

## **Improved Facial Recognition for Attendance Systems**

Sravan Gangishetti, Harshal Bohra, Hari Purnapatre, Satvik Shukla

Information Technology, Pune Institute of Computer Technology, Pune, Maharashtra, India

## ABSTRACT

Article Info Volume 7, Issue 6 Page Number: 456-467 Publication Issue : November-December-2020

Attendance is an important part of educational life. The attendance methodology which exists currently is either totally manual or requires some human assistance. In traditional methods, the teacher manually marks the attendance of each student on paper. There are many flaws in this method. As the process is manual there is a chance of error in the marking of attendance. Similarly, marking of false attendance by the students is also possible. Moreover, the attendance is then required to be updated in the online database of the institute through which it is accessible to the students. Also, after every month a report of all the students is to be generated. This overall process is cumbersome and time consuming. There also exist some automated systems for attendance like fingerprint verification and RFID but they also require some human support. Here, the automated attendance system using facial recognition plays an important role which requires no human intervention and is fully automatic. Face recognition has always remained a major focus of research because of its non-invasive nature and because it is people's primary method of person identification.

## Article History

Accepted : 15 Dec 2020 Published : 30 Dec 2020

**Keywords :** Spatio-Temporal Crime Prediction, Machine Learning, Alert System.

## I. INTRODUCTION

Face Recognition is relevant to a variety of places like educational institutes, offices and many places where attendance is to be monitored. The methodology of face recognition can also be extended for security purposes as well as in defense. The whole world is moving towards automation hence this project is the step towards this direction. The project covers a lot of domains like Image processing, Machine learning, Networking, Python programming, etc. Automation has been emerging tremendously in the recent days in each and every domain. A lot of time is irrelevantly wasted on manual attendance daily in schools and colleges. The accuracy of the records is also not properly maintained. Moreover manual updating of the database is not real time. Hence automation in the attendance system is greatly required. This attendance system using face recognition solves all the above problems

**Copyright:** © the author(s), publisher and licensee Technoscience Academy. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License, which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited

making the process fully automated. Facial recognition attempts will be made to detect and recognize the students seated in the classroom by experimenting with various facial detection and recognition techniques. The recognized students' attendance will be marked automatically in the attendance database.

#### II. Related Work

Currently the attendance system used in classrooms is manual. This requires a certain amount of time at the beginning of every lecture. Also, the teacher marks attendance on paper and has to update it manually into the database at regular intervals. This is prone to errors and wastes a lot of time and resources which could be utilized better. Hence, there is a need for improvement in this process. In face recognition a simple camera is required to be installed in the classroom and connected to a computer. Various reliable face recognition techniques are available and most of them are open source. As the camera keeps functioning during the lecture without any intervention it satisfies the one basic objective of not disturbing the class. Also, as the system is completely digital it satisfies the objective of making the process automated as final output is directly updated in the database. There are three kinds of approaches which are extensively used and studied. In feature-based approach the local features like nose, eyes are segmented and it can be used as input data in face detection to ease the task of face recognition. In a holistic approach the whole face is taken as the input in the face detection system to perform face recognition. The hybrid approach is a combination of feature based and holistic approach. In this approach both local and whole face is used as the input to face detection system. It should however be noted that the existing face recognition techniques are not one hundred percent (100%) efficient just yet. Typical efficiencies range between 40% to 60%.

#### III. System Architecture

To solve the above-mentioned problem firstly, a camera will be installed in the classroom. Further capturing images from the classroom at certain time intervals, extracting the faces from the images, preprocessing of the face images and running them through different facial recognition algorithms like PCA, LBPH, etc. will be done and accuracy of each method will be checked. After application of the algorithms, the best one is chosen on the basis of accuracy of the parameters. The implementation of algorithms will take place in Python programming language. The accurately recognized students will be obtained from the entire process.



Fig 1. Block Diagram of Basic System

- Camera is installed in the classroom, takes photos of the classroom periodically in onehour lectures.
- The processor receives this image and using a neural network detects all faces in these images.
- These detected faces are cropped and passed to recognizer.
- 4) The recognizer predicts the person to whom the face belongs and passes and stores the roll number of the students detected in each frame till the end of lecture.
- 5) At the end, the final output is calculated using

a threshold and final attendance is output as a .csv file.

#### 6)

## A. Face Detection

1) Haar-Cascade method (Initial Approach)

The first step in recognizing any face is detecting the faces in the image. There are various methods to detect faces, the most common and easiest to use being the Viola-Jones algorithm. [6] Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Haar cascade consists of a large number of filters which are passed repeatedly over the image. If the first filter gives positive another filter is passed and so on till a face is declared as detected. Hence it is called a cascade (waterfall structure). [6] OpenCV provides easy use of this by the way of Haar Cascades library. This method reduces computing time and gives satisfactory results. Hence, Haar- cascades were used for initial face detection.

**Step 1:** First we need to load the required XML classifiers.

**Step 2:** The next step is to load our input image (or video) in grayscale mode. If the image is in RGB, we can convert it into grayscale by using cv.cvtColor() function.

**Step 3:** Find the faces in the image. If faces are found, it returns the positions of detected faces as Rect(x,y,w,h).

**Step 4**: Once we get these locations, we can create a ROI for the face and apply eye detection on this ROI.

**Step 5:** These faces can then be cropped by using the coordinates obtained.

2) Detection using TensorFlow Object Detection API

TensorFlow is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library, TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open source license on November 9, 2015and is also used for machine learning applications such as neural networks. Object Detection API is an open source framework built on top of TensorFlow that makes it easy to construct, train, and deploy object detection models. The model used is a high-speed model that can work on detecting video feed at high fps, the single shot detection (SSD) network. Training a neural network for Object detection requires huge dataset, high end GPU and approximately a week of training time. We have used a pre-trained model which is trained on the Wider face dataset. TensorFlow is easily installed along with python. The model is loaded using TensorFlow interfacing functions. When an image is supplied to the model it detects all faces in the image and gives output as bounding boxes around the face in form of a 3D array. The coordinates of the faces are extracted in the image from the array.

## B. Image Enhancement using CLAHE

## 1) Image enhancement

It is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. When we increase the contrast of an image and filter it to remove the noise "it looks better". [7] Improvement in quality of degraded images can be achieved by using application of enhancement techniques.

# 2) Contrast limited adaptive histogram equalization (CLAHE)

Histogram equalization (HE) is a widely used image contrast enhancement technique. The basic

disadvantage of HE is it changes the brightness of the image. In order to overcome this drawback, various HE methods have been proposed. [2] These methods preserve the brightness on the output image but do not have a natural look. In order to overcome this problem, the idea of histogram equalization technique for image enhancement using contrast limited adaptive histogram equalization (CLAHE) method is used. [7] In Contrast Limited Adaptive Histogram Equalization, we limit the contrast and take the histogram.

**Step 1:** Input the image and calculate the grid size based on maximum image dimension.

**Step 2:** Starting from the top left corner find grid points on the image.

Step 3: For each grid point-

-Calculate histogram with area equal to window size and grid point as centre.

-Clip the histogram above the clip limit and use it to find CDF.

**Step 4**: For each pixel

-Find 4 neighbour grid points for the pixel and based on their CDF find the mapping of that pixel at 4 grid points.

**Step 5:** Apply Interpolation among these values to get mapping at the current pixel location.

**Step 6:** Now map this intensity into the output image with range [min max].

## D. Face Alignment

Face alignment is the process of:

1) Identifying the geometric structure of faces in digital images.

2) Attempting to obtain a canonical alignment of the face based on translation, scale, and rotation.

Face alignment relies only on the facial landmarks themselves (particularly, the eye regions) to obtain a normalized rotation, translation, and scale representation of the face. The reason we perform this normalization is that facial recognition algorithms like Eigenfaces, LBPs for face recognition, Fisher faces, and deep learning/metric methods can all benefit from applying facial alignment before trying to identify the face. By using face alignment higher accuracy is achieved from the face recognition model. [8]. All faces across an entire dataset should be centred in the image, rotated such that the eyes lie on a horizontal line (i.e., the face is rotated such that the eyes lie along the same y-coordinates) and should be scaled such that the size of the faces is approximately identical.



Fig 2 : Face Alignment

## E. Face Recognition

PCA based Eigenface Method (Initial Approach)

Principal Component Analysis (PCA) is а mathematical procedure that uses an orthogonal transformation to convert a set of values of possibly correlated M variables (faces) into a set of values of K uncorrelated variables called principal components (eigenvectors). [4] The number of principal components (Eigenfaces) is always less than or equal to the number of original variables (face images) i.e. K  $\leq$  M. This transformation is defined in such a way that the first eigenface shows the most dominant "direction/feature" of the training set of images and each succeeding component in turn shows the next most possible dominant "direction/feature"; all under the constraint that it be uncorrelated to the preceding Eigenface. To reduce the calculations needed for finding these eigenfaces, the dimensionality of the original training set is reduced before they are calculated. Since eigenfaces show the "directions" of data and each preceding eigenface shows less "directions" and more "noise", only few first eigenface (say K) are selected whereas the last of the eigenfaces are discarded.



**Fig 3 :** The average face, the first and last eigenface which is mainly noise

The above image shows the average face and the first and last eigenfaces that were generated from a collection of 30 images each of 4 people. The average face shows the smooth face structure of a generic person, the first few eigenfaces will show some dominant features of faces, and the last 19 eigenfaces (e.g.: Eigenface 119) are mainly image noise. Figure 2.7 shows the first 32 eigenfaces. Images 32 -119are discarded because they are mainly noise.



Fig 4 : The dominant 32 eigenfaces representing all the images in the set

Representing an image as a combination of K Eigenfaces reduces the number of values needed to

recognize it from M to K. This makes the recognition process faster and free of errors caused by noise.

## 1) PCA Algorithm

**Step 1**: Create a training set and load it. The training set consists of total M images. Where each image is of size N X N.

**Step 2**: Convert the images in the training set to face vectors.

A training set consisting of total M image





**Step 3**: Normalize the face vectors i.e. remove all the common features. This is done by calculating the average face vector and then subtracting it from all the face vectors. Figure shows the average face (U) which is then subtracted from all the face vectors.



Øi = Ti - U

Fig 6 : The average face is represented by the blue column labelled U

**Step 4:** Calculate the eigenvectors from the covariance matrix.

$$C = A.A^T$$

Where  $\mathbf{A} = [\mathbf{Ø}_1, \mathbf{Ø}_2, \mathbf{Ø}_3 \dots \mathbf{Ø}_M]$  and is of dimension  $N^2 XM$ 

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{A}^{\mathsf{T}} = \mathbf{N}^2 \mathbf{X} \cdot \mathbf{M} \quad \cdot \mathbf{M} \mathcal{X} \mathbf{N}^2 = \mathbf{N}^2 \mathbf{X} \cdot \mathbf{N}^2$$



Fig 7 : The covariance matrix

This is a very huge matrix as depicted in Figure 2.15 where N = 50, N2=2500. This would yield 2500 x 2500 Eigenfaces. To find K eigenvectors from M = 2500 eigenvectors with K<M would take a lot of time due to the many calculations that would need to be done and hence the need for Dimensionality reduction.

**Step 5**: Reduce the dimensionality of the training set.To reduce these calculations on the needed eigenfaces, we calculate them from a covariance matrix of reduced dimensionality.

Where,

C=AT.A = M x N2. N2 x M = M x M= 100 x100...... f M = 100

This would give 100 eigenvectors as shown in the Figure.





Fig 8 : The reduced dimensionality matrix in the low dimensional space

**Step 6**: Select K best eigenvectors such that K < M and can represent the whole training set. The yellow bars

represent the selected K Eigenvectors that are sufficient to represent the whole training set.



Fig 9: The selected K eigenvectors that can represent the whole set of images

**Step 7**: Convert lower dimensional K eigenvectors into original face dimensionality.

 $U_i = AV_i$ 

 $U_i = ith$  eigenvector in the higher dimensional space  $V_i = ith$  eigenvector in the lower dimensional space

**Step 8**: Represent each image as a linear combination of all K eigenvectors plus the mean/average face.



a weighted sum of the K eigen faces + the average face



**Step 9**: For each image in the training set calculate and store the associated weight vectors. A weighted face vector  $\Omega$  which is the eigenface representation of the ith face weight vector for each face is calculated.

#### 2) PCA Algorithm Flow Chart :

The face recognition algorithm flow chart is basically a method of checking which training image is most similar to the input image, out of the whole training set.



Fig 11 : Flowchart of PCA Algorithm

## 3) LBPH Algorithm :

The face image is first divided into small regions from which the Local Binary Pattern (LBP) features are extracted and concatenated into a single feature histogram efficiently representing the face image. [9]. The textures of the facial regions are locally encoded by the LBP patterns while the whole shape of the face is recovered by the construction of the face feature histogram. The idea behind using the LBP features is that the face images can be seen as composition of micro-patterns which are invariant with respect to monotonic grey scale transformations. Combining these micro patterns, a global description of the face image is obtained.

**Step 1:** Define the parameters (radius, neighbors, grid x and grid y) using the Parameters structure from the LBPH package. Init function is then called passing the structure with the parameters. If parameters are not set, it will use the default parameters.

**Step 2:** Training the algorithm. Call the Train function passing a slice of images and a slice of labels by parameter. All images must have the same size. The labels are used as IDs for the images, so if you have more than one image of the same texture/subject, the labels should be the same.

**Step 3:** The Train function will first check if all images have the same size. If at least one image has not the same size, the Train function will return an error and the algorithm will not be trained.

**Step 4:** Then, the Train function will apply the basic LBP operation by changing each pixel based on its neighbours using a default radius defined by the user. The basic LBP operation can be seen in the following image (using 8 neighbours and radius equal to 1).



Fig 12:3 x 3 neighborhood in LBPH

**Step 5:** After applying the LBP operation we extract the histograms of each image based on the number of grids (X and Y) passed by parameter. After extracting the histogram of each region, we concatenate all histograms and create a new one which will be used to represent the image.



Fig 13 : Histogram generation from each block of grid

**Step 6:** The images, labels, and histograms are stored in a data structure so we can compare all of it to a new image in the Predict function.

**Step 7:** Now, the algorithm is already trained and we can Predict a new image.

**Step 8:** To predict a new image we just need to call the Predict function passing the image as parameter!. The Predict function will extract the histogram from? the new image, compare it to the histograms stored in? the data structure and return the label and distance corresponding to the closest histogram if no error has occurred. Closer to zero the distance, the greater is the confidence.

$$LBP(\mathbf{x}_{c},\mathbf{y}_{c}) = \sum_{p=0}^{P-1} 2^{p} s(\mathbf{i}_{p} - \mathbf{i}_{c})$$

Where,  $(\mathbf{x}_c, \mathbf{y}_c)$  as central pixel with intensity  $\mathbf{i}_c$ and  $\mathbf{i}_n$  being the intensity of the neighbor pixel. **s** is the sign function defined as:

A more formal description of the LBP operator can be given as:

$$s(x) = \begin{cases} 1 & \mathrm{if} \ x \ge 0 \\ 0 & \mathrm{else} \end{cases}$$

A fixed neighborhood fails to encode details differing in scale. Hence, the LBPH operator was further extended to use a variable neighborhood. The idea is to align an arbitrary number of neighbors on a circle with a variable radius.



Fig 14 : Extension of LBPH Operator

For a given Point  $(x_c, y_c)$ 

4) Comparison of Histograms

LBPH package provides the following metrics to compare the histograms:

Chi Square

Euclidean Distance

Normalized Euclidean Distance

Absolute Value

Euclidean distance metric is used as the default metric to compare the histograms.[9]

The returned calculated distance can be used as a confidence measurement.

Lower confidences are better because it means the distance between the two histograms is closer.

#### IV. Results and Evaluation

#### 1) Face Detection using Haar Cascades

Initially, Face detection was tried using the Haar-Cascade method. The frame captured from the classroom was passed through the face detection function. It was observed that using the Haar-Cascade method gave some false positive results. These false positives if passed to the recognizer may result in wrong face recognition.



Fig 15 : Result of detection using Haar-Cascade

Due to low accuracy and detection of false positives we decided to shift the face detection part to TensorFlow Object Detection API.



**Fig 16** : Result of detection using TensorFlow API

The problem of false detections in the Haar- Cascade method is successfully solved using the TensorFlow API.

Also, the accuracy of this method is better than Haar-Cascades

Technique used	Haar- Cascade	TensorFl ow API	TensorF low API
Total no. of students	22	27	14
No. of students detected	12	23	14
False	3	0	0

Positives			
Accuracy	54.5%	85.2%	100%

 Table 1 : Face Detection analysis

## 2) Database of Students

The images of students were taken for various face positions & expressions and a database was created.

Below are some of the samples of faces from the database.



Fig 21 : Sample database of students

3) Installation of camera in classroom

The camera was installed in the classroom at a location from where most of the classroom will be visible.

The camera was turned on and feed was recorded.

Number of frames were captured from the feed.

It was later studied to determine lighting conditions and minimum and maximum face size in pixels.





Fig 22 : Frames from camera feed in classroom

## 4) Face Recognition

LBPH method is used for Face recognition. The classifier was trained on a database of faces created and then cropped detected faces were passed to it.

Number of Training Images per Subject are given as follows:

No. of students	No. of Training Images
1	60
2	65
3	65
4	40
5	35
6	35
7	40
8	40
9	25

10	40
11	70
12	10
13	25
14	20

Table 2 : Training	images per person
--------------------	-------------------

The Results of Recognition are as follows:

Total Number of	14
Students	
Number of	12
Students	
successfully	
detected	
Error	2
Accuracy	85.71%

## Table 3 Face recognition results

## 5) Accuracy Check

The accuracy of recognition was cross verified with the help of a confusion matrix. In a confusion matrix it is checked whether a particular subject is correctly recognized as itself or is incorrectly matched with some other person. Fig 3.7 Confusion Matrix. Ideally, the diagonal elements on the confusion matrix should have the maximum values and other elements should be zero. Practically, some of the non-diagonal elements are non-zero due to the presence of false recognitions as seen in the above matrix. This problem of some false recognition is taken care of by the threshold set for the final calculation of the output.



Fig 23 : Confusion Matrix

## V. CONCLUSION

- A system has been designed and implemented [3]. which uses face recognition to mark attendance of students present in the classroom.
- The system successfully returns a .csv file with numbers of students present which can easily be linked to the database.
- The current system is not feasible for large classrooms as the camera is not able to give good quality images of distant students.
- The designed system thus replaces the manual system with an automated system which is fast, efficient and time saving.
- 5) This system will save time, reduce the amount of work the administration has to do and will [7]. replace the stationery material with electronic apparatus.
- 6) Hence a system with expected results has been developed however It should be noted that the existing face recognition techniques are not one hundred percent (100%) efficient just yet. Typical efficiencies range between 40% to 60%.

#### **VI. REFERENCES**

- [1]. Sushma Jaiswal, Dr. (Smt.) Sarita Singh Bhadauria, Dr. Rakesh Singh Jadon, "Comparison between face Recognition Algorithm - Eigenfaces, Fisher Faces and Elastic Bunch Graph Matching", Volume 2, No. 7, July 2011 Journal of Global Research in Computer Science.
- [2]. Ravindra Pal Singh and Manish Dixit, "Histogram Equalization: A Strong Technique for Image Enhancement". International Journal of Signal Processing, Image Processing and Pattern Recognition. Vol.8, No.8(2015), pp.345-35
  - A. L. Rekha and H. K. Chethan, "Automated Attendance System using face Recognition through Video Surveillance," Int. J. Technol. Res. Eng., vol. 1, no. 11, pp. 1327–1330, 2014.
- [4]. Mathew A. Turk, Alex P. Pentland, "Face Recognition using Eigenfaces" Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference.
- [5]. S. Z. Li and A. K. Jain, Eds., Handbook of face recognition. New York: Springer,2005
- [6]. P. Viola and M. J. Jones, "Robust real-time face detection," Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, 2004.
  - 7]. Mithilesh Kumar and Ashima Rana, "Image Enhancement using Contrast Limited Adaptive Histogram Equalization and Wiener filter", International Journal of Engineering And Computer Science, Volume 5, Issues 6, June 2016, Page No. 16977-16979
- [8]. Face alignment Tutorial, https://github.com/nlhkh/face-alignment-dlib
- [9]. LBPH OpenCV Documentation, https://docs.opencv.org/2.4/modules/contrib/doc /facerec/facerec\_tutorial.html45
- [10]. Divyansh Methi, Abhishek Chauhan, Divyanshu Gupta, "Attendance System Using

Face Recognition" International Journal of Advanced Research in Science, Engineering and Technology Vol. 4, Issue 5, May 2017

- [11]. W. ZHAO, R. Chellappa, P.J. Phillips, A. Rosenfeld, "Face Recognition – A Literature Survey", ACM Computing Surveys, Vol. 35 Dec-2003.
- [12]. Samuel Lukas, Aditya Rama Mitra, Ririn Ikana Desanti, Dion Krisnadi, "Student attendance system in classroom using face recognition technique", Information and Communication Technology Convergence, (ICTC), 2016
- [13]. E.Varadharajan, R.Dharani, S.Jeevitha, B.Kavinmathi, S.Hemalatha, "Automatic Attendance Management System using Face Detection ", Green Engineering and Technologies (IC-GET).
- [14]. Andrew Wagner, Arvind Ganesh, Hossein Mobahi, "Toward a Practical Face Recognition System: Robust Alignment and Illumination by Sparse Representation", IEEE transactions on pattern analysis and machine intelligence, Vol. 34, No. 2, February 2012
- [15]. Anil K Jain, Lin Hong, Sharath Pankanti, and Ruud Bolle, "Biometric Identification", IEEE, 2004.

## Cite this article as :

Hari Purnapatre, Satvik Shukla, Shravan Gangishetti, Harshal Bohra, "Improved Facial Recognition for Attendance Systems", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 7 Issue 6, pp. 456-467, November-December 2020. Available at doi : https://doi.org/10.32628/IJSRST207678 Journal URL : http://ijsrst.com/IJSRST207678