2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) | 978-1-6654-7436-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICAC3N56670.2022.10074166

# Smart Face Recognition Based Attendance System Using ML Algorithm

Rohan Habu
*School of Electronics & Communication Engineering*
*Dr. Vishwanath Karad MIT World Peace University,* Pune, Maharashtra
haburohan@gmail.com

Sumitra Motade
*School of Electronics & Communication Engineering*
*Dr. Vishwanath Karad MIT World Peace University,* Pune, Maharashtra
sumitra.motade@mitwpu.edu.in

Shweta Kukade
*School of Electronics & Communication Engineering*
*Dr. Vishwanath Karad MIT World Peace University,* Pune, Maharashtra
shweta.kukade@mitwpu.edu.in

Kishanprasad Gunale
*School of Electronics & Communication Engineering*
*Dr. Vishwanath Karad MIT World Peace University,*
Pune, Maharashtra
kishanprasad.gunale@mitwpu.edu.in

Arunkumar Nair
*Tech Head*
*Canspirit Artificial Intelligence*
Pune, Maharashtra
arunkumar.nair@canspirit.ai

***Abstract*—**
**The system's goal is to make the attendance marking process quick and easy because the teacher's mundane job in class is ti me consuming in monitoring the students while marking atten dance and ensuring that no proxy attendance is marked.**
**To solve the problem efficiently, the system employs a machin e approach that makes use of Python's OpenCV library. The Haar cascade algorithm and the LBPH algorithm are used for face detection and recognition. The system is designed to be s ufficiently accurate in comparison to other systems.**

***Keywords*— *OpenCV, Python, LBPH, Haar cascade, image processing, face recognition, face detection.***

## I. INTRODUCTION

Normally the schoolteacher has to mark attendance of all students present in the class either before the start of class or after the class or for both times. The teacher either calls out each student's name and confirms attendance or instructs the students to note their attendance on paper themselves. And later the teacher has to do the extra work again of marking the attendance in an excel sheet or csv file or any attendance marking system. With this traditional system, as teacher has to mark it manually, there may be the chances of proxy attendance marking by student, wrong marking of attendance and wastage of time during the class. To avoid these cases, many researchers have introduced identification systems like RFID based System, Biometric based Recognition System like Iris, Fingerprint, Palm print. These systems, however, have limitations in terms of range and contact-based systems.

RFID technology is used to implement the attendance system [1]. The system's output is satisfactory, but it has some limitations. The RFID reader must be of high quality. The RFID tags must be carried by the students throughout the campus in order to record attendance for the lecture. This can be inconvenient at times, and if the student does not have the RFID tag, his or her attendance cannot be tracked even if the student is present in class. This shows that the system does not completely solve the problem. In this case, there is a high likelihood that students will mark proxy attendance by exchanging the tags. Furthermore, the system's cost rises due to the high cost of accurate and long-range RFID readers.

The fingerprints is used to track students' attendance [2]. The solution obtained is good, but during the covid era, these systems were not at all promoted, so even though these systems were present in many institutions and organizations, attendance was manually recorded. So, the point is that the system has no limitations, but it cannot be used in difficult situations such as covid.

The mention of these systems and their limitations sparks the idea of developing a low-cost hands-free attendance system. As a result, face recognition technology is best suited to marking the student's attendance without requiring any contact with any device. Python is used in the system for both GUI and backend development. In short, Python is used to create both the system's front end and backend.

## II. LITERATURE SURVEY

As mentioned, the system's core is face recognition, which provides the most value to the system. A face recognition-based attendance system is proposed [3], however it had an accuracy of 77 percent and a false positive rate of 28 percent, which is considered a challenge to solve, and this system has an accuracy of 90 percent and a false positive rate of 18 percent. This is accomplished by training the machine learning model while keeping the

dataset size as modest as possible in order to avoid model underfitting and overfitting. This can be accomplished by maintaining a constant ratio between all of the students' images. Calculating the total number of correct and incorrect predictions made by the model, as well as the total number of observations, provides a foundation for determining the trained model's accuracy.

Face detection should be thorough before moving on to face recognition. Because faces must be present in the image in order for face recognition to work. And this is possible, thanks to the machine learning algorithm of Haar cascades developed by [4] in 2001. This is the most efficient algorithm to date, capable of detecting most faces in an image in a short amount of time and with an accuracy of 96.24 percent. Although the model may make false detections, this can be avoided if an image is captured in a controlled environment. There are also various face detection APIs available, but they require the image to be sent over the internet to their database, where faces are extracted from the image and then sent as a JSON response. However, reliable internet connectivity is required for this process to transfer the data. However, the algorithm proposed by [4] can be run locally without the need for internet connectivity and has acceptable accuracy.

As mentioned in [4], the concept of integral image is used, which allows it to perform calculations for a very high-resolution image in a very short amount of time. The sum of all pixels above that pixel and all pixels preceding that pixel in that row is the integral image pixel value. Haar features are calculated using the relative intensities from the integral image. Figure 1 depicts an explanation of the integral image. The Haar features are classified as edge, rectangle, and line as mentioned in Figure 2.

When it comes to face recognition, the system employs the Linear Binary Pattern Histogram (LBPH) algorithm. The algorithm extracts binary patterns from grayscale face images, stores the image's histogram, and uses those histograms to perform real-time face recognition. The LBPH algorithm [5] is used efficiently by training a Convolutional Neural Network (CNN). This system also uses the LBPH algorithm efficiently by making that the ratio of training images is equal, avoiding cases of underfitting and overfitting of the machine learning model. The system maintains the image ratio, which improves the model's accuracy. The LBPH algorithm is useful for running locally and does not require high performance hardware.

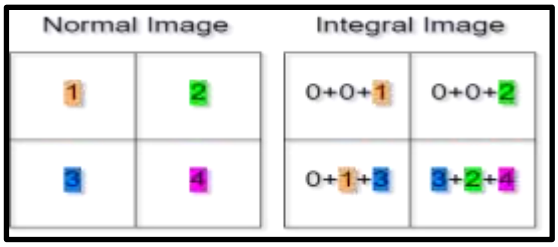


Fig1: Integral Image

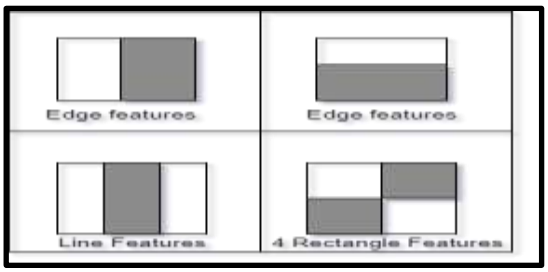


Fig2: Haar Features

A face recognition system [6] is proposed that detects Haar features using opencv and recog nises faces using Histogram of Oriented Gradients (HOG). The HOG algorithm works best on a machine with high co mputational power, but on a standard machine, latency is pr oduced, which may prevent the application from working in real time. To deploy using only HOG, cloud services are re quired, which require a reliable and stable internet connecti on to function. Because of this limitation, that approach is n ot universally accepted.

A system is proposed in [7] that uses Convolutional Neural Networks (CNN) to feed an image as a feature vector to the trained network and then co mputes the recognition result.However, training CNN takes longer than training the LBPH model. As a result, as the dat abase size grows, so does the training time, making it slowe r to train the model and use it incrementally.

Face recognition is performed using a deep learning approach Face recognition is performed using a deep learning approach [8]. Deep learning does not require a feature extraction phase be cause the model extracts the features and returns the result after the fully connected layer and pooling layers are attach ed to the model's convolutional layers. This deep learning model necessitates a lot of computational power, and the a mount of memory required grows exponentially as the data set size grows. It must be used only with cloud services bec ause local machines cannot run these models at all. As a result, deep learning cannot be applied.

The HOG approach [9] is used to perform face recognition, which, as previously state d, requires more computational power. Because LBPH perf orms well on low computational power devices, the HOG a pproach cannot be used.

Other approaches to face recognition exist. The first method is to train the model using Google's TensorFlow Python library, which employs the transfer learning method. The computation power required for this is higher and using TensorFlow on a computer without a GPU is slower. As a result, there will be a lag between face recognition and the user experience will suffer. TensorFlow has a lightweight library called tflite that can run on a microcontroller and perform image analysis. However, in order to perform accurate face recognition, the image must have sufficient resolution to detect faces in the image and perform face recognition over it. This is difficult and expensive to achieve and thus cannot be used.

Another method is to use Python's face recognition library, which makes use of the face encodings concept.

ISBN: 978-1-6654-7436-8/22/$31.00 ©2022 IEEE 

However, any of the processes cannot be changed that generate the face encodings, so tuning the trained model is not possible. As a result, the accuracy of cannot be guaranteed because it uses a single image to generate the encoding and has been observed to work practically in the same lighting conditions as that of the input image. So, if the lighting conditions in the environment are controlled, this is useful; however, in the case of a classroom, the conditions are controlled but have a slight variance, and thus implementing face recognition using this library may fail.

After considering all of these methods of implementing face recognition, the machine learning method is chosen because it efficiently trains the machine learning model based on the LBPH algorithm and performs computations locally without any dependency. As a result, the system is lag-free and accurate in producing satisfactory results.

## III. DESCRIPTION

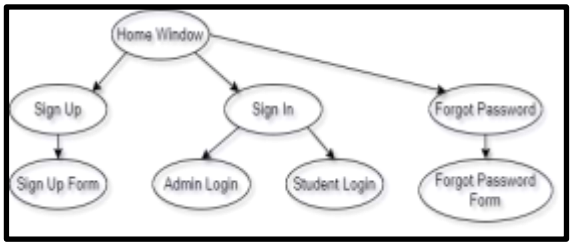


Fig3: Frontend Window Links

The system employs a Graphical User Interface (GUI) created with Python's Tkinter library. Tkinter is a flask-like framework used to host applications on Windows, Linux, and Mac OS. It also provides native support for some components, which can be customized. The program's logic is written entirely in Python. During the operation of the GUI, Python accesses all data at the backend. To make things easier to understand, the project is divided into frontend and backend components.

The frontend is made up of the following windows: the home window, the sign-up window, the sign in window, the forgot password window, the student window, and the admin window. All of these windows are made with Tkinter. The GUI buttons execute various backend functions and keep the application running. To gain control of the application, basic functionalities such as new user registration with model training, deletion of the user from the system, forgot password utility, and admin login facility are implemented. The main highlight of the software is its email sending capability. After the administrator logs into the system and selects a specific attendance sheet, the software retrieves the names of the present students in the class and sends emails to the students and their parents, which were collected during registration. As a result, students can receive notification of their attendance status with a single click. Figure 3 depicts all of the wireframe links for the frontend window.

The frontend windows are all children of the main home window. As shown in Figure 3, the windows are descended from the home window. As a result, when the application is launched, the home window is loaded and displayed first. The next window loads based on the user's selection. The frontend has been designed in this manner.

To perform the tasks triggered by the frontend GUI, the backend of the application is implemented using python libraries such as NumPy, pandas, smtp server, imutils, OpenCV, and others. NumPy and pandas are used to analyse data and store it in a structured format. Smtp server library is used to connect to the smtp server and send emails to students and parents. Imutils is used to access the system camera, capture the video stream from the camera sensor, and deliver it to the application. OpenCV is used for image pre-processing and face detection, as well as face recognition on detected faces. After capturing the image from the video stream, it is converted to grayscale to drastically reduce the number of calculations required while losing one dimension, namely the colour of the image. However, the face detection and recognition models are only trained for grayscale images, so feeding colour images into the application will fail. As a result, the algorithms use fewer computer resources while producing excellent results.

The home window has four buttons: signup, sign in, forgot password, and quit. The signup window contains the signup form, which accepts the candidate's basic information, as well as a button that says Sign Up & Take Images, which when clicked, opens the webcam, and displays the webcam feed on the display. The image is not captured until the candidate presses the c for capture key on the keyboard. The candidate can also leave the session by pressing the q for quit key on the keyboard. If the candidate fails to capture an image, a pop-up message appears, no model is trained, and the process is aborted. The candidate can capture images in the range of 1 to 10 and if the user explicitly exits the camera in the middle, it is checked whether the user is already present in the system or not by verifying the details entered by the user along with the details stored in the studentdetails.csv file.

If present, the model is trained for the user's most recent images, and the candidate's other information is updated. If the user does not exist in the system, the model is trained, and a welcome email with the login credentials attached is sent to the user. Then, depending on the condition, a pop-up message is displayed for the program's action. The sign in window contains a sign in form that requests the user's login credentials. When the login button is clicked, the details entered in the form are submitted to the code and verified. The admin and student sign in screens are identical. If the entered credentials are of admin, the admin screen is displayed, which contains a variety of options and full control over the system. If the credentials entered are those of a student, the student screen is displayed, with a single option for marking attendance. If a student forgets their login credentials, they can use the forgot password utility. The credentials can be retrieved again by entering basic information for student verification, and the credentials will be emailed to the student immediately.

The administrator can perform the following

ISBN: 978-1-6654-7436-8/22/$31.00 ©2022 IEEE 

actions in the application: Send email, Train model, View attendance of any time, Mark attendance, View student details, view training images, full reset, delete model, delete student data, and delete all attendance data.

## IV. METHODOLOGY

1. Accessing the device's webcam. The device's webcam is accessed via the Python imutils (image-utilities) library, which contains basic image processing functions (capture, resize, rotate, etc.).
2. Creating the application's home screen. Tkinter renders the home screen, which includes sign up, sign in, and forgot password options.
3. If the sign-up option is selected, render the sign-up screen and, after entering all of the required information and clicking the sign-up button, display the webcam feed on the screen and wait until the user does not press the c key to capture the image. If no image is clicked and the user exits the process, the process is discarded, and the model is not trained. If the user clicks some images and exits the process, train the model and check whether the user already exists in the system, and if so, display a retraining and details update pop-up message. If the user does not exist in the system, send the email with the login credentials attached to the user.
4. If logged in as an administrator, then the admin can use the admin screen to perform several actions.
5. If logged in as a student, display the student screen to mark the attendance.
6. If the forgot password option is selected, display the forgot password screen and, after entering the details, email the user's credentials if the user exists on the system.

| Roll Id | Student Name | Student Email | Parent Email |
|---|---|---|---|
| 1 | Manoj Joshi | manojj@gmail.com | manojp@gmail.com |
| 2 | Ramesh Sai | rameshs@gmail.com | rameshsp@gmail.com |
| 3 | Rahul Gupta | rahulg@gmail.com | rahulgp@gmail.com |
| 4 | Abhi Vora | abhiv@gmail.com | abhivp@gmail.com |

Table 1: Student Details Format

| Id | Name | Date | Time | Student Email | Parent Email |
|---|---|---|---|---|---|
| 2 | Ramesh Sai | 23-07-2022 | 2:45 PM | rameshs@gmail.com | rameshsp@gmail.com |
| 3 | Rahul Gupta | 23-07-2022 | 2:45 PM | rahulg@gmail.com | rahulgp@gmail.com |
| 4 | Abhi Vora | 23-07-2022 | 2:46 PM | abhiv@gmail.com | abhivp@gmail.com |
| 1 | Manoj Joshi | 23-07-2022 | 2:46 PM | manojj@gmail.com | manojp@gmail.com |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | m |

Table 2: Student Attendance Format

## V. FLOWCHART

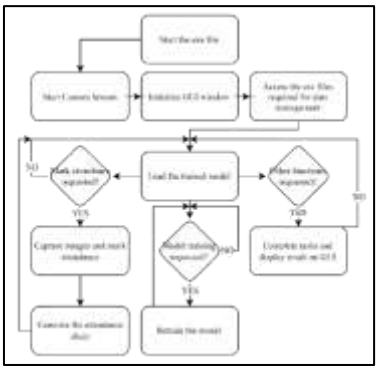

Fig4: Backend Flowchart

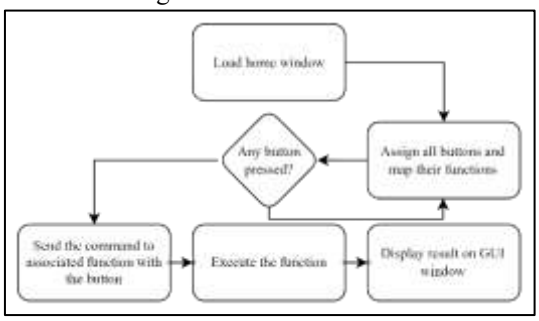


Fig5: Frontend Flowchart

## VI. RESULTS

Figure 6 depicts the application's home window. It allows the user to sign in, sign up, use the forget password utility, and exit the application. The user can log in using either admin or student credentials. If the user is new to the system, he or she must sign up before using it. The user's credentials are emailed to the email address provided by the user during the sign-up process. If the user forgets their password, they can use the forgot password utility to recover their credentials.

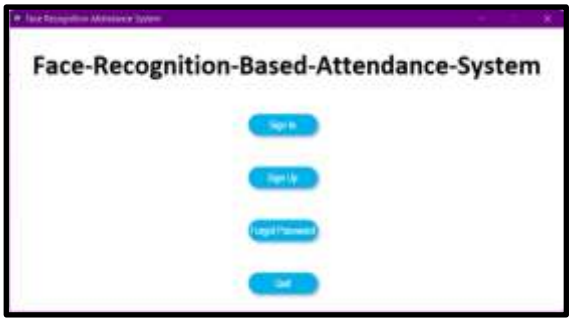


Fig6: Application Home Window

Figure 7 depicts the Sign In window, which accepts the user's username and password to login. When a user clicks on login, the data in the system is compared to the information entered. If the entered credentials are admin, the admin utility screen appears; otherwise, the student utility screen appears; otherwise, a pop-up message stating incorrect credentials appears.


Fig7: Sign in Window

The user can register for the system by using the Sign-up window, as shown in Figure 8. The form accepts information such as name, ID, student email, and parent email. These specifics may be expanded in the future based on the needs of the organisation.

Following the collection of these details, the details are checked to see if they are relevant, such as no integers in the name section, no names in the roll id section, and the domain name in the email section. If all of the information is entered correctly, it is saved on the system and the camera is opened for image capture. The user can take images ranging from 1 to 10. Taking ten images is a better option for improving model training and recognition accuracy. Following that, the model is trained, and if the training is successful, the training completed message is displayed. And if there are any complications, they are only displayed in the message. If the user is new, an email is sent to the student with the student's credentials for logging into the system to mark attendance. If the images are taken by an existing user, the model is retrained on the new images and no email is sent to the user.

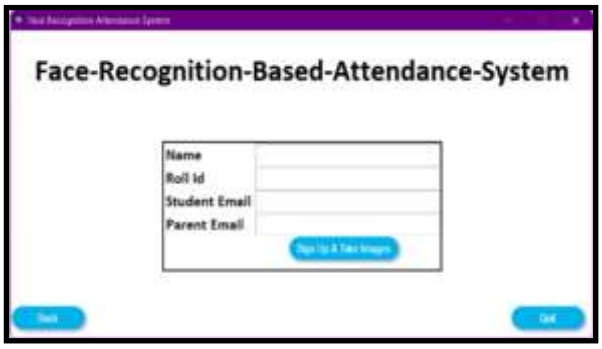


Fig8: Sign Up Window

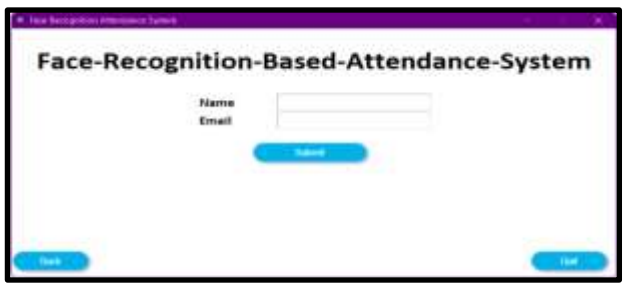


Fig9: Forgot Password Window

Figure 9 depicts the Forgot password window, which accepts the user's name and email address to confirm the user's existence in the system. If the user exists in the system, the credentials are emailed to the user's registered email address; if the user does not exist in the system, a prompt appears on the screen.

Figure 10 depicts the admin task window, which contains numerous administrative utilities. The administrator can automatically send emails to students and their parents. The model can be retrained. The most recent attendance can be viewed with a single click. Any previous attendance can be viewed. If a student encounters a problem during the sign-in or sign-up process, the administrator can resolve it by using the create folders button, which creates the necessary files for the system to function. The student's attendance can be tracked. The .csv file contains information of all students. All reset features, such as full reset, delete model, delete all student data, and delete all attendance data, are also included.

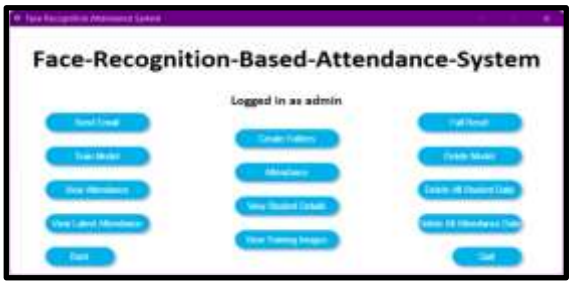


Fig10: Admin Window

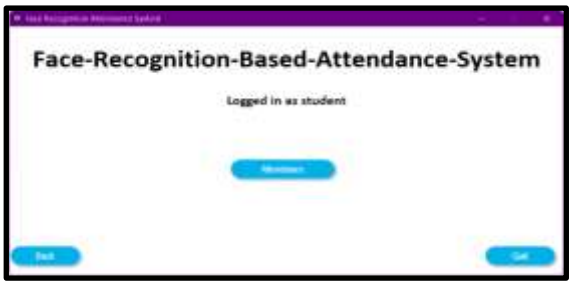


Fig11: Student Window

After logging in with the necessary credentials, the student can mark his or her attendance using the window depicted in Figure 11. When a student clicks the attendance button, the camera stream is accessed, and the face detection and recognition algorithm is activated. If the face is detected, a square is drawn on the face. If the face is recognised, the attendance is marked, and the image displays the student's roll number and name, indicating that the attendance was successfully marked. After marking attendance, the student can stop the camera stream by pressing the q key.

Fig12: Image Capture Window

Figure 12 depicts the images captured by the system during the student registration or sign-up process to

ISBN: 978-1-6654-7436-8/22/$31.00 ©2022 IEEE 

train the machine learning model. These images are pre-processed, and only the face image surrounded by the rectangle on the main image is fed into the model. After taking the images, the model is trained, and the user is prompted about the model's training status.

Figure 13 depicts the attendance marking window, which includes the real-time camera stream as well as the bounding boxes for any detected faces in the image. If the face is recognised, the student's attendance is marked in the attendance sheet, and the roll id and name of the recognised student are displayed in real time on the same window; otherwise, the attendance is considered unknown, and the name is displayed as unknown.


Fig13: Attendance Marking Window

Figures 14, 15 show only the attendance marking windows from a variety of angles. As this is a ML approach rather than a computer vision approach, the user does not need to be at the same angle to be recognised by the system. The system can achieve up to 35 degrees of angle in all four directions (up, down, left, and right). Going outside the angle will produce false results because the model will not detect the face and thus no face recognition can be used.


Fig14: Horizontal plane movement of face


Fig15: Vertical plane movement of face

## VII. COMPARISON OF VARIOUS FACE RECOGNITION METHODS

| Parameter | LBPH | Tensorflow | Deep Learning |
|---|---|---|---|
| Accuracy | 85% - 90% | >90% | 80% - 90% |
| Application | Controlled environments | Dynamic environments | Controlled environments |
| CPU Consumption | Less | High | Very high |
| Dependencies required | No | Yes | Yes |
| False positives/negatives rate | 20% - 25% | 10% - 15% | 25% -30% |
| Latency | 10-20ms | 100-110ms | 90-100ms |
| Reliability | Moderate | High | Moderate |
| Training time | Less than 5 seconds for data size > 100 | More than 10 seconds for data size > 100 | More than 15 seconds for data size>100 |

## REFERENCES

[1] Lim, T. S., S. C. Sim, and M. M. Mansor. "RFID based attendance system." 2009 IEEE Symposium on Industrial Electronics & Applications. Vol. 2. IEEE, 2009.

[2] Akinduyite, C. O., et al. "Fingerprint-based attendance management system." Journal of Computer Sciences and Applications 1.5 (2013): 100-105.

[3] Chinimilli, Bharath Tej, et al. "Face recognition based attendance system using Haar cascade and local binary pattern histogram algorithm." 2020 4th international conference on trends in electronics and informatics (ICOEI) (48184). ("2020 4th International Conference on Trends in Electronics and ...") IEEE, 2020.

[4] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." ("Rapid Object Detection using a Boosted Cascade of Simple Features") "Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition." ("Proceedings of the 2001 IEEE Computer Society Conference on Computer ...") CVPR 2001. Vol. 1. Ieee, 2001.

[5] Anand, Akshit, Vikrant Jha, and Lavanya Sharma. "An improved local binary patterns histograms techniques for face recognition for real time application." International Journal of Recent Technology and Engineering 8.2S7 (2019): 524-529.

[6] Hl, Dhanush Gowda, et al. "Face recognition based attendance system." International Journal of Engineering Research & Technology (IJERT) 9.6 (2020).

[7] Damale, Radhika C., and Bazeshree V. Pathak. "Face recognition based attendance system using machine learning algorithms." 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). ("2018 Second International Conference on Intelligent Computing and ...") IEEE, 2018.

[8] Sutabri, Tata, Ade Kurniawan Pamungkur, and Raymond Erz Saragih. "Automatic attendance system for university student using face recognition based on deep learning." ("Automatic Attendance System for University Student Using Face ...") International Journal of Machine Learning and Computing 9.5 (2019): 668-674.

[9] Madhu, Shrija, et al. "Face recognition based attendance system using machine learning." Int J Mang, Tech and Engr 9 (2019).