Smart Attendance System using Face-Recognition

Mr. Rajvardhan Shendge[1], Mr. Aditya Patil[2], Mrs. Tejashree Shendge[3]

[1, 2]*Student, Computer Engineering, Ramrao Aidik Institute of Technology(India)*
[3]*Student, Electronics and Telecommunication Engineering, Fr. C. Rodrigues Institute of Technology(India)*

Abstract - Every institute, college, and organization, large or little, has an attendance marking system. We've built a method that uses the latter to simplify the former, thanks to huge advances in the field of image processing. Face Recognition is becoming more popular than other biometric verification methods due to its simplicity, non-invasiveness, and lack of touch. The system's major goal is to identify and recognize faces in a real-time environment, match them with data in the database, and record their attendance. This is intended to make the time-consuming manual attendance process more efficient. This also overcomes the issue of authentication and proxies because biometrics are one-of-a-kind, and facial traits used for Face Recognition are one of them. For face detection and recognition, the designed system uses OpenCV, dlib, Face Recognition libraries, and One-Shot Learning, which takes just one image per person in the database and so saves space when compared to standard training-testing models.

Key Words: face recognition, image processing, face detection, Siamese Networks

1.INTRODUCTION

There is an inherent positive relationship between students' attendance in schools and colleges and their academic performance, according to research [1]. And, in order to maintain this relationship, it is necessary to encourage their presence and performance in the classrooms, so that students are motivated to keep up with the progress of the subjects being taught in class, thereby increasing their participation in school/college. Attendance management systems have been implemented in schools, colleges, and universities all over the world using a variety of methods. Despite their high usability, the practicality of these systems is a little questionable. The face recognition-based attendance system is one such system that has recently gained traction. Face recognition is a technique for identifying, verifying, or distinguishing a subject based on an image or video of the subject's face. It employs a biometric identification method that uses facial and head measurements to verify a person's identity. Face recognition biometric systems use computer algorithms to pick out specific, distinguishing features of a person's face, such as the space between the eyes or the shape of the face. These characteristics are converted into a mathematical representation, such as an array or matrix, and compared to the characteristics of other faces in a face recognition database. A face encoding is data about a specific face that differs from a photograph in that it is designed to only include certain details that can be used to distinguish one face from another. This system requires any device with digital photographic technology, such as a webcam or a CCTV camera, to generate and obtain the images and data needed to create and record the biometric facial pattern [11]. Various facial recognition algorithms have been developed over the years to recognize people regardless of their environment, lighting, angle, or facial expression. Based on its performance in other security applications, it appears to be a promising approach for student attendance systems that can help solve problems associated with current systems. A system that uses facial recognition to assess students' attendance using machine learning algorithms is proposed in the proposed paper. The One-Shot Learning approach is used in the proposed system, which requires only one image of each student to train the system that will be used to detect their faces, generate their face encodings, and mark their attendance. The attendance will be recorded on an Excel sheet, which staff and faculty members will be able to assess and evaluate. We used a pre-trained deep neural network called face-recognition, which was built using dlib and has an accuracy of nearly 99.38 percent on the LFW dataset [12] to implement the OneShot Learning approach. To test the system's stability and robustness, we tested it on a few students from our college in various light settings, camera settings, and occlusions.

2. LITERATURE SURVEY

Various systems are currently in use to manage and assess student attendance at universities. Even though these systems are extremely usable, their practicality and constraints pose a problem in the process, as previously stated. The following are a few of the systems in place:

2.1 Manual attendance system

Manual attendance systems are traditional systems in which a teacher or lecturer takes students' attendance by calling names or signing an attendance sheet. Such attendance systems rely entirely on students acting in a fair and consistent manner. Although it is a low-cost system, it is extremely vulnerable to human error or manipulation. A student may be mistakenly marked present by the teacher if another student answers it on a roll-call, or a student can forge signatures on the sheet, resulting in 'proxy attendance' [16].

2.2 RFID-Based attendance system

Rfid is the abbreviation for Radio Frequency Identification. Students are given RFID cards, which are scanned using an RFID reader to mark their attendance at universities. The RFID system's main flaw is its lack of practicality. RFID tag cards are prohibitively expensive, and purchasing them for an entire university is not feasible. Another disadvantage is that if students are not supervised, they can scan multiple RFID cards on the reader, resulting in proxy attendance. If not supervised or cared for properly, the RFID reader is also susceptible to damage [17].

2.3 Bluetooth-based attendance system

Bluetooth-based systems collect information about the students present in the class and record their attendance using Bluetooth signals from their phones. This system appears to be very practical, as nearly 95% of college students have their phones with them. To make the system perfect, it can also implement proxy removal methods. However, the system's main flaw is its lack of usability. A Bluetooth-based device can only connect to 8 other devices at a time. This is due to the Master and Slave concept, which limits a device's connection to only eight other devices at a time. As a result, this system can only be used when the number of students in a classroom is in the single digits [18]. After more thought, it was discovered that every existing attendance management system had flaws that tainted the process. Problems caused by 'proxy attendances' will be eliminated by using facial detection and recognition as a parameter of attendance generation, as only those students present in the lecture will be marked present. Because every classroom has a laptop and a webcam, the components are also inexpensive. The main strategy is to compare the face encodings of the image captured in real-time with those already stored in the database, which can then be used to mark attendance if a match is found. A Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System was proposed in the paper by author Saypadith et al. [9]. Face detection and tracking were implemented using a Convolutional Neural Network (CNN). Author Deeba et al. [10] used a similar approach to develop a Local Binary Pattern Histogram (LBPH)-based Enhanced Real-Time Face Recognition system that can recognize faces in low and highlevel images in real time. A model of an automated attendance system was proposed by authors Akbar et al. [7]. Their system detects and counts students as they enter and exit the classroom using a combination of Radio Frequency Identification (RFID) and Face Recognition. It keeps track of each student's attendance records and provides pertinent information as needed. Author Smitha et al. [8] used Haar-Cascade Classifier and Local Binary Pattern Histogram (LBPH) for face detection and recognition in their automated attendance system. Faces were captured using a live stream video of the class in their system, and attendance was recorded, which could be accessed as a CSV file. For face recognition, author Sawhney et al. [3] use a hybrid algorithm that combines Eigenface, Principal Component Analysis (PCA), and Linear Discriminant Analysis (LDA). The facial features obtained through these algorithms can then be used to identify students and, as a result, mark their attendance. Authors Kiran et al. [6] developed a face recognition attendance system that employs Eigenface, Haar Cascade Classifier, and Principal Component Analysis (PCA) algorithms. Their method was to take real-time images of students, compare the extracted Eigenvalues to those in the database, and mark attendance based on the recognition result from PCA analysis. This system had a 97% accuracy rate when tested on a database with images from 70 students. The attendance system proposed by D'Souza et al. [4] is based on the Haar Cascade Classifier and the Local Binary Pattern Histogram (LBPH) algorithm. Their proposed system would take group photos of students during class hours, perform facial segmentation and identification, and update attendance accordingly. Harikrishnan et al. [5] created an attendance system using the Haar Cascade Classifier and the Local Binary Pattern Histogram (LBPH) algorithm in another implementation of a similar system. Their system achieved a maximum accuracy of 74% when used in various environments such as lighting and occlusions.

3. METHODOLOGY

3.1 One-Shot Learning Model

The One-Shot Learning Model is the foundation of our system. It's a classification task in which one sample is used to classify a large number of future samples. Face-recognition systems based on the One-Shot Learning Model learn a rich low-dimensional feature representation known as a face encoding, which can be easily calculated for faces and compared for verification and identification tasks [14]. Consider the case of a face recognition system for a timekeeping system. Images of multiple faces make up the input dataset. Convolutional Neural Networks (CNN)-trained models require a large number of images to train and achieve high accuracy. If there are a few minor changes in the dataset, these models must be trained iteratively. If a student drops out of college, the dataset must be updated by deleting the student's images, and the model must be retrained. In addition, when a new student is admitted to college, their images must be collected, and the model must be retrained. This procedure consumes a significant amount of time and manpower. To address this, the One-Shot Learning model can be used, which takes a lot less time to train because only one image of the student is required. The Siamese Network is widely used to implement the One-Shot Learning Model. Figure 1 shows how the Siamese Network performs facial recognition. A similarity function is the foundation of any Siamese Network. A Siamese Network's architecture consists of two parallel networks, each taking a different input, and combining their outputs to generate a prediction. A Siamese Network for face recognition is a neural network that learns a function f(d) that takes two input images, one from the dataset called the actual image and the other from outside the dataset called the candidate image, and the output is the similarity between the two images. If the two images passed through the network have a small distance between them, they can be classified as the dataset's actual image [21].

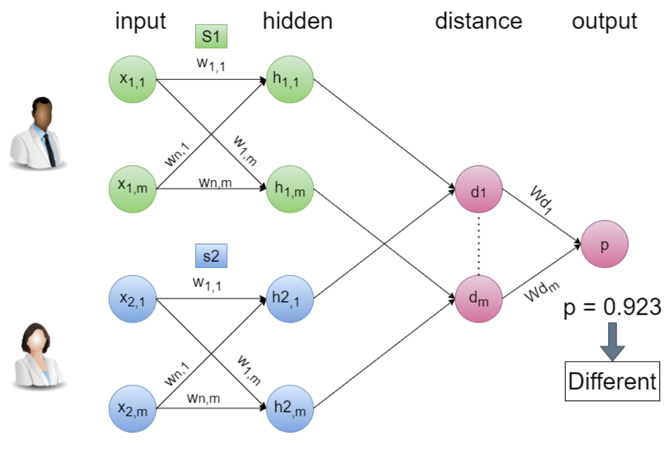


Fig. 1. Siamese Network for Face Recognition

These images are passed through similar networks called sister networks, which are similar in terms of their parameters and shared weights, in order to train the neural network to learn how to compute similarities between two images, the actual image and candidate image. These sister networks are made up of a series of convolutional, pooling, and fullyconnected layers that produce a fixed-size feature vector denoted by h1 as an encoding of the actual image Image1. The difference —h2- h1— between the encodings of the two images passed is the distance between their encodings. The value of —h2 -h1— is relatively small if the two images passed are of the same person. The workings of sister networks are depicted in Figure 2.

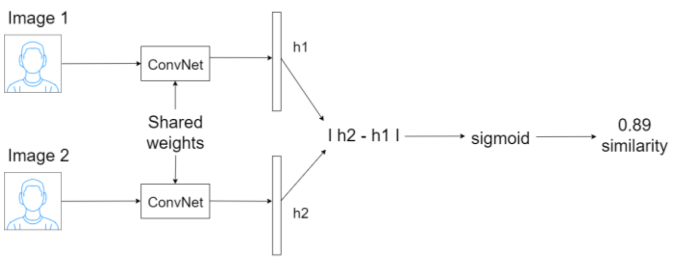


Fig. 2. Feature vector extraction in sister networks

3.2 dlib's HoG Face Detection

Histogram of Oriented Gradients (HoG) is an abbreviation for Histogram of Oriented Gradients. HoG's main idea is to turn facial features from an image (or a real-time video) into a vector and feed it into a classifier like SVM (Support Vector Machine) to detect the presence of a face in an image. The histograms of directions of gradients, or oriented gradients of the image, are the names given to these extracted features. Gradients are large around edges and corners in general, and they allow us to detect regions of interest (ROI) [20]. This method for detecting human bodies was developed by Dalal et al. [19]. The images are first preprocessed by being cropped and scaled to the appropriate size. The image gradients must be calculated as the first step in face detection. These gradients are calculated to remove all non-essential elements from an image, such as background noise, leaving only the region of interest (ROI). Kernels are used to compute the horizontal and vertical gradients, as shown in fig.3. [20].

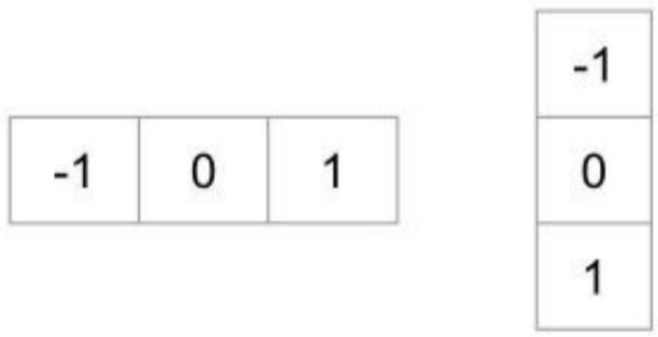


Fig. 3. Kernels applied to compute gradients [20]

After gradient computation, the image is divided into 8x8 cells to create a compact representation, making our HoG more noise resistant. Then, for each of these cells, HoG is calculated. The gradient's direction inside a region is estimated by creating a histogram from the 64 gradient directions and their magnitudes within each region. The histogram is divided into nine categories that correspond to angles ranging from 0 to 180 degrees. The temperatures are 0°, 20°, 40°, 80°, and 160°[20].

While building an HoG, 3 subcases arise as follows:

1.)If the angle is smaller than 160° and it is not halfway between 2 classes, the angle will be categorized in the right category of HoG [20].

2.)If the angle is smaller than 160°and it is exactly between 2 classes, then the angle contributes equally to both the bounds, and the magnitude is divided by 2 [20].

3.)If the angle is greater than 160°, the pixel is considered to contribute proportionally to 160° and 0° [20].

Finally, a 16x16 block is used to normalize the image, making it insensitive to things like lighting. The value of the 8x8 sized HoG is divided by the L2- norm of the HoG of the 16x16 block that contains it, which is a vector of length 36. The feature vector is created by concatenating all of the 36x1 vectors into a single large vector that can be used to train an SVM (Support Vector Machine) classifier and used for face detection using the dlib library [20].

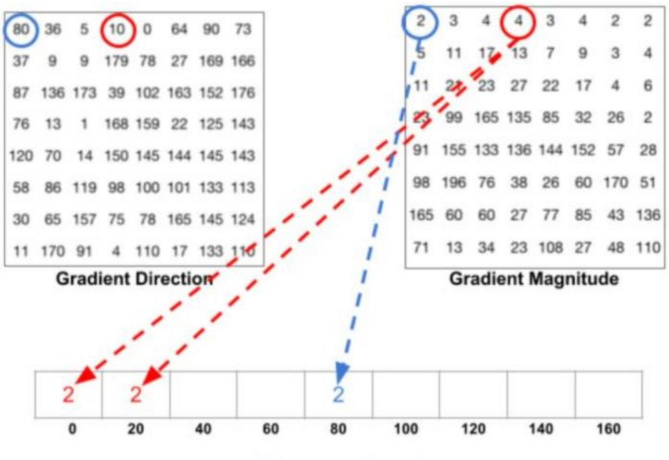


Fig. 4. Subcase 2 of HoG building [20]

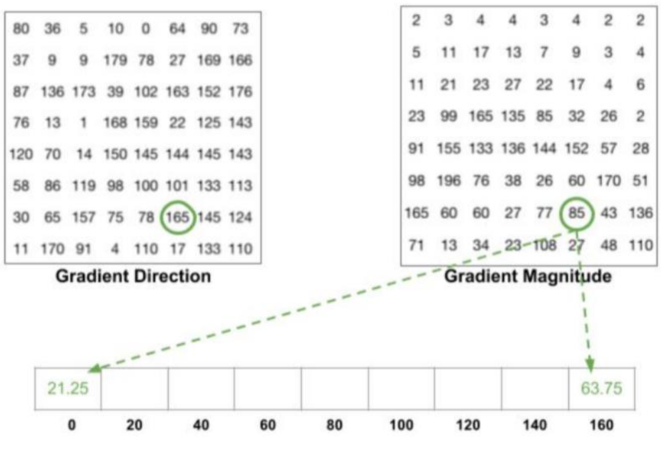


Fig. 5. Subcase 3 of HoG building [20]

4. CONCEPTUAL ARCHITECTURE

Our method entails assessing student attendance using only one image per student from the class, captured using a webcam connected to a laptop or desktop computer. All of the students in the class must register on the device by entering their information, and each student's image will be captured and saved in the dataset. The student will be asked to register their attendance using the device that the system is running on before each lecture. The system will then detect the face, calculate the encodings of the face, and compare them to those in the dataset. The student will be marked present for the lecture if there is a match. This attendance information will be saved in a CSV file that the class's faculty/lecturer can easily access.

This process can be primarily divided into 4 stages:

4.1 Image acquisition for dataset creation

We used images of 50 students from our own college to create our dataset. These photographs only show the students frontal faces. Only one image is used per student. After that, the images in the dataset are preprocessed. The images are first cropped in preprocessing so that only the Region of Interest (ROI) is available for further detection. After that, the cropped images are resized to a specific pixel position. After resizing the images, the cv2 module of the OpenCV library is used to convert them from BGR to RGB. Finally, the processed images are saved in the dataset along with the student's name.

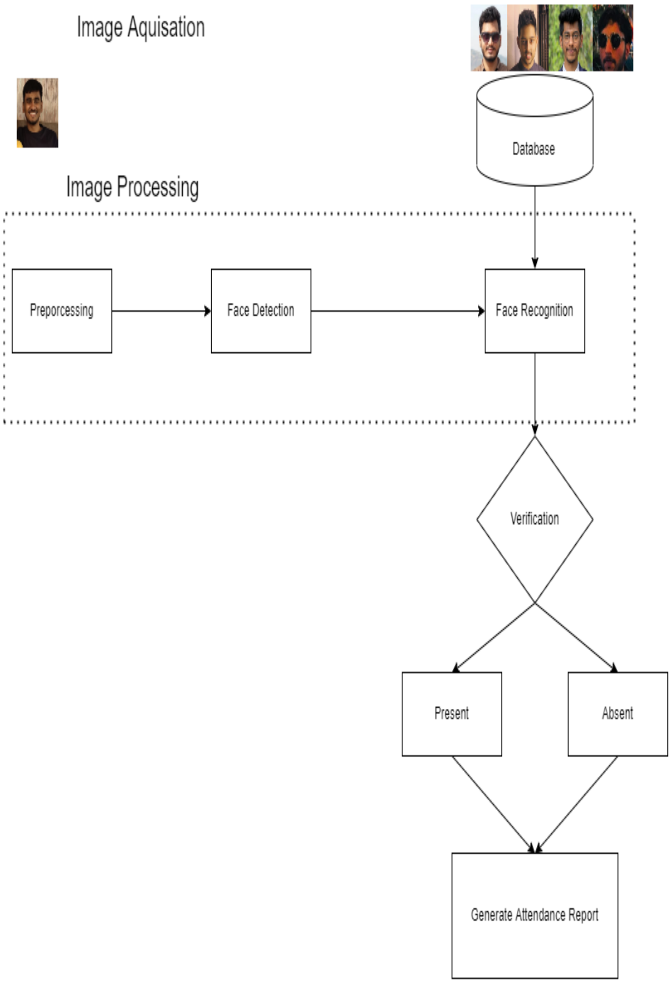


Fig. 6. Conceptual Architecture of the System

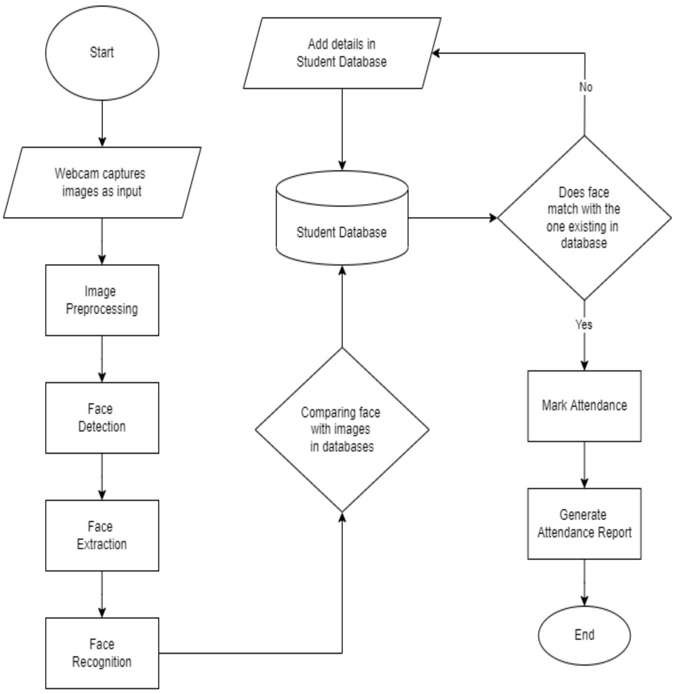


Fig. 7. Modular Diagram of the system

4.2 Face Detection

The face-recognition model [12], which is heavily based on dlib, is used for face detection. The color and size of the slant eyes, the gap between the eyebrows, the distance between the lips and the chin, and other details are noted in this model. When all of these values are added together, a face encoding is created, which is a vector array with 128 values. This model is looped through the dataset in our system to calculate the face encodings of each image. In the next step of face recognition, this face encoding aids in identifying the students. 128-valued face encoding vector array.

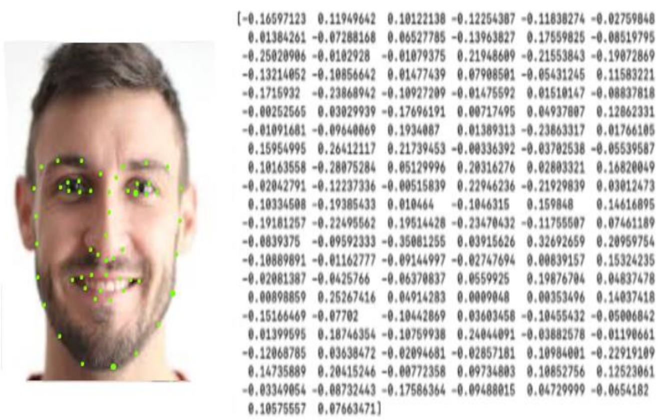


Fig. 9. 128-valued face encoding vector array

4.3 Face Recognition

Real-time image processing and detection are involved in this step. The student's face is detected and the student is recognized using a webcam to record live video of the student. Because the image is captured in real time, image distortion can occur if a student is not fully facing the camera. Face landmark estimation [15] is used to detect the pose of the face, which solves the problem. There are 68 distinct landmarks on every face. The top of the chin, the outside edge of each eye, the inner edge of each eyebrow, and other landmarks can be found.

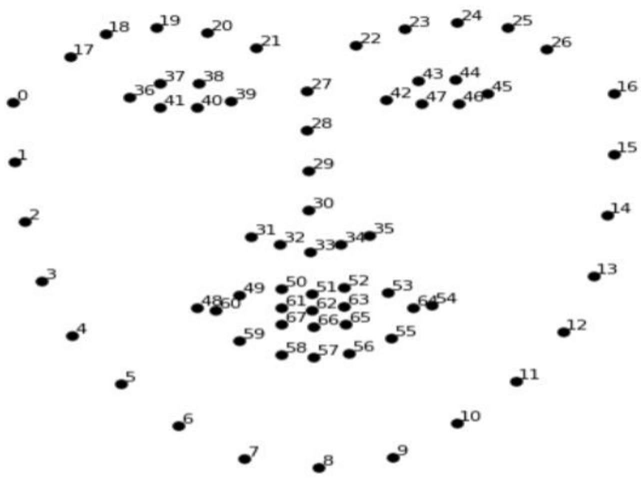
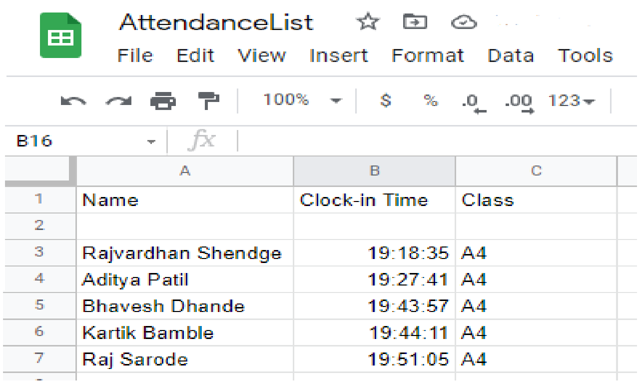


Fig. 10. 68 landmarks present on the face

By rotating, scaling, and shearing the face image, these landmarks can be used to center it. This image can now be used to calculate face encodings, which are then compared to encodings already stored in the database, and the student is identified as such.

4.4 Attendance Generation



	A	B	C
1	Name	Clock-in Time	Class
2			
3	Rajvardhan Shendge	19:18:35	A4
4	Aditya Patil	19:27:41	A4
5	Bhavesh Dhande	19:43:57	A4
6	Kartik Bamble	19:44:11	A4
7	Raj Sarode	19:51:05	A4

Fig. 11. Attendance List as observed using Google Sheets

The recognized faces are then marked present on a CSV file, which can be generated and assessed in a soft copy format on Excel, following the recognition process.

5. IMPLEMENTATION SETUP

Students and faculty can use our PyQT-based GUI to interact with the system. When the students open the app, they will be taken to a screen where they can see the live video feed as well as the date and time when the attendance is taken. To begin their attendance process, the student must first clock in. If the system recognizes the student's face when they clock in, a label with the student's name and the match index will be generated around the face.

Fig. 12. GUI of application capturing real-time video

6. RESULT AND ANALYSIS

According to the implementation setup, the match index is the smallest difference between the face encodings of the student's face in the live video capture and those in the database. Figure 13 depicts the system's implementation as it is being worked on using real-time video capture. The matchindex's confidence threshold is set to 0.6 by default. If the match index value is less than this confidence threshold, the face will be recognized and identified as the same. Figure 14 shows the relationship between the live image capture's confidence score and the face distance (i.e. match index) of the images in the dataset. After that, the student's attendance is recorded in a CSV file. On the attendance sheet, only the names of students whose faces were scanned and recognized will be written. Any app that supports CSV files, such as Excel, Google Sheets, or Numbers, can then be used to evaluate this file.

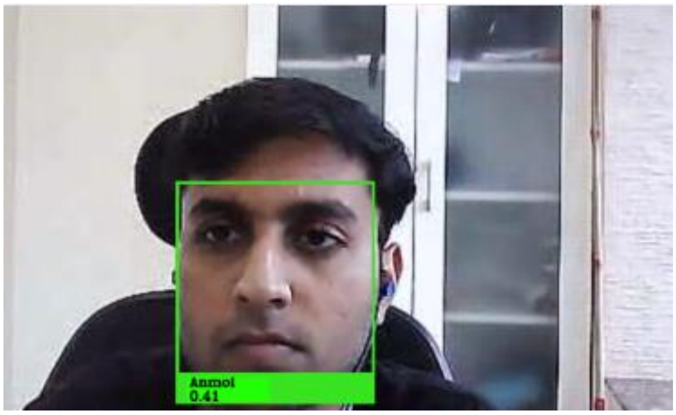


Fig. 13. Live webcam capture of the Students with Identification. The match index is mapped to the identified student's label in our system.

7. CONCLUSIONS

Individual classroom attendance is currently feasible using the system we developed. It can be widely used at the collegiate level with the necessary enhancements and the creation of a proper database containing all of the details of each student in the college or university. This system can be used to manage not only students, but also faculty, staff, and nonstaff members' students. Another development that we want to make sure of is the system's complete automation. To avoid any discrepancies, such as tampering with the devices, the system must currently be supervised. Our goal is to completely automate the process by using a real-time live feed captured by a CCTV camera to mark students' attendance without the need for manual supervision, resulting in legitimate and untampered attendance reports.

8. FUTURE SCOPE

The system that we have developed is currently viable for individual classroom attendance. With the required enhancements and creation of a proper database consisting of all the details of each student in the college or university, it can be widely used at the collegiate level. This system can also be used to manage the students of not only the students but also the faculty members, staff, and non-staff members as well. Another development which we wish to ensure is the complete automation of the system. Currently, the system has to be supervised to avoid any discrepancies such as tampering with the devices. Our goal is to completely automate the process by using a real-time live feed capture using a CCTV camera, which can mark the attendance of students without any manual supervision, thereby producing legitimate and untampered attendance reports.

REFERENCES

[1] T. Fadelelmoula and Almaarefa Colleges, Riyadh, Saudi Arabia, "The impact of class attendance on student performance," Int. Res. J. med. Med. Sci., pp. 47–49, 2018.

[2] A. G. Menezes, J. M. D. da C. Sa, E. Llapa, and C. A. EstombeloMontesco, "Automatic attendance management system based on deep oneshot learning," in 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), 2020, pp. 137–142.

[3] S. Sawhney, K. Kacker, S. Jain, S. N. Singh, and R. Garg, "Realtime smart attendance system using face recognition techniques," in 2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence), 2019, pp. 522–525.

[4] J. W. S. D'Souza, S. Jothi, and A. Chandrasekar, "Automated attendance marking and management system by facial recognition using histogram," in 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), 2019, pp. 66–69.

[5] J. Harikrishnan, A. Sudarsan, A. Sadashiv, and R. A. S. Ajai, "Visionface recognition attendance monitoring system for surveillance using deep learning technology and computer vision," in 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019, pp. 1–5.

[6] T. A. Kiran, N. D. K. Reddy, A. I. Ninan, P. Krishnan, D. J. Aravindhar, and A. Geetha, "PCA based Facial Recognition for Attendance System," in 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 248–252.

[7] M. S. Akbar, P. Sarker, A. T. Mansoor, A. M. Al Ashray, and J. Uddin, "Face recognition and RFID verified attendance system," in 2018 International Conference on Computing, Electronics Communications Engineering (iCCECE), 2018, pp. 168–172.

[8] Smitha, P. S. Hegde, Afshin, and Yenepoya Institute of Technology, "Face Recognition based Attendance Management System," Int. J. Eng. Res. Technol. (Ahmedabad), vol. V9, no. 05, 2020.

[9] S. Saypadith and S. Aramvith, "Real-time multiple face recognition using deep learning on embedded GPU system," in 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018, pp. 1318–1324.

[10] F. Deeba, H. Memon, F. Ali, A. Ahmed, and A. Ghaffar, "LBPH-based enhanced real-time face recognition," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 5, 2019.

[11] https://www.electronicid.eu/en/blog/post/face-recognition/en

[12] https://pypi.org/project/face-recognition/

[13] https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

[14] https://machinelearningmastery.com/one-shot-learning-with-siamese-networks-contrastive-and-triplet-loss-for-face-recognition/

[15] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1867–1874.

[16] https://www.softwaresuggest.com/blog/manual-vs-automated-attendance-system-pros-cons/

[17] https://www.edusys.co/blog/rfid-attendance-system

[18] S. K. Baharin, Z. Zulkifli, and S. B. Ahmad, "Student absenteeism monitoring system using Bluetooth smart location-based technique," in 2020 International Conference on Computational Intelligence (ICCI), 2020, pp. 109–114.

[19] 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), 2005, vol. 1, pp. 886–893 vol. 1.

[20] https://maelfabien.github.io/tutorials/face-detection/1-theory-1

[21] https://heartbeat.comet.ml/one-shot-learning-part-1-2-definitions-and-fundamental-techniques-1df944e5836a