

# FACE ATTENDANCE SYSTEM

---

## Project Overview

The Face Attendance System is an innovative application designed to automate the attendance process using facial recognition technology. The system utilizes Python, OpenCV, and machine learning techniques to register, recognize, and mark attendance based on facial data. This report outlines the project's objectives, methodology, implementation, and results.

## Objectives

- Develop a reliable and efficient system for marking attendance using facial recognition.
- Enable user registration by capturing and storing facial data.
- Recognize registered users in real-time through video feed.
- Log attendance records with timestamps for each recognized user.

## Methodology

### Tools and Libraries

- **Python:** Programming language used for developing the system.
- **OpenCV:** Library for real-time computer vision tasks.
- **NumPy:** Library for handling numerical operations.
- **CSV:** For storing and retrieving user and attendance data.

## Implementation Steps

1. **Initialization:**
  - Initialize video capture to access the webcam.
  - Load the Haar Cascade classifier for face detection.
  - Load previously stored known faces and names from files.
  - Load registered users from a CSV file.
2. **User Registration:**
  - Capture user's facial data through the webcam.
  - Detect faces using the Haar Cascade classifier.
  - Resize and store the detected face along with the user's name.
  - Save the registered data to disk and update the CSV file.
3. **Face Recognition:**
  - Continuously capture frames from the webcam.
  - Convert frames to grayscale for face detection.
  - Detect and recognize faces using template matching.
  - Compare detected faces with registered faces to identify users.

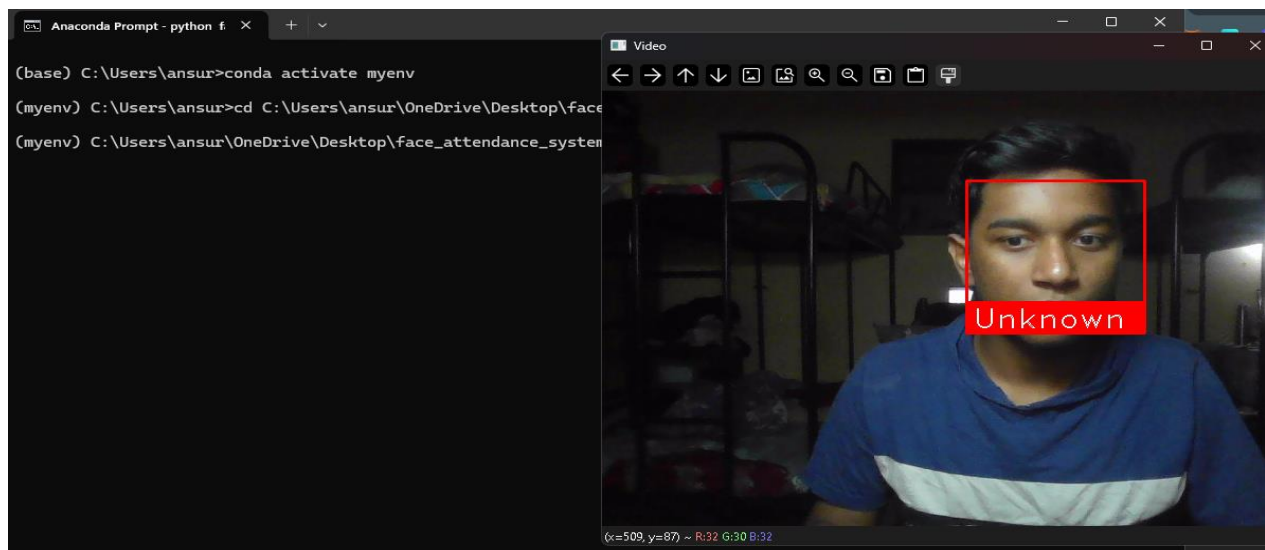
#### 4. Marking Attendance:

- Log recognized users' attendance with the current timestamp in a CSV file.
- Display the user's name on the video feed when recognized.

## Data Storage

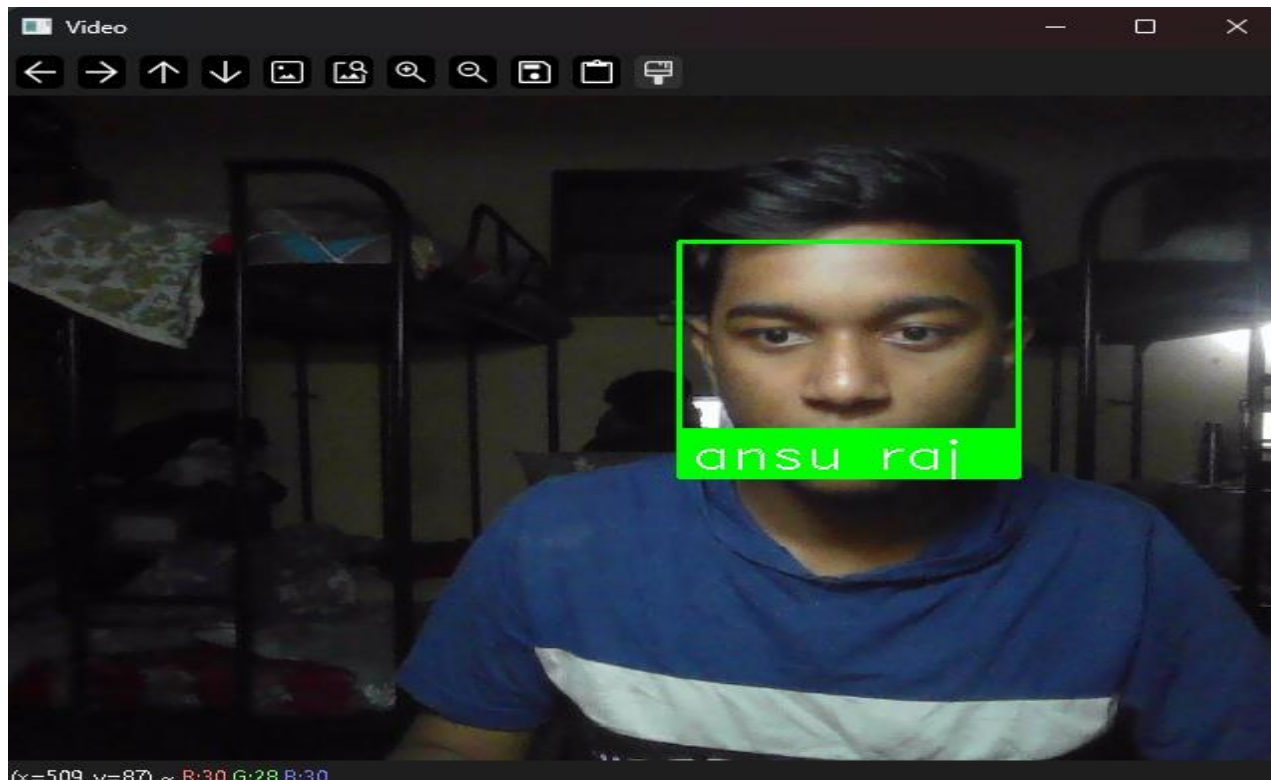
- **known\_faces.npy**: Numpy array file storing registered facial data.
- **known\_names.npy**: Numpy array file storing names of registered users.
- **registered.csv**: CSV file storing user registration details.
- **attendance.csv**: CSV file logging attendance records with timestamps.

## Implementation



```
(base) C:\Users\ansur>conda activate myenv
(myenv) C:\Users\ansur>cd C:\Users\ansur\OneDrive\Desktop\face_attendance_system
(myenv) C:\Users\ansur\OneDrive\Desktop\face_attendance_system>python face_attendance.py
Enter your name: ansu raj
Please look at the camera and press 's' to capture your face.
```

```
(base) C:\Users\ansur>conda activate myenv
(myenv) C:\Users\ansur>cd C:\Users\ansur\OneDrive\Desktop\face_attendance_system
(myenv) C:\Users\ansur\OneDrive\Desktop\face_attendance_system>python face_attendance.py
Enter your name: ansu raj
Please look at the camera and press 's' to capture your face.
Face registered successfully!
Attendance marked for ansu raj
(myenv) C:\Users\ansur\OneDrive\Desktop\face_attendance_system>s
```



```
registered.csv > data
1  ansu raj,2024-07-17 22:55:24.018272
2
3
```

```
attendance.csv > data
1  ansu raj,2024-07-17 22:57:04.611846
2
```

## Results

The Face Attendance System successfully performs the following functions:

- **Registration:** Users can register their faces, which are stored for future recognition.
- **Real-Time Recognition:** The system accurately recognizes registered users in real-time.
- **Attendance Logging:** Attendance is marked and logged with timestamps for recognized users.

## Conclusion

The Face Attendance System demonstrates an effective approach to automating the attendance process using facial recognition technology. The integration of Python, OpenCV, and machine learning techniques ensures accurate and efficient performance. Future enhancements could include improving recognition accuracy and integrating additional features such as email notifications and detailed reporting.