3rd International Conference on Electrical &Computer Engineering
ICECE 2004, 28-30 December 2004, Dhaka, Bangladesh

MODIFIED FREQUENCY SENSITIVE LEARNING ALGORITHM FOR IMAGE COMPRESSION

C. A. Al Sayeed, A. B. M. Ishteak Hossain and M. F. Uddin

Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology
E-mail: sayeedbuet@yahoo.com, ishtiupal@yahoo.com

ABSTRACT

At medium to low bit rates, Frequency Sensitive Competitive Learning (FSCL) algorithm is particularly effective for adaptive vector quantization in image compression systems. This paper presents a little modified FSCL scheme for the compression of grayscale still images in which the codebook is consecutively trained by six training images. This generated codebook is then used to decode an encoded test image *'Lena'*. Both SNR and PSNR and certainly the visual quality of the test image that we have achieved are found better as compared to the conventional FSCL scheme and most of the other existing methods.

1. INTRODUCTION

The core technique used for image compression consists of three important processing steps – pixel transformation, quantization and coding. Among the methods developed so far, vector quantization (VQ) is widely accepted as the method of data compression that provides a necessary tradeoff between speed and accuracy. VQ is mainly based on the representation of an input vector by the index of a codebook. A large number of algorithms exists for VQ. In this paper, we shall mainly discuss on the application of frequency sensitive competitive learning algorithm in performing VQ of images.

As a model to simulate the learning function of human brains, neural network approach remains very promising due to its adaptive methodology and massive parallel architecture. Several neural networks have been developed so far for image compression. Back-propagation (BP) is one of the neural networks, which are directly applied to image compression coding [2,3,4]. Such neural network structure consists of three layers: one input layer, one output layer and one hidden layer. Compression is achieved by designing the number of neurons at the hidden layer, less than that of neurons at both input and the output layers.

The basic back-propagation network is further extended to construct a hierarchical neural network by adding two more hidden layers into the existing network as proposed in [4].

Finite state vector quantization (FSVQ) is another scheme that provides better performance by incorporating memory into vector quantization as proposed in [5] where a distinct small codebook is associated with each state. The input vector is quantized using the current state codebook.

Kohonen's algorithm or self-organizing kohonen map [7] is another popular method for achieving the vector quantization of image compression. It realizes a mapping between an input and an output space that preserves the topology. This method is useful to increase the compression ratio significantly for a given image quality in a lossy compression scheme.

This paper is organized as follows: FSCL algorithm is first reviewed and then our proposed training scheme for codebook generation is presented. Simulation results for the test image *'Lena'* are presented in the next section. The obtained results are then compared with different existing methods. The final section concludes the paper.

2. FREQUENCY SENSITIVE LEARNING ALGORITHM

There are several existing algorithms for competitive learning, such as unsupervised, supervised and differential competitive learning – all vary in the definitions of the learning law [6]. A major problem of using a basic neural network is that some of the neural units may be underutilized. As described in [6], the frequency sensitive competitive learning (FSCL) algorithm is actually a modification of the unsupervised competitive learning (UCL) algorithm. It modifies UCL by introducing an additional

ISBN 984-32-1804-4

parameter called 'winning frequency'. The upper limit associated with this parameter provides the regularity in updating the synaptic vectors.

In the FSCL method [8], a codebook is generated for vector quantization during a single pass through the training set. At first, the codebook can be initialized at random or from the first M input vectors of the training image where M represents the codebook length. The winner is selected by choosing the neuron with minimum distortion and its output is set at high. The synaptic vectors are then updated with the training rules as described in [6,8]. In this way, only the winning synaptic vectors are updated. The training rule moves it toward the current training vector by a fractional amount. The training of the winner is performed until its winning frequency exceeds the upper threshold frequency. The actual selection of the threshold frequency depends on the input data statistics and is normally chosen to be three to four times larger than the average winning frequency over the entire training set.

2.1 Codebook Generation

The principal task in FSCL algorithm is to generate the codebook. In conventional FSCL design, codebook is initialized using the first M input vectors from the training image, where M is the length of the codebook. In this paper, our concept is to generate a highly enriched codebook using as much training image as possible. We can send this codebook to different receiving points those are expecting to receive an image data. We can then simply encode our image and send that encoded image to different receivers. At the receiving end, the receiver will be able to reconstruct the image easily using the vector indices of the encoded image from the codebook that was received earlier.

In this paper, we have generated our codebook using six grayscale training images e.g. *pepper*, *mountain*, *mandrill*, *zelda, frog* and *barb.* At first, we have trained our codebook using training image '*pepper*' and then the trained codebook is updated using the next training image '*mountain*' and so on. However, all the training and test images we used are of size 512 x 512 and are in 'png' format. The images are encoded using the one iteration FSCL with the upper threshold frequency 600.

3. SIMULATION RESULTS

Computer simulations were performed using MATLAB 5.3 in order to implement the FSCL scheme. Codebooks were generated for different block sizes and lengths and the trained network is then tested on test image '*Lena'*. To evaluate the performance of the FSCL scheme, we mainly discussed two parameters: SNR (signal to noise ratio) and PSNR (peak signal to noise ratio).

Table 1: Simulation results for the test image 'Lena'

BLOCK SIZE	NO. OF BITS	COMPRESSION RATIO	SNR	PSNR
3 X 3	10	1.1111	26.9923	32.8522
	9	1.0000	26.3740	32.2339
	8	0.8889	26.0487	31.9086
	7	0.7778	24.6991	30.5590
4 X 4	10	0.6250	24.9672	30.8271
	9	0.5625	24.5744	30.4343
	8	0.5000	23.8002	29.6601
	7	0.4375	23.0306	28.8905
5 X 5	10	0.4000	23.0157	28.8756
	9	0.3600	22.7070	28.5669
	8	0.3200	22.2200	28.0799
	7	0.2800	21.8018	27.6618
6 X 6	10	0.2778	21.7677	27.6276
	9	0.2500	21.6153	27.4752
	8	0.2222	21.2862	27.1462
	7	0.1944	20.4879	26.3479

The signal to noise ratio or SNR is defined as

$$\text{SNR}=10\log_{10}\frac{E[\| X \|^2]}{E[d(X,\hat{X})]}\ \text{dB}.$$

Here, the numerator defines the estimation or summation of all the squared matrix elements of matrix X (original image) and the denominator measures the product of the difference of vector X and $\hat{X}$ (decoded image) and its conjugate.

However, the most acceptable measurement is done to evaluate the quality of the reconstructed image with PSNR (peak signal to noise ratio). For a grayscale image with n blocks of size $N,$ the PSNR is given as follows:

$$\text{PSNR}=10\log_{10}\frac{255^2}{\frac{1}{nN}\sum_{i=1}^{n}\sum_{j=1}^{N}\left(\left\|X-\hat{X}\right\|^2\right)}\ \text{dB}$$

At first, we have fixed the block size at 3x3 and performed the simulation for different lengths of the codebook (from M=2^7=128 to M=2^{10}=1024). The generated codebook was then used for encoding and decoding the test image '*Lena'*. The whole procedure was repeated for block sizes 4x4 to 6x6.

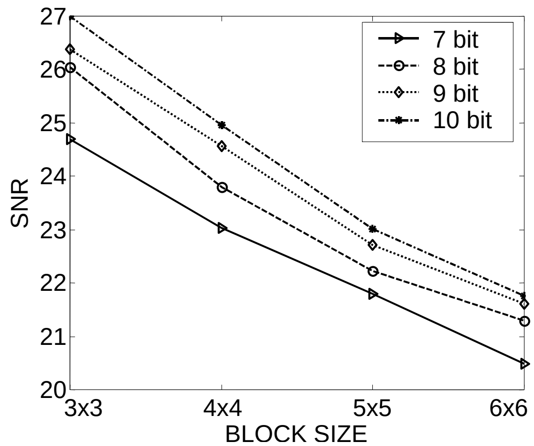

Fig. 1 SNR vs. block size for different codebook lengths for image Lena

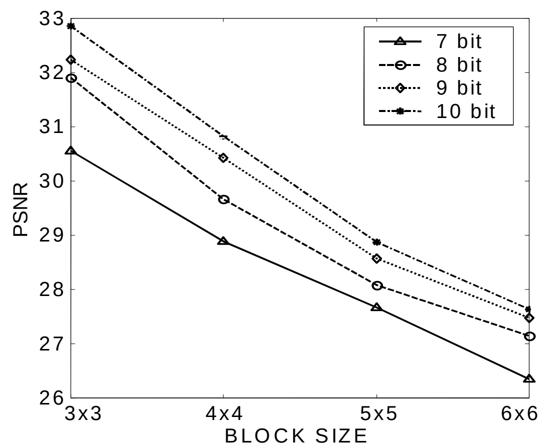

Fig. 2 PSNR vs. block size for different codebook lengths for image Lena

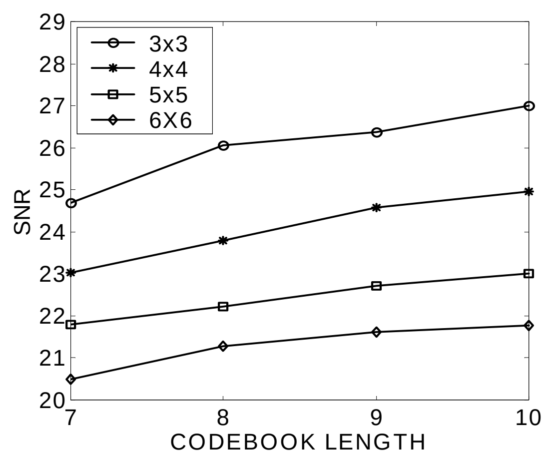

Fig. 3 SNR vs. block size for different codebook lengths for image Lena

As seen from fig. 1 and fig. 2, both SNR and PSNR of the decoded image decrease significantly as the

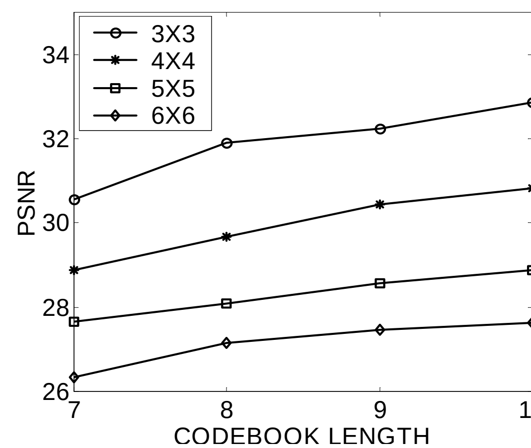

Fig. 4 PSNR vs. block size for different codebook lengths for image Lena

block sizes are increased for a fixed codebook length. Again, from fig. 3 and fig. 4, it is clear that they increase slightly as the codebook lengths are increased for a fixed block size. Hence, we can say that using this FSCL method, the image can be compressed significantly with a slight loss in SNR and PSNR. For a block size of 5x5, as the codebook length is reduced from 10-bit to 7-bit, the PSNR is reduced by around 1.04 times for the test image *'Lena'* but at the same time the compression is achieved by around 1.43 times.

4. COMPARISON WITH OTHER METHODS

Table 2 shows the performance comparison of our modified FSCL method with different existing methods for the test image Lena.

Table 2: Performance comparison for the test image Lena

Compression Ratio (bpp)	Method	Performance
0.24	FSVQ [5]	PSNR = 27.47 dB
0.25	SOM [7]	PSNR = 24.70 dB
0.25	Our result	PSNR = 27.48 dB
0.75	Basic BP [1]	PSNR = 26.17 dB
0.66	Adaptive BP [1]	PSNR = 27.93 dB
0.625	Our result	PSNR = 30.83 dB
1.00	Hierarchical BP [1]	SNR = 25.670 dB
1.00	FSCL [8]	SNR = 25.230 dB
1.00	Our result	SNR = 26.374 dB

Fig. 5 Original image Lena (512 x 512)

Fig. 6 Encoded image Lena with PSNR = 31.9086 dB at Compression Ratio = 0.8889 bits/pixel

Fig. 7 Encoded image Lena with PSNR = 29.6601 dB at Compression Ratio = 0.50bits/pixel

As seen from Table 2, our FSCL method is found truly excellent as compared to the other existing algorithms like self-organized kohonen mapping and different back propagation algorithms.

Fig. 8 Encoded image Lena with PSNR = 27.1462 dB at Compression Ratio = 0.2222 bits/pixel

5. CONCLUSION

In this paper, we presented the FSCL scheme in a modified way by generating a codebook using several grayscale still images and then applied the network on a test image. Definitely, our results outperform some of the well-known image compression schemes for the same compression ratio in both PSNR and SNR and obviously in visual quality. And also as we have seen earlier, we can achieve a significant compression ratio with a slight loss in SNR and PSNR.

REFERENCES

[1] J. Jiang, "Image compression with neural networks – a survay," *Signal Processing: Image Communication* 14, pp. 737-760, 1999.

[2] S. Carrato, "Neural networks for image compression, Neural Networks: Adv. and Appl. 2 ed.," *Gelenbe Pub,* North-Holland, Amsterdam, 1992, pp. 177-198.

[3] M. Mougeot, R. Azencott, B. Angeniol, "Image compression with back propagation: improvement of the visual restoration using different cost functions," *Neural Networks* 4 (4) (1991) pp. 467-476.

[4] A. Namphol, S. Chin, M. Arozullah, "Image compression with a hierarchical neural network," *IEEE Trans. Aerospace Electronic Systems* 32 (1), pp. 326-337, January 1996.

[5] R. Arvind and A. Gersho, "Image compression based on vector quantization with finite memory," *Opt. Eng.,* vol. 26, no. 7, pp. 570-580, July 1987.

[6] B. Kosko, "Neural networks for signal processing," *Prentice Hall,* 1992.

[7] C. Amerijckx, P. Thissen, "Image compression by self-orgainzed kohonen map," *IEEE Transactions on Neural Net.*, vol. 9, no. 3, pp. 503-506, May 1998.

[8] M. Shnaider, A. P. Paplinski, "Frequency sensitive competitive neural network with an application to image compression," *Tech. Report 94-2, Monash Univ.* June 1994.