COMPARING ARTIFICIAL NEURAL NETWORKS (ANN) IMAGE COMPRESSION TECHNIQUE WITH DIFFERENT IMAGE COMPRESSION TECHNIQUES

Bekir KARLIK*, Serkan AYDIN*, İlker KILIÇ* * Celal Bayar U., Electric-Electronic Eng. Dep., Manisa

Abstract : In this paper it is presented that, 256x256 16 gray level images can be compressed fast and efficiently by using neural networks . Compression results were compared with the other methods according to mean square error (MSE) and visual of the image and it is seen that by further works on ANN the images can be compressed better.

1. INTRODUCTION

An 640x480, 256 gray level image needs about 310 Kbytes memory. According to the standard of Europe CCIR or USA-RS170 monochrome, an image is renewed 25 times in a second. 640x480, 30 video images are stored into the video RAM in a second by using a scorpion card. This means in real time we have to transfer 9 Mbytes data into the computer memory in a second. It is quite difficult almost impossible. But if we achieve this work we need 3.2 GBytes memory for an hour video signal. So it is not practical. We have different image compression algorithms as JPEG and MPEG that the compressed image data needs less memory and the compression work doesn't take much time.

Image compression and reconstruction are important problems. An other problem is decreasing of the compression and reconstruction time duration. Especially, in real time, decreasing the computation time is an important advantage.

The purpose of image compression is decreasing the memory where the compressed image is stored in. The reconstructed image data quality must be reasonable.

2. IMAGE COMPRESSION TECHNIQUES

There are quite a lot of methods about image compression. These methods are divided in two classes ; lossless and lossy. In lossless algorithm like Huffman coding the compression ratio is limited and very low. In lossy compression algorithms we can obtain high compression rates but by the way the MSE also increases.

Run-length coding, vector quantization [1] [7], transform coding [2], predictive coding [3] and block truncation coding [8] are some of the lossy compression techniques.

In run-length algorithm first pixel value is assumed as the reference .If the following pixel values do not exceed a threshold, then run counter is increased one for each pixel. Otherwise "1" is loaded into the run counter and the following first pixel becomes the new reference value. Because there are likely to be runs of length>>1 in the compressed image representation, the resulting compressed data set will be smaller than the raw image data. These effects are highly dependent on the threshold T, which must be chosen to yield both a reasonable image representation(i.e., the reconstructed image should appear subjectively

"close" to the uncompressed image and a reasonable number of runs, each consisting of a reasonable number of pixels.

In Run-length method if the compression ratio is increased, horizontal lines are seen on the image causing noise.

In Vector Quantization the image is formed with two dimensional blocks. Each block is represented by a vector. All vectors of this algorithm are selected from a codebook table which is formed before the compression. The problems of this algorithm are block effect and edge distortion.

In Discrete Fourier Transform algorithm again the image is formed by two dimensional blocks. Discrete Fourier Transform of each block consist of coefficients. Number of these coefficients is as same as the number of pixels in each block. Each coefficient is consist of real and imaginary parts. These coefficients are stored into the memory one by one in the zig-zag order. In the compression algorithm some of these coefficients are ignored and the rest of the coefficients are stored into the memory. So the compression is achieved. The reconstructed image quality is dependent on the amount of ignored coefficient. The more coefficients are used in the compression the better image can be obtained in the reconstruction algorithm. Equation 1 presents the DFT (Discrete Fourier Transform) and equation 2 presents the IDFT (Inverse Discrete Fourier Transform)

$$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \exp\left[-j2\pi(ux+vy)/N\right]$$
(1)

$$u,v = 0,1,2,...,N-1$$

$$f(x,y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \exp\left[j2\pi(ux+vy)/N\right]$$
(2)

$$x,y = 0,1,2,...,N-1$$

In figure 2 the zig-zag order is shown. The fist coefficient of the zig-zag order is called DC value the rest of the coefficients are called AC values. DC value consist of more image information than the AC values do.

As the equations exhibit, DFT and IDFT need quite amount of calculations while Discrete Cosine Transform (DCT) doesn't. The formulation of DCT and IDCT can be seen in Eq. 3 and Eq. 4.

3. ANN APPROACH FOR THE IMAGE COMPRESSION

In this study, image compression and reconstruction were performed by using ANN. 256x256 digital gray-level images were used. Images were compressed and reconstructed by using two different ANN stage, as shown in Fig. 3.

A multi-layered and feedforward ANN network, which has error-backpropagation algorithm, were used in this paper (shown in Fig. 4). ANN architecture is in the form of

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{N}} & u = 0\\ \sqrt{\frac{2}{N}} & u = 1,2,3 \dots N-1 \end{cases}$$
(3)
$$\alpha(v) = \begin{cases} \frac{1}{\sqrt{N}} & v = 0\\ \sqrt{\frac{2}{N}} & v = 1,2,3 \dots N-1 \end{cases}$$

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$
$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{N}} & u = 0\\ \sqrt{\frac{2}{N}} & u = 1,2,3 \dots N-1 \end{cases}$$
(4)
$$\alpha(v) = \begin{cases} \frac{1}{\sqrt{N}} & v = 0\\ \sqrt{\frac{2}{N}} & v = 1,2,3 \dots N-1 \end{cases}$$

64:70:4:70:64. Compressed data were obtained from the hidden layer, which has four neurons. Learning speed and rate were taken 0.9 and 0.7, respectively. Network architecture in Fig. 4 was used for teaching. After the teaching phase, the architecture was split into two part; the first part was used for compression and the other part was used for reconstruction. In Fig. 5 original (left) the axial cranium section passing through sifenoidal sinus (in bone algorithm, in pontin sifenoid and in two temporal, there is subarachnoidal air collections) and (right) the axial cranium section passing through the venticular plate (the algorithm in soft tissues, mass is observed in gray matter, white matter and intraventicular) images are shown. Images, compressed by using ANN at a rate of R=0.5 bpp, are seen in Fig. 6. Run-length coding, vector quantization and DFT results are given in Fig. 7, 8, 9, respectively. Compression results from ANN were taken in real-time. If a codebook, which is formed by 16 elements, is used then the MSE and SNR values are 9.792, and 0.589302 for the image on the left handside of the page, respectively. For the image on the right handside of the page, these values are 8.859 and 1.04087. MSE formulation is shown in Eq. 5 and SNR formulation is given by Eq. 6.

$$MSE = \frac{1}{N*N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y) \right]^2$$
(5)

$$SNR = 10Log \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) \right]^2}{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y) \right]^2}$$
(6)

4. RESULTS

This study shows that ANN can be used for image compression efficiently. Reconstructed images, which have been compressed with run-length coding algorithm, is shown in Fig. 7. But, there are horizontal lines in this reconstructed images. In Fig. 8 a vector quantization study with R=0.5 bpp, is shown. It is clearly seen that, there is a block-effect in this Figure. In Fig. 9, DFT results are shown. In this algorithm again compression ratio is R=0.5 bpp. It is obvious that, compression with ANN is better in keeping most of the original data about the image.

In forward works; enhancing the quality of reconstructed images with ANN and analyzing ANN results for 256 gray level images can be done.

REFERENCES

1. Gray R. M., "Vector Quantization", IEEE ASSP Mag., vol :1, pp : 4-29, April 1984.

2. Torres-Urgell L., Kirlin R. L., "A New Adaptive Method for Image Compression Using Karhunen-Loeve Transform", in Proc. Int. Conf. on Acoustics, Speech and Signal Proc., pp : 32.3.1-32.3.4, 1987.

3. Ozturk, Y., Abut H., "Multichannel Linear Prediction and Applications to Image Coding" Archiv fur Elektronik und Ubertragungstechnik, AEU Band 43, Heft 5, 1989.

4. Delp E. J., Mitchell O. R., "Image Compression Using Block Truncation Coding", IEEE Trans. on Communication, pp : 532-540, April 1983.

5. Karlık B., "Çok Fonksiyonlu Protezler İçin Yapay Sinir Ağları Kullanarak Miyoelektrik Kontrol", Doktora Tezi Y.T.Ü. Fen Bilimleri Enstitüsü, İstanbul, s. 22-40, Mart, 1994 6. Karlık B, Aydın S., Kılıç İ., "Yapay Sinir Ağları İle Görüntü Sıkıştırma ve Yeniden Oluşturma",

Harran Üniversitesi Gap 1. Mühendislik Kongresi-1996 Bildiriler Kitabı, pp:626-632, Mayıs, 1996.

7. Vijay S. Sitaram, Chien-Min Huang, and Paul D. Israelsan, "Efficient Codebooks for Vector Quantization Image Compression with an Adaptive Tree Search Algorithm", IEEE_Trans. on Communication, Vol. 42, No : 11, pp : 3027-3032, November, 1994.

8. Sherif A. Mohamed and Moustafa M. Fahmy, "Image Compression Using VQ-BTC", IEEE Trans. on Communication, Vol. 43, No : 7, pp : 2177-2182, July, 1995.





Figure 1. 8×8 pixels two dimensional image block





Figure 3. Block diagram of ANN technique for image compression and reconstruction



Figure 4. ANN network architecture.



Figure 5. Original images.



Figure 6 Reconstructed images by using ANN with R=0.5 bpp.



Figure 7 Reconstructed images of run-length coding (R=0.5 bpp).



Figure 8 Reconstructed images of vector quantization (R=0.5 bpp).



Figure 9 Reconstructed images of DFT (R=0.5 bpp).