EXTENSHIP PROJECT

TITLE:

SMART ATTENDANCE SYSTEM USING FACIAL RECOGNITION

TEAM MEMBERS :

1.S Sai Vignesh Reddy

2.Sai Dixith Villa

3.Narapareddy Abhilash Reddy

4. Javed Mohamed Ashiq

1 INTRODUCTION:

As an era of modern technology has becoming more digital making world more simple according to the data storage and data transformation which makes people more simple with the data and easy maintaining reduces the time ,without the physical or manual enter, example in the schools are in the offices analysing the student data and in the offices salary with manual attendance takes lot of time. There is also a problem of getting a hardcopy of the attendance records when the attendance sheet may get misplace. This all will be misplaced by digital ***Facial Recognition System*** . As some of the attendance systems with other technologies. Fingerprint-based attendance: system, students fingerprints must be collected and recognised by a portable biometric device in order to record their attendance. RFID cards: Students must show their RFID cards to an ID card reader in order for the RFID-based attendance system to record their attendance.

As with facial recognition system will be make the application more accurate than other models .

In the project we are building a live attendance system using facial recognition using deep learning models with live image capturing and with that record attendance in various settings such as schools, offices, or events will be easy. It involves capturing and analyzing the unique facial features of individuals to accurately identify and verify their presence.

2 .LITERATURE SURVEY:

2.1 PROBLEM STATEMENT

Concept In a classroom with large number of students, and in office with large numbers of employees , it is a very difficult and time-consuming task to take the attendance manually every day . It provides the data of the students or employee As we got some attendance system like biometric which has so many concerns with the modern world as recent Covid -19 effect, know a days .Users are generally more comfortable with facial recognition as it does not require physical contact . As manual attendance needed more attention recording maintainability and storage is a big ,know a days .

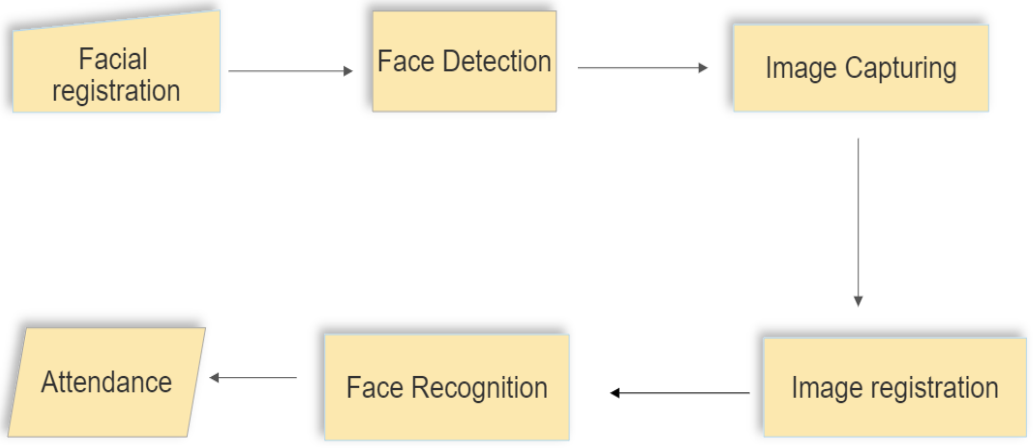
Therefore, we can implement an effective system which will mark the attendance of students automatically by recognizing their faces. The process of this face recognition is divided into various steps, but the important steps are detection of face and recognition of face.

2.2 PROPOSED SOLUTION

To develop an automated attendance system using face recognition. All the students of the class or in the office must register themselves by entering the required details like their name and registration number and then their images will be captured and stored in the dataset with the help of open cv library. During each session, faces will be detected from live streaming video of classroom or in the office . The faces detected will be compared with images present in the dataset. If match found, attendance will be marked for the respective student or employee. The task of the proposed system is to capture the face of each student and to store it in the database for their attendance. The face of the student is captured in such a manner that all the feature of the students face needs to be detected. There is no need for the teacher to manually take attendance in the class because the system records a video and through further processing steps the face is being recognized and the attendance is updated in the excel sheet. With that feature if teacher what to see the full list of students are present in the class at the end of the session the teacher can download the saved file respective day as the document contain the details of the students with name, registration number and particular date and time of the present student

3. THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM:



3.2 HARDWARE REQURIMENTS :

- ✓ Dell Laptop with 8 GB RAM or above
- ✓ Minimum Storage Capacity: 5GB
- ✓ Camera 720p or above
- ✓ Ref lap

SOFTWARE REQURIMENTS :

- ✓ PyCharm and Google collab
- ✓ Microsoft Office
- ✓ Flask
- ✓ Open cv and Dlib Library
- ✓ Cuda application

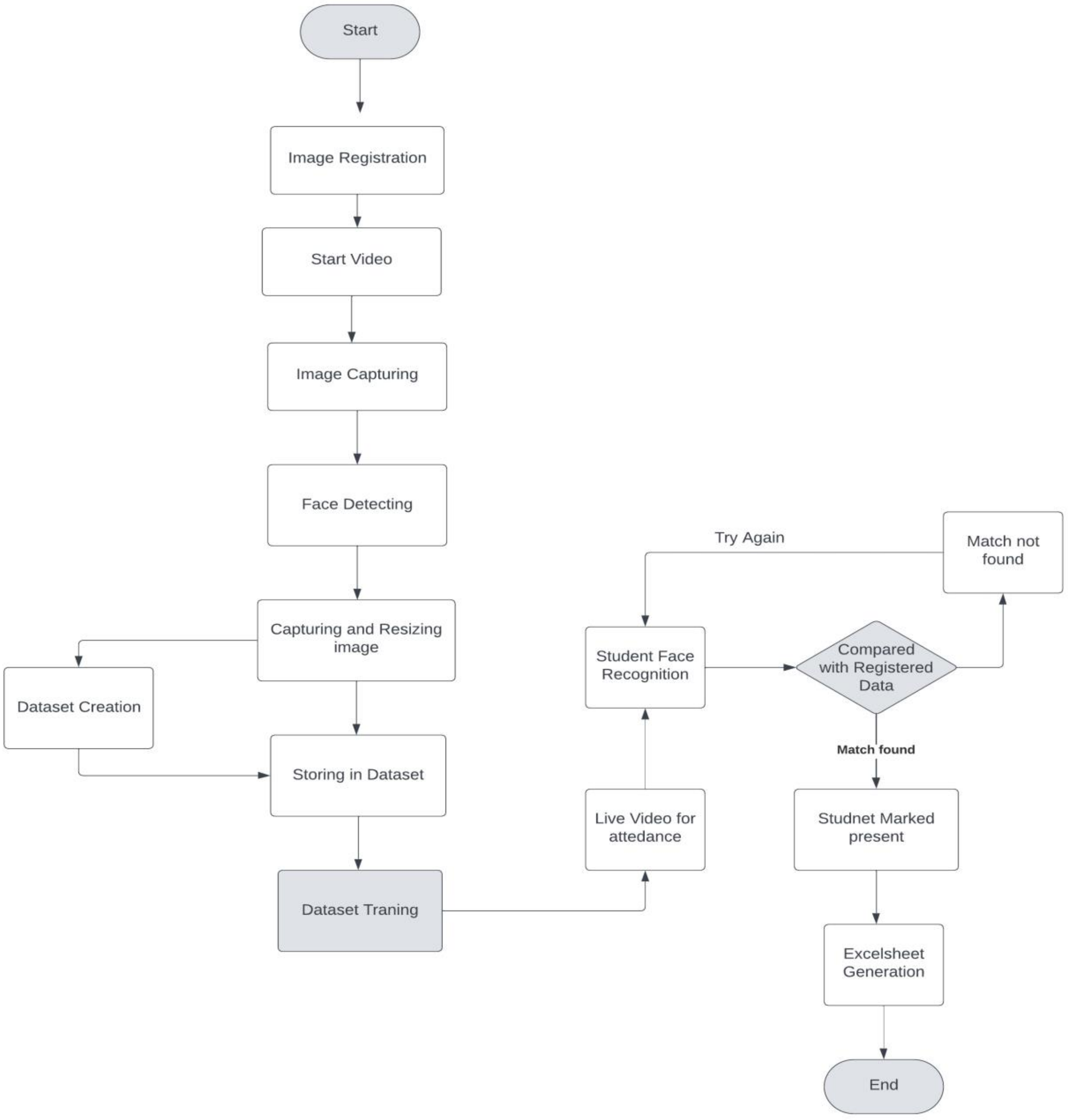
4. EXPERIMENTAL INVESTIGATIONS

During the development and implementation of your project smart attendance system using facial recognition, researchers some development has be conducted to evaluate the system's performance, accuracy, and usability. Here are some common areas of experimental investigation we made of your project

Algorithm Selection is the main aspect of creating a model and in the smart attendance system it the big aspect of choosing the best algorithm which is suitable for the facial recognition in an accurate way as there are Traditional and Deep learning algorithm as choosing traditional model will limited discriminative power, meaning they may not be able to effectively identify the individuals with similar facial features So as we are doing this project with Deep learning algorithm such as CNN

Accuracy Assessment is the best way to maintain a model is working with the best possible way It involves measuring the system ability to correctly identify individuals and measuring the accuracy of the recognition process. Preprocessing techniques, such as face alignment, normalization, and noise reduction, may be applied to the collected dataset to standardize the input data and improve recognition accuracy which should be investigated and identifies person and false occur when the system does not identifies the person. Images in dataset should include variations images in lighting conditions, facial expressions, angles, and other factors that may impact recognition accuracy.

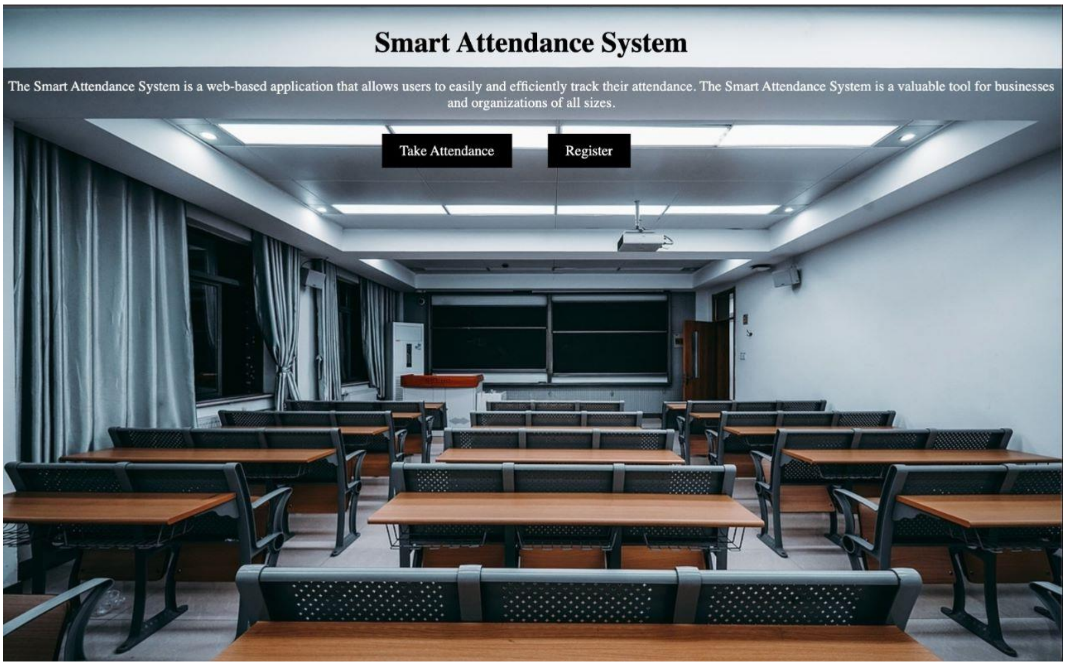
5 .FLOWCHART



FLOW OF SMART
ATTEDANCE SYSTEM

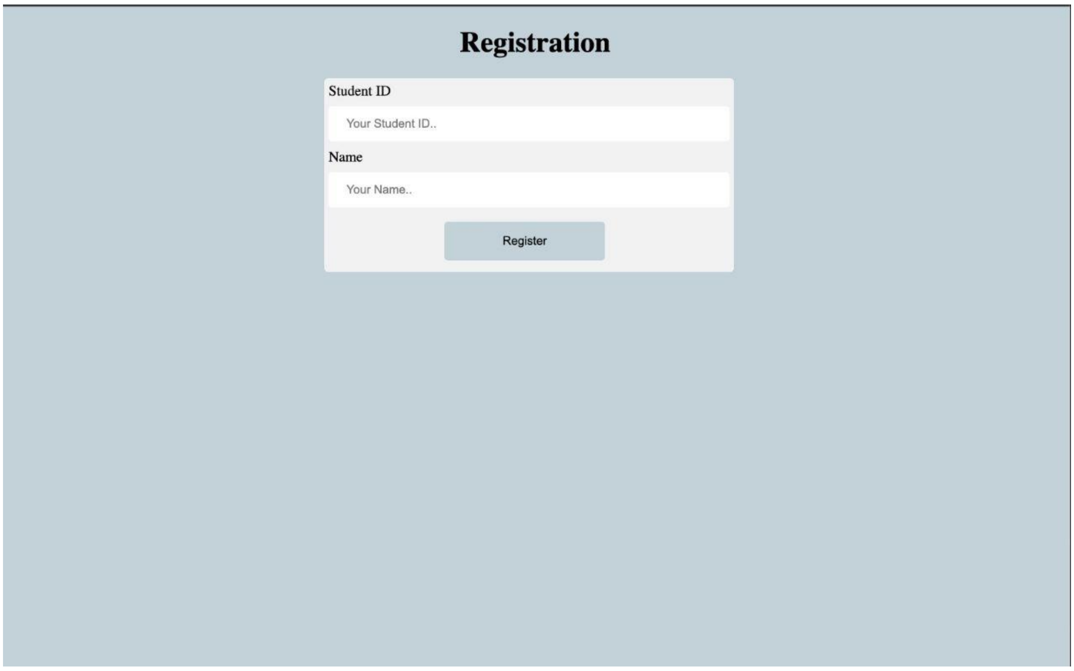
6 RESULTS:

- ## Registration:



Smart Attendance System

The Smart Attendance System is a web-based application that allows users to easily and efficiently track their attendance. The Smart Attendance System is a valuable tool for businesses and organizations of all sizes.



Registration

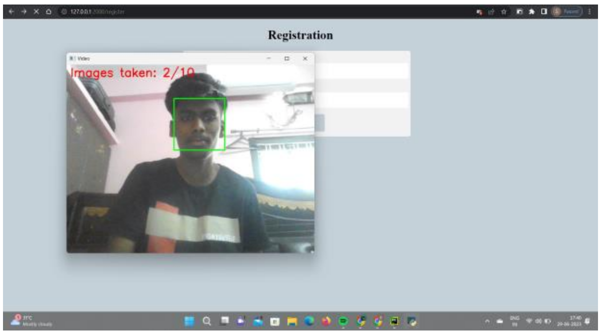
Student ID

Your Student ID..

Name

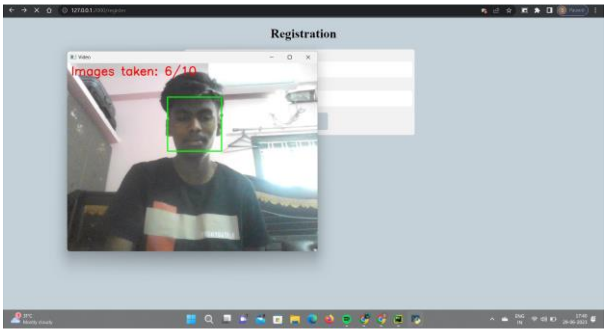
Your Name..

If a student has to register themselves, they first need to finish the registration process where a sample of 10 images are captured in real-time as shown below.



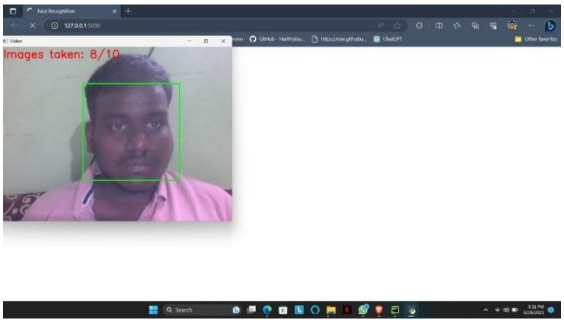
Registration

Images taken: 2/10



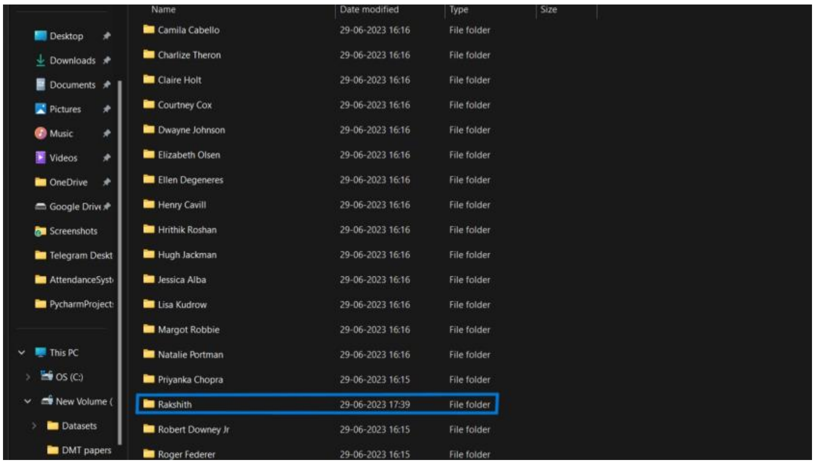
Registration

Images taken: 6/10

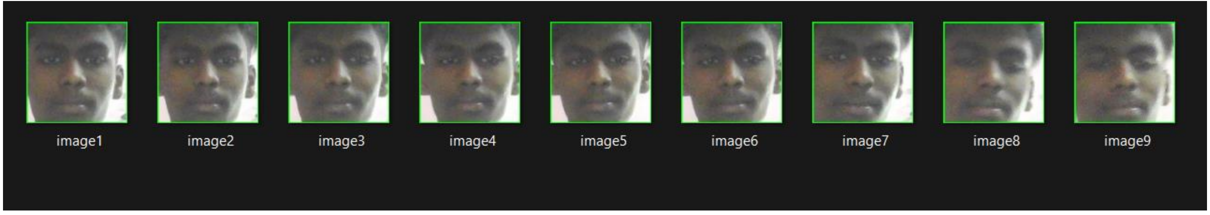


Images taken: 8/10

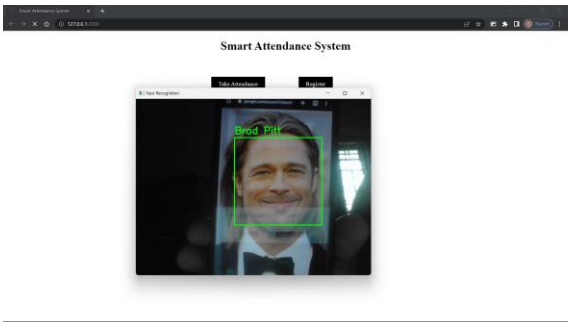
- A distinct folder is created for every registered student where the sample images are divided into training and testing process and the later model is trained to identify the student.

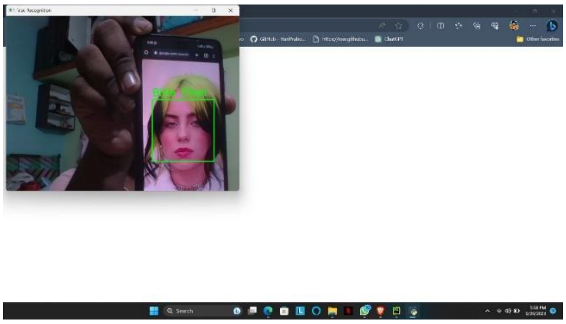


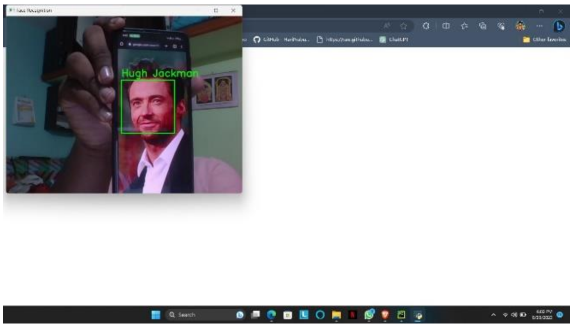
Name	Date modified	Type	Size
Camila Cabello	29-06-2023 16:16	File folder	
Charlize Theron	29-06-2023 16:16	File folder	
Claire Holt	29-06-2023 16:16	File folder	
Courtney Cox	29-06-2023 16:16	File folder	
Dwayne Johnson	29-06-2023 16:16	File folder	
Elizabeth Olsen	29-06-2023 16:16	File folder	
Ellen Degeneres	29-06-2023 16:16	File folder	
Henry Cavill	29-06-2023 16:16	File folder	
Hrithik Roshan	29-06-2023 16:16	File folder	
Hugh Jackman	29-06-2023 16:16	File folder	
Jessica Alba	29-06-2023 16:16	File folder	
Lisa Kudrow	29-06-2023 16:16	File folder	
Margot Robbie	29-06-2023 16:16	File folder	
Natalie Portman	29-06-2023 16:16	File folder	
Priyanka Chopra	29-06-2023 16:15	File folder	
Rakshith	29-06-2023 17:39	File folder	
Robert Downey Jr	29-06-2023 16:15	File folder	
Roger Federer	29-06-2023 16:15	File folder	

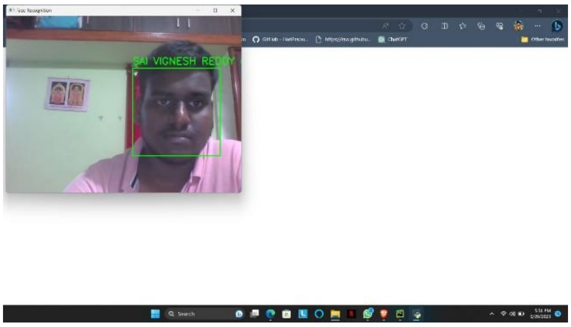


- As we can see the model is detecting the faces as registerd

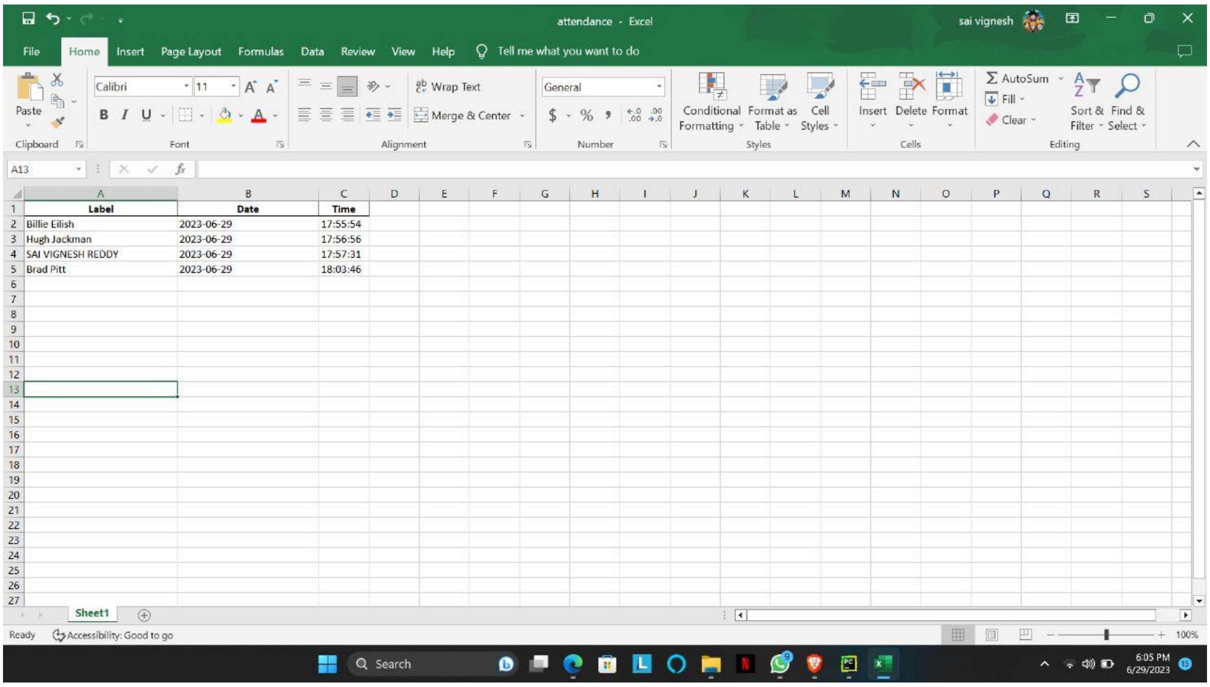








- Data is stored in the excel sheet with name of the student, date and time



Label	Date	Time
Billie Eilish	2023-06-29	17:55:54
Hugh Jackman	2023-06-29	17:56:56
SAI VIGNESH REDDY	2023-06-29	17:57:31
Brad Pitt	2023-06-29	18:03:46

7 .ADVANTAGES & DISADVANTAGES

Advantages:

- ✓ Eliminates the need for manual attendance
- ✓ Less time consuming
- ✓ With unique facial features it provide hight accuracy of find the individuals
- ✓ It provide extra security of fake attendances
- ✓ Provides the attendance date more precisely with adding date and time when the attendance is taken

Disadvantages:

- ✓ Inaccurate facial recognition
- ✓ Poor environmental conditions such as dull lighting conditions which makes the model more inaccurate to prediction
- ✓ The system involve significant costs including the acquisition of specialized hardware, cameras which makes difficult to some schools and industries
- ✓ Less privacy

8.APPLICATIONS:

Academy :

The main purpose of Smart attendance system is Schools, colleges, and universities can benefit from facial recognition-based attendance systems. It simplifies the attendance process from both students and faculty, eliminates the need for manual attendance, and reduces the chances of errors or proxy by the students.

Healthcare Center :

Some of the hospitals ,health care institutions need this kind of assistance for maintaining employees facial recognition will track the presence of staff and will ensure the proper maintenance of staff members improves efficiency in health care .

9.CONCLUSION:

This system aims to build an effective class attendance system using face recognition techniques. The proposed system will be able to mark the attendance via face Id. Facial recognition has been a aspiring topic for many developed for a long time. To achieve good accuracy in a facial recognition model one must keep on experimenting with the data and algorithms. To improve a facial recognition model one can input more number of pictures of a person into the dataset. It allows the model to compare with a wide variety of pixel matrices. One can also input pictures having different environment settings with different lighting it will help the model train in different environments and increase precision

10.FUTURE SCOPE:

Some enhancements can be made is :

As last two years we have suffered from the covid 19 from the time face mask has be mandatory to all the people masked face recognition can be introduced with developing algorithms that can accurately recognize faces even when partially covered can be a significant advancement.

And incorporating some of the best Ai techniques will make the system much more accurate to recognising the people and objects more seamlessly wile coming days technology is improving constantly which makes a good sign of finding new Deep learning models for further enhancement of the face recognition.

11.BIBILOGRAPHY

[1] B.K. Mohamed and C. Raghu, "Fingerprint attendance system for classroom needs," India Conference (INDICON), Annual IEEE, 2012, pp. 433-438.

[2] S. N. Shah and A. Abuzneid, "IoT based smart attendance system (SAS) using RFID," IEEE Long Island Systems, Applications and Technology Conference (LISAT), 2019.

[3] A. Khatun, A.K.M.Fazlul Haque, S. Ahmed, and M. M. Rahman, "Design and implementation of Iris recognition based attendance management system," 2nd Int'l Conf. on Electrical Engineering and Information & communication Technology (ICEEICT) 2015, Bangladesh.

[4] S.Sawhney, K.Kacker, S. Jain, S. N.Singh, and R.Garg, "Real-time smart attendance system using face recognition techniques,", 9th Int'l Conf. on Cloud Computing, Data Science & Engineering, 2019, pp. 522-525.

[5] S.Sveleba, I. Katerynchuk, I.Karpa, I.Kunyo, S.Ugryn, and V.Ugryn, "The real time face recognition," 3rd Int'l Conf. on Advanced Information and Communication Technologies, 2019, pp.294-297.

[6] R.Nandhini, N. Duraimurugan, and S.P.Chokkalingam, "Face recognition based attendance system," Int'l Journal of Engineering and Advanced Technology (IJEAT), Vol-8, Issue-3S, February 2019, pp.574-577

SOURCE CODE:

```
from flask import Flask, render_template, request, redirect, url_for
import os
import cv2
import numpy as np
from keras.models import load_model
import datetime
import pandas as pd

app = Flask(_name_)


def create_folder(path, name):
    folder_path = os.path.join(path, name)
    os.makedirs(folder_path, exist_ok=True)
    return folder_path


def capture_images(folder_path, num_images, frame_interval):
    cap = cv2.VideoCapture(0)

    face_cascade =
cv2.CascadeClassifier('data/haarcascade_frontalface_default.xml.xml')

    image_count = 0
    frame_count = 0

    while True:
        ret, frame = cap.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5,
minSize=(30, 30))

        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

            if frame_count % frame_interval == 0 and image_count < num_images:
                face_image = frame[y:y + h, x:x + w]

                image_path = os.path.join(folder_path, f"image{image_count + 1}.jpg")
                cv2.imwrite(image_path, face_image)
                image_count += 1

        cv2.putText(frame, f"Images taken: {image_count}/{num_images}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                    (0, 0, 255), 2)
        cv2.imshow("Video", frame)

        key = cv2.waitKey(1) & 0xFF
        if key == ord('q') or image_count >= num_images:
            break

        frame_count += 1

    cap.release()
    cv2.destroyAllWindows()

    return image_count


def mark_attendance(label, attendance_df):
    now = datetime.datetime.now()
```

```
 date_string = now.strftime("%Y-%m-%d")
  time_string = now.strftime("%H:%M:%S")

  if label not in attendance_df['Label'].values:
      attendance_df.loc[len(attendance_df)] = [label, date_string, time_string]


def recognize_person(frame, model, face_cascade, label_mapping, attendance_df):
  gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

  faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5,
minSize=(30, 30))

  for (x, y, w, h) in faces:
      face_image = gray[y:y + h, x:x + w]
      face_image = cv2.cvtColor(face_image, cv2.COLOR_GRAY2RGB)
      face_image = cv2.resize(face_image, (224, 224))

      face_image = np.expand_dims(face_image, axis=0)
      face_image = face_image / 255.0

      predictions = model.predict(face_image)
      person_index = np.argmax(predictions)
      confidence = predictions[0][person_index]

      if confidence > 0.5:
          label = label_mapping[person_index]
          mark_attendance(label, attendance_df)
          cv2.putText(frame, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9,
(0, 255, 0), 2)
          cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

  return frame
@app.route('/')
def home():
  return render_template('index.html')
@app.route('/register', methods=['GET', 'POST'])
def register():
  if request.method == 'POST':
      name = request.form['name']
      id = request.form['id']
      # Process the registration data and perform necessary actions
      path = "D:/face"
      folder_path = create_folder(path, name)
      num_images = 10
      frame_interval = 10
      image_count = capture_images(folder_path, num_images, frame_interval)
      print(f"{image_count} images captured and stored in folder: {folder_path}")
      return "Registration complete"
      filename = 'venv/model.py'
```

```
        exec(open(filename).read())
    return render_template('register.html')


@app.route('/attendance', methods=['GET', 'POST'])
def attendance():
    model = load_model("data/face.h5")
    face_cascade =
cv2.CascadeClassifier('data/haarcascade_frontalface_default.xml.xml')

    dataset_path = 'D:/face'
    label_mapping = {}
    folders = os.listdir(dataset_path)
    for i, folder in enumerate(folders):
        label_mapping[i] = folder
    attendance_df = pd.DataFrame(columns=['Label', 'Date', 'Time'])

    cap = cv2.VideoCapture(0)

    while True:
        ret, frame = cap.read()

        result_frame = recognize_person(frame, model, face_cascade, label_mapping,
attendance_df)

        cv2.imshow("Face Recognition", result_frame)

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

    cap.release()
    cv2.destroyAllWindows()

    file_path = 'data/attendance.xlsx'

    attendance_df.drop_duplicates(subset=['Label'], inplace=True)  # Remove duplicate
entries

    attendance_df.to_excel(file_path, index=False)

    print("Attendance saved to:", file_path)

    return 'Attendance marked'


if _name_ == '_main_':
    app.debug = True
    app.run(host='127.0.0.1', port=2000)
```