International Journal of Current Engineering and Technology ISSN 2277 – 4106
©2013 INPRESSCO. All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Real Time Facial Expression Recognition System Using 2D-DCT and Neural Network

Keerti Keshav Kanchi[a*] and Vijaya C[a]

[a]Department of Electronics and Communication Engineering, SDMCET, Dharwad, India

Abstract

The Face recognition is an important and secured way to protect the frauds at everywhere like government agencies which are investing a considerable amount of resources into improving security systems as result of recent terrorist events that dangerously exposed flaws and weaknesses in today's safety mechanisms. In this paper discrete cosine transform and neural network is used in the development of facial expression recognition system capable of operating in real time. The proposed technique uses video input as input image taken through the webcam in real time on which two dimensional discrete cosine transform (2D-DCT) is performed for image compression and self organizing map (SOM) for recognition purpose. The DCT extracts features from face image; these feature-vectors are constructed by computing DCT coefficients. A self-organizing map(SOM) using an unsupervised learning techniques used to classify DCT-based features vectors into groups to identify if the subject in the input image is present or not present in the image database. Evaluation of the procedure is performed in MATLAB using an image database of 25 people containing 5 subjects and each subject have 5 different facial expressions. After training about 1000 epochs system achieved approximately 89.18% recognition rate.

Keywords:*2D-Discrete Cosine Transform (2D-DCT); Facial Expression Recognition; Image Processing Neural Network; Kohonen Self Organizing Map (SOM).*

1. Introduction

Recognizing faces is something that people usually do effortlessly and with-out much conscious thought, yet it has remained a difficult problem in the area of computer vision, where some 20 years of research is just beginning to yield useful technological solutions. Face recognition has become a very active area of research in recent years mainly due to increasing security demands and its potential commercial and law enforcement applications. The recognition of individuals without their full co-operation is in high demand by security and intelligence agencies requiring a robust person identification system [RC Gonzalez et al, 1992]. A facial recognition and face verification system can be considered as a computer application for automatically identifying or verifying a person in a digital image in as much as the processing is carried out on digital video input image taken through the webcam in real time.

This paper describes real time facial expression recognition system using video input. Here an image database of 25 people containing five subjects and each subject have five different facial expressions which are taken in real time using webcam. All the 25 face images are compressed using two dimensional discrete cosine transform (2D-DCT). When the 2D DCT is applied with a mask, high-coefficients in an image are discarded. Then the 2D IDCT is applied to regenerate the compressed image, which is blurred due to the loss of quality and also smaller in size. The next stage uses a Kohonen self-organizing map (SOM) with an unsupervised learning technique which is trained to classify vectors into groups to recognize if the subject in the input image is present or not present in the image database. After training all the 25 face images, we now take a single video input image, compress it using 2D-DCT and regenerate it using IDCT. All the 25 trained images and untrained input image are simulated. The untrained input image is compared with all trained images, if the face image is classified as present, the best match image is found in the training database using minimum absolute deviation and that image is displayed otherwise if the image is not found then image is not found in the database is displayed.

1.1 Image Processing

Image processing is a form of signal processing in which the input is image, like video frame or photograph and output may be image or characteristics associated with that image. A digital image is composed of a finite number of elements, each of which has a particular location and value .These elements are referred to as picture elements, image

*Corresponding author: **Keerti Keshav Kanchi**

elements or pixels. An image is digitized to convert it to a form which can be stored in a computer's memory or on some form of storage media, such as a hard disk or CD ROM [Simon Haykin, 2004]. This digitization procedure can be done by a scanner, or by a video camera connected to a frame grabber board in a computer. Once image has been digitized, it can be operated upon by various image processing operations.

Some applications of Image processing are
1. Robotics 2.Medical field 3.Graphics and animation 4.Satellite imaging

1.2 Image Processing Techniques

Image processing techniques are used to enhance, improve or alter an image and to prepare it for image analysis. Image processing is divided into many sub processes, including Histogram analysis, thresholding, masking, edge detection, segmentation and others.

1.3 Real Time Image Processing

In many of the techniques considered so far the image is digitized and stored before processing .In other situations ,although the image is not stored, the processing routine requires long computation times before they are finished. This means that in general there is a long lapse between the time and image is taken and the time a result obtained [Simon Haykin, 2004]. This may be acceptable in situations in which the decisions do not affect the processing other situations there is a need for real time processing such the results are available in real time or in a short enough time to be considered real time processing.

1.4 Image Processing Toolbox:

MATLAB is complemented by a family of application specific solutions called toolboxes. The image processing toolbox is a collection of MATLAB functions (called m-functions or m-files) that extend the capability of the MATLAB environment for the solution of digital image processing problems .Other toolboxes that used to complement the image processing tool boxes are the signal processing, Neural Networks, Fuzzy logic and wavelet Toolboxes.

1.4.1 Image Types:

The toolbox supports four types of images. They are grey scale images, binary images, indexed images and RGB images. A gray scale image is a data matrix whose value represents shades of gray. A binary image is a logical array of 0s and 1s. An indexed image does not explicitly contain any colour information .Its pixel values represent indices into a colour lookup table .Colours are applied by using these indices into a colour lookup table (LUT).Colours are applied by using these indices to look up the corresponding RGB triplet in the LUT. An RGB image is a three dimensional byte array that explicitly stores a colour value for each pixel. RGB image array are made up of width , height and three channels of colour information

2. Discrete Cosine Transform

2.1 Overview

The discovery of the Discrete Cosine Transform (DCT) in 1974 is an important achievement for the research community working on image compression. The Discrete Cosine Transform possess some fine properties, i.e., de-correlation, energy compaction, separability, symmetry and orthogonality, due to which it is virtually used in every image/video processing standard such as signal and image processing and especially for lossy data compression because it has a strong energy compaction property: most of the signal information tends to be concentrated in a few low-frequency components of the DCT and high-frequency components are eliminated, approaching the Karhunen-Loève Transform (KLT) for signals based on certain limits of Markov processes. Compression standards like JPEG, for compression of still images, MPEG, for compression of motion video, MP3, for compression of audio streams and the H.263, for compression of video telephony and teleconferencing all employ the basic technique of the DCT.

2.2 Definition

The DCT is regarded as a discrete-time version of the Fourier-cosine series. Hence, it is considered as a Fourier-related transform similar to the Discrete Fourier Transform (DFT), using only real numbers. Since DCT is real-valued, it provides a better approximation of a signal with fewer coefficients. Figure 1 illustrates the two-dimensional representation of the DCT.

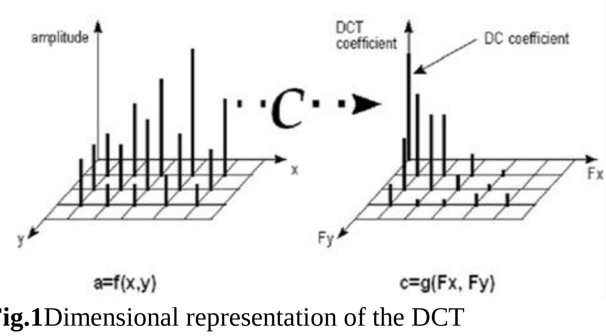

Fig.1 Dimensional representation of the DCT

The DCT is a transform which transforms a signal or image from the spatial domain to the elementary frequency domain. Lower frequencies are more obvious in an image than higher frequencies an image is transferred into its frequency components and higher frequency coefficients are discarded, the amount of data needed to describe the image without sacrificing too much image quality will reduce [Simon Haykin, 2004]. Thus, DCT can be computed with a Fast Fourier Transform (FFT) like algorithm. Hence it, can be concluded that: The DCT decorrelates image data, after which each transform coefficient is encoded independently without losing compression efficiency.

2.2.1 One-dimensional DCT

The discrete cosine transform is a linear invertible function F: RN -> RN (where R denotes the set of real numbers), or equivalently an N x N square matrix. Mathematically, the 1D discrete cosine transform (1D DCT) X[k] of a sequence x[n] of length N is defined as:

$$X[k]=\alpha[k]\sum_{n=0}^{N-1}x[n]\cos\left(\frac{\pi(2n+1)k}{2N}\right)$$
$$,k=0,1,,..............,N-1 \qquad (2.1)$$

Also, the inverse 1D DCT is defined as:

$$X[k]=\sum_{n=0}^{N-1}\alpha[k]X[k]\cos\left(\frac{\pi(2n+1)k}{2N}\right)$$
$$,n=0,1,..........N-1 \qquad (2.2)$$

where in both Equations 2.1 and 2.2, α[k] is defined as

$$\alpha[k]=\begin{cases}\sqrt{\frac{1}{N}}, & for\ \ k=0\\ \sqrt{\frac{2}{N}}, & for\ \ k=1,2,...,N-1\end{cases}$$

The basis sequences of the 1D DCT are real, discrete-time sinusoids are defined by:

$$C_N[n,k]=\cos\left(\frac{\pi(2n+1)k}{2N}\right) \qquad (2.4)$$

Each element of the transformed list X[k] in equation 2.1 is the inner dot product of the input list x[n] and a basis vector. Constant factors are chosen so the basis vectors are orthogonal and normalized. The DCT can be written as the product of a vector (the input list) and the N x N orthogonal matrix whose rows are the basis vectors.

2.2.2 Two-dimensional DCT

The two-dimensional discrete cosine transform (2D-DCT) is used for processing signals such as images. The 2D DCT resembles the 1D DCT transform since it is a separable linear transformation; that is if the two-dimensional transform is equivalent to a one-dimensional DCT performed along a single dimension followed by a one-dimensional DCT in the other dimension. For e.g, in an n x m matrix, S, the 2D DCT is computed by applying it to each row of S and then to each column of the result.
The 2-D DCT is similar to a Fourier transform but uses purely real math. It has purely real transform domain coefficients and incorporates strictly positive frequencies. The 2D DCT is equivalent to a DFT of roughly twice the length, operating on real data with even symmetry, where in some variants the input and/or output data are shifted by half a sample [Anil K. Jain, 1989]. As the 2D DCT is simpler to evaluate than the Fourier transform, it has become the transform of choice in image compression standards such as JPEG. The 2D DCT represents an image as a sum of sinusoids of varying magnitudes and frequencies. It has the property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients of the DCT. The series form of the 2D discrete cosine transform (2D DCT) is defined as:

$$X[k_1,k_2]=\alpha[k_1]\alpha[k_2]=\sum_{n=0}^{N_1-1}\sum_{n=0}^{N_2-1}x[n_1,n_2]\cos\left(\frac{\pi(2n_1+1)k_1}{2N_1}\right)\cos\left(\frac{\pi(2n_2+1)k_2}{2N_2}\right) \qquad (2.5)$$

for k_1 = 0, 1, ... N_1-1 and k_2 = 0, 1, ... N_2-1 and,

$$x[n_1,n_2]=\sum_{n=0}^{N_1-1}\sum_{n=0}^{N_2-1}\alpha[k_1]\alpha[k_2]X[k_1,k_2]\cos\left(\frac{\pi(2n_1+1)k_1}{2N_1}\right)\cos\left(\frac{\pi(2n_2+1)k_2}{2N_2}\right) \qquad (2.6)$$

for n_1= 0, 1, ...,... N_1-1 and n_2 = 0, 1, ..., N_2-1, with α[k] defined in equation 2.3
Equation 2.5 is called the analysis formula or the 'forward transform', while equation 2.6 is called the synthesis formula or 'inverse transform'. Mathematically, the DCT is perfectly reversible and there is no loss of image definition until coefficients are quantized.

2.2.3 DCT Image Compression

The proposed design technique calculates the 2D-DCTof the image blocks of size 8 × 8 pixels using '8' out of the 64 DCT coefficients for masking. The other 56 remaining coefficients are discarded (set to zero). The image is then reconstructed by computing the 2D-IDCT of each block using the DCT transform matrix computation method. Finally, the output is a set of arrays. Each array is of size 8 × 8 pixels and represents a single image [A S. Raja et al, August 2012]. Empirically, the upper left corner of each 2D-DCT matrix contains the most important values, because they correspond to low-frequency components within the processed image block.

3. Facial expression recognition

Face recognition is a pattern recognition task performed specifically on faces. It can be described as classifying a face either known or unknown, after comparing it with stored known individuals. It is desirable to have a system that has the ability of learning to recognize unknown faces. Facial expression recognition involves comparing an image with a database of stored faces in order to identify the individual in the input image [A S. Raja et al, August 2012]. The related task of face detection has direct relevance to recognition because images must be analyzed and faces identified, before they can be recognized. Face recognition, although a trivial task for the human brain and has proved to be extremely difficult to imitate artificially, because although commonalities exist between faces, they can vary considerably in terms of age, skin color, orientation, facial expression and presence of facial furniture such as glasses or facial hair. In this paper real time based facial expression recognition is used which automatically detects face regions, ex-tracts features from the video input, and recognizes facial identity if a face is present [L. Ma et al, Oct 2004]. In surveillance, information security, and access control applications, face recognition and identification from a video sequence is an important problem. Face recognition based on real time is preferable over using the still images. Though recognition of faces from video sequence is a direct extension of still-image-based recognition.

3.1 Kohonen Self-Organizing Maps

Self-Organizing Maps (SOM) were introduced by a Finnish Professor, Teuvo Kohonen in 1982, thus SOM's are also sometimes referred to as Kohonen Maps. Self-Organizing Maps are a subtype of Artificial Neural Network. They are trained using unsupervised learning to produce low dimensional representation of the training samples while preserving the topological properties of the input space. Thus, SOM are reasonable for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling. The principal goal of self-organizing maps is to transform an incoming signal pattern of arbitrary dimension into a one or two dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion. Self-organizing maps learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors [L. Ma et al, June 2004]. They provide 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.

3.2 Overview

Self-organizing maps are a single layer feed forward network where the output syntaxes are arranged in low dimensional (usually 2D or 3D) grid. Each input is connected to all output neurons. Attached to every neuron there is a weight vector with the same dimensionality as the input vectors. Hence, in this system, a SOM is employed to classify DCT-based vectors into groups to identify if the subject in the input image is present or not present in the image database [L. Ma et al, June 2004] .The number of input dimensions is usually a lot higher than the output grid dimension. A Self-organizing map is shown in Figure 2.

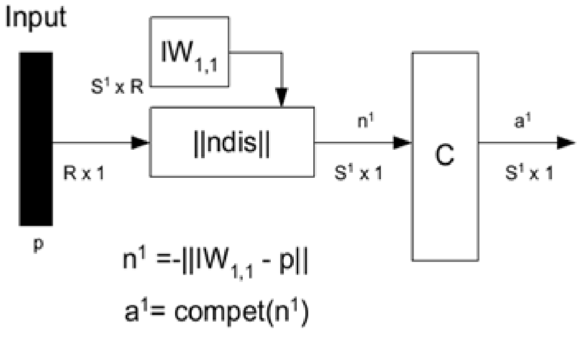

Fig 2: Self-Organizing map

3.3 Network Architecture

Self-organizing maps are single layer feed forward networks where the output syntaxes are arranged in low dimensional (usually 2D or 3D) grid. Each input is connected to all output neurons. Attached to every neuron there is a weight vector with the same dimensionality as the input vectors. The number of input dimensions is usually a lot higher than the output grid dimension. SOMs are mainly used for dimensionality reduction rather than expansion. The architecture for a simple self-organizing map is shown in Figure 3.

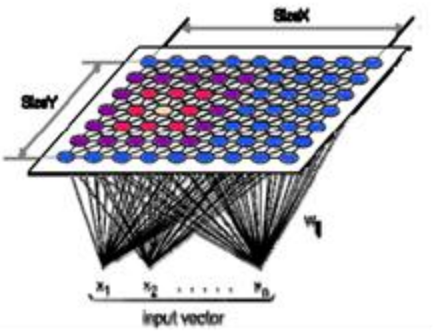

Fig 3: Self-Organizing map architecture

The input vector *p* is the row of pixels of the image. The ||ndis|| box in the Figure 3 accepts the input vector *p* and the input weight matrix $IW_{1,1}$ produces a vector having S1 elements. The elements are the negative of the distances between the input vector and vectors $IW_{1,1}$ formed from the rows of the input weight matrix [Fasel. B, 2003]. The competitive transfer function accepts a net input vector for a layer and returns neuron outputs of 0 for all neurons except for the winner, the neuron associated with the most positive element of net input n^1. The winner's output is 1. The neuron whose weight vector is closest to the input vector has the least negative net input and, therefore, wins the competition to output a 1. Thus the competitive transfer function produces a 1 for output element $a^1{}_i$ corresponding to i*, the winning neuron. All other output elements in a^1 are 0.

A self-organizing feature map network identifies a winning neuron using the same procedure as employed by a competitive layer. However, instead of updating only the winning neuron, all neurons within a certain neighbourhood of the winning neuron are updated using the Kohonen rule. Thus, when a vector *p* is presented, the weights of the winning neuron and its close neighbors move toward p. Consequently, after many presentations, neighboring neurons learn vectors similar to each other. Hence, the SOM network learns to categorize the input vectors it sees. Typically, a SOM has a life cycle of three phases: the learning phase, the training phase and the testing phase.

3.4 Unsupervised Learning

During the learning phase, the neuron with weights closest to the input data vector is declared as the winner. Then weights of all of the neurons in the neighbourhood of the winning neuron are adjusted by an amount inversely proportional to the Euclidean distance. It clusters and classifies the data set based on the set of attributes used. The learning algorithm is summarized as follows:

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

$$w_i = [w_{i1}, w_{i2}, \ldots\ldots, w_{il}]^T \in \Re^n \qquad (4.3.1)$$

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

$$x = [x_{1,} x_{2,} \ldots\ldots\ldots\ldots x_l]^T \quad \in \Re^n \qquad (4.3.2)$$

3. Similarity Matching: Find the best matching (winning) neuron $i(x)$ at time *t*, $0 < t \leq n$ by using the minimum distance Euclidean criterion:

$$i(x) = \arg\min_j \|x(n) - w_j\|, \; j = 1,2\ldots.l \qquad (4.3.3)$$

3. Updating: Adjust the synaptic weight vector of all neurons by using the update formula: $w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n)(x(n) - w_j(n))$ (4.3.4)

Where $\eta(n)$ is learning rate parameter, and $h_{j,i}x(n)$ is the neighbourhood function centred around the winning neuron. Both $\eta(n)$ and are $h_{j,i}x(n)$ varied dynamically during learning for best results.

4. Continue with step 2 until no noticeable changes in the feature map are observed.

Training images are mapped into a lower dimension using the SOM network and the weight matrix of each image stored in the training database. During recognition trained images are reconstructed using weight matrices and recognition is through untrained test images using Euclidean distance as the similarity measure.

3.5 Training

During the training phase, labelled DCT-vectors are presented to the SOM one at a time. For each node, the number of wins is recorded along with the label of the input sample [Hyun-chul choi et al, Oct 2006]. The weight vectors for the nodes are updated as described in the learning phase. By the end of this stage, each node of the SOM has two recorded values: the total number of winning times for subject present in image database, and the total number of winning times for subject not present in image database.

3.6 Testing

During the testing phase, each input vector is compared with all nodes of the SOM, and the best match is found based on minimum Euclidean distance [Hyun-chul choi et al, Oct 2006]. The final output of the system based on its recognition, displays if the test image is present or not present in the image database.

4. Proposed work

The proposed block diagram of real time facial expression recognition using 2D-DCT and SOM neural network is as shown below:

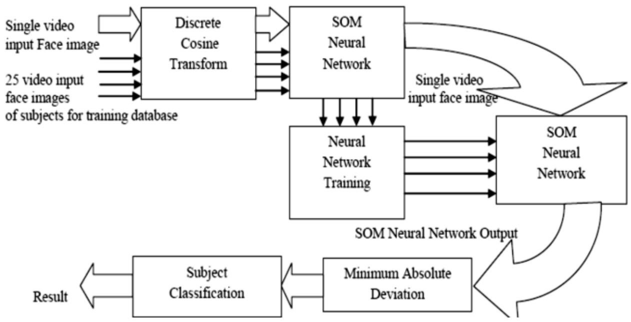

Fig 4: Block Diagram of Real Time Facial Expression Recognition using SOM Neural Network

Face images of different subjects through video input in real time were taken for the training database. Similarly five additional images of each individual subject with different facial expression were taken as shown in figure 4. The five facial expressions comprise of: happy, sad, neutral, angry, smiling. All face images were resized to 8 x 8 pixels and saved, the next step was to compress them by applying the 2D blocked DCT. When the 2D DCT is applied with a mask, high-coefficients in an image are discarded. Then the 2D IDCT is applied to regenerate the compressed image, which is blurred due to the loss of quality and also smaller in size. For the image data to be input into the neural network, it should follow the form of only one column, despite the number of rows. Currently, all the resized and DCT compressed face images are in the form of 8 x 8 pixels. Hence the image data needed to be reshaped from an 8 x 8 matrix to a 64 x 1 array for it to be used both for the input and training database of the neural network. All face image data was then reshaped to a 64 x 1 array. The actual data which will be input into the neural network as an video input face image or a set of more data combined together for the training database.

Self-Organizing Maps (SOM's) were chosen due to the failure of RBF (Radial Basis Function Neural Network) and FCM (Fuzzy C-Means Clustering) and). In RBF all outputs after continuous program simulations and change in input face images resulted in incorrect outputs; classifying wrong images from the input with the trained database and generating incorrect answers. Hence, the idea of using RBF for the design of the face recognition system was dropped whereas in FCM, it is proved to be efficient in managing image data, but on every trial it calculated cluster centres at different positions. Cluster centres were calculated based on iterations. The position difference of cluster centres for every trial made it difficult to correctly match the input image with the trained database for every trial. Since the output result on some trials used to be correct and on some trials used to be incorrect, due to the output un-certainty of this technique the idea was dropped for the face recognition system. After research and study, self organizing map (SOM) were found to be efficient for image data management and proved to be an accurate closest matching technique of untrained input images with trained database of images. For the design of SOM, a set of 25 video input image data with 5 different subjects and each subject have 5 different facial expressions are taken in real time for the training database and are loaded into MATLAB. A SOM was then created and the parameters for the SOM network were selected to be a minimum and maximum point for each row on vector; training database. There were 64 minimum and 64 maximum points selected for the training database in a 64 x 2 array. After the SOM neural network was created, it was trained for 1000 epochs. After the SOM neural network was trained and simulated for the 25 images in the training database, the SOM neural network was then simulated for the single video input face image. After the simulation of the input face image, the image in the training database which is the closest match by the SOM neural network for the video input face image is found by finding the minimum absolute deviation. After the closest matched training database images are found, they are then classified. Classification of the subject is the answer of the face recognition system.

5. Experimental results

5.1 Image Database

The training database consists of 25 video input face images taken in real time, containing 5 subjects and each subject having 5 different facial expressions as shown in figure below:

Fig 5 : Training database of 25 video input face images with different facial expressions.

6.2 Untrained Video Input Face Image

The untrained input face image in Figure 6 was simulated in the SOM network for the trained database in Figure 6 for 1000 epochs. The untrained input matched with the closest image of the same subject in the training database, displaying a correct answer, with the recognition rate of 89.18%. Video Input face images were then changed and the network was simulated repeatedly and the SOM neural network generated correct answers. Thus, through the tests mentioned above the SOM neural network validation proves the face recognition system to be accurate for untrained face images with different facial expressions. After SOM training, the 25-dimensional trained image database is transformed into a 64-dimensional map where the magnitude of the layer weights is increased and Euclidean distance for feature vector for trained image is smaller than untrained image, as shown in figure 8.This transformation produces better classification by grouping similar clusters together. Figure 9 shows the SOM weight vectors.

Fig 6 : untrained video input face image with different facial expression

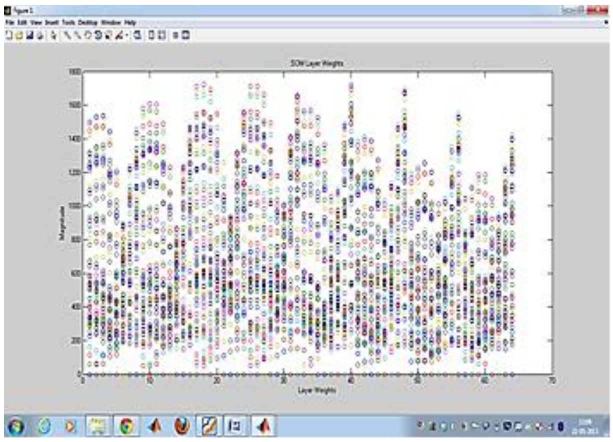

Fig 7 : Euclidean Distance Weight for Trained Image Database with SOM Network**.**

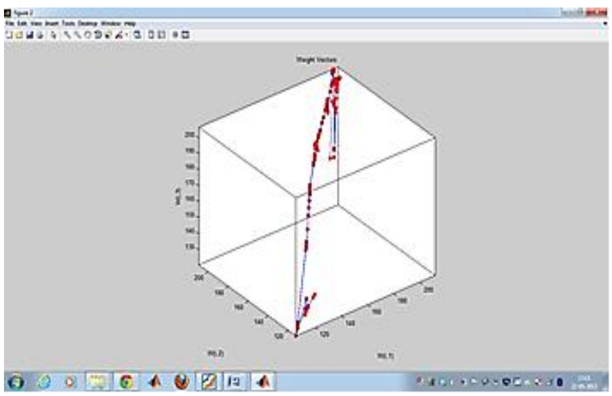

Fig 8 : SOM weight vectors

The output consists of different types of plots used for neural network training of self organizing map (SOM) which are as shown below:

5.1 SOM Topology (plotsomtop)

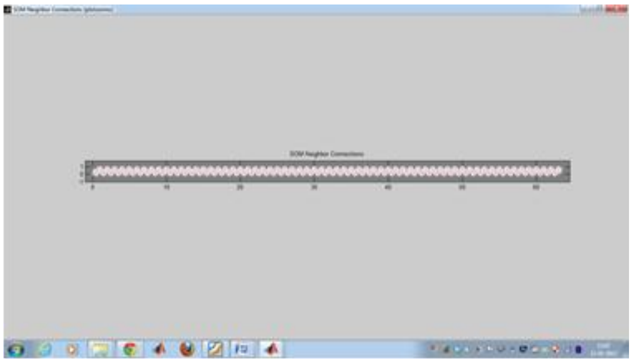

Fig 9: SOM Topology

For SOM training, the weight vector associated with each neuron moves to become the centre of a cluster of input vectors. In addition, neurons that are adjacent to each other in the topology should also move close to each other in the input space. The default topology is hexagonal; to view it, click SOM Topology from the network training window. In this figure, each of the hexagons represents a neuron. The grid is 64-by-64, so there are a total of 4096 neurons in this network.

5.2 SOM Neighbor Connections (plotsomnc)

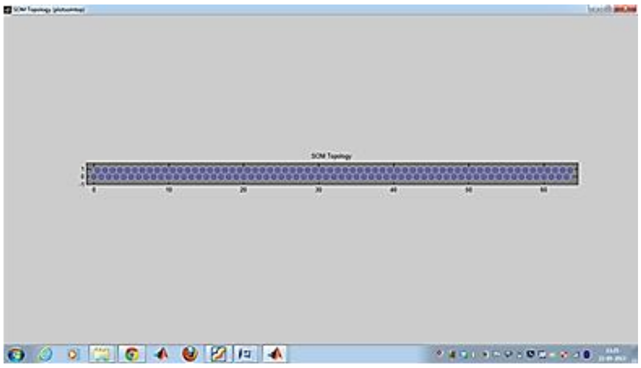

Fig 10: SOM Neighbor Connections

It plots a SOM layer showing neurons as gray-blue patches and their direct neighbor relations with red lines.

5.3 SOM Neighbor Distances (plotsomnd)

In the figure 11, the blue hexagons represent the neurons. The red lines connect neighboring neurons. The colors in the regions containing the red lines indicate the distances between neurons. The darker colors represent larger distances, and the lighter colors represent smaller distances. A band of dark segments crosses from the lower-centre region to the upper-right region. The SOM network appears to have clustered the flowers into two distinct groups.

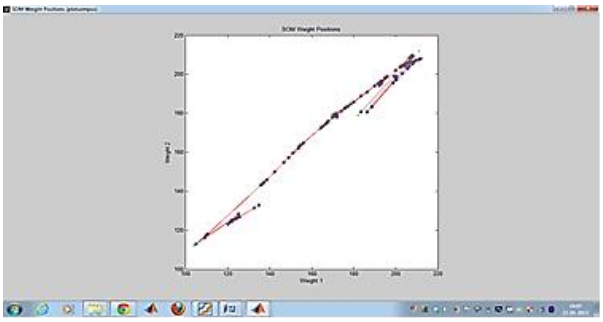

Fig 11: SOM Neighbor Distances

5.4 SOM Weight positions (plotsompos)

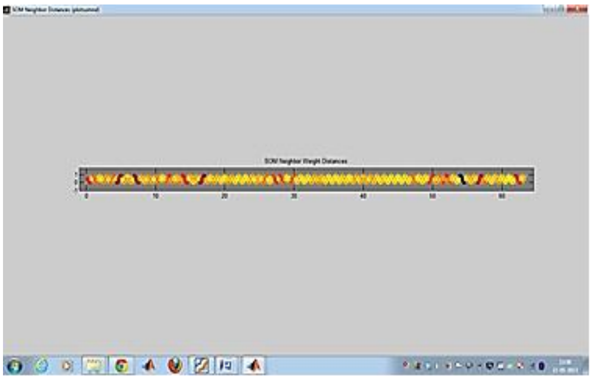

Fig 12: SOM Weight Positions

It plots the input vectors as green dots and shows how the SOM classifies the input space by showing blue-gray dots for each neuron's weight vector and connecting neighboring neurons with red lines.

5.5 Determining Optimal Number of Epochs for Training

Epochs are neural network training parameters. They are defined as one complete cycle through the neural network for all cases, which present the entire training set to the neural network. Each time a record goes through the net, it is one trial, one sweep of all records is termed as an Epoch. Less number of epochs used for training leads to less training time for the training data set. The goal is to find the optimal number of epochs for training which will produce accurate neural network results and at the same time require the least amount of time for program execution. The SOM neural network was tested to determine the optimal number of epochs to be used for neural network training. This test was performed by varying the number of epochs and network training time to find the best possible recognition rate.

Table I shows the tabulated results for the test. The training database for this test consists of 25 face images of 5 subjects, each subject having 5 different facial expressions.

Table I. Comparison of number of epochs vs. network training time and
Recognition rate

Number of epochs	Network Training Time (in sec)	Recognition Rate (%)
50	18.21	58.13
100	15.46	56.12
150	67.23	65.54
200	85.57	75.32
250	110.32	82.35
300	150.19	85.05
500	180.23	86.03
800	220.49	87.45
1000	279.13	89.18

Conclusion

This paper has presented a novel approach of real time facial expression recognition technique that uses features derived from DCT coefficients, along with a self organizing map (SOM) neural network-based classifier. The system was evaluated in MATLAB using an image database of 25 face images which are taken in real time, containing five subjects and each subject having 5 different facial expressions. After training for approximately 1000 epochs the system achieved a recognition rate of 89.18% for fastest network training time. The system has less computational requirement, this makes a system well suited for low cost and real-time hardware implementation. Commercial Implementations of this technique do not currently exist. However, it is conceivable that a practical SOM-based face recognition system may be possible in the future

Acknowledgment

The authors wish to thank the authorities at AICTE for granting RPS to the department of ECE, SDM college of Engineering and Technology Dharwad. This work is towards partial fulfilment of said project. The authors are also grateful to the authorities of SDMCET Dharwad, for the facility extended to carry out the present work. The authors also would like to thank the unknown reviewers for their invaluable comments which help in improving the paper.

References

R.C. Gonzalez Richards E. Woods (1993), Digital Image Processing, Addison- Wesley Publishing.

Simon Haykin, (2004) Neural Network-A Comprehensive Foundation, 2nd Ed., Pearson Education.

Anil K. Jain, (1989). Fundamentals of Digital Image Processing, Prentice-Hall.

A.S. Raja and V. Joseph Raj, (August 2012), Neural network based supervised self Organizing maps for face recognition International Journal on Soft Computing (IJSC) Vol.3, No.3.

L. Ma, Y. Xiao, and K. Khorasani,(Oct 2004), A new facial expression recognition technique using 2D DCT and k-means algorithm, in Proc. International Conference on Image Processing, pp. 1269-1272.

L. Ma and K. Khorasani, (June 2004), Facial expression recognition using constructive feed forward neural networks, IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 34, No. 3, pp. 1588-1595.

Fasel. B and Luettin. J, (2003) Automatic Facial Expression analysis: A survey, Pattern Recognition, , 36: 259–75.

Hyun-Chul Choi and Se-Young Oh,(Oct 2006) Real time Facial Expression Recognition using Active Appearance Model and Multilayer Perceptron, SICE-ICASE International Joint Conference. 18-21, in Bexco, Busan, Korea.