**PROJECT REPORT**

**on**

**A FULLPROOF AUTOMATED ATTENDANCE SYSTEM**

**(CSE III Semester Mini project)**

**2023-2024**

****

**Submitted to: Submitted by:**

**Ms. Sonal Malhotra Malya Shukla**

**(Class Coordinator) Class Roll no-37**

**CSE Roll. No:2219039**

**Class-(C1) III-Sem**

**Session: 2023-2024**

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**GRAPHIC ERA HILL UNVERSITY, DEHRADUN**

**CERTIFICATE**

### Certified that Ms. Malya Shukla (Roll No.-2219039) has developed mini project on “A full proof automated Attendance System” for the CS III Semester Mini Project Lab in Graphic Era Hill University, Dehradun. The project carried out by Student is her own work as best of my knowledge.

Date:

Ms. SONAL MALHOTRA

**Class Co-ordinator**

**CSE-C1-III-Sem**

(CSE Department)

GEHU Dehradun

**ACKNOWLEDGMENT**

I would like to express my gratitude to The Almighty, the most Beneficent and the most Merciful, for completion of project.

I wish to thank my parents for their continuing support and encouragement. I also wish to thank them for providing me with the opportunity to reach this far in my studies.

I would like to thank particularly my project Co-ordinator **Ms Sonal Malhotra** and for her patience, support and encouragement throughout the completion of this project and having faith in me.

At last but not the least I greatly indebted to all other persons who directly or indirectly helped me during this work.

**Ms. Malya Shukla**

**Roll No.- 2219039**

**CSE-C1-III-Sem**

**Session: 2023-2024**

**GEHU, Dehradun**

**EXCEUTIVE SUMMARY:**

**AUTOMATED ATTENDANCE SYSTEM**

In the ever-evolving landscape of technology, traditional methods of attendance tracking in various settings, such as educational institutions, workplaces, and events, have faced challenges related to accuracy, efficiency, and security. To address these issues, the integration of face recognition technology into automated attendance systems has emerged as a cutting-edge solution.

Automated attendance systems utilizing face recognition leverage the power of artificial intelligence (AI) and computer vision to streamline the attendance tracking process. This innovative approach offers a host of advantages over traditional methods, including increased accuracy, real-time data processing, and a reduction in administrative burden.

The core principle behind face recognition technology lies in the ability to identify and verify individuals based on unique facial features. A facial recognition system captures, analyses, and compares facial patterns, allowing for the swift and accurate recognition of individuals within a database. This technology has seen significant advancements in recent years, making it both reliable and robust for applications like attendance tracking.

**THE PROJECT ON AUTOMATED ATTENDANCE SYSTEM USING FACE RECOGNITION IS BUILT BY USING MODULES AND LIBRARIES OF PYTHON ALONG WITH USING WEBCAM AND STORING THE ATTENDANCE IN A CSV FILE AND SENDING THE NOTIFICATION WHEN A STUDENT IS MARKED PRESENT.**

**1.Introduction**

**1.1 Purpose**

The primary purpose of this project is to streamline the attendance-taking process by automating the identification of students using facial recognition. The system captures video from a webcam, compares detected faces with a database of known faces, and records attendance in a CSV file.

**1.2 Scope**

The scope of the project covers face recognition, attendance logging, and database management. The system is designed to be scalable for use in educational institutions, reducing manual effort and improving accuracy.

**2.System Architecture**

**2.1 Technologies Used**

**OpenCV:** Capturing and processing video frames.

**face\_recognition:** A Python library for face recognition.

**numpy**: A library for numerical operations in Python.

**os:** A module for interacting with the operating system.

**datetime:** A module for working with dates and times in Python.

**plyer:** A library for sending desktop notifications.

**2.2 Workflow**

**Initialization:**

* **Establish a connection to the MySQL database.**
* **Create the 'attendance' table if it doesn't exist.**

**Face Recognition Setup:**

* **Load known face encodings and names from student photos.**
* **Initialize variables for face recognition.**

**Real-time Face Recognition:**

* **Capture video frames from the webcam.**
* **Detect faces and compare them with known faces.**
* **Mark attendance for recognized students.**

**Attendance Logging:**

* **Update the CSV file with the student's name and the time of attendance.**
* **Send a notification for each student marked present.**

**Display:**

* **Display real-time video with attendance status.**

**Termination:**

* **Release video capture and close windows.**
* **Close the CSV file at the end.**

**3. Implementation**

**3.1 Face Recognition**

The system uses the face\_recognition library to encode and compare faces. Known face encodings are loaded from student photos, and real-time face recognition is performed using OpenCV.

**3.2 Database Integration**

CSV file is employed for storing attendance data. The system updates the CSV file with the student's name and timestamp when attendance is marked.

**3.3 Notification System**

The Plyer library is utilized to send notifications for each student marked present, providing real-time updates.

**4. Results**

The system has been tested in various scenarios, demonstrating accurate face recognition and reliable attendance logging. The notification feature enhances user awareness of attendance updates.

**5. Challenges and Future Enhancements**

**5.1 Challenges**

* Lighting conditions and camera angles may affect face recognition accuracy.
* Limited scalability for a large number of students.

**5.2 Future Enhancements**

Integration with additional biometric features for improved accuracy.

Implementation of a web-based interface for easy administration.

Optimization for scalability in larger educational institutions.

**6. Conclusion**

The Automated Attendance System using Face Recognition provides an efficient and accurate method for tracking student attendance. The integration of face recognition, csv file and notifications make it a valuable tool for educational institutions seeking to automate attendance processes.

**7. References**

**face\_recognition Library: https://github.com/ageitgey/face\_recognition**

**OpenCV: https://opencv.org/**

**Plyer Library:** [**https://github.com/kivy/plyer**](https://github.com/kivy/plyer)

**SOURCE CODE:**

import face\_recognition

import cv2

import numpy as np

import csv

import os

from datetime import datetime

from plyer import notification

# Function to send notifications

def send\_notification(title, message):

    notification.notify(

        title=title,

        message=message,

        app\_icon=None,

    )

# Capture video from webcam

video\_capture = cv2.VideoCapture(0)

# Create empty lists to store face encodings and names

known\_face\_encodings = []

known\_face\_names = []

# Specify the folder containing student photos

students\_folder = "photos/students/"

# Iterate through files in the folder

for filename in os.listdir(students\_folder):

    if filename.lower().endswith((".jpeg", ".jpg", ".png")):

        student\_image = face\_recognition.load\_image\_file(os.path.join(students\_folder, filename))

        face\_encodings = face\_recognition.face\_encodings(student\_image)

        if face\_encodings:

            student\_encoding = face\_encodings[0]

        else:

            print(f"No face found in {filename}. Skipping...")

            continue

        known\_face\_encodings.append(student\_encoding)

        known\_face\_names.append(os.path.splitext(filename)[0])  # Use filename without extension as the name

# List to keep track of students

students = list(known\_face\_names)

# Set up CSV file for attendance

now = datetime.now()

current\_date = now.strftime("%Y-%m-%d")

csv\_file\_name = f"attendance\_{current\_date}.csv"

with open(csv\_file\_name, 'w+', newline='') as f:

    lnwriter = csv.writer(f)

    # Add headers to the CSV file

    lnwriter.writerow(["Student Name", "Time Present", ""])

    while True:

        \_, frame = video\_capture.read()

        small\_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

        rgb\_small\_frame = small\_frame[:, :, ::-1]

        face\_locations = face\_recognition.face\_locations(rgb\_small\_frame)

        face\_encodings = face\_recognition.face\_encodings(rgb\_small\_frame, face\_locations)

        face\_names = []

        for face\_encoding in face\_encodings:

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

            name = ""

            face\_distance = face\_recognition.face\_distance(known\_face\_encodings, face\_encoding)

            best\_match\_index = np.argmin(face\_distance)

            if matches[best\_match\_index]:

                name = known\_face\_names[best\_match\_index]

            face\_names.append(name)

            if name != "" and name in students and students:

                font = cv2.FONT\_HERSHEY\_SIMPLEX

                bottomLeftCornerOfText = (10, 100)

                fontScale = 1.5

                fontColor = (0, 255, 0)

                thickness = 3

                lineType = 2

                cv2.putText(frame, f'{name} Present',

                            bottomLeftCornerOfText,

                            font,

                            fontScale,

                            fontColor,

                            thickness,

                            lineType)

                students.remove(name)

                print(students)

                current\_time = now.strftime("%H-%M-%S")

                lnwriter.writerow([name, current\_time, ""])

                # Send a notification when a student is marked present

                notification\_title = "Attendance Update"

                notification\_message = f"{name} marked present at {current\_time}"

                send\_notification(notification\_title, notification\_message)

        cv2.imshow("attendance system", frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):

            break

# Release video capture and close windows

video\_capture.release()

cv2.destroyAllWindows()

import face\_recognition

import cv2

import numpy as np

import csv

import os

from datetime import datetime

from plyer import notification

# Function to send notifications

def send\_notification(title, message):

    notification.notify(

        title=title,

        message=message,

        app\_icon=None,

    )

# Capture video from webcam

video\_capture = cv2.VideoCapture(0)

# Create empty lists to store face encodings and names

known\_face\_encodings = []

known\_face\_names = []

# Specify the folder containing student photos

students\_folder = "photos/students/"

# Iterate through files in the folder

for filename in os.listdir(students\_folder):

    if filename.lower().endswith((".jpeg", ".jpg", ".png")):

        student\_image = face\_recognition.load\_image\_file(os.path.join(students\_folder, filename))

        face\_encodings = face\_recognition.face\_encodings(student\_image)

        if face\_encodings:

            student\_encoding = face\_encodings[0]

        else:

            print(f"No face found in {filename}. Skipping...")

            continue

        known\_face\_encodings.append(student\_encoding)

        known\_face\_names.append(os.path.splitext(filename)[0])  # Use filename without extension as the name

# List to keep track of students

students = list(known\_face\_names)

# Set up CSV file for attendance

now = datetime.now()

current\_date = now.strftime("%Y-%m-%d")

csv\_file\_name = f"attendance\_{current\_date}.csv"

with open(csv\_file\_name, 'w+', newline='') as f:

    lnwriter = csv.writer(f)

    # Add headers to the CSV file

    lnwriter.writerow(["Student Name", "Time Present", ""])

    while True:

        \_, frame = video\_capture.read()

        small\_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

        rgb\_small\_frame = small\_frame[:, :, ::-1]

        face\_locations = face\_recognition.face\_locations(rgb\_small\_frame)

        face\_encodings = face\_recognition.face\_encodings(rgb\_small\_frame, face\_locations)

        face\_names = []

        for face\_encoding in face\_encodings:

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

            name = ""

            face\_distance = face\_recognition.face\_distance(known\_face\_encodings, face\_encoding)

            best\_match\_index = np.argmin(face\_distance)

            if matches[best\_match\_index]:

                name = known\_face\_names[best\_match\_index]

            face\_names.append(name)

            if name != "" and name in students and students:

                font = cv2.FONT\_HERSHEY\_SIMPLEX

                bottomLeftCornerOfText = (10, 100)

                fontScale = 1.5

                fontColor = (0, 255, 0)

                thickness = 3

                lineType = 2

                cv2.putText(frame, f'{name} Present',

                            bottomLeftCornerOfText,

                            font,

                            fontScale,

                            fontColor,

                            thickness,

                            lineType)

                students.remove(name)

                print(students)

                current\_time = now.strftime("%H-%M-%S")

                lnwriter.writerow([name, current\_time, ""])

                # Send a notification when a student is marked present

                notification\_title = "Attendance Update"

                notification\_message = f"{name} marked present at {current\_time}"

                send\_notification(notification\_title, notification\_message)

        cv2.imshow("attendance system", frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):

            break

# Release video capture and close windows

video\_capture.release()

cv2.destroyAllWindows()