Implementing a Face Recognition System using Principal Component Analysis

Aniketh Sukhtankar, UF ID 7819-9584, Student, *EEL 6825 Pattern Recognition*

***Abstract*— Face recognition is the most useful techniques for security purposes in current century. It has many applications apart from its most important and obvious use in biometrics. It can be used for security in various offices and such small scale economical purposes. The major advantages of face recognition are that it is user friendly compared to various other security measures which have complex steps involved. Also, these days since everything is handled online and each task has a password to remember it becomes very tedious. So, face recognition is the best alternative because in that case you do have to memorize the passwords. If many methods of face recognition are combined then it becomes very efficient and powerful. So, the goal of this project is to use the existing methods of face recognition and combine them and propose a new algorithm such that it is economical and at the same time very powerful cos multiple methods are combined to form a single unified powerful algorithm.**

***Index Terms*—Real time, Face Recognition, PCA**

I. Introduction

Face recognition has become a very active part of the research in recent years because it has many advantages and is the most powerful security mechanism compared to its counterparts. The major reason why face recognition has gained so much momentum is because it is not easy to artificially fake it. Even though faces can be similar they will never have the same expressions and mannerisms. There are vast amounts of differences like skin color texture, birth marks, various other marks which are specific to an individual. So even if with plastic surgery you try to copy some individual's exact features, there will be many noticeable differences which would be caught by the face detector even if it cannot be seen with naked eyes. A face recognizer studies various expressions and attributes in minute details and so we cannot manipulate them. A generic illustration of a face recognition system is shown in Fig. 1.

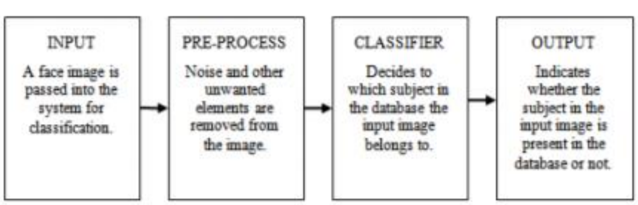


Fig. 1. Generic representation of a face recognition system

After years and years of research there is still no algorithm which is efficient accurate for the traditional conditions. Either it is not very efficient as we do not have a unified algorithm which has a mix of all the powerful face recognition features or then the algorithm is not economical. Due to these reasons face recognition cannot be used on small scale and for every simple application. But can be only used by the highly-funded forensics departments and the Federal Bureau of Investigation and if we consider the economical one like the one in our laptops that is used. Those algorithms are so weak that sometimes even your friends face it recognizes as the owner and thus poses a great security breach or then sometimes it does not recognize the owner and thus he must manually type the password and so it is not very sophisticated as proposed by the Face Recognition vender check (FRVT). Thus, there is a need for a face recognition algorithm which is efficient and economical. The algorithms using latest technology like machine learning are very efficient but their disadvantage is that the training and processing data takes up a lot of time and effort and thus can be used only for some applications sparingly.

Literature Review

A. *Architecture*

Traditional face recognition algorithm was mainly dependent on the facial features. What it did was from the image provided it found the relative positions of eyes, nose, ears, jawline and various other factors and made a template which it then compared with all the image to detect the face from amongst the set of images provided to it. This is the traditional approach and not very efficient because it only considers a small subset of the actual problem and leaves many other important factors behind which actually helps to make the algorithm more powerful by considering the minor details which cannot be tampered with at all and thus making the most efficient algorithm. Some of the popular recognition algorithms are described below.

Face recognition systems architecture broadly consists of the three following tasks:

1 Acquisition Phase which includes detection of face like images from an entire bunch of different types of images. So, it narrows down to all the face images first which is it broadly classifies the images.

2 Feature Extraction Phase this includes a much deeper

analysis where it aligns and segments the various features for a much better comparison.

3 Recognition Phase which is the main phase where based on all the features described and its accurate alignment it tries to figure out the exact face with the same alignment and relative positioning which is rare.

B. *Face Detection Approaches*

Some of the important face detection strategies are discussed here.

1) Knowledge based methods are the ones where data is obtained from researcher's data after studying human faces for years and years. But in this case the data is not reliable enough because it is very difficult to define rules. Expressing the data by reading the data and converting it into actual faces is very difficult since no set properties are defined.

2) Featured-based methods: To make the algorithm more efficient we define the other features of the face like texture, color to differentiate and make the image unique but this is very difficult to get the exact features in this case because the person must be looking fair in the picture than he actually is just because of a certain angle or just because there was light falling on him when the picture was click or because of the flash of the camera. So, we cannot reply on such factors to make the algorithm efficient and refined.

3) Template matching: This process is very straight forward but not efficient because in this case the image is compared with the template which is pre-made. But in this case the efficiency suffers because the algorithm is not dynamic enough to incorporate a few changes like the pose or form of the person or the angle and the expressions that change according to the mood. Also, the external differences because of change in hairstyle or little make up worn by women which is quite common.

4) Appearance-based method: All the disadvantages of the template matching algorithm are overcome in this one. In this case, due to statistical analysis and machine learning all the external added features are easily recognized and ignored while comparing the face to the template. Because in this case it does not search for the perfect matching but due to machine learning it has the capacity to incorporate a few changes and still recognize the face with minor external differences giving importance only to the most important aspects which can be used to recognize the face accurately to some extent.

C. *Face Recognition Approaches*

The LFA technique of recognition makes use of features like eyes, nose as described before so it is very efficient with local variations but does not work well with universal features. The model matching is another method which makes use of neural network learning and thus uses training data to understand the exact image and thus trains it to select the most appropriate match based on all the examples it has learnt from but in this case a wide variety of examples are required to make it work for real world scenarios.

II. Face Recognition Algorithms

A. Principal Component Analysis (PCA):

PCA is a technique which calculates the space between the features and reduces the dimensionality of it and then uses it to recognize the various faces but poor discriminating power among the category and enormous computation makes this technique less efficient. This limitation is overcome by Linear Discriminant Analysis which is one of the most dominant algorithms for feature selection in appearance based strategies. But initially LDA used PCA to use the best properties of both the features.

B. Support Vector Machine (SVM)

Support Vector Machines (SVM) is a highly useful technique in classification issues. After feature extraction, this algorithm becomes very efficient however it cannot be applied if the value of the feature is missing. The advantage Support Vector Machine Classifiers is that they can give better generalization performance as compared to the native neural networks.

C. Independent Component Analysis (ICA)

Independent component analysis (ICA) is a technique is used for multidimensional statistical information. If the image and all the features are a single entity then it becomes very difficult to accurately determine the face using face recognition. So, in this case all the features are treated as separate components and thus the separate components can be analyzed from all the dimensions and thus this gives more depth to the entire face recognition algorithm as separate components are completely analyzed individually and then connected to recognize the entire image.

D. Linear Discriminant Analysis (LDA):

The linear discriminant analysis (LDA) converts the data into low dimension space and thus information is well separated. Thus, in singular form which is not efficient. A nonlinear subspace is required which protects the native features but at the same time emphasizes discriminate data.

III. Proposed System

A. *Eigenfaces*

As a part of my project I have focused my attention on building a real-time face recognition system based on Principal Component Analysis. The major challenge for a face recognition system is extracting features in an efficient manner. The python script built utilizes the eigen face technique for information reduction for the pictures. Even a very small image stores a large amount of data in it. A technique should summarize the information contained in an image to effectively store data useful for facial recognition and get rid of all the irrelevant data. First, Base faces are constructed from the images supplied as part of the training set. When a new image is passed to the program for analysis, it is represented as a linear combination of these base faces. Every face that we need to classify is projected into face-space generated by the recognizer and then analyzed as a vector. The problem we face is that images in computers are represented using very high dimensions. A 2D x by y grayscale image constitutes a m = xy-dimensional vector area, thus a picture with say 40 x 40 pixels uses around 1600-dimensional image area already. However, all the dimensions stored are not equally helpful from a face recognition standpoint. What is most important to us are the components that are used for many of the data points. The Principal Component Analysis (PCA) technique takes a group of probably correlated variables and then convert it into a smaller set of unrelated variables. The basic idea behind this approach is that a dataset of high dimensions is usually represented by correlated variables and only a few important dimensions' matter. The PCA technique identifies these so called principal components and uses them for matching faces.

Algorithmic Description

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the vector generated with observation values $x_i \in R^d$.

First the mean is calculated using the formula

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Then the Covariance Matrix is calculated as follows

$$S = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T$$

Calculate the eigen-values λ_i as well as eigen-vectors v_i of S

$$Sv_i = \lambda_i v_i, i = 1, 2, \ldots, n$$

The eigen-vectors are then sorted in non-increasing order of their eigen-value. The k principal components are the eigen-vectors corresponding to the top k eigen-values.

The k principal components will be:

$$y = W^T(x - \mu)$$

where $W = (v_1, v_2, \ldots, v_k)$.

The reconstruction from the Principal Components is done as follows:

$$x = Wy + \mu$$

where $W = (v_1, v_2, \ldots, v_k)$.

The Eigen faces method performs face recognition in three steps. First all training samples generated are projected into the PCA subspace. Second the image to be analyzed is projected into the same. Third, the closest neighbor from projected training images similar to the provided projected query image is identified.

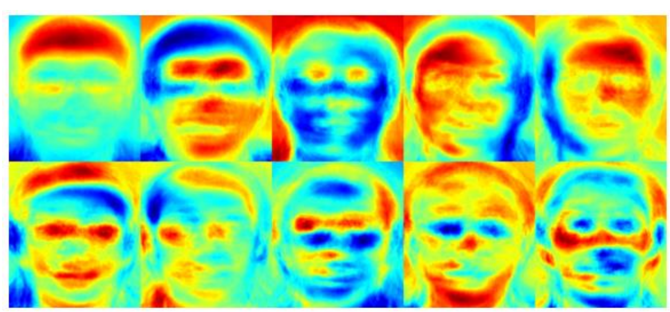
However, we still have to solve a problem. Imagine we are given 500 images sized 40 x 40 pixel. The Principal Component Analysis calculates the covariance matrix $S = XX^T$, where Size of (x) = 1600 x 500 in our example. You would end up with a 1600x 1600 matrix, which is tremendously high for a single image. In order to solve this problem, we make use of a concept in linear algebra, a P x Q matrix with P much greater than Q can only have Q-1 non-zero eigenvalues. So taking a Q x Q sized eigenvalue decomposition $S = X^TX$ is sufficient:

$$X^TXv_i = \lambda_i vi$$

and get the original value of the corresponding vectors of $S = XX^T$ with a simple left multiplication as follows:

$$XX^T(Xv_i) = \lambda_i(Xv_i)$$

The resulting eigenvectors are orthogonal. In addition to features illumination is also encoded as a part of these:



B. Fisher faces

The Principal Component Analysis (PCA), that forms the basis of the Eigenfaces technique discussed earlier, finds a linear combination of features that maximizes the entire variance in information. However, this does not take into account any categories and so plenty of data which might be otherwise useful may be lost when we get rid of elements that do not form a part of principal components. Imagine a scenario where an external light source is responsible for variance in your dataset. The so called Principal Components selected by PCA do not contain any discriminative data at all, therefore the projected samples are smeared together and a classification that makes use of these images becomes highly inefficient. Linear Discriminant Analysis that forms the bases of the Fisher face technique performs dimensionality reduction only for specific classes and was proposed by the great statistician Sir R. A. Fisher. It was first used to group flowers into categories in his 1936 paper the employment of multiple measurements in taxonomic issues [Fisher36]. Rather than maximizing the general the Linear Discriminant Analysis technique maximizes the quantitative relation of between-classes to within-classes scatter which effectively detects the mixture of features that separates best among distinct categories. This is based on the idea that same categories will cluster tightly along, whereas different categories are positioned as far apart within the lower-dimensional illustration. This was identified and applied to recognize faces.

Algorithmic Description

Let X be a randomly generated vector with samples drawn from each of the c categories:

$$\begin{aligned} X &= \{X_1, X_2, \ldots, X_c\} \\ X_i &= \{x_1, x_2, \ldots, x_n\} \end{aligned}$$

First, the scatter matrices Sb and Sw are computed using:

$$\begin{aligned} S_B &= \sum_{i=1}^{c} N_i (\mu_i - \mu)(\mu_i - \mu)^T \\ S_W &= \sum_{i=1}^{c} \sum_{x_j \in X_i} (x_j - \mu_i)(x_j - \mu_i)^T \end{aligned}$$

, where μ is the total mean given by the following formula:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

And μ_i is mean of a particular category $i \in \{1, \ldots, c\}$:

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j$$

The Fisher face algorithm next searches for a projection W, such that it maximizes the class separation criteria given by the following formula:

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$

Following [BHK97], a solution for this maximization problem is produced by solving the General Eigen-value Problem:

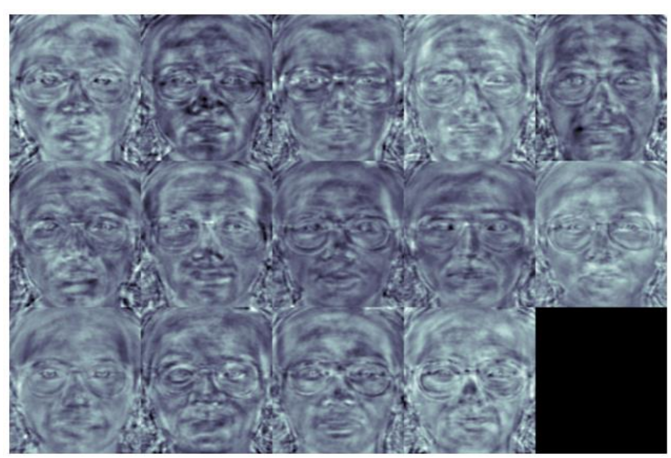
$$\begin{aligned} S_B \nu_i &= \lambda_i S_w \nu_i \\ S_W^{-1} S_B \nu_i &= \lambda_i \nu_i \end{aligned}$$

The maximization problem can then be rewritten as:

$$\begin{aligned} W_{pca} &= \arg\max_W |W^T S_T W| \\ W_{fld} &= \arg\max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \end{aligned}$$

Thus the transformation matrix W, that projects a sample into the $(c-1)$-dimensional space is given by the formula:

$$W = W_{fld}^T W_{pca}^T$$



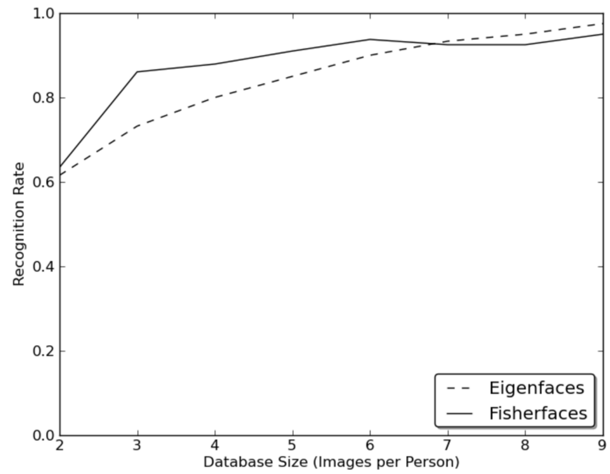
The Fisher faces technique learns a category-specific transformation matrix and so it does not capture illumination as clearly as the Eigenfaces technique. The Discriminant Analysis instead only focusses on the facial features for facial recognition. The performance of the Fisher face algorithm is also heavily dependent on the supplied input file. If the training set used by the Fisher face algorithm only consists of well-lit pictures and we then supply it pictures that are taken in badly-lit conditions, the algorithm will search out the incorrect components. The algorithm does this because it has no probability data from which to learn the illumination contained in the various images. Just like the Eigenfaces, the Fisher face algorithm allows a reconstruction of the projected image. However, since we only identify the facial features that are used to tell apart the subjects, the corresponding image reconstruction is not as good as Eigen face method. For the Fisher faces technique we'll project the sample image onto each one of the Fisher faces.

The differences are quite subtle for the naked eye to catch, however they become noticeable if you pay close attention:

C. Local Binary Patterns Histograms

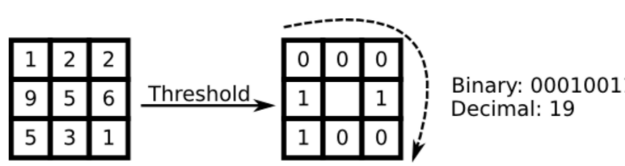
In the Eigen face and Fisher faces techniques the information is projected as a vector in an image space with large dimensions. The high-dimensionality of the image space can cause problems; therefore, we make use of a lower-dimensional subspace to preserve helpful data. The total scatter is maximized in the Eigen-face technique, which causes problems if an external source such as light is responsible for the variance generated, because components that have maximum variance over all categories do not help in classification and hence are not useful.

Therefore, Linear Discriminant Analysis which forms the basis of the Fisher faces technique was employed to preserve the discriminative data and optimize. The Fisher faces technique works great in constrained scenarios that gurantee perfect illumination conditions, but fails in real time applications where we just can't guarantee perfect illumination settings. It also fails if the training set contains only one image of the individual. The resulting covariance estimates for the subspace can be highly incorrect, and in turn the recognition rate suffers. Following is a graph that compares the recognition rates of the Eigenfaces and Fisher faces technique on the AT&T ORL Face database, which is not a very complex image database:



A minimum of 9+-1 pictures per individual are needed to give good recognition rates. Please refer [KM01] for a in depth analysis of both the techniques for smaller datasets. Both of these techniques worked by extracting only the important features from every image. The high dimensional vector representing the image is not completely examined. Instead only a set of native features come into play which implicitly results in a lower dimensionality subspace.

However image representation in a real-time scenario suffer not from illumination variations, expansion/shrinking, translation or rotation. The native features that are picked by the algorithms should be affective against such problems. The native Binary Patterns technique was initially used for texture analysis in two dimensions. The Local Binary Patterns technique works by summarizing the local structure in an image by examining every pixel with its neighboring pixels. The central pixel is compared against its neighbors. If the intensity of this pixel is less than its neighbor, it is denoted with a 0 and 1 if not. Every pixels can be represented as a binary range in this manner, similar to 10110010. Thus for every set of 9 pixels there are 8 neighbours and you can end up with 2 to the power 8 possible combinations, that are called local Binary Patterns and also known as LBP codes. the primary LBP operator actually used a strict 3 x 3 neighborhood that can be represented like this:



Algorithmic Description

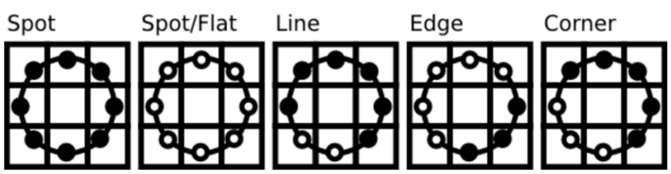
The LBP operator can be more formally defined as follows:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

, with (Xc, Yc) as pixel in the centre. The corresponding intensity is denoted by Ic and the intensity of all the neighboring pixels given by In. The Sign function is defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \quad (1)$$

This description allows us to capture even the extremely minute details in pictures. The authors were originally satisfied with state of the art results for texture classification. However, when the operator was revealed it was noted, that a set neighborhood fails to encode details that vary in scale. The operator was later extended to make use of a variable neighborhood in [AHP04]. An arbitrary range of neighbors is aligned on a circle with a changing radius, which enables to capture the resulting neighborhood pixels:



For a input Point (x_c, y_c) the position of the neighbor pixel $(x_p, y_p), p \in P$ can be calculated by the following formulas:

$$x_p = x_c + R\cos\left(\frac{2\pi p}{P}\right)$$

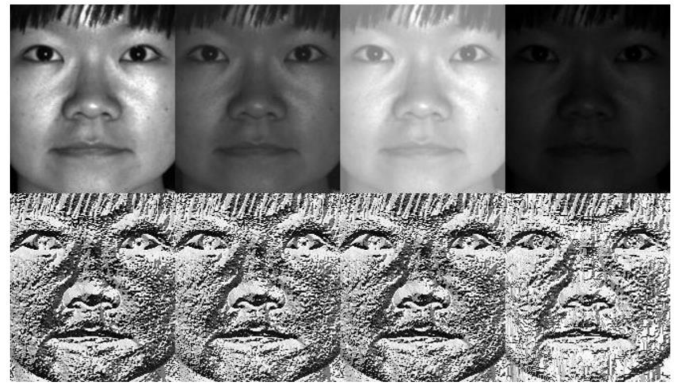
$$y_p = y_c - R\sin\left(\frac{2\pi p}{P}\right)$$

Where P is the total count of sample points and R is the radius of the circle.

This operator referred to as Extended LBP or Circular LBP because it is an extension to the original LBP codes. The point on the circle is interpolated if it doesn't correspond to image coordinates. There are many interpolation schemes to make use of, a bilinear interpolation is done in OpenCV:

$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}.$$

The LBP operator can work great against monotonic gray scale transformations. This can be verified by looking at the LBP picture of a modified picture given below:



So the next step is to incorporate the spatial information in the face recognition model. The technique used by Ahonen et. al [AHP04] works by dividing the image that is generated by LBP into N local regions. A histogram is then extracted from each of these regions. We then concatenate all the local histograms to obtain the spatially enhanced feature vector which is further used for face recognition. These histograms are called Local Binary Patterns Histograms (LPBH).

IV. Face Databases Used

A. The Database of Faces

The ORL database of Faces was utilized for this project. The image database contains a collection of face pictures of different taken over a span of 2 years beginning in 1992. The database consists of 10 different pictures of each of the 40 individuals. For some of the individuals, the photographs were taken during different times of the day, varying the illumination conditions as well as facial expressions (Subject is smiling or not) and facial details (Subject with glasses or not). All the photographs were taken against a dark background with the faces in an upright, frontal position (some degree of side movement was allowed). The image files are in PGM format. the dimensions of every image are 92x112 pixels, with 256 gray levels per pixel. the photographs are separated over forty directories for each individua. Each directory has name of the form sN, wherever N varies between 1 and 40. Each folder contains 10 corresponding to that individual. Each image has names of the form P.pgm, where P is a number between 1 and 10 and specifies image range

B. Yale Face Database A

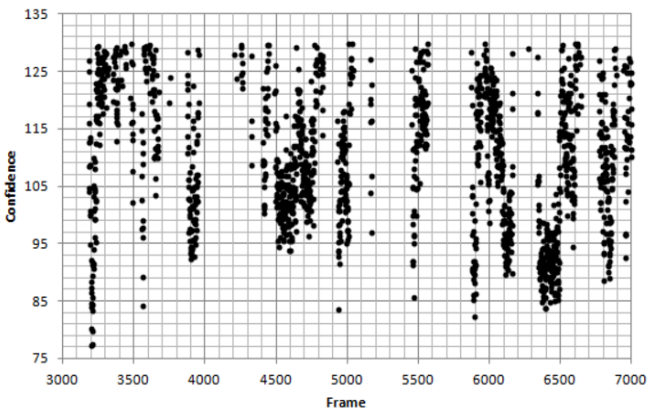
The Yale Face database A (size 6.4MB) contains 165 pictures of 15 individuals in grayscale GIF format. There are eleven pictures per subject. The database includes one image per different facial expression or configuration: center-light, left-light, right-light, wearing glasses, smiling, not wearing glasses, standard, surprised, sad face, sleepy face and winking.

C. Yale Face Database B

The extended Yale Face database B contains 16128 pictures of 28 individuals under 9 poses and 64 lighting conditions. All images in the database are in .pgm format and are manually aligned, cropped, and finally re-sized to a 168x192 dimension picture. Please refer to the homepage of the Yale Face database for additional elaborate data of the information format. Please refer the following papers, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose", "Acquiring Linear Subspaces for Face Recognition under Variable Lighting, PAMI, May 2005 [pdf]."

V. EVALUATION

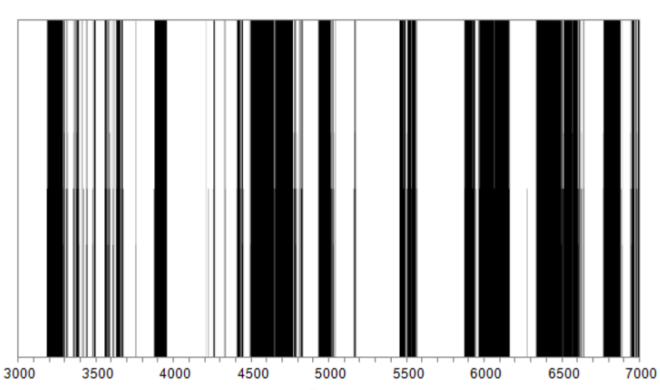
By using the Yale Face Database-A along with a few images added to training set, we obtain the confidence level in each frame of the real-time video input provided by the webcam:

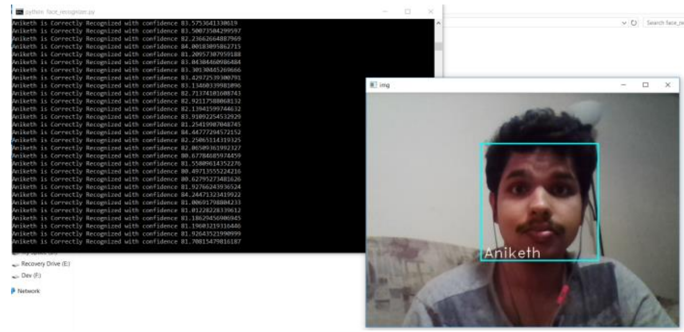


The results of running the codes on the Training Set using each of the 3 methods are described below in Table 1

Method Used	Mean Confidence	Max Confidence	Min Confidence
Eigen Faces	43.407	45.675	38.773
Fisher Faces	40.623	43.226	37.682
LPBH	41.700	48.897	32.124

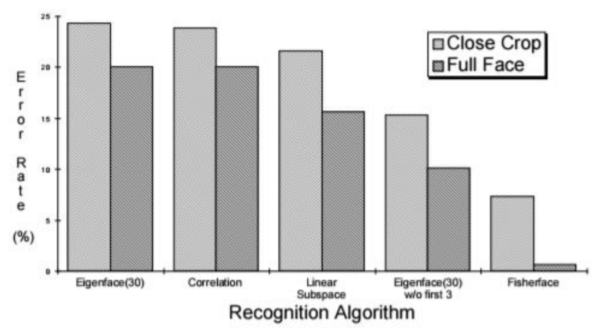
By setting some threshold value, we can obtain a binary determination (yes/no) of which frames contain the frontal view of my face:





Frame 4600

The relative performance of the algorithms can be seen in the figure below. Fisher face technique had error rates that were better than all other techniques. It seems that the Fisher face method performs better than others in varying illumination conditions, for different expression and regardless of whether the subject has glasses on or not. Linear subspace performs badly because of the variation in facial expressions due to which the pictures no longer lie in a linear subspace. The Fisher face algorithm ignores the less important parts of the image that might not be useful for facial recognition. The resulting projections, mask the areas of the face that vary a lot. On the contrary, the cheeks, nose and brow stay the same over within-class variations and are considered more important for recognition. We can conclude that Fisher face technique produces projection directions that are good for recognizing faces that lie outside the training set by reducing inter class scatter. Such that, once the projection directions are determined, everyone can be modeled by only one image.



All the algorithms performed better on full face images. Note that the Fisher face technique showed a good improvement by reducing error rate from 7.2% to 0.7% When training is done by making use of an image that captures the entire face, the pixel elements that correspond to the contour of the face are chosen as good features to discriminate among people. Practically though, it can be expected that the recognition rates would drop if background or hairstyle had varied from person to person and may have resulted in performance that would be lower than cropped set of images.

VI. RELATED WORK

A recently proposed facial recognition method makes use of PCA, LDA and neural network. The algorithm consists of four steps: a) Preprocessing b) Using PCA for Dimension reduction b) Using LDA for feature extraction and d) neural network for classification. Combination of LDA and PCA are used in the technique for improving the capability of LDA when dealing with small sample size. The algorithm makes use of a neural classifier to reduce number misclassification resulting due to non-linearly separable classes. The proposed technique has been tested on Yale face database. The results show that the algorithm displays less misclassification when compared to previous techniques.

VII. CONCLUSION

Implementing this project gave me an opportunity to study about many popular techniques used in the field of face recognition. The detailed literature survey provided me with the pros and cons of the existing recognition algorithms and the trade-off associated with them. We can see that a combination of two or more techniques can improve the accuracy of system greatly. Several conclusions can be put forward as a result off the experiments conducted on the implemented system

1. If an image that is like a preexisting image in the training set is provided, all methods perform well.

2. Removing initial 3 principal components improves the performance displayed by the Eigenface technique in different lighting conditions.

3. In the limit as a higher number of principal components are used in Eigen face technique, performance becomes similar to correlation. When the first 3 principal components are taken out the performance becomes better as dimensionality of the feature space increases. Note that the performance levels off at 45 components as found by Sirovitch and Kirby when making use of Eigenface technique.

4. Fisher face technique seems to be the best at handling varying expressions and lighting conditions. Linear Subspace suffers with changing facial expressions.

Many interesting questions remain unanswered: How well does Fisher face work with large datasets? Does performance deteriorate if some of the subjects are only observed under certain illumination coditions? How can information about class of faces be used to an advantage?

REFERENCES

[1] D. Kumar, C.S. Rai, and S. Kumar, “Face Recognition using Self Organizing Map and Principal Component Analysis” in Proc. on Neural Networks and Brain, ICNNB 2005.

[2] Kar, S., Hiremath, S., Joshi, D.G. and Chadda, V.K. and Bajpai, A. “A Multi-Algorithmic Face Recognition System”, International Conference on Advanced Computing and Communications, 2006. ADCOM 2006. pp 321 -326, 2006.

[3] Delac K., Grgic M., Grgic S., “Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set, International Journal of Imaging Systems and Technology, Vol. 15, Issue 5, 2006, pp. 252-260.

[4] The Extended Yale Face Database B http://vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html

[5] The AT& T Database of Faces formerly “The ORL Database of Faces” http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[6] S Nicolas Morizet, Frdric Amiel, Insaf Dris Hamed, Thomas Ea A Comparative Implementation of PCA Face Recognition Algorithm, ICECS’07

[7] M. Turk and A. Pentland, "Eigenfaces for recognition," J. Cognitive Neuroscience,vol. 3, 71-86., 1991.

[8] Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV 2004 (2004), 469–481.

[9] Belhumeur, P. N., Hespanha, J., and Kriegman, D. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 7 (1997), 711–720.

[10] Brunelli, R., and Poggio, T. Face recognition through geometrical features. In European Conference on Computer Vision (ECCV) (1992), pp. 792–800.

[11] Duda, R. O., Hart, P. E., and Stork, D. G. Pattern Classification (2nd Edition), 2 ed. November 2001

[12] Rodriguez, Y. Face Detection and Verification using Local Binary Patterns. PhD thesis, ´Ecole Polytechnique F´ed´erale De Lausanne, October 2006.

[13] Turk, M., and Pentland, A. Eigenfaces for recognition. Journal of Cognitive Neuroscience 3 (1991), 71–86.

[14] Zhao, W., Chellappa, R., Phillips, P., and Rosenfeld, A. Face recognition: A literature survey. Acm Computing Surveys (CSUR) 35, 4 (2003), 399–458.

[15] http://docs.opencv.org.html

[16] A Real-time Face Recognition system using PCA and various Distance Classifier – Deepesh Raj home.iitk.ac.in