
VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

**School of Computer Science and Engineering (SCOPE)**

**B.Tech – Computer Science and Engineering with Specialization in Artificial Intelligence and Robotics**

**Fall Semester 2022-23**

**November, 2022**

*A project report on*

# AUTOMATED ATTENDANCE MARKER

*Submitted in partial fulfillment for the J Component project of*

**ECE3501 IOT FUNDAMENTALS**

*by*

**NETHRA SAI M (20BRS1092)**

**GANTI VENKATA VARSHINI(20BRS1046)**

**VIGNESH SURENDRAN(20BRS1217)**

**ISHAAN SAHAY(20BRS1042)**

Signature of the Candidate

Signature of the Faculty

## TABLE OF CONTENTS

# 1. ABSTRACT

This project is based on the study and the implementation of the Automated attendance marker project for customized, automated, highly secured class room management system known as fully automated classroom attendance system. Through this project, the strategies, design, development and implementation of an automated attendance marker project for class room attendance management systems is explained in detail.

Automated attendance marker is an autonomous robot which can move in the classroom and mark attendance of students using facial recognition . Prototype of the proposed system provides a high level of authentication by embedding face recognition and obstacle avoidance. Though given model features are specially designed for our college needs, this model can be modified for general usage and can be applied for all academic levels i.e. school, college, universities.

In every field Robots have a scope for their presence and complete the tasks given. The area we are going to introduce robots is educational institutions.

For every Educational institution there are Attenders who are being used for taking attendance, passing circulars and calling students/faculty from class rooms. Automated Attendance marker can perform the above tasks in place of humans with speed and accuracy.

For taking attendance the latest process used is Biometric and for circular, calling process attender is being used.

After taking attendance it needs to be updated in the database of the institution which is another burden for the database manager of the institution. Multiple attenders need to be used to complete circular and calling tasks; this is another cost and time consuming process.

## 2. INTRODUCTION

Many clinical advances and technology had been created on this current epoch of automation to store time, enhance accuracy, and decrease charges enhance the high-satisfactory of our lifestyles System of automatic assistance is the development that has been made withinside the discipline of conventional jobs will get replaced through automation Call motion on one presence. Assistive generation Bio-metrics are typically utilized in those structures. data, web-primarily based totally and primarily based totally on clever playing cards These In loads of organizations, structures are typically employed. The conventional manner of calculating attendance time is as follows: When the energy is not there, it is time-ingesting and complicated.

The automatic attendance device has a bonus over the traditional approach due to the fact It saves time and also can be applied for safety purposes. This additionally aids withinside the prevention of faulty assistance. A well-advanced attendance monitoring device The utilization of bio metrics, that's steeply-priced in our situation, generally includes photograph seizure and database management. Face detection, pre-processing, and improvement of characteristic extraction and class After then, the post-processing segment is completed. Machine studying strategies are used. We can take automated attendance on paper. Convolutional Neural Networks are used to create this model. Face recognition set of rules withinside the classroom.

In the current time automated face recognition has emerge as a fashion and has been evolved very much , that is in particular because of reasons; first it's far because of availability of contemporary-day technology and 2nd is because of the cap potential to shop time the use of face recognition withinside the technique of taking attendance of college students.
Its utilization will develop widespread withinside the destiny because it saves a number of time. It consumes a number of time to take attendance manually and few may additionally faux the attendance, in an effort to save you time intake and keep away from faking the attendance face recognition is used to perceive the man or woman gift withinside the magnificence and mark his attendance , that is achieved

with the assist of picture or video frame. We proposed an automated attendance control machine the use of device mastering strategies along with CNN set of rules.

The face detection and recognition will robotically hit upon the scholars withinside the school room and mark the attendance through spotting the man or woman.. The college has get entry to feature the pupil info along with name, USN, phone number, email-id. Then the picture is captured through an excessive definition digital digicam for the duration of the magnificence hours. When the lecture goes on, faces of college students are detected, segmented and saved for verification with database the use of the Convolutional Neural Networks (CNN) set of rules of device mastering technique.

## 3. PROPOSED SYSTEM

The suggested system’s goal is to build an autonomous wheel robot which can move around in the class room avoiding obstacle and capture the images of students to mark their attendance using facial recognition.

The algorithm used for this system is Histogram of oriented gradients(HOG) and Convolutional Neural Network(CNN).

The student's face must be captured in such a way that all of the student's facial traits, including the student's location and posture, are detected. The teacher does not need to physically take attendance in the classroom because the technology takes a video, which is then processed to recognize the face and update the attendance database. The key benefit of the system is that the faculty doesn’t need to move around the class since the autonomous robot is designed to do that and a text file sent via mail with time and date which makes it easy for the faculty to identify the time of class and mark attendance.

1. Image capture: The students' images are recorded and forwarded to be detected using face detection.

2. Face Detection: Using a face detection technique, the face identification process is greatly improved.

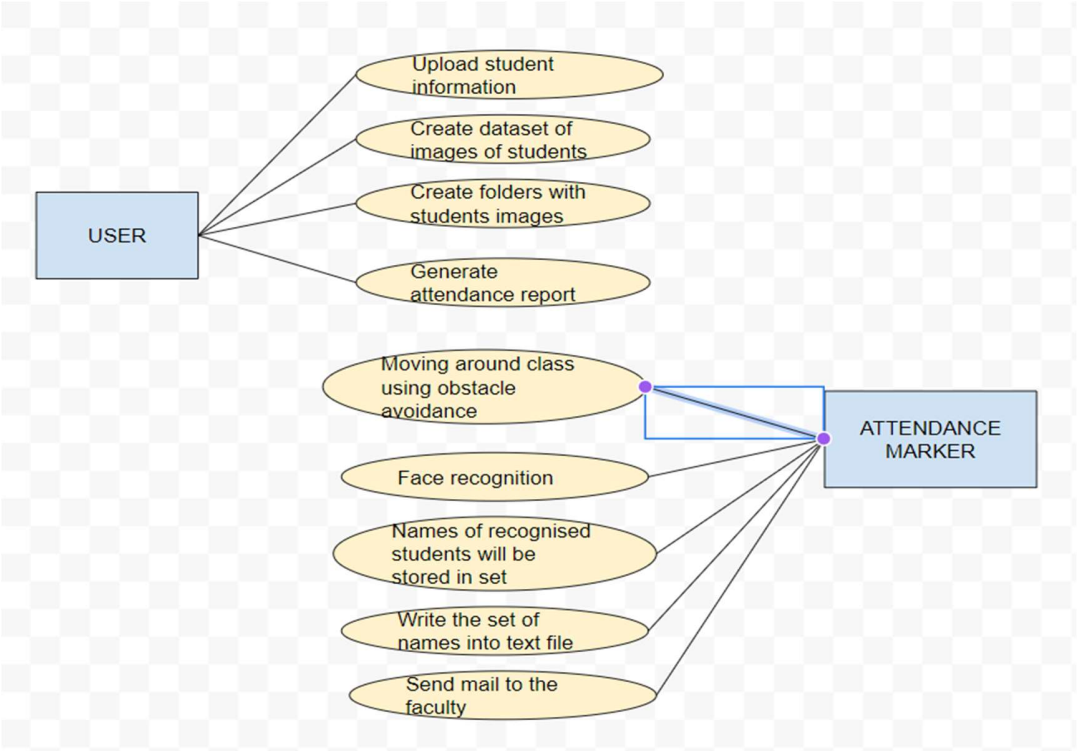
3. Obstacle avoidance: The system is built with an obstacle avoidance module which allows the system to move around in the classroom to move freely.

# 4. IMPLEMENTATION

All students in the class are marked present, and a database in the form of a csv file is delivered to the email address provided, along with the date the attendance was taken.

The smart attendance system has been demonstrated to be an effective classroom attendance system. This technique is non-intrusive and helps to eliminate the use of proxies and fraudulent attendance. Many ways to a smart attendance system have been presented, but the optimum option for a smart attendance system has been discovered to be a face recognition-based approach. It is necessary to have a dependable method for taking attendance. We have successfully applied the same in a classroom. Our system is simple to use attendance marker built using raspberry pi, arduino and usb camera.

## 4.1 **USE-CASE DIAGRAM**



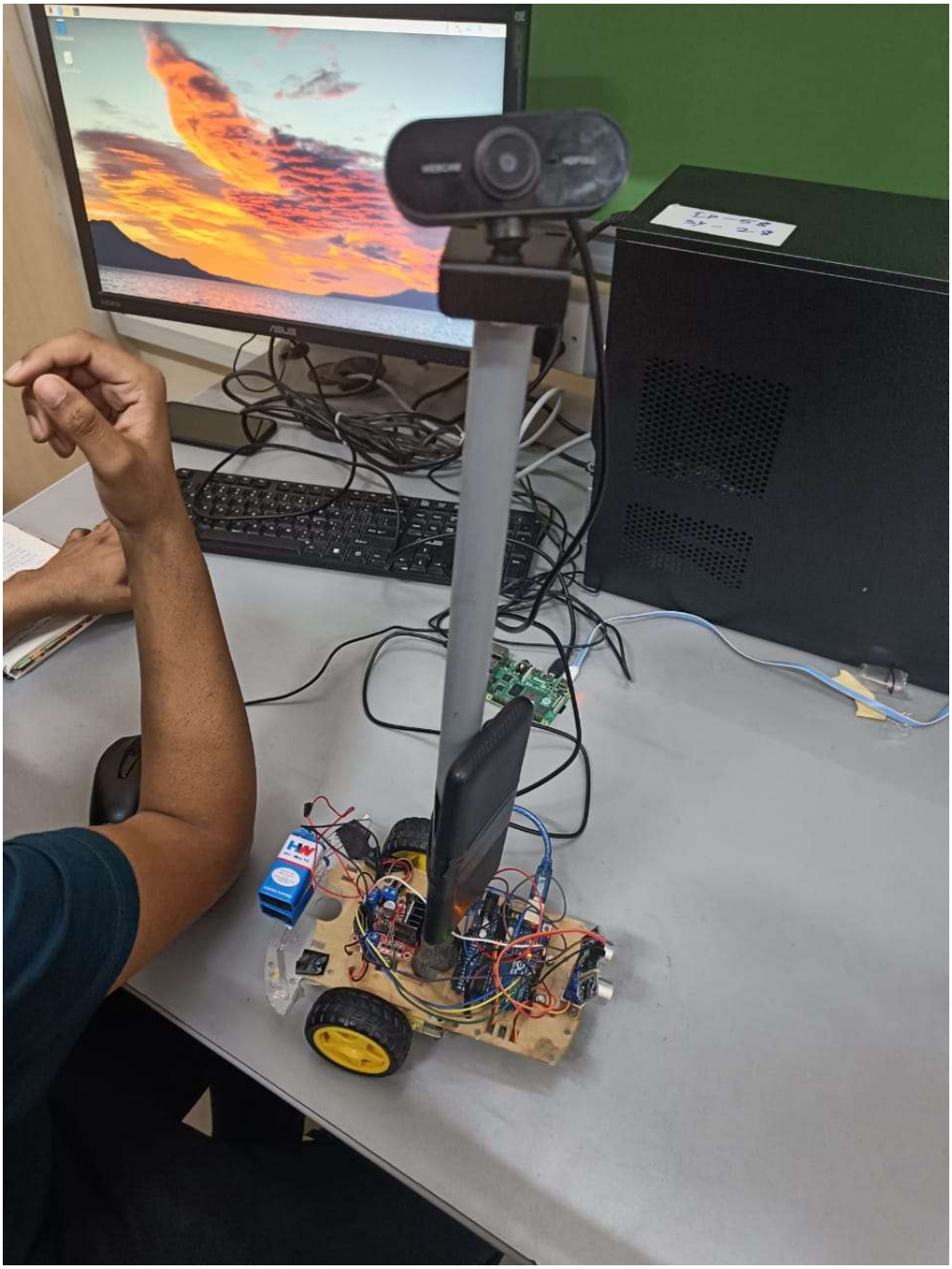
# 5. MODULES

## 5.1 OBSTACLE AVOIDANCE MODULE
**(Implemented by Vignesh Surendran**
**Code - Vignesh Surendran and Ishaan Sahay)**

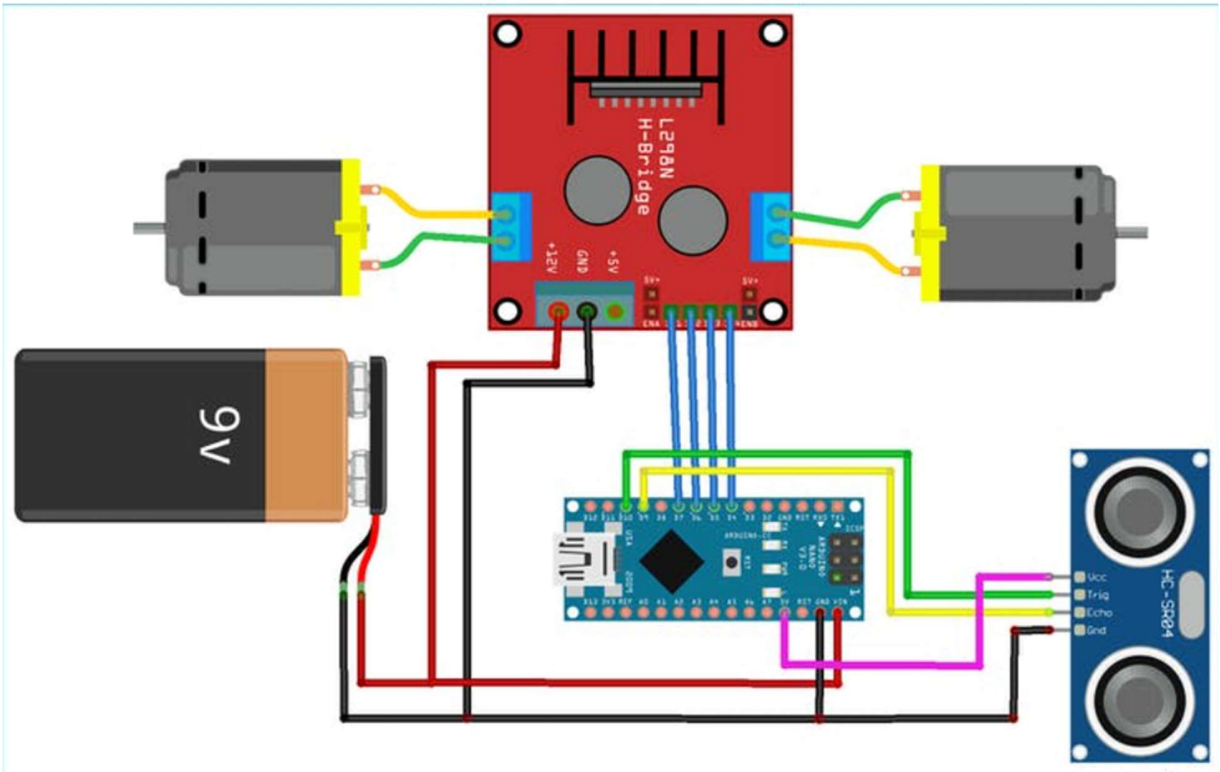
For the system to move in the classroom without clashing with the benches and chairs we have built an obstacle avoidance robot.
We have used a chassis on which an ultrasonic sensor is placed along with an arduino and motor drive for the obstacle avoidance part.
The ultrasonic sensor detects the presence of an object which can possibly block the path of the chassis and changes its direction accordingly.
The basic working of obstacle avoidance relies on ultrasonic sensor. The basic principle behind the working of an ultrasonic sensor is to note down the time taken by the sensor to transmit ultrasonic beams and receive the ultrasonic beams after hitting the surface. Then further the distance is calculated using the formula

### 5.1.1 COMPONENTS REQUIRED:

- Arduino NANO or Uno (any version)
- HC-SR04 Ultrasonic Sensor
- LM298N Motor Driver Module
- 5V DC Motors
- Battery
- Wheels
- Chassis
- Jumper Wires



## 5.1.2 CIRCUIT DIAGRAM:




## CODE:

micrp | autonomous_bot

```
#include <Servo.h>
Servo myservo;
const byte servostart = 72; //servo motor start point
int distanceleft = 0;
int distanceright = 0;
long t, cm;
void setup() {
  Serial.begin(9600);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(4, INPUT);
  myservo.attach(12);
  start();
  rotete();
}
void loop() {
  analogWrite(3, 50);
  analogWrite(11, 50);
  getdistance();
  int leftdistance = 0;
  int rightdistance = 0;
```

```
if (cm <= 20) {
  robostop();
  Serial.println("robo stop");
  delay(100);
  backward();
  Serial.println("robo backward");
  delay(300);
  robostop();
  Serial.println("robo stop");
  delay(200);
  leftdistance = leftsee();
  Serial.println(leftdistance);
  delay(200);
  rightdistance = rightsee();
  Serial.println(rightdistance);

  if (leftdistance >= rightdistance) {
    turnleft();
    delay(500);
    robostop();
    Serial.println("turnleft");
  } else {
    turnright();
    delay(500);
    robostop();
    Serial.println("turnright");
```

```
    }
  } else {
    forward();
    Serial.println("forward");
  }
}

void start() {
  //myservo.write(servostart);
  delay(3000);
  for (int a = 0; a < 4; a++) {
    myservo.write(servostart);
    delay(50);
    myservo.write(40);
    delay(50);
    myservo.write(90);
    delay(50);
    myservo.write(servostart);
  }
}

void rotete() {
  delay(500);
  digitalWrite(6, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(8, LOW);
```

```
  digitalWrite(9, HIGH);
  delay(2000);
  digitalWrite(6, LOW);
  digitalWrite(5, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
}

void forward() {
  digitalWrite(6, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(8, LOW);
  digitalWrite(9, HIGH);
}
void backward() {
  digitalWrite(6, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(8, HIGH);
  digitalWrite(9, LOW);
}
void robostop() {
  digitalWrite(6, LOW);
  digitalWrite(5, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
}
```

```
void turnright() {
  digitalWrite(6, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(8, HIGH);
  digitalWrite(9, LOW);
}
void turnleft() {
  digitalWrite(6, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, HIGH);
}
int leftsee() {
  myservo.write(servostart);
  delay(1000);
  myservo.write(155);
  delay(1000);
  distanceleft = getdistance();
  //Serial.println(distanceleft);
  myservo.write(servostart);
  return distanceleft;
}
int rightsee() {
  myservo.write(servostart);
  delay(1000);
  myservo.write(5);
```

```
  delay(1000);
  distanceright = getdistance();
  //Serial.println(distanceright);
  myservo.write(servostart);
  return distanceright;
}
int getdistance() {
  digitalWrite(2, LOW);
  delayMicroseconds(4);
  digitalWrite(2, HIGH);
  delayMicroseconds(10);
  digitalWrite(2, LOW);
  t = pulseIn(4, HIGH);
  cm = t / 29 / 2;
  //Serial.println(cm);
  return cm;
}
```

## 5.2 FACIAL RECOGNITION
**(Implemented by Varshini Ganti and Nethra Sai M)**

The facial recognition part in the system is used to detect students' face and mark their attendance. Names of students whose faces were recognized by the facial recognition will be stored in a set to avoid duplicate entries and this set will be written into a text file which will be sent to the user through mail.

### 5.2.1 MACHINE LEARNING AND DEEP LEARNING ALGORITHMS USED:

Histogram of Oriented Gradients(HOG) and CNN:

Feature descriptors simply mean the representation of an image that simply extracts the useful information and disregards the unnecessary information from the image.
In the HOG feature descriptor, the distribution ( histograms ) of guidelines of gradients ( oriented gradients ) are used as features. Gradients ( x and y derivatives ) of a photo are beneficial due to the fact the importance of gradients is massive round edges and corners ( areas of abrupt depth changes ) and we recognize that edges and corners percent in lots of extra data approximately item form than flat areas.

To address the problem of human face popularity on small unique dataset, a brand new method combining convolutional neural network (CNN) with augmented dataset was used for our system. The unique small dataset is augmented to be a massive dataset through numerous modifications of the face images. Based on the augmented face photo dataset, the characteristics of the faces may be correctly extracted and better face popularity accuracy may be accomplished with the aid of the usage of the imaginative CNN. The effectiveness and superiority of the

proposed method may be tested with the aid of using numerous experiments and comparisons with a few often used face popularity methods.

### 5.2.2 LIBRARIES USED:

**Tensorflow:**

TensorFlow is the most used Deep Learning framework and it has pre-trained models that easily help with image classification.
The images are classified using CNN. In most cases, to generate a model means the classification of the images only needs to provide a similar image which is the positive image. The image is then trained and retrained through a process known as anchoring or Transfer Learning.

**OpenCV:**

OpenCV is the massive open-source library for computer vision, machine learning, and image processing and now it performs a chief function in real-time operation which may be very essential in today's systems. By the usage of it, one could process images and movies to become aware of objects, faces, or maybe handwriting of a human. When included with diverse libraries, along with NumPy, python is able to process the OpenCV array shape for analysis. To Identify photo samples and its diverse capabilities we use vector areas and carry out mathematical operations on those capabilities.

**Dlib:**

Dlib is a landmark's facial detector with pre-skilled models, the dlib is used to estimate the area of sixty eight coordinates (x, y) that map the facial factors on a person's face.

**Facial recognition:**

Facial Recognition library is used to map an individual's facial features mathematically and stores the data as a faceprint. The library uses Deep Learning algorithms to compare a live capture or digital image to the stored faceprint in order to verify an individual's identity.

**Imutils:**

A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and Python

### 5.2.3 STEPS FOR IMPLEMENTATION:

**1. Creating the dataset:**

We have 3 python files which help in implementing the facial recognition module of our project. In the headshots python file we will activate the usb camera using open cv and will capture the images of the students to use as a dataset for face detection.

We will first create a folder in the path set for the virtual environment and will name it with the student name. This folder name will be entered in the headshots python file which is used to store the captured images of the student in the right folder with the same name.

2. **Training the model with created dataset**

After generating the dataset with students' images we need to train the model with the generated dataset.
This is done with train_model.py python file
It will resize the captured images which helps in detecting the image in all angles
This python file uses the images saved in the folder and will be trained with folder's name as variable

3. **Facial detection and saving the students names into a text file**

This is the main file of the project and is done with the facereg_mail.py python file. This file when run will start the camera and detect the faces of students. The names of the students will be stored in the set attendees. We have used set instead of list to avoid duplicate entries.
The set attendees is written into a text file iotstudents.txt which is sent as an attachment through mail to the user.
The text file consists of the date and time to make it more accurate.
To send the mail we have used Mailgun software which generates an api key for registered mail with which we can send mail.

**5.2.4 COMPONENTS REQUIRED:**

1. Raspberry pi 4
2. USB camera
3. Pipe (to hold the camera)
4. Power bank for power supply

**CODE:**

To implement facial recognition we created a virtual environment in Raspberry Pi4. To direct to the path in which we have saved the files we have to follow the below - given commands:
-cd Project
-source env/bin/activate
-cd facial_recognition
-ls

The last command will give us the list of folders and files saved in that particular directory

**Headshots.py:**

In the facial_recognition folder we have one more folder called dataset to store all the student images.
So when we want to create a dataset for any student we have to create another folder in the dataset folder with the student name.

**Code:**

```
import cv2

name = 'nethra'
cam = cv2.VideoCapture(0)

cv2.namedWindow("press space to take a photo", cv2.WINDOW_NORMAL)
cv2.resizeWindow("press space to take a photo", 500, 300)

img_counter = 0

while True:
    ret, frame = cam.read()
    if not ret:
        print("failed to grab frame")
        break
    cv2.imshow("press space to take a photo", frame)

    k = cv2.waitKey(1)
    if k%256 == 27:
        # ESC pressed
        print("Escape hit, closing...")
        break
    elif k%256 == 32:
        # SPACE pressed
        img_name = "dataset/"+ name +"/image_{}.jpg".format(img_counter)
        cv2.imwrite(img_name, frame)
        print("{} written!".format(img_name))
        img_counter += 1

cam.release()

cv2.destroyAllWindows()
```

We can capture the images using space button and the captured images will be saved in the corresponding folder and will be labeled from 0 and so on

**train_model.py:**

After generating the dataset we will train the images with the variable name provided in the headshots file using this train_model.py file. In this we will implement HOG (Histogram of Oriented Gradients algorithm)

**Code:**

```
from imutils import paths
import face_recognition
import pickle
import cv2
import os


print("[INFO] start processing faces...")
imagePaths = list(paths.list_images("dataset"))

knownEncodings = []
knownNames = []

for (i, imagePath) in enumerate(imagePaths):

        print("[INFO] processing image {}/{}".format(i + 1,
                len(imagePaths)))
        name = imagePath.split(os.path.sep)[-2]
        image = cv2.imread(imagePath)
        rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

        boxes = face_recognition.face_locations(rgb,model="hog")

        encodings = face_recognition.face_encodings(rgb, boxes)
        for encoding in encodings:

                knownEncodings.append(encoding)
                knownNames.append(name)

print("[INFO] serializing encodings...")
data = {"encodings": knownEncodings, "names": knownNames}
f = open("encodings.pickle", "wb")
f.write(pickle.dumps(data))
f.close()
```

**facereg_mail.py:**

This is the file which implements the main part of the project. This file detects the face and then stores the names of students whose faces were detected in a set and writes into a text file. We are using set to avoid duplicate entries.
This text file will be sent as an attachment to the faculty through mail.
To send the mail we are using Mailgun software.
Mailgun software:
Mailgun is an email automation software company. The platform's ease of use, world-class support, and powerful APIs empower smart development teams to reach real customers at scale with a data-driven approach so their organization can grow faster.

## Code:

```
from imutils.video import VideoStream
from imutils.video import FPS
import face_recognition
import imutils
import pickle
import time
import cv2
import requests
from datetime import datetime
now = datetime.now()
timeout = 300   # [seconds]
timeout_start = time.time()
currentname = "unknown"
encodingsP = "encodings.pickle"
cascade = "haarcascade_frontalface_default.xml"

Attendees = set()
def send_message(name):
    return requests.post(
        "https://api.mailgun.net/v3/sandboxc0755b6249704d0b8847720c8f4df716.mailgun.org",
        auth=("api", "ea2bf452bf9eacc418f5447a2acbcc85-31eedc68-2bbd8bc0"),
        files = [("attachment", ("iotstudents.txt", open("iotstudents.txt", "rb").read()))],
        data={"from": 'hello@example.com',
            "to": ["vignesh.surendran2020@vitstudent.ac.in"],
            "subject": "Attendance for IOT",
            "html": "<html>"+now.strftime("%d/%m/%Y %H:%M:%S")+"</html>"})

print("[INFO] loading encodings + face detector...")
data = pickle.loads(open(encodingsP, "rb").read())
detector = cv2.CascadeClassifier(cascade)

print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
time.sleep(2.0)
fps = FPS().start()

while True:
        frame = vs.read()
        frame = imutils.resize(frame, width=500)
```

```
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        rects = detector.detectMultiScale(gray, scaleFactor=1.1,
                minNeighbors=5, minSize=(30, 30),
                flags=cv2.CASCADE_SCALE_IMAGE)

        boxes = [(y, x + w, y + h, x) for (x, y, w, h) in rects]

        encodings = face_recognition.face_encodings(rgb, boxes)
        names = []

    for encoding in encodings:

                matches = face_recognition.compare_faces(data["encodings"],
                        encoding)
                name = "Unknown"

                if True in matches:

                        matchedIdxs = [i for (i, b) in enumerate(matches) if b]
                        counts = {}

                        for i in matchedIdxs:
                                name = data["names"][i]
                                counts[name] = counts.get(name, 0) + 1

                        name = max(counts, key=counts.get)

                        if currentname != name:
                                currentname = name
                                print(currentname)
                if(currentname not in Attendees and currentname!="Unknown")
                    Attendees.add(currentname)
                names.append(name)
```

```
        for ((top, right, bottom, left), name) in zip(boxes, names):
                cv2.rectangle(frame, (left, top), (right, bottom),
                        (0, 255, 225), 2)
                y = top - 15 if top - 15 > 15 else top + 15
                cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
                        .8, (0, 255, 255), 2)

        cv2.imshow("Facial Recognition is Running", frame)
        if time.time() < timeout_start + timeout:
            break

        fps.update()

f = open("iotstudents.txt",a)
f.write("\nAttendance for Iot\n")
f.write("now.strftime("%d/%m/%Y %H:%M:%S")")
f.write(Attendees)
f.close()

request = send_message(iotstudent.txt)
print ('Status Code: '+format(request.status_code))
fps.stop()
print("[INFO] elasped time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))
cv2.destroyAllWindows()
vs.stop()
```

## 6. RESULT

In this project we have implemented both obstacle avoidance and facial recognition system to build an automated attendance marker which can go in a classroom avoiding obstacles at the same time mark attendance of students using facial recognition.

The final output is a mail with a text file attachment of names of students who were present for the class along with the date and time of the class to make it easy for the faculty to understand class timing and to which class this attendance sheet belong to.

## OUTPUT:

varshini
ishaan
nethra
vignesh
Attendance for ECE3501
11/19/2022, 11:55:57
ishaan
varshini
vignesh
nethra
Attendance for ECE3501
11/19/2022, 12:03:06
vignesh
ishaan
Attendance for ECE3501
11/19/2022, 12:27:11
varshini
vignesh
ishaan
Attendance for ECE3501
11/19/2022, 12:44:31
ishaan
vignesh
Attendance for ECE3501
11/19/2022, 12:46:10
vignesh
nethra
varshini
ishaan
Attendance for ECE3501
11/19/2022, 12:51:49
varshini
ishaan
Attendance for ECE3501
11/19/2022, 12:58:50
vignesh
ishaan

## 7. CONCLUSION

In this study, we develop an automatic attendance system for a classroom that uses the CNN algorithm to recognise students' faces in the classroom, section, or laboratory and automatically record their attendance. It is more efficient in terms of time and effort, particularly when there are many pupils in the class. The suggested system makes use of a camera and face recognition to track a student's attendance. The faces are entered in a database once they have been matched to confirm their presence or absence. After comparing the stored photos in the database, it automatically stamps the student's attendance in the classroom.
The faculty can utilize the date to keep track of a student's attendance status and get email notifications. Send a text message to their phone number if they aren't available. This method of automatically and covertly monitoring students during lectures and classes is a better approach to keep track of attendance.

## 8. REFERENCES


[1] Attendance Management System using Face-Recognition - Mrunal Aware, Prasad Labade, Manish Tambe, Aniket Jagtap, Chinmay Beldar
[2]https://www.ijraset.com/research-paper/automated-attendance-system-using-face-recognition
[3]https://pyimagesearch.com/2015/02/02/just-open-sourced-personal-imutils-package-series-opencv-convenience-functions/
[4]https://www.tfcertification.com/blog/building-face-recognition-model-tensorflow#:~:text=Using%20Tensorflow%20For%20Face%20Recognition&text=The%20images%20are%20classified%20using,as%20anchoring%20or%20Transfer%20Learning.
[5]https://towardsdatascience.com/cnn-based-face-detector-from-dlib-c3696195e0c
[6]https://techvidvan.com/tutorials/face-recognition-project-python-opencv/
[7] https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/