A Secure Attendance System using Raspberry Pi Face Recognition

Rhouma Bin Hamed
College of Applied Sciences (CAS), UTAS, Salalah, Sultanate of Oman
Tarek Fatnassi
College of Applied Sciences (CAS), UTAS, Salalah, Sultanate of Oman

Abstract: This study aims to develop a machine-learning-based attendance management system using face recognition and Raspberry Pi. The proposed system is composed of two main subsystems. The first is a Raspberry Pi, to be installed in each class, and the second is a web application fed by data from the Raspberry Pi. To take attendance, an instructor commands a Raspberry Pi camera through a web-based subsystem. Then, the camera takes a picture of the whole class and detects faces using trained Haar Cascades. It sends back a file with the class picture and Cartesian coordinates of the detected faces. The web application parses the file, looking for the coordinates of faces. For each Region of Interest, it uses the Support Vector Machine algorithm to recognize faces based on their HOG (Histogram of Oriented Gradients) features. The recognizer uses a pre-built dataset of that particular class containing the students' personal photos, names and ID numbers. Features of each face were extracted using HOG and trained to construct the model over a given class of students. Once every detected face is recognized, the application generates a report for the instructor showing the list of students' names and attendance status.

Keywords: Machine learning, SVM, Histogram of Oriented Gradients (HOG), Haar Cascade, Raspberry Pi

Introduction

In this digital era that relies on the use of artificial intelligence (AI), setting up a fully automatic system in a university that allows the recording and management of students' attendance based on face recognition has become an urgent need. Indeed, the attendance system that is currently used by our university is semi-automatic and has many limitations, especially in terms of the time and energy teachers spend on entering attendance. The majority of access control and attendance management systems that exist in the present time are based on

http://doi.org/10.18080/jtde.v10n2.530

various technologies, such as fingerprint, RFID (Radio-Frequency IDentification), and Iris recognition. These systems also have time-consuming issues, as they are queue-based and of an intrusive nature.

In contrast to these systems, having an attendance system based on face recognition is more effective as it has been successfully used in several contactless and non-invasive processes. Nowadays, artificial intelligence plays a significant role in all areas, as it helps in improving efficiencies and augmenting human capabilities. In the educational area, for example, AI techniques have been applied to manage the attendance of students in an easy and fast manner (Kamalahasan *et al.*, 2020). Every organization requires a strong and stable system to record class attendance, but they still mainly use a manual system that consists of using hard copies of student lists and asking students to sign next to their names during each class. Other organizations, such as UTAS, are using semi-automatic systems which have many limitations, especially in terms of time and energy. In addition, many organizations have adopted access control and attendance management systems that are fully automatic and use biometric characteristics for the identification of individuals, such as fingerprint, RFID, and iris. These systems are time-consuming, because they are queue-based and of an intrusive nature. However, the use of face recognition methods in attendance systems is considered efficient and smart. In fact, face recognition is faster and more accurate than other techniques, and it is efficient as it reduces the amount of effort a teacher can make while taking attendance (Thaware, 2018; Smitha, Hegde & Afshin, 2020) this paper we propose a smart attendance system that uses sophisticated methods to reduce the cost and reduce the error rate during the record of attendance. This system uses a Raspberry Pi camera to capture and recognize the students' faces. This automatic attendance system is cost-effective, easy and fast, and reduces the amount of effort teachers used to waste while dealing with manual systems. Besides, time and health safety in the present times of COVID-19 constitute a priority for this new system. Class attendance can be taken in a safer and more time-efficient way than the iris and fingerprint methods that are currently applied.

Literature Review

Having an automatic attendance management system became a necessity for educational institutions. Marking attendance manually is very difficult and error prone. Many researchers who are interested in this field tried to give relevant solutions to this problem. Most of the popular methods for attendance management include face recognition (Amritha & Sudhakar, 2019; Qian *et al.*, 2020; Evanjalin *et al.*, 2021; Kowsalya *et al.*, 2019).

http://doi.org/10.18080/jtde.v10n2.530

In Qian *et al.* (2020), they proposed an image decomposition scheme to explore the intrinsic structure from different gradient orientations and extend the matrix regression-based classifier for solving the robust face recognition problem. They provided a nice analysis of different image decomposition schemes and have shown that their algorithm succeeds in parsing the complex structure of the image in order to handle facial images with real disguises and illumination changes. However, their method costs more computation time when handling large-scale datasets, since they should compute the nuclear norm minimization problem by using the singular value shrinkage operator iteratively.

In Evanjalin *et al.* (2021), an automatic attendance system based on OpenCV and Raspberry Pi is proposed. The Raspberry Pi camera is used to take video of staff from many angles and under different illuminations. Still images are extracted from the registered videos. Then, Haar-cascades were used for face detection. A Deep Neural Network (DNN) was used to train the staff faces and store them in a mySQL database. This system requires too many computations and raises too many questions: from taking videos, when and what exactly the Raspberry Pi camera is ordered to take videos; how the trimming is done to extract face images; are Haar-cascades turned on videos or on extracted frames from the video. The use of the DNN is also not clearly explained for which features exactly the model is trained on: what features from the faces is the DNN working on? Their system also shows the output timing of a staff recognition. In that case, does the model detect when a staff member is exiting a door? If it does not recognise the existence of a staff member in a room, does it declare the staff member absent? There are too many issues that are not explained in that paper.

Automated Attendance Management Systems based on Face Recognition are used in classes to automatically detect faces of any students while entering the classroom, recognize them, then edit the database by marking them present. Face recognition requires samples to be collected, identified, extracted with necessary information, and stored for recognition. Face Detection is the basis in any operation performed in the face recognition process. The Haar Feature-based Cascade Classifier is a widely used mechanism for detecting faces. In order to train a classifier to detect faces, two large sets of images are formed, with one set containing images with faces, and the other set without. These images are then used to generate classifier models. The classifier is generated by extracting Haar features from the images. The detector and Haar-cascade extracts features quickly and it is easy to use. In our proposed attendance system, we use Haar cascades only in the detection stage. Then, we pass the detected faces to the SVM classifier to analyse the detected faces and label them with the right names/IDs.

Histogram of Oriented Gradients

HOG is a feature extraction technique that is used in computer vision and image processing (Dalal & Triggs, 2005). It is essentially based on HOG descriptors, which are used also for quantifying and representing both shape and texture. This technique is described according to the distribution of the edge trends in the picture, which will be divided into small cells. There are multiple pixels in each cell. For every group of each cell, a histogram of gradient directions is computed. The HOG feature is formed by concatenating all the histograms over all the image. For better resolution, HOG contrast is done by calculating the intensity over the largest region of multiple cells, which is named a block, and then this value is used for normalizing all cells within that block. The normalized result gives better performance on the variation in illumination and intensity. The HOG feature returns a real-valued feature vector. The dimensionality of this feature vector is dependent on the parameters chosen for the orientations, pixels per cell, and cells per block parameters. HOG features are extracted from large numbers of facial images to be used as part of the recognition mechanism (Rosebrock, 2021). These HOG features are then labelled together for a face/user and a Support Vector Machine (SVM) model is trained to predict faces that are fed into the system.

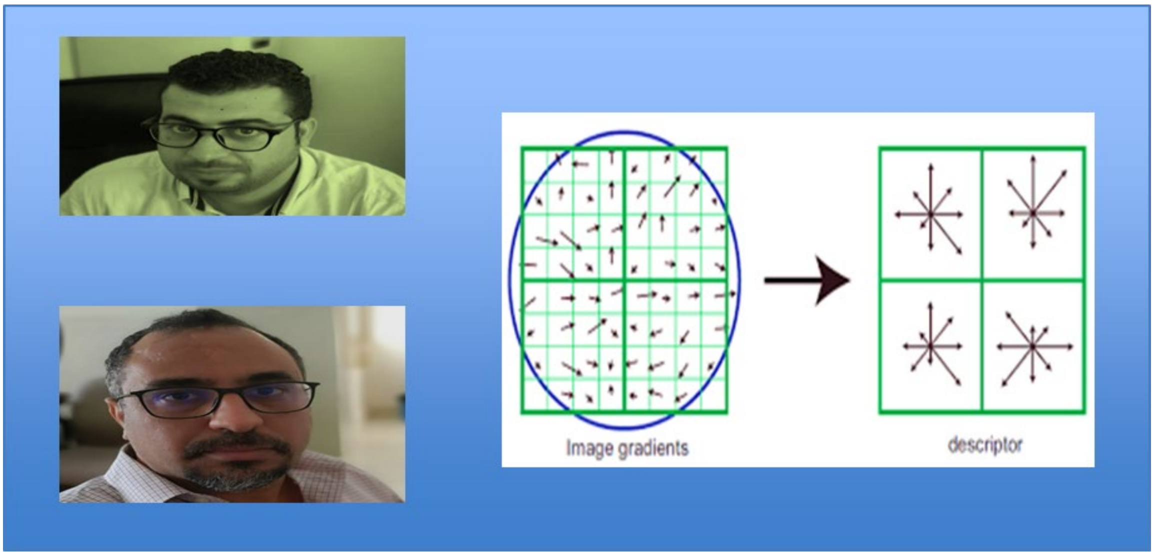

Figure 1. Histogram of Oriented Gradients from different faces

Support Vector Machines

Support Vector Machines (SVMs) are designed to solve the classic two-layer pattern recognition problem (Vapnik, 1998; Fradkin & Muchnik, 2006). SVMs are a set of supervised learning methods used for classification, regression and outlier detection. SVM uses a subset of training points in the decision function, called support vectors. A support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which

http://doi.org/10.18080/jtde.v10n2.530

can be used for classification, regression, or other tasks like outlier detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since, in general, the larger the margin, the lower the generalization error of the classifier. Figure 2 reflects the separation of groups of samples by a hyperplane where the classifier chooses the perfect hyperplane giving the maximum margin between the hyperplane and the nearest sample representing a specific class.

In this paper, we adopt SVM for face recognition by modifying the interpretation of the SVM classifier output and creating representations of facial images that are compatible with a two-layer problem. We are concerned with the following two categories: the differences between images of the same individual; and dissimilarities between images of different people. These two classes are the inputs to the SVM algorithm. The SVM algorithm creates a decision surface that separates the two classes. For face recognition, we re-interpret the resolution surface to produce a measure of similarity between two facial images. This allows us to build face recognition algorithms. We demonstrate the SVM-based algorithm in both validation and identification applications. In identification, the algorithm is presented with a photo of an anonymous person. The algorithm provides the best estimate of the identity of the person from a database of known individuals. In a more general response, the algorithm will present a list of the most similar individuals in the database. In verification (also referred to as authentication), the algorithm is presented with a photo and alleged identity of the person. The algorithm either accepts, rejects or calculates a confidence measure of the validity of the claim (Qutbuddin & Larik, 2021).

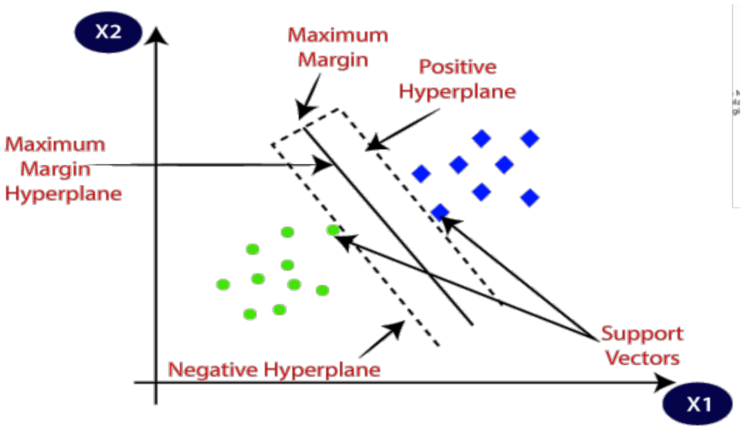

Figure 2. Support Vector Machines

The recognition of faces in the video sequence of the class is divided into three basic tasks: face detection, face prediction, and face tracking. Before starting the process of face recognition, a vector of HOG facial features is extracted. This vector is then used in the SVM model to determine the degree of matching of the input vector with each of the labels. The SVM returns the label of the face that represents the confidence to the closest match within the trained face data (Thaware, 2018).

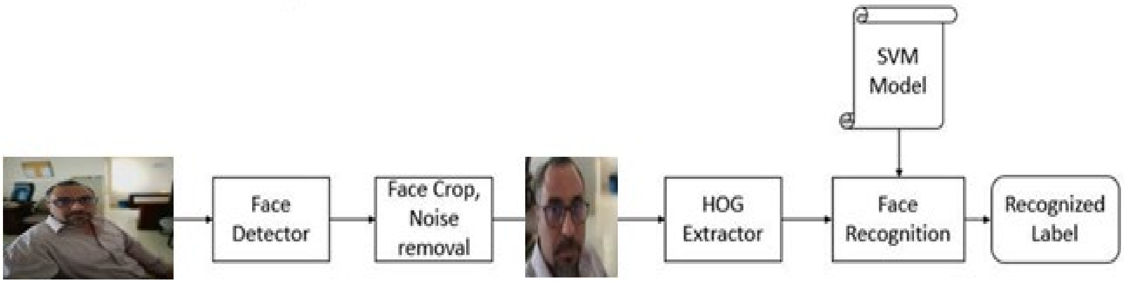

Figure 3. Block diagram of the face recognition process with Support Vector Machines

Methodology

In this paper we propose an accurate and cost-effective smart attendance management system that uses face recognition techniques. Additionally, this system uses a Raspberry Pi to capture and detect the students' faces.

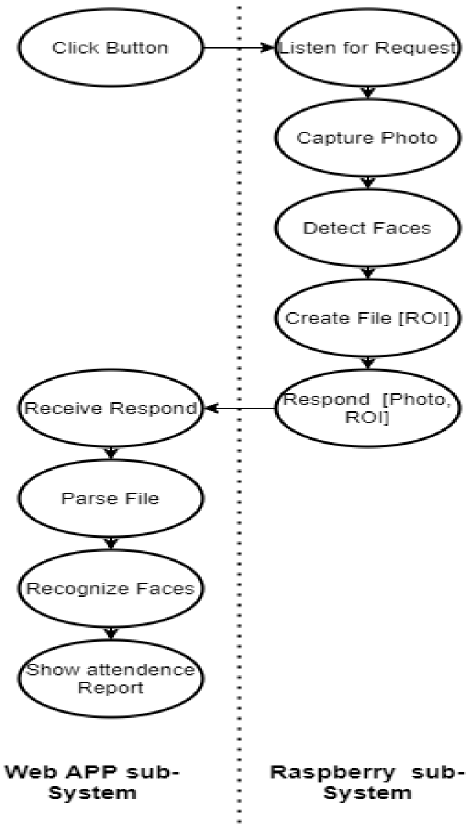

Figure 4. The proposed attendance system workflow

The proposed system is composed of two main subsystems that can communicate together by exchanging relevant data. The first one is a Raspberry Pi system, which is going to be installed

http://doi.org/10.18080/jtde.v10n2.530

in each class, and the second one is a web application that will be fed by data sent by the first subsystem.

The procedure is shown in Figure 4. First, an instructor will start a Raspberry Pi camera by pushing a button from the web application (SIS). After that, the Raspberry Pi camera will take a picture of the whole class. This picture will be analyzed locally in the Raspberry Pi system via a script that detects all the faces existing in that image using the Haar feature-based cascade classifier. The output of this script is a file that contains Cartesian coordinates of the detected faces. This file will be sent to the web application to accomplish the face recognition task.

Once the file is received by the web application, it will be parsed and the coordinates of the faces will be fetched. Then for each Region of Interest (ROI), an SVM algorithm is used to recognize the detected face. The recognizer uses the pre-built dataset of that particular class, which contains personal photos of each student in that class saved with their particular name or ID number. Finally, once every detected face is recognized, the web application would update the attendance database and generate a report for the instructor showing the list of students' names along with their attendance status and a timestamp.

Just like any other biometric identification, face recognition requires samples to be collected, identified, extracted with the required information, and stored for a later recognition process. Moreover, the accuracy of face recognition is highly dependent on the quality and variety of the sample images. The variety of sample images can be obtained by capturing multiple images with multiple facial expressions and angles for the same face.

For illustration, Figure 5 shows an example of sample face capture with several emotions.

Figure 5. Sample face capture

In this project, the dataset that is used for training the face recognition model contains 77 images which are stored inside 6 different folders. Each folder holds specific individual photos and is labelled with his/her ID number. Prior to the recognition phase, a model should be trained by applying the SVM algorithm on the images of the dataset. As illustrated in Figure 6, the training phase is achieved in two basic stages, face detection and feature extraction. The detection process is accomplished by applying the Haar Cascade Classifier on the images of all the registered students, then delivering an xml file. Once a face is detected, it can be cropped and stored as a sample face image for analysis and feature extraction.

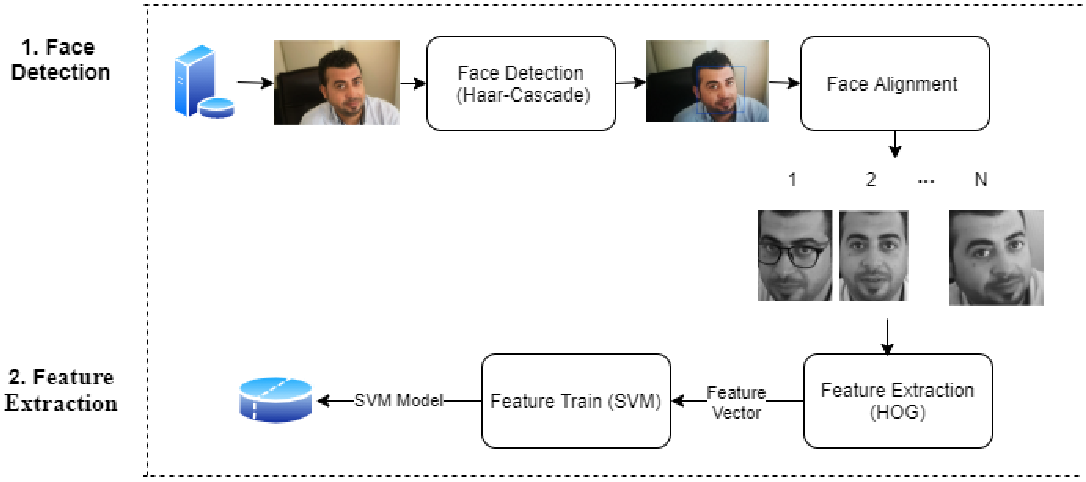

Figure 6. Face Training

After that, HOG (Histogram of Oriented Gradients) features are extracted from all the samples of face images that are related to a particular student. These HOG vectors of features are then labelled together for that specific user, forming an SVM label within an SVM model. As all the facial images of all users are extracted, an SVM model is generated and saved to recognize faces that are fed into the system afterward.

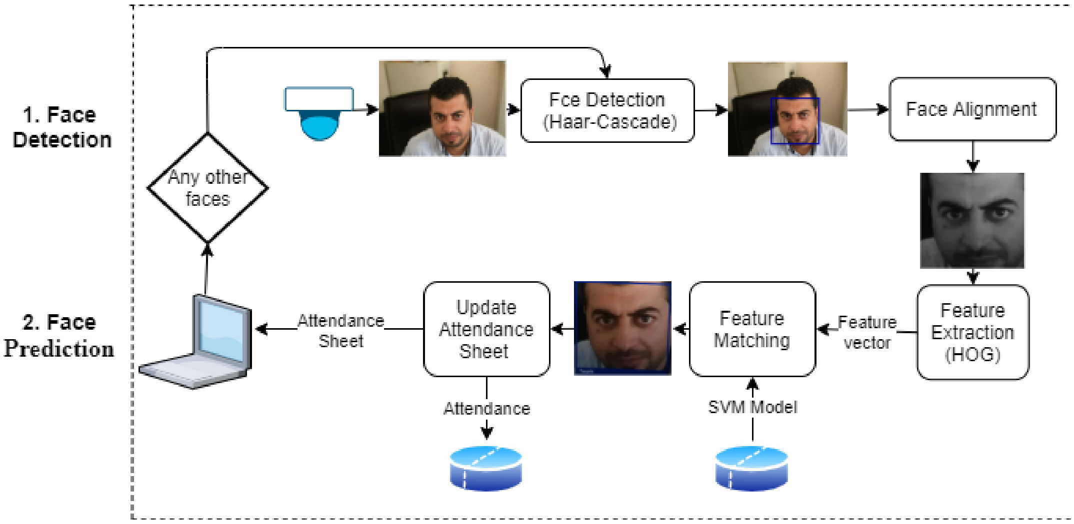

Figure 7. Face Recognition Model

The recognition of a face in an image is divided into two primary tasks, Face Detection and Face Prediction, as shown in Figure 7. The instructor accesses the system (SIS) and clicks a button to start recording the attendance. Accordingly, a request is sent to the Raspberry Pi camera, which is located in the classroom, to open and capture a photo of the whole class. The captured image is then passed through the Haar Cascade Classifier to detect all the faces shown in the photo. The classifier presents a bounding box around the detected faces. Next, the image is cropped to one certain face, normalized and kept as a facial image sample. Afterward, the facial image HOG features are extracted and compared with each SVM label in

 http://doi.org/10.18080/jtde.v10n2.530

the SVM model to predict the student's label (name/ID). The SVM classifier returns a label with the maximum matching score. Later on, the recognized label is assigned with the presence status in the attendance database. Back at the class photo, the recognized face is now shown with its label. The same process is then repeated with all the detected faces within the captured image. Finally, a completed attendance sheet is viewed by the instructor without his/her intervention.

Results and Discussion

The algorithm used to produce a smart attendance system utilizes the face recognition and dlib libraries besides the HOG method in Python. For the testing and evaluation of the attendance system, two experiments are done: one for measuring the performance of the face detection module; and another one for measuring the performance of the face recognition module, which is highly dependent on the accuracy of the first module.

To measure the performance of the face detection module, 100 images are placed in one folder (70 facial images and 30 not facial images). The results obtained by this test are: 65 True Positive (TP), 27 True Negative (TN), 5 False Negative (FN) and 3 False Positive (FP). Based on the result, the face detection accuracy is estimated using a Balanced Accuracy formula as follows:

$$BA = \frac{1}{2}\,(TPR + TNR)$$

where TPR (true Positive Rate) is given as the Ratio of TP over the P (Positive):

$$TPR = \frac{TP}{P}$$

and TNR (True Negative Rate) is given as the Ratio of TN over the N (Negative):

$$TNR = \frac{TN}{N}.$$

In the above experiment, we get TPR = 92.8 % and TNR = 90 %. Hence, we get an accuracy Rate BA = 91.4 %. Figure 8 shows a sample of true positive outputs and Figure 9 shows an example of a false negative case.

Moreover, a Python script has been created to calculate the face detection accuracy score, the error rate, a drawing of the Receiver Operating Characteristic (ROC) curve, and the area under the ROC curve (AUC), based on the above results.

Figure 10 shows the plotted ROC curve along with the AUC percentage for the face detection module. The ROC curve shows the trade-off between sensitivity (TPR) and specificity (1 –

http://doi.org/10.18080/jtde.v10n2.530

FPR). The blue dash line represents a random classifier where FPR = TPR. The orange curve symbolizes our face detection classifier. Generally, classifiers that give curves closer to the top-left corner indicate a better performance (with higher TPR and Lower FPR). Thus, the orange curve denotes a great classifier for face detection. Similarly, we can indicate that an AUC = 0.8 summarizes the performance of the classifier.

Figure 8. Correctly detected faces (True Positives)

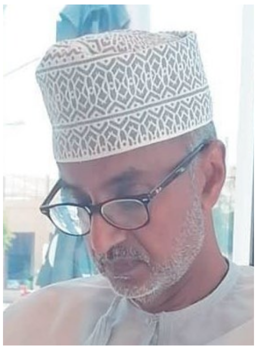

Figure 9. Incorrectly detected face (False Negative)

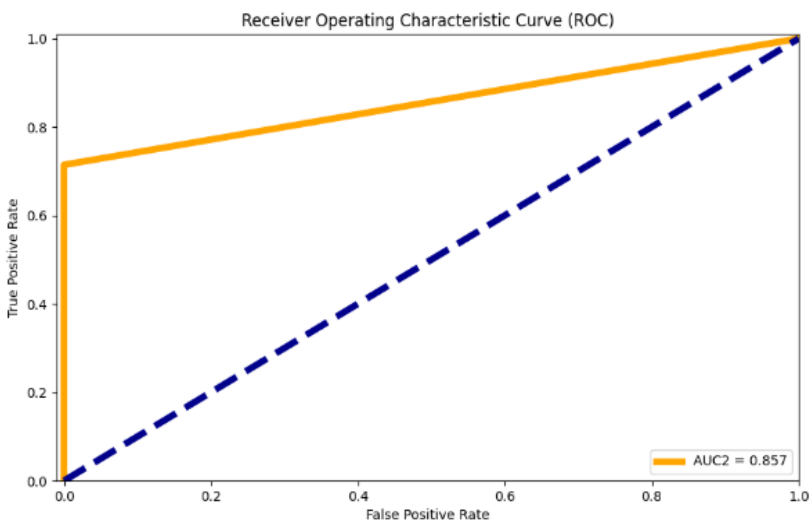

Figure 10 Face detection ROC curve

http://doi.org/10.18080/jtde.v10n2.530

To test the face recognition module of the attendance management system, we placed in a folder 8 images of 8 different faces. Five of these images are of persons who have been used for the training of the model while the rest are of other persons (which means unknown to the system). In this particular experiment, TP stands for a correctly recognised face; TN stands for a correct unrecognised face classified as "unknown"; FP stands for an unknown face recognised as existing in the class; FN stands for a face in the class classified as "unknown". A special concern here is to minimize the FN rate due the fact that the attendance system will mark the student as absent when he/she is present, which should be avoided if deployment of the system is to be considered in the near future.

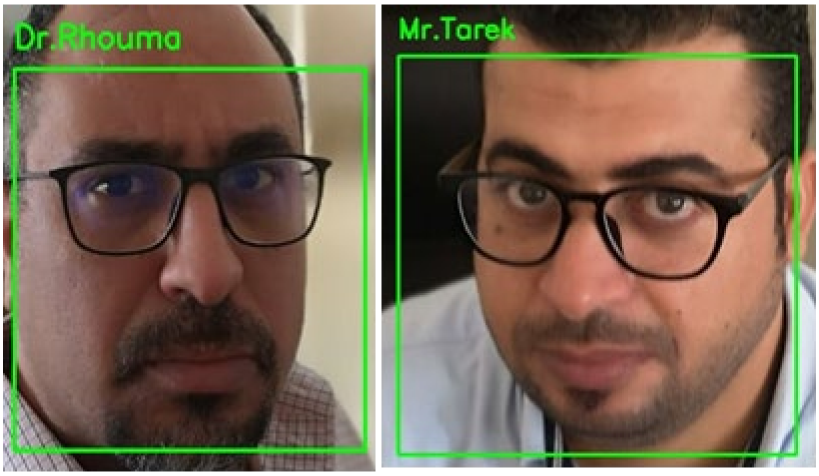

Figure 11. Correctly recognised faces (True Positives)

The obtained result indicates 5 TP, 1 TN, 0 FN and 2 FP. These values give a balanced accuracy of BA = (TPR + TNR)/2 = (100 + 33.3)/2 = 66%; but note that FNR = 1 – TPR = 0%, which is the most important feature that we need to accomplish. Figure 11 displays a sample of true positive outputs. Consequently, Figure 12 shows an example attendance.json file updated according to the recognized faces. Additionally, Figure 13 shows an example of a true negative output.

Fig.14 shows the plotted ROC curve along with the AUC percentage that reflects the face identification efficiency. The face recognition classifier (orange curve) demonstrates a good performance with AUC = 0.6.

```
[
{
"name": "Mr.Mickel",
"status": "present"
},
{
"name": "Dr.Brown",
"status": "present"
},
{
"name": "Mr.Layeeq",
"status": "absent"
}
]
```

Figure 12. An example attendance.json file

Figure 13. True negative output

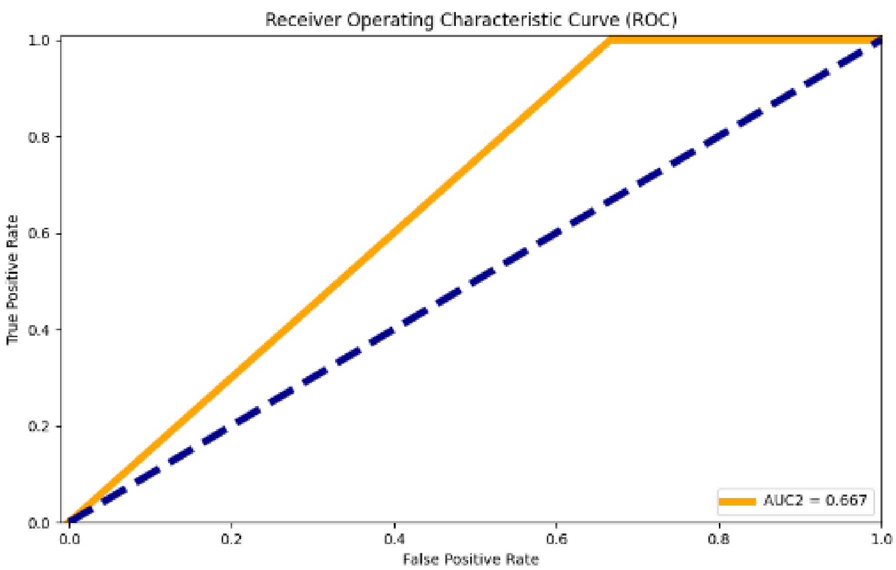

Figure 14. Face Recognition ROC curve

http://doi.org/10.18080/jtde.v10n2.530

Conclusion

The attendance system that we have designed using a Raspberry Pi, Python programming, machine learning/deep learning libraries and Django web Framework, needs further work regarding the SIS system which is the front-end for the instructor. Switching to a semi-automatic attendance system can gain time spent in every class marking the attendance register manually by calling students one by one. However, one should recheck the results since no system is 100% efficient. The recognition phase can be done by using another algorithm besides HOG/SVM like HOG/Deep Learning approach.

Acknowledgement

The authors are supported by a 2020 TRC Project number BFP/RGP/ICT/140 with Dr Rhouma as Principal Investigator (PI) and Mr Tarek as Co-PI for the benefits of the College of Applied Sciences (CAS) in Salalah. Currently, CAS is being merged with the University of Technology and Applied Sciences (UTAS).

References

Amritha & Sudhakar (2019). Face Recognition based Attendance System using Machine Learning. *International Journal of Engineering Development and Research*, *7*(3), 541–549.

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 886—893. https://doi.org/10.1109/CVPR.2005.177

Evanjalin, A. B., Christy, D., Karthika, N., & Reshma, R. S. (2021). Face Recognition System Attendance System using Raspberry Pi. *Irish Interdisciplinary Journal of Science & Research*, *5*(2), 84–91.

Fradkin, D., & Muchnik, I. (2006). Support Vector Machines for Classification. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. *70*, 13–20.

Kamalahasan, M., Gayathri, S., Swapna, B., Hemasundari, H., Apoorva, K., Wasim Akthar, J., Sudharsan, R., Saritha, V., & Nithin Kumar, N. (2020). AI Powered Face Recognition based Attendance Marking System. *Journal of Advanced Research in Dynamical and Control Systems*, *12*(04), 1607–1611. https://doi.org/10.5373/JARDCS/V12SP4/20201640

Kowsalya, P. K., Pavithra, J., Sowmiya, G., & Shankar, C. K. (2019). Attendance Monitoring System Using Face Detection & Face Recognition. *International Research Journal of Engineering and Technology (IRJET)*, *6*(3), 6629–6632.

Qian, J., Yang, J., Xu, Y., Xie, J., Lai, Z., & Zhang, B. (2020). Image decomposition based matrix regression with applications to robust face recognition. *Pattern Recognition*, *102*, 107204. https://doi.org/10.1016/j.patcog.2020.107204

Qutbuddin, S., & Larik, S. (2021). Face Recognition for Automated Attendance using HOG & Machine Learning. *Conference on Face Recognition using HOG model.* Karachi, Pakistan.

Rosebrock, A. (2021). Histogram of Oriented Gradients and Object Detection. *PyImageSearch.* Available at https://pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/

Smitha, Hegde, P. S., & Afshin. (2020). Face Recognition based Attendance Management System. *International Journal of Engineering Research*, *9*(5), 1190–1192. http://dx.doi.org/10.17577/IJERTV9IS050861

Thaware, R. (2018). Real-Time Face Detection and Recognition with SVM and HOG Features. [online] EEWeb. Available at https://www.eeweb.com/real-time-face-detection-and-recognition-with-svm-and-hog-features

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience, New York.