

Intelligent Attendance Automation System by Face Recognition using PCA

Deven Mistry¹, Tejas Gupta²

¹(Computer Engineering, Thakur College of Engineering and Technology, India)

²(Computer Engineering, Thakur College of Engineering and Technology, India)

Abstract : Face being a multidimensional structure is a complex identity to a human. There is requisite of a highly efficient algorithm and technique to detect and recognize the faces. Our attempt on face recognition is done by Principal Component Analysis (PCA). Face will be categorized as familiar or unfamiliar by checking against the data set. If the face is unfamiliar then his/her face-based data will be stored in the database and on approval of the administrator, the data set for the student is created. The proposed system is an attendance system that enables the faculty to mark the attendance of the students in the lecture efficiently.

Keywords: Principle Component Analysis, Data Flow Diagram, Facial Recognition Technology, Haarcascade Classifier

I. Introduction

Whenever a lecture or a session commences, the professor or the teaching assistant always needs to spare time to record students' attendance. It is an onerous task and requires a lot of effort if the lecture has huge number of students. Also, the attendance sheet is subjected to damage and loss. At the end these attendance records are used by the staff to monitor each student's attendance rates. This banal practice is effective with a small number of students but extremely painstaking if there are a large number of students.

II. Motivation

The system aims at developing an intelligent system that consists of a camera, which will capture the image of the students in the class. Further the system is supposed to detect the human faces in the image, recognize the students registered and mark their attendance. Although there are challenges such as uneven illumination and changes in posing of the face for the system to work efficiently, detection and recognition of students will be enhanced over time as the system will be trained over and over again. The system will be based on the concept of Principle Component Analysis and Eigenface Approach of Face Recognition

III. Background Work

Proposed automated attendance system which can be achieved by the concept of Principle Component Analysis (PCA). In PCA, there are many algorithms/approaches such as Eigenfaces Approach, FERET, Fisher faces approach, Support Vector Machine, etc. These algorithms are available in packages such as OpenCV and TensorFlow. Challenges such as uneven illumination, noise detection and distant placing of subjects occur in the working of the system. PCA is considered to be one of the most used unsupervised algorithms and can be seen as the most popular dimensionality reduction algorithm. The goal of PCA is to identify and detect the correlation between variables. If there is a strong correlation between the variables and it's found then one could reduce the dimensionality.

- [1] Prof. V.P. Kshirsagar, M.R.Baviskar, M.E.Gaikwad in their paper proposed the use of Eigenfaces for feature extraction and feed forward back propagation neural network.
- [2] Ahmet Ozdhil, in his paper titled "A Survey on Comparison of Face Recognition Algorithms" provides a comparative study of many algorithms such as Eigenfaces, Fischerfaces and Local Binary Pattern Histogram methods. The paper also informs about the various platforms on which the various algorithms work differently.

IV. Proposed Work

In a given data set, we find the directions of maximum variance in high dimensional data and then we project it into a smaller dimensional subspace while retaining most of the information. The goal of PCA is to reduce the dimensions of a D dimensional dataset by projecting onto a k dimensional subspace where $k < D$.

Before marking the attendance on the proposed system, we need to prepare the data set (i.e. the data from which the system will recognize the faces). The data set can be prepared using the following steps:

1. Activate the camera and capture the image. The administrator has the option of registering the student under their observation and providing each one with their respective ID and input their names. During the lecture the administrator has to open the camera and capture the image.



Fig. 1 (Captured Image)

2. Extract frontal face: The frontal faces are extracted from the image with the help of the Haar Cascade Classifier.



Fig. 2 (Extracted Frontal Face)

The Haar Cascade Classifier is basically used for detecting objects from the source. The 'haarcascade_frontalface_default.xml' comes by default with the OpenCV package of python. The following classifier is already trained and is capable of recognizing and detecting various features of the face. The classifier works by passing a box over the entire image looking for features. (Refer Fig. 3 and Fig. 4). For improving the efficiency of this detection, one can even create their own classifier which would be able to do the desired result.



Line Features



Edge Features



Fig. 3

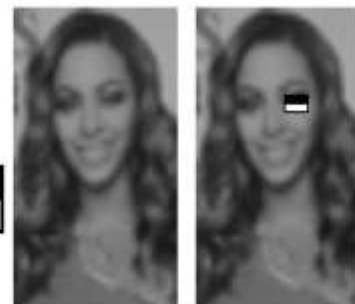


Fig. 4 (Feature Extraction)

- After the frontal faces are extracted, the system captures multiple images of the frontal face extracted and saves them in '. npy' format. The images captured are stored digitally in the form of arrays.

Storing the data in '. npy' format offers a lot of advantages. This file format makes incredibly fast reading speed enhancement over reading from plain text or CSV files. The '. npy' format is a standard binary file format for persisting a single arbitrary NumPy array on disk. The format stores all of the shape and data type information necessary to reconstruct the array correctly even on another machine with a different architecture. The format is designed to be as simple as possible while achieving its limited goals. For instance, 'deven.npy' consists:

```
934e 554d 5059 0100 7600 7b27 6465 7363
7227 3a20 277c 7531 272c 2027 666f 7274
7261 6e5f 6f72 6465 7227 3a20 4661 6c73
652c 2027 7368 6170 6527 3a20 2832 352c
98a5 a58a 9996 7280 7d5b 6467 3d41 441f
1f1f 1719 1914 1616 1214 1411 1313 1213
1314 1414 1313 1313 1313 1414 1415 1515
1717 171b 1b1b 1c1c 1f21 2228 2022 2a1f
232d 2023 3120 2433 1f24 321e 2435 1b22
351c 2235 1c21 321b 202f 1b21 3018 1d2c
.....
.....
.....
6366 5d60 645a 5d61 575d 5d55 5c5b 5559
5958 5a5a 686b 6b73 787b 747a 7c7a 7e80
```

For every individual the .npy file is created and the data regarding the shape and other necessary information while training is stored in the files. The format is as such that the images can be reconstructed using the data array correctly.

While in the front end a spreadsheet is created that stores the id and the name of the student as put at the time for training the system with the faces initially. An example of the spreadsheet is in Fig. 5.

	A	B
1	ID	Name
2		
3	CMPA1	Rajesh
4	CMPA2	Mukesh
5	CMPA3	Jay
6	CMPA4	Deven
7	CMPA5	Tejas
8	CMPA6	Aditya
9	CMPA7	Rishabh
10	CMPA8	Ashutosh
11	CMPA9	Rohit
12	CMPA10	Ved

Fig. 5 (Spreadsheet of Students)

Name	Type	Size
aditya.npy	NPY File	93 KB
ashutosh.npy	NPY File	39 KB
jay.npy	NPY File	261 KB
mukesh.npy	NPY File	89 KB
rajesh.npy	NPY File	145 KB
rishabh.npy	NPY File	117 KB
rohit.npy	NPY File	89 KB
tejas.npy	NPY File	45 KB
ved.npy	NPY File	48 KB

Fig. 6 (Data set of the Students)

After the data set is ready, the system is ready to mark the attendance. The data set is visible in the following format.



Fig. 7 (Image Captured by The Camera)



Fig. 8 (Extracted Frontal Faces from The Image Captured)

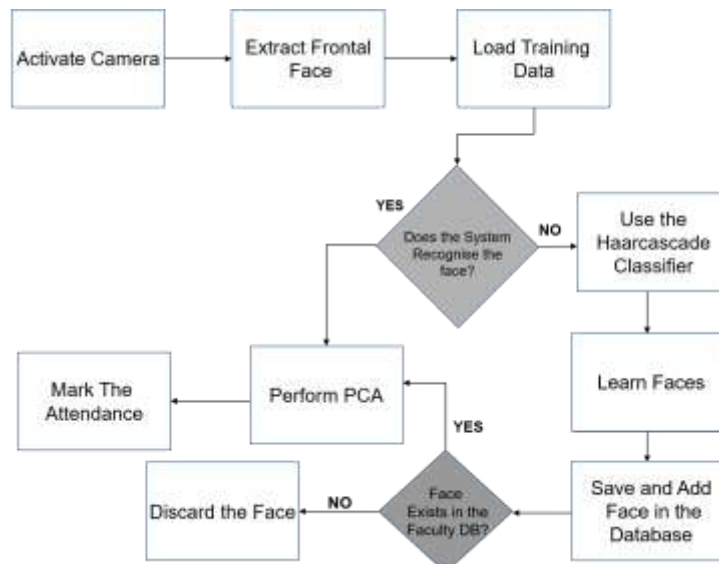


Fig. 9 (Flowchart for the System)

It works as follows:

We first activate the camera and the camera captures an image of the classroom.

1. Extract frontal face: The frontal faces for all the students in the view of the camera are extracted in the same manner as they were used to create the data set i.e. the .npy is created for all the faces using the haarcascade classifier.
2. Load training data: To be able to recognize the faces, Recognition: If the If the system is able to recognize the frontal face, it performs PCA and marks the attendance for the same; else, it uses the Haar Cascade classifier to learn the face and saves it in the database. PCA itself works in multiple stages. It fundamentally works on the principle of dimension reduction.
3. If the saved face does not exist in the faculty database, it is discarded, else, PCA is performed and the attendance is marked.

Working of PCA

PCA is one of most commonly used unsupervised learning algorithms which is used for feature extraction. PCA works by learning from the loaded database. This is done using the 'train_test_split()' function from the sklearn.model_selection package. The train_test_split function is used in order to make the split. It takes test_size as a parameter. If one enters, test_size=0.2 inside the function, it indicates the percentage of the data that should be held over for testing. It's usually around 80/20 or 70/30. At times we then need to fit the data on the training model. This is done using the various methods from the StandardScaler class from the sklearn.preprocessing package. We use two methods to fit the data model, fit_transform() and transform(). These methods are used to center the data (make it have zero mean and unit standard error), we subtract the mean and then divide the result by the standard deviation.

$$x' = \frac{x - \mu}{\sigma}$$

We do that on the training set of data. But then we have to apply the same transformation to the testing set (e.g. in cross-validation), or to newly obtained examples before forecast. But we have to use the same two parameters μ and σ (values) that we used for centering the training set.

Hence, every sklearn's transform's fit() just calculates the parameters (e.g. μ and σ in case of StandardScaler) and saves them as an internal objects state. Afterwards, we can call its transform() method to apply the transformation to a particular set of examples.

fit_transform() joins these two steps and is used for the initial fitting of parameters on the training set x , but it also returns a transformed x' . Internally, it just calls first fit() and then transform() on the same data.

After fitting the data, we can now apply PCA. The PCA class is readily available from the sklearn.decomposition package. Here the PCA function takes n_components (number of principle components) as a parameter.

The function internally uses SVD (Singular Value Decomposition) to project it to a lower dimensional space. Out of the multiple dimensions in the training data, SVD first finds the dimensions around which there is maximum variation. Then considering those dimensions SVD plots the data and finds the best fitting line. The best fitting line is determined by measuring the distances of points from the origin.

The measuring the distances, they are squared so that the negative values do not cancel out each other. Then we calculate the sum of these distances and choose the line for which the sum of squared distances is the maximum. When such a line is determined, that becomes our Principle Component 1 (PC1). After finding PC1, we find the unit vector along the direction of PC1. This unit vector along PC1 is known as the EigenVector for PC1, also the sum of the squared distances for PC1 is known as the Eigenvalue for PC1.

The square root of this Eigenvalue is called the Singular value for PC1. In 2-D, Principle Component 2 will simply be a line perpendicular to PC1. The unit vector along PC2 will be called the EigenVector for PC2. For the final plot, we rotate everything so that PC1 is horizontal, then using the projected points, we can find where the sample points go.

V. Expected Outcome

Once the data set is ready, the system is ready to function. The system is now capable of recognizing the faces and can be used to mark the attendance of the class.

The system is able to recognize faces from the data set and mark the attendance for the respective students.

	A	B	C
1	ID	Name	03-Feb-19
2			
3	CMPA1	Rajesh	Present
4	CMPA2	Mukesh	Absent
5	CMPA3	Jay	Absent
6	CMPA4	Deven	Present
7	CMPA5	Tejas	Present
8	CMPA6	Aditya	Present
9	CMPA7	Rishabh	Present
10	CMPA8	Ashutosh	Present
11	CMPA9	Rohit	Absent
12	CMPA10	Ved	Present

Fig. 5

VI. Future Scope

An acknowledgement section may be presented after the conclusion, if desired. At present the system performs its computations in 2 dimensions instead a 3-D depth sensing camera can be used to capture the data. Doing so, the system would be able to capture in an additional dimension which will help us overcome the existing drawback and even improve the efficiency of the system. Also, from the data set, images can be further used for sentiment analysis. Examining the facial expressions of the students we will be able to determine whether a student is able to understand the subject which is being taught or not. It will even help us to determine the attention span of a student.

VII. Conclusion

In this system we have implemented an attendance system for a lecture, section or laboratory by which lecturer or teaching assistant can record student's attendance. It saves time and effort, especially if it is a lecture with huge number of students. The complete system is implemented in Python and Django. This attendance system shows the use of facial recognition techniques for the purpose of student attendance and for the further process this record of student can be used in exam related issues.

References

Proceedings Papers:

- [1]. Prof. V.P. Kshirsagar, M.R.Baviskar, M.E.Gaikwad, "Face Recognition Using Eigenfaces" *2011 3rd International Conference on Computer Research and Development*
- [2]. Sang-Jean Lee, Sang-Bong Jung, Jang-Woo Kwon*, Seung-Hong Hong, "Face Detection and Recognition Using PCA", *Proceedings of IEEE. IEEE Region 10 Conference. TENCON 99. 'Multimedia Technology for Asia-Pacific Information Infrastructure' (Cat. No.99CH3703)*
- [3]. Shruti Sehgal, Harpreet Singh, Mohit Agarwal, V. Bhasker, Shantanu, "Data Analysis Using Principal Component Analysis" *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*
- [4]. M. Sharkas, M. Abou Elenien, "Eigenfaces vs. Fisherfaces vs. ICA for Face Recognition; A Comparative Study," *9th International Conference on Signal Processing, 2008, ICSP 2008., 2008, pp. 914–919*
- [5]. D. Huang , C. Shan , M. Ardabilian , Y. Wang and L. Chen "Local binary patterns and its application to facial image analysis: A survey", *IEEE Trans. Syst., Man, Cybern., C, vol. 41, pp.765 -781 2011*
- [6]. H. Yang and Y. Wang, "A LBP-based face recognition method with Hamming distance constraint," in *Proc. Int. Conf. Image Graph., Aug., 2007, pp. 645–649.*
- [7]. T. D. Russ, M. W. Koch, and C. Q. Little, "3D Facial Recognition: A Quantitative Analysis," *38th Annual 2004 International Carnahan Conference on Security Technology, 2004.*
- [8]. P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About," in *Proceedings of the IEEE, vol. 94, Issue 11, 2006.*