Image Compression using PCA and Improved Technique with MLP Neural Network

Vilas Gaidhane[1] , Vijander Singh[2], Mahendra Kumar[3]
[1] Department of Electronics & Communication,
G.L.A Institute of Technology and Management, Mathura, U.P, India, vilasgla98@gmail.com
[2] ICE Division, N.S.I.T. New Delhi, India, vijaydee@gmail.com
[3] U.P. Technical University, Lucknow, U.P, mahendra1812@gmail.com

Abstract **–Computer images consist of large data and hence require more space to store in the memory. The compressed image requires less storing space of memory and less time to transmit. In this paper, feed forward back propagation neural network method with PCA technique is used for image compression. It is found that the results using PCA techniques are not satisfactory. To improve these results author suggest another technique which has better results. The Artificial Neural Network (ANN) used in these technique is trained by considering the different numbers of hidden neurons, epoch and reconstructed image compared with original image.**

Keywords: Image compression, Feed forward neural network, Principle Component Analysis, PSNR.

I. INTRODUCTION

Image compression is playing an important role in communication applications. The main aim of the image compression is to remove the redundancy from the image data in such a way that it allows the same image reconstruction at the receiver end. There are mainly two types of image compression techniques, lossy and lossless image compression. In medical applications like X-ray images and EGS images are compressed by lossless compression method because each bit of information is essential. On other hand, digital or video images compressed by lossy compression techniques. Transform coding techniques like cosine transform are very effective techniques which gives better result but it requires more time for data compression [4]. Artificial neural network have been incorporate with the PCA Algorithm to solve an image compression problem discuss in papers [1],[2],[3]. For data compression, researchers focus on the principal component analysis [1] with the neural network approaches which is statistical method; transform n-dimensional subspace to m-dimensional subspace.

In 1982, Oja[10] proposed a method using a single neural element with the capability of extracting the principal component. It can be extended to a lot methods of estimating several principal components [1]. The Karhunnen-Oja's symmetric subspace learning rule and adaptive principal component extraction algorithm (APEX) can be used for principal component extraction [1].

But PCA technique depends upon the threshold value at which the iteration process of learning is stopped. In PCA technique, it is very difficult to obtain covariance of the matrix. Similarly the precise value of eigenvalues and eigenvectors is not possible due to computational error and hence the results are not satisfactory.

For best results author proposed new technique which has best results. Rest of the paper is organized as follows. Section II provides the information regarding the Principal Component Analysis Algorithm. Section III, IV and V presents the architecture of neural network and training algorithm. Experimental result is presented in section VII. Finally Section VIII presents conclusion and future work.

II. PRICIPAL COMPONENT ANALYSIS TECHNIQUE

The quality of images obtained from the satellite is mostly depends upon the conditions and environment in which the satellite works. These conditions are not favorable for the image reading. The images obtained from the satellite are noisy and require large memory space to store. To compress these images PCA based neural network model is used. This technique is also known as the Karhaunen Loeve transform, Hotelling Transform or proper orthogonal transformation. It is based on the factorization technique which is generally used in mathematics. PCA technique takes the collection of data and transforms it to the new data which has the same statistical properties. Transform is performing in such a way that it the originality remains at the end. In this data reduction technique, transform the data from n- dimensional space to m-dimensional space where m < n [3]. Generally data represented by set of m vectors:

$X = \{X_1, X_2, \ldots\ldots\ldots\ldots Xn\}$, Where each vector X_i has n element. The vector X_i is depending on application. For example in image compression each vector can represent a measure component of each vector features like color and size. Anyone can group together the features of an image by considering the element of each vector. Thus, the features column vector K for the data set

X can be represented as
$$C_{XK} = \begin{bmatrix} X_{1nK} \\ X_{2nK} \\ X3nK \\ . \\ . \\ X_{inK} \end{bmatrix} \qquad (1)$$

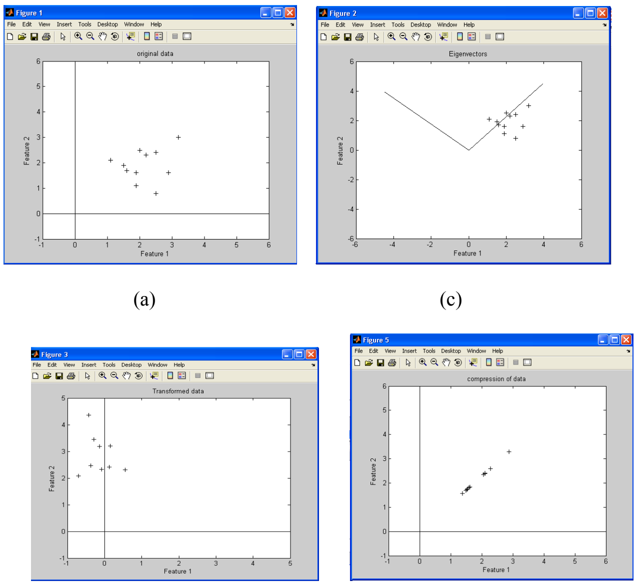
This approach requires the computation of the input data convergence matrix C_X and extraction of the eigenvalues and

978-0-7695-4201-0/10 $26.00 © 2010 IEEE
DOI 10.1109/ARTCom.2010.15

eigenvectors [2]. We can group such feature column vectors in a matrix form for easy processing. That is,

$$C_X = \lfloor C_{X,1}, C_{X,2}, \ldots\ldots C_{X,n} \rfloor \ . \qquad (2)$$

Fig.1 shows the example of original data and the eigenvectors where it has large values which are define by a line that goes across the points.



(a) (c)

(b) (d)

Fig .1. a) Original data c) Eigenvectors
b) Transformed data d) Compressed Data

Fig.1 shows the results obtained by transforming the features $C_Y = C_x W^T$. and eigenvectors become our main axes. For compression of data, we want to eliminate the component which has less importance that is less variation. So in this work data is reconstructed by setting to zero value. Fig.1 d) shows the compressed data using PCA.

The PCA method transforms the feature column vectors C_X to obtain the new vectors which give the better classification properties. PCA method confirms that the important data which account maximum variation measure by the covariance. The covariance measures the linear dependence between two random variables. Therefore, by computing the covariance, we can determine the relation between two data sets. If $X_i = \{X_{i,1}, X_{i,2}, \ldots\ldots\ldots X_{i,n}\}$, then the covariance is defined as:

$$\sigma_{X1,n} = E[(C_{x1} \quad \mu_{x1})\ldots(C_{xn} \quad \mu_{xn})] \qquad (4)$$

E[ ] is the average value of the element of the vector and μ_{xn} is the column vector obtained by multiplying the scalar value $E\lfloor C_{X,k} \rfloor$ by unitary vector.

It is important to note that the covariance measures a linear relationship between the two values of sets. The data may be related with each other in different ways. PCA method gives the simple solution in many applications like linear modeling, data compression, image compression, image restoration and pattern recognition etc.

The image data can also be compressed using the concept of data transformation. The PCA algorithm is summarized in the following steps [3].

1. Obtain the feature matrix Cx from the given image data. Each column of the matrix defines a feature vector.
2. Compute the covariance matrix $\bullet_X$.
3. Obtain the eigenvalues by solving the characteristics equation. These values forms the covariance matrix $\bullet_Y$.
4. Obtain the eigenvectors by solving characteristic equation for W_i in ($_i$**I** - $\bullet_X$) **W**i = 0 for each eigenvalues. Eigenvectors should be normalized.
5. Transformation W is obtained by considering the eigenvectors as their columns.
6. Obtained the transform features by computing $\mathbf{C}_Y = \mathbf{C}_x\mathbf{W}^T$. The new features are linearly independent.
7. For compression, reduce the dimensionality of the new feature vector by setting to zero components with low $_i$ values.

The dimensionality of input feature vector can be change by many ways. Here neural network is used to change the dimension of feature vector matrix as well as reconstruction of an original image. The threshold value plays an important role in PCA algorithm. The accuracy of the obtained results depends upon the threshold value at which the iteration process of learning is stopped. The feed forward back propagation neural network is used to obtain the desired results.

III. ARTIFICIAL NEURAL NETWORK ARCHITECTURE FOR IMAGE COMPRESSION

A neural network architecture shown in Fig.2 is suitable for solving the image compression problems. In this type of architecture the large input feed to a small hidden layer, which is further feed to large output layer. Such type of network is referred as feed forward neural network. One of the most important types of feed forward network is the back propagation neural network. It is a multi-layer feed forward network using extend gradient decent based delta learning rule commonly known as back propagation rule [7].

The input layer encodes the input and transmits the output of the hidden layer. The out layer receives the 16 hidden output of hidden layer and generates the 64 output. [8].

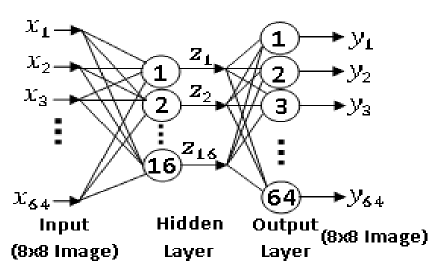


Fig .2. Architecture of feed forward neural network

IV. TRAINING ALGORITM

For processing of an image using neural network, it necessary to convert image into non-overlapping blocks of n x n pixels. In this paper 32 x 32 images is considered and subdivided into 8 x 8 pixels matrix. Each block represents N = n x n dimensional space. The goal of an image compression is to obtain the replica of original input image.

Each output unit receives a target pattern corresponding to an input pattern, error information term δk is calculated as:

$$\delta k = (t_k - y_k) f(y_{ink}) \tag{5}$$

And new updated weighted weights are,

$$W_{jk}(new) = W_{jk}(old) + \Delta W_{jk}\,, \tag{6}$$

$$W_{ok}(new) = W_{ok}(old) + \Delta W_{ok} \tag{7}$$

The weight correction term $\Delta V_{ij} = \alpha\delta_j x_i$

The bias correction term $\Delta V_{oj} = \alpha\delta_j$

Therefore New biases;

$$V_{ij}(new) = V_{ij}(old) + \Delta V_{ij}\,, \tag{8}$$

$$V_{oj}(new) = V_{oj}(old) + \Delta V_{oj} \tag{9}$$

V. PROPOSED TECHNIQUE AND ALGORITHM

The proposed feed forward back propagation neural network algorithm for image compression using neural network is as follows:

i) Image divided into Small 8x8 Chuck in matrix form.
ii) Normalized the values (0 to 255) of matrix in the range -1 to 1, which is called pixel to real mapping. Real values are converted to number of bits.
iii) Apply the feed forward back propagation training algorithm as explained in section IV.
iv) Once the training is complete, the 8x8 chucks are selected in the sequence.
v) Digital bits converted to real values.
vi) Matrix ranges from -1 to 1 reconverted from real to pixel mapping.
vii) Recall phase to demonstrate the decompressed image.

The training of the neural net precedes as follows, a 32X 32 and 256 X 256 training images are used to train the bottleneck type network to learn the required identity map. Training input-output pairs are produced from the training image by extracting small 8X8 chunks of the image chosen at a uniformly random location in the image. The real values 0 to 255 required 7 bits or more for representation in binary (1111111). The larger decimal values (0 to 255) are converted into smaller decimal values -1 to 1 and its equivalent binary representation. The number between -1 to 1 may be divided into 8 or 16 different values. 8 and 16 values needs 3 and 4 bits respectively. Thus the small decimal real value required less number of binary bits. For example: consider the first hidden output, when the value is between -0.25 and -0.50, then the code 010 is transmitted and when the value is between 1.00 and 0.75, the code 111 is transmitted. The pixel to real and real to pixel mapping is shown in Fig.3.

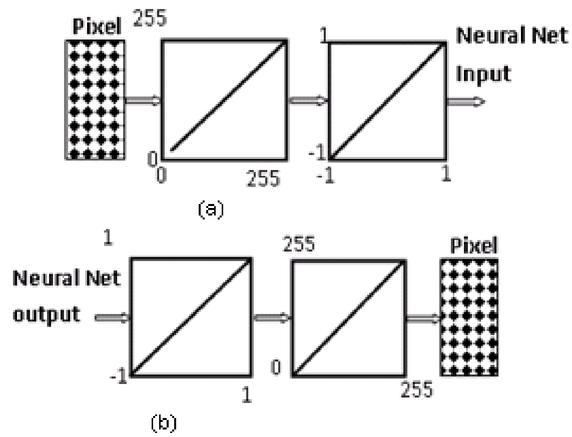


Fig.3. (a) Pixel to real conversion (b) Real to pixel conversion

VI. EVALUATION PARAMETERS

The quality of an image is measured using the mathematical expressions also. MSE, PSNR and CR are some parameters which define the quality of obtained image. The compression- decompression error is evaluated by comparing the input image and decompressed image using normalized mean square error formula [7]:

$$E = (t - y_{in})^2 \tag{10}$$

Where t is the target value and y_{in} is the net input to the output unit.

The phrase peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR is most commonly used as a measure of quality of reconstruction of lossy compression. The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs it is used as an approximation to human perception of reconstruction quality, therefore in some cases one reconstruction may appear to be closer to the original than another, even though it has a lower PSNR (a higher PSNR would normally indicate that the reconstruction is of higher quality). Mathematically PSNR is given by:

$$PSNR = 10\log_{10}\left(\frac{L^2}{MSE}\right)(dB) \tag{11}$$

Where L is the maximum value in pixels of an image.

VII. EXPERIMENTAL EVALUATION

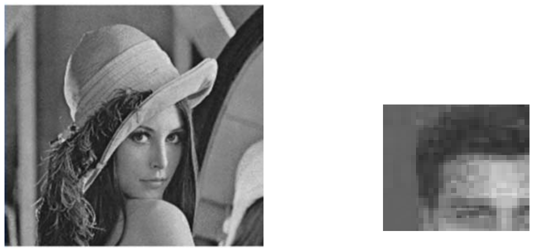
For various experiment in this paper, Leena 256.bmp and Lucky 32.bmp image is considered which are shown in Fig. 4, a and b. Specifications of images are obtained by reading images in Adobe Photoshop 7.0.1.

- Spécifications of image Leena 256.bmp:

Dimensions = 256 256 pixels,
Size: 64 K,
Resolution = 28.346 pixels /cm.
Width = 9.03 cm, Height = 9.03 cm.

- Spécifications of image Lucky 32.bmp:

Dimensions = 32 32 pixels,
Size: 1 K,
Resolution = 28.346 pixels /cm.
Width = 1.13 cm, Height = 1.13 cm



(a) (b)

Fig. 4. a) Leena 256.bmp b) Lucky 32.bmp

The input images shown in Fig.4 is first converted into 8 x 8 pixels blocks. The image compression is experimented with various numbers of neurons in the hidden layer along with the different parameters of the network. The network configuration with hidden layer neurons 8, 16 & 32 can be easily represented by 3, 4 and 5 bits respectively. Therefore 8,16 and 32 hidden neurons generally considered for experiments while 64 input neurons and 64 output neurons are used according to the requirement of selected input image pixels blocks size**.**

A number of experiments are carried out to test the PCA algorithm using feed forward back propagation neural network. For various experiment the common images are considered which are shown in Fig 4. The Epoch = 300 are considered, as it gives the best result at 300 Epoch for experiments.

Table 1
Results for 32 x 32 pixel Lucky 32.bmp image using PCA technique.

Hidden Neurons	Epoch	PSNR(dB)	Error
8	300	-30.2964	30.9774
16	300	-30.3337	30.9712
32	300	-30.3078	30.9772

Fig. 5. Decompressed image a) Leena 256. Bmp b) Lucky 32. Bmp

The results of 32 x 32 pixels Lucky 32.bmp and 256 X 256 Leena 256.bmp image using PCA technique are summarized in table 1. These are the measurements for constant values of alpha = 0.4 and mf = 0.6. The decompressed images for 32 hidden layer neurons are shown in Fig. 5.

A number of experiments are carried out to test the various configurations of neural networks for considered images. The blocks of 8 x 8 pixels are used in the experiment. The image compression is experimented with various numbers of neurons in the hidden layer along with the different parameters of the network. The network configuration with 1 to 16 neurons in the hidden layers are used while 64 input neurons and 64 output neurons are used according to the requirement of selected input image blocks sizes. It is not easy to find the covariance of a vector matrix in PCA technique. Similarly the precise value of eigenvalues and eigenvectors is not possible due to computational error and therefore data lost during transformation and calculation. Due to lose data the decompressed image result is not satisfactory.

From above results it is clear that the decompressed image is distorted due to the loss of data during PCA conversion. Also the error remain constant and therefore decompressed image for 8, 16 and 32 hidden neurons also approximately same which is shown in Fig.6. Fig.6 shows that the error E saturates and become constant which leads to more distortion in the output.

Now consider the Fig. 4 (a) and (b) as input image with same specifications as explain above. Table 2 shows experimental results for proposed feed forward back propagation algorithm. These are the measurements for constant values of alpha = 0.4 and mf = 0.6 for 300epoch.

Table 2
Results for 32 x 32 pixel Leena 32 image using proposed technique

Hidden Neurons	Epoch	PSNR(dB)	Error
8	300	80.5359	0.0049
16	300	81.3672	2.2553E-05
32	300	85.0179	5.1149E-06

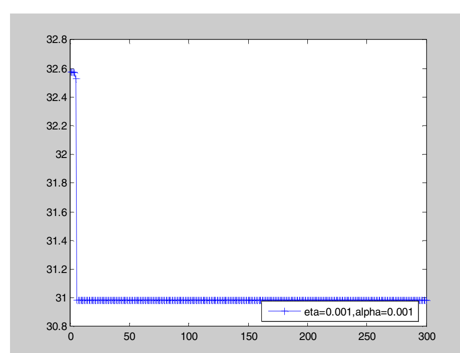


Fig. 6. Error E versus Epoch.

Fig. 7. Decompressed image a) Leena 256. Bmp b) Lucky 32. Bmp

Also Fig.7 shows the experimental result for 32 hidden layer neurons which is an obtained decompressed image.

The variation of error E with respect to number of Epoch performed is shown in Fig. 8. From above results it is clear that the proposed feed forward back propagation algorithm gives better resuslts than convensional PCA method.

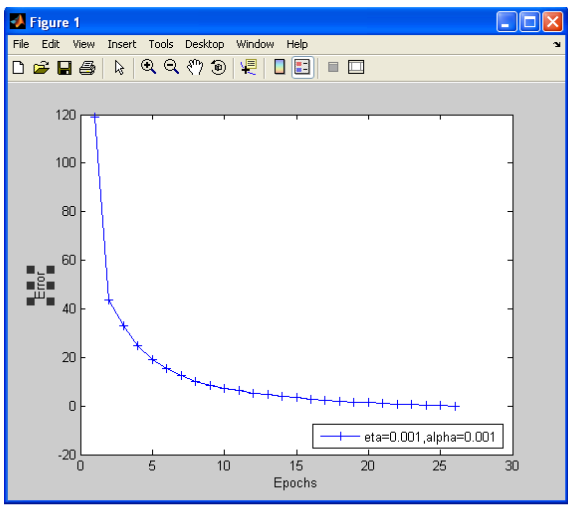


Fig.8 Error E versus Epoch

Both techniques can also be compared using the parameters speed, Error, PSNR with respect to the Epoch performed. Table 3 shows the comparison of PCA and Proposed technique. These reading are for Hidden neuron = 32, Alpha=0.6 and mf= 0.2. Figure 4 (a) is considered here. As the number of epoch increses, the error and time required for ituration reduced for proposed technique. For following results system with following specifications is used.

Intel (R) Core ™ 2 Duo CPU, T5670 @1.80 GHz, 789 MHz Processor.

Table 3
Comparison of PCA and Proposed technique

No. of Epoch	PCA Technique			Proposed Technique		
	Error	PSNR (dB)	Time (Min:Sec)	Error	PSNR (dB)	Time (Min:Sec)
10	32.9837	-30.8348	00:02.0	0.0025	86.6210	00:02.6
50	31.5755	-30.3351	00:02.9	2.1E-05	88.5439	00:04.8
100	30.4602	-30.2250	00:05.1	5.47E-06	89.3468	00:08.5
200	30.2200	-30.1612	00:09.2	2.53E-07	90.2951	00:15.8
300	30.2000	-30.1523	00:22.9	8.53E-07	90.9951	00:24.5
500	30.2000	-30.1523	00:36.9	2.10E-08	91.9942	00:03.6
1000	30.2000	-30.1523	01:14.4	4.82E-09	92.9900	01:12.6
1500	30.2000	-30.1523	01:55.2	8.06E-09	93.2123	01:48.8

VIII. CONCLUSION

The work presented consisting of PCA technique along with feed forward neural network that allows to concentrate on the original information contained in the remotely sensed images provided by the satellite. The accuracy of an obtained results is depends upon the threshold value at which the iteration process of learning is stopped. In this technique some of the information below the threshold value is removed or replace by zero and therefore more information removed from the feature vector matrix and hence from image data. In this paper an improved method of image compression using feed forward back propagation network is presented. The network is trained with blocks of image and tested. The result shows the good quality of reconstructed image. According to the obtained result it is concluded that is worth to continue developing new algorithm for image compression.

REFERENCES

[1] Jianxun MI, De-Shuang Huang, "Imag Compression Using Principal Component Neural Network," *8th Inernational Conference on Control, Automation, Robotics and Vision*, Vol. 1, pp. 698-701,6-9 Dec. 2004.

[2] Cebrail TASKIN, Serdar Kursat SARICAZ, "An Overviw of Image Compression approach," *Third international conference on Digital Telecommunications*, Vol. 1, pp. 174-179, IEEE 2008.

[3] S. Costa, S. Fiori, " Image Compression Using principal component neural networks," *Image and Vision Computing 19*, pp. 649-668, 2001.

[4] J. B. O. Souza Filho, L.P. Calba, J.M. Seixas, " An accurate and fast neural method for PA extraction," *International Joint Conference on Neural Networks-IJCNN*, 2003.

[5] S. Haykin, *Neural Networks: A comprehensive Foundation*, New York: MacMillan, 1994.

[6] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Third Edition, Pearson Education 2008.

[7] N Sivanandam, S Sumathi,S N Deepa, *Introduction to Neural Network Using MATLAB 6.0*, Tata Mc-Graw Hill Publication.

[8] Vijander Singh, Indra Gupta and H.O Gupta, "ANN Based Estimator for Distillation Using Levenberg-Marquardt Approach," *Engineering Applications of Artificial Intelligence*, International Journal of Elsevier Science, pp.249-259, 20, 2007.

[9] E. Oja and J. Koruna, "On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix," Journal of Mathematical Analysis and Applications, vol.104. pp.69-84,1985

[10] E Oja, "A simplified neuron model as a principal component analyzer," J. Math. Biol., vol. 15, pp. 267-273, 1982..