system

streamlines attendance management processes, reducing administrative burden and improving efficiency.

2. **Enhance Accuracy and Reliability**:
   - Objective: The FRBAS aims to enhance the accuracy and reliability of attendance records by leveraging facial recognition algorithms to identify and verify students' identities. By minimizing errors and inconsistencies associated with manual data entry, the system ensures the integrity of attendance data, fostering trust and confidence among stakeholders.
3. **Improve Administrative Efficiency**:
   - Objective: Another key objective of the FRBAS is to improve administrative efficiency within educational institutions. By automating attendance management tasks, the system enables administrators to allocate their time and resources more effectively, focusing on strategic initiatives and student support services.
4. **Ensure Compliance and Accountability**:
   - Objective: The FRBAS ensures compliance with regulatory requirements and promotes institutional accountability in attendance tracking. By maintaining accurate and auditable attendance records, the system enables institutions to demonstrate compliance with relevant regulations and standards, mitigating risks and liabilities.
5. **Facilitate Real-Time Monitoring**:
   - Objective: The FRBAS facilitates real-time monitoring of student attendance patterns, enabling administrators to identify trends, address issues, and make data-driven decisions proactively. By providing timely insights into attendance metrics, the system supports strategic planning and resource allocation efforts.
6. **Enhance Security and Privacy**:
   - Objective: Security and privacy are paramount considerations in the design of the FRBAS. The system prioritizes the protection of student data through robust encryption mechanisms, access controls, and compliance with data protection regulations. By safeguarding sensitive information, the system fosters trust and confidence among users.
7. **Promote Technological Innovation**:
   - Objective: By embracing facial recognition technology, the FRBAS promotes technological innovation and digital transformation in educational administration. The system serves as a catalyst for advancing the adoption of cutting-edge technologies to improve operational efficiency and student outcomes.

Overall, the proposed system represents a comprehensive solution for attendance management in educational institutions, addressing key challenges and objectives through the integration of facial recognition technology, advanced algorithms, and user-friendly interfaces. By aligning with the intended objectives, the FRBAS aims to deliver tangible benefits in terms of efficiency, accuracy, compliance, and innovation, ultimately enhancing the educational experience for students and administrators alike.

3.3 FEASIBILITY STUDY

The feasibility study assesses the practicality and viability of implementing the proposed face recognition-based attendance system using Python, considering various dimensions: technical, social, operational, economic, and legal.

- **Technical Feasibility**: The system's technical feasibility is robust, leveraging widely available hardware components such as standard webcams and modern computing devices. With Python and open-source libraries like OpenCV and face_recognition, the system can efficiently detect and recognize faces, ensuring compatibility and ease of integration. Additionally, existing network infrastructure supports real-time data transmission and remote access, further enhancing technical feasibility.

- **Operational Feasibility**: Operationally, the system demonstrates strong feasibility by streamlining attendance management processes. Its intuitive interface and automated attendance marking reduce manual effort, improving operational efficiency. Real-time processing capabilities enable immediate database updates, ensuring timely and accurate attendance records. Additionally, the system's scalability allows for easy expansion to accommodate growing organizational needs, further enhancing operational feasibility.

- **Social Feasibility**: The system's social feasibility is promising, as it addresses user needs and concerns effectively. The user-friendly interface minimizes the learning curve for administrators and users, ensuring widespread acceptance. By enhancing efficiency and accuracy in attendance management, the system fosters positive user experiences. Moreover, stringent measures for data protection and privacy address user concerns, building trust and confidence in the system.

- **Economic Feasibility**: The system's economic feasibility is favourable, with low initial and operational costs. Hardware components like webcams are affordable and readily available, while Python and open-source libraries incur no licensing fees. Development costs are minimized, and maintenance expenses are low, resulting in a quick return on investment. By reducing labour costs associated with manual attendance management, the system offers significant economic benefits to organizations.

- **Legal Feasibility**: Legally, the system complies with data protection laws and regulations, ensuring the privacy and security of users' biometric data. Strong encryption methods safeguard sensitive information, mitigating legal risks associated with data breaches. Regulatory compliance measures uphold the system's integrity and legitimacy, fostering trust among users and stakeholders.

The proposed face recognition-based attendance system exhibits strong feasibility across all dimensions, making it a viable and advantageous solution for attendance management in various organizational settings.

3.3.1. TECHNICAL FEASIBILITY

The technical feasibility of the face recognition-based attendance system using Python evaluates the ability to meet system requirements with available resources.

- **Hardware Requirements**
  1. **Cameras**:
     - **Type**: Standard webcams or IP cameras for image capture.
     - **Resolution**: Moderate to high resolution for clear and accurate face detection.
  2. **Computing Power**:
     - **Processors**: Modern multi-core processors (e.g., Intel i5/i7, AMD Ryzen).
     - **Memory (RAM)**: At least 8GB for smooth performance.
     - **Storage**: Sufficient storage capacity, preferably SSDs, for fast data access.
  3. **Network Infrastructure**:
     - **LAN**: For internal data transfer.
     - **Internet Access**: For remote access and synchronization.
     - **Bandwidth**: Adequate to handle data transmission without latency.
- **Software Requirements**
  1. **Programming Language**:
     - **Python**: For its powerful libraries and ease of use.
  2. **Libraries**:

    - **OpenCV**: For image processing.
    - **face_recognition**: For detecting and recognizing faces.
    - **NumPy**: For numerical operations.
    - **Django/Flask**: For developing a web-based user interface if needed.
3. **Database Management**:
    - **SQLite**: For lightweight database needs.
    - **MySQL**: For handling larger datasets and complex queries.

- **Development and Integration**
1. **Tools**:
    - **IDEs**: PyCharm, VSCode, or Jupyter Notebooks.
    - **Version Control**: Git for tracking changes and collaboration.
2. **Integration**:
    - **APIs**: RESTful APIs for frontend-backend communication.
    - **Cross-Platform Compatibility**: Ensuring operation on various OS (Windows, Linux, macOS).
3. **Testing**:
    - **Unit Testing**: For individual components.
    - **Integration Testing**: Ensuring components work together.
    - **Performance Testing**: Checking system performance under different conditions.

3.3.2. OPERATIONAL FEASIBILITY

Operational feasibility assesses the practicality of implementing and maintaining the face recognition-based attendance system within an organization. This involves evaluating the system's usability, efficiency, reliability, scalability, cost efficiency, and security.

- **User-Friendliness**
    - **Intuitive Interface**: The system offers a user-friendly interface that requires minimal training for both administrators and users, facilitating quick adoption and ease of use.
- **Efficiency and Productivity**
    - **Automated Attendance Marking**: Automates attendance recording, significantly reducing time and effort while minimizing human errors.
    - **Real-Time Processing**: Provides immediate updates to attendance data, ensuring accurate and timely information.
- **Reliability**
    - **Accuracy**: Utilizes advanced face recognition algorithms for high accuracy, reducing false positives and negatives.
    - **Maintenance**: Regular software updates and hardware checks ensure minimal downtime and reliable operation.
- **Scalability**
    - **Expandable System**: Easily scalable to accommodate organizational growth by adding new users and hardware components without major changes.
    - **Adaptability**: Flexible to adapt to various operational needs and integrate with other systems (e.g., HR management).
- **Cost Efficiency**
    - **Operational Costs**: Low ongoing costs with significant savings in labor and physical resources, leading to a quick return on investment.
    - **Maintenance**: Involves minimal costs, primarily for routine maintenance and occasional upgrades.
- **Security and Privacy**
    - **Data Protection**: Employs strong encryption and secure authentication to protect biometric data and ensure compliance with data protection regulations.
    - **Privacy Compliance**: Stores only necessary biometric data, maintaining transparency

and respecting user privacy.

3.3.3. ECONOMICAL FEASIBILITY

Economic feasibility assesses the financial viability of implementing the face recognition-based attendance system, focusing on costs, savings, and overall return on investment (ROI).

- **Initial Investment**
  - **Hardware Costs**: Initial investment involves purchasing cameras and modern computing devices. Standard webcams and existing computing infrastructure can be utilized to minimize costs.
  - **Software Costs**: Python and its associated libraries (OpenCV, face_recognition) are open-source and free, eliminating licensing fees.
  - **Development Costs**: Development involves expenses related to system design, programming, and testing. Using open-source tools and libraries reduces these costs significantly.
- **Operational Costs**
  - **Maintenance**: Regular software updates and hardware maintenance incur low ongoing costs. These include routine checks and minor upgrades to ensure the system remains functional and up to date.
  - **Training**: Minimal training costs are required due to the system's user-friendly interface, allowing staff to quickly learn and operate the system.
- **Cost Savings**
  - **Labor Costs**: Automation of the attendance process reduces the need for manual tracking, leading to significant savings in labor costs.
  - **Resource Savings**: The system eliminates the need for physical resources like paper-based attendance logs, contributing to cost reductions in stationery and storage.
- **Return on Investment (ROI)**
  - **Quick ROI**: The efficiency gains from automated attendance tracking and the reduction in manual labor result in a quick return on investment. The initial costs are offset by the immediate savings in labor and resource expenses.
  - **Long-Term Benefits**: Over time, the system continues to provide financial benefits through sustained operational cost savings and enhanced productivity.

CHAPTER 4. SYSTEM ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

1. **Video Capture and Face Detection:**
   - **Requirement**: The system must capture video frames from the webcam in real-time.
   - **Implementation**: Utilize the OpenCV library (cv2.VideoCapture) to continuously capture video frames. Apply face detection using the face_recognition.face_locations function.
2. **Facial Recognition**:
   - **Requirement**: The system must recognize faces and compare them with pre-registered faces.
   - **Implementation**: Utilize the face_recognition library to encode and compare faces. Implement the findEncodings function to preprocess images and calculate facial encodings. Use these encodings to compare and identify faces in real-time.
3. **Attendance Marking:**
   - **Requirement**: The system must mark attendance by associating recognized faces with timestamps.
   - Implementation: Implement the markAttendance function to write the name, time, and **date** of recognized faces to a CSV file (Attendance.csv). Ensure that duplicate entries are not recorded for the same person during a session.
4. **Real-Time Display:**
   - **Requirement**: The system must display the video frames in real-time, with annotated bounding boxes and recognized names.
   - **Implementation**: Use the OpenCV library (cv2.imshow, cv2.rectangle, cv2.putText) to display the video frames. Annotate the frames with rectangles around detected faces and display the recognized names above the rectangles.
5. **Continuous Execution:**
   - **Requirement**: The system must operate continuously until manually terminated.
   - **Implementation**: Employ a continuous loop (while True) to ensure continuous video capture, face detection, and attendance marking. Allow the user to exit the system by pressing the 'q' key.
6. **File Handling**:
   - **Requirement**: The system must handle image files and attendance records.
   - **Implementation**: Utilize the os module to read image files from the 'student_images' directory. Use the csv module for reading and writing attendance records in the 'Attendance.csv' file.
7. **Time and Date Handling:**
   - **Requirement**: The system must record the time and date of each attendance entry.
   - **Implementation**: Use the datetime module to obtain the current time and date. Include this information when marking attendance entries in the CSV file.
8. **Dynamic Recognition and Updating:**
   - **Requirement**: The system should dynamically recognize and update faces during runtime.
   - **Implementation**: Ensure that the system can dynamically recognize and update faces by re-reading images and recalculating encodings during execution. This is important for scenarios where new individuals need to be added to the recognition system without stopping the application.

These functional requirements outline the core features and behaviors of the facial recognition-based attendance system based on the provided code. They serve as a foundation for the development and testing of the system.

4.2 NON-FUNCTIONAL REQUIREMENTS

1. **Performance**:
    - **Requirement**: The system must provide real-time face detection and recognition.
    - **Criteria**: The system should achieve a frame processing rate of at least 10 frames per second (FPS) to ensure timely attendance marking.
2. **Accuracy**:
    - **Requirement**: The facial recognition system must have a high accuracy rate.
    - **Criteria**: Achieve a recognition accuracy of at least 95% under varying lighting conditions and facial expressions.
3. **Scalability**:
    - **Requirement**: The system must handle a scalable number of pre-registered faces.
    - **Criteria**: The system should be able to efficiently recognize and mark attendance for up to 1000 pre-registered individuals without a significant decrease in performance.
4. **Usability**:
    - **Requirement**: The user interface should be intuitive for administrators.
    - **Criteria**: Administrators should be able to easily add, update, or remove pre-registered faces. The system should provide clear visual feedback during attendance marking.
5. **Security**:
    - **Requirement**: Ensure the security of attendance records and face encodings.
    - **Criteria**: Attendance records and face encodings should be stored securely. Implement encryption and access controls to prevent unauthorized access or tampering.
6. **Reliability**:
    - **Requirement**: The system should be reliable and operate without frequent failures.
    - **Criteria**: The system should be capable of continuous operation for at least 8 hours without unexpected crashes. In the event of a failure, it should gracefully recover without data loss.
7. **Portability**:
    - **Requirement**: The system should be portable across different environments.
    - **Criteria**: The system should be easily deployable on different operating systems (Windows, Linux) and hardware configurations without significant modifications.
8. **Maintainability**:
    - **Requirement**: The system should be easy to maintain and update.
    - **Criteria**: Code should be well-documented, and modular, allowing for straightforward updates or enhancements. Regular updates should be possible without disrupting normal system operation.
9. **Interoperability**:
    - **Requirement**: The system should be interoperable with common hardware and software.
    - **Criteria**: The system should work seamlessly with commonly used webcams and adhere to standard data exchange formats for interoperability with other systems or databases.
10. **Ethical Considerations**:
    - **Requirement**: The system should adhere to ethical considerations regarding privacy and consent.
    - **Criteria**: Ensure that the system complies with privacy laws and regulations. Users should be informed about the purpose of data collection, and consent should be obtained.

11. **Response Time**:
    - **Requirement**: The system must provide quick responses to user interactions.
    - **Criteria**: The user interface should have a responsive design, and interactions such as updating pre-registered faces or marking attendance should have minimal latency.

These non-functional requirements ensure that the facial recognition-based attendance system not only performs its primary functions effectively but also meets key performance, security, and usability criteria. They guide the system's development and provide a basis for evaluation during testing and deployment.

4.3 REQUIREMENT SPECIFICATION

Purpose

The purpose of this document is to provide a comprehensive specification for the development of the Facial Recognition-Based Attendance System.

Scope

The system aims to streamline attendance tracking using facial recognition technology, catering to educational institutions and organizations.

4.3.1 Software Requirement Specification

The software requirements define the functionalities and constraints of the system.

1. **Functional Requirements**

   Video Capture and Face Detection:
   - The system shall capture video frames from the webcam in real-time.
   - Face detection algorithms shall identify faces in each frame.

   Facial Recognition:
   - The system shall encode and compare detected faces with pre-registered faces.
   - A database shall store facial encodings for pre-registered individuals.
   - Dynamic recognition and updating of faces during runtime shall be supported.

   Attendance Marking:
   - The system shall mark attendance by associating recognized faces with timestamps.
   - Attendance records, including name, time, and date, shall be saved to a CSV file.

   Real-Time Display:
   - Video frames shall be displayed in real-time with annotated bounding boxes around recognized faces.
   - Recognized names shall be displayed above the bounding boxes.

2. **Non-Functional Requirements**

   Performance:
   - The system shall achieve a frame processing rate of at least 10 frames per second.
   - Facial recognition accuracy shall be at least 95%.

   Scalability:
   - The system shall efficiently recognize and mark attendance for up to 1000 pre-registered individuals.

   Usability:
   - The user interface shall be intuitive for administrators to manage pre-registered faces.

   Security:
   - Attendance records and face encodings shall be stored securely with encryption.
   - Access controls shall be implemented to restrict unauthorized access.

   Reliability:
   - The system shall operate continuously for at least 8 hours without unexpected

crashes.

4.3.2 Implementation Tool & Language

The Facial Recognition Based Attendance System (FRBAS) is implemented using a combination of tools and programming languages designed to create a robust, efficient, and user-friendly application. The key tools and languages utilized in the development of this system are as follows:

1. Python

Python is the primary programming language used for the implementation of the FRBAS. Known for its simplicity, readability, and extensive library support, Python is well-suited for developing applications that involve machine learning, computer vision, and GUI development. The following libraries and frameworks in Python are crucial for the system:

- **OpenCV (cv2)**: OpenCV (Open Source Computer Vision Library) is an open-source library that provides a wide range of computer vision and image processing functions. In the FRBAS, OpenCV is used for:
    - Capturing live video feed from the camera.
    - Performing face detection using Haar cascades or other models.
    - Image preprocessing tasks like resizing and converting colour spaces.
    - Displaying video frames and processed images.
- **face_recognition**: The face_recognition library is built on top of dlib and provides simple and powerful tools for face recognition tasks. It is used for:
    - Encoding facial features from student images.
    - Comparing live video frames with stored facial encodings to recognize students.
    - Managing facial data efficiently.
- **tkinter**: tkinter is Python's standard GUI library, which is used to create graphical interfaces for the application. In the FRBAS, tkinter is employed for:
    - Designing the main application window and user interface elements such as buttons, labels, and entry fields.
    - Handling user interactions for tasks like registering new students, taking attendance, and viewing attendance records.
- **PIL (Python Imaging Library) / Pillow**: Pillow is a fork of the original PIL and provides tools for opening, manipulating, and saving many different image file formats. It is used in the FRBAS for:
    - Loading and displaying profile images of students.
    - Resizing images to fit within the application's GUI.
- **csv**: The csv module is used for reading and writing CSV files, which are used to store:
    - Student registration data, including names and enrollment numbers.
    - Attendance records with timestamps and other relevant details.

2. Libraries and Modules

The FRBAS relies on several other libraries and modules to enhance functionality and performance:

- **os**: Used for interacting with the operating system, such as managing file paths and directories.
- **shutil**: Used for high-level file operations like copying and removing files.
- **threading**: Used to handle concurrent operations, such as running the facial recognition process in a separate thread to keep the GUI responsive.
- **datetime**: Used for handling date and time operations, including timestamping attendance records.
- **numpy**: Used for numerical operations and handling data arrays, particularly useful in processing image data and computing face distances.

3. Development Environment

The development environment for the FRBAS includes:

- **Integrated Development Environment (IDE)**: Popular Python IDEs like PyCharm, Visual

Studio Code, or Jupyter Notebooks can be used for writing and testing code.
- **Version Control System**: Git is used for version control, enabling collaborative development and tracking changes.
- **Dependency Management**: Pip is used to manage Python packages and dependencies.

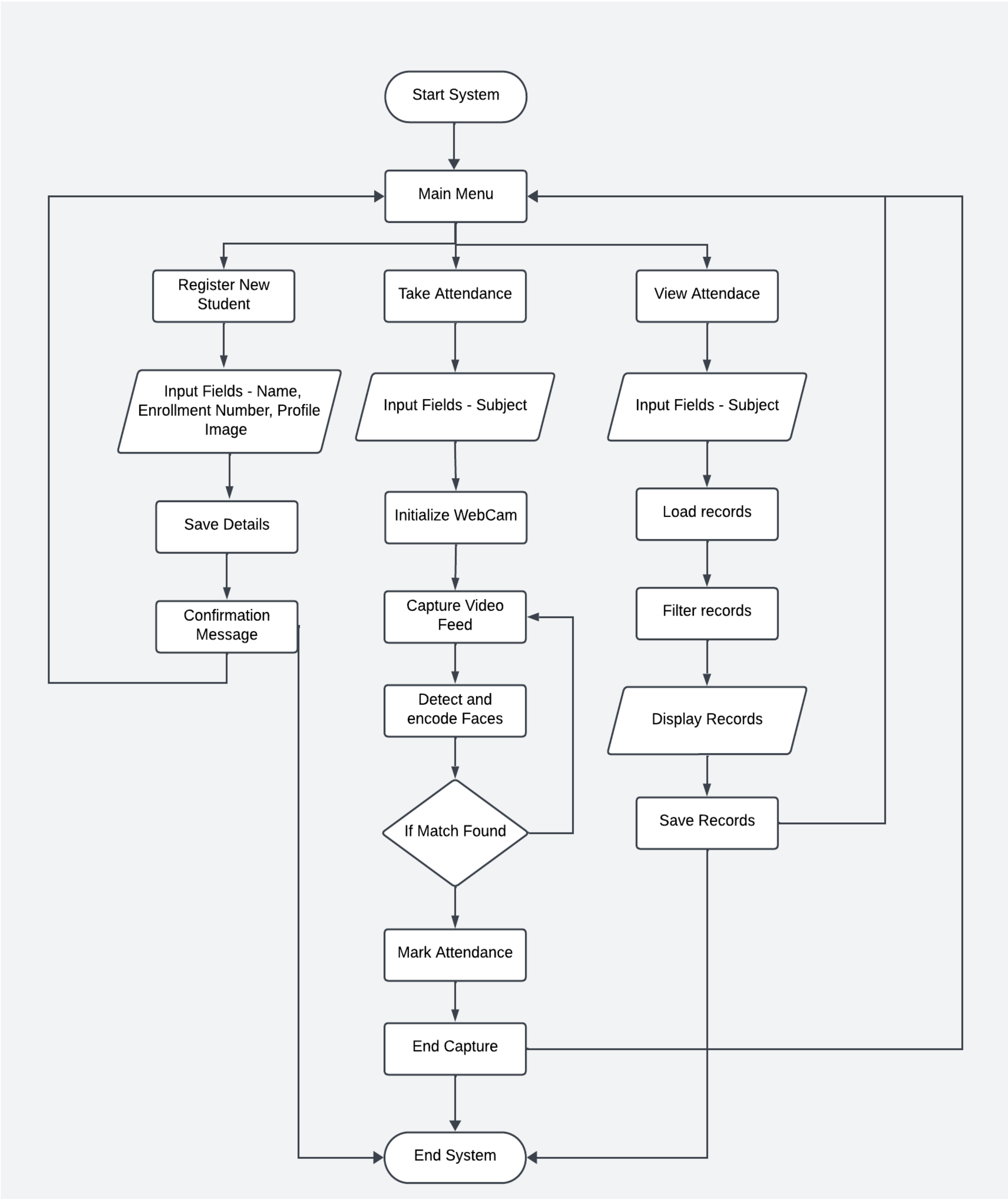
Implementation Overview

The FRBAS is implemented as follows:

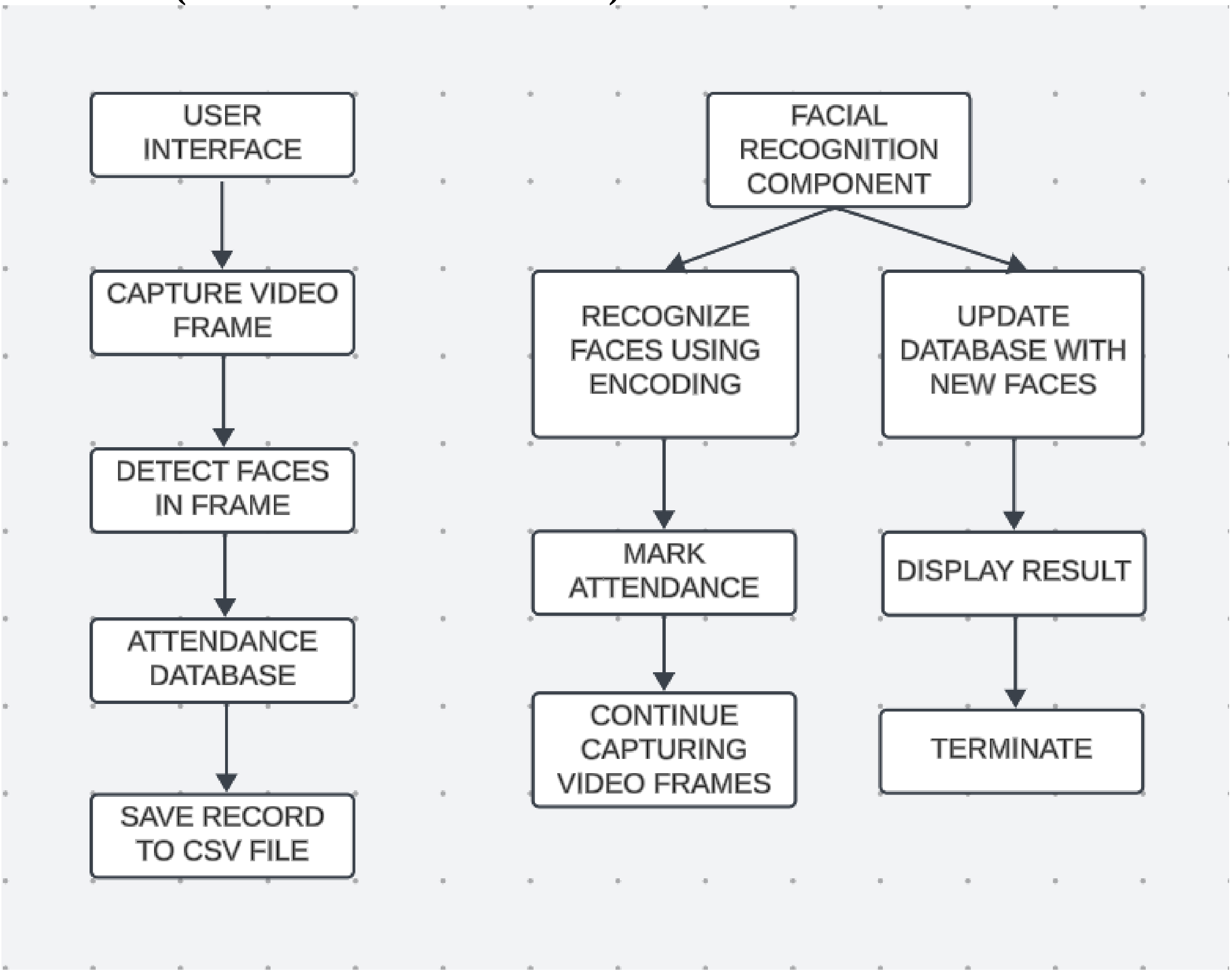
1. **Registration Module (register.py)**:
    - Provides a GUI for entering student details and capturing profile images.
    - Stores student data in a CSV file and saves images in a designated directory.
2. **Attendance Module (attendance.py)**:
    - Captures live video feed from the camera.
    - Detects and recognizes faces using the face_recognition library.
    - Marks attendance by recording recognized faces along with timestamps in a CSV file.
3. **View Attendance Module (viewAttendance.py)**:
    - Provides a GUI for viewing attendance records.
    - Allows filtering and displaying attendance data based on subjects and other criteria.
    - Supports exporting attendance records to CSV files.
4. **Main Application (main.py)**:
    - Integrates all modules and provides the main user interface.
    - Allows users to navigate between registration, attendance, and viewing attendance functionalities.

    By leveraging these tools and languages, the FRBAS offers a comprehensive and efficient solution for automating attendance management in educational institutions, ensuring accuracy, compliance, and ease of use.

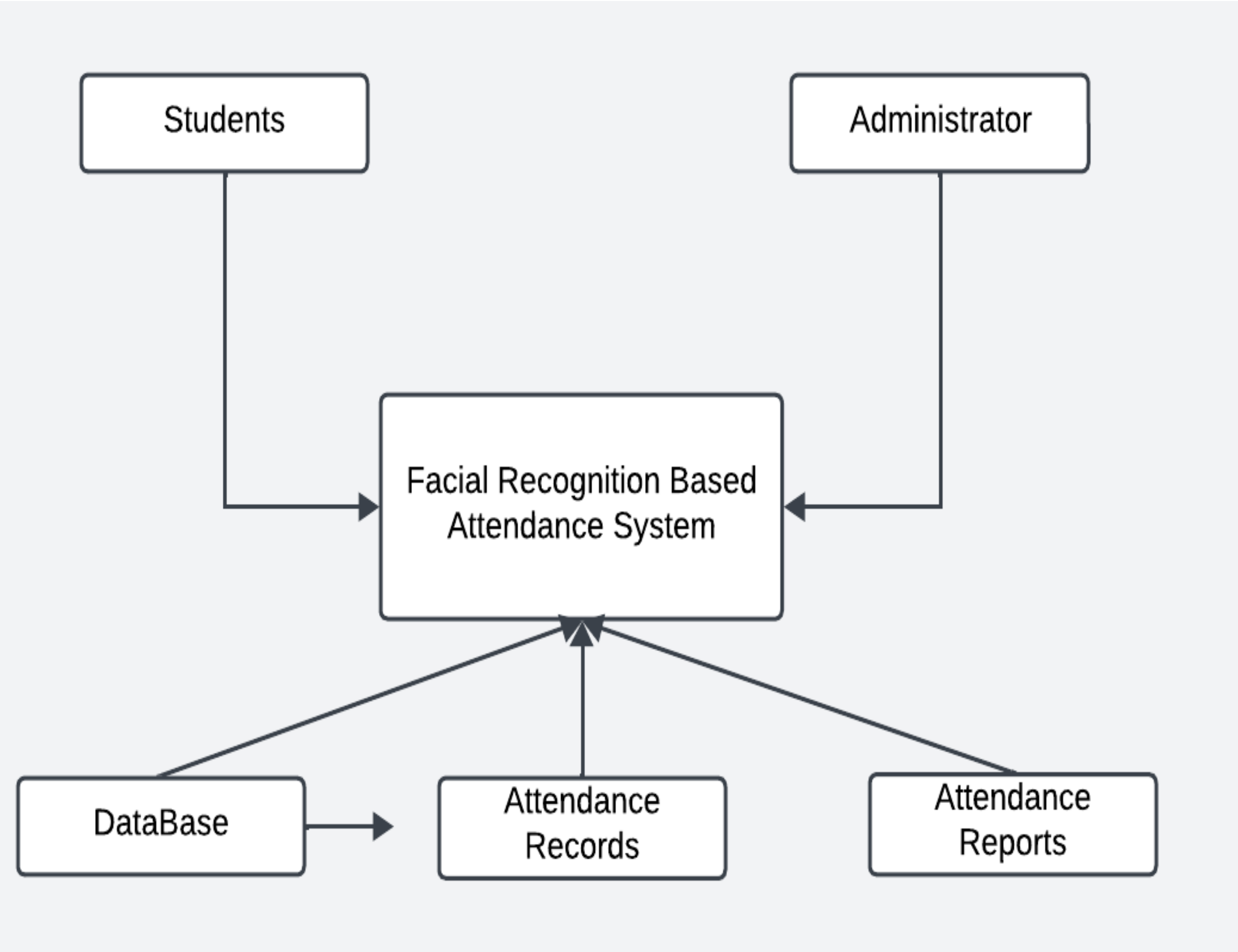
4.4 SYSTEM FLOWCHART



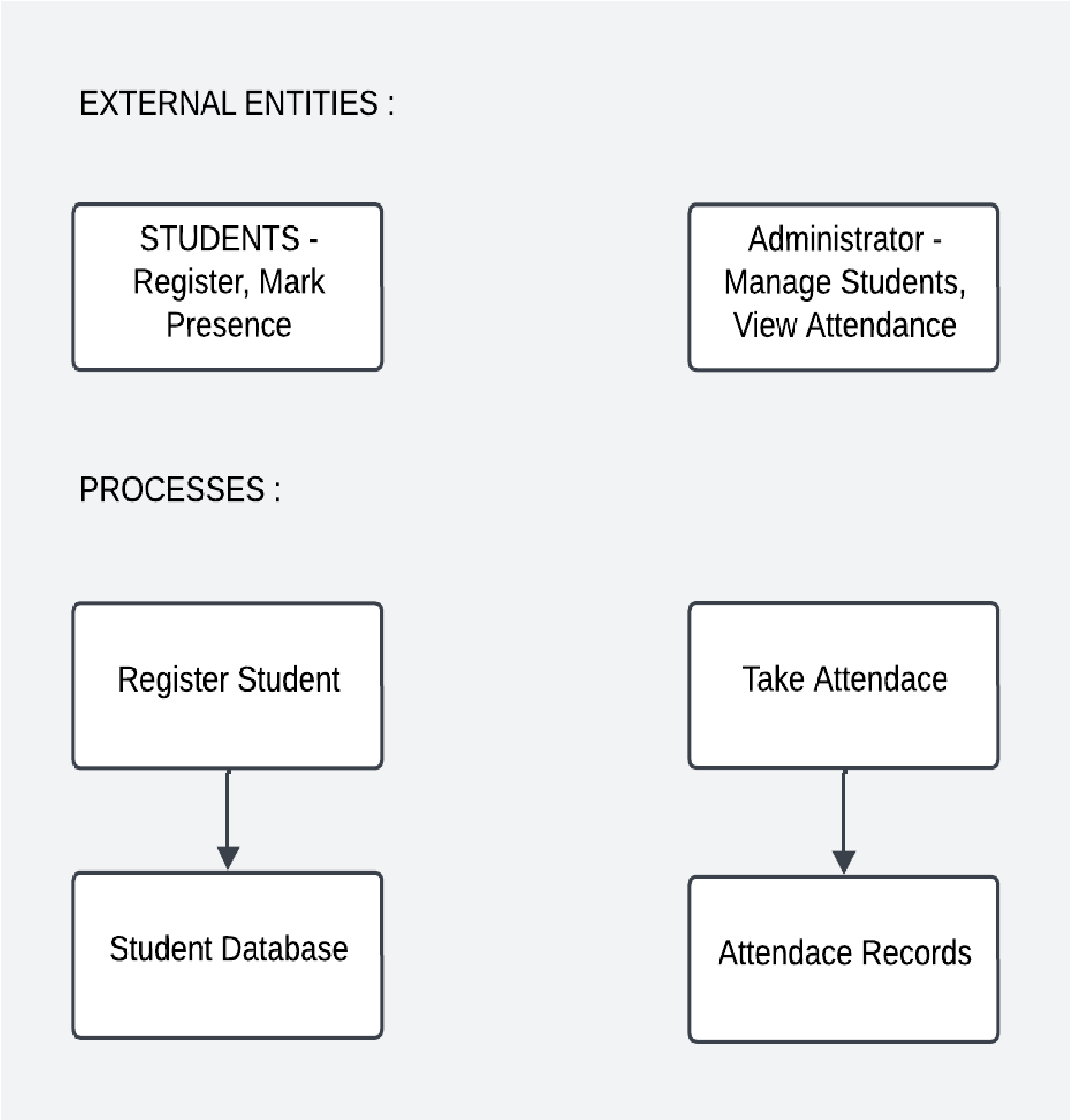
4.5 DFD (DATA FLOW DIAGRAMS)



4.5.1 LEVEL 0 DFD



4.5.2 LEVEL 1 DFD



4.5 SYSTEM DESIGN

System design involves specifying the architecture, components, modules, data, interfaces, and algorithms for a system to satisfy specified requirements.

1. **System Architecture:**
   Components:
   - User Interface: Allows interaction with the system.
   - Capture Module: Captures video frames from the webcam.
   - Detection Module: Identifies faces in the captured frames.
   - Recognition Module: Encodes and compares detected faces with pre-registered faces.
   - Attendance Module: Marks attendance updates the database, and saves records to a CSV file.
   - Display Module: Displays real-time results, such as recognized names.

   Communication Channels:
   - Communication between modules occurs through well-defined interfaces.
   - Data flow is managed to ensure efficient processing.
2. **Data Storage and Handling:**
   Attendance Database:
   - Stores attendance records, including names, timestamps, and other relevant information.
   - Utilizes a secure and efficient database management system.

   CSV File:
   - Attendance records are also saved to a CSV file for additional data backup and analysis.
3. **Facial Recognition Algorithms:**
   Encoding and Comparison:
   - Utilizes facial recognition algorithms to encode and compare facial features.
   - Dynamically updates the database with new faces during runtime.
4. **User Interface:**
   Intuitive Design:
   - Provides a user-friendly interface for administrators.
   - Allows easy addition, removal, or updating of pre-registered faces.
5. **Continuous Execution:**
   Infinite Loop:
   - Implements a continuous loop for video frame capture, face detection, and attendance marking until the user terminates the application.
6. **Real-Time Display:**
   Visual Feedback:
   - Displays video frames in real-time with annotated bounding boxes around recognized faces.
   - Presents recognized names above the rectangles for visual feedback.
7. **Security Measures:**
   Data Encryption:
   - Implements encryption mechanisms to secure attendance records and face encodings.

   Access Controls:
   - Enforces access controls to restrict unauthorized access to sensitive data.
8. **Performance Considerations:**
   Frame Processing Rate:
   - Strives for a frame processing rate of at least 10 frames per second for real-time performance.

Accuracy:

- Ensures facial recognition accuracy of at least 95% under varying conditions.

9. **Scalability:**
   Database Scalability:
   - Designs the attendance database to handle a scalable number of pre-registered individuals.
10. **Interoperability:**
    Compatibility:
    - Ensures compatibility with common webcams and adheres to standard data exchange formats.
11. **Ethical Considerations:**
    Privacy Measures:
    - Complies with privacy laws and regulations.
    - Incorporates user consent mechanisms.
12. **Maintainability:**
    Modular Code:
    - Writes modular and well-documented code for ease of maintenance.
    - Allows for straightforward updates and enhancements.
13. **Portability:**
    Cross-Platform Compatibility:
    - Designs the system for deployment on different operating systems (e.g., Windows, Linux).
14. **Testing and Quality Assurance:**
    Testing Procedures:
    - Develops comprehensive test cases for functionality, security, and performance.
    - Conducts thorough testing at each development stage.
15. **Documentation:**
    Comprehensive Documentation:
    - Prepares detailed documentation covering system architecture, modules, algorithms, and usage guidelines.

This outline provides a systematic approach to system design for a facial recognition-based attendance system. The actual implementation would involve coding and integrating these design elements to create a functional and reliable system.

4.6 SYSTEM TESTING

System testing is a critical phase in the software development life cycle where the entire system is tested as a whole. The goal is to ensure that all components work seamlessly together, meet the specified requirements, and perform as expected. Here's an overview of the key aspects of system testing for a facial recognition-based attendance system:

1. **Types of System Testing:**
   Functional Testing:
   - **Attendance Marking:** Verify that the system correctly marks attendance for recognized faces.
   - **Dynamic Recognition:** Test the system's ability to dynamically recognize and update faces during runtime.
   - **Database Operations:** Ensure proper updating and saving of attendance records to the database and CSV file.
   - **User Interface:** Check the user interface for intuitive interactions and functionality.

Performance Testing:

- **Frame Processing Rate:** Evaluate if the system achieves the specified frame processing rate (e.g., at least 10 frames per second).
- **Scalability:** Assess the system's performance with varying numbers of pre-registered **individuals.**

Security Testing:

- **Data Encryption:** Verify that attendance records and face encodings are stored securely through encryption.
- **Access Controls:** Test the access controls to prevent unauthorized access to sensitive data.

Usability Testing:

- **User Interface Design:** Assess the usability of the user interface for administrators to add, update, or remove pre-registered faces.
- **Feedback Mechanisms:** Check if the system provides clear feedback to users during attendance marking and termination.

Reliability Testing:

- **Continuous Operation:** Evaluate the system's ability to operate continuously for the specified duration without unexpected crashes.
- **Error Handling:** Test how well the system handles unexpected errors and recovers gracefully.

Compatibility Testing:

- **Webcam Compatibility:** Ensure that the system works seamlessly with common webcams.
- **Cross-Platform Compatibility:** Test the system's deployment on different operating systems (e.g., Windows, Linux).

Ethical Considerations:

- **Privacy Compliance:** Verify that the system complies with privacy laws and regulations.
- **User Consent:** Ensure that appropriate mechanisms are in place to obtain user consent for data collection.

2. **Testing Scenarios:**
   Normal Operation: Simulate normal scenarios where individuals are recognized, attendance is marked, and records are updated.
   Error Scenarios: Introduce errors, such as incorrect face recognition or database errors, to assess how the system handles and recovers from them.
   Performance Scenarios: Test the system's performance with different numbers of individuals, evaluating scalability and responsiveness.
   Security Scenarios: Attempt unauthorized access to sensitive data and assess the system's response.
3. **Testing Tools:**
   Testing Frameworks: Utilize testing frameworks (e.g., PyTest for Python) to automate and streamline the testing process.
   Simulation Tools: Use simulation tools to mimic real-time scenarios and assess system behavior under various conditions.
4. **Documentation:**
   Test Plans: Develop comprehensive test plans outlining testing objectives, scenarios, and acceptance criteria.
   Test Cases: Create detailed test cases covering functional, performance, security, and usability aspects.

Bug Reports: Document and prioritize any bugs or issues discovered during testing.

5. **User Acceptance Testing (UAT):**
   Conduct UAT with actual users or stakeholders to validate that the system meets their expectations and requirements.
6. **Regression Testing:**
   Perform regression testing to ensure that new updates or changes do not negatively impact existing functionalities.
7. **Performance Metrics:**
   Measure and document key performance metrics, such as frame processing rate, recognition accuracy, and response time.
8. **Feedback and Iteration:**
   - Collect feedback from testers and users
   - Iterate on the system based on feedback and make necessary improvements.

System testing ensures that the facial recognition-based attendance system is reliable, performs well, and meets the specified requirements in a variety of scenarios. It plays a crucial role in delivering a high-quality and robust software product.

4.7 SYSTEM IMPLEMENTATION

System implementation involves translating the design specifications into actual code and deploying the software. Here's a step-by-step guide for implementing a facial recognition-based attendance system:

1. **Coding:**
   Set Up Development Environment:
   - Ensure that the development environment is set up with the necessary tools and libraries (OpenCV, face_recognition, etc.).

   Module-wise Coding:
   - Implement code for each module identified during system design (Capture Module,
   - Detection Module, Recognition Module, etc.).
   - Follow best coding practices, maintain code modularization, and include comments for clarity.

   Integration:
   - Integrate individual modules into a cohesive system.
   - Ensure proper communication and data flow between modules.
2. **Database Implementation:**
   Design Database Schema:
   - Create a database schema that includes tables for storing attendance records.

   Implement Database Operations:
   - Write code to handle database operations, such as saving attendance records and updating the database with new faces.

   Test Database Connectivity:
   - Verify that the system can successfully connect to and interact with the database.
3. **User Interface Implementation:**
   Design User Interface:
   - Implement the user interface for administrators to interact with the system.
   - Design forms or screens for adding, updating, or removing pre-registered faces.

   User Input Handling:
   - Write code to handle user input and trigger relevant system functionalities.

4. **Facial Recognition Implementation:**
   Integration with Libraries:
   - Integrate facial recognition libraries (e.g., face_recognition) into the system.
   - Utilize pre-trained models or train custom models based on requirements.

   Dynamic Recognition:
   - Implement code for dynamic recognition and updating of faces during runtime.
5. **Real-Time Display Implementation:**
   Display Video Frames:
   - Write code to capture video frames from the webcam and display them in real-time.

   Annotate Faces:
   - Implement code to annotate recognized faces with bounding boxes and display names in real-time.
6. **Security Implementation:**
   Data Encryption:
   - Implement encryption mechanisms for sensitive data, such as attendance records and face encodings.

   Access Controls:
   - Write code to enforce access controls and restrict unauthorized access to sensitive functionalities.
7. **Performance Optimization:**
   Frame Processing Rate:
   - Optimize code to achieve the specified frame processing rate.

   Algorithm Efficiency:
   - Optimize facial recognition algorithms for better performance.
8. **Testing:**
   Unit Testing:
   - Conduct unit testing for individual modules to ensure they function as expected.

   Integration Testing:
   - Test the integration of all modules to verify proper communication and data flow.

   System Testing:
   - Perform comprehensive system testing based on the previously defined test plans and scenarios.

   User Acceptance Testing (UAT):
   - Involve actual users or stakeholders to conduct UAT and gather feedback.
9. **Documentation:**
   Code Documentation:
   - Provide detailed documentation for the code, including comments, function descriptions, and usage guidelines.

   User Manual:
   - Create a user manual explaining how to use the system, especially for administrators.
10. **Deployment:**
    Prepare for Deployment:
    - Ensure all dependencies and libraries are installed on the deployment environment.

    Deploy the System:
    - Transfer the compiled code and necessary files to the deployment environment.

    Configuration:
    - Configure the system for the specific deployment environment, including database connection settings.
11. **Monitoring and Maintenance:**

Monitoring:

- Implement monitoring mechanisms to track system performance and detect any anomalies.

Bug Fixes and Updates:

- Address any bugs or issues reported during testing or after deployment.
- Provide updates or patches as needed.

12. **User Training:**

    Administrator Training:

    - Train administrators on how to use the system effectively.
    - Provide guidance on troubleshooting common issues.

13. **Feedback and Iteration:**

    Collect Feedback:

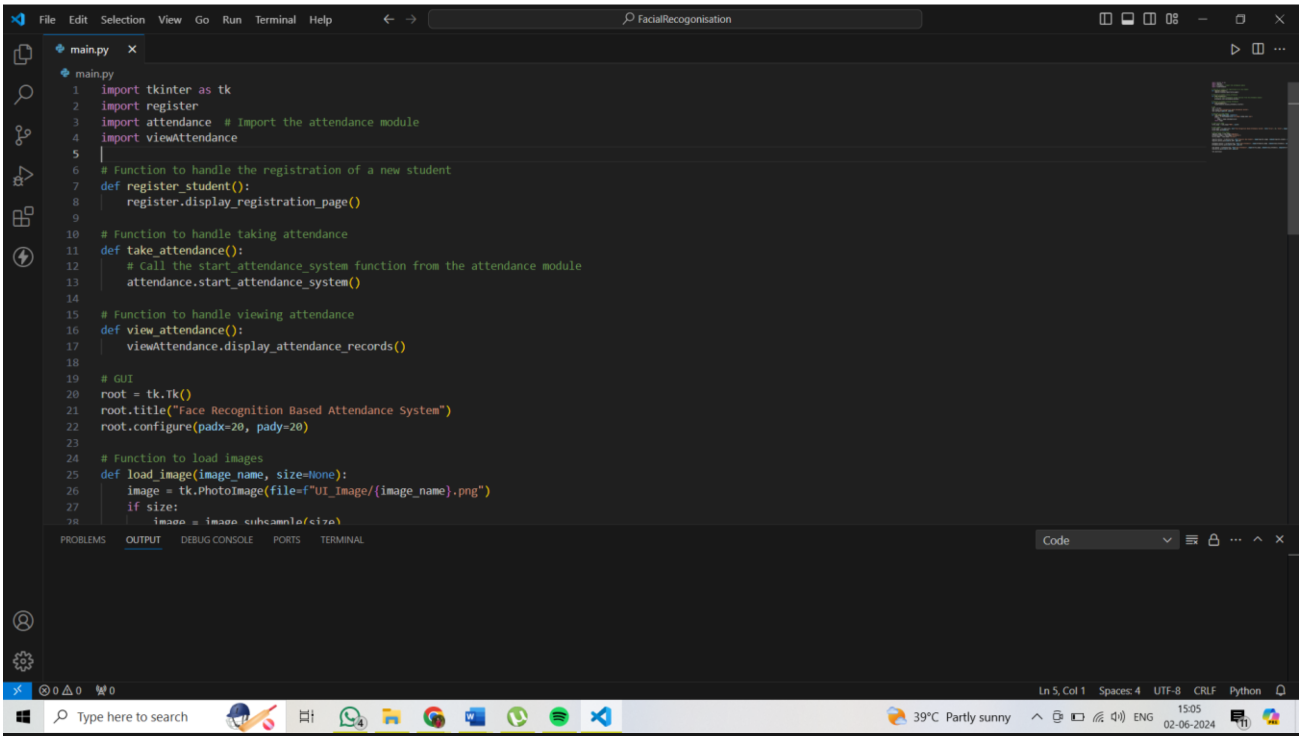
    - Gather feedback from users, administrators, and stakeholders.

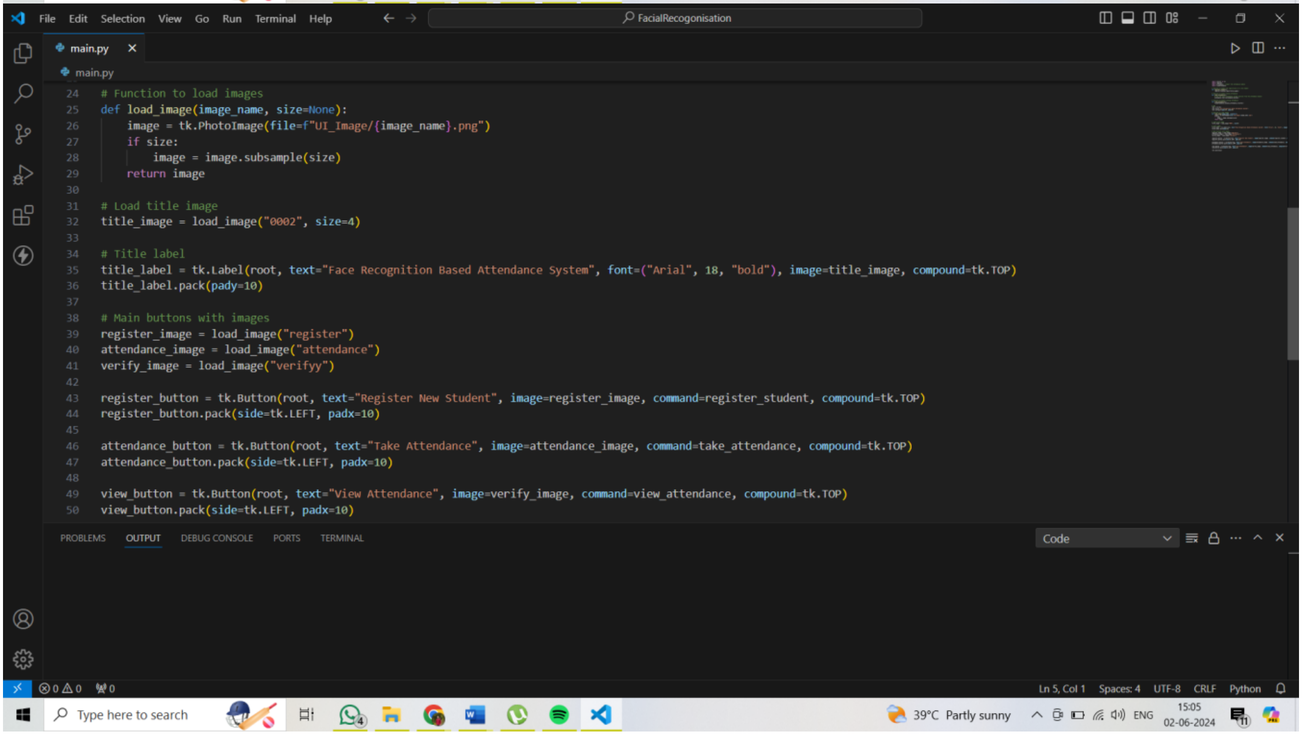
    Iterate and Improve:

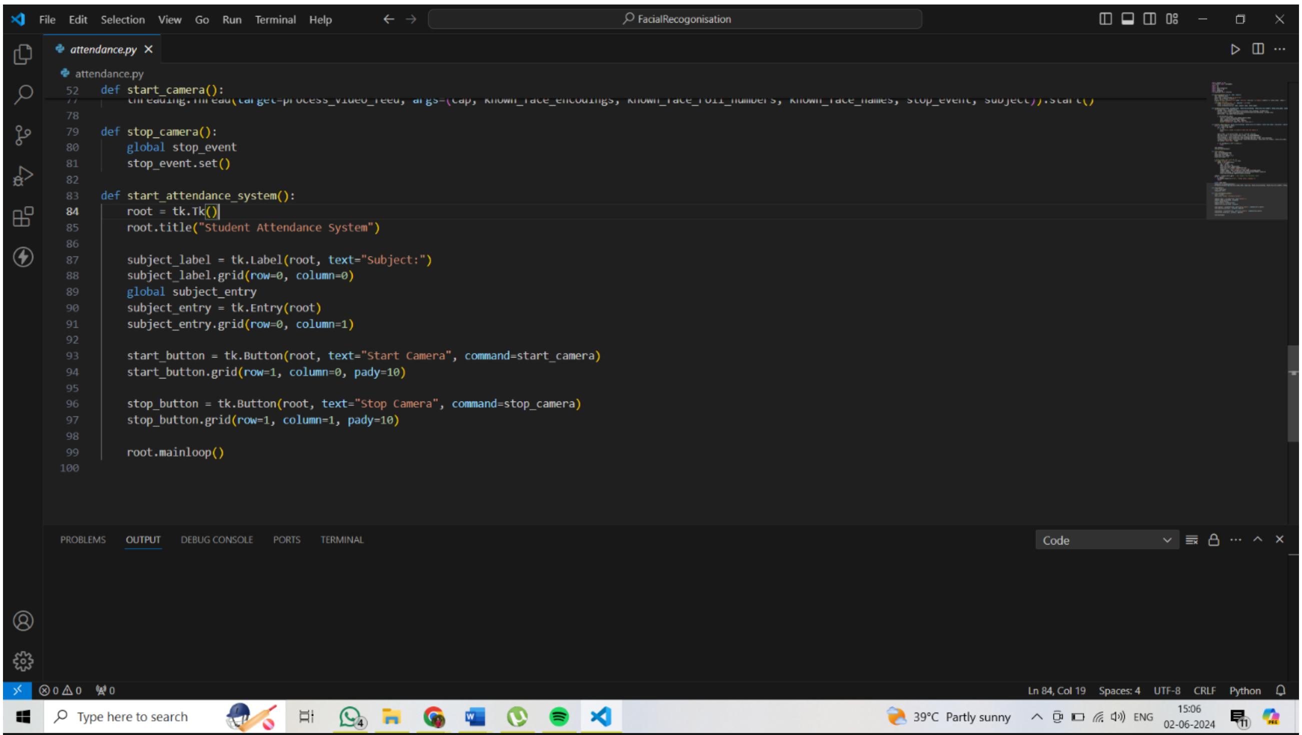
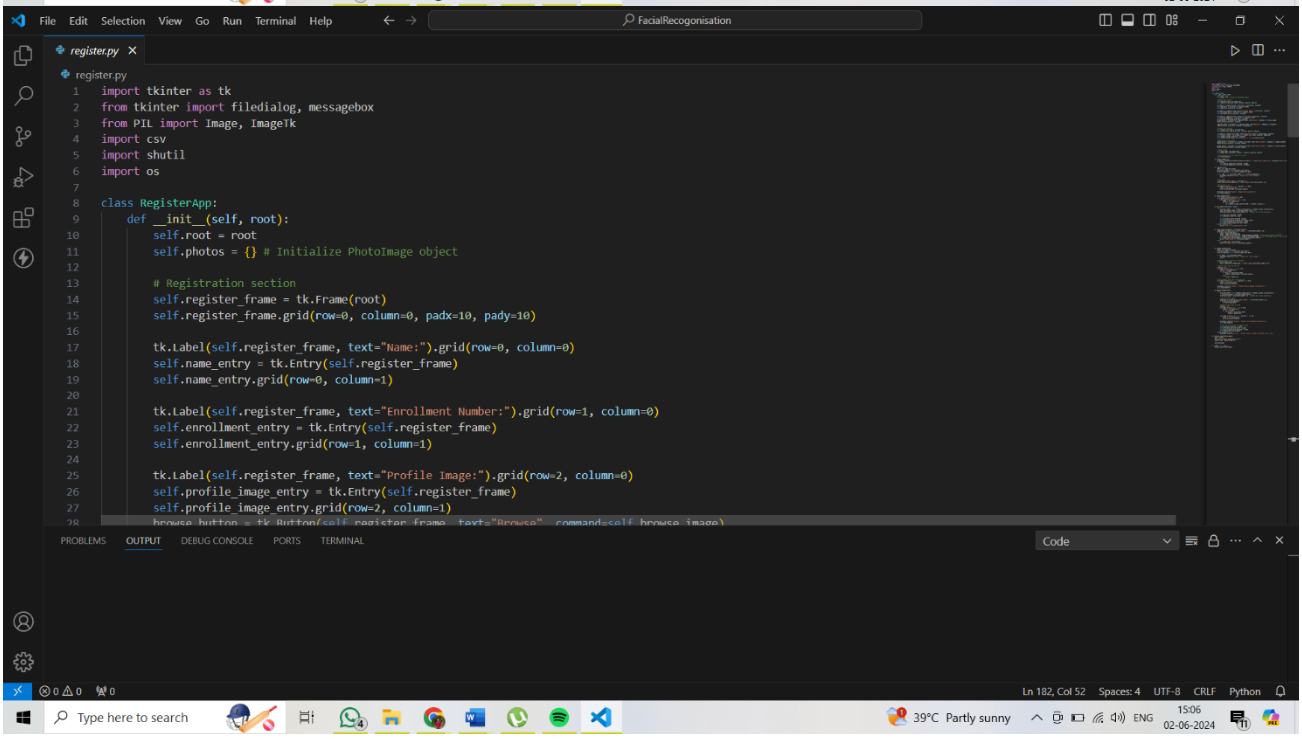
    - Use feedback to make necessary improvements and iterate on the system.

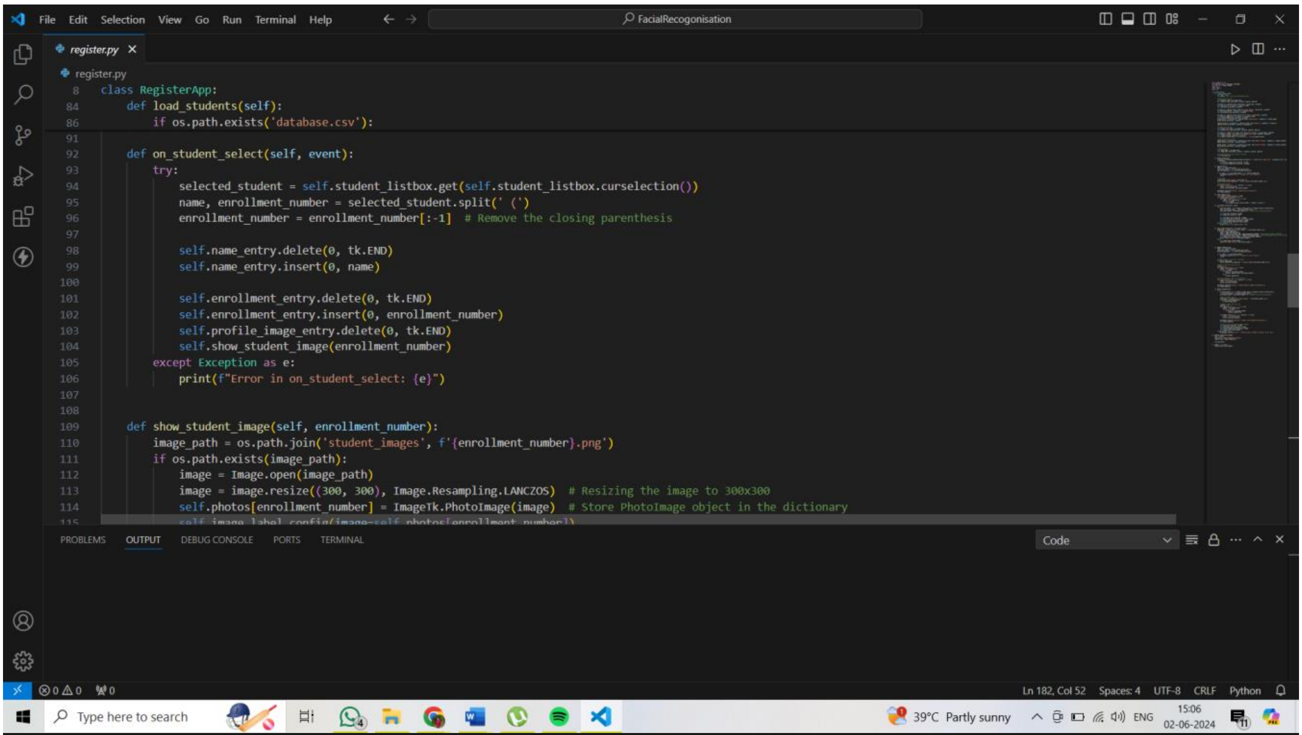
By following these steps, you can successfully implement and deploy a facial recognition-based attendance system, ensuring that it meets the specified requirements and functions reliably in real-world scenarios.

4.9 SCREENSHOTS OF THE PROJECT






File Edit Selection View Go Run Terminal Help
FacialRecogonisation
attendance.py
def start_camera():
def stop_camera():
global stop_event
stop_event.set()
def start_attendance_system():
root = tk.Tk()
root.title("Student Attendance System")
subject_label = tk.Label(root, text="Subject:")
subject_label.grid(row=0, column=0)
global subject_entry
subject_entry = tk.Entry(root)
subject_entry.grid(row=0, column=1)
start_button = tk.Button(root, text="Start Camera", command=start_camera)
start_button.grid(row=1, column=0, pady=10)
stop_button = tk.Button(root, text="Stop Camera", command=stop_camera)
stop_button.grid(row=1, column=1, pady=10)
root.mainloop()
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
Code
Ln 84, Col 19 Spaces: 4 UTF-8 CRLF Python
Type here to search
39°C Partly sunny
ENG
15:06
02-06-2024

File Edit Selection View Go Run Terminal Help
FacialRecogonisation
register.py
import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk
import csv
import shutil
import os
class RegisterApp:
def __init__(self, root):
self.root = root
self.photos = {} # Initialize PhotoImage object
# Registration section
self.register_frame = tk.Frame(root)
self.register_frame.grid(row=0, column=0, padx=10, pady=10)
tk.Label(self.register_frame, text="Name:").grid(row=0, column=0)
self.name_entry = tk.Entry(self.register_frame)
self.name_entry.grid(row=0, column=1)
tk.Label(self.register_frame, text="Enrollment Number:").grid(row=1, column=0)
self.enrollment_entry = tk.Entry(self.register_frame)
self.enrollment_entry.grid(row=1, column=1)
tk.Label(self.register_frame, text="Profile Image:").grid(row=2, column=0)
self.profile_image_entry = tk.Entry(self.register_frame)
self.profile_image_entry.grid(row=2, column=1)
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
Code
Ln 182, Col 52 Spaces: 4 UTF-8 CRLF Python
Type here to search
39°C Partly sunny
ENG
15:06
02-06-2024

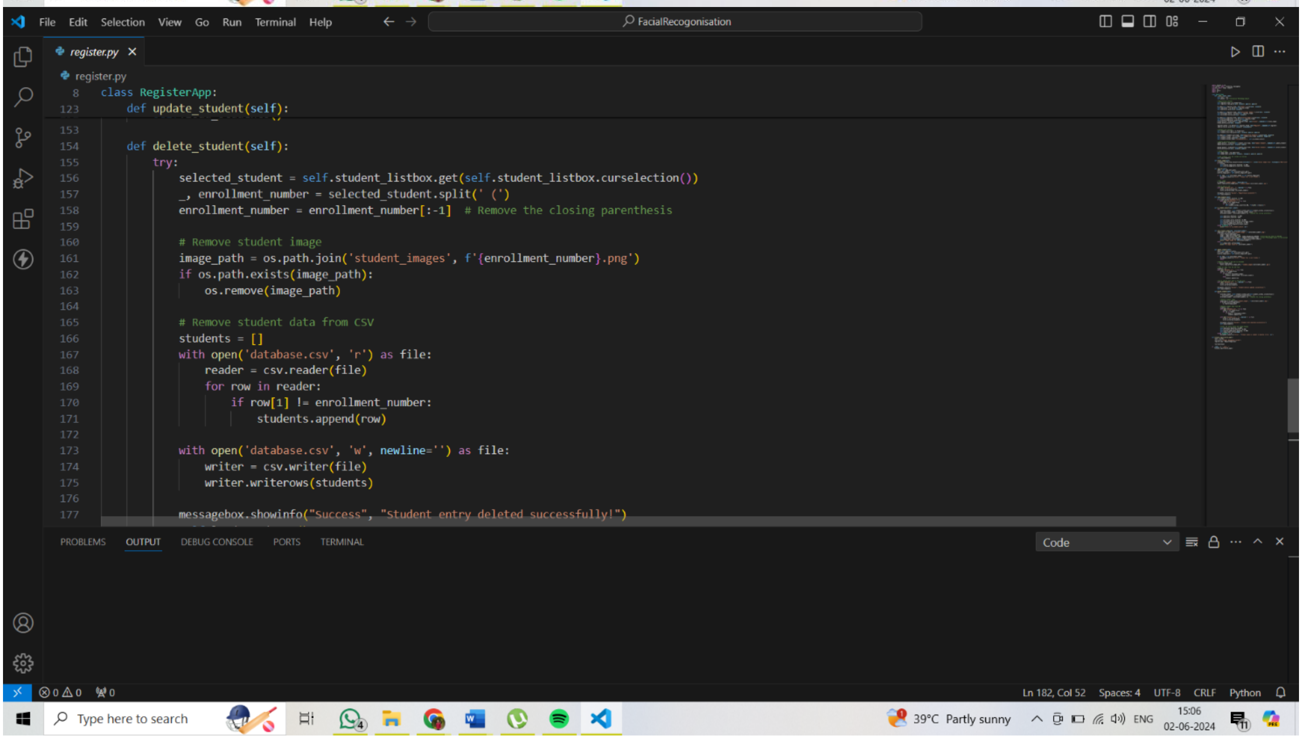

File Edit Selection View Go Run Terminal Help
FacialRecogonisation
register.py
class RegisterApp:
    def load_students(self):
        if os.path.exists('database.csv'):

    def on_student_select(self, event):
        try:
            selected_student = self.student_listbox.get(self.student_listbox.curselection())
            name, enrollment_number = selected_student.split(' (')
            enrollment_number = enrollment_number[:-1] # Remove the closing parenthesis

            self.name_entry.delete(0, tk.END)
            self.name_entry.insert(0, name)

            self.enrollment_entry.delete(0, tk.END)
            self.enrollment_entry.insert(0, enrollment_number)
            self.profile_image_entry.delete(0, tk.END)
            self.show_student_image(enrollment_number)
        except Exception as e:
            print(f"Error in on_student_select: {e}")

    def show_student_image(self, enrollment_number):
        image_path = os.path.join('student_images', f'{enrollment_number}.png')
        if os.path.exists(image_path):
            image = Image.open(image_path)
            image = image.resize((300, 300), Image.Resampling.LANCZOS) # Resizing the image to 300x300
            self.photos[enrollment_number] = ImageTk.PhotoImage(image) # Store PhotoImage object in the dictionary
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
Code
Ln 182, Col 52 Spaces: 4 UTF-8 CRLF Python
Type here to search
39°C Partly sunny
ENG
15:06
02-06-2024


File Edit Selection View Go Run Terminal Help
FacialRecogonisation
register.py
class RegisterApp:
    def update_student(self):

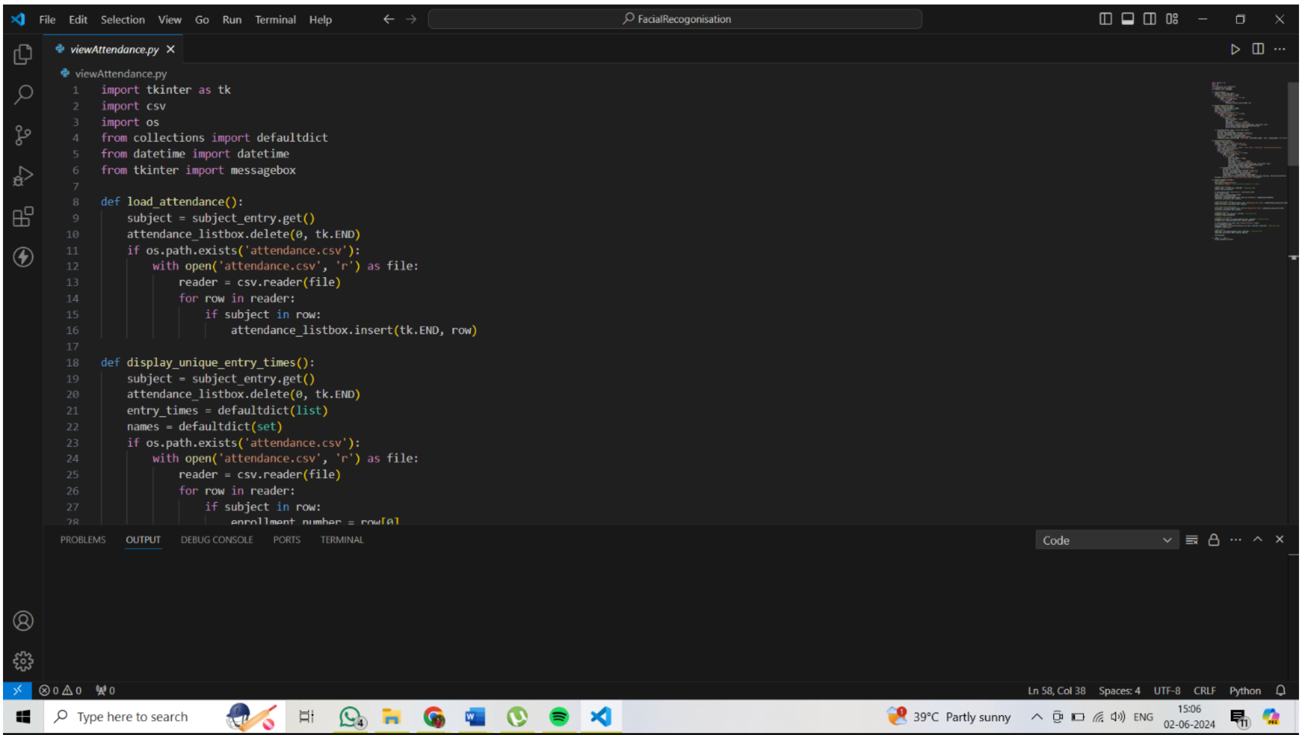
    def delete_student(self):
        try:
            selected_student = self.student_listbox.get(self.student_listbox.curselection())
            _, enrollment_number = selected_student.split(' (')
            enrollment_number = enrollment_number[:-1] # Remove the closing parenthesis

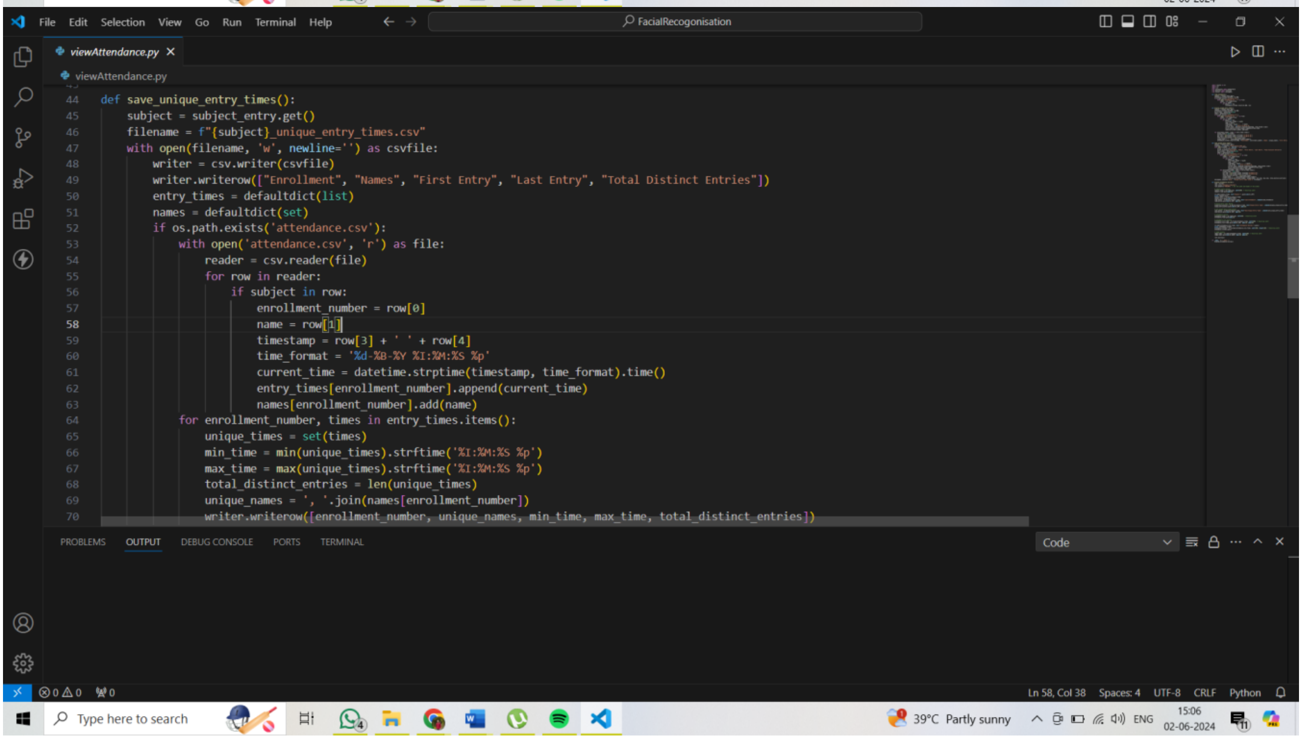
            # Remove student image
            image_path = os.path.join('student_images', f'{enrollment_number}.png')
            if os.path.exists(image_path):
                os.remove(image_path)

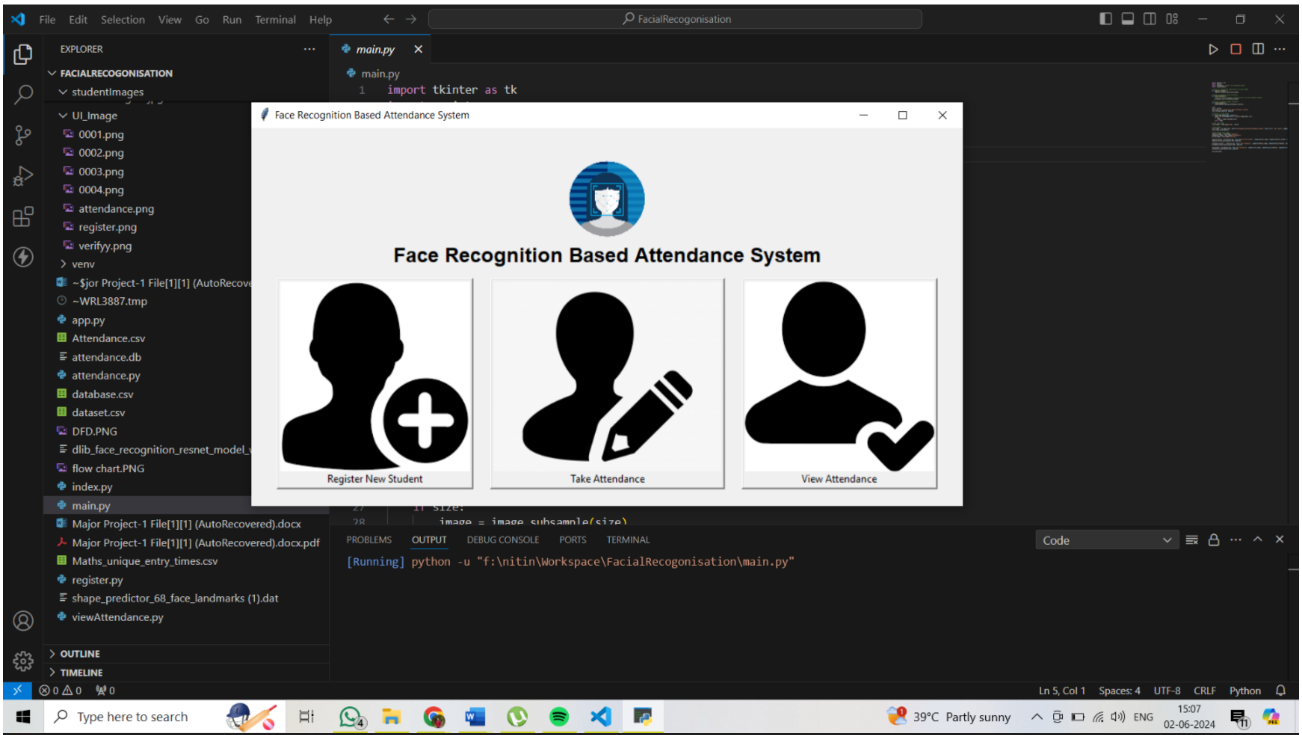
            # Remove student data from CSV
            students = []
            with open('database.csv', 'r') as file:
                reader = csv.reader(file)
                for row in reader:
                    if row[1] != enrollment_number:
                        students.append(row)

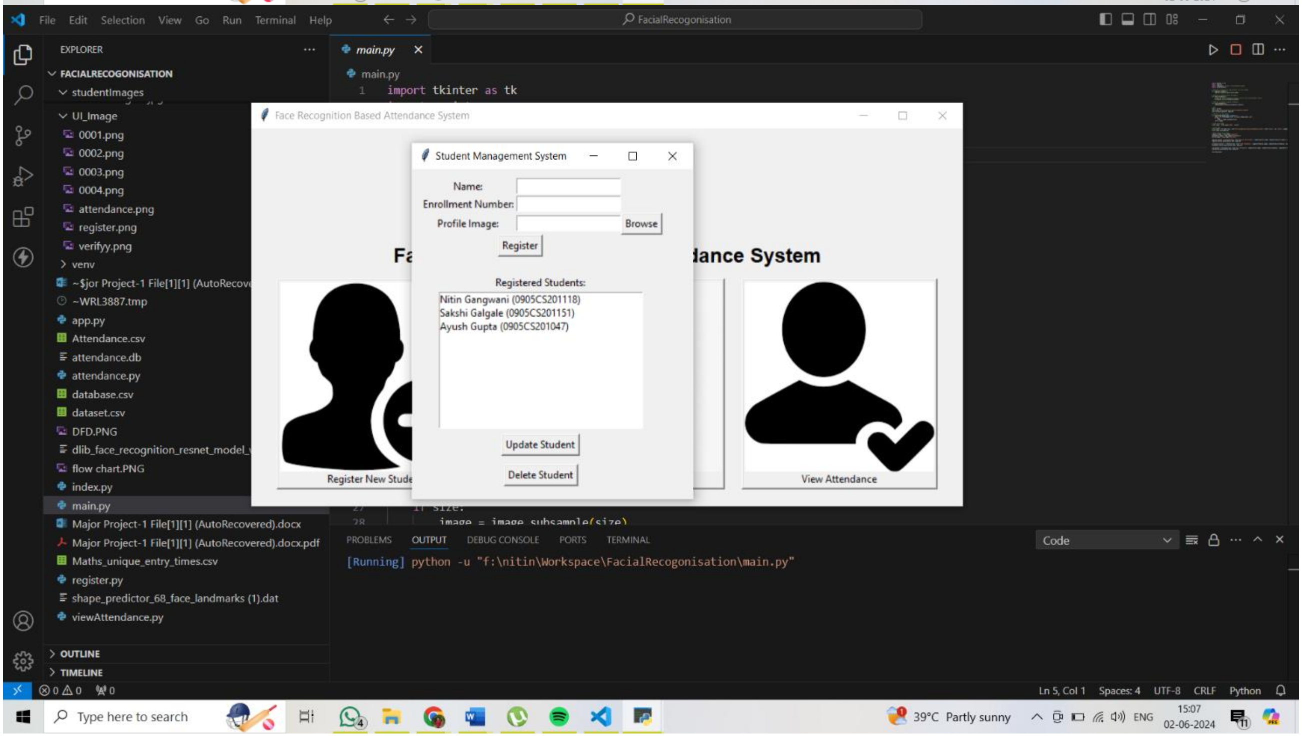
            with open('database.csv', 'w', newline='') as file:
                writer = csv.writer(file)
                writer.writerows(students)

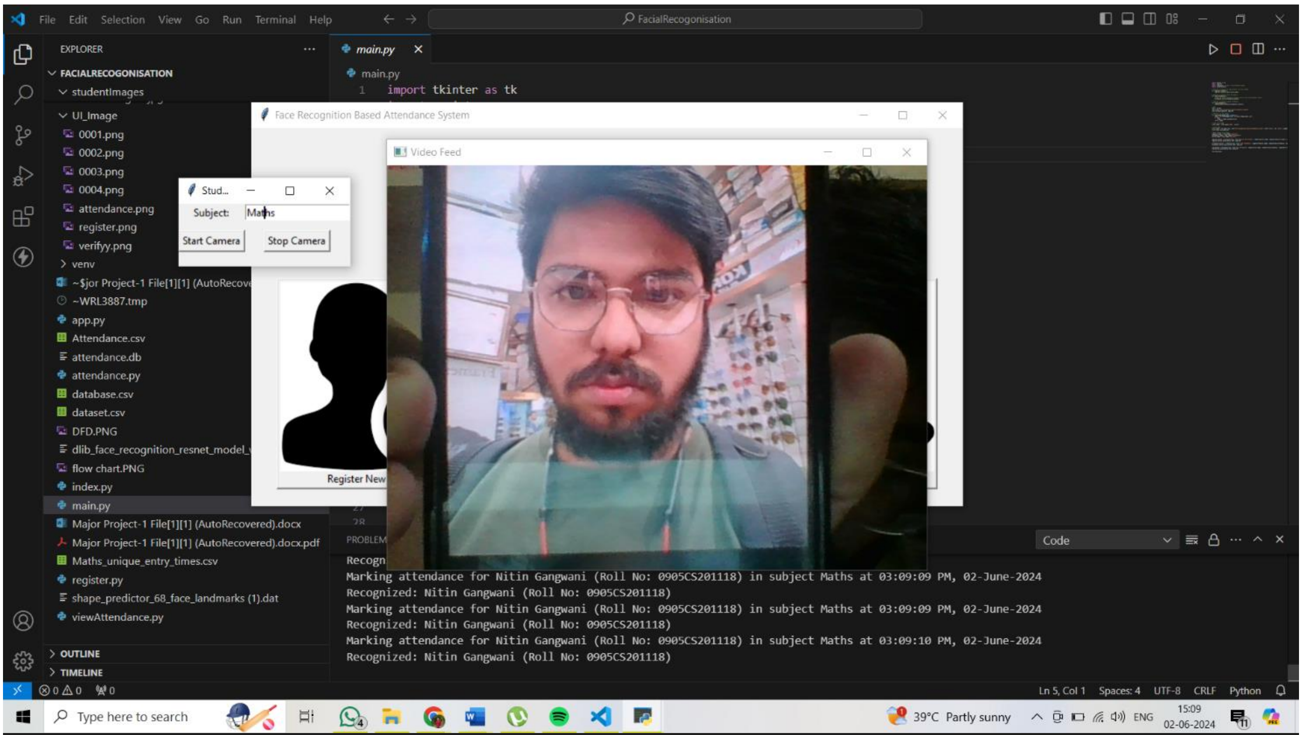
            messagebox.showinfo("Success", "Student entry deleted successfully!")
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
Code
Ln 182, Col 52 Spaces: 4 UTF-8 CRLF Python
Type here to search
39°C Partly sunny
ENG
15:06
02-06-2024

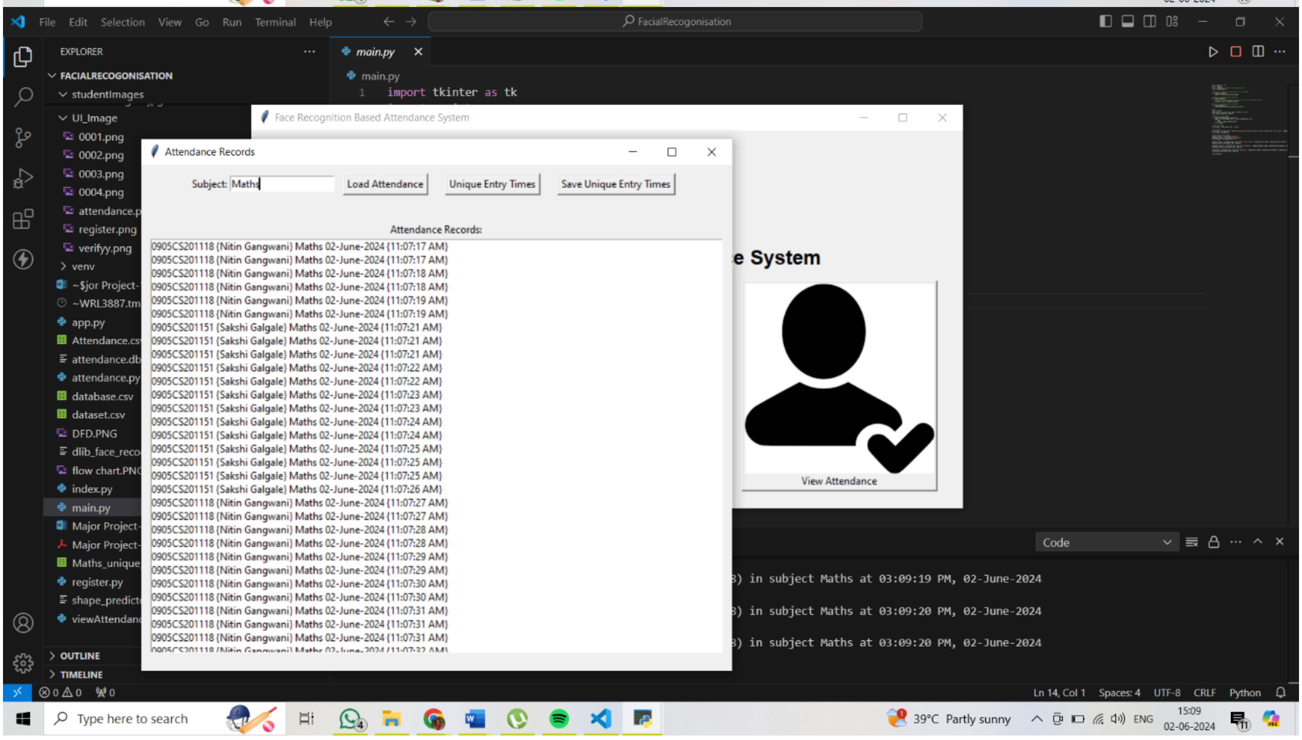

viewAttendance.py
import tkinter as tk
import csv
import os
from collections import defaultdict
from datetime import datetime
from tkinter import messagebox
def load_attendance():
    subject = subject_entry.get()
    attendance_listbox.delete(0, tk.END)
    if os.path.exists('attendance.csv'):
        with open('attendance.csv', 'r') as file:
            reader = csv.reader(file)
            for row in reader:
                if subject in row:
                    attendance_listbox.insert(tk.END, row)
def display_unique_entry_times():
    subject = subject_entry.get()
    attendance_listbox.delete(0, tk.END)
    entry_times = defaultdict(list)
    names = defaultdict(set)
    if os.path.exists('attendance.csv'):
        with open('attendance.csv', 'r') as file:
            reader = csv.reader(file)
            for row in reader:
                if subject in row:
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
Ln 58, Col 38 Spaces: 4 UTF-8 CRLF Python

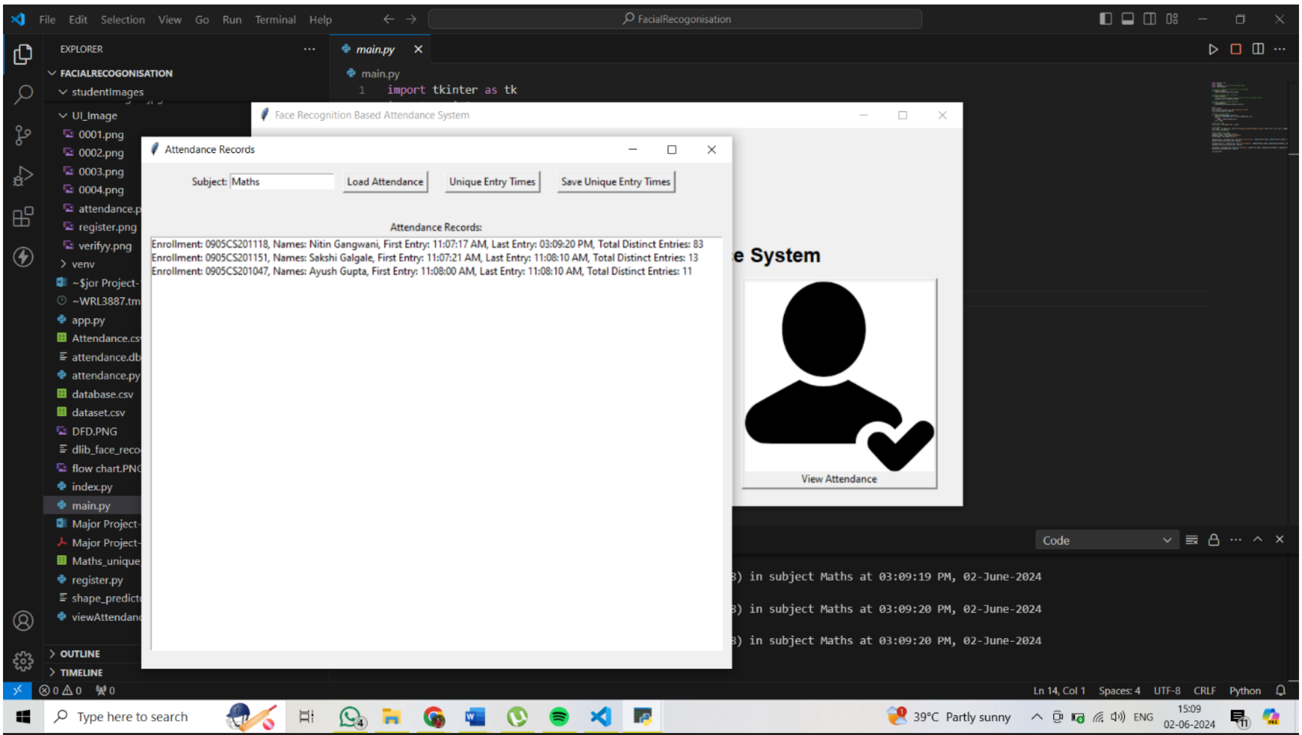

viewAttendance.py
def save_unique_entry_times():
    subject = subject_entry.get()
    filename = f"{subject}_unique_entry_times.csv"
    with open(filename, 'w', newline='') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(["Enrollment", "Names", "First Entry", "Last Entry", "Total Distinct Entries"])
        entry_times = defaultdict(list)
        names = defaultdict(set)
        if os.path.exists('attendance.csv'):
            with open('attendance.csv', 'r') as file:
                reader = csv.reader(file)
                for row in reader:
                    if subject in row:
                        enrollment_number = row[0]
                        name = row[1]
                        timestamp = row[3] + ' ' + row[4]
                        time_format = '%d-%B-%Y %I:%M:%S %p'
                        current_time = datetime.strptime(timestamp, time_format).time()
                        entry_times[enrollment_number].append(current_time)
                        names[enrollment_number].add(name)
            for enrollment_number, times in entry_times.items():
                unique_times = set(times)
                min_time = min(unique_times).strftime('%I:%M:%S %p')
                max_time = max(unique_times).strftime('%I:%M:%S %p')
                total_distinct_entries = len(unique_times)
                unique_names = ', '.join(names[enrollment_number])
                writer.writerow([enrollment_number, unique_names, min_time, max_time, total_distinct_entries])
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
Ln 58, Col 38 Spaces: 4 UTF-8 CRLF Python

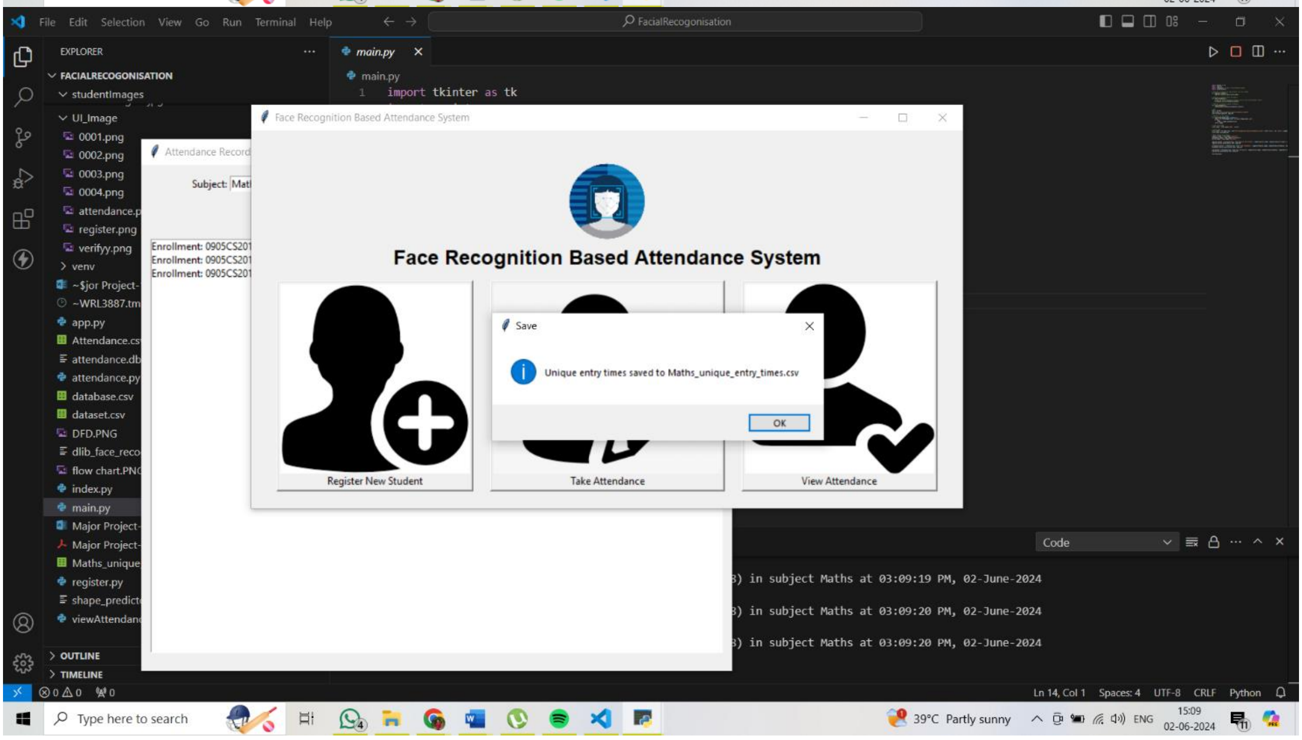

Face Recognition Based Attendance System
Face Recognition Based Attendance System
Register New Student
Take Attendance
View Attendance
[Running] python -u "f:\nitin\Workspace\FacialRecogonisation\main.py"


Face Recognition Based Attendance System
Student Management System
Name:
Enrollment Number:
Profile Image:
Browse
Register
Registered Students:
Nitin Gangwani (0905CS201118)
Sakshi Galgale (0905CS201151)
Ayush Gupta (0905CS201047)
Update Student
Delete Student
View Attendance
[Running] python -u "f:\nitin\Workspace\FacialRecogonisation\main.py"


Face Recognition Based Attendance System
Video Feed
Subject: Maths
Start Camera
Stop Camera
Register New
Marking attendance for Nitin Gangwani (Roll No: 0905CS201118) in subject Maths at 03:09:09 PM, 02-June-2024
Recognized: Nitin Gangwani (Roll No: 0905CS201118)
Marking attendance for Nitin Gangwani (Roll No: 0905CS201118) in subject Maths at 03:09:09 PM, 02-June-2024
Recognized: Nitin Gangwani (Roll No: 0905CS201118)
Marking attendance for Nitin Gangwani (Roll No: 0905CS201118) in subject Maths at 03:09:10 PM, 02-June-2024
Recognized: Nitin Gangwani (Roll No: 0905CS201118)


Face Recognition Based Attendance System
Attendance Records
Subject: Maths
Load Attendance
Unique Entry Times
Save Unique Entry Times
Attendance Records:
0905CS201118 (Nitin Gangwani) Maths 02-June-2024 (11:07:17 AM)
0905CS201151 (Sakshi Galgale) Maths 02-June-2024 (11:07:21 AM)
0905CS201118 (Nitin Gangwani) Maths 02-June-2024 (11:07:27 AM)
e System
View Attendance


Attendance Records
Subject: Maths
Load Attendance
Unique Entry Times
Save Unique Entry Times
Attendance Records:
Enrollment: 0905CS201118, Names: Nitin Gangwani, First Entry: 11:07:17 AM, Last Entry: 03:09:20 PM, Total Distinct Entries: 83
Enrollment: 0905CS201151, Names: Sakshi Galgale, First Entry: 11:07:21 AM, Last Entry: 11:08:10 AM, Total Distinct Entries: 13
Enrollment: 0905CS201047, Names: Ayush Gupta, First Entry: 11:08:00 AM, Last Entry: 11:08:10 AM, Total Distinct Entries: 11
View Attendance


Face Recognition Based Attendance System
Save
Unique entry times saved to Maths_unique_entry_times.csv
OK
Register New Student
Take Attendance
View Attendance

CHAPTER 5. METHEDOLOGY

5.1 DATA COLLECTION

Data collection is a critical component of any facial recognition-based attendance system. The accuracy and effectiveness of the system largely depend on the quality and quantity of the data collected. This section outlines the process of collecting the necessary data, including the types of data required, the methods used for data acquisition, and the procedures for ensuring data quality and integrity.

Types of Data Collected

1. **Student Information:**
    - **Personal Details:** Includes the student's name, enrollment number, and other identifying information.
    - **Profile Images:** High-quality images of each student’s face, which are essential for training the facial recognition model.
    - **Enrollment Records:** Data pertaining to the enrollment status of each student, which helps in managing active and inactive students.
2. **Attendance Records:**
    - **Timestamps:** The date and time when attendance is marked for each student.
    - **Subjects:** Information about the subjects or classes for which attendance is being recorded.
    - **Status:** Presence or absence status of students.

Data Collection Methods

1. **Manual Data Entry:**
    - **Description:** Personal details and enrollment records are manually entered into the system by an administrator.
    - **Tools Used:** User interfaces created using Tkinter allow administrators to input and update student information.
2. **Image Acquisition:**
    - **Description:** Profile images of students are captured using a camera or uploaded from existing photo files.
    - **Tools Used:** The Tkinter interface facilitates image browsing and uploading, while OpenCV is used for capturing images through a camera.
3. **Automated Attendance Recording:**
    - **Description:** The facial recognition system automatically captures and records attendance based on the images taken during each class session.
    - **Tools Used:** OpenCV and face_recognition libraries are utilized to detect and recognize faces in real-time, and attendance data is stored in CSV files.

Ensuring Data Quality and Integrity

1. **Validation Checks:**
    - **Description:** Input fields for student details are validated to ensure completeness and correctness. For instance, enrollment numbers must be unique, and profile images must be in acceptable formats (JPEG, PNG).
    - **Tools Used:** Tkinter validation functions and message boxes to prompt the user for correct inputs.
2. **Image Quality:**
    - **Description:** Profile images must be of high resolution and properly framed to ensure accurate facial recognition.
    - **Tools Used:** Image processing techniques using the PIL (Python Imaging Library) to resize and enhance images.
3. **Data Backup and Security:**
    - **Description:** Regular backups of the student database and attendance records are

maintained to prevent data loss. Access to data entry and modification is restricted to authorized personnel.
- **Tools Used:** CSV file handling for backups, and proper directory management using Python's os and shutil libraries.

5.2 FEATURE SELECTION

Feature selection is a crucial step in the development of a facial recognition-based attendance system. It involves identifying the most relevant attributes from the data that will be used to train the facial recognition model. Effective feature selection enhances the model's accuracy and efficiency, allowing for precise identification of students and reliable attendance recording. This section outlines the features selected for the facial recognition model and the rationale behind their selection.

Features for Facial Recognition

1. **Facial Landmarks:**
   - **Description:** Key points on the face, such as the eyes, nose, mouth, and jawline.
   - **Rationale:** Facial landmarks are essential for capturing the unique geometric properties of a person's face. They are invariant to changes in lighting and pose, making them reliable features for facial recognition.
   - **Tools Used:** The **face_recognition** library is utilized to detect and extract facial landmarks.
2. **Face Encodings:**
   - **Description:** High-dimensional vectors representing the facial features.
   - **Rationale:** Face encodings provide a compact and robust representation of the face, enabling efficient comparison between different faces. These encodings are generated using deep learning models trained on large datasets.
   - **Tools Used:** The **face_recognition** library provides functionality to generate and compare face encodings.
3. **Profile Images:**
   - **Description:** The raw images of students' faces used for training and recognition.
   - **Rationale:** High-quality profile images are the basis for extracting facial landmarks and generating face encodings. Clear and well-lit images ensure better model performance.
   - **Tools Used:** OpenCV for image capture and processing, and the PIL library for image resizing and enhancement.
4. **Attendance Metadata:**
   - **Description:** Information such as timestamps, subject names, and attendance status.
   - **Rationale:** Metadata is crucial for contextualizing the attendance records and ensuring that the correct data is stored for each session. This information helps in tracking student participation over time.
   - **Tools Used:** Tkinter for user input and CSV files for data storage.

Feature Selection Process

1. **Data Preprocessing:**
   - **Description:** Involves cleaning and transforming raw data into a suitable format for feature extraction.
   - **Steps:**
     - Convert profile images to grayscale to standardize the input format.
     - Resize images to a fixed dimension to maintain uniformity.
     - Normalize pixel values to enhance the model's robustness.
   - **Tools Used:** OpenCV and PIL for image preprocessing.
2. **Feature Extraction:**
   - **Description:** Extracting meaningful features from the preprocessed data.
   - **Steps:**

    - Detect facial landmarks using the **face_recognition** library.
    - Generate face encodings from the detected landmarks.
  - **Tools Used:** The **face_recognition** library for landmark detection and encoding generation.
3. **Feature Selection:**
  - **Description:** Identifying the most relevant features for training the facial recognition model.
  - **Steps:**
    - Evaluate the importance of different facial landmarks and encodings using statistical methods.
    - Select the most significant features that contribute to accurate facial recognition.
  - **Tools Used:** Python libraries such as NumPy and scikit-learn for statistical analysis and feature selection.

5.3 DATA PREPROCESSING

Data preprocessing is a critical step in the development of a facial recognition-based attendance system. It involves preparing raw data, particularly images, to be in a suitable format for feature extraction and model training. Proper data preprocessing ensures that the input data is clean, standardized, and enhances the performance of the facial recognition model. This section outlines the various preprocessing techniques applied to the dataset used in the system.

Preprocessing Steps

1. **Image Acquisition:**
  - **Description:** Collecting raw images of students' faces for registration and attendance purposes.
  - **Tools Used:** OpenCV for capturing images from a camera, Tkinter for file selection.
2. **Image Resizing:**
  - **Description:** Adjusting the dimensions of images to a fixed size to ensure uniformity.
  - **Process:**
    - All images are resized to a standard dimension (e.g., 300x300 pixels) to maintain consistency across the dataset.
    - This step helps in reducing computational complexity and ensures that the facial recognition model receives input images of the same size.
  - **Tools Used:** PIL (Python Imaging Library) for image resizing.
3. **Grayscale Conversion:**
  - **Description:** Converting color images to grayscale to reduce computational load and focus on structural features.
  - **Process:**
    - Convert each image from RGB (Red, Green, Blue) format to grayscale.
    - This step helps in minimizing the effect of color variations and lighting conditions, making the recognition process more robust.
  - **Tools Used:** OpenCV for image color conversion.
4. **Normalization:**
  - **Description:** Scaling pixel values to a standard range (0 to 1) to enhance model performance.
  - **Process:**
    - Normalize the pixel values by dividing each pixel value by 255 (the maximum pixel value for 8-bit images).
    - Normalization ensures that the neural network processes the data more efficiently, leading to faster convergence during training.
  - **Tools Used:** NumPy for array manipulation and normalization.

5. **Face Detection:**
    - **Description:** Identifying and extracting the region of interest (the face) from each image.
    - **Process:**
        - Use the **face_recognition** library to detect facial landmarks and crop the face region from the image.
        - This step focuses the model on the facial features, improving accuracy by eliminating background noise.
    - **Tools Used: face_recognition** library for face detection and landmark extraction.
6. **Data Augmentation (Optional):**
    - **Description:** Generating additional training data by applying various transformations to existing images.
    - **Process:**
        - Apply transformations such as rotation, flipping, zooming, and brightness adjustment to augment the dataset.
        - Data augmentation helps in improving the model's generalization by exposing it to a variety of conditions.
    - **Tools Used:** Augmentor or other image augmentation libraries in Python.
7. **Data Cleaning:**
    - **Description:** Removing duplicates, corrupt images, and ensuring data integrity.
    - **Process:**
        - Identify and remove duplicate images from the dataset.
        - Check for and eliminate any corrupt or incomplete images that could negatively impact model training.
    - **Tools Used:** Custom Python scripts using OS and PIL libraries for file handling and image verification.
8. **Encoding and Storing Face Data:**
    - **Description:** Generating face encodings and storing them for later use in recognition.
    - **Process:**
        - Extract face encodings using the **face_recognition** library, which generates a unique 128-dimensional vector for each face.
        - Store the encodings along with student metadata (name, enrollment number) in a structured format (e.g., CSV or database).
    - **Tools Used: face_recognition** library for generating encodings, CSV module for storing data.

5.4 MODEL SELECTION

Model selection in an attendance management system involves choosing the most suitable algorithm or method to analyze attendance data effectively. This process is crucial to ensure the system meets its objectives, handles data efficiently, and provides accurate and insightful analyses. The following sections outline the key aspects of model selection, including system requirements, data characteristics, commonly used models, and the criteria for evaluating and selecting the best model.

System Requirements and Data Characteristics

1. **System Requirements:**
    - **Objective Alignment:** The model must align with the primary goals of the attendance management system, such as tracking attendance accurately, predicting attendance trends, and detecting anomalies.
    - **Real-time Processing:** The ability to handle real-time data inputs and outputs is crucial for maintaining up-to-date attendance records.
    - **Scalability:** The model should be able to scale with the increase in data volume and

    variety over time.
  - **Efficiency:** Efficient data processing and minimal computational overhead are essential for a smooth and responsive system.
2. **Data Characteristics:**
   - **Volume:** The system must manage potentially large volumes of attendance data generated daily.
   - **Variety:** Data can include various forms such as timestamps, student identifiers, and class information, as well as images for facial recognition.
   - **Velocity:** Real-time or near-real-time data processing is often required to ensure timely attendance tracking and reporting.

Commonly Used Models

1. **Classification Models:**
   - **Logistic Regression:** Suitable for binary classification tasks, such as determining whether a student is present or absent.
   - **Support Vector Machines (SVM):** Effective for classification tasks with clear margins of separation.
   - **Random Forests:** A robust model for handling imbalanced datasets and providing feature importance insights.
2. **Regression Models:**
   - **Linear Regression:** Used for predicting continuous outcomes, such as forecasting future attendance rates based on historical data.
   - **Polynomial Regression:** Useful when the relationship between the variables is non-linear.
3. **Clustering Models:**
   - **K-Means Clustering:** Helps in grouping students based on attendance patterns, which can be useful for identifying groups with similar behaviors.
   - **Hierarchical Clustering:** Useful for creating nested clusters and understanding the hierarchy of attendance patterns.

Evaluation Criteria for Model Selection

1. **Accuracy:** The model should provide high accuracy in predicting attendance and identifying patterns.
2. **Performance:** The model must be able to process data quickly, especially for real-time applications.
3. **Scalability:** The model should handle an increasing amount of data without significant performance degradation.
4. **Robustness:** The model should be resilient to noise and able to handle incomplete or imperfect data.
5. **Ease of Implementation:** The complexity of implementing the model and integrating it with the existing system should be manageable.

Proposed Model Selection Process

1. **Initial Assessment:**
   - Conduct an initial assessment of the system's objectives and the nature of the data.
   - Identify the key requirements for the model, such as real-time processing, scalability, and accuracy.
2. **Model Evaluation:**
   - Evaluate different models based on their suitability for the system's objectives.
   - Use historical attendance data to train and test each model, comparing their performance metrics.
3. **Model Testing:**
   - Implement a pilot test of the selected models in a controlled environment.
   - Assess the models' performance in real-time data processing and their ability to

provide accurate predictions and analyses.

4. **Final Selection:**
    - Select the model that best meets the system's objectives and performance criteria.
    - Consider factors such as ease of integration, scalability, and the ability to handle future data growth.

5.5 MODEL TRAINING

Model training is a crucial phase in the development of a facial recognition-based attendance system. During this phase, the selected machine learning or deep learning model learns from the preprocessed data to make predictions or classifications accurately. The training process involves feeding the model with labeled data, adjusting its internal parameters through iterative optimization techniques, and evaluating its performance to ensure it meets the system's requirements. This section outlines the steps involved in model training, including data preparation, model selection, training, and evaluation.

Steps in Model Training

1. **Data Preparation:**
    - **Feature Extraction:** Extract relevant features from preprocessed data, such as facial landmarks and encodings.
    - **Labeling:** Assign labels to the data, indicating the corresponding student identities or attendance statuses.
    - **Data Splitting:** Divide the dataset into training, validation, and testing sets to assess the model's performance effectively.
2. **Model Selection:**
    - **Choice of Algorithm:** Select the appropriate machine learning or deep learning algorithm based on the nature of the problem and the characteristics of the data.
    - **Hyperparameter Tuning:** Tune the model's hyperparameters, such as learning rate, batch size, and regularization strength, to optimize performance.
3. **Model Training:**
    - **Initialization:** Initialize the model's parameters randomly or using pre-trained weights (in the case of transfer learning).
    - **Forward Propagation:** Feed the training data through the model to obtain predictions.
    - **Loss Calculation:** Compute the loss or error between the predicted outputs and the ground truth labels.
    - **Backward Propagation:** Propagate the error backward through the network and update the model's parameters using gradient descent optimization algorithms, such as stochastic gradient descent (SGD) or Adam.
    - **Epochs and Batch Iterations:** Repeat the forward and backward propagation steps for multiple epochs, with each epoch comprising one pass through the entire training dataset. Optionally, perform training in mini-batches to improve computational efficiency.
4. **Model Evaluation:**
    - **Validation Set Performance:** Evaluate the model's performance on the validation set at the end of each epoch to monitor training progress and prevent overfitting.
    - **Metrics Calculation:** Calculate evaluation metrics such as accuracy, precision, recall, and F1-score to assess the model's performance comprehensively.
    - **Early Stopping:** Implement early stopping based on validation set performance to prevent overfitting and save computational resources.
5. **Model Fine-Tuning (Optional):**
    - **Regularization Techniques:** Apply regularization techniques such as dropout or L2 regularization to prevent overfitting and improve generalization.

    - **Architecture Modifications:** Modify the model's architecture based on insights gained during evaluation, such as adding or removing layers, adjusting layer sizes, or changing activation functions.
6. **Model Deployment:**
    - **Testing Set Evaluation:** Assess the trained model's performance on the testing set to obtain unbiased estimates of its generalization ability.
    - **Deployment Considerations:** Prepare the model for deployment in the production environment, ensuring compatibility with the system's infrastructure and performance requirements.

5.6 MODEL VALIDATION

Model validation is a critical step in the development of a facial recognition-based attendance system. It involves assessing the trained model's performance and generalization ability using independent datasets or cross-validation techniques. Proper validation ensures that the model accurately identifies students and records their attendance, even when faced with unseen data. This section outlines the importance of model validation, common validation techniques, and key considerations in validating the facial recognition model.

Importance of Model Validation

1. **Generalization Assessment:** Model validation helps determine how well the trained model performs on unseen data, indicating its ability to generalize to new scenarios.
2. **Performance Evaluation:** Validation provides insights into the model's accuracy, precision, recall, and other performance metrics, enabling developers to assess its effectiveness in real-world applications.
3. **Overfitting Detection:** Validation helps detect overfitting, where the model learns to memorize training data rather than capturing underlying patterns, leading to poor performance on new data.
4. **Hyperparameter Tuning:** Validation aids in optimizing hyperparameters by comparing the performance of different configurations and selecting the most effective ones.
5. **Decision-Making Support:** Validated models instill confidence in stakeholders and decision-makers, facilitating informed choices regarding system deployment and usage.

Common Validation Techniques

1. **Train-Validation-Test Split:**
    - **Description:** Divide the dataset into three disjoint sets: training, validation, and testing.
    - **Process:**
        - Train the model on the training set.
        - Evaluate the model's performance on the validation set during training to monitor progress and prevent overfitting.
        - Finally, assess the model's generalization ability on the testing set, which it has not seen during training or validation.
2. **K-Fold Cross-Validation:**
    - **Description:** Split the dataset into K folds and iteratively use each fold as the validation set while training the model on the remaining data.
    - **Process:**
        - Divide the dataset into K subsets (folds).
        - Train the model K times, each time using a different fold as the validation set and the remaining folds for training.
        - Average the performance metrics across all iterations to obtain an overall estimate of the model's performance.
3. **Leave-One-Out Cross-Validation (LOOCV):**
    - **Description:** Similar to K-fold cross-validation but with K equal to the number of

samples in the dataset.
- **Process:**
    - Train the model K times, each time leaving out one sample as the validation set and using the remaining samples for training.
    - Compute the average performance across all iterations to evaluate the model's performance.
4. **Stratified Sampling:**
    - **Description:** Ensure that each class or group in the dataset is represented proportionally in both the training and validation sets.
    - **Process:**
        - Stratify the dataset based on class labels or other relevant attributes to maintain class balance in the training and validation sets.
        - This technique is particularly useful for imbalanced datasets, where certain classes may be underrepresented..

5.7 MODEL DEPLOYMENT

Model deployment is the final phase in the development lifecycle of a facial recognition-based attendance system. It involves integrating the trained model into the production environment, where it can be used to identify students and record their attendance in real-time. Successful deployment requires careful planning, infrastructure setup, and monitoring to ensure the system operates efficiently and reliably. This section outlines the key steps involved in model deployment, including system architecture design, implementation, testing, and ongoing maintenance.

Steps in Model Deployment

1. **System Architecture Design:**
    - **Component Identification:** Identify the components required for model deployment, including servers, databases, APIs, and user interfaces.
    - **Scalability Considerations:** Design the system architecture to accommodate potential increases in workload and data volume over time.
    - **Security Measures:** Implement security measures such as encryption, access controls, and authentication to protect sensitive data and ensure user privacy.
2. **Implementation:**
    - **Model Integration:** Integrate the trained facial recognition model into the production environment, ensuring compatibility with existing systems and infrastructure.
    - **API Development:** Develop APIs or service endpoints to enable communication between the model and other system components, such as user interfaces or backend servers.
    - **Database Setup:** Configure databases to store attendance records, student information, and other relevant data securely.
3. **Testing:**
    - **Unit Testing:** Conduct unit tests to verify the functionality of individual components, such as APIs and database connections.
    - **Integration Testing:** Test the integration of different system components to ensure seamless communication and data flow.
    - **End-to-End Testing:** Perform end-to-end tests to validate the system's behavior under real-world conditions, including scenarios such as student identification and attendance recording.
4. **Deployment:**
    - **Rollout Strategy:** Plan the rollout strategy for deploying the system into production, considering factors such as deployment environment, user base, and potential impact on existing operations.
    - **Gradual Deployment:** Deploy the system gradually, starting with a small subset of

users or departments before scaling up to the entire organization.
    - **Monitoring and Feedback:** Monitor the system closely during deployment to identify any issues or performance bottlenecks, and collect user feedback for further improvements.
5. **Maintenance and Monitoring:**
    - **Performance Monitoring:** Continuously monitor the system's performance, including response times, resource utilization, and error rates, to ensure optimal operation.
    - **Security Updates:** Stay up-to-date with security patches and updates to protect the system against vulnerabilities and emerging threats.
    - **Model Retraining:** Periodically retrain the facial recognition model using new data to maintain its accuracy and effectiveness over time.
    - **User Support:** Provide ongoing support to users and administrators, addressing any issues or concerns they may encounter during system usage.

Deployment Considerations

1. **Regulatory Compliance:** Ensure compliance with relevant regulations and standards governing data privacy, security, and facial recognition technology.
2. **User Training:** Provide training and guidance to users on how to use the system effectively and responsibly.
3. **Backup and Recovery:** Implement backup and recovery procedures to safeguard data integrity and availability in the event of system failures or data loss.
4. **Scalability Planning:** Plan for future scalability by designing the system to accommodate growth in user base, data volume, and system workload.
5. **Feedback Mechanisms:** Establish mechanisms for collecting user feedback and incorporating it into future system updates and enhancements.
6. **Conclusion**

Model deployment is a critical phase in the lifecycle of a facial recognition-based attendance system, marking the transition from development to real-world usage. By following best practices in system architecture design, implementation, testing, and maintenance, developers can ensure the successful deployment and operation of the system. Effective deployment enables organizations to leverage facial recognition technology to accurately track attendance, enhance security, and streamline administrative processes, ultimately improving efficiency and productivity.

CHAPTER 6. FUTURE OUTLOOK & BIBLIOGRAPHY

6.1 SCOPE OF THE PROJECT

The scope of the facial recognition-based attendance system encompasses various aspects, including its functionalities, target users, deployment environment, and potential future enhancements. By defining the project's scope clearly, stakeholders can establish realistic goals, allocate resources effectively, and ensure successful project execution. This section outlines the scope of the project, highlighting its key components and boundaries.

Functionalities

1. **Facial Recognition:** The system will employ facial recognition technology to identify students based on their facial features captured by a camera.
2. **Attendance Recording:** Upon successful identification, the system will record the student's attendance automatically, along with the date, time, and subject.
3. **Registration:** Administrators will have the ability to register new students by capturing their facial images and associating them with their enrollment details.
4. **Data Management:** The system will manage student data, including profiles, enrollment numbers, and attendance records, in a secure and organized manner.
5. **User Interface:** The system will feature user-friendly interfaces for both administrators (for registration and management) and end-users (for attendance marking).

Target Users

1. **Administrators:** Responsible for system setup, maintenance, and user management.
2. **Teachers:** Utilize the system to take attendance quickly and accurately during classes.
3. **Students:** Experience streamlined attendance processes and enhanced security measures.

Deployment Environment

1. **On-Premises:** The system will be deployed on local servers within educational institutions, ensuring data privacy and compliance with institutional policies.
2. **Integration:** It will integrate with existing infrastructure, such as cameras, databases, and network systems, to facilitate seamless operation.

Boundaries

1. **Non-Real-Time Updates:** While attendance records will be updated in real-time during class sessions, other system updates, such as registration changes, may require manual intervention by administrators.
2. **Limited Scalability:** The system's scalability may be limited to the capacity of the deployed hardware and software infrastructure, with considerations for future expansion.
3. **Single-Subject Attendance:** Initially, the system may focus on recording attendance for individual subjects or classes, with the potential for multi-subject support in future iterations.

Future Enhancements

1. **Multi-Subject Support:** Extend the system to handle attendance tracking across multiple subjects or courses simultaneously.
2. **Enhanced Security Features:** Introduce additional security measures, such as multi-factor authentication or activity logging, to enhance system integrity.
3. **Mobile Integration:** Develop mobile applications to allow administrators and teachers to access the system remotely and perform essential functions on-the-go.
4. **Analytics and Reporting:** Incorporate analytics and reporting capabilities to provide insights into attendance trends, student participation, and class engagement.

6.2 LIMITATIONS

1. **Hardware Dependency:**
    a. **High-Quality Cameras:** The accuracy of facial recognition heavily relies on the quality of the cameras used. Low-quality cameras may result in reduced accuracy and reliability of the system.

   b. **Processing Power:** Facial recognition algorithms require significant processing power, which may pose limitations on the types of devices capable of running the system efficiently.
2. **Environmental Factors:**
   a. **Lighting Conditions:** Variations in lighting conditions, such as shadows or glare, can affect the performance of facial recognition algorithms, leading to false positives or negatives.
   b. **Obstructions:** Facial recognition may be impeded by obstructions like facial coverings, accessories, or hair, reducing its effectiveness in real-world scenarios.
3. **Privacy and Ethical Concerns:**
   a. **Data Privacy:** Storing and processing facial data raise privacy concerns, requiring robust security measures to protect sensitive information from unauthorized access or misuse.
   b. **Ethical Implications:** Facial recognition technology raises ethical questions regarding consent, surveillance, and potential biases, necessitating careful consideration of its implications on individuals' rights and freedoms.
4. **Accuracy and Error Rates:**
   a. **False Positives/Negatives:** Facial recognition systems may exhibit false positives (misidentification of individuals) or false negatives (failure to recognize authorized individuals), impacting the system's reliability and trustworthiness.
   b. **Demographic Biases:** Biases in facial recognition algorithms may lead to disparities in accuracy across demographic groups, raising concerns about fairness and equity in system outcomes.
5. **Legal and Regulatory Compliance:**
   a. **Legislation:** Compliance with regulations such as GDPR, CCPA, and biometric data protection laws requires careful adherence to data handling, consent, and transparency requirements.
   b. **Consent Requirements:** Obtaining explicit consent from individuals for capturing and processing their facial data is essential to comply with privacy regulations and ethical standards.
6. **Scalability and Maintenance:**
   a. **Scalability Challenges:** Scaling the system to accommodate a growing number of users or expanding functionalities may present technical and logistical challenges, requiring careful planning and resource allocation.
   b. **Maintenance Burden:** Regular maintenance, updates, and patches are necessary to address security vulnerabilities, algorithm improvements, and evolving user requirements, adding to the operational overhead of the system.
7. **Integration Complexity:**
   a. **Legacy Systems:** Integrating facial recognition technology with existing infrastructure, such as legacy attendance management systems or databases, may require extensive customization and integration efforts.
   b. **Interoperability:** Ensuring interoperability with diverse hardware and software platforms, including cameras, operating systems, and database systems, poses additional challenges during system implementation.
8. **User Acceptance and Training:**
   a. **Training Needs:** Users, including administrators, teachers, and students, may require training to understand the system's operation, privacy implications, and best practices for effective usage.
   b. **Resistance to Change:** Resistance to adopting facial recognition technology or concerns about privacy and surveillance may hinder user acceptance and adoption, requiring change management strategies to address.

6.3 CONCLUSIONS

The development and implementation of a facial recognition-based attendance system represent a significant advancement in attendance management technology. Throughout the project lifecycle, various components, methodologies, and considerations were meticulously addressed to ensure the system's effectiveness, efficiency, and compliance with regulatory standards. As the project concludes, several key conclusions emerge:

1. **Technological Advancement:** The utilization of facial recognition technology in attendance management signifies a leap forward in automation and accuracy, streamlining traditional attendance-taking processes and reducing administrative burdens.
2. **Enhanced Security:** By leveraging biometric authentication, the system strengthens security measures, mitigating risks associated with proxy attendance, identity fraud, and unauthorized access.
3. **Efficiency Gains:** The automation of attendance recording minimizes manual efforts, enabling teachers and administrators to focus on core educational tasks, fostering a more productive learning environment.
4. **Data Privacy and Ethics:** Despite its benefits, facial recognition technology raises significant concerns regarding data privacy, consent, and ethical implications. Adhering to stringent privacy regulations and ethical guidelines is paramount to safeguarding individuals' rights and fostering trust in the system.
5. **Continuous Improvement:** The project's conclusion marks the beginning of a journey toward continuous improvement and refinement. Ongoing monitoring, feedback collection, and iteration are essential to address limitations, enhance system performance, and adapt to evolving user needs and technological advancements.
6. **Collaboration and Stakeholder Engagement:** The success of the project hinges on effective collaboration and engagement with stakeholders, including educators, students, administrators, policymakers, and regulatory bodies. Building consensus, addressing concerns, and fostering transparency are vital elements of sustainable project outcomes.
7. **Future Directions:** Looking ahead, the project sets the stage for further innovation and exploration in attendance management and biometric authentication. Future directions may include expanding system functionalities, integrating with emerging technologies, and conducting research on privacy-preserving methods and equitable deployment practices.

6.4 BIBLIOGRAPHY

1. Jain, A. K., Ross, A., & Nandakumar, K. (2016). Introduction to Biometrics. Springer.
2. Li, S. Z., & Jain, A. K. (2011). Handbook of Face Recognition (2nd ed.). Springer.
3. Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 3(1), 71–86.
4. OpenCV Library Documentation. (n.d.). Retrieved from https://opencv.org/
5. Tkinter Documentation. (n.d.). Retrieved from https://docs.python.org/3/library/tkinter.html
6. Python Software Foundation. (n.d.). Python 3.9.5 Documentation. Retrieved from https://docs.python.org/3/
7. General Data Protection Regulation (GDPR). (2016). Regulation (EU) 2016/679. Official Journal of the European Union.
8. California Consumer Privacy Act (CCPA). (2018). California Civil Code Sections 1798.100–1798.199.5.
9. Bhattacharjee, S., & Kumar, S. (2020). Privacy-Preserving Face Recognition: A Survey. ACM Computing Surveys, 53(3), 1–37.
10. Federal Trade Commission. (2012). Facing Facts: Best Practices for Common Uses of Facial Recognition Technologies.
11. Bollmann, P., & Tiddens, B. (2019). Face Recognition in Law Enforcement: A European

Perspective on Police Powers and Data Protection. European Data Protection Law Review, 5(2), 230–241.
12. National Institute of Standards and Technology (NIST). (2020). Face Recognition Vendor Test (FRVT) Part 6A. NIST Interagency/Internal Report (NISTIR) 8280.
13. Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. Proceedings of the British Machine Vision Conference (BMVC), 2015.
14. Schoenmakers, S., & Lubbers, M. J. (2021). Attendance Tracking in Education: A Review of Current Practices and Future Directions. Computers & Education, 162, 104071.

6.5 WEB REFERNECES

1 Face Recognition Homepage. (n.d.). Retrieved from https://www.face-rec.org/
2 OpenCV Documentation. (n.d.). Retrieved from https://docs.opencv.org/
3 Tkinter Documentation. (n.d.). Retrieved from https://docs.python.org/3/library/tkinter.html
4 Python Software Foundation. (n.d.). Python 3.9.5 Documentation. Retrieved from https://docs.python.org/3/
5 European Union. (2016). General Data Protection Regulation (GDPR). Regulation (EU) 2016/679. Official Journal of the European Union. Retrieved from https://eur-lex.europa.eu/eli/reg/2016/679/oj
6 California Legislative Information. (2018). California Consumer Privacy Act (CCPA). California Civil Code Sections 1798.100–1798.199.5. Retrieved from https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?lawCode=CIV&division=3.&title=1.81.5.&part=4.&chapter=&article=
7 Federal Trade Commission. (2012). Facing Facts: Best Practices for Common Uses of Facial Recognition Technologies. Retrieved from https://www.ftc.gov/reports/facing-facts-best-practices-common-uses-facial-recognition-technologies/
8 National Institute of Standards and Technology (NIST). (2020). Face Recognition Vendor Test (FRVT) Part 6A. NIST Interagency/Internal Report (NISTIR) 8280. Retrieved from https://www.nist.gov/publications/face-recognition-vendor-test-frvt-part-6a
9 TensorFlow Documentation. (n.d.). Retrieved from https://www.tensorflow.org/
10 PyTorch Documentation. (n.d.). Retrieved from https://pytorch.org/docs/stable/index.html