



National University of Modern Languages

PROJECT

FRAS (Facial Recognition Attendance System)

May 20, 2024

ARTIFICIAL INTELLIGENCE

Mam Hira Arshad



Team Members

Umair Farhat CS-25
Abdullah Tariq CS-02
Rehan Raheem Ali CS-18
Ghufran CS-09

Table of Contents

1. Introduction	1
2. Project Overview.....	1
3. Libraries and Dependencies.....	1
4. System Components.....	2
User Interface	2
Face Registration	2
Face Recognition with Liveness Detection	2
Attendance Marking	2
5. Detailed Explanation of Functions and Features.....	2
User Interface Functions	2
Image Loading and Encoding Functions	2
Eye Aspect Ratio Calculation.....	3
Face Registration Function	3
Facial Recognition Function with Liveness Detection	3
Attendance Marking Function	3
6. Why These Libraries and Methods?	4
OpenCV	4
face_recognition	4
dlib.....	4
Tkinter	4
scipy	4
imutils.....	4
7. Conclusion	5

1. Introduction

The Facial Recognition Attendance System is designed to automate the attendance tracking process using facial recognition technology. This system utilizes a webcam to capture live video feed, detects faces, and recognizes registered faces while ensuring liveness detection by monitoring eye blinks.

2. Project Overview

The system comprises several components including a user interface for registration and recognition, face registration functionality, liveness detection, and attendance marking. The primary goal is to provide a reliable, automated, and user-friendly attendance system.

3. Libraries and Dependencies

The project utilizes several libraries:

OpenCV: For image and video processing.

face_recognition: For facial recognition operations.

dlib: For facial landmark detection.

Tkinter: For creating the graphical user interface.

scipy: For calculating distances in liveness detection.

imutils: For convenient handling of image processing functions.

4. System Components

User Interface

The user interface is built using Tkinter. It provides buttons for registering a new face, starting the facial recognition process, and exiting the application.

Face Registration

This feature captures multiple images of a user's face via the webcam, saves them, and prepares them for future recognition.

Face Recognition with Liveness Detection

The recognition process detects faces in real-time, compares them with registered faces, and ensures liveness by checking for eye blinks.

Attendance Marking

Recognized faces are logged with the current date and time into a CSV file.

5. Detailed Explanation of Functions and Features

User Interface Functions

start_facial_recognition()

Starts the facial recognition process in a new thread to keep the UI responsive.

register_new_face()

Prompts the user for their name and starts the face registration process in a new thread.

exit_application()

Exits the application gracefully.

Image Loading and Encoding Functions

load_images_from_directory(directory)

Loads images from a specified directory, handling any errors encountered during the process. It returns a list of images and corresponding class names.

findEncodings(images)

Converts images to RGB and extracts facial encodings. Encodings are unique features extracted from faces that can be used for recognition.

Eye Aspect Ratio Calculation

eye_aspect_ratio(eye)

Calculates the Eye Aspect Ratio (EAR) which is used to detect eye blinks. EAR is computed using specific facial landmarks around the eyes.

Face Registration Function

register_face(name)

Captures multiple images of a user's face via the webcam and saves them for later use in recognition. The images are saved with the user's name and a sequence number.

Facial Recognition Function with Liveness Detection

run_facial_recognition()

Performs facial recognition and liveness detection. It:

- Loads known face encodings.
- Captures video from the webcam.
- Detects faces and facial landmarks in each frame.
- Checks for eye blinks to ensure liveness.
- Matches detected faces with known encodings and marks attendance for recognized faces.

Attendance Marking Function

markAttendance(name)

Logs the recognized name along with the current date and time into a CSV file. Ensures that duplicate entries are avoided.

6. Why These Libraries and Methods?

OpenCV

OpenCV is a powerful library for real-time computer vision and image processing. It is used for capturing video from the webcam, converting image formats, and drawing rectangles around detected faces.

face_recognition

This library simplifies the process of face detection and recognition. It uses deep learning models to extract facial encodings which are highly accurate and efficient for face matching.

dlib

dlib provides robust tools for machine learning and data analysis. In this project, it is primarily used for detecting facial landmarks, which are critical for calculating the Eye Aspect Ratio and ensuring liveness.

Tkinter

Tkinter is a standard Python library for creating graphical user interfaces. It provides an easy way to create buttons, dialogs, and windows for user interaction.

scipy

The scipy library is used for mathematical computations. In this project, it helps in calculating Euclidean distances which are essential for the Eye Aspect Ratio calculations.

imutils

imutils is a set of convenience functions to make basic image processing functions such as translation, rotation, resizing, and displaying images easier with OpenCV.

7. Conclusion

In conclusion, the Facial Recognition Attendance System represents a cutting-edge solution for modern attendance tracking needs. By harnessing the power of advanced image processing and machine learning algorithms, it offers unparalleled accuracy and efficiency in identifying individuals and recording their attendance.

Moreover, the system's adaptability and scalability make it suitable for various environments, ranging from educational institutions and corporate offices to large-scale events. Its user-friendly interface and seamless integration capabilities ensure a smooth user experience, while its customizable features allow for tailored implementation according to specific requirements.