International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

IOATS: an Intelligent Online Attendance Tracking System based on Facial Recognition and Edge Computing

Manoranjan Parhi[1], Abhinandan Roul[2], Bravish Ghosh[3], Abhilash Pati[4]

Submitted: 10/03/2022 Accepted : 02/05/2022

Abstract: Since the Coronavirus (COVID19) pandemic, all activities have been held digitally, necessitating the surveillance of guests' real-time attendance. Previously, online attendance included getting the list of attendees, which was inconvenient because many people chose to keep silent or leave the meeting completely. As a result, a technique for collecting attendance using facial recognition that can correctly identify participants who remain online for the duration of the lecture is required. The goal of this work is to develop a system named IOSTS, an intelligent online attendance tracking system, that can track attendance while using minimum bandwidth and maintaining user privacy. This proposed work is based on the concepts of facial recognition and edge computing. The entire utility will be run on the client's PC. From random experiments, it is observed that achieving an accuracy of 98 % in facial recognition. This new approach is a foolproof method of tracking attendance and increasing digital transparency.

Keywords: Edge Computing, Facial Recognition, Attendance Tracking, COVID19, Video Conferencing Platform

1. Introduction

The world has become silent for some months due to the pandemic situation created by COVID19, which stands for Corona Virus Disease 2019 [1]. Educational systems are widely hampered due to COVID19. All events have been planned online since the COVID19 epidemic. Video conferencing software is used to hold educational sessions, corporate meetings, and self-development seminars. The new classes are Google Meet, WebEx, and Zoom. Students from all across the world join forces to learn together. The attendance system is one part of teaching that is difficult to control. A new strategy is being developed to maintain academic integrity and maintain student engagement in classrooms at the same level as previously.

1.1. Edge Computing

Edge computing refers to the process in which all the possible computations take place at the client-side, thus limiting the data transfer, and computational load on the server and reducing bandwidth consumption Olaniyan et al [2]. This process utilizes the users' device to compute collected data. It values privacy by not transferring critical private information over the network. The proposed tool as shown in Figure 1 uses edge computing to detect and recognize faces on users' devices.

[1] *Faculty of Engineering and Technology, ITER, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India*
ORCID ID: 0000-0002-1625-6022
[2] *Faculty of Engineering and Technology, ITER, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India*
ORCID ID: 0000-0002-0132-8919
[3] *Faculty of Engineering and Technology, ITER, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India*
ORCID ID: 0000-0003-4919-3264
[4] *Faculty of Engineering and Technology, ITER, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India*
ORCID ID: 0000-0002-3418-4202
* *Corresponding Author Email: er.abhilash.pati@gmail.com*

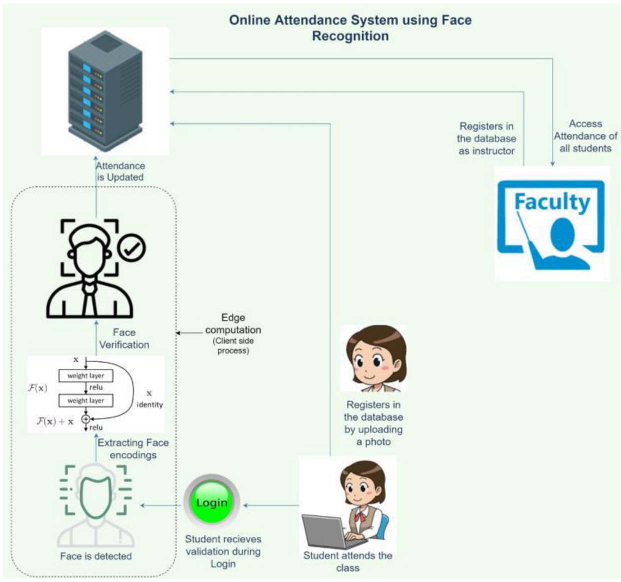


Fig. 1. Proposed Attendance Tracking System

The benefits obtained by using edge computing are as follows:

- Low network utilization: Images are not transmitted to the cloud;
- Data Privacy: No photos are transmitted over the internet;
- Faster processing: No time is spent on sending the image to the server and awaiting the inference;
- Operational efficiency: Servers with low computational capacity can handle the tasks;
- Reduced power consumption: Since the servers do not need to process expensive operations, they consume less power.

1.2. Facial Recognition

Facial recognition is the technique of verifying a person's identity by comparing it with a known image of the person. The use of face recognition is planned to make the tool accurate and reliable. This tool can help the administration to monitor the students' presence in a class comprehensively and with ease. The whole process revolves around continuous capture of students' images using OpenCV, localization of face images from the captured photo and extracting the face encodings, and running the verification process [3]. The video capture, image extraction, and facial recognition will all happen at the client end i.e., on the student's laptop or PC.

1.3. Research Gap and Questions

Earlier works focused on the recognition of students in physical classes with the use of IoT devices. The information, i.e., the images of the students were also being stored in the cloud thereby posing a threat to user privacy. None of the studied research works focused on attendance capture in online classes and meetings.

There exist some **research questions (RQs)**:

- To what extent this proposed work can accurately detect the attendees?
- What is the effectiveness of using this system in various educational organizations?
- Is the tool limited by internet bandwidth constraints?

1.4. Motivation and Objective

Generally, many issues are raised for effective attendance during an online class, that should be solved. We tried to address the issue by using a unique online attendance system with a focus on edge computing to improve user privacy.

The **key objectives** for developing this tool are:

- To achieve a high-accuracy, fail-proof, and highly resilient method of attendance labeling with auto verification of attendees' participation. It would necessitate little or no human intervention in marking the participation of a specified list of students;
- To develop an intelligent system that is capable of detecting and recognizing faces in an online environment, tracking students' attention and mood. It will help the host to identify whether the content being delivered is being received by the audience;
- To create a user-friendly way to register students and faculty members, and students' face images directly through a web component;
- By classifying images at the client end and then submitting the final attendance (Present/Absent) to the central archive, network utilization and application processing are minimized;
- Be able to display to the student/user the number of active and failed facial recognition attempts.

1.5. Contribution

The **key contributions** of the paper are described as follows:

- Recognize the face from Live Video / Recorded Video;
- Track the attendance of all students for the entire duration of the class using face recognition and edge computing;
- Update the attendance in a central database server;
- Highlight Images on RED if that student is absent from previous classes;
- Highlight Images on GREEN if that student attendance is regular;
- Dashboard of overall attendance for the entire class;
- This tool can be deployable as a web component;

1.6. Organization of the paper

The rest of this paper is organized as follows. In Section 2, the significant state-of-the-art approaches relating to attendance tracking systems using face recognition techniques are discussed. The implementation of the proposed attendance capturing tool is explained with the necessary graphical user interface in Section 3. Section 4 describes the experimental setup and implementation. Finally, Section 5 describes the conclusion and future work.

2. Existing Works

Balcoh et al [4] proposed a technique where captured images are converted into a grayscale image and histogram normalization takes place for contrast enhancement and median filtering is used for noise removal. The accuracy on a clear normal face is found to be 95%. Chintalapati et al. [5] utilize the Viola-Jones algorithm for face detection, AdaBoost learning algorithm as the classifier, PCA/LDA/LBPH for feature extraction, and Distance Classifier/Bayesian/SVM for classification. The accuracy for PCA with the distance classifier technique has the highest accuracy for static images at 93%. Nithya [6] proposed a PCA algorithm for face recognition and IoT for attendance systems with notification systems. Surekha et al [7] proposed face detection using the Viola-Jones algorithm, face recognition using the Multi-Keypoint Descriptor method, and the Sparse Representation based Classification (MKD-SRC) method of partial face recognition. The system produces a 60% accuracy in a regular environment, with a substantial computation time when a large number of images are considered. Arsenovic et al [8] proposed a CNN for face detection and recognition, as they're effective for large datasets, and provide accurate mean results with smaller error margins. The experiment was conducted with people in five different positions, and the model was trained using augmented model images. The overall accuracy was 95.02%. Rekha et al. [9] proposed a technique where the PCA (Eigenface database) algorithm is used for face recognition, with preprocessing steps such as image cropping, resizing, and image enhancement to produce the eigenface database, data compression of face dataset for identification. Suresh et al [10] proposed a technique that consists of face recognition by Eigenfaces with a limitation of 1 person at a time, with data being stored locally on a MicroSD card. Kakarla et al [11] proposed a technique where face recognition is implemented using 20 layered CNN model architecture which received an accuracy of 99.86% with a loss of 0.0057%. Patil et al [12] proposed face detection using Viola-Jones, HAAR Cascade algorithm, and face recognition is done by LDA, KNN, and SVM. Excel sheets were incorporated to store the student attendance. Sharanya et al. [13] proposed face detection using Haar-Cascades which is included in the OpenCV library (a computer vision library). CNN is used for the face recognition part, which is used inside the Keras library (an open-source software library that facilitates Python interface for artificial neural networks) as a sequential model for image processing, face recognition, and liveness detection. Bussa et al [14] proposed to use the OpenCV library and implement the LBPH algorithm for face recognition with a Tkinter interface, with excel sheet attendance updating. The summary of existing works considered in this research is provided in Table 1.

Table 1: A Summary of the considered works

Work	Methodology	Findings	Pros	Cons
Balcoh et al. [4]	RGB is converted to grayscale, Histogram normalization for contrast enhancement, HAAR cascade for face detection, and Eigenface for face verification	It was found that face detection accuracy with veil -was 40%, unveil-95%, and beard -75%. The face recognition with veil 2%, unveil- 95%, beard- 63%.	No usage of any specialized hardware.	Inaccurate to recognize people in veils.
Chintalapati et al. [5]	Use of Viola-Jones algorithm for face detection, AdaBoost learning algorithm as the classifier, PCA/LDA for feature extraction, SVM/Bayesian for classification.	For PCA+distance classifier false positive rate=55%, Recognition rate of static images=93%, Recognition rate of Real time video= 61%. False positive rate of PCA+Distance classifier is 55%, LDA+Distance classifier is 53%, PCA+SVM is 51%, PCA+Bayes is 52%, LBPH+Distance classifier is 25%. Recognition rate (static images) for PCA+Distance classifier is 93%, LDA+Distance classifier 91%, PCA+SVM is 95%, PCA+Bayes is 94%, LBPH+Distance classifier is 95%.	Can handle spoofing threats.	Only detect faces for up to 30° angle variations
Nithya [6]	PCA algorithm for Face recognition, SMTP (Simple Mail Transfer Protocol) is used to send mail over the internet using IoT	Attendance of the student is sent to their parents.	Integration of IoT Attendance of the student is sent through emails to their parents and teachers	Works only in the classroom with completely visible faces
Surekha et al [7]	Viola-Jones algorithm and MKD-SRC method of partial face recognition	60% accuracy in an uncontrolled environment, and 100% accuracy under a controlled environment	Capable of recognizing partially visible faces, with limited view.	The efficiency of the system is reduced in the case of partially visible faces
Arsenovic et al. [8]	CNN cascade, OpenCV and Dlib Face recognition API and integration with web application	The system was tested in an IT company with five volunteers. Using augmented images in the dataset the algorithm could adopt DNN for partially noised data. 95.02% accuracy in testing	DNN for partially noised data Deep Learning face recognition using Dlib Python script adds random accessories to face images: mustaches, glasses, etc. and creates new images for training the dataset.	CNN's methods on smaller datasets are challenging as CNNs achieve better results for larger datasets.
Rekha et al. [9]	Eigen Face and PCA algorithm with MATLAB GUI.	Image Cropping, Image Resize, resized images are converted from RGB to Gray Level, followed by Image enhancement Attendance is updated in the Microsoft Excel sheet	NXN resolution Image enhancement Normalize the vector to find unique features	To generate an Eigenface database, a large number of digitized images are normalized. Eigen's faces are used for data compression of faces for the identification process. It is generated for training set images and test images
Suresh et al. [10]	Eigenfaces, Micro SD is used as a storage database to compensate for the volume of the face data.	Attendance is automatically recorded in the excel sheet.	Background subtraction and face cropping. OpenCV is used.	The facial recognition process can only be done for 1 person at a time.
Kakarla et al. [11]	A novel CNN architecture with Smart Attendance Management System (SAMS)	accuracy and loss of the CNN model are recorded as 99.86% and 0.0057 respectively. (practical)	20 layered CNN architecture Data collection and data augmentation for developing the CNN Data Augmentation to generate new samples by manipulating the existing data	A large training dataset is required for better accuracy
Patil et al [12]	Viola-Jones and HAAR Cascade algorithm. For image pre-processing, histogram equalization is used to enhance input image quality. Excel sheet attendance marking.	Precision. Recall and Accuracy in KNN is 96%, 97%, 97% and in SVM is 93%, 95%, 95%.	Face Recognition done using LDA along with KNN and SVM Time-efficient	Illumination and aging effect can make problems with recognition
Sharanya et al [13]	Haar Cascades and CNN. OpenCV used integral image concepts to compute the features.	Single image-based face liveness detection.	AdaBoost Learning Administrator role and student role	Lacks algorithm Tuning Tkinter GUI used for UI
Bussa et al. [14]	LBPH algorithm	Uses an algorithm with the least noise interference.	Open CV library is used to implement the LBPH algorithm	Tkinter GUI Microsoft excel for attendance report

3. Proposed Work: IOATS

This section describes the main workflow of the proposed attendance tracking system. The operational flow of the complete process, as depicted in Figure 2, is discussed in a step-by-step approach in Algorithm1. Based on this mechanism, an intelligent online attendance capturing tool has been designed which includes the following seven modules:

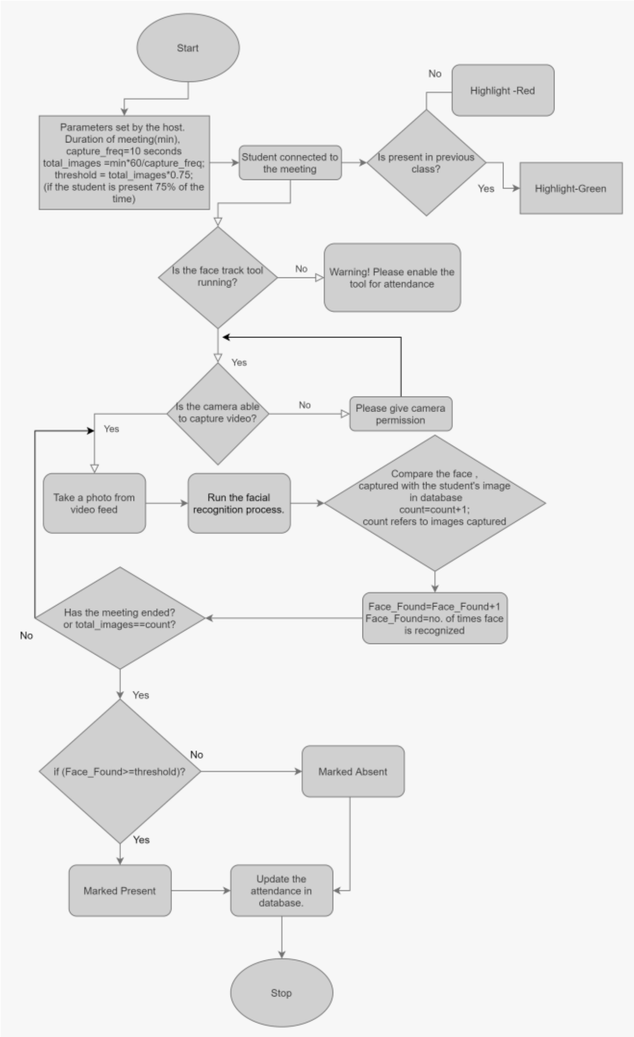


Fig. 2. Working Procedure of Proposed Model

A. Student Registration

The student registration module enables the students to register themselves in this system. Each student has to enter his/her Name, Email, and Registration ID, set a password, and upload a passport photo. It is automatically checked whether the entered Reg ID is unique and if it's true then the student is registered successfully in the database.

B. Faculty Registration

The faculty registration module enables the faculty members to register themselves who take the various course at the university. Registering in the portal will enable them to gain access to an analytics platform wherein they can access the overall class attendance along with individual students' attendance.

C. Image Capture

This module helps to capture images for further processing from the live video feed at specified intervals (say 10 seconds. It ensures whether the camera permission is given by the user and runs the capture device until the end of the class.

D. Face Detection

After images are captured by webcam at certain intervals, they are now passed to the face recognition library powered by dlib to detect face locations. It uses a Histogram of Oriented Gradients (HOG) to detect the face in the image.

E. Extraction of Face Encodings

After a face is successfully detected, it is passed through the Residual Networks (ResNet) deep learning model to extract the face encodings in a 128D vector format. The image passes through several convolutional layers which convolve over the image to extract features. This deep learning algorithm tries to find the facial pattern all over the given image. After a few convolutions, a pooling layer shrinks the information retaining the most important part. In ResNet, the connections between layers can be skipped by having the same dimensions [15]. In the end, after the average pooling facial encodings are extracted.

3.1. Datasets Used

The dataset contains 31 distinct classifications as well as 2562 celebrity face photos, by Vasuki Patel on Kaggle. Another 90 non-face images are introduced to make 2652 total images. It is used for the experimental testing of our tool.

3.2. Working of IOATS

The tool captures the facial images from the user's PC and compares them with available data in the dataset. It establishes a connection between the user's PC and server to download the registered image of the user from the server. Then, for the whole duration of the meeting, the tool captures data and uses the local image from the server to verify the face and consider the attendance. After the end of the duration of the meeting, the tool connects to the internet again to upload the attendance in the database.

Algorithm 1: Online Attendance Tracking System

Input*: Receives class details such as Regd. No of the student, Course ID, etc.*

Output*: Attendance for current class (Present (1) or Absent (0))*

```
function attendanceTracker(classdetails)
Initialize FaceFound = 0
Compute TotalPictures = (durationinmins * 60=10)
Compute Threshold = 0.75 *TotalPictures
while (Is Class Not Over) do
  Take a photo through Webcam
  Find face locations
  if (Face locations are found) then
    Extract the facial encodings
    Compare with encodings of the known image of student
    if (Student is recognized) then
      if (the student was present in a previous class) then
        Draw a bounding box in Green
      else
        Draw a bounding box in Red
        Update the FaceFound counter by 1
      end if
    else
      Print "Another person found/ You aren't recognized"
    end if
  else
    Print "Face Not Detected"
  end if
    Wait 10 seconds
end while
if (FaceFound > Threshold) then
  Attendance = Present
else
  Attendance = Absent
end if
Update the attendance in the database
end function
```

Face Recognition

After the extraction of face encodings of a known and unknown image captured from the camera, it compares both of them using Euclidian distance with the help of dlib's machine learning model. The model is trained upon labeled faces in the wild benchmark with an accuracy of 99.38% [16][17]. The model is derived from a ResNet-34 model. If the images are similar and match above a certain threshold then the face is recognized.

F. *Attendance Updation*

Finally, after face recognition, the student attendance is updated in the database as either present (1) or Absent (0) date-wise.

4. Experimental Set-up

The proposed approach has been implemented using Python Language Ver 3.7 and its various libraries such as Streamlit 0.79 for web GUI development, OpenCV-Python 4.5.1.48, Open-Source Computer Vision Library, for continuous video capturing, Dlib 19.21.1 King (2009) for face detection and face recognition. The implementation of this work has been carried out with the following hardware specifications: Intel Core i3 2 GHz processor (6th Gen), 4 GB RAM running under Windows 10 Professional. The tool has been built using PyCharm IDE. MySQL 8.0 is used as a back-end database system for managing student registration and attendance.

4.1. Implementation

Figure 3 depicts the registration page, where students must register themselves to utilize the attendance tool. They must submit information such as their name, email address, password, and personal photograph. Figure 4 shows the Faculty registration page, wherein the faculty members are required to register themselves to be able to access insights into students' attendance. They are required to provide details such as name, email, password, courses taught, and a unique authorization code (provided from the university end). The purpose of the authorization code is to prevent students from signing up as faculty in the portal.

Figure 5 shows the screenshot of the proposed attendance capturing tool which is the primary focus of this work. Students are required to log in at their class time providing course ID and registration ID. The tool will automatically start capturing the images for analysis until the end of the class. If the student was absent in the previous class, then the face is to be surrounded by a red boundary otherwise a green boundary. Figure 6 shows the Faculty Dashboard where the faculty members will be required to log in using their faculty ID and password. This dashboard will show them the attendance of all the students enrolled in the course taken by them. It enlists each student with no. of classes present and their attendance percentage for the respective course. Figure 7 shows the student dashboard which provides students the accessibility to verifying their attendance. It is a 24x7 service for students where they can track their attendance for several subjects in, which they are enrolled. They can monitor their attendance percentage and ensure they attend sufficient classes to be eligible to pass the course.

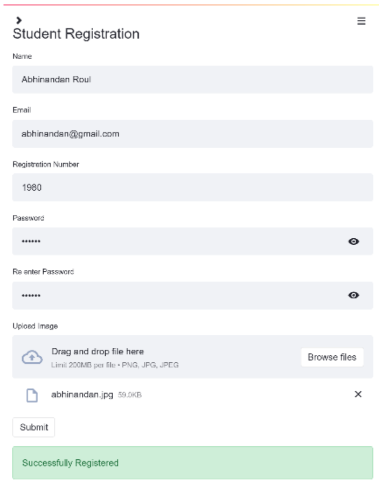


Fig. 3. Student Registration

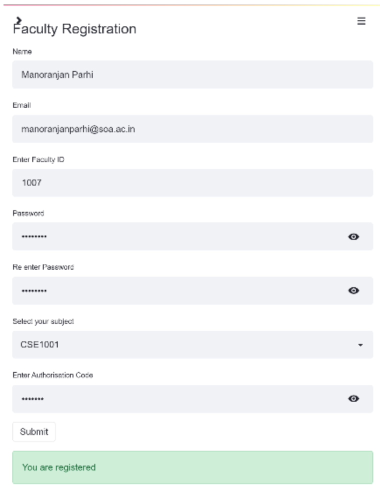


Fig. 4. Faculty Registration

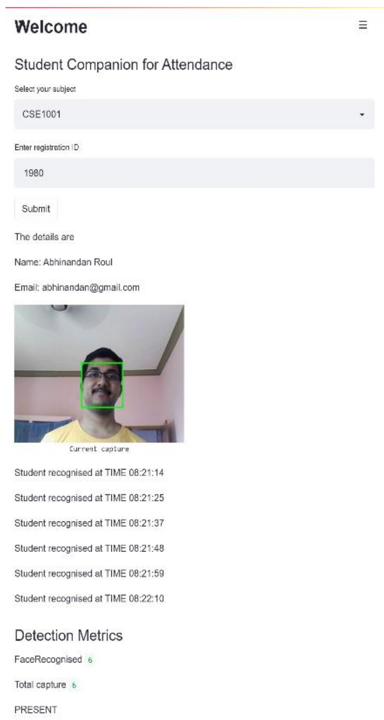


Fig. 5. Attendance Capture Tool

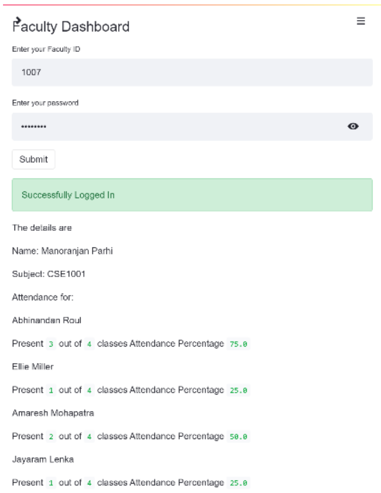


Fig. 6. Faculty Dashboard

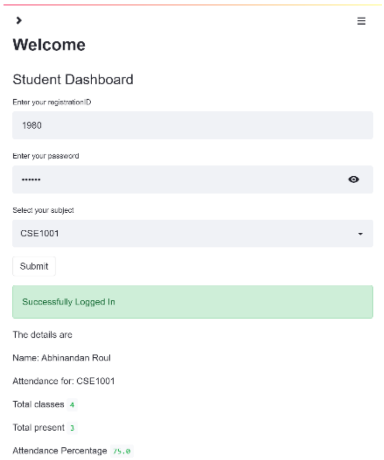


Fig. 7. Student Dashboard

5. Empirical Results

We utilized the Face Recognition Dataset given by Vasuki Patel from Kaggle to calculate detection accuracy. The collection includes face data from 31 distinct classifications as well as 2652 celebrity face photos along with non-face images. With a tolerance of 0.6, we compared the face encoding of known labeled pictures against the face encoding of unknown images. Accuracy (Acc), precision (Pre), recall (Rec), and F-Measure (F-M) are some of the performance metrics for classification purposes that have been studied in this research as depicted in equations 1, 2, 3, and 4 [18, 19, 20]. These evaluative measures are based on T_1 for True Positive, T_0 for True Negative, F_1 for False Positive, and F_0 for False Negative. The number of accurate predictions divided by the total number of input samples is known as accuracy (Acc). The ratio of accurately anticipated positive observations to the total number of correctly predicted positive observations is known as precision (Pre). The ratio of successfully anticipated positive observations to the total number of correctly predicted positive observations is referred to as recall (Rec). The weighted average of Precision and Recall is known as F1-Score (F1-S). This proposed technique was shown to be 97.73 percent accurate as depicted in Table 2.

$$Acc = (T_1 + T_0) / (T_1 + T_0 + F_1 + F_0) \quad (1)$$

$$Pre = T_1 / (T_1 + F_1) \quad (2)$$

$$Rec = T_1 / (T_1 + F_0) \quad (3)$$

$$F1\text{-}S = (2 * Pre * Rec) / (Pre + Rec) \quad (4)$$

A model's performance requires more than accuracy, generally in the case of imbalanced data i.e. the Area under the curve (AUC). Each threshold is represented by a Receiver Operating Characteristic Curve (ROC curve). The AUC assesses how well a model discriminates between two classes. The AUC is higher, better the performance of the model. The AUC measures a classifier's ability to distinguish across classes. The ROC curve is a graph that shows how well a classification model works across all categorization levels (FPR) [21]. Our model's ROC curve is plotted to test its classification performance as depicted in Fig. 9. In the figure, the wide distance between the green and blue lines defines the model's AUC, which is 0.88. Where the optimum diagnostic test of the classes is 1.0. Also, AUC = 0.88 reflects the model's accuracy in predicting classes, ensuring outperformance and capacity in categorizing classes.

Table 2: A Comparison of IOATS with the considered works

Work	Acc (in %)	Pre (in %)	Rec (in %)	F1-S (in %)
Balcoh et al [4]	85	-	-	-
Chintalapati et al [5]	95	-	-	-
Surekha et al [7]	80	-	-	-
Arsenovic et al [8]	95.02	-	-	-
Kakarla et al [11]	99.87	99.86	99.87	99.86
Patil et al [12]	97	96	97	-
IOATS [Proposed]	97.73	99.60	98.04	98.81

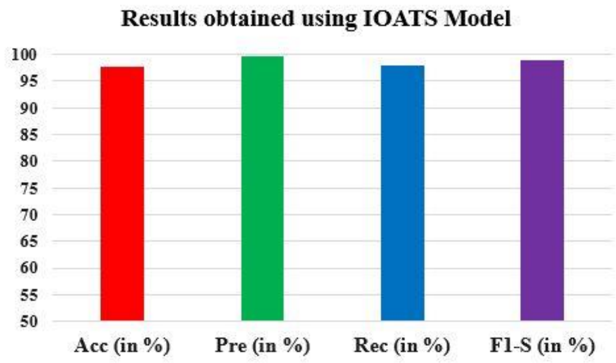


Fig. 8. Results obtained using this proposed IOATS Model

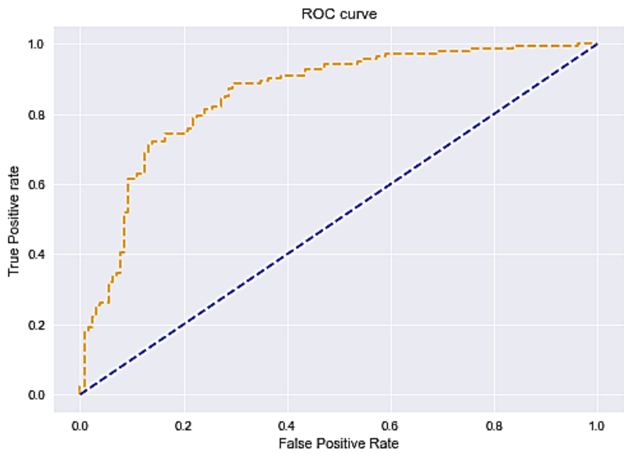
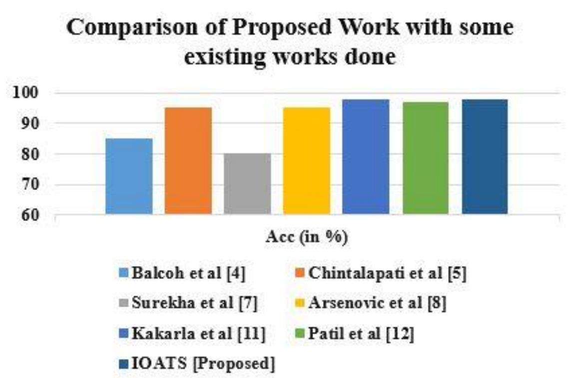


Fig. 9. The ROC curve was obtained from the proposed model.

Balcoh et al. [4] used Haar Cascade classifiers for the detection of faces, and EigenFace for recognition. Chintalapati et al. [5] use the Viola-Jones method which is fast and has a higher detection rate. PCA is used for feature extraction and SVM for classification. Surekha et al. [7] use Viola Jones in conjunction with MKD-SRC representation for face recognition. Arsenovic et al. [8] employ the use of CNN Cascade for face detection and CNN for the generation of face embeddings. Kakarla et al. [11] used a CNN approach with 20 layers which requires a higher training time and the use of better processing power. Patil et al. [12] proposed the use of the Viola-Jones technique with LDA, KNN, and SVM for face recognition. The above-mentioned techniques are traditional techniques with various limitations concerning lighting conditions and training data. In addition, LBPH is a traditional method that works in blocks of 3x3 pixels. Then after a set of operations, the final histogram is compared with the original histogram of sample data. Eigenface is a handy face recognition algorithm. It also saves time and space in processing and storage. PCA reduces an image's size greatly in a short period. Eigenface's frontal face accuracy is also good (around 90 percent). Because training and recognition data are strongly linked, Eigenface's accuracy is highly variable. Because the projection uses the pixel value as a reference, the accuracy diminishes with light intensity. The size and orientation of a photograph also affect its accuracy. Preprocessing the image is required for a good result. Fisherface is similar to Eigenface but better at identifying photos. We can use FLD to categorize the training set to deal with various people and facial expressions. We may acquire better facial expression accuracy than Eigenface. The fisherface is also more invariant to changes in light intensity since it eliminates the first three basic components.

The suggested work draws inspiration from ResNet-34. This is a ResNet model composed of 29 Conv layers. It takes the architecture of the residual neural network into consideration. In ResNet, we use an identity shortcut across layers with identical input and output dimensions. When the dimensions of the shortcut expand, the shortcut continues to conduct identity mapping with additional zero entries padded to compensate for the increased dimensions. In conventional CNNs, as we go further into the network, the error rate gradually decreases until it reaches a minimal value, at which point it begins to increase again. This occurs as a result of the exploding and vanishing gradient descent issue, which also results in model overfitting, which increases the error. The neural network extracts feature automatically during training, without requiring any handmade features. Deeper ResNets are capable of outperforming shallow ResNets in picture classification tasks due to their capacity to overcome the vanishing gradient issue. A Comparison of this proposed work IOATS with the considered works done outperforms others although falls short in some cases, as depicted in Table 2 and Fig. 10. We took into account each model's average accuracy % in Celebrity Face Dataset that is provided in various researches. Fig. 11 shows a comparative analysis of proposed IOATS with the performance of the Eigenface, Fisherface, and LBPH models.

Fig. 10. The comparison of this proposed work IOATS with some considered existing works done



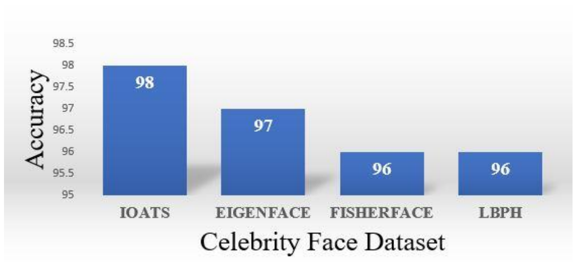


Fig. 11. Comparing the accuracy of different face recognition models

6. Conclusion And Future Scope

The proposed attendance tracking system, IOATS, is cantered on both students and staff. We think that this tool will aid in maintaining academic integrity when it comes to attendance. Even when classes are offered online, it will allow schools to implement stringent attendance standards. It ensures that all students who participated in the class are marked as present. During the lesson, the face recognition program does not utilize any network traffic because it operates locally. It continually takes images during the lesson and validates the student's identity. This program automates the attendance procedure, giving educators more time to lecture. We have planned to extend the functionalities of the proposed tool by adding the following features in the future.

- Automatic SMS notification regarding attendance;
- OTP based email verification during registration;
- Improvements in UI/UX Design;
- Providing support for mobile app users;

References

[1] A. Pati, M. Parhi and B. K. Pattanayak, “COVID19 Pandemic Analysis and Prediction Using Machine Learning Approaches in India”, In Advances in Intelligent Computing and Communication, Springer, Singapore, 2021, pp. 307–316.

[2] R. Olaniyan, O. Fadahunsi, M. Maheswaran, and M. F. Zhani, “Opportunistic Edge Computing: Concepts, opportunities and research challenges,” Future Generation Computer Systems, vol. 89. Elsevier BV, pp. 633–645, December 2018.

[3] I. A. Ujan and I. A. Ismaili, "Biometric attendance system," The 2011 IEEE/ICME International Conference on Complex Medical Engineering, 2011, pp. 499-501.

[4] N. K. Balcoh, M. H. Yousaf, W. Ahmed, M. I. Baig,” Algorithm for efficient attendance management: Face recognition based approach”, IJCSI International Journal of Computer Science, vol. 9, no. 4, pp. 146–150, 2012.

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

[6] D. Nithya, "Automated Class Attendance System based on Face Recognition using PCA Algorithm", International Journal of Engineering Research & Technology (IJERT) Volume 04, Issue 12 (December 2015).

[7] Surekha, B., Nazare, K. J., Viswanadha Raju, S., & Dey, N. (2016). Attendance Recording System Using Partial Face Recognition Algorithm. Intelligent Techniques in Signal Processing for Multimedia Security, 293–319.

[8] M. Arsenovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "FaceTime — Deep learning-based face recognition attendance system," 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), 2017, pp. 000053-000058.

[9] E. Rekha and P. Ramaprasad, "An efficient automated attendance management system based on Eigen Face recognition," 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, 2017, pp. 605-608.

[10] V. Suresh, S. C. Dumpa, C. D. Vankayala, H. Aduri, J. Rapa, (2019), "Facial Recognition Attendance System Using Python and OpenCV", Quest Journals - Journal of Software Engineering and Simulation, vol. 5, no. 2, pp. 18–29.

[11] S. Kakarla, P. Gangula, M. S. Rahul, C. S. C. Singh, and T. H. Sarma, "Smart Attendance Management System Based on Face Recognition Using CNN," 2020 IEEE-HYDCON, 2020, pp. 1-5

[12] V. Patil, A. Narayan, V. Ausekar, A. Dinesh, (2020), "Automatic Students Attendance Marking System Using Image Processing and Machine Learning", IEEE International Conference on Smart Electronics and Communication pp. 542–546.

[13] T. Sharanya, U. Kasturi, P. Sucharith, T. Mahesh and V. Dhivya, (2020) 'Online Attendance using Facial Recognition', IJERT International Journal of Engineering Research & Technology, Vol. 9, no.6, pp. 202–207.

[14] S. Bussa, A. Mani, S. Bharuka, and S. Kaushik, "Smart Attendance System using OPENCV based on Facial Recognition," International Journal of Engineering Research and, vol. V9, no. 03. ESRSA Publications Pvt. Ltd., Mar. 11, 2020.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[16] S. Sharma, K. Shanmugasundaram and S. K. Ramasamy, "FAREC-CNN based efficient face recognition technique using Dlib," 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2016, pp. 192-195

[17] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments", University of Massachusetts, Amherst, Technical Report, pp. 07–49, 2007.

[18] A. Pati, M. Parhi and B. K. Pattanayak. (2021, January)." IDMS: An Integrated Decision Making System for Heart Disease Prediction." In 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON) (pp. 1-6). IEEE.

[19] A. Pati, M. Parhi and B. K. Pattanayak, (2022). "HeartFog: Fog Computing Enabled Ensemble Deep Learning Framework for Automatic Heart Disease Diagnosis." In Intelligent and Cloud Computing (pp. 39-53). Springer, Singapore.

[20] A. Pati, M. Parhi and B. K. Pattanayak, (2022). "IADP: An Integrated Approach for Diabetes Prediction Using Classification Techniques." In Advances in Distributed Computing and Machine Learning (pp. 287-298). Springer, Singapore.

[21] A. Bhandari, "AUC-ROC Curve in Machine Learning Clearly Explained," 16 06 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/. [Accessed 13/08/2021]