Automated Student Attendance System Using Face Recognition

E. Omer Akay
Dept. of Mechatronics Engineering
Istanbul Bilgi University
Istanbul, Turkey
omer.akay@bilgiedu.net

K. Oguz Canbek
Dep. of Mechatronics Engineering
Istanbul Bilgi University
Istanbul, Turkey
oguz.canbek@bilgi.edu.tr

Yesim Oniz
Dept. of Mechatronics Engineering
Istanbul Bilgi University
Istanbul, Turkey
yesim.oniz@bilgi.edu.tr

***Abstract*—In this study, an automated attendance taking system is developed and implemented.Two different face detection algorithms, namely Histogram of Oriented Gradients and Haar-Cascade algorithms, are applied and their performances are compared. Deep learning based on convolutional neural networks (CNNs) is employed for the identification of the students in the classroom. Furthermore, a mask checking feature is also included as a measure against the Covid-19 pandemic. A graphical user interface (GUI) system is designed using Python.**

***Index Terms*—computer vision, face recognition, automated attendance**

I. Introduction

Being one of the sub-branches of computer vision, face detection and recognition systems were initially intended for public surveillance purposes. However, the ever-decreasing size, enhanced computational power and affordable prices of the transistors lead to an increase in the variety and number of the applications of face recognition systems in every day life, industry and academia.

In this study, a face recognition system is developed for taking attendance of students. Research suggests that students' attendance is directly proportional to the effective learning and student retention [1], [2]. Several automated models of attendance are used in schools and universities and they are extensively studied in the literature. One of these methods is utilizing a finger print reading device, which can be either placed in front of the classroom, or handheld [3]. A similar approach is employing an RFID card for taking attendance [4]. As the devices used in both methods can be used by a single student at a time, these methods pose a problem of time. Furthermore the RFID method is prone to fraudulent attempts, as any student can use their companion's card instead of them. Applications using Bluetooth communication protocol constitute another alternative, in which either the Bluetooth connection of students' mobile device is directly used [5] or special Bluetooth tags [6] or beacons [7] are utilized to generate the attendance list automatically.

Academic research on computer vision based approaches to attendance taking has gained has gained momentum in the recent years as these methods provide lower time consumption and higher accuracy compared to the conventional counterparts. In [8], the images taken by a video camera installed to classroom are used to detect and recognize the students as they enter the classroom, and to mark the attendance list. The performance of various feature extraction and classification methods is tested for a database of 80 individuals. In [9], Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) methods are used for feature extraction and Radial Basis Function (RBF) Network is used for the recognition task with a success rate of 82% for a dataset of 148 images of 16 individuals. In [10], succesive layers of Convolutional Neural Networks (CNN) are used to create face embedding as features for the face recognition task. Then a SVM is used as a classifier with a succes rate of 95% for a five-individual dataset, whereas in [11], an additional PCA method is applied to the CNN feature extractor, and Mahalanobis distance is used as a clasification method.

The main objective of this study is to develop a computer vision-based face recognition system for the purpose of automated attendance taking. For the face recognition classifier both traditional and machine learning computer vision methods are used. Additionally, a mask checking feature is also included to the system as a measure against the Covid-19 pandemic, which can be used to identify students who are violating the rules on face coverings.

The organization of the paper is as follows: In Section-II the methodology of the study is presented where the methods and algorithms used for the face detection, face recognition and medical face detection are explained and sample application results are presented. In Section-III, the details on the developed graphical User Interface are introduced. Finally in Section-IV concluding remarks and future work are summarized.

II. Methodology

In this study, an automated attendance taking system along with medical mask detector is implemented. In the face detection, structures of the same geometric shape such as eyes, nose and mouth along with their relations to each other are used to detect a human face in a digital image. On the other hand, in face recognition the similarity rates between previously recorded face data and a new face image are utilized to identify or verify a person. In general, the pose of the head,

978-1-7281-9090-7/20/$31.00 ©2020 IEEE

the light intensity of the environment, and the occlusion of the face can affect the accuracy of a face recognition system.

A. Face Detection

The first step in the face recognition is the detection of the face. For this purpose, two different face detection algorithms, namely Histogram of Oriented Gradients (HOG) [12] and Haar-Cascade [13], are applied and their performances are compared.

The histogram of oriented gradients (HOG) method relies on the contrast in different regions of an images. The main advantage of this method is that the effect of lighting, either excess or lack of it, has little to no effect on the gradients.

Assuming that for each pixel, $f(x, y)$ is corresponding to the color of the greyscale image at the pixel location (x, y), the gradient vector is corresponding to the color difference between the adjacent pixels in both x-axis and y-axis and can be computed as follows:

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} f(x+1,y) - f(x-1,y) \\ f(x,y+1) - f(x,y-1) \end{bmatrix} \quad (1)$$

The magnitude and the direction of the image gradient at each pixel can be determined using $\sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$ and $tan^{-1}\left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}\right)$, respectively.

Next, the image is divided into several cells, each consisting of adjacent, non-overlapping C-by-C pixels. For each block, a histogram of gradient orientations with B bins is generated. In the block normalization step, overlapping blocks with 2-by-2 cells are formed and the block feature b is formed as follows:

$$b = \frac{v}{\sqrt{||v||_2^2 + e}} \quad (2)$$

where v is corresponding to the non-normalized vector of all histograms in a block, and e is small constant used to avoid division-by-zero. These features are then used for detection of the objects, which is corresponding to faces in this study.

Haar-cascade is another method, in which the transition between the light and the dark areas are used to infer the type of the features which are then utilized to detect objects such as lines, edges, faces, eyes, vehicles etc. on a given image. When used for the face detection, generally three different types of Haar-like features are employed to determine face parts. These features are presented in Figure 1. In this method,the sum of the pixel values under the light area is subtracted from the sum of the pixel values corresponding to the dark area.

Applying Haar-like features with different sizes at different locations and performing the related calculations might be time consuming and computationally cumbersome. To alleviate this issue and to detect the most suitable features, generally AdaBoost algorithm is employed, which determines a threshold value which will classify positive and negative images. Once the suitable features are identified, the next step is grouping the features into different stages and applying them in a cascade manner one-by-one. If a region fails in one of the stages, that region will be discarded and the subsequent stages will not be implemented. Using the Haar-Cascade method, faces of different scales can be determined. However, the performance of this method drastically change with the pose of the head and it is less robust to occlusion.

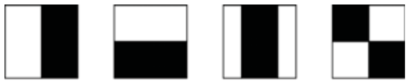

Fig. 1. Haar-like Features

In this study, both HOG and Haar-cascade methods are implemented using dlib [14] and OpenCV [15] libraries and their performances in terms of accuracy and required time are compared. A picture taken from a classroom with 19 students is used as the test image, which is first converted into greyscale. The results of both methods are presented in Figure 2 and 3, and summarized in Table I. Regarding the results, it can be observed that the HOG-algorithm can provide higher accuracy in comparable time. Therefore, in the subsequent parts of this study, HOG algorithm is utilized instead of the Haar-cascade method.

Fig. 2. Faces detected using HOG

Fig. 3. Faces detected using Haar-Cascade

TABLE I
HOG vs. Haar-Cascade Comparison

Metrics	*HOG*	*Haar-Cascade*
True Positive	18	18
True Negative	1	1
False Positive	0	8
Precision (%)	100	69
Recall (%)	100	100
F1-Score	100	82
Training Time (s)	40.3	37.3

B. Face Recognition

To reduce the dependency of the performance of the face recognition algorithm from the pose of the head, the face landmark estimation algorithm developed in [16] is applied, by which the 68 coordinates related with different facial structures like eyes, lip and nose are determined on the image. Figure 4 shows these coordinates on a face image from the dataset. Using these coordinates, various measurements including the width of the nose, the distance between eyes etc. can be made for each person. Instead of making direct comparisons between the measurements obtained from the image and those stored in the database, deep learning techniques can be utilized to identify the person on the image, as they provide better accuracy under changing environmental conditions.

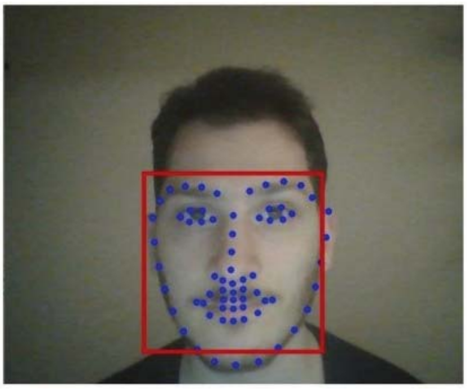

Fig. 4. Face landmarks

In this study, a convolutional neural network (CNN) architecture is used to encode faces into 128-dimensional vectors. The hidden layers of the network consist of two sets of consecutive layers, where each set comprises two convolutional layers followed by a max-pooling layer.. In the convolutional layers, the number of $3-by-3$ kernels is set to 32 and rectified linear unit (ReLU) is preferred over sigmoid or hyperbolic tangent as the activation function to decrease the computational load. In the output layer of the CNN, three densely connected layers are used. In the last fully connected layer, softmax activation function is used to compute the output of the network. To prevent overfitting, batch normalization and dropout methods are adopted.

During the training phase of the CNN algorithm images are used in triplets to increase the accuracy of the encoding, where two images belong to the same person whereas the third one is of another person. The objective function of the CNN algorithm updates the filters such that the 128 dimensional encodings of the images from the same person become closer while the third encoding shift further away from the two.

The database used to train the network consists of the images of students, who are taking the same class. The pictures are downloaded from the Student Management System of the university by the instructor and uploaded to the database with the name information for each one. After completing the "triplet" training on the image dataset, the weights of CNN model are adjusted to generate a 128-dimensional vector for each person, which are then used by a Support Vector Machine (SVM) classifier to identify the students.

To test the performance of the algorithm, the same image used in the face detection part is employed and the results of face recognition operation is presented in Figure 5. First the faces on the image are determined. For each face, a 128-dimensional vector is generated, which is compared with the pre-trained vector measurements in the dataset and the name for the person is retrieved from the database.

Fig. 5. Results of the face recognition

C. Medical Mask Detection

With the recent global pandemic, it has become essential to wear a mask when in public to keep both yourself and others safe. In this study, a new feature is included to the regular attendance system, which will enable to identify the students who do not wear masks in the classroom.

First, a dataset of masked faces is generated using the dataset of unmasked faces compiled by webscraping [17]. After locating the face and the facial landmarks on the unmasked face images, a mask image is placed on top of the face to create masked faces. During the training phase, data was loaded to detect the face mask. The same CNN structure explained in the previous subsection is trained for this new dataset. The flow diagram of the mask detection system is illustrated in Figure 6 and one of the sample results is presented in Figure 7.

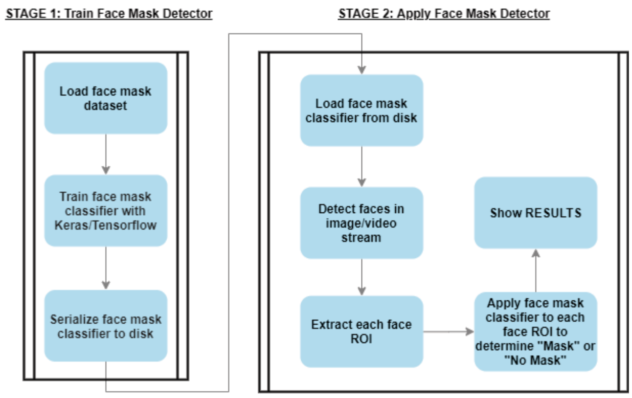

Fig. 6. Flow Diagram of mask detection classifier

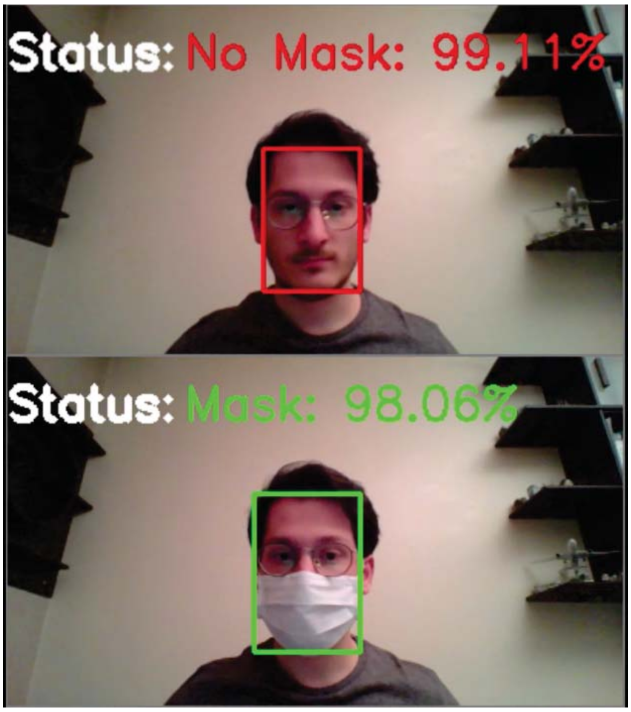

Fig. 7. Results of the mask detector

III. Graphical User Interface

A graphical user interface (GUI) is developed using the Python library PyQt5. The proposed automatic attendance system has the following capabilities (see also Figure 8):

- ”Register student” to enter the names and face images of the students to the database
- ”Take attendance” to compare face encoding in the image captured by the camera with the encodings available in the database and genertae the attendance list
- ”Train the system” to create student face encodings from their face images
- ”Send attendance using e-mail” to send the attendance list of a desired date to the instructor(s) via e-mail
- ”Delete the attendance list” to delete attendance lists of the specified dates
- ”Check the list” to see a list of the previously generated attendance lists

Fig. 8. Main menu window

In the developed system, instructors have their own usernames and passwords to enter the system. When the ”Attendance” button is pressed, a pop-up window will be opened to enable the user to enter the username and password. If the information provided by the user is correct, user can login the system by clicking the login button and use the provided features. On the other hand, if the provided information is incorrect, a window with an error message will be opened and all functions will be deactivated.

When the ”Take an Attendance” option is selected, the program first takes the photo of the classroom using the camera attached to the computer. Using CNNs, face recognition operations will be performed on the captured image and a list of the students identified on the image will be generated. A sample attendance list can be seen in Figure 9. This attendance list can be sent to the instructors both in text and Excel format.

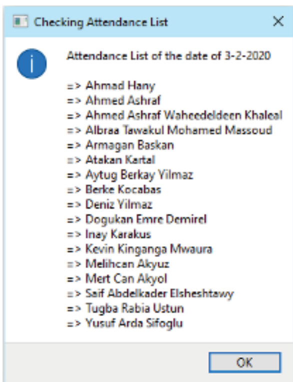

Fig. 9. Sample attendance list

The flow chart of the GUI operations is presented in Figure 10.

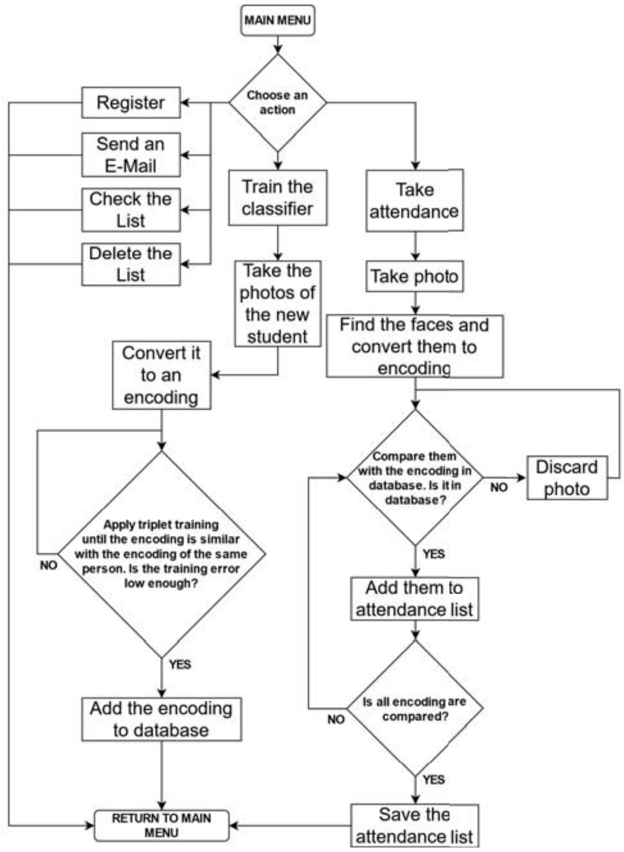

Fig. 10. Flowchart for the GUI

IV. Conclusion

In this study, an automated attendance taking system is developed and implemented. For the face detection part, two widely used face detection algorithms, namely Histogram of Oriented Gradients and Haar-Cascade algorithms, are applied and tested on an image taken from a classroom. Regarding the obtained results, it is observed that HOG algorithm provides better accuracy under changing lighting conditions. For the face recognition, deep learning based on convolutional neural networks (CNNs) along with a SVM classifier is employed. In addition, a mask checking algorithm is also developed which can be used in the classrooms for the identification students, who are not wearing a mask or wearing it improperly. A graphical user interface (GUI) system is designed, which enable instructors to generate the attendance automatically, to add new students or to check the attendance of a previous date.

In this work, the images of the students are saved to the database manually by the instructor. As a future work, it is intended to integrate the developed system into the Student Management System of the university/school, which can provide automatic generation of the database for each class.

References

[1] T. Fadelelmoula, "Impact of class attendance on student performance," *International Research Journal of Medicine and Medical Sciences*, Jun. 2018.

[2] D. O. Allen and D. J. Webber, "Attendance and exam performance at university: a case study," *Research in Post-Compulsory Education*, vol. 15, no. 1, pp. 33–47, Mar. 2010, publisher: Routledge _eprint: https://doi.org/10.1080/13596740903565319.

[3] B. K. P. Mohamed and C. V. Raghu, "Fingerprint attendance system for classroom needs," in *2012 Annual IEEE India Conference (INDICON)*, Dec. 2012, pp. 433–438, iSSN: 2325-9418.

[4] B. Dias, A. Mohammad, H. Xu, and P. Tan, "Intelligent Student Attendance Management System Based on RFID Technology," in *Complex, Intelligent, and Software Intensive Systems*, ser. Advances in Intelligent Systems and Computing, L. Barolli, F. K. Hussain, and M. Ikeda, Eds. Cham: Springer International Publishing, 2020, pp. 578–586.

[5] V. Bhalla, T. Singla, A. Gahlot, and V. Gupta, "Bluetooth based attendance management system," *International Journal of Innovations in Engineering and Technology (IJIET)*, vol. 3, no. 1, pp. 227–233, 2013.

[6] R. Lodha, S. Gupta, H. Jain, and H. Narula, "Bluetooth Smart Based Attendance Management System," *Procedia Computer Science*, vol. 45, pp. 524–527, Jan. 2015.

[7] M. Borić, A. Vilas, and R. Díaz Redondo, "Automatic Attendance Control System based on BLE Technology," Jan. 2018, pp. 289–295.

[8] S. Chintalapati and M. V. Raghunadh, "Automated attendance management system based on face recognition algorithms," in *2013 IEEE International Conference on Computational Intelligence and Computing Research*, Dec. 2013, pp. 1–5.

[9] S. Lukas, A. R. Mitra, R. I. Desanti, and D. Krisnadi, "Student attendance system in classroom using face recognition technique," in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2016, pp. 1032–1035.

[10] M. Arsenovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "FaceTime — Deep learning based face recognition attendance system," in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, Sep. 2017, pp. 000 053–000 058, iSSN: 1949-0488.

[11] E. Winarno, I. Husni Al Amin, H. Februariyanti, P. W. Adi, W. Hadikurniawati, and M. T. Anwar, "Attendance System Based on Face Recognition System Using CNN-PCA Method and Real-time Camera," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2019, pp. 301–304.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[13] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.

[14] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009, publisher: JMLR. org.

[15] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[16] V. Kazemi and J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," 2014, pp. 1867–1874.

[17] P. Bhandary, "prajnasb/observations," Oct. 2020, original-date: 2020-03-25T09:22:27Z. [Online]. Available: https://github.com/prajnasb/observations