Final Project Report: Face Recognition Based Attendance System

Table of Contents

1. **Executive Summary**
2. **Introduction**
    - 2.1 Background
    - 2.2 Objectives
3. **System Architecture**
    - 3.1 Overview
    - 3.2 Components
4. **Technologies Used**
5. **Implementation Details**
    - 5.1 Attendance Configuration
    - 5.2 Attendance System Class
    - 5.3 User Interaction
6. **Data Management**
7. **Challenges Faced**
8. **Future Enhancements**
9. **Conclusion**
10. **References**
11. **Appendices**

Team Members:

Name	ID	signature
Sheikh Shariar Nehal	*242220005101260*	
Shahrear Hossain	*242220005101888*	
Shouvik Dhali	*242220005101159*	
MD Himel Ahamed	*242220005101284*	
Mubinul Islam	*242220005101626*	

1. Executive Summary

The Face Recognition Based Attendance System is an innovative solution designed to automate the attendance marking process using advanced facial recognition technology. This project integrates computer vision and machine learning techniques to accurately identify individuals and record their attendance in real-time. The system is built using Python and employs various libraries such as Flask, OpenCV, and scikit-learn, providing a user-friendly interface for both administrators and users.

2. Introduction

2.1 Background

Traditional attendance systems often rely on manual processes, which can be time-consuming and prone to errors. The advent of facial recognition technology offers a more efficient and accurate method for tracking attendance. This project aims to leverage this technology to streamline the attendance process in educational institutions and workplaces.

2.2 Objectives

- To develop a system that automatically recognizes faces and marks attendance.
- To create a user-friendly web interface for managing attendance records.
- To ensure the system is robust and can handle various lighting and environmental conditions.
- To provide a scalable solution that can be adapted for different user bases.

3. System Architecture

3.1 Overview

The system architecture consists of a client-server model where the client interacts with a web interface, and the server processes requests and manages data.

3.2 Components

- **Frontend**: A web interface built using HTML, CSS, and JavaScript, allowing users to interact with the system.
- **Backend**: A Flask application that handles requests, processes images, and manages attendance records.
- **Database**: CSV files are used to store attendance records and user data, with potential for future integration with a relational database.

4. Technologies Used

- **Programming Language**: Python
- **Web Framework**: Flask
- **Computer Vision Library**: OpenCV
- **Machine Learning Library**: scikit-learn
- **Data Handling**: pandas
- **Model Serialization**: joblib
- **Frontend Technologies**: HTML, CSS (Tailwind CSS for styling)

5. Implementation Details

5.1 Attendance Configuration

The system is configured using a class **AttendanceConfig**, which defines parameters such as:

- **REQUIRED_IMAGES**: Number of images required for training.
- **MIN_FACE_SIZE**: Minimum size of the face to be detected.
- **MODEL_FILENAME**: Filename for the trained model.
- **CASCADE_FILE**: Haar cascade file for face detection.
- **IMAGE_SIZE**: Size to which images are resized for processing.

5.2 Attendance System Class

The core functionality is encapsulated in the **AttendanceSystem** class, which includes methods for:

- **Setting up directories**: Creates necessary directories for storing attendance records and user images.
- **Extracting faces**: Uses a Haar cascade classifier to detect faces in images.
- **Identifying faces**: Loads a pre-trained KNN model to identify faces.
- **Training the model**: Trains the KNN model using images of known users.
- **Adding attendance**: Records attendance in a CSV file.
- **Setting up routes**: Defines the web routes for the application, including home, start, and add user functionalities.

5.3 User Interaction

The web interface allows users to:

- View today's attendance.
- Start the attendance marking process using the webcam.
- Add new users by capturing their images.

5.4 Attendance Marking Process

When the attendance marking process is initiated:

- The system captures video from the webcam.
- It detects faces in real-time and identifies them using the trained model.
- If a recognized face is found, the system records the attendance.

---

6. Data Management

Attendance records are stored in CSV files, with each entry containing the name, roll number, and time of attendance. The system automatically creates a new CSV file for each day, ensuring organized data management.

---

7. Challenges Faced

- **Lighting Conditions**: Variations in lighting can affect face detection accuracy, leading to missed identifications.
- **Model Training**: Ensuring the model is trained with sufficient and diverse images for accurate recognition.
- **User Experience**: Designing an intuitive interface that is easy to navigate for users with varying

---

8. Future Enhancements

To further improve the Face Recognition Based Attendance System, the following enhancements are proposed:

- **Database Integration**: Transition from CSV file storage to a relational database (e.g., MySQL, PostgreSQL) for better data management, scalability, and security.
- **Mobile Application Development**: Create a mobile version of the application to allow users to mark attendance from their smartphones, increasing accessibility.
- **Enhanced Security Features**: Implement additional security measures, such as user authentication and data encryption, to protect sensitive user information.
- **Real-time Notifications**: Introduce a notification system to alert users when their attendance is marked or if there are any discrepancies.
- **Multi-Factor Authentication**: Add multi-factor authentication for user login to enhance security.
- **Integration with Learning Management Systems (LMS)**: Allow the attendance system to integrate with existing LMS platforms for seamless data exchange and reporting.

9. Conclusion

The Face Recognition Based Attendance System successfully automates the attendance marking process, providing a modern solution to traditional methods. By leveraging advanced technologies such as computer vision and machine learning, the system demonstrates significant improvements in efficiency and accuracy. The user-friendly interface and robust backend ensure a positive user experience, making it a valuable tool for educational institutions and workplaces alike. With potential for future enhancements, this project lays the groundwork for further innovations in attendance management.

10. References

- OpenCV Documentation: OpenCV
- Flask Documentation: Flask
- scikit-learn Documentation: scikit-learn
- Python Official Documentation: Python

11. Appendices

A.1 Attendance Configuration Class

Python code

```
@dataclass
class AttendanceConfig:
    """Configuration class for attendance system"""
    REQUIRED_IMAGES: int = 10
    MIN_FACE_SIZE: Tuple[int, int] = (20, 20)
    MODEL_FILENAME: str = 'face_recognition_model.pkl'
    CASCADE_FILE: str = 'haarcascade_frontalface_default.xml'
    IMAGE_SIZE: Tuple[int, int] = (50, 50)
```

A.2 Attendance System Class

python code

```
class AttendanceSystem:
    def __init__(self, config: AttendanceConfig):
        self.config = config
        self.app = Flask(__name__)
        self.setup_directories()
        self.face_detector = cv2.CascadeClassifier(self.config.CASCADE_FILE)
        self.background = cv2.imread("background.png")
        self.setup_routes()
```

Appendix B: Sample Attendance Record

| Name | Roll | Time | |----------|-------|----------| | nehal | 1260 | 15:29:52 | | iamnehal | 2003 | 19:25:40 | | UER63 | 34545 | 19:25:54 |

Appendix C: Project File Structure

```
MultipleFiles/
|
├── app.py
├── background.png
├── haarcascade_frontalface_default.xml
├── requirements.txt
├── home.html
```

```
├── Attendance-10_30_24.csv
├── Attendance-11_02_24.csv
├── Attendance-12_09_24.csv
├── face_recognition_model.pkl
└── static/
    └── faces/
```