VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590 014

A Mini - Project Report

On

“ATTENDANCE FACE RECOGNITION SYSTEM”

Submitted in partial fulfillment of the requirements for the **MINI PROJECT (BCD586)** course of the 5th semester

Bachelor of Engineering

In

Computer Science & Engineering (DATA SCIENCE)

Submitted by

Mr. Sankalp s	**Mr. Md Shakir**
(4AI22CD046)	**(4AI22CD036)**
Mr. Sohan Arya	**Mr. Ganesh Naik**
(4AI22CD057)	**(4AI22CD021)**

Under the guidance of

Mrs. Shilpa K V B.E., M.Tech.

Assistant Professor

Department of CS&E (DATA SCIENCE)

Adichunchanagiri Institute of Technology

CHIKKAMAGALURU - 577102

2024-25

ADHICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY

Jyothinagar, Chikkamagaluru-577102

DEPARTMENT OF CS&E (DATA SCIENCE)

CERTIFICATE

This is to certify that the Mini project work entitled "ATTENDANCE FACE RECOGNITION SYSTEM" is a bonafied work carried out by Mr. Sankalp S (4AI22CD046),Mr. Md Shakir (4AI22CD036), Mr. Sohan Arya (4AI22CD057), Ganesh Naik (4AI22CD021) in partial fulfillment for the Mini Project (BCS586) course of 5th semester Bachelor of Engineering in Computer Science and Engineering (Data Science) of the Visvesvaraya Technological University, Belagavi during the academic year 2024-2025. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The Mini project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said Degree.

Signature of the Guide	**Signature of Coordinator**
Mrs. Shilpa K V B.E., M.Tech	**Mrs. Shilpa K V. B.E., M.Tech**
Assistant Professor	**Assistant Professor**

Signature of the HOD
Dr. Adarsh M J B.E., M.Tech., Ph.D
Associate Professor and Head

ABSTRACT

The Attendance Face Recognition System is a modern solution aimed at automating the process of marking attendance through facial recognition technology. Traditional attendance methods, such as manual registers or biometric systems like fingerprint scanning, are prone to errors, time wastage, and security concerns. This system leverages deep learning and computer vision techniques to accurately identify individuals based on their facial features, ensuring a quick, reliable, and contactless method of attendance tracking.

The core of the system involves capturing and processing facial images in real-time, comparing them against a pre-stored database of enrolled faces. Once a face is recognized, the system automatically logs the individual's attendance, timestamping the event for record- keeping. The technology is highly accurate and reduces the possibility of fraudulent activities, such as proxy attendance, by matching facial features with a high degree of precision.

By integrating this system with a user-friendly interface and a centralized database, the project ensures that data can be easily accessed and analyzed. Additionally, it offers the flexibility to be used in various environments, such as schools, universities, and workplaces.

ACKNOWLEDGEMENTS

We express our humble pranamas to his holiness Divine Soul Parama Poojya Jagadguru Padmabushana Sri Sri Sri Dr.Balagangadharanatha Maha Swamiji and Parama Poojya Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji Pontiff, Sri Adichunchanagiri Maha Samsthana Matt and Sri Sri Gunanatha Swamiji, Chikkamagaluru branch, Sringeri who have showered their blessings on us.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of help and support from all quarters during the making of this project, so with gratitude, we take this opportunity to acknowledge all those whose guidance and encouragement helped us emerge successful.

We express our gratitude to Dr. C K Subbaraya, Director, Adichunchanagiri Institute of Technology.

We express our sincere thanks to our beloved principal, Dr. C T Jayadeva for having supported us in our academic endeavors.

We are thankful to the resourceful guidance, timely assistance and graceful gesture of our guide Mrs. Shilpa K V Asst. Professor, Department of CS&E (DATA SCIENCE), who has helped us in every aspect of our project work.

We thank our project coordinator Mrs. Shilpa K V Asst. Professor, Department of CS&E (DATA SCIENCE), for her lively correspondence and assistance in carrying on with this project

We are also indebted to Dr. Adarsh M J, HOD of CS&E (DATA SCIENCE) Department, for the facilities and support extended towards us.

We would be very pleased to express our heart full thanks to all the teaching and non- teaching staff of CS&E (DATA SCIENCE) Department and our friends who have rendered their help, motivation and support.

Sankalp S

Md Shakir

Sohan Arya

GaneshNaik

CONTENTS

List of test cases

List of Snapshots

Chapter 1

Introduction

1.1 Background

- **Context:**
  Face recognition technology has gained popularity in various applications like attendance systems, security, and personal identification. This system leverages computer vision to detect and identify individuals in real time using facial features.
- **Problem:**
  Traditional attendance systems rely on manual input, which is prone to errors and inefficiencies. Additionally, tracking attendance in large groups can be time-consuming and labour-intensive.
- **Opportunity:**
  Implementing an automated face recognition system allows for accurate, real-time attendance tracking, reducing human error and improving efficiency in environments such as classrooms, offices, and events. It also offers the potential for integration with other security and identification systems.

1.2 Problem Statement

- **Over view of program**

  Traditional attendance tracking methods, such as manual entry or card swiping, are inefficient and prone to human error. This often leads to inaccuracies and delays in monitoring attendance in various settings like schools, offices, or events.

- **Specific Issues:**
  Manual attendance systems require constant human intervention, leading to time waste and potential mistakes. Additionally, they lack real-time automation, making them unsuitable for large-scale or high-traffic environments where quick and accurate tracking is essential.

1.3 Objective of the System

The objective is to automate attendance tracking using face recognition technology to enhance efficiency, accuracy, and security in various environments.

Key Goals:

1. **Real-time face recognition:** Accurately detect and identify individuals in real time using a webcam.
2. **Automated attendance marking:** Automatically record attendance by matching detected faces with a pre-encoded list of known individuals.
3. **Error reduction:** Minimize human error in attendance tracking by eliminating manual entry.

1.4 Significance of the System

- Automated Attendance Tracking: The system uses facial recognition to automatically identify individuals and record their attendance, eliminating the need for manual attendance tracking.
- Enhanced Accuracy and Security: By leveraging face recognition technology, the system ensures a high level of accuracy and security in attendance recording, preventing proxy attendance.
- Real-Time Monitoring: The system continuously monitors the video feed in real-time, instantly detecting faces and updating the attendance records whenever a known face is detected.
- Ease of Use: The system operates through a simple interface that displays live video and attendance details, with no need for complex user interaction.

1.5 Scope of the Project

In Scope:

1. Face Recognition-Based Attendance System:
2. The system utilizes facial recognition to identify individuals and automatically mark their attendance in real-time.
3. Webcam Integration:
4. The system works with a webcam to capture live video feeds and processes them for face detection and recognition.
5. Training with Known Faces:
6. The system is designed to train with a set of known images (faces) that it can recognize during live video capture.
7. CSV File for Attendance:
8. Attendance records are stored in a CSV file with timestamps, allowing for easy tracking and retrieval of attendance data.
9. Real-Time Face Matching:
10. Once the system detects a face, it matches it with known faces and records the name and time in real-time.

Out of Scope:

11. Multi-Factor Authentication (MFA):
12. The system does not incorporate additional authentication methods, such as passwords or biometric verification beyond facial recognition.
13. Handling Poor Image Quality:
14. The system does not specifically address issues with low-quality or blurry images, either for training or recognition purposes.

1.6 Methodology

- **Image Collection:**
  Collect a set of training images containing faces for recognition. These images are stored in a specified directory on the system.
- **Image Preprocessing:**
  Load valid image files (with extensions .jpg, .jpeg, .png) from the specified directory. Convert each image to a format suitable for face recognition.
- **Real-Time Face Detection and Recognition:**
  Capture video from the webcam in real-time.
  Resize and process each frame for faster face detection and encoding.
- **Face Comparison:**
  Detect faces in each video frame using the face_recognition library.
  Compare detected faces against stored face encodings and find the best match.

1.7 Target Audience

- **Teachers:**
  Teachers can use the system to easily track student attendance without manual intervention. It helps save time during classes and ensures accurate attendance records.
- **Students:**
  Students benefit from automated attendance tracking, ensuring that their presence is recorded accurately. It reduces the chances of errors and missed entries.
- **Administrators:**
  Administrators can manage and access real-time attendance data for efficient monitoring. The system simplifies administrative tasks related to attendance record maintenance.

1.8 Overview of the Report

- 1.1 Background
- 1.2 Problem statement
- 1.3 objective of the System
- 1.4 Significance of the System
- 1.5 Scope of the Project
- 1.6 Methodology
- 1.7 Target Audience

Chapter 2

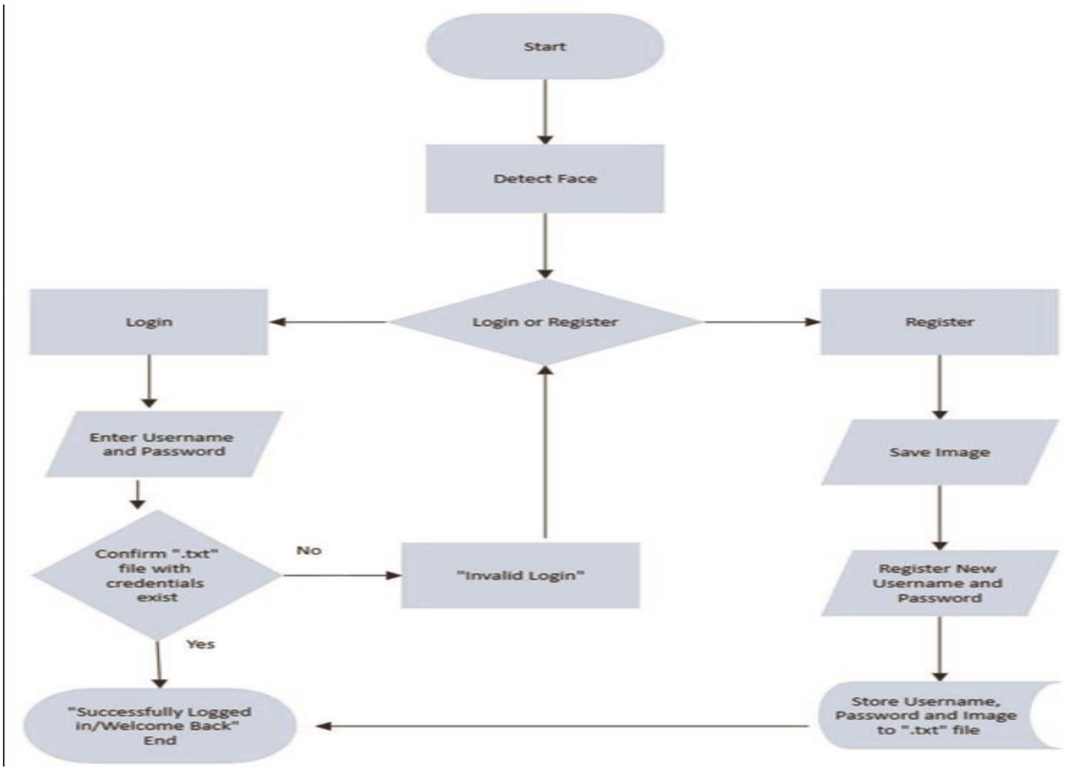
System Design

The system is designed to recognize faces from images captured via a webcam, compare them to a pre-stored dataset, and log attendance automatically when a match is detected. It uses basic image processing techniques, such as face detection and encoding, and handles various edge cases like missing faces and duplicate entries.

2.1 System Architecture

- **High-Level Overview**: The system captures real-time video from a webcam, detects faces, and compares them with pre-stored face encodings to identify individuals. Once recognized, it logs the person's attendance with a timestamp into a CSV file.

- **Architecture Diagram**:


Start
Detect Face
Login or Register
Login
Register
Enter Username and Password
Save Image
Confirm ".txt" file with credentials exist
No
"Invalid Login"
Register New Username and Password
Yes
"Successfully Logged in/Welcome Back" End
Store Username, Password and Image to ".txt" file

- **Components**:
  1. **User Interface (UI)**: Displays the webcam feed with real-time face recognition results and feedback on detected individuals.
  2. **Webcam (Capture Device)**: Captures live video frames and sends them for processing in the face recognition system.
  3. **Real-Time Face Detection**: Detects faces in the captured frames and generates face encodings for recognition.
  4. **Face Recognition**: Matches the detected faces with pre-stored encodings and identifies the person.
  5. **Attendance Marking**: Logs recognized individuals' names and timestamps into a CSV file for attendance tracking.

2.2 Module Design

- This modular design allows for easy debugging, testing, and updates to individual components like image loading, face encoding, or attendance marking without affecting the entire system.

2.2.1 Image Preprocessing Module.
2.2.2 Face Encoding Module.
2.2.3 Attendance Marking Module.
2.2.4 Real-Time Face Recognition Module.

2.3 User Interface (UI) Design

- **Main Screens**:

- **Navigation Menu**: Buttons for navigating to different sections such as "Start Recognition", "View Attendance", and "Settings".
- **Live Camera Feed**: Display the webcam video stream with detected faces highlighted by a bounding box.
- **Attendance Log**: A scrollable list or table displaying names of recognized individuals along with their timestamp.
- **Face Recognition Sensitivity**: Users can adjust the recognition threshold for better accuracy or speed.

2.4 Technology Stack

- **Frontend: Python (OpenCV)**
- **Backend: Python (Face Recognition, Flask for web interface)**

Implementation

Chapter 3

This code performs real-time face recognition using a webcam, where it matches faces detected in video frames to pre-encoded known faces from training images and logs recognized individuals' names with timestamps into an "Attendance.csv" file. It continuously displays the webcam feed and allows the user to quit by pressing 'q'.

3.1 Backend Implementation

The backend implementation of the code involves several key steps to load, process, and use face recognition to mark attendance. Here's an explanation of the backend implementation of each major function:

1. Load and Process Training Images:

- **Directory Setup**: A specific directory (Training_images) is set where images of known faces are stored. These images are used to generate face encodings, which are unique representations of the faces.
- **Image Loading:** The code filters out valid image files (e.g., .jpg, .jpeg, .png) from the directory using os.listdir() and loads them using cv2.imread(). For each valid image, the filename is stored as a "class name," which corresponds to the person's name.

python

Copy code

```
valid_extensions = ('.jpg', '.jpeg', '.png')
myList = [file for file in os.listdir(path) if file.lower().endswith(valid_extensions)].
```

2. Face Encoding:

- **Face Detection:** The first step involves detecting a face in an image or video frame. This can be done using a pre-trained model (like the ones used by the face_recognition library) which finds the locations of faces in the image.
- **Face Encoding**: Once a face is detected, the face_recognition.face_encodings function is used to convert the face into a numerical vector. This vector represents the unique features of the face (such as the distance between key facial landmarks). Each face has a different encoding, which can be used for comparison.
- **Matching Faces:** To recognize a person, the encoded face is compared to a database of previously stored encodings. The face_recognition.compare_faces function compares the new face's encoding to known face encodings and checks for matches by calculating the "distance" between them. If the distance is small enough, the faces are considered a match.

3. Attendance Logging:

1. **Attendance Function:**

- The mark Attendance(name) function writes the detected person's name and the current timestamp to an Attendance.csv file.
- The Attendance.csv file is opened in append mode (a+), and each entry contains the name and the date/time of detection.

2. **Main Process**:

- When a match is found between a detected face and one of the stored encodings, the system logs the detected person’s name and the timestamp into the Attendance.csv file.
- The mark Attendance() function is called every time a new match is detected.

4. Real-Time Webcam Processing:

- **Webcam Capture:** The script opens the webcam (cv2.VideoCapture(0)) to capture real-time video frames for face detection and recognition.
- **Face Detection and Encoding:** Each frame is resized and converted to RGB for face detection. Detected faces are encoded using face_recognition.face_encodings().
- **Face Matching:** The encoded faces are compared with stored encodings to identify a match, and the best match is highlighted with a rectangle and name label on the frame.
- **Attendance Logging**: When a match is found, the person's name and timestamp are logged into an Attendance.csv file to record their presence in real-time.

5. Run the System:

- The startCamera() function is called to begin the process, capturing webcam frames and recognizing faces in real time.

Backend Overview:

- **Input**: The system receives a series of training images stored in a folder (Training_images).
- **Processing**: The system processes each image to extract face encodings, compares those encodings with faces detected from webcam feed, and identifies matching faces.
- **Output**: Recognized names are logged with timestamps into an attendance CSV file (Attendance.csv), and feedback is shown by displaying the webcam feed with recognized faces highlighted.

Dependencies:

- **OpenCV**: For webcam interaction and image processing.
- **face_recognition**: For face detection and encoding.

3.2 Frontend Implementation

The frontend provides the user interface for teachers, students, and administrators to interact with the system.

User Interface (UI) Components

- **Login Page**: Allows users to log in with their credentials, validating input before sending data to the backend.

- **Dashboard**: Displays navigation options tailored to each user role (Admin, Teacher, or Student).

- **Attendance Screen**: Enables teachers to view a list of students in a selected class and mark attendance status for each student.

- **Report Screen**: Provides options to filter and view attendance data, enabling users to generate reports.

3.3 Database Implementation

- **Database Setup**: Used MySQL as the database system to store user andattendance data in a relational structure.
- **Database Schema**:
  - **User Table**: Stores user data, including user_id, username, password_hash, and role.
  - **Student Table**: Stores student-specific data, such as student_id, name, email, and class_id.
  - **Attendance Table**: Records attendance entries with attendance_id, student_id, class_id, date, and status.
  - **Class Table**: Contains class_id, course_id, and teacher_id for each class session.
  - **Course Table**: Contains course details, such as course_id and course_name.

Chapter 4

Testing

This chapter covers the testing processes and methodologies applied to the Student Attendance System. Testing is essential to identify and correct any issues, validate that the system meets functional and non-functional requirements, and ensure that it performs reliablyunder various conditions.

4.1 Testing Objectives

- Verify that the system functions as intended by executing a series of test cases.
- Ensure that all user roles (Admin, Teacher, Student) can access the intended features without any errors.
- Test the accuracy of attendance records and data retrieval.
- Confirm that the system is secure and handles invalid inputs or unauthorized access appropriately.
- Evaluate the system's performance and reliability under load.

4.2 Testing Environment

- **Hardware**: Laptop/PC with minimum 8GB RAM and multi-core processor.
- **Software**:
  - Backend and frontend hosted on local servers (Node.js/Django for backend, React/Angular for frontend).
  - Database: MySQL .
  - Testing Tools: Face Recognition, Image Loading and Encoding: The cv2 (OpenCV) and face_recognition libraries are used to load and encode training images for later comparison.
- **Operating System**: Windows 10/macOS/Linux.
- **Browser**: Google Chrome, Firefox, and Microsoft Edge .

4.3 Types of Testing

4.3.1 Unit Testing

- **Objective**: To test individual components or functions in isolation to verify their correctness.
- **Tools**: Jest or Mocha for backend logic testing, and React Testing Library for frontend component testing.
- **Example Test Cases**:
  - **User Authentication**: Verifies that the login function correctly authenticates users based on their credentials.
  - **Attendance Recording**: Tests that attendance is accurately marked and saved in the database.
  - **Report Generation**: Confirms that the report function generates data correctly for given filters (e.g., by date or student).

4.3.2 Integration Testing

- **Objective**: To test the interaction between different modules of the system (e.g., frontend and backend, backend and database).
- **Example Test Cases**:
  - **Attendance Submission**: Ensures that the frontend sends attendance data to the backend, and the backend stores it correctly in the database.
  - **Data Retrieval for Reports**: Verifies that the report module retrieves and displays accurate attendance data for specified criteria.
  - **User Access Control**: Confirms that only authorized users can access certain features (e.g., only teachers can mark attendance, only admins can manage users).

4.3.3 Functional Testing

- **Objective**: To test the system against functional requirements to ensure it meets specified user needs.
- **Test Scenarios**:
  - **Login and Registration**: Tests login and registration processes for all user roles and checks that users are directed to the appropriate dashboard.
  - **Attendance Management**: Verifies that system can mark attendance for each student and that the system accurately reflects the attendance status.
  - **Viewing Reports**: Checks that students, teachers, and admins can access reports and that data displayed matches attendance records.

4.4 Test Cases

Below are sample test cases for various components:

1. Test Case: Image Loading and Encoding

- **Test Case ID**: TC-001
- **Description**: Verify that valid images are correctly loaded from the specified directory and encoded for face recognition.
- **Steps**:
  1. Place valid images (.jpg, .jpeg, .png) in the training images directory.
  2. Run the script to load images and generate encodings.
- **Expected Outcome**: The images should be loaded without errors, and encodings should be successfully created for each image.

2. Test Case: Invalid Image Handling

- **Test Case ID**: TC-002
- **Description**: Ensure the system handles invalid image formats and skips them appropriately.
- **Steps**:

1. Place non-image files (e.g., .txt, .pdf) in the training images directory.
2. Run the script.

- **Expected Outcome**: The system should skip these files and print a warning without breaking the execution.
- **Result**: Pass/Fail.

3. Test Case: Face Detection and Recognition

- **Test Case ID**: TC-003
- **Description**: Verify that the system correctly detects and recognizes faces from the webcam feed.
- **Steps**:
  1. Ensure a known face is in the webcam frame.
  2. Run the face recognition code.
- **Expected Outcome**: The system should detect the face, match it with the known dataset, and display the correct name.
- **Result**: Pass/Fail

4. Test Case: Unrecognized Face Handling

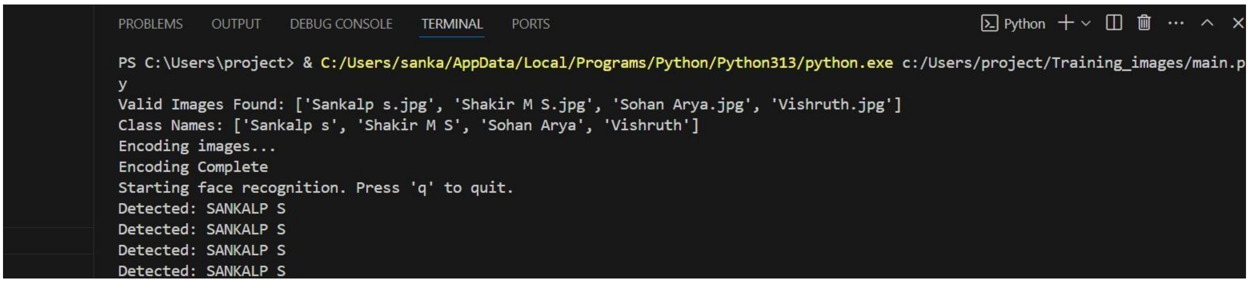
- **Objective**: Ensure that the system does not mark attendance for faces it cannot recognize.
- **Steps**:
  1. Present an unregistered person in front of the camera.
  2. Verify that the system does not mark attendance for this person.
- **Expected Outcome**: The system should not add any attendance entry for unrecognized faces in the Attendance.csv file.

Chapter 5 Results and Discussion

This chapter summarizes the results of the Attendance Face Recognition System project, discussing its effectiveness, reliability, and alignment with the intended objectives. The chapter also covers any challenges encountered, key insights, and recommendations for future improvements.

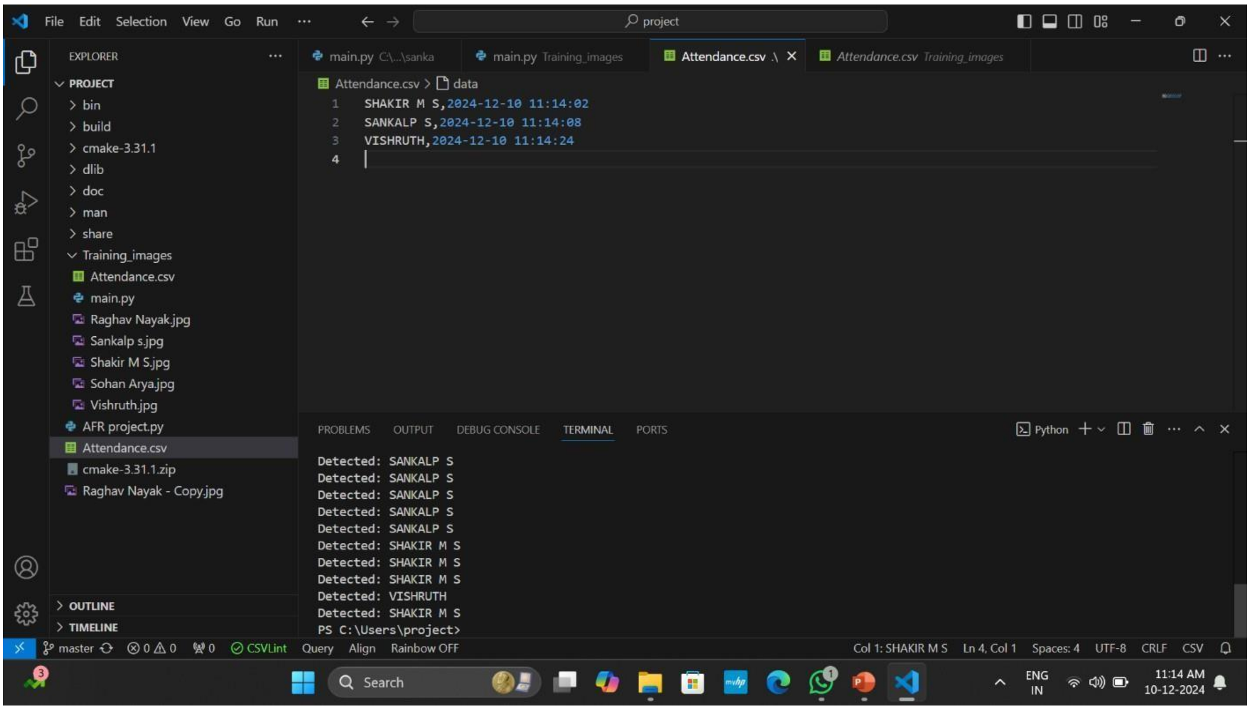
5.1 Results

Snapshots of the Project with description:



```
PS C:\Users\project> & C:/Users/sanka/AppData/Local/Programs/Python/Python313/python.exe c:/Users/project/Training_images/main.p
y
Valid Images Found: ['Sankalp s.jpg', 'Shakir M S.jpg', 'Sohan Arya.jpg', 'Vishruth.jpg']
Class Names: ['Sankalp s', 'Shakir M S', 'Sohan Arya', 'Vishruth']
Encoding images...
Encoding Complete
Starting face recognition. Press 'q' to quit.
Detected: SANKALP S
Detected: SANKALP S
Detected: SANKALP S
Detected: SANKALP S
```

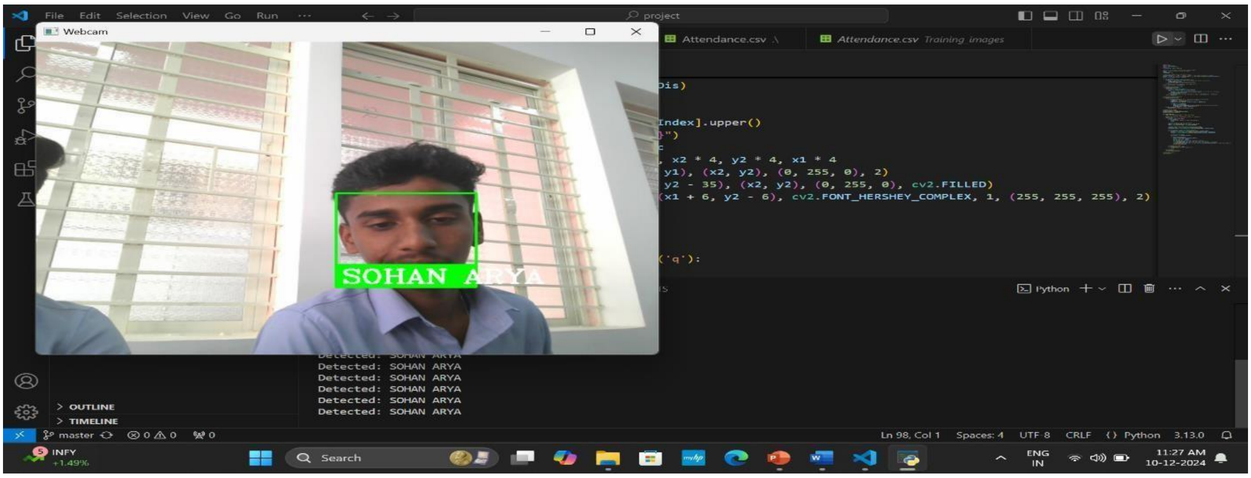
Snapshot 5.1.1 showing which all valid images are stored in the database.



```
SHAKIR M S,2024-12-10 11:14:02
SANKALP S,2024-12-10 11:14:08
VISHRUTH,2024-12-10 11:14:24
```

```
Detected: SANKALP S
Detected: SANKALP S
Detected: SANKALP S
Detected: SANKALP S
Detected: SANKALP S
Detected: SHAKIR M S
Detected: SHAKIR M S
Detected: SHAKIR M S
Detected: VISHRUTH
Detected: SHAKIR M S
PS C:\Users\project>
```

Snapshot 5.1.2 showing the Attendance data .



```
Dis)


Index].upper()
}")
c
, x2 * 4, y2 * 4, x1 * 4
y1), (x2, y2), (0, 255, 0), 2)
y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
(x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)



('q'):
```

```
Detected: SOHAN ARYA
Detected: SOHAN ARYA
Detected: SOHAN ARYA
Detected: SOHAN ARYA
Detected: SOHAN ARYA
Detected: SOHAN ARYA
```

Snapshot 5.1.3 Face Recognizing in camera.

5.2 Discussion

Effectiveness of the System

- The Attendance Face Recognition System effectively achieved its goal of streamlining attendance management, enhancing accuracy, and improving access to attendance records.
- The successful implementation of user roles allowed each type of user to access only the necessary features, improving both security and user experience.
- Overall, the system demonstrated reliable performance, accurate data management, and ease of use, making it a valuable tool for educational institutions.

5.3 Limitations of the Current System:

1 Lighting Conditions

- **Limitation: The system's performance heavily depends on good lighting conditions. Poor lighting, such as very bright or very dim environments, can negatively impact face detection and recognition accuracy.**

- **Impact: If the lighting is not optimal, the system may fail to detect faces, or it may misidentify them, leading to incorrect attendance marking.**

2. Face Occlusions

- **Limitation: The system struggles with faces that are partially obscured by items such as hats, glasses, masks, or hands.**

- **Impact: When individuals wear face coverings or have their faces partially blocked, the system may not recognize them, leading to missed attendance entries or inaccurate recognition.**

Chapter 6

Conclusion and Future Enhancements

6.1 Conclusion

- The Attendance Face Recognition System successfully achieved its primary goal of simplifying and automating the process of attendance management in an educational setting. By enabling System to mark attendance digitally, students to view their records easily, and administrators to generate detailed reports, the system offers an efficient and reliable solution that addresses the needs of all user roles.
- Overall, the Attendance Face Recognition System demonstrates the advantages of digitizing future attendance methods, enhancing both efficiency and accuracy, and reducing the administrative workload on teachers and staff. It has the potential to be an essential tool in educational institutions, contributing to better attendance tracking and management practices.

6.2 Future Enhancements

To further increase the effectiveness and usability of the Attendance Face Recognition System, the following enhancements are recommended:

1 Real-Time Face Recognition:

- **Parallel Processing:** Handle multiple faces in the frame simultaneously, improving the system's ability to recognize and track several individuals at once**.**
-

2 Multi-Camera Support:

- **Multiple Camera Inputs**: Enable support for multiple cameras in large spaces to handle more people at once, such as in office buildings or classrooms.
- **360-Degree Recognition**: Integrate cameras placed at different angles to enable better face recognition from all directions.

3 Dynamic Face Enrollment:

- **Automatic Updating:** Implement a mechanism to automatically update user profiles when there are changes (e.g., a significant change in appearance such as change in face texture , aging, etc.).

REFERENCES

Citation Format:

- **Books**: Vadivel R V and Naveen raj M World Journal of Advance Research 18/04/2023.
- **Journal Articles**: Naveen raj M and Vadivel R V . (2023). "World Journal of Advance Research ".
- Article DOI: https://doi.org/10.3057/wjarr.2023.18.1.0705
- **Online Resources**:
  https://github.com/krishnaik06/Face-Recognition-Attendance-Projects
- **Software Tools**: Python
  Open CV
  JSON