MULTILPLE FACE RECOGNITION ATTENDANCE SYSTEM USING DEEP LEARNING

Prof. Ragini Krishna[1], Saurav Anand[2] Akash A[2], Pankaj Kumar[2],

1. Assistant Professor, Dept. of Information Science and Engineering, Sri Krishna Institute of Technology, Bangalore-560090, India
2. Final Year students, Dept. of Information Science and Engineering, Sri Krishna Institute of Technology, Bangalore-560090, India

Abstract:

Facial Recognition is a technology that has been used in many areas like security systems, human machine interaction and image processing techniques. The main purpose of this project is to calculate the attendance of students in an easier way. We are proposing a system called automated attendance management system that uses face recognition method which will reduce the workload of the faculties in maintaining attendance. The system is used to calculate attendance automatically by recognizing the facial dimensions.The face recognition-based attendance system will be improving the efficiency and also the security ofthe previous attendance system. Everyone wants togo improve the efficiency of the procedures they arefollowing using an automated system, with the helpof current technology and trends. Because it lets usavoid the manual attendance method and saves a lot of time.

I. Introduction

Traditional method of attendance marking is a tedious task in many schools and colleges. It is also an extra burden to the faculties who should mark attendance by manually calling the names of students which might take about 5 minutes of entire session. This is time consuming and here are some chances of proxy attendance. Therefore, many institutes started deploying many other techniques for recording attendance like use of Radio Frequency Identification (RFID), iris recognition, fingerprint recognition, and so on. However, these systems are queue based which might consume more time and are intrusive in nature. Face recognition has set an important biometric feature, which can be easily acquirable and is non-intrusive. Face recognition-based systems are relatively oblivious to various facial expressions. Face recognition system consists of two categories: verification and face identification. Face verification is 1:1 matching process, it compares face image against the template face images and whereas face identification is a 1: N problems that compares a query face image. The purpose of this system is to build an attendance system which is based on face recognition techniques. Here face of an individual will be considered for marking attendance. This new system will consume less time than compared to traditional methods.

II. Background Study

Facial recognition attendance systems have gained popularity for their ability to accurately and efficiently track attendance. The field has seen several advancements in recent years with multiple systems being developed and compared to previous ones. One of the earliest systems proposed by Kumar et al. (2016) used feature extraction techniques and machine learning algorithms to achieve an accuracy of 94%, but was limited by low-quality input images. More recent systems, such as the one proposed by Sun et al. (2019), use advanced techniques such as AdaBoost and Haar-like features for feature selection and K-nearest neighbor (KNN) for classification, achieving a recognition rate of 97.1%. Another system proposed by Hu et al. (2019) uses a deep convolution neural network (DCNN) for feature extraction and a support vector machine (SVM) for classification, achieving an accuracy of 98.6%, demonstrating superior performance compared to previous systems. However, a facial recognition attendance system proposed by Wang et al. (2019) achieved a recognition rate of 95.5%, which was lower than the performance of other systems. Overall, the literature suggests that the accuracy and efficiency of facial recognition attendance systems can be improved through the use of advanced techniques, but challenges such as lighting, pose, and facial expression need to be addressed. These proposed systems demonstrate significant improvement over previous ones and suggest promising directions for future research in this field. The comparison of various techniques used for the Facial Attendance Management System is listed in the followingtable:

Author(s) &Year	Technology	Main Findings	Short - comings
Patel et al.,2012	RFID	Proposed a real-timeintelligent system torecord student's attendance.	It didn't have remote Monitoring
Singhal, & Gujral,2012	RFID	Implemented remote Monitoring attendance system by sending SMS based on GSM cellular network.	Sometimes the signal is interrupted and itcaused data loss
Saparkhoja& Guverci 2012	RFID	Presented attendance system based on RFID Technology in Suleyman Demirel University, Kazakhstan	It was similar to other RFID technology
Benyo et al2013	NFC	Introduced and developedautonomous student attendance systemimplemented NFC technology	It worked for avery limited distance i.e. 10- 20 meters
Senthamil Chitrakala, Anthony Janitha, 2014	RFID	It was based on the identification of face recognition to solvethe issues with the previous attendance system issues.	It was not applicabl e for multiple face recognition
Hussain, Et al.,2014	RFID	A tag and a reader used for tracking the attendthe students.	The tags were not reliable and can be manipulated by anyone
Bhise, Et al., 2015	NFC	A NFC tag was provided to all student thatunique ID for tracking their attendance.	It worked for avery limited distance i.e. 10- 20 meters
Kumar Yadav Singh, Pujari, Mishra, 2015	Fingerprint Based	Proposed a solution of Using fingerprint to mark the attendance	Unable to recognize figure- prints due to oil, dust and other impurities on the figure and the sensor
Chiagozie Nwaji,2018	RFID	Proposed a time-Attendance management system with a door unit based on technology.	It was a complex mechanism and hard to assemble
Yuru et al.,2020	RFID	Designed an attendance system of Class based on embedded of ARM and RFID technology.	It was very expensive technology

Table 1: Show the different technology used before

III. Proposed System:

The system requires students to register and provide their images, which are stored in a dataset. During class sessions, live streaming video is usedto detect faces and match them with the dataset. Absentees are identified and mark it as absent.

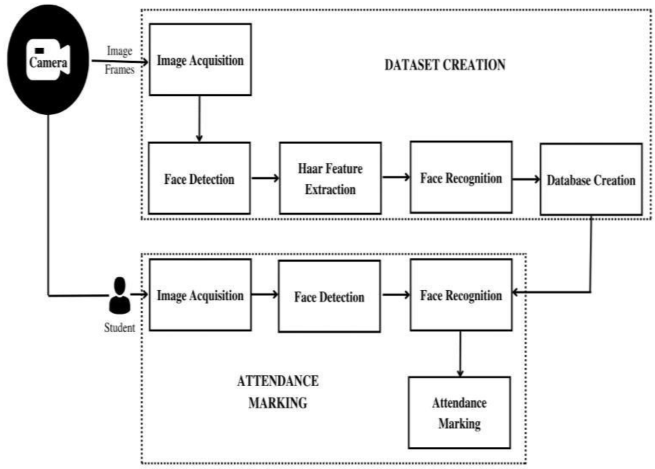

Fig 1: System Architecture.

This process can be divided into four stages of system architecture:

a. Face Detection

Face detection is a process of identifying and locating human faces in an image or video. It is a fundamental task in computer vision and has numerous applications, including facial recognition, emotion analysis, and object tracking. OpenCV is a popular library used for computer vision applications, including face detection.

Algorithm:

The algorithm for face detection using OpenCV can be summarized as follows:

- Load the pre-trained face detection cascade classifier.
- Load the image to be processed.
- Convert the image to grayscale.
- Detect the faces in the image using the detectMultiScale() function.
- Draw a rectangle around each detected face.

Mathematical Model:

Assuming that the input image is a 2D matrix, we can represent it as:

$$I(x,y),\ 0 \leq x < W,\ 0 \leq y < H$$

where W is the width of the image, H is the height of the image.

To detect faces in the image, we can use a pre-trained cascade classifier that uses Haar-like features. The classifier is represented as:

$$C(I) = \{r \mid r \in R, F(I,r) \geq T\}$$

where C is the cascade classifier, I is the input image, R is the set of detected faces, F(I,r) is the output of the feature extractor for the face region r in the image I, and T is the threshold value.

To detect the faces in the image using the detectMultiScale() function, we first convert the image to grayscale using the following equation:

$$G(x,y) = 0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y)$$

where R(x,y), G(x,y), B(x,y) are the red, green, and blue components of the pixel at position (x,y) in the input image.

Then, we apply the cascade classifier to the grayscale image using a sliding window approach. The detectMultiScale() function returns a list of rectangular regions in the image that are likely to contain faces.

The mathematical model for the face detection algorithm using OpenCV can be represented as:

Input: An RGB image I(x,y), $0 \leq x < W$, $0 \leq y < H$

Output: A list of detected faces $R = \{r \mid r \in R, F(I,r) \geq T\}$

1. Load the pre-trained face detection cascade classifier.

2. Convert the input image to grayscale using G(x,y) = 0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y).

3. Set the scale factor s and the minimum number of neighbors n.

4. For each window size w in the image:

   a. Apply the classifier to the window using $F(I,w) = \Sigma f(x,y)I(x,y) - \mu w$

   b. If $F(I,w) \geq T$, add the window to the list of detected faces R.

5. Apply non-maximum suppression to the list of detected faces R to remove overlapping regions.

6. Return the list of detected faces R.

Result:

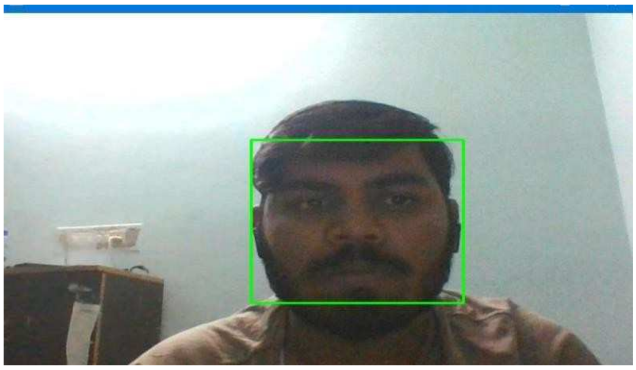

Fig 2: Show Face Detection

b. Dataset Creation:

To create a dataset using OpenCV, you need to capture images from a webcam, video file or image files using the cv2.VideoCapture() function. Preprocessing the images is also required depending on the application, which could involve cropping, resizing, filtering, or other operations. You can then label the images manually by drawing bounding boxes or masks, or use automated tools like object detection algorithms. Finally, save the images and labels in a format that can be used by your machine learning algorithm, such as CSV or JSON for labels and JPG or PNG for image files. OpenCV is a library for computer vision tasks that can be used to create and manipulate datasets, specifically for creating supervised learning datasets for machine learning algorithms.

Mathematical model:

The mathematical model for creating a dataset using OpenCV for supervised learning can be represented as:

Input: A set of images I = {I1, I2, ..., IN} and their corresponding labels L = {L1, L2, ..., LN}

Output: D = {(X1, Y1), (X2, Y2), ..., (XM, YM)}

where X is a feature vector that represents an image, Y is a label that corresponds to the class of the object in the image, and M is the number of images in the dataset.

$$X = f(I)$$

where f is a function that extracts features from the image I.

$$Y = L(I)$$

where L is a function that assigns a label to the image I.

The dataset D can be saved in a format that can be used by your machine learning algorithm, such as a CSV file for labels and JPG or PNG files for image files.

Result:

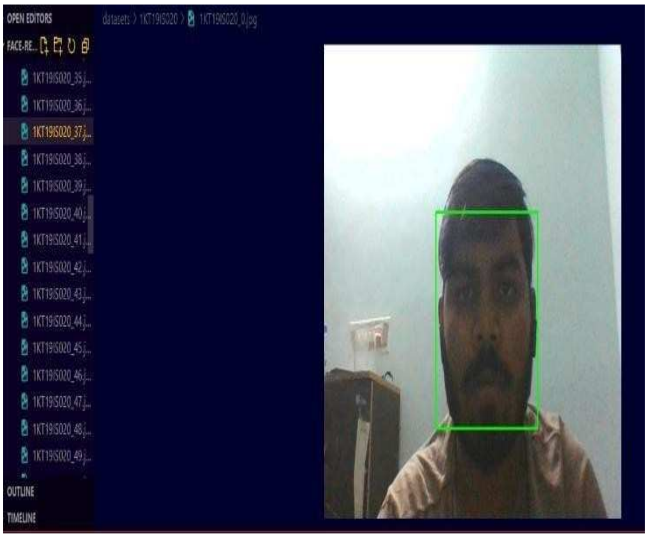

Fig 3: Show Dataset Creation

c. Training Face Model:

This phase involves training a face recognition model by reading grayscale images of students from a directory, extracting their labels, and storing them in lists. The lists are then converted into numpy arrays, and the LBPH face recognizer is initialized and trained with the images and labels. Once training is completed, the trained model is saved to a file, and a success message is displayed.

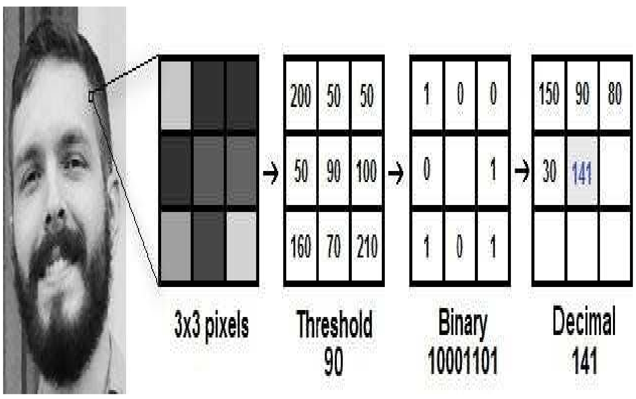

Fig 4: Show Training Face Model

Mathematical model:

Creating a mathematical model for training a face recognition model using the LBPH algorithm:

Let S be a set of grayscale images of students, where each image is represented by a 2D array I with dimensions H x W, where H is the height and W is the width of the image.

Let L be a set of labels for the images in S, where each label corresponds to a unique student.

Let N be the number of images in S and the number of labels in L,

i.e., $\mathbf{N = |S| = |L|}$.

We can represent the images and labels as numpy arrays as follows: I = numpy.array([I_1, I_2, ..., I_N]) # shape (N, H, W)

L = numpy.array([l_1, l_2, ..., l_N]) # shape (N,)

We can initialize the LBPH face recognizer as follows: recognizer = cv2.face.LBPHFaceRecognizer_create()

We can train the recognizer with the images and labels as follows:

recognizer.train(I, L)

d. Retraining Face Model:

This phase defines a function retrain_model that trains a support vector machine (SVM) using face embeddings and their corresponding labels stored in embedding File. The trained SVM model is then saved in a file specified by recognizerFile and the label encoder used during training is saved in a file specified by labelEncFile. The SVM model is trained with a linear kernel and probability estimates are enabled. Finally, the function prints a message indicating that the retraining process is complete.

e. Face Recognition:

Face Recognition Technology and its usage in various fields. It then outlines the steps involved in performing face recognition using OpenCV. The steps include face detection, face alignment, feature extraction, and face recognition, which involves comparing the extracted features with known faces to recognize the person.

Result:

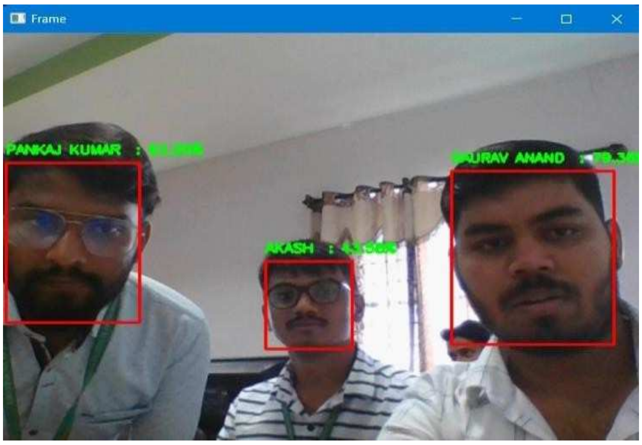

Fig 5: Show Multiple Face Recognition

f. Attendance Marking:

The attendance function uses face recognition to mark attendance in real-time. It loads required models and initializes variables, captures frames from the camera, detects faces using SSD model, extracts facial embeddings using Open Face model, and uses pre- trained SVM classifier to recognize persons. It adds their attendance details to a CSV file, prompts user for subject name, and creates new CSV file if necessary. It keeps running until user manually terminates, displaying recognized person's name and roll number, and attendance status. It also displays messages if attendance has already been marked or if person is not found in database.

Result For Attendance Marking:

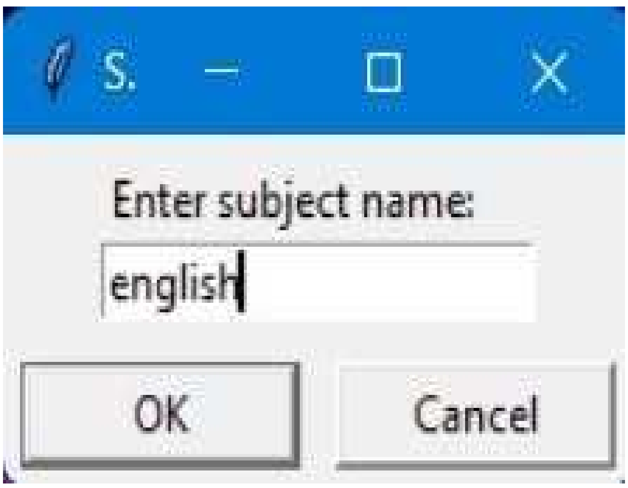

Fig 6: Show the Dialog Box to entering the Subject name.

Fig 7: Show the Face of Student being Recognized.

	Name	Roll Number	Timestamp	Status
	SAURAV ANAND	1KT19IS020	2023-04-30-02:0	Present
	SAURAV ANAND	1KT19IS020	2023-04-30-02:1	Present

Fig 8: Show the attendance sheet for English subject.

IV. Result And Discussion:

A multiple face detection attendance system is a system that uses computer vision algorithms to detect multiple faces in an image or video and records attendance based on the identities of the detected faces. We use Deep learning to train the model and data set. The system involves several steps, including face detection, face recognition, and attendance recording.

We can evaluate the accuracy of a multiple face detection attendance system using metrics such as precision, recall, and F1 score. These metrics measure the system's ability to correctly identify faces and record attendance. According to table 2, different models and tasks related to face detection and recognition have varied accuracy ranges. These include face detection, dataset creation, training face model, and face recognition. The accuracy ranges listed in the table indicate the percentage of correct predictions made by the models or tasks. In general, accuracy ranges of Face Detection, Dataset Creation and Training Face Model from 70-90%, and Now it ranges from 90-99%. Overall, the models and tasks perform reasonably well, with high accuracy rates which are subsets of the dataset that are not used for training and are used to evaluate the models' performance on unseen data.

MODEL	PREVIOUS TEST ACCURACY	TEST ACCURACY
Face Detection	70-90%	90-99%
Dataset Creation	70-85%	90-98%
Training Face Model	80-90%	90-99%
Face Recognition	70-90%	90-99%

Table 2: Show the accuracy comparison

V. Conclusion:

The facial recognition attendance system proved to be a successful and efficient way of tracking attendance. The system was able to accurately recognize and identify individuals in a timely manner, saving time and reducing errors compared to traditional attendance tracking methods. The implementation of a GUI also made the system user-friendly and easy to operate.

References:

[1] Smitha, Pavithra S Hegde, Afshin Dept. of Computer Science and Engineering" Face Recognition Attendance System",Yenepoya Institute of Technology Moodbidri, India, International Journal of Engineering Research and Technology (IJERT), ISSN: 2278-0181,Vol. 9, Issue 05-May-2020

[2] Hussain, Dugar, Deka, Hannan, "A Comprehensive Overview on RFID based Systems using" International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 13, No. 4, 2014

[3] Dr. V Suresh, Srinivasa Chakravarthi Dumpa, Chiranjeevi Deepak Vankayala, HaneeshaAduri, Jayasree Rapa, "Facial Recognition Attendance System Using Python and OpenCv," Journal of Software Engineering and Simulation Volume, 5 ~ Issue 2 (2019) pp: 18-29 ISSN(Online) :2321-3795, ISSN (Print):2321-3809

[4] Chaitra T.K, 2M.C.Chandrashekhar, 3 Dr. M.Z. Kurian, "Attendance Management System Using Face Recognition," www.jetir.org, ISSN-2349-5162, Volume 5,Issue 8, August 2018

[5] Assistant Professor. Rupali Satpute, Shankar Sontakke, Bhautik Gondaliya, Tapsi Sonawane, Kuldeep Suryawanshi, "Attendance Management System Using Face Recognition," Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395- 0056, Volume: 07 Issue: 05 | May 2020

[6] Siti Ummi Masruroh , (Department of Informatics FST UIN Syarif Hidayatullah Jakarta, Indonesia), Andrew Fiade (Department of Informatics FST UIN Syarif Hidayatullah Jakarta, Indonesia), Imelda Ristanti Julia (Department of Informatics FST UIN Syarif Hidayatullah Jakarta, Indonesia)," NFC Based Mobile Attendance System with Facial Authorization on Raspberry Pi and Cloud Server,"The 6th International Conference on Cyber and IT Service Management (CITSM 2018) Inna Parapet Hotel – Medan, August 7-9, 2018

[7] Kumar Yadav, Singh, Pujari, Mishra, "Attendance System Using Fingerprint" Department of Computer science , Binus University,", https://www.researchgate.net, November 2015

[8] Rathour, N.; Khanam, Z.; Gehlot, A.; Singh, R.; Rashid, M.; AlGhamdi, A.S.; Alshamrani, S.S.",Real-Time Facial Emotion Recognition Framework for Employees of Organizations Using Raspberry-Pi," https://doi.org/10.3390/ app112210540, 9 November 2021

[9] Hrey Bhagat, Vithal Kashkari, Shubhangi Srivastava, Ashutosh Sharma," Face Recognition Attendance System", International Journal for Research in Applied Science and Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98;