# 1. Title Page

**Project Title**: Face Recognition Attendance System  
**Submitted by**: *Vinay*  
**Technology Stack**: Python, OpenCV, face\_recognition, CSV  
**Date**: June 2025

# 2. Introduction

The Face Recognition Attendance System is a real-time solution that uses a webcam and machine learning techniques to automatically recognize faces and record attendance. Unlike traditional methods such as fingerprint scanners or manual signing, this system provides a contactless and faster way of maintaining attendance.

# 3. Problem Statement

Manual attendance systems are time-consuming, prone to error, and require physical interaction. The goal of this project is to automate the attendance process using computer vision and facial recognition to:

* Reduce human effort.
* Increase accuracy.
* Prevent proxy attendance.

# 4. Technologies Used

| **Component** | **Description** |
| --- | --- |
| **Python** | Primary programming language |
| **OpenCV** | Real-time video processing and drawing |
| **face\_recognition** | Face encoding, detection, and matching |
| **NumPy** | Array and distance calculations |
| **CSV** | For saving attendance data |
| **Jupyter Notebook** | Used for prototyping the project |

# 6. Code Explanation & Implementation

**✅ Step 1: Install Libraries**

pip install face-recognition

Installs the face\_recognition library which is built on top of dlib.

✅ Step 2: Import Modules

import face\_recognition, cv2, numpy as np, csv, os

from datetime import datetime

import time

These libraries are used for face detection, timekeeping, and file operations.

**✅ Step 3: Load Known Faces**

PHOTO\_DIR = "photos"

known\_face\_encodings = []

known\_face\_names = []

for filename in os.listdir(PHOTO\_DIR):

image = face\_recognition.load\_image\_file(os.path.join(PHOTO\_DIR, filename))

encodings = face\_recognition.face\_encodings(image)

known\_face\_encodings.append(encodings[0])

known\_face\_names.append(name)

This loop loads and encodes faces from the photos/ folder.

**✅ Step 4: Initialize Webcam and Start Face Recognition**

video\_capture = cv2.VideoCapture(0)

Activates the webcam. A loop reads each frame and performs recognition.

**✅ Step 5: Match Faces**

matches = face\_recognition.compare\_faces(known\_face\_encodings, face\_encoding)

Each detected face is compared to the known faces using Euclidean distance.

**✅ Step 6: Mark Attendance in CSV**

with open(csv\_filename, 'w', newline='') as f:

csv\_writer.writerow(['Name', 'Time'])

csv\_writer.writerow([name, current\_time])

When a known face is detected, the name and time are written into the CSV file.

**✅ Step 7: Display Results**

cv2.rectangle(...) # Draw bounding boxes

cv2.imshow("Face Recognition Attendance", frame)

Faces are displayed with bounding boxes and names on the webcam feed.

# 7. Results

The system accurately identified faces stored in the photos/ folder and marked them present in the CSV file. It could:

* Recognize multiple faces.
* Mark attendance once per person.
* Save logs with accurate timestamps.

# 8. Challenges

* **Lighting conditions**: Poor lighting affected recognition.
* **Camera warm-up**: Needed a slight delay before reading frames.
* **Face not detected**: Some images in the photos/ folder didn’t contain clear faces.

# 9. Future Scope

* Integrate with Google Sheets or cloud database.
* Add face re-training features using GUI.
* Use deep learning for more robust recognition.
* Deploy on Raspberry Pi for offline systems in classrooms.

# 10. Conclusion

The Face Recognition Attendance System is an efficient solution for automating attendance in classrooms or offices. It’s fast, contactless, and saves records systematically in CSV format, demonstrating practical usage of AI and computer vision.