

Design of face detection and recognition using

Machine learning

BY

VIGNESHWARAN . S (Reg. No. 39290125)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

MARCH 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VIGNESHWARAN . S (Reg. No. 39290125)** who carried out the project entitled **“Design of face detection and recognition system to monitor students during online examinations using Machine learning algorithms”**.

Project Supervisor

DR MAGESHWARI,

Head of the Department

Dr. L. LAKSHMANAN

DECLARATION

VIGNESHWARAN . S (Reg. No. 39290125) hereby declare that the Project Report entitled **“Design of face detection and recognition system to monitor students during online examinations using Machine learning algorithms”** done by me under the guidance of **DR MAGESHWARI**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

Alternative Title

Mechanizing Student's Attendance During Online Examination via Face Recognition

AIM

- To ease the process of attendance during online classes and to prevent malpractice during online assessments.

ABSTRACT

- Today's pandemic situation has changed the way of education to students. Education system had completely undertaken by online platforms. There are many platforms that provide services to schools, colleges and other educational intuitions. In addition to the online teaching, examination also has gone online. In this online education system, it is important to monitor the students presence i.e., attendance of the students, which plays vital role in making system. Even though educational institutions have adapted to online mode which is done through online portal, it is very challenging for staffs in taking attendance for a students. With the integrated webcam in online portals we can monitor the activities of students and malpractices done by them. But the attendance of the students is quite difficult and it is done manually by staffs by just seeing who are all in the streaming video. This can be bridled by face detection and recognition techniques using KNN algorithm. With this integration we can extract a facial feature vector which is otherwise known as embeddings and train it using Python Face Recognition Library, then identify the faces of the students before entering into examination dashboard. If it is matched the attendance system auto updates the presence of the particular student and we can also find the attendance percentage of the students, which is useful in analyzing the willing percentage of the students to write the exam online.

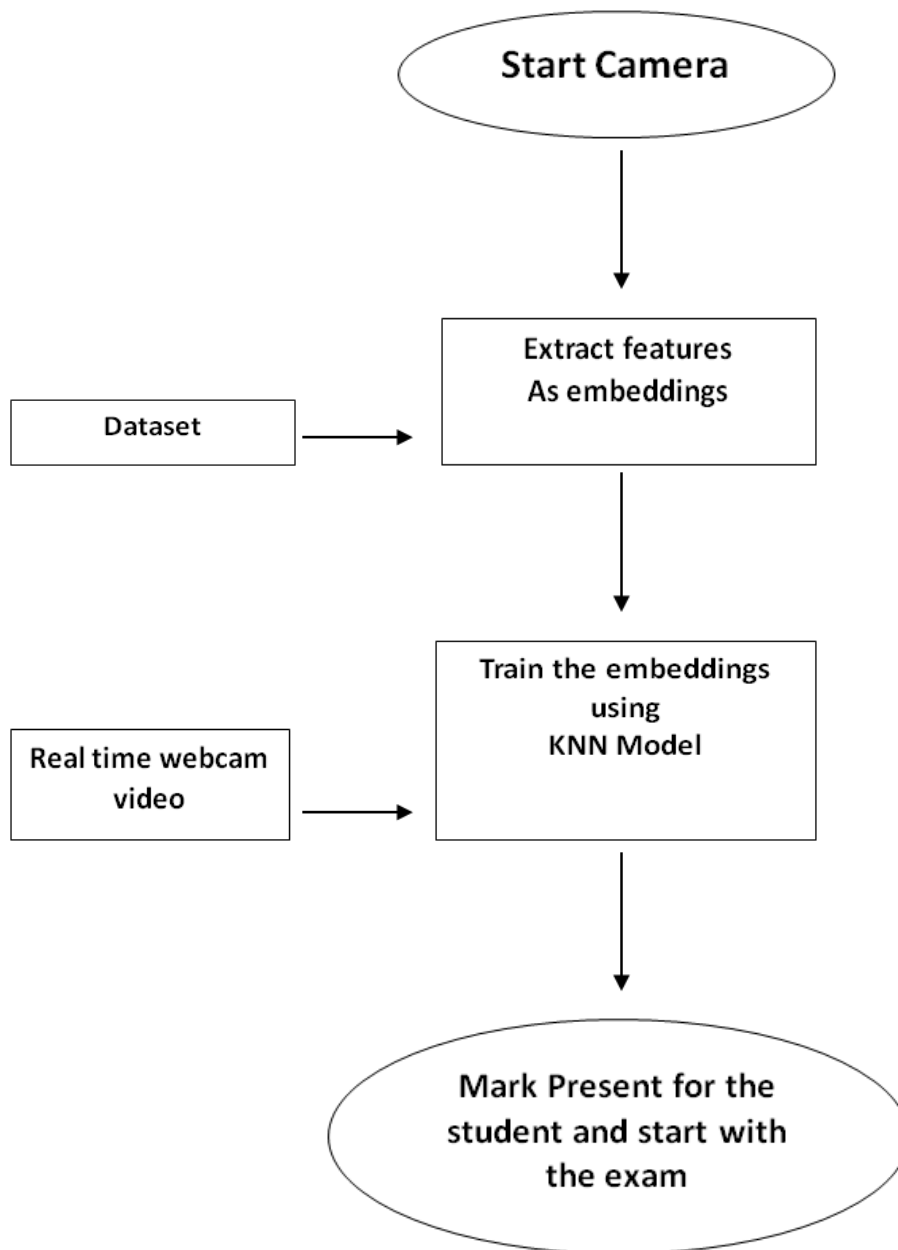
Existing system

- In our education system, exam plays vital role in accessing and evaluating the student's knowledge.
- These exams have several strict set of rules to be followed by the students in order to write their exams and these rules ensure that the student write their exams in proper way without malpracticing.
- Now as it's a pandemic period this system is completely made online and the monitoring is done via webcam during the classes and exams.
- The educational institutions conduct exams for large number of students and it is difficult to monitor each one manually while some students take this as an advantage and started to impersonate someone like themselves during exams and scores better than the one who are true.
- To prevent this we are going to integrate a face recognition technique and machine learning algorithm in proposed system.

Proposed system

- In our prototype we implement Python Face Recognition Module to train the Face Images.
- We use KNN algorithm to analyze the nodes in face image then marks the patterns in various images which is taken in different angles.
- These images get trained as models in python server. We develop a web application as our ground work to mark student's attendance during Online Exams.
- We develop with AJAX Api calls java-script functions to get our response and request more responsive.
- The status of the application and all details of student will be stored and retrieved from MYsql Database Server which is maintained periodically.
- We implement JDBC connection in java Servlet to access our database. All the requests are sent to the Backend Business Logics which is written in Java Servlets using J2EE technology.

SYSTEM ARCHITECTURE



Modules

- Admin Sign in
- Student Sign in
- Attendance Update
- Exam Attendance Audit

Admin Sign-In

- Admin or a Teacher (as Admin) can only sign in to the admin page, where they can add a new student to their university with student's basic details and their recent photo which is sent to the Python Coded Machine Learning Algorithm for future face recognition in attendance system.
- Also admin can put on college events, seminars etc. The new students added are annexed to the DB maintained by the administrator.

Student Sign-In

- Students who are all added on by the admin can only sign in via Student Sign in page.
- The students have to use their Student-Id and Password which is provided by admin while registration, to sign in their online class portal.
- In this online portal they can avail online class as well as online exam. The students need to plump for what they need.

Attendance Update

- After students picked up their choice, they are directed to the face verification page where their faces get captured by the live camera and compared with the trained KNN Set and their attendance is casted up.
- After successful verification the students are allowed to enter their class or exam as per their choice.
- If the student's pick is online exam, they have to complete the test within the timer runs out. After their allocated session is done, once again they are diverted to the face verification page to add a final touch to end up the attendance process.

Exam Attendance Audit

- After the exam gets wind up, their marks are updated and with the attendance report an analysis of student's willingness to attend exam online will be noted down and compared.

Algorithm Used

- Face Recognition
- KNN Algorithm

Software Requirements:

- Windows 7(1no)
- JDK 1.8
- My SQL 8.0
- Tomcat 8.5

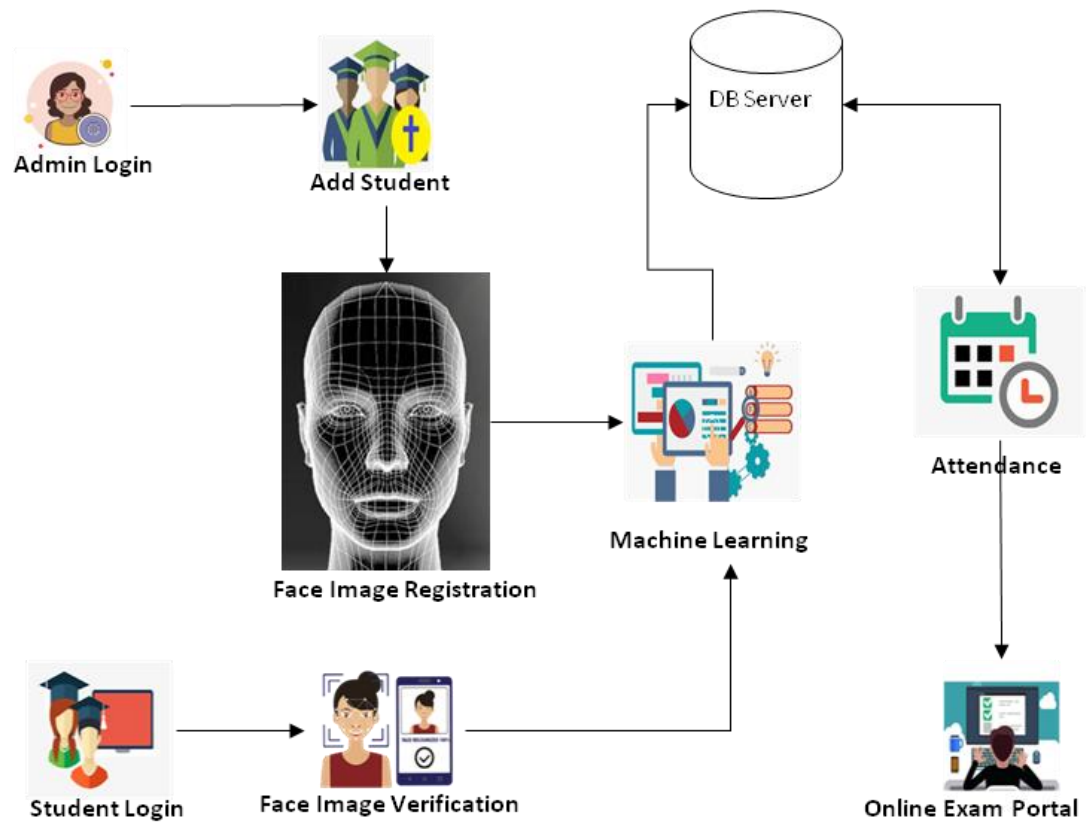
Hardware Requirements

- Hard Disk : 250GB and Above
- RAM : 4GB and Above
- Processor : i3 and Above

Technologies Used

- Java Servlet, Python
- HTML,CSS,JS

Architecture Diagram



DATABASE DETAILS

edumaster Attendance

Database Name : edumaster

Server version 8.0.26

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0
*/;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

Table structure for table `attendance`

```
CREATE TABLE `attendance` (
  `Name` varchar(255) DEFAULT NULL,
  `InTime` varchar(255) DEFAULT NULL,
  `OutTime` varchar(255) DEFAULT NULL,
  `Date` varchar(255) DEFAULT NULL,
  `Attendance` varchar(255) DEFAULT NULL,
  `StudentId` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

Dumping data for table `attendance`

```

LOCK TABLES `attendance` WRITE;
/*!40000 ALTER TABLE `attendance` DISABLE KEYS */;
INSERT INTO `attendance` VALUES ('Karthick','11:02','11:10','2022-02-09','Present','EM01'),('Karthick','6:00 pm','6:01 pm','2022-2-18','Present','EM01'),('Daniel','6:29 pm','6:29 pm','2022-2-18','Present','EM06');
/*!40000 ALTER TABLE `attendance` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```

Table structure for table `student`

```

CREATE TABLE `student` (
  `Fname` varchar(255) DEFAULT NULL,
  `Lname` varchar(255) DEFAULT NULL,
  `Phone` bigint DEFAULT NULL,
  `Email` varchar(255) DEFAULT NULL,
  `Password` varchar(255) DEFAULT NULL,
  `City` varchar(255) DEFAULT NULL,
  `Country` varchar(255) DEFAULT NULL,
  `File` varchar(255) DEFAULT NULL,
  `StudentId` varchar(255) DEFAULT NULL
)

```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

Dumping data for table `student`

```
INSERT INTO `student` VALUES
('Karthick','R',9876543210,'karthi@gmail.com','karti123','chennai','India',
['object
File'],'EM01'),('Princy','J',9870654313,'princy@gmail.com','princy123','
chennai','India',['object
File'],'EM02'),('Sangeetha','Jai',9876512304,'sangeetha@gmail.com','san
gee123','chennai','India',['object
File'],'EM03'),('Kavitha','Guna',9876512340,'kavi@gmail.com','kavi123',
'chennai','India',['object
File'],'EM04'),('Kayal','Vivek',89045671234,'kayal@gmail.com','kayal12
3','madurai','India',['object
File'],'EM05'),('Daniel','Maran',9876012354,'daniel@gmail.com','dani12
3','Tirunelveli','India',['object
File'],'EM06'),('Priyanka','S',8905674321,'priyanka@gmail.com','priya1
23','Chennai','India',['object
File'],'EM07'),('Sakthi','Eswar',8907651234,'sakthi@gmail.com','sakthi1
23','Chennai','India',['object
File'],'EM08'),('Shankar','Manikandan',8907654312,'shankar@gmail.co
m','shankar123','Trichy','India',['object
File'],'EM11'),('Kathir','Ganesh',8907651234,'kathir@gmail.com','kathir
123','Trichy','India',['object
File'],'EM10'),('Yalni','Mohan',8904561237,'yalni@gmail.com','','Chenna
i','I','undefined',''),('Yalni','Mohan',8904561237,'yalni@gmail.com','yal1
23','Chennai','India',['object File'],'EM12');
/*!40000 ALTER TABLE `student` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

Table structure for table `test`

```
CREATE TABLE `test` (  
  `Name` varchar(255) DEFAULT NULL,  
  `ExamName` varchar(255) DEFAULT NULL,  
  `Mark` varchar(255) DEFAULT NULL,  
  `InTime` varchar(255) DEFAULT NULL,  
  `OutTime` varchar(255) DEFAULT NULL,  
  `Date` varchar(255) DEFAULT NULL,  
  `Attendance` varchar(255) DEFAULT NULL,  
  `StudentId` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;  
/*!40101 SET character_set_client = @saved_cs_client */;
```

Dumping data for table `test`

```
INSERT INTO `test` VALUES ('Karthick','English','100','6:02 pm','6:03  
pm','2022-2-  
18','Present','EM01'),('Daniel','OOPS','80','06:10','06:30','2022-02-  
18','Absent','EM06'),('Princy','JAVA','60','07:10','07:30','2022-02-  
18','Present','EM02'),('Daniel','English','100','6:30 pm','6:31 pm','2022-  
2-18','Present','EM06'),('Daniel','English','100','6:30 pm','6:31  
pm','2022-2-18','Absent','EM06');
```

SOURCE CODE

Face Recognition KNN

```
import os
import json
from math import sqrt
from sklearn import neighbors
from os import listdir
from os.path import isdir, join, isfile, splitext
import pickle
from PIL import Image, ImageFont, ImageDraw, ImageEnhance
import face_recognition
from face_recognition import face_locations
from face_recognition.cli import image_files_in_folder
#This is not working for me
#from face_recognition.face_recognition_cli import
image_files_in_folder          #comment this line if you
got error
from flask import Flask, jsonify, request,
redirect,render_template
from flask_cors import CORS
from os import path

ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}
REGISTER_PATH="./Register/"
TEMP_PATH="./Temp/"
if not os.path.exists(REGISTER_PATH):
    os.makedirs(REGISTER_PATH )

app = Flask(__name__)
CORS(app)
def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1].lower() in
ALLOWED_EXTENSIONS

@app.route('/register', methods=['GET', 'POST'])
def upload_register():
    # Check if a valid image file was uploaded
    if request.method == 'POST':
        if 'file' not in request.files:
            return json.dumps({"status": "Error", "msg": "Image
cannot be empty"})
        name = request.form.get('name')
```

```

email = request.form.get('email')

if(name == ''):
    return json.dumps({"status": "Error", "msg": "Name
cannot be empty"})

    file = request.files['file']
    # src =
path.realpath(REGISTER_PATH+"/"+email+"/"+name+"/"+file.filename
)

    print(name)
    print(file)

    if file.filename == '':
        return json.dumps({"status": "Error", "msg": "Image
cannot be empty"})

    if file and allowed_file(file.filename):
        if os.path.exists(REGISTER_PATH+email):

#file.save(REGISTER_PATH+email+"/"+file.filename)
        foo = Image.open(file)
        #foo = foo.rotate(90, PIL.Image.NEAREST, expand
= 1)

foo.save(REGISTER_PATH+email+"/"+file.filename.strip(),optimize=
True,quality=85)

os.rename(REGISTER_PATH+email+"/"+file.filename,REGISTER_PATH+em
ail+"/"+name+".jpg")

        #return json.dumps({"status": "Error", "msg":
"User Already Registered"})
        else:
            os.makedirs(REGISTER_PATH+email)
            foo = Image.open(file)

#file.save(REGISTER_PATH+email+"/"+file.filename)
        #foo = foo.rotate(90, PIL.Image.NEAREST, expand
= 1)

```

```
foo.save(REGISTER_PATH+email+"/"+file.filename.strip(),optimize=
True,quality=85)
```

```
os.rename(REGISTER_PATH+email+"/"+file.filename,REGISTER_PATH+em
ail+"/"+name+".jpg")
```

```
        print(file)
        return json.dumps(train(REGISTER_PATH))
    else:
        return json.dumps({"status": "Error", "msg": "Image
Format not supported <png,jpg,jpeg> "})
```

```
    # If no valid image file was uploaded, show the file upload
form:
```

```
    return json.dumps ({"status":"Error","msg":"GET Not Allowed
"})
```

```
@app.route('/', methods=['GET', 'POST'])
```

```
def upload_image():
```

```
    # Check if a valid image file was uploaded
```

```
    if request.method == 'POST':
```

```
        if 'file' not in request.files:
            return redirect(request.url)
```

```
        file = request.files['file']
```

```
        if file.filename == '':
```

```
            return redirect(request.url)
```

```
        foo = Image.open(file)
```

```
        if file and allowed_file(file.filename):
```

```
            # The image file seems valid! Detect faces and
```

```
return the result.
```

```
        print(file)
```

```
        #foo = foo.rotate(90, PIL.Image.NEAREST, expand = 1)
```

```
foo.save(TEMP_PATH+"/"+file.filename.strip(),optimize=True,quali
ty=85)
```

```
        return json.dumps(predict(file))
```

```
    # If no valid image file was uploaded, show the file upload
form:
```

```
    return render_template('index.html')
```

```

def predict(X_img_path, knn_clf = None, model_save_path
="train", DIST_THRESH = .5):

    # if not isfile(X_img_path) or splitext(X_img_path)[1][1:]
not in ALLOWED_EXTENSIONS:
    #         raise Exception("invalid image path:
{}".format(X_img_path))
    if knn_clf is None:
        with open(model_save_path, 'rb') as f:
            knn_clf = pickle.load(f)

    # X_img = face_recognition.load_image_file(X_img_path)
    #X_img = face_recognition.load_image_file(X_img_path)
    X_img =
face_recognition.load_image_file(TEMP_PATH+X_img_path.filename)
    X_faces_loc = face_locations(X_img)
    if len(X_faces_loc) == 0:
        return []

    faces_encodings = face_recognition.face_encodings(X_img,
known_face_locations=X_faces_loc)

    closest_distances = knn_clf.kneighbors(faces_encodings,
n_neighbors=1)

    is_recognized = [closest_distances[0][i][0] <= DIST_THRESH
for i in range(len(X_faces_loc))]

    return [{"result":pred,"status":str(rec)} if rec else
{"result":"N/A","status":str(rec)} for pred, loc, rec in
zip(knn_clf.predict(faces_encodings), X_faces_loc,
is_recognized)]

def train(train_dir, model_save_path="train", n_neighbors=None,
knn_algo='ball_tree', verbose=False):
    X = []
    y = []
    for class_dir in listdir(train_dir):
        if not isdir(join(train_dir, class_dir)):
            continue
        for img_path in image_files_in_folder(join(train_dir,
class_dir)):
            image = face_recognition.load_image_file(img_path)
            faces_bboxes = face_locations(image)
            if len(faces_bboxes) != 1:
                if verbose:

```

```

        print("image {} not fit for training:
{}".format(img_path, "didn't find a face" if len(
        faces_bboxes) < 1 else "found more than
one face"))

        continue
        X.append(face_recognition.face_encodings(image,
known_face_locations=faces_bboxes)[0])
        y.append(class_dir)

    if n_neighbors is None:
        n_neighbors = int(round(sqrt(len(X))))
        if verbose:
            print("Chose n_neighbors automatically as:",
n_neighbors)

    knn_clf =
neighbors.KNeighborsClassifier(n_neighbors=n_neighbors,
algorithm=knn_algo, weights='distance')
    knn_clf.fit(X, y)

    if model_save_path != "":
        with open(model_save_path, 'wb') as f:
            pickle.dump(knn_clf, f)
    return {"status": "Success", "msg": "Trained successfully"}

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5001, debug=True)

'''
http://localhost:5001/          GET          web interface
http://localhost:5001/          POST          for face search variable
name is search return JSON
http://localhost:5001/register POST          for Registering face.
variable names photo= file name = name
can be tested with postman

sample html will load
'''

```

DATABASE CONNECTION CODE

```
package com.db;

import java.sql.Connection;
import java.sql.DriverManager;

import javax.servlet.ServletContext;
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;

public class DbConnection implements ServletContextListener{

    public static Connection con;

    ServletContext servletcontext;

    @Override
    public void contextDestroyed(ServletContextEvent arg0) {
        // TODO Auto-generated method stub
    }

    @Override
    public void contextInitialized(ServletContextEvent arg0) {
        // TODO Auto-generated method stub
        servletcontext=arg0.getServletContext();
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");

            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/edumaster","root","root");

            servletcontext.setAttribute("connection",con);
            System.out.println("---Education Master db Connection created ---!!!---");

        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {

    }
}
```



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119
www.sathyabama.ac.in



Conclusion

- Thus our system is proposed to enhance the attendance using automatic face recognition.

Thank You