

POST-PROCESSING BLOCK CODED IMAGES USING ARTIFICIAL NEURAL NETWORKS

G. Qiu[⊙], Z. He^{*} and S. Chen^{*}

[⊙]University of Derby, UK

^{*}University of Portsmouth, UK

Abstract: In this paper, a technique employing artificial neural networks for post-processing block coded images is presented. Visually important image features are extracted from the de-compressed image and used as input to a feedforward neural network. The neural network learns to reconstruct the difference image between the original (uncompressed) and the de-compressed image. Coding artifact reduction is achieved by adding the neural networks output to the de-compressed image. Experimental results using the new technique for post-processing quadtree coded images are presented. It is shown the new technique can significantly improve the compressed image both in terms of peak signal to noise ratio (PSNR) and visual quality of the image.

1. INTRODUCTION

Image coding techniques, such as JPEG [1] and Vector Quantization [2] are "block based", in the sense that the original images are divided into small blocks, for example 8 x 8 pixel sub-images, and code the blocks individually. One major defects for these type of coding schemes is the "blocking effects", i.e., the edges of the blocks are visible in the decoded images, especially at high compression ratios. Efforts have been made in the past to alleviate this annoying problem, different degrees of success have been reported [3, 4].

Recently, neural networks have been shown to be able to reconstruct visual images by using visually important image features, such edges [5, 6]. These approaches can also be used for image coding [7]. In [6] multi-resolution features, edge information from different resolution scales of the image have been used for the reconstruction of visual image. In fact, the neural networks were used for recovering the high frequency contents (edges),

and the low frequency contents were coded separately.

In many image coding methods, such as mean removal Vector Quantization and Block Truncation Coding schemes [8], the low frequency components, the block means, are coded separately. In transform based coding methods, such as JPEG, low frequency components are given priority. At high compression ratios, the high frequency components are not adequately coded, therefore blocking effects occurs because the abrupt changes in block intensities. Human visual system is also sensitive to multi-resolution features [9], whilst in block coded images, this factor has not been taken into consideration (emerging techniques such as wavelet coding [10] is multi-resolution based coding method and blocking effect is not a problem).

The ultimate aim of image compression is to use as little data as possible to represent an image. If lossless compression techniques are used, the decoded image will be identical to the original image. In most cases, compression are lossy, the decoded image will differ from the original image. The smaller the difference, the better is the quality of the coded image. There is a trade off between quality and compression ratio, the higher the compression ratio, the poorer is the quality of the coded image. The goal of post processing is to improve the quality of the image without incurring bit rate overheads.

A method utilizing neural networks for improving block encoded images is presented in this paper. The method extracts *visually important* image features from decoded images, and no extra bits are required. These visually important features are used as input to a multi-layer feedforward neural network which learns from these features to reconstruct the difference image between the original and the de-compressed image [5, 6]. As

an application example, we present results of applying this technique to process quad-tree coded images.

2. ANN FOR POST-PROCESSING BLOCK CODED IMAGES

We have developed a neural network based technique for post-processing block coded images. In neural network applications, the first task is to decide which neural network model to use and define the network architecture. Second, to collect sufficient training samples. Third, to train the network use appropriate training algorithms. Finally, apply the trained network. We will describe these steps in sequence in the context of processing de-compressed images.

2.1 Systems Architecture

A three layer feedforward network will be used. The input layer accepts visually important features extracted from de-compressed images. The output of the network will be added to the de-compressed image to improve its quality. The number of input neurons corresponds to the number of visual features used, and the number of output neurons corresponds to the number of pixels in the block used. Notice here the block size we refer to is different from those used by the image coders, normally, our block size should be at least four times as large as those used by the coder, and ideally should be as large as possible. However, too large a block size would make computation and storage impractical. A schematic diagram of the systems architecture is shown in Fig. 1.

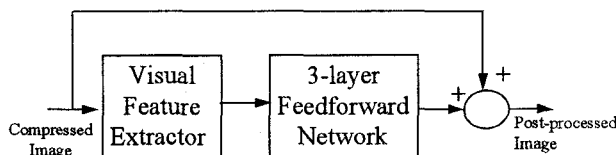


Fig. 1 Schematics diagram of the post-processing system

2.2 Collecting Training Samples

The feedforward networks we shall be using are trained by supervised learning, therefore, apart from input, the corresponding desired output needs

to be defined as well. The images are first compressed to a given bit rate, and then de-compressed. Let $B(x, y)$ be a $2^c \times 2^c$ block in the original image, $B_r(x, y)$ be the decoded corresponding block, and $D(x, y) = B(x, y) - B_r(x, y)$. The following multi-resolution features are extracted from $B_r(x, y)$ as input to the network: $2^{c-1} \times 2^{c-1}$ pairs of directional derivatives of 2×2 blocks, $2^{c-2} \times 2^{c-2}$ pairs of directional derivatives of 4×4 blocks, ..., and 1 pairs of directional derivatives of $2^c \times 2^c$ block. For example, if 16×16 blocks are used, there will be 64 pairs of directional derivatives of 2×2 blocks, 16 pairs of directional derivatives of 4×4 blocks, 4 pairs of directional derivatives of 8×8 blocks, and 1 pairs of directional derivatives of 16×16 block. The pair of directional derivatives are the horizontal and vertical edge information calculated by the convolution of the following masks and the image blocks.

$$G_h(i) = \begin{vmatrix} 1(2^i) & -1(2^i) \\ 1(2^i) & -1(2^i) \end{vmatrix} \quad G_v(i) = \begin{vmatrix} 1(2^i) & 1(2^i) \\ -1(2^i) & -1(2^i) \end{vmatrix}$$

where $i = 0, 1, \dots, c-1$, and $1(2^i)$ is a $2^i \times 2^i$ matrix with all of its elements equal to 1. Using these features as input, the corresponding desired output for supervised learning is the difference image block $D(x, y)$. We shall use as many image as possible to collect sufficient training samples.

2.3 Training the Network

Once the block size used is decided, the network architecture is decided, the number of hidden neurons are decided through experiment. The backpropagation algorithm [11] is used to train the network.

2.4 Utilizing the Network

Once the network is trained, its architectural information and the values of the weights are fixed and used with the decoder. After an image is decoded, the visual features are extracted as described in 2.2, and input to the network, the output of the network is added to the decoded image.

3. APPLICATION TO QUAD-TREE CODED IMAGES

The post-processing scheme described in Section 2 has been applied to quad-tree coded images [12]. We use four 512 x 512, 8-bit per pixel gray scale images in our experiment. We use three to collect training samples, and the fourth is used as testing image. Table 1 shows the PSNR performance improvements of quad-tree coded images at the bit rate of 0.5 bit per pixel. The result for each image represents testing result, in which the other three images are used to train the network, and the current image is used to test the trained network. It is seen that the improvements are quite significant. Also, the PSNR improvements are quite consistent for different images, this means the network "generalize" well. Apart from PSNR improvements, the visual quality of the images has been improved accordingly as shown in Fig. 2.

4. CONCLUDING REMARKS

We have shown in this paper the usefulness of artificial neural network technology in processing block coded images. The results are encouraging. However, the current scheme is computationally very intensive. Because neural networks are intrinsically parallel, special hardware can be designed to speed up the computation. In future research, we will investigate methods to reduce the computational costs of the technique and apply the technique to vector quantization and JPEG coded images.

References

1. W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993
2. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992
3. D. G. Sampson, D. V. Papadimitriou, M. Ghanbri and T. J. Dennis, "A method for enhancing the picture quality of low bite-rate block coded images", Proc. of the 5th IEE International Conf. on Image Processing and It Applications, Heriot-Watt University, Edinburgh, UK, July 1995, pp. 40- 44
4. B. Ramamurthi and A. Gersho, "Nonlinear space-variant post-processing of block coded images", *IEEE Trans Acoustic, Speech and Signal Processing*, Vol. 34, pp. 1258-1267, 1986
5. G. Qiu, "Reconstruction of Images using an Artificial Neural Network with Local-feature Extraction", Proc. of SPIE, Vol. 2353, *Intelligent Robots and Computer Vision XIII: Algorithms and Computer Vision*, Boston, MA, USA, pp. 460-468, 1994
6. G. Qiu, "Modelling the Visual Cortex using Artificial Neural Networks for Visual Image Reconstruction", Proc. of IEE 4th International Conference on Artificial Neural Networks, Churchill College, Cambridge, UK, June 1995, pp. 127 - 132
7. G. Qiu, M. R. Varley and T. J. Terrell, "Image Coding based on Visual Vector Quantization", Proc. IEE 5th International Conference on Image Processing and Its Applications, Heriot-Watt University, Edinburgh, UK, July 1995, pp. 301 - 305
8. Qiu, M. R. Varley and T. J. Terrell, "Improved Block Truncation Coding using Hopfield Neural Network", *IEE Electronics Letters*, Vol. 27, No. 21, pp. 1924-1926, 1991
9. D. Marr and E. Hildrith, "Theory of edge detection", Proc. Royal Society, London, B207, pp. 187-217, 1980
10. M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform", *IEEE Trans. Image Processing*, Vol. 1, pp. 205-220, 1992
11. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation", in *Parallel Distributed Processing*, Vol. 1, Chapter 8, D. E. Rumelhart and J. L. McClelland (Eds.), Cambridge, MA, MIT Press, 1986
12. E. Shusterman and M. Feder, "Image compression via improved quadtree decomposition", *IEEE Trans. Image processing*, vol. 3, no.2, pp. 207-215, 1994

Table 1 PSNR Improvements after post-processing, bit rate = 0.5 bpp

Images	Quad-tree Coded	Post-processed	Improvements
Lena	32.38 dB	33.30 dB	0.92 dB
Peppers	33.28 dB	34.15 dB	0.87 dB
Airplane	32.86 dB	33.86 dB	1.00 dB
Sailboat	29.53 dB	30.48 dB	0.95 dB



(a)



(b)

Fig. 2. (a) Quadtree coded Lena image at 0.5 bpp, (b) ANN processed Lena image.