



Face Photo-Sketch Synthesis and Recognition

Amit R.Sharma
Bharati Vidyapeeth Deemed
University,College Of
Engineering, Pune-46
Information Technology
Department

Prakash.R.Devale
Bharati Vidyapeeth Deemed
University,College Of
Engineering, Pune-46
Information Technology
Department

ABSTRACT

Today in Modern Society Face Recognition has gained much attention in the field of network multimedia access. After the 9/11 tragedy in India, the need for technologies for identification, detection and recognition of suspects has increased. One of the most common biometric recognition techniques is face recognition since face is the convenient way used by the people to identify each other. In this paper we are going to study a method for representing face which is based on the features which uses geometric relationship among the facial features like mouth, nose and eyes .Feature based face representation is done by independently matching templates of three facial regions i.e eyes, mouth and nose .Principal Component Analysis method which is also called Eigen faces is appearance based technique used widely for the dimensionality reduction and recorded a greater performance in face recognition. Here we are going to study about PCA followed by Feed Forward Neural Network called PCA-NN.

Keywords

Face Recognition ,PCA, Neural Network.

1. INTRODUCTION

Face is a very important part of the human body through which and individual can be identified. The Face is a primary focus in the society and it plays a major role in conveying identity and emotions of an individual. Other than identical twins every individual has unique facial features. Facial recognition is a form of computer vision which uses human faces to attempt to identify an individual or verify a person's claimed identity.

Face recognition has become an important issue in many applications such as security systems, credit card verification, criminal identification etc. Even the ability to merely detect faces, as opposed to recognizing them, can be important.

Although it is clear that people are good at face recognition, it is not at all obvious how faces are encoded or decoded by a human brain. Human face recognition has been studied for more than twenty years. Developing a computational model of face recognition is quite difficult, because faces are complex, multi-dimensional visual stimuli. Therefore, face recognition is a very high level computer vision task, in which many early vision techniques can be involved. In Face recognition, there are two techniques-verification and identification. During verification the system compares the individual with who they say they are and gives the decision yes or no. and in Verification the system compares the individual with all other individual present in the database and gives the ranked list matches.

2. LITERATURE ANALYSIS

In psychology study, researchers have long been using various face drawings, especially line drawings of faces, to investigate face recognition by the human visual system [10], [11], [12], [13], [14]. Human beings can recognize caricatures quite well, which is a special kind of line drawings of faces, with particular details of a face accentuated, compared with the ability to recognize face photos. Presumably, the details which get accentuated in caricaturing are those which are characteristics of that individual. Someone even question whether caricatures are in any way better representations than natural images, since caricatures may contain not only the essential minimum of information but also some kind of “superfidelity” due to the accentuated structures [10]. It is also shown that computer-drawn “cartoons” with edges, pigmentation, and shading of the original image can be well recognized by human beings. Some computer-based sketch synthesis systems have been proposed in recent years. Most of them have the linedrawing output without much sketch texture which is useful to convey 3D shading information. In [8], [9], face shape was extracted from a photo and exaggerated by some rules to make the result more similar to a sketch in shape. They were not based on learning. Freeman et al. [15] proposed an example-based system which translated a line drawing into different styles. Chen et al. [16] proposed an example-based face cartoon generation system. It was also limited to the line drawings and required the perfect match between photos and line drawings in shape. These systems relied on the extraction of face shape using face alignment algorithms such as Active Appearance Model (AAM) [17]. These line drawings are less expressive than the sketches with shading texture. In this paper, we work on sketches with shading texture.



Fig. 1. Examples of a face photo and a sketch.



It requires modeling both face shape and texture. There was only limited research work on face sketch recognition because this problem is more difficult than photo-based face recognition and no large face sketch database is available for experimental study. Methods directly using traditional photo-based face recognition techniques such as the eigenface method [1] and the elastic graph matching method [2] were tested on two very small sketch data sets with only 7 and 13 sketches, respectively. A face sketch synthesis and recognition system using eigentransformation was proposed. It was not limited to line drawing and could synthesize sketches with more texture. The transformation was directly applied to the whole face image. In [4], it was shown that a synthesized sketch by eigentransformation would be a good approximation to a sketch drawn by an artist only if two conditions are satisfied: 1) A face photo can be well reconstructed by PCA from training samples and 2) the photo-sketch transformation procedure can be approximated as linear. In some cases, especially when the hair region is included, these conditions are hard to be satisfied. Human hair varies greatly over different people and cannot be well reconstructed by PCA from training samples. PCA and Bayesian classifiers were used to match the sketches drawn by the artist with the pseudosketches synthesized from photos. Liu et al. [5] proposed a nonlinear face sketch synthesis and recognition method. It followed the similar framework as in [3], [4]. However, it did eigentransformation on local patches instead of the global face images. It used a kernel-based nonlinear LDA classifier for recognition. The drawback of this approach is that the local patches are synthesized independently at a fixed scale and face structures in large scale, especially the face shape, cannot be well learned. Zhong et al. [6] and Gao et al. [7] proposed an approach using an embedded hidden Markov model and a selective ensemble strategy to synthesize sketches from photos. The transformation was also applied to the whole face images and the hair region was excluded.

2. PROPOSED SYSTEM

We develop a new approach to synthesize local face structures at different scales using a Markov Random Fields model. It requires a training set containing photo-sketch pairs. We assume that faces to be studied are in a frontal pose, with normal lighting and neutral expression, and have no occlusions. Instead of directly learning the global face structure, which might be too complicated to estimate, we target at local patches, which are much simpler in structure. The face region is divided into overlapping patches. During sketch synthesis, for a photo patch from the face to be synthesized, we find a similar photo patch from the training set and use its corresponding sketch patch in the training set to estimate the sketch patch to be synthesized. The underlying assumption is that, if two photo patches are similar, their sketch patches should also be similar. In addition, we have a smoothness requirement that neighboring patches on a synthesized sketch should match well. The size of patches decides the scales of the face structures which can be learned. We use a multiscale Markov Random Fields model to learn face structures at different scales. Thus, local patches in different regions and scales are learned jointly instead of independently as in [5]. This approach can also be used to synthesize face photos given sketches. Our sketch/photo algorithm is relevant to [18], which used MRF to estimate scenes, such as motion and range map, from images. During the face sketch recognition stage, there are two options to reduce the modality difference between photos and sketches: 1) All of the face photos in the gallery are first transformed to sketches using the sketch synthesis algorithm and a query sketch is matched with the synthesized sketches, and 2) a query sketch is

transformed to a photo and the synthesized photo is matched with real photos in the gallery. We will evaluate both options in Section 3. After the photos and sketches are transformed into the same modality, in principle, most of the proposed face photo recognition approaches can be applied to face sketch recognition in a straightforward way. In this paper, we will evaluate the performance of several appearance-based face recognition approaches. Biometrics is a form of bioinformatics that uses biological properties to identify individuals. Examples of biometrics are fingerprinting, facial recognition, iris scanning, signature authentication, and voice recognition and hand geometry. Facial recognition is simply using characteristics of the face to identify an individual.

There are several practical reasons for favoring facial recognition over other biometrics for the purposes of identification. Since the biometric data can be captured at a distance, it does not require active participation on the part of the subject - the individual need not pose, push a button or click a mouse to activate a system, stare into a lens or press an ink pad. Facial Recognition is unobtrusive and discrete. The infrastructure for its implementation is already widespread and inexpensive. Security cameras are common in airports, ATM machines, or at any location a business owner, governmental agency or private homeowner may choose to keep secure. Every government agency and many private companies keep photo ID records. In addition, intelligence agencies have massive surveillance databases of images and video. There is a growing need to match these legacy photos to live individuals: for example, to check that someone is authorized to enter a building that the user of a debit card is the owner of that card, or that someone entering the country does not match a photo of a known or suspected terrorist. Unfortunately, current facial recognition technology suffers from failure rates too high to be implemented more pervasively. Most facial recognition applications today use 2-dimensional technology, which measures height, width and distance between feature points to make identification. This technique introduces a fundamental flaw since faces are 3-dimensional, with irregularly shaped features - noses, lips, ears, hair - that change in appearance as the face turns. Faces also reflect light and produce shadows, essentially creating new and different images. With 2-dimensional technology, failure rates rise with changes in pose or expression or variable lighting. In 2002 the US Government's Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST) conducted a Facial Recognition Vendor Test. Vendors with facial recognition applications competed with their individual technologies. The failure level was disappointing and unacceptable. The conclusion of the test was that 3-dimensional technology is the only hope to significantly improve the performance of existing facial recognition solutions. A metrics is the first company to solve critical problems associated with facial recognition. Our software, which is based on new 3-dimensional technology, addresses variations in (1) pose, (2) lighting and (3) facial expressions. It can be used with any camera, making it cheaper to implement than 3D solutions that require specialized cameras. Our software is fast, and it is approaching real-time speed.

Biometrics is the science of using biological properties to identify individuals. Examples of biometrics include fingerprinting, facial recognition, iris scans, signature authentication, voice recognition and hand geometry. Facial recognition has obvious advantages for personal identification:



unlike other biometrics, it is non-invasive, unobtrusive and discrete.

At present the demand for Facial Recognition technologies are fueled by the needs of Homeland Security. With improvements in reliability, future applications will come increasingly from the commercial sector. Applications for Facial Recognition are varied and vast:

- a) Document control (digital chip in passports, drivers' licenses)
- b) Transactional authentication (credit cards, ATMs, point-of-sale);
- c) Computer security (user access verification);
- d) Physical access control (smart doors);
- e) Voter registration (election accuracy);
- f) Time and attendance (entry and exit verification);
- g) Computer games (a virtual "you" plays against virtual opponents)

Technology can be inserted into all of these applications. Clearly this is an expanding market, which will remain strong even in economic downturns.

3. GENERAL FRAMEWORK OF FACE RECOGNITION SYSTEM

Face recognition software is based on the ability to recognize a face and then measure the various features of the face.

Every face has numerous nodal points that make up facial features. Some of the features measured by the software are

1. Distance between the two eyes
2. The shape of the cheek bones
3. Eye socket's depth
4. Jaw line's length
5. Nose width

These nodal points are measured creating a numerical code called Face Printed is stored in the data base as a face representation.

Regardless of any specific method, face recognition comprises of five steps

1. Acquiring the image of individual face using digital scan of a photograph or live picture of the individual.
2. Location the face in the image
3. Analysis of facial image according to nodal points and creating a face print
4. Comparison of face print obtained with all the face prints present in database
5. Declaration of match or no match

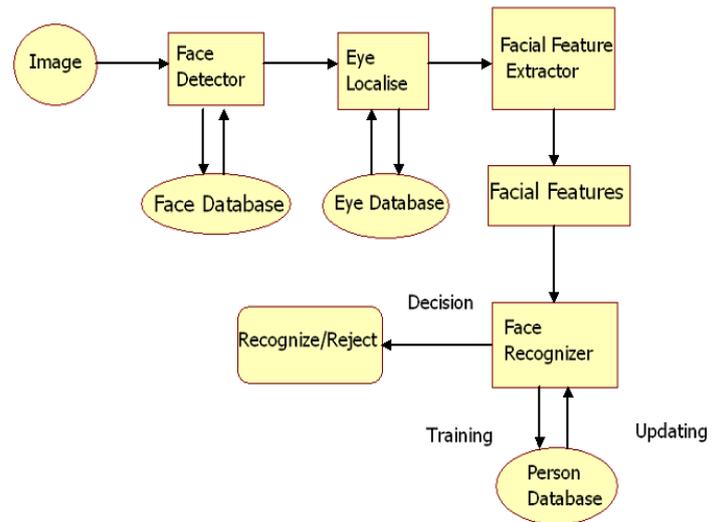


Fig1: General Framework of Face recognition System

We develop a new approach to synthesize local face structures at different scales using a Markov Random Fields model. It requires a training set containing photo-sketch pairs.

4. SYSTEM BLOCK DIAGRAM

System Block Diagram for face recognition using PCA is as shown in fig. In to this architecture we convert data base photo in to sketch by using PCA algorithm (by using above steps). Then this sketch is converted into train sketch. The Train sketch is then compare with sketch drawn by the Artist. Then find out the maximum match that will be our result.

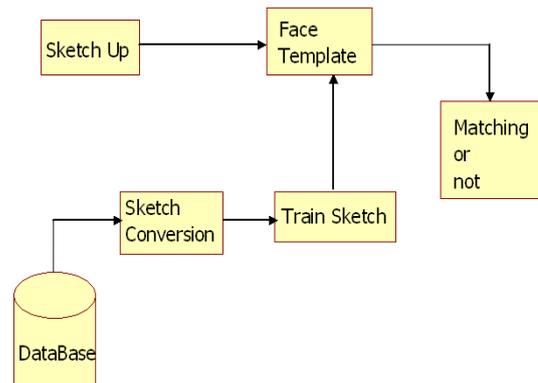


Fig2: System block diagram



5. PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis is a technique used for pattern recognition and signal processing in data reduction and feature extraction. The main aim of PCA is to find set of orthogonal components to minimize error in the reconstructed data. The pattern often contains redundant information, hence it has to be mapped to feature vector to get rid of redundancy and yet preserve the intrinsic features of the pattern.

A 2-dimension face image with size $N \times N$ can also be considered as one dimensional vector of dimension N^2 . An ensemble of images maps to a collection of points in this huge space. Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principle component is to find the vectors that best account for the distribution of face images within the entire image space. These vectors define the subspace of face images, which we call “face space”. Each of these vectors is of length N^2 , describes an $N \times N$ image, and is a linear combination of the original face images.

According to mathematics, eigenfaces are the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images. To represent the different variations in faces we use Eigen vectors. Each face can be represented exactly by a linear combination of the eigenfaces. It can also be approximated using only the “best” eigenvectors with the largest eigenvalues. The best M eigenfaces construct an M dimensional space, i.e., the “face space”.

Here are the steps to perform Principal Component Analysis.

Step 1: Get some data.

Here we are considering a simple data set of 2 dimensions so that PCA at step can be viewed easily ‘

Data Set =

X	2.5	0.5	2.2	1.9	3.1	2.3	2	1
y	2.4	0.7	2.9	2.2	3	2.7	1.6	1.1

Data Adjust =

X	69	-1.31	0.39	0.09	1.29	0.49	0.19	-0.81
y	0.49	-1.21	0.99	0.29	1.09	0.79	-0.31	-0.81

Step 2: Subtract the mean

For PCA to work properly, you have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. So, all the x values have \bar{x} (the mean of the values of all the data points) subtracted, and all the y values have \bar{y} subtracted from them. This produces a data set whose mean is zero.

Step 3: Calculate the covariance matrix

Since the data is 2 dimensional, the covariance matrix will be 2×2

$$C_{CV} = \begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555555 \end{pmatrix}$$

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix Since the covariance matrix is square, we can calculate the eigenvectors and eigenvalues for this matrix. These are rather important, as they tell us useful information about our data.

$$\text{Eigenvalues} = \begin{pmatrix} 0.049083398 \\ 1.28402771 \end{pmatrix}$$

$$\text{Eigen Vectors} = \begin{pmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{pmatrix}$$

Step 5: Choosing components and forming a feature vector Here is where the notion of data compression and reduced dimensionality comes into it. If we look at the eigenvectors and eigenvalues from the previous section, we notice that the eigenvalues are quite different values. In fact, it turns out that the eigenvector with the highest eigenvalue is the principle component of the data set. In our example, the eigenvector with the largest eigenvalue was the one that pointed down the middle of the data. It is the most significant relationship between the data dimensions. In general, once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives us the components in order of significance. Now, if you like, you can decide to ignore the components of lesser significance. We do lose some information, but if the eigenvalues are small, we



don't lose much. If we leave out some components, the final data set will have less dimensions than the original. To be precise, if we originally have n dimensions in our data, and so we calculate n eigenvectors and eigenvalues, and then we choose only the first k eigenvectors, then the final data set has only k dimensions. What needs to be done now is we need to form a feature vector, which is just a fancy name for a matrix of vectors. This is constructed by taking the eigenvectors that we want to keep from the list of eigenvectors, and forming a matrix with these eigenvectors in the columns.

$$\text{FeatureVector} = (eg_{i_1}, eg_{i_2}, \dots, eg_{i_n})$$

Given our example set of data, and the fact that we have 2 eigenvectors, we have two choices. We can either form a - feature vector with both of the eigenvectors:

$$\begin{pmatrix} .677873399 & -.735178656 \\ -.735178656 & .677873399 \end{pmatrix}$$

or, we can choose to leave out the smaller, less significant component and only have a

singlecolumn:

$$\begin{pmatrix} .677873399 \\ -.735178656 \end{pmatrix}$$

We shall see the result of each of these in the next section.

Step 6: Deriving the new data set: This the final step in PCA, and is also the easiest. Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the original data set, transposed.

$$\text{FinalDataSet} = (\text{RowFeatureVector} \times \text{RowDataAdjust})$$

Where RowFeatureVector is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and RowDataAdjust is the mean-adjusted data transposed, ie. the data items are in each column, with each row holding a separate dimension. If we take the transpose of the feature vector and the data first, rather than having a little T symbol above their names from now on. FinalDataSet is the final data set, with data items in columns, and dimensions along rows. What will this give us? It will give us the original data solely in terms of the vectors we chose. Our original data set had two axes x and y , so our data was in terms of them. It is possible to express data in terms of any two axes that you like. If these axes are perpendicular, then the expression is the most efficient. This was why it was important that eigenvectors are always perpendicular to each other. We have changed our data from

being in terms of the axes x and y , and now they are in terms of our 2 eigenvectors. In the case of when the new data set has reduced dimensionality, ie. We have left some of the eigenvectors out, the new data is only in terms of the vectors that we decided to keep. To show this on our data, we have done the final transformation with each of the possible feature vectors. We have taken the transpose of the result in each case to bring the data back to the nice table-like format. we have also plotted the final points to show how they relate to the components.

If we compare this data set with the one resulting from using both eigenvectors, you will notice that this data set is exactly the first column of the other. So, if we were to plot this data, it would be 1 dimensional, and would be points on a line in exactly the x positions of the points in the plot We have effectively thrown away the whole other axis, which is the other eigenvector. So what have we done here? Basically we have transformed our data so that is expressed in terms of the patterns between them, where the patterns are the lines that most closely describe the relationships between the data. This is helpful because we have now classified our data point as a combination of the contributions from each of those lines. Initially we had the simple x and y axes. This is fine, but the x and y values of each data point don't really tell us exactly how that point relates to the rest of the data. Now, the values of the data points tell us exactly where (ie. above/below) the trend lines the data point sits. In the case of the transformation using both eigenvectors, we have simply altered the data so that it is in terms of those eigenvectors instead of the usual axes. But the single-eigenvector decomposition has removed the contribution due to the smaller eigenvector and left us with data that is only in terms of the other.

6. FACE RECOGNITION USING PCA

Now let us see how PCA algorithm works. Here faces from database is selected one by one .then face image is removed from photo excluding background detail. Then face image is converted in to sketch, by using PCA sketch is then converted in to Eigen Face by considering the intensity of image. In this way average of all Eigen face is find out. Now convert artist sketch in to eigen face & mix it with average of all faces from databases. Now find out the maximum match of two faces which will be the output . Here sketch is converted in to eigen face which is then compare with the sketch which is generated from photo from the criminal record of police. The face region is divided into overlapping patches. During sketch synthesis, for a photo patch from the face to be synthesized, we find a similar photo patch from the training set and use its corresponding sketch patch in the training set to estimate the sketch patch to be synthesized. underlying assumption is that, if two photo patches are similar, their sketch patches should also be similar.

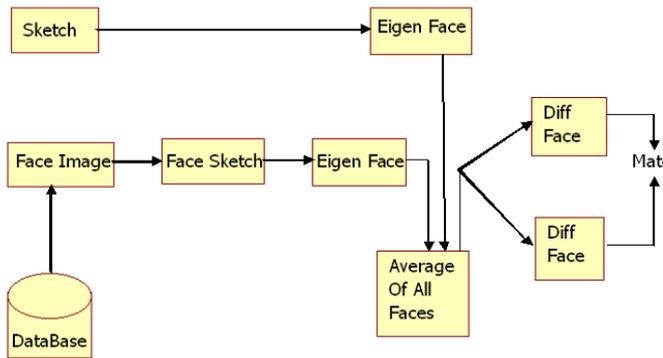


Fig3: Face recognition with eigenface.

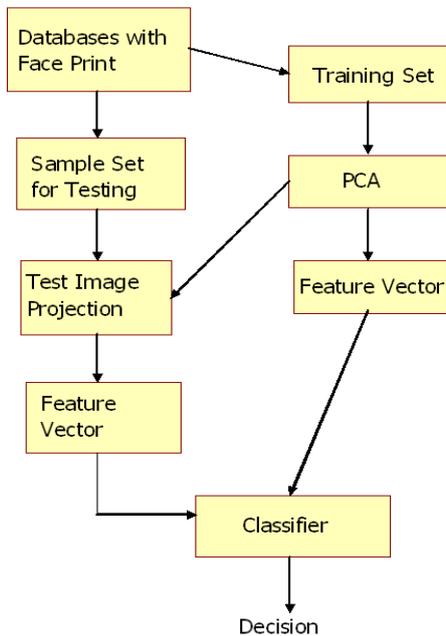


Fig4: PCA Approach for Face Recognition

7. NEURAL NETWORKS

Neural networks importance has increased due to massive parallelism in its structure .It has a high computation rates and it provides a great alternative with other conventional classifiers and decision making systems.

A successful face recognition method depends heavily on the choice of the features used by the pattern classifier. The back propagation is the best known and widely used learning algorithm in training multilayer perceptions(MLP).The MLP refer to the network consisting of a set of sensory units that constitute the input layer ,one or more hidden layers of computation nodes and an output layer of computation nodes. The input signal propagates through the network in a forward direction from left to right and layer by layer. For even an

image of moderate size however the network can be very complex and difficult to train.

A Feed forward Neural Networks is suitable structure for nonlinear separable input data .In FFNN model the neurons are organized in the form of layers. The neuron n one layer gets the input from the previous layer and feed their output to the next layer. In this type the neurons in the same or previous layers are not permitted for network connections.

Another example for neuron classifier is Probabilistic Decision-based Neural Network(PDNN).It does not have fully connected network topology ,instead it divides the network into K subnets Each subnet is dedicated to recognize one person in the database.PDNN uses Gaussian activation function for its neurons and the output of each subnet is the weighted summation of the neuron output.

Training and Testing of Neural Networks

The training is performed by n poses from each subject and the performance testing is performed by 10n poses of same subject. After calculation the Eigen Faces using PCA the projection vectors are calculated for the training set and then used to train the neural networks. This architecture is called PCA-NN.

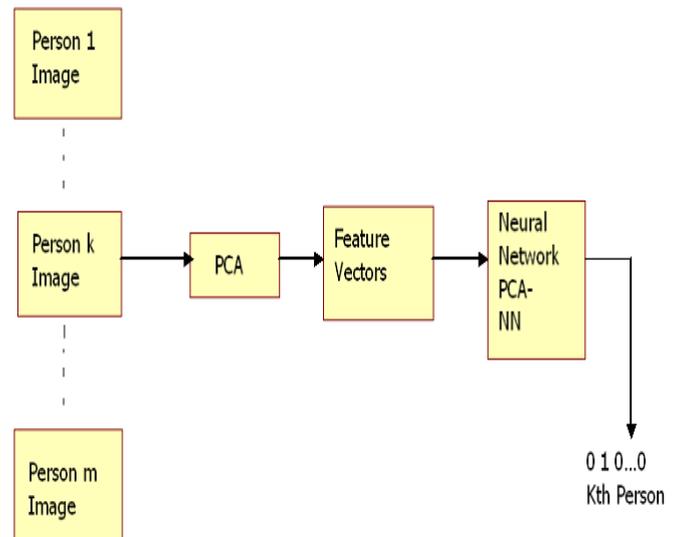


Fig5: Training Phase of Neural Networks

During the face sketch recognition stage, there are two options to reduce the modality difference between photos and sketches:

- 1) All of the face photos in the gallery are first transformed to sketches using the sketch synthesis algorithm and a query sketch is matched with the synthesized sketches, and
- 2) a query sketch is transformed to a photo and the synthesized photo is matched with real photos in the gallery.



8. CONCLUSION

Face Recognition Is a challenging as well as important recognition technique. It has great importance due to its friendliness. In this paper we have given an introductory survey for the face recognition technology, discussed about generic framework for face recognition. We have discussed about steps for PCA algorithm .We have covered basics of neural networks used in face recognition. We hope this paper can provide the readers a better understanding about face recognition. Face recognition can be applied in various fields like passport recognition, criminals list verification etc.

9. REFERENCES

- [1] “A tutorial on Principal Components Analysis” Lindsay I Smith , February 26, 2002
- [2] K. Sung and T. Poggio, Example-based Learning for View-based Human Face Detection, A.I. Memo 1521, MIT A.I. Laboratory, 1994.
- [3] H.A. Rowley, S. Baluja, and T. Kanade, Neural Network Based Face Detection, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 20, 1998.
- [4] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, Face Recognition: A Literature Survey, UMD CFAR Technical Report CAR-TR-948, 2000.
- [5] M. Turk and A. Pentland, Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, Vol.3, pp. 72-86, 1991.
- [6] W. Zhao, Robust Image Based 3D Face Recognition, PhD thesis, University of Maryland, 1999.
- [7] K.S. Huang and M.M. Trivedi, Streaming Face Recognition using Multicamera Video Arrays, *16th International Conference on Pattern Recognition (ICPR)*. August 11-15, 2002.
- [8] P.J. Phillips, P. Rauss, and S. Der, FERET (Face Recognition Technology) Recognition Algorithm Development and Test Report, *Technical Report ARL-TR 995*, U.S. Army Research Laboratory.
- [9] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, XM2VTSDB: The Extended M2VTS Database, in *Proceedings, International Conference on Audio and Video-based Person Authentication*, pp. 72-77, 1999.
- [10] Face Photo-Sketch Synthesis and Recognition Xiaogang Wang and Xiaoou Tang, Fellow, *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 11, November 2009
- [11] PCA and LDA based Neural Networks for Human Face Recognition Alaa Eleyan and Hasan Demirel *Eastern Mediterranean University Northern Cyprus* June 2007
- [12] Face Recognition using Neural Networks P.Latha Dr.L.Ganesan, Dr.S.Annadurai *Signal Processing: An International Journal (SPIJ) Volume (3) : Issue (5)*
- [13] An Introduction to Face Recognition Technology Shang-Hung Lin, Ph.D. *IC M*, 2000
- [14] Belhumeur, P.; Hespanha, J. & Kriegman, D. (1997). Eigen faces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, (July 1997) 711-720, 0162-8828
- [15] Brunelli, R. & Poggio, T. (1993). Face recognition: Features versus Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15 No. 10, (October 1993) 1042-1052, 0162-8828
- [16] Computational and performance aspect of PCA based face recognition algorithm. Hyeonjoon moon New York. & P Jonathon Phillips USA 16 March 2000.