IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

REAL-TIME SMART ATTENDANCE SYSTEM USING FACE RECOGNITION TECHNIQUE

[1]Kaushik Goswami, [2]Priyanshu Acharya, [3]Rupsa Paul, [4]Sulagna Chatterjee

[1] Faculty, [2,3,4] Student

[1,2,3,4] Postgraduate and Research Department of Computer Science

[1,2,3,4] St. Xavier's College (Autonomous), Kolkata, India

Abstract: One's face is an essential physical feature that gives everyone their own identity. The face recognition system can be used by using the features of the face as a biometric. Any organization's most difficult responsibility is recording attendance. Huge amount of time is often wasted to record the attendance.

Under the conventional attendance system, teachers call on the students, and their attendance is recorded, accordingly. These conventional methods are laborious and time-consuming. In this project, we suggest Facial Recognition Method through which the photo of the entire class can be taken and then the attendance can be marked. This model combines an input image-capturing camera, an algorithm for facial recognition from input images, facial encoding, and identification, and finally attendance tracking in a CSV file for the required office. The database is created by training the system with the faces of the authorized students of every class and then their facial encodings are taken and then the model is trained. The library of Python is trained by HOG algorithm and SVM Classifier for identifying faces.

Keywords: Face Recognition, HOG, SVM, Euclidean Distance, Python, MySQL.

I. INTRODUCTION

Schools, colleges, and universities over the years keep record of student's presence in the class as studies have shown a connection between student's educational achievements and their attendance rates. So, teachers and professors keep the record manually which is not efficient and, on an average, if it takes around 7 minutes to take the attendance and considering the guidelines of 52 periods for a paper, it is a wastage of around 364 minutes or 6 hours per paper.

So, nowadays biometric methods for attendance are used, such as fingerprint scanning or face recognition. However, the use of fingerprint scanning is impractical in classes with high student populations as it requires a one-to-one interaction with the machine and the user. Face recognition is another way of checking whether a student is present or not. But here also, the available methods mostly require a one-to-one interaction with the camera and the student. Considering this, the idea of marking the attendance of the entire class through one image from a camera came, which would essentially save a lot of time and make the process of taking the attendance much faster and efficient with higher accuracy.

HOG or Histogram of Oriented Gradients is a feature descriptor which is used in computer vision and image processing tasks involving the detection of objects. It is mostly used to detect pedestrians. HOG relies on the property of objects in an image to generate the gradients and directions. Gradients are calculated for each pixel and is constituted for a pixel grid where the magnitude is constituted and the direction of change is

accounted together for the computation. The gradient for each pixel constituted by the magnitude and direction is generated by the following formula,

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\Theta = \arctan(\frac{g_y}{g_x})$$

where, g_x and g_y are the horizontal and vertical components of change in pixel intensity respectively.

SVM or Support Vector Machine is a supervised machine learning technique mostly used for classification problems. It is sometimes also used for regression problems. Searching for an ideal hyperplane to divide data points of various classes in a high-dimensional space is its main objectives. Since, the project has more characteristics than samples, it is a better choice. SVM tries to draw the hyperplane by maximizing the margin between the different classes. The margin is the distance between the hyperplane and the support vectors, which are the nearest data points from each class.

The dataset had a large number of images whose HOGs were present and were classified as whether it is a face or not. This dataset being a classification problem was best fitted using the SVM and face_recognition library used this method to train the model and the model generated can now predict whether a new HOG is a face or not. As a result, the SVM can now easily find the new faces and locate them in a new picture.

Euclidean Distance is a common metric used in facial recognition algorithms to compare the similarity between two feature vectors. The feature vectors that are generated from the HOG for various faces are then checked for similarity with the already existing ones. Euclidean Distance is a method through which the similarity is determined between the two feature vectors using the formula,

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

where, p and q are the feature vectors of two different face images and n is the number of elements in each vector, which for this model is 128 as a 128-dimensional vector is generated for each face in the Python library used. A smaller distance indicates a greater similarity between the face images.

II. LITERATURE REVIEW

Traditional ways of Fingerprint Authentication would result in a huge cost while applying in each class and waste a huge amount of time while recording. The use of QR codes and RFID also increases the chance of the misuse of the system. Retinal scanning also suffers from similar problems as pointed out in the work of Helmi R. et. al.[1] in 2019. Their method of Eigenface Algorithm not only provides more efficiency but also helps to match faces when there is a change of angles or some features.

In 2019, Xing Y. proposed a method using SL4A and TTS (Text-to-Speech Recognition) where the student's voice is recorded each time when the attendance system calls out their names and the students answer their presence in their mobile phones. In this process, although the teacher presses the attendance button throughout the procedure, it still wastes enough time which could be solved by using Facial Recognition.[2]

Mijić D et. al. [3] proposed the method of using RFID technologies where an RFID card will be issued to every student and when they enter the classroom their attendance is recorded. However, the cost of initial setup is huge and also suffers from the disadvantage of students bringing the tags of their friends to mark their attendance. However, their software structure influenced the database system of this project.

One common problem in facial recognition is that the images might become blurry at the edges while clicking. Timo A. et. al. [4] in their paper discusses the technique of the Local Phase Quantization (LPQ) model for

recognition and show that the LPQ model reaches a higher recognition rate than the widely used local binary pattern.

Single-face detection in an image can be achieved with high accuracy by using a Convolutional Neural Network as mentioned in the paper of F. P. Filippidou et al. [5] However, the paper requires accuracy in multiple-face detection. To improve upon the problem of intensity of light and head pose problem the Viola and Jones Algorithm and Principal Component Analysis are used. And in the detection of faces, the Ada-Boost algorithm is the most effective algorithm for multiple face detection as mentioned by Wagh P. et. al. [6] in their paper.

Face Recognition is one of the best methods for attendance at present leaving out certain problems. This model is going to be an advanced model for what is already shown by P. Visalakshi et. al.[7] in their paper where they are using LBPH organisers along with LBP and paves the way for the multiple face detection we are going to do. Similar model from the papers of Ch. V. Kumar et. al. [8] is upon which the project has been improved. Their work is mainly based upon OpenCV which is a popular computer vision library started by Intel in 1999. The cross-platform library sets its focus on real-time image processing and includes patent-free implementations of the latest computer vision algorithms.[9] This paper uses the Python based face_recognition library for its implementation.[10]

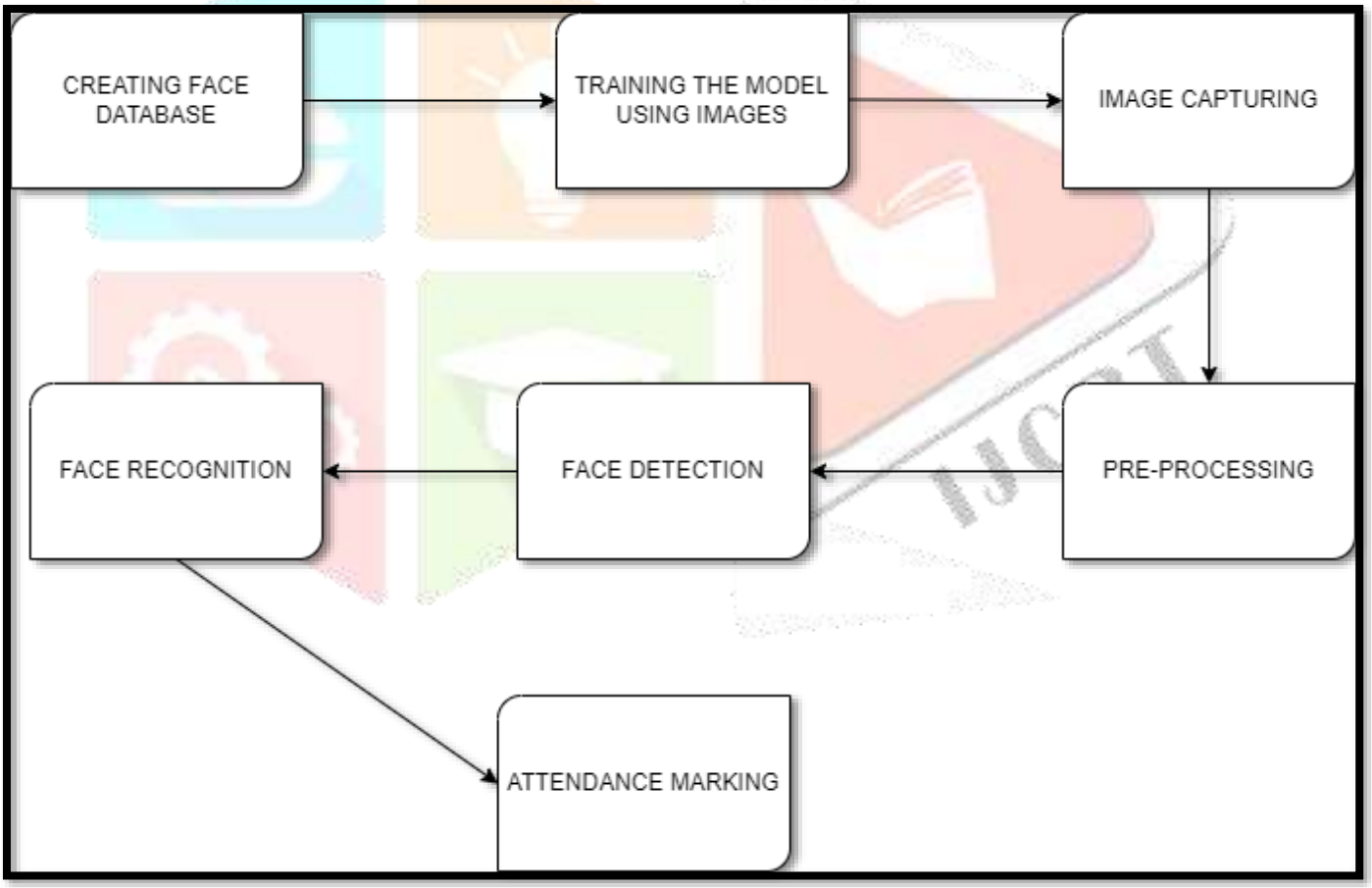

Figure 1: - Methodology

III. METHODOLOGY

The system must follow these steps needed to be processed in following order as shown in Figure 1:

1. Creating Face Database
2. Training the Model using the images
3. Image Capture
4. Pre-processing
5. Face Detection
6. Face Recognition
7. Registering Attendance

1. Creating Face Database: The database, which serves as the system's training set, contains pictures of enrolled students. For this research, the system was trained using five photos per student in the training set, with a total of forty-one students. This means that a total of 205 photos were used to train the system. Here, a GUI was used to input relevant information from each student, which was then fed into the Student Database.

2. Training the Model using the Images: The images are then sorted out according to the year of the students and their face encodings are trained separately in separate files. They are also stored in the server for reducing the time complexity of searches made during the facial recognition.

3. Image Capture: To effectively detect and recognize faces, a high-quality camera is needed. Correct drivers for the camera must be installed, to ensure that the drivers are compatible with Python, and the camera is connected to the server. Once the camera is turned on; it will capture images which will then be processed to identify the faces. The images are then sent to the database from where the next steps follow.

4. Pre-processing: In the initial processing stage, the following steps are performed on the input image. The image is converted from the BGR format to the RGB format as the library supports the use of RGB format images for proper detection.

5. Face Detection: The system looks at the photos once they are received and then sends them for face detection. To detect faces, the system uses the HOG model along with SVM classifier.

6. Face Recognition: The comparison between the test and training photos is carried out in this, the most crucial module of this system. Although there are several algorithms available to carry out the recognition process, Euclidean distance is used to compare the faces. The lesser the distance, more similar the faces.

7. Registering Attendance: The attendance registration module is completed right after the face recognition module. If the identified face is recognized, the attendance will be recorded in the database and then it is taken out into a CSV file.

IV. ALGORITHM

The algorithm to train the database with the student's image is as follows: -

Step 1: - A database is created with 5 tables. One table consists of the details of the students and their images, another consists of the teacher timetable, one consists of a limited details of the students along with a column to mark their attendance, and another allocates the teachers in it for a particular period which updates dynamically after every period to store the new teachers for each class in it after every period. The last table

stores the image from the particular camera with the details about the year and the classroom. They are named as student, teacher, student_attendance, teacher_allocation and camera respectively.

Step 2: - Students enter their details and five images of their faces through a front end. These details are stored in the student and student_attendance table in the database.

Step 3: - The images are retrieved from the student table separately for each year and the facial encodings of each student is taken out, which is a 128-dimensional feature vector which contains various information regarding measurements of various facial features. These are then stored as flat files in the server with the students' roll and year.

Step 4: - With this the training of the model is over and the encodings are stored for use when required.

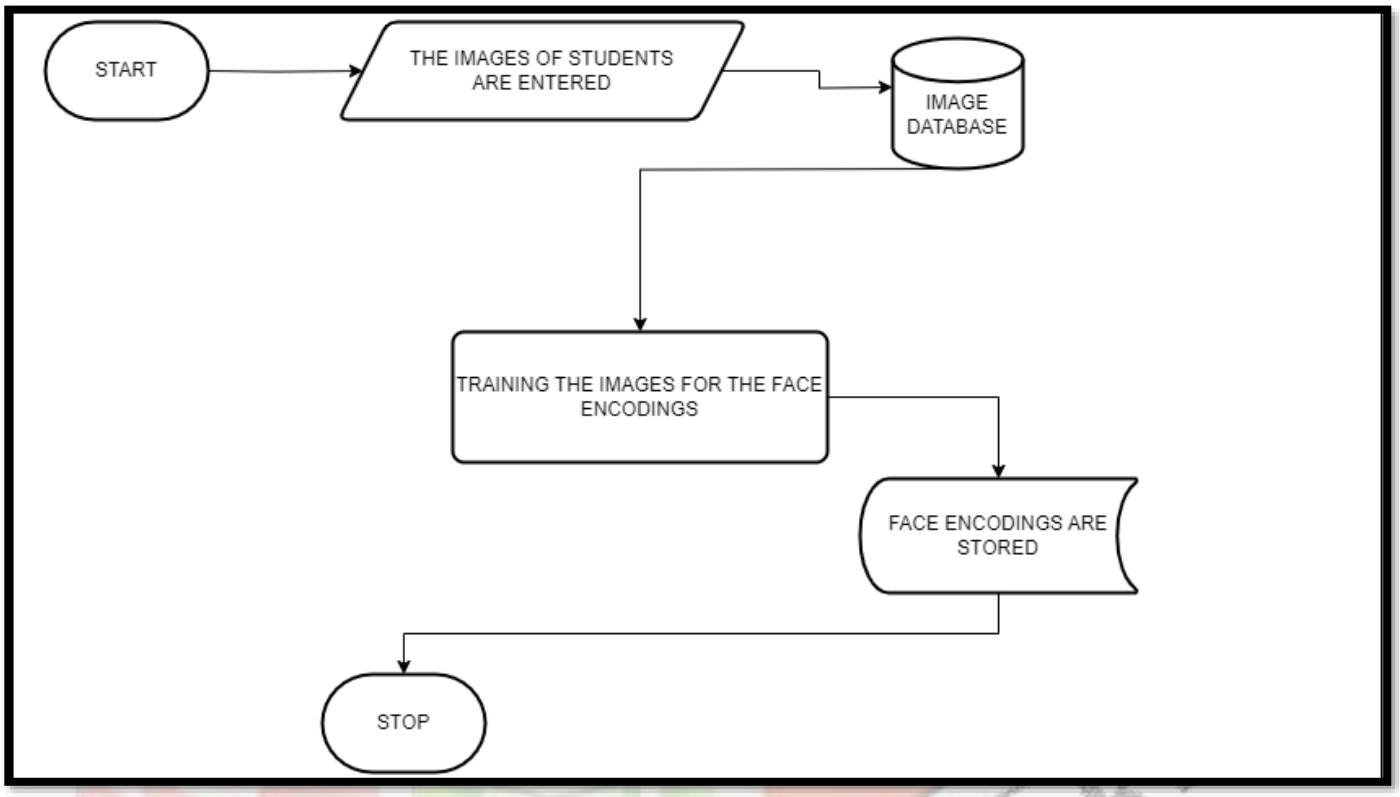

Figure 2: - Flow Chart for Training

The algorithm to click the image and mark the attendance from it is as follows: -

Step 1: - The particular day and period are retrieved from the system date and time and accordingly the teacher_allocation table is filled with the information of a particular period from the teacher table.

Step 2: - Now, the images are captured from the classes using a camera that is already installed at times already specified for each period in the class. These images are sent to the camera table in the database.

Step 3: - The images are retrieved and are preprocessed by changing them from BGR format to RGB format as the library can work only on RGB images.

Step 4: - The model then detects the locations of all the faces in the image using HOG and SVM. These locations are then used to get the facial encodings of each of those faces.

Step 5: - The year of the class is already known from the camera table and now the known face encodings of that particular year is received from the server and their faces are matched with the encodings found from the image captured.

Step 6: - The Euclidean Distance between the face encodings are calculated and only when it is below a certain threshold value, the face is taken for a match else it is rejected. If it matches then the roll is noted.

Step 7: - Since, all the rolls are achieved students are marked in the student_attendance table as "Present", using the roll and year.

Step 8: - Next the teacher_allocation table is joined with the student_attendance table over a particular year, and the name of the teacher along with his/her students' attendance can now be retrieved.

Step 9: - The attendance is marked in a CSV file with the students and the teacher's name for the official record. At last, for the next period the steps are again repeated from 1, and the cycle continues till the last period.

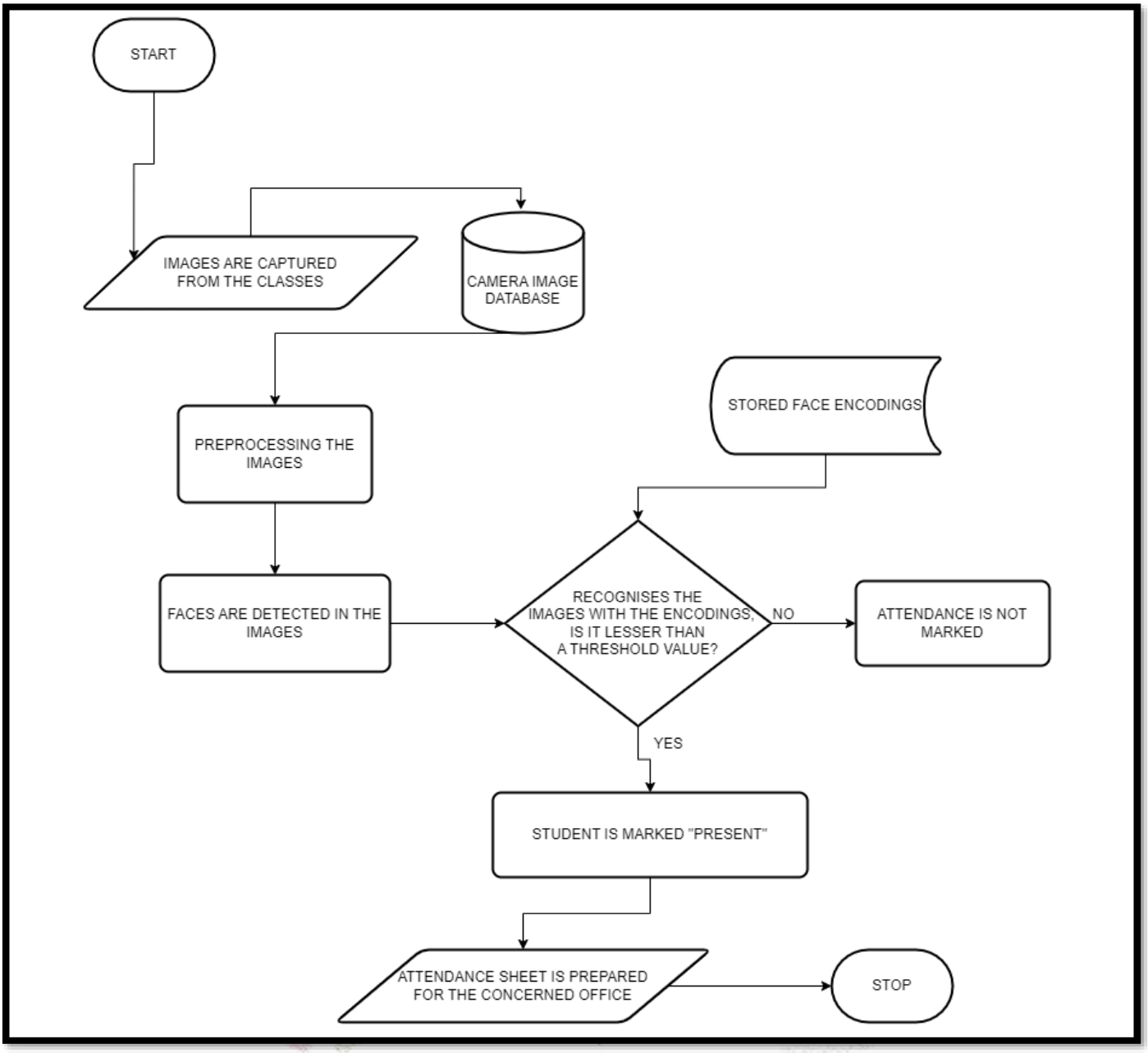

Figure 3: - Flow Chart for Recognition and Attendance

V. RESULTS AND DISCUSSION

The face_recognition library of Python recommends a default tolerance value of 0.6 to be used as the Euclidean Distance for a match between a known and an unknown face. For this model the default tolerance value was changed from 0.6 to 0.48 by Hit and Trial Method. The tolerance level was playing a great role in determining the faces correctly.

Figure 4: - Testing using tolerance value 0.6

The value of 0.6 was giving a lot of false results as is evident from the figure 4. Here, the same roll number is being identified against various persons.

Figure 5: - Testing using tolerance value 0.3

After testing a lower value of 0.3, it was found that the number of recognitions has decreased to a large extent and most of them were rejected. So, then a higher value was required for proper identifications.

Figure 6: - Testing using tolerance value 0.48

At last, after various trials, it was found out that at around a value of 0.48 the model is able to generate a proper number of true results and the false values have decreased significantly with accuracy reaching even around 100% in certain well captured images.

The class was divided into three rows and pictures of each row were taken separately and the attendance sheet was generated.

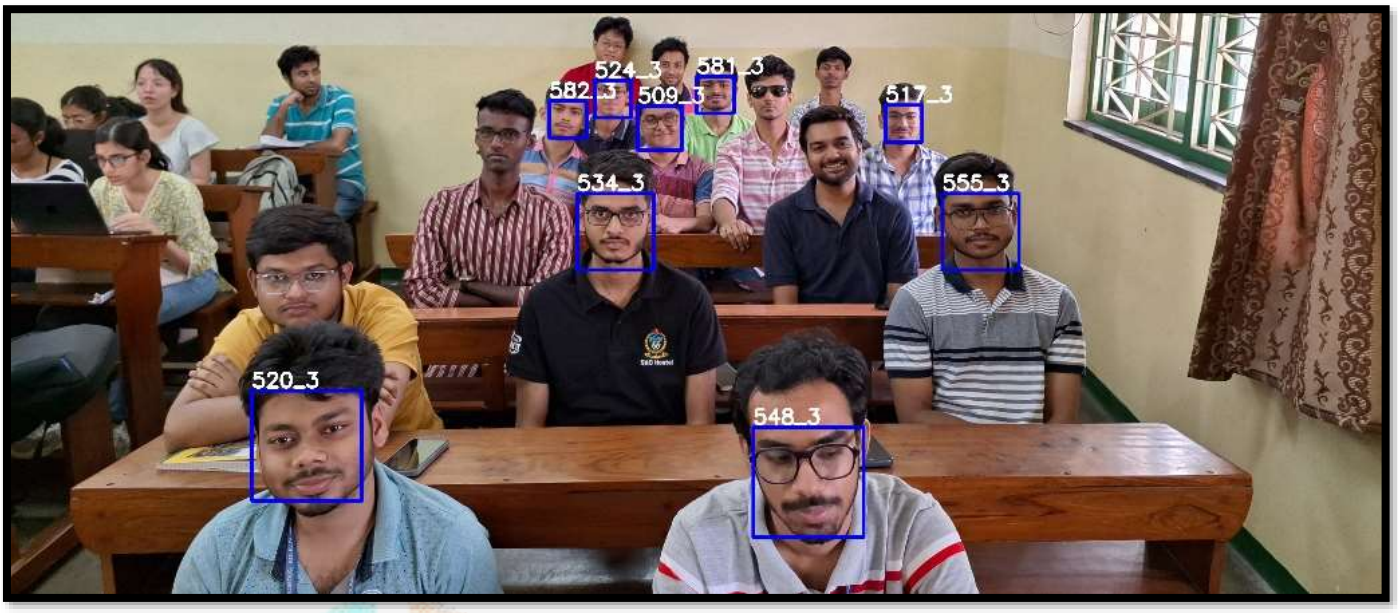

Figure 7: - Row 1 Recognition

Row 1: -

In this row, there was a false result being reported as roll 555. Since, the sample for the actual person was not present so, it showed it as a different person. However, if the sample would have been present in the database, there might have been a positive result.

Also, after having the data of another person, the model did not recognize him since he was wearing sunglasses, making the model a better choice to be implemented in classroom as someone who might be sleeping or will not show their actual facial features, the model will not recognize them.

Figure 8: - Row 2 Recognition

Row 2: -

In this row as well, a false result is being reported as 555. All the other rolls of whose data was available to the model, it gave us a correct prediction and a highly accurate result.

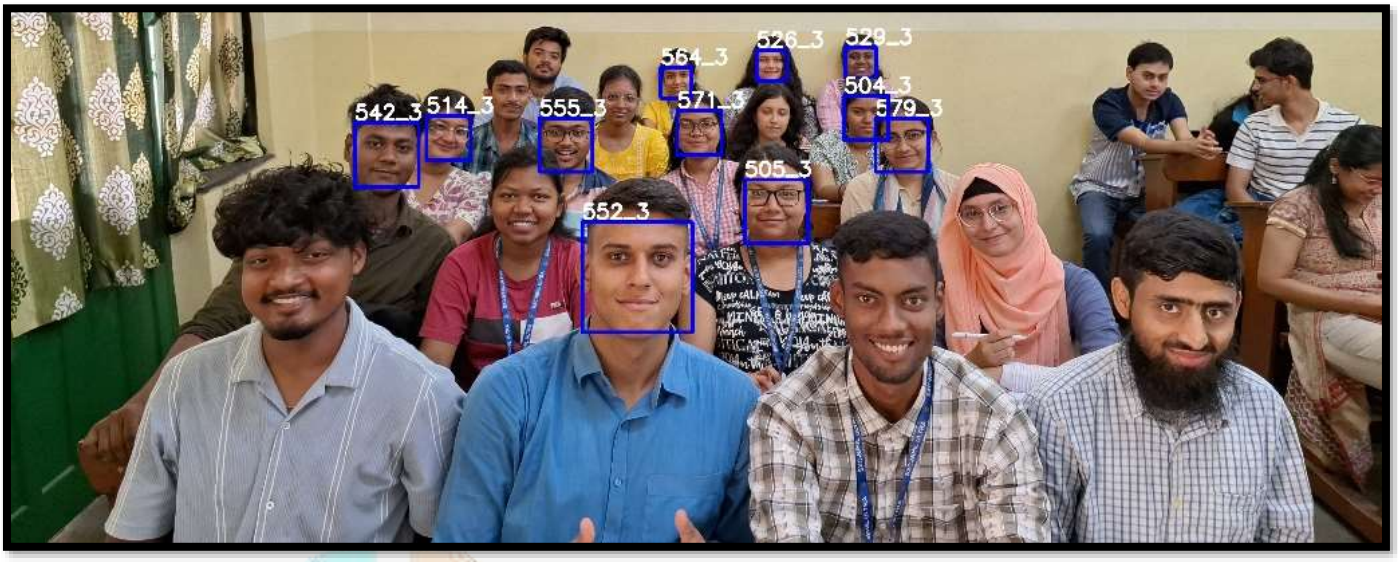

Figure 9: - Row 3 Recognition

Row 3: -

Here, even though the data of a person was present, he was not recognized due to the fault of the model. Also, one false result of 505 was reported.

All the students were not identified as their data was not present in the model due to privacy issues and not due to the fault of the model. A data of 41 students were used in the research with 5 images from each thereby a total of 205 images.

After all these recognitions, the accuracy of the model was taken out by the formula,

$$Accuracy = \frac{Total\ Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

Using this above-mentioned formula, we get,

Total Number of Predictions = 37

Total Number of Correct Predictions = 34

Therefore,

Accuracy = 34/37

= 0.918

Accuracy Percentage = 91.8%

However, if the missing predictions for which the model had data but could not predict or find it is also included then,

Total Number of Correct Predictions = 34

Total Number of Available Datapoints in the Image = 38

Therefore,

Accuracy = 34/38

= 0.894

Accuracy Percentage = 89.4%

VI. CONCLUSION AND FUTURE SCOPE

The model was able to provide an accuracy of 89.4%, just by using HOG and SVM classifier. With advanced GPU, MMOD CNN can be used to increase the accuracy even further. The face_recognition library of Python itself has a recorded accuracy of 99.38% on the Labeled Faces in the Wild benchmark. Thus, the accuracy can easily be heightened further.

While the smart attendance systems using face recognition offer several advantages, they also come with some limitations:

1. High resolution cameras are required to be installed which are very costly and should be placed in proper locations which can easily cover the entire classroom.
2. For greater accuracy we need to use MMOD CNN instead of HOG and SVM, which would require GPU processing for faster results, however can also detect tilted faces as well.
3. If someone's facial features changes, although there will be provisions for updation but if it changes in the middle of the academic year, maintaining the database might pose as a challenge.
4. If students occlude their faces, the attendance will not be recorded.

This smart attendance system has a large number of future applications. They are: -

1. If this is combined with the seat locations, it can be implemented in examination halls which would save the time of the students by saving them the time of signing the attendance sheets.
2. The accuracy needs to be increased much more for stricter real-world applications.
3. Since, the data of the students will be available, miscreants around the college can easily be tracked down.
4. The algorithm has been written for a single department. For total college, the tables need to be changed accordingly and needs to be expanded.

REFERENCES

[1] Helmi, R., Yusuf, S., & Jamal, A. (2019), Face recognition automatic class attendance system (FRACAS), IEEE international conference on automatic control and intelligent systems (I2CACIS 2019), Selangor, Malaysia, June 29, 2019.

[2] Xing, Y. (2019), A class attendance system based on SL4A, IEEE 4th international conference on cloud computing and big data analytics, April 12-15,2019 in Chengdu, China

[3] Mijić, D., Durutović, J., Bjelica, O., & Ljubojević, M. (2019), An improved version of student attendance management system based on RFID, 18th international symposium INFOTEH-JAHORINA, March 20–22, 2019 in Jahorina mountain, Republic of Srpska, Bosnia and Herzegovina.

[4] Timo Ahonen et al., "Recognition of blurred faces using local phase quantization", ICPR19th International Conference on Pattern Recognition, Dec 8-11,2008 in Tampa, Florida, USA.

[5] F. P. Filippidou and G. A. Papakostas, "Single Sample Face Recognition Using Convolutional Neural Networks for Automated Attendance Systems", Fourth International Conference on Intelligent Computing in Data Sciences (ICDS), October 21-23,2020 in Valencia, Spain.

[6] P. Wagh, R. Thakare, J. Chaudhari and S. Patil, "Attendance system based on face recognition using eigen face and PCA algorithms", International Conference on Green Computing and Internet of Things

(ICGCIoT), pp. 303-308, October 8, 2015 in Greater Noida, Delhi, India.

[7] P. Visalakshi, Sushant Ashish, Assistant Professor, Department of Computer Science and Engineering SRM Institute of Science and Technology, Chennai, Tamil Nadu, India, Attendance System Using Multi-Face Recognition, International Journal of Pure and Applied Mathematics, Volume 118, Issue Number 22, ISSN:1314-3395, Year of Publication: 2018.

[8] Ch. Vinod Kumar, Dr. K. Raja Kumar, PG Scholar, Dept of CS& SE, Andhra University, Vishakhapatnam, AP, India, Face Recognition Based Student Attendance System with OpenCV, International Journal of Advanced Technology and Innovative Research, Volume 8, Issue Number 24, ISSN: 2348-2370, Year of Publication: December 2016.

[9] OpenCV Documentation https://docs.opencv.org/2.4/modules/contrib/doc//facerec_tutorial.html.

[10] Face Recognition Documentation https://face-recognition.readthedocs.io/en/latest/index.html.