

Face Recognition Using SOM Based Neural Networks and PCA Algorithm

M.Vasavi¹, B.SriSowmya², B.Prudhvi³, Dr.S.Sridhar⁴

¹(ECE, LIET/ JNTUK, India)

²(ECE, LIET/ JNTUK, India)

³(ECE, LIET/ JNTUK, India)

Abstract: In recent networked world, the need to maintain the security of information or physical property is becoming both increasingly important and increasingly difficult. Face recognition is one of the few biometric methods, which is very complicated system since the human faces change depending on their age, expressions etc. A human being has lots of expressions. So it is not possible to learn all types of expressions into the network. As a result, there is a solution for the unrecognized. Moreover due to the performance variations of the input device, face cannot be detected correctly and pattern may change extremely. In this paper we have developed and illustrated a recognition system for human faces using Kohonen self-organizing map (SOM) and PCA algorithm. The main objective of our face recognition system was to obtain a model that is easy to learn i.e. minimization of learning time, react well with different facial expressions with noisy input and optimize the recognition as possible.

Keywords: ANN (Artificial Neural Networks), PCA (Principal Component Analysis), SOM(Self Organizing Mapping).

I. Introduction

Biometric-based techniques have emerged as the most promising option for recognizing individuals in recent years since, instead of authenticating people and granting them access to physical and virtual domains based on passwords, PINs, smart cards, plastic cards, tokens, keys and so forth, these methods examine an individual's physiological and/or behavioural characteristics in order to determine and/or ascertain his identity. Passwords and PINs are hard to remember and can be stolen or guessed; cards, tokens, keys and the like can be misplaced, forgotten, purloined or duplicated; magnetic cards can become corrupted and unreadable. However, an individual's biological traits cannot be misplaced, forgotten, stolen or forged. Biometric-based technologies include identification based on physiological characteristics (such as face, fingerprints, finger geometry, hand geometry, hand veins, palm, iris, retina, ear and voice) and behavioral traits (such as gait, signature and keystroke dynamics). Face recognition appears to offer several advantages over other biometric methods, a few of which are outlined here: Almost all these technologies require some voluntary action by the user, i.e., the user needs to place his hand on a hand-rest for fingerprinting or hand geometry detection and has to stand in a fixed position in front of a camera for iris or retina identification. However, face recognition can be done passively without any explicit action or participation on the part of the user since face images can be acquired from a distance by a camera. This is particularly beneficial for security and surveillance purposes. Furthermore, data acquisition in general is fraught with problems for other biometrics: techniques that rely on hands and fingers can be rendered useless if the epidermis tissue is damaged in some way (i.e., bruised or cracked). Iris and retina identification require expensive equipment and are much too sensitive to any body motion. Voice recognition is susceptible to background noises in public places and auditory fluctuations on a phone line or tape recording. Signatures can be modified or forged. However, facial images can be easily obtained with a couple of inexpensive fixed cameras. Good face recognition algorithms and appropriate preprocessing of the images can compensate for noise and slight variations in orientation, scale and illumination. Finally, technologies that require multiple individuals to use the same equipment to capture their biological characteristics potentially expose the user to the transmission of germs and impurities from other users. However, face recognition is totally non-intrusive and does not carry any such health risks.

II. Proposed Methodology

The identification of a person by their facial images can be done in a number of different ways such as by capturing an image of the face in the visible spectrum using an inexpensive camera or by using the infrared patterns of facial heat emission. Facial Recognition in visible light typically model key features from the central portion of the facial image using a wide assortment of cameras in visible light system extract features from the captured images that do not change over time while avoiding superficial features such as facial expression or

hair. Several approaches to model facial images in the visible spectrum are Principal Component Analysis (PCA), local feature analysis, Neural Network, multi-resolution analysis etc. The two main methods for character recognition are SOM(self organized map) PCA(principle component analysis).

2.1 The Self-Organizing Map

T. Kohonen introduced the Self-Organizing Map (SOM). It is an unsupervised learning process, which learns the distribution of a set of patterns without any class information. It has the property of topology preservation. There is a competition among the neurons to be activated or fired. The result is that only one neuron that wins the competition is fired and is called winner-takes-all neuron. SOMs may be one-dimensional, two-dimensional or multidimensional, but the most common ones are either one-dimensional or two-dimensional maps. The number of input connections depends on the number of attributes to be used in the classification. The neuron with weights closest to the input data vector is declared the winner during the training. Then the weights of all of the neurons in the neighbourhood of the winning neuron are adjusted by an amount inversely proportional to the distance. It clusters and classifies the data set based on the set of attributes used.

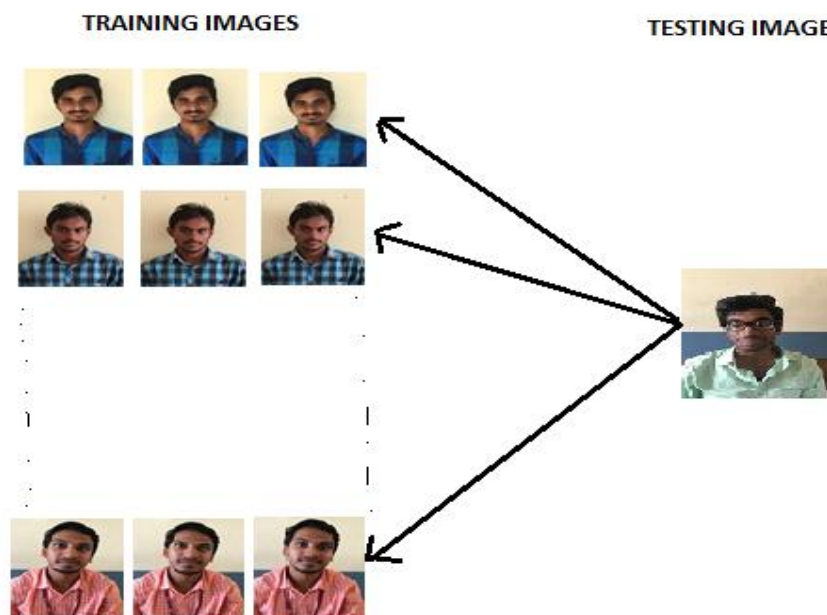
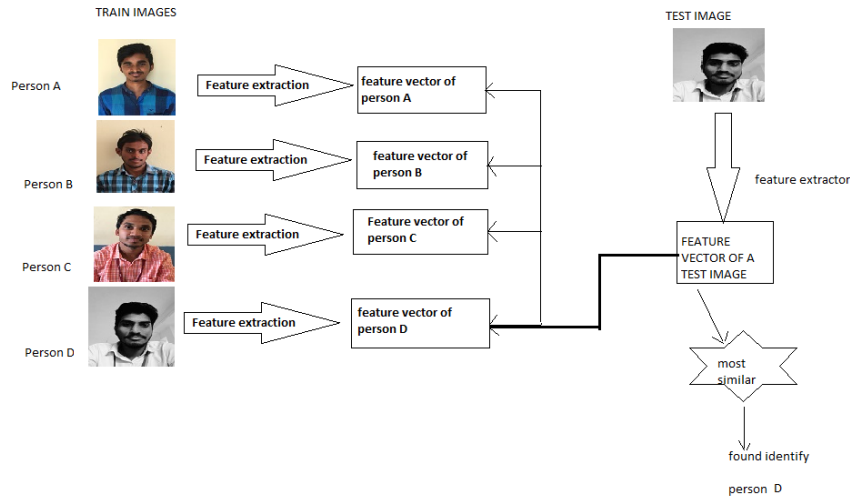


Fig.1. Comparing a new image with the database

2.2 PCA BASED FACE RECOGNITION

Many techniques can be used for face recognition but Principle Component Analysis is mostly followed and good technique. Principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. Now it is mostly used as a tool in exploratory data analysis and for making predictive models. PCA involves the calculation of the eigenvalue decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after mean centering the data for each attribute. The results of a PCA are usually discussed in terms of component scores and loadings. PCA is theoretically the optimal linear scheme, in terms of least mean square error, for compressing a set of high dimensional vectors into a set of lower dimensional vectors and then reconstructing the original set. It is a non-parametric analysis and the answer is unique and independent of any hypothesis about data probability distribution. However, the PCA compressions often incur loss of information.



III. Proposed Algorithm

3.1 SOM Based Algorithm

The algorithm is summarized as follows

Step 1- Initialization: Choose random values for the initial weight vectors $w_j(0)$, the weight vectors being different for $j = 1, 2, \dots, l$ where l is the total number of neurons

Step 2- Sampling: Draw a sample x from the input space with a certain probability.

Step 3- Similarity Matching: Find the best matching (winning) neuron $i(x)$ at time steps n by using the minimum distance Euclidean criterion

$$i(x) = \arg \min_j \|X(n) - w_j\|, \quad j = 1, 2, \dots, l$$

Step 4- Updating: Adjust the synaptic weight vector of all neurons by using the update formula

$$W_j(n+1) = W_j(n) + \eta(n) h_{j,i(x)}(n) (x(n) - w_j(n))$$

where, $\eta(n)$ is the learning rate parameter, and $h_{j,i(x)}(n)$ is the neighbourhood function centered around the winning neuron $i(x)$. Both $\eta(n)$ and $h_{j,i(x)}(n)$ are varied dynamically during learning for best results. Step 5- Continue with Step-2 until no noticeable changes in the feature map are observed.

Steps:-

The steps in constructing a SOM:

Step 1 - Initializing the Weights

Step 2 - Obtain Best Matching Unit

Step 3 - Scale Neighbours:

- a) Determining Neighbours and
- b) Training

Step 4 - Mapping : Determining the Quality of SOMs

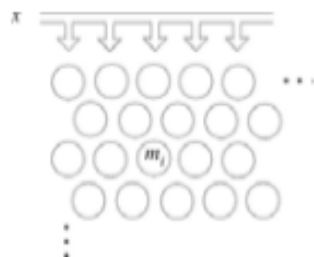


Fig. Cell arrangement for map

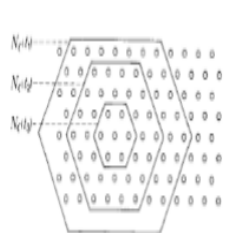


Fig. Examples of topological neighbourhood $N_0(t)$, where $t_1 < t_2 < t_3$

Faces represent complex, multidimensional, meaningful visual stimuli and developing a computational model for face recognition is difficult. The self-organizing map provides a quantization of the image samples into a topological space where inputs that are nearby in the original space are also nearby in the output space, thereby providing dimensionality reduction and invariance to minor changes in the image sample, and the Convolutional neural network provides for partial invariance to translation, rotation, scale, and deformation. The self-organizing map provides a quantization of the image samples into a topological space where inputs that are nearby in the original space are also nearby in the output space, which results in invariance to minor changes in the image samples

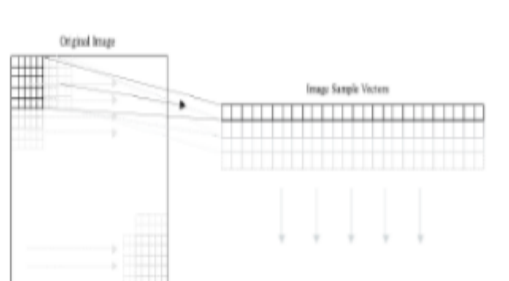


Fig. A Depiction of the local image sampling process. A window is stepped over the image and a vector is created at each location

3.2 PCA Based Algorithm

The whole recognition process involves two steps:

- A. Initialization process
- B. Recognition process

The Initialization process involves the following operations:

- i. Acquire the initial set of face images called as training set.
- ii. Calculate the Eigenfaces from the training set, keeping only the highest eigenvalues. These M images define the face space. As new faces are experienced, the eigenfaces can be updated or recalculated.
- iii. Calculate distribution in this M-dimensional space for each known person by projecting his or her face images onto this face-space.

These operations can be performed from time to time whenever there is a free excess operational capacity. This data can be cached which can be used in the further steps eliminating the overhead of re-initializing, decreasing execution time thereby increasing the performance of the entire system. Having initialized the system, the next process involves the steps:

- i. Calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the Eigenfaces.
- ii. Determine if the image is a face at all (known or unknown) by checking to see if the image is sufficiently close to a —free space.
- iii. If it is a face, then classify the weight pattern as either a known person or as unknown. iv. Update the eigen faces or weights as either a known or unknown, if the same unknown person face is seen several times then calculate the characteristic weight pattern and incorporate into known faces. The last step is not usually a requirement of every system and hence the steps are left optional and can be implemented as when the there is a requirement.

IV. Eigenface Algorithm

Let a face image $\Gamma(x, y)$ be a two dimensional M by N array of intensity values. In this thesis, I used a set of image by 200×149 pixels. An image may also be considered as a vector of dimension $M \times N$, so that a typical image of size 200×149 becomes a vector of dimension 29,800 or equivalently a point in a 29,800 dimensional space.

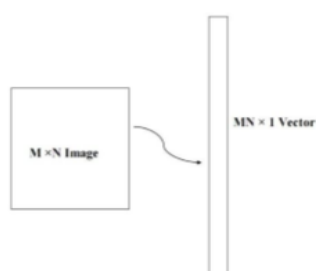


Fig. Conversion $M \times N$ image into $MN \times 1$ vector

Step1: prepare the training faces Obtain face images I1, I2, I3, I4 , IM (training faces). The face images must be centered and of the same size.

Step 2: Prepare the data set Each face image Ii in the database is transformed into a vector and placed into a training set S.

$$S = \{ \phi_1, \phi_2, \phi_3, \phi_4, \dots, \phi_M \}$$

In My example M = 34. Each image is transformed into a vector of size MN × 1 and placed into the set. For simplicity, the face images are assumed to be of size N × N resulting in a point in dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Step 3: compute the average face vector The average face vector (Ψ) has to be calculated by using the following formula:

$$\Psi = \frac{1}{M} \sum_{N=1}^M \Gamma_n$$

Step 4: Subtract the average face vector The average face vector is subtracted from the original faces and the result stored in the variable ,

$$\Phi_i = \Gamma_i - \Psi$$

Step 5: Calculate the covariance matrix We obtain the covariance matrix C in the following manner,

$$C = \frac{1}{M} \sum_{N=1}^M \Phi_n \Phi_n^T$$

$$= AA^T \quad (N^2 \times N^2 \text{ matrix where,}$$

$$A = [\phi_1, \phi_2, \phi_3, \phi_4, \dots, \phi_M] \quad (N^2 \times M \text{ matrix))}$$

Step 6: Calculate the eigenvectors and eigenvalues of the covariance matrix

The covariance matrix C in step 5 has a dimensionality of N² × N² so one would have N² eigenface and eigenvalues. For a 256 × 256 image that means that on must compute a 65, 536 × 65, 536 matrix and calculate 65,536 eigenfaces. Computationally, this is not very efficient as most of those eigenfaces are not useful for our task. In general, PCA is used to describe a large dimensional space with a relative small set of vectors.

Step 6.1: consider the matrix

Compute the eigenvectors u_i of AA^T

The matrix AA^T is very large -- ► not practical!!!

$$L = A^T A \quad (M \times M \text{ matrix})$$

Step 6.2: compute eigenvectors V_i of L=A^TA

$$A^T A V_i = \mu_i v_i$$

What is the relationship between u_i and v_i ?

$$A^T A V_i = \mu_i v_i$$

$$A A^T A V_i = \mu_i v_i$$

$$C u_i = \mu_i v_i \quad \text{where, } u_i = A v_i$$

Then C=A A^T and L=A^TA have the same eigenvalues and their eigenvectors are related as follows:

$$u_i = A v_i$$

Note 1: C=A A^T can have upto N² eigen values and eigen vectors.

Note 2: L= ATA can have upto M eigen values and eigen vectors.

Note 3: The M eigenvalues of C=A A^T (along with their corresponding eigenvectors) correspond to the M largest eigenvalues of L= A^TA (along with their corresponding eigenvectors).

Where is an eigenvector of L= A^TA . From this simple proof we can see that AV_i is an eigenvector of C=A A^T.

The M eigenvectors of L= A^TA are used to find the M eigenvectors u_i of C that form our eigenface basis:

$$u_i = \frac{1}{M} \sum_{i=1}^M v_i \phi_i$$

where u_i are the eigen vectors i.e. Eigenfaces.

Step 7: keep only K eigenvectors (corresponding to the K largest eigenvalues) Eigen faces with low eigenvalues can be omitted, as they explain only a small part of Characteristic features of the faces.

V. Projection of Training Samples Into The Eigenface Space

Next we have to project the training sample into the Eigen face space. The feature weight for the training images can be calculated by the following formula:

$$\omega_i = \mu_i^T (\Gamma_i - \psi)$$

Where, μ_i is the i^{th} Eigenfaces and $i=1,2,3,\dots,K$. The weight is obtained as above form a vector as follows

$$\Omega_i^T = [\omega_1, \omega_2, \omega_3, \dots, \omega_k]$$

VI. Testing Sample Classifications

- a) Read the test image and separate face from it.
- b) Calculate the feature vector of the test face. The test image is transformed into its eigenface components. First we compare line of our input image with our mean image and multiply their difference with each eigenvectors. Each value would represent a weight and would be saved on a vector Ω^T

$$\omega_{test} = \mu_i^T (\Gamma_{test} - \psi)$$

Where, μ_i is the i^{th} Eigenfaces and $i=1,2,3,\dots,K$.

$$\Omega_{test}^T = [\omega_1, \omega_2, \omega_3, \dots, \omega_k]$$

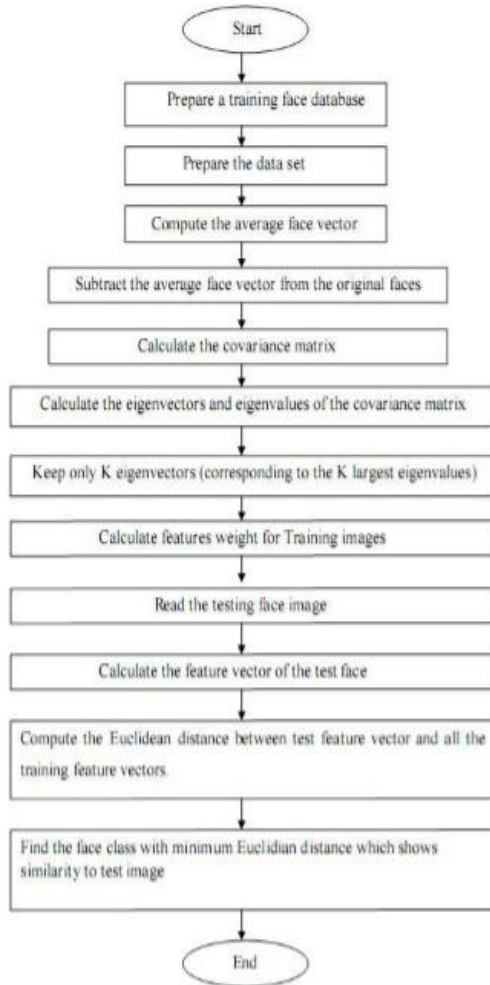
- c) Compute the average distance (Euclidean distance) between test feature vector and all the training feature vectors. Mathematically, recognition is finding the minimum Euclidean distance, between a testing point and a training point given in the following equation.

$$\epsilon_k = \sqrt{\|\Omega_{test} - \Omega_i\|^2}$$

Where, $i = 1, 2, 3, \dots, K$. The Euclidean distance between two weight vectors thus provides a measurement of similarity between the corresponding images.

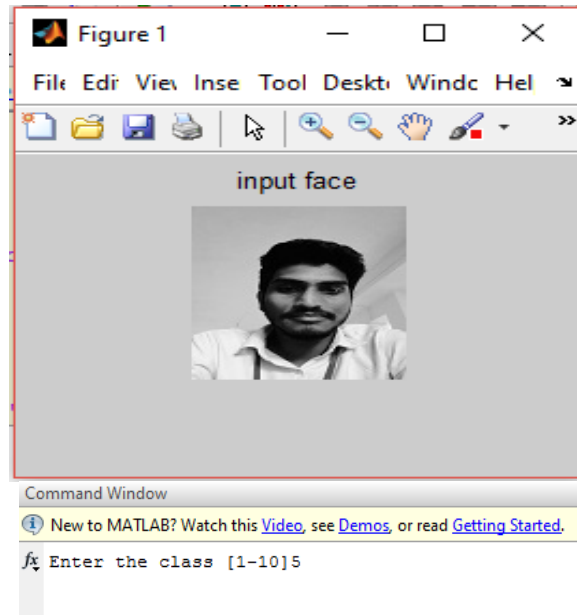
- d) The face class with minimum Euclidian distance shows similarity to test image.

VII. Flowchart

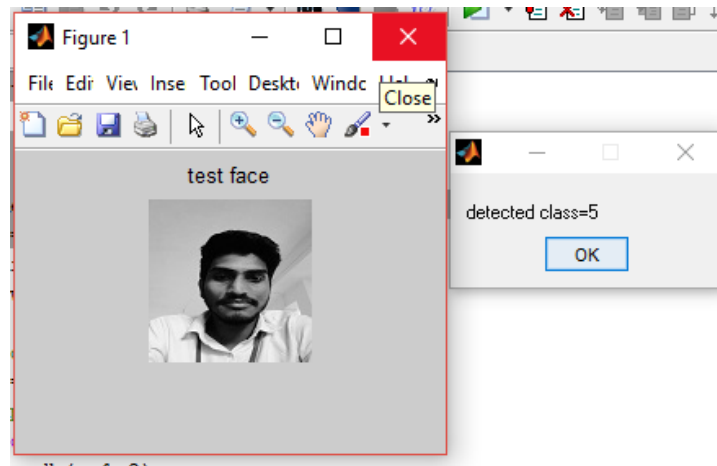


VIII. Results And Analysis

SOM BASED RESULT
TRAIN IMAGE



TEST IMAGE

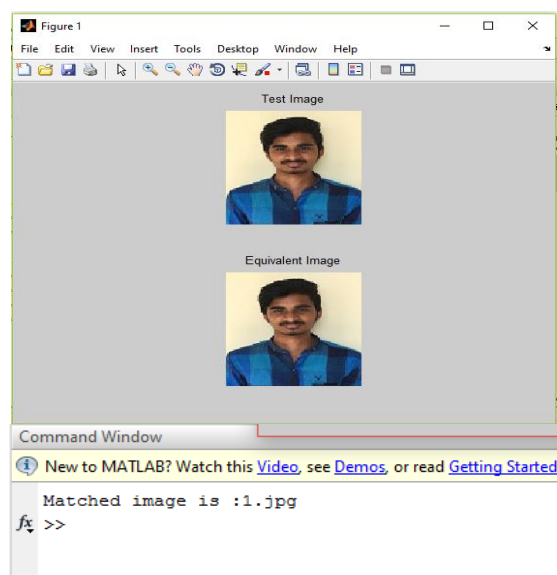
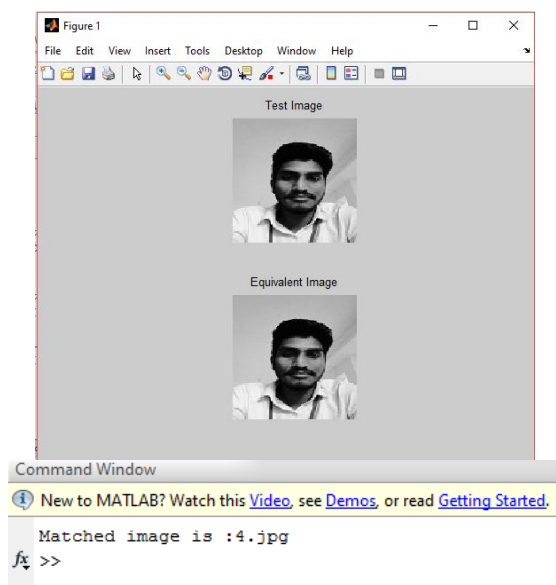


DATA BASE

Editor - moduletraining.m

db <6x3 double>				
	1	2	3	4
1	126.1845	66.9296	1	
2	126.8992	67.1737	1	
3	106.0463	60.5129	2	
4	106.8224	60.9225	2	
5	153.9004	72.0863	3	
6	154.7742	72.0892	3	
7				

PCA BASED RESULTS



IX. Conclusion

In this paper we implemented the face recognition system using Self Organizing map and Principal Component Analysis. The system successfully recognized the human faces and worked better in different conditions of face orientation. SOM is sheet-like artificial neural network, the cells of which become specially tuned to various input signal patterns or classes through an unsupervised learning process. Each cell or local cell group acts like separate decoder for the same input it is thus the presence or absence of an active response at the location and not much the exact input-output signal transformation or magnitude of the response, that provides an interpretation of the input information. SOM reduce dimensions and display similarities. Self-Organizing Maps are topologically ordered, which leads to good extracting feature ability.

References

- [1]. Self Organizing Maps, Tom Germano, March 23, 1999.
- [2]. Self-Organizing formation of topologically correct feature maps, Biol. Cybern., vol. 43, pp.59- 69, 1982.
- [3]. Self-Organizing Map, Teuvo Kohonen, vol. 78, No.9, 1990.
- [4]. Face Recognition Using Self-Organizing Maps, Qiu Chen, Koji Kotani, Feifei Lee and Tadahiro Ohmi, Tohoku University, Japan
- [5]. AT&T Laboratories Cambridge, The database of faces at <http://www.cl.cam.ac.uk/research/dtg/attarchive/facesataglance.html>.
- [6]. Face Recognition Using PCA versus ICA versus LDA cascaded with the Neural Classifier of Concurrent Self Organising Maps, Victor-Emil Neagoe and Ioan-Anton Stanculescu, 2008, IEEE Transaction.
- [7]. Face Recognition Using Self-Organizing Map and Principal Component Analysis, Dinesh Kumar, C S Rai and Shakti Kumar, 2007.
- [8]. <http://www.face-rec.org>

- [9]. Shemi P M, Ali M A, A Principal Component Analysis Method for Recognition of Human Faces: Eigenfaces Approach, International Journal of Electronics Communication and Computer Technology (IJECCT), Volume 2 Issue 3 (May 2012).
- [10]. M. Turk, A. Pentland: Face Recognition using Eigenfaces, Conference on Computer Vision and Pattern Recognition, 3 – 6 June 1991,
- [11]. Prof. Y. Vijaya Lata , Chandra Kiran Bharadwaj Tungathurthi , H. Ram Mohan Rao , Dr. A. Govardhan , Dr. L. P. Reddy, Facial Recognition using Eigenfaces by PCA, Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering & Technology, Jawaharlal Nehru Tech. University.
- [12]. Parvinder S. Sandhu, Iqbaldeep Kaur, Amit Verma, Samriti Jindal, Inderpreet Kaur, Shilpi Kumari, Face Recognition Using Eigen face Coefficients and Principal Component Analysis, International Journal on Electrical and Electronics Engineering 3:8 2009.