MLP for Adaptive Postprocessing Block-Coded Images

Guoping Qiu, *Member, IEEE*

***Abstract*—A new technique based on the multilayer perceptron (MLP) neural network is proposed for blocking-artifact removal in block-coded images. The new method is based on the concept of learning-by-examples. The compressed image and its original uncompressed version are used to train the neural networks. In the developed scheme, inter-block slopes of the compressed image are used as input, the difference between the original uncompressed and the compressed image is used as desired output for training the networks. Blocking-artifact removal is realized by adding the neural network's outputs to the compressed image. The new technique has been applied to process JPEG compressed images. Experimental results show significant improvements in both visual quality and peak signal-to-noise ratio. It is also shown the present method is comparable to other state of the art techniques for quality enhancement in block-coded images.**

***Index Terms*—Block coding, image coding, image enhancement, JPEG, neural network, postprocessing.**

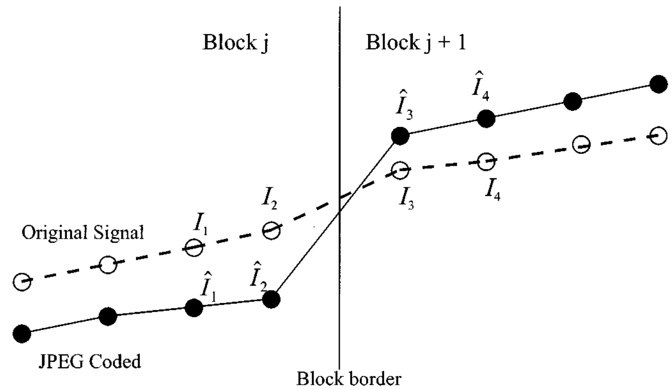

Fig. 1. Abrupt changes in block-border pixel values after quantization causes blocking artifacts.

I. Introduction

Many well-known image-compression techniques such as JPEG [1] are block based. In these techniques, an image is partitioned into small blocks (typically 8×8) and each block is coded independently. However, at low bit rates, the reconstructed images generally suffer from visually annoying artifacts due to very coarse quantization. One major such artifact is the blocking effect, which appears as artificial boundaries between adjacent blocks.

Although emerging image-compression techniques such as wavelet transform [2] do not suffer from blocking artifacts, no international standard based on this new technique exists at this stage, and software implementation is not yet widely available to ordinary image users. On the other hand, JPEG has been international standard for many years, and its implementation software is available in a variety of application environments. To date, JPEG is a widely used image-compression tool, and we believe it will continue to be so in the foreseeable future. Based on this rationale, we continue the effort to investigate new methods to improve the quality of block-based image-compression methods (JPEG is a special case).

There are many techniques developed to reduce the blocking effect. Some use image filtering techniques [3]–[5], some formulate the blocking-effect removal as an image restoration problem [6], and yet others use the theory of projections onto convex set (POCS) to process block-coded images [7].

In this paper, we investigate a new technique for artifact removal in block based image coding. The technique is based on the concept of *learning by examples* and implemented using multilayer perceptron (MLP) neural networks [8]. Unlike previous methods, we make explicit use of the original (uncompressed) image and use it to train the networks. Once trained, these networks are used to remove blocking effects without unnecessary blurring the images. Simulation results show that the new method improves PSNR and visual quality of JPEG compressed images and its performance is comparable to that of other known blocking removal methods.

The rest of the paper is organized as follows. In Section II, the problem of blocking artifact is first described, then a technique for blocking-effect removal using MLP networks is introduced. Section III presents simulation results on JPEG-compressed images, and concluding remarks are given in Section IV.

II. Blocking-Artifact Removal Based on Adaptive Learning

A. Problem Statement

In most natural image signals, the intensity values of neighboring pixels tend to change slowly. Although step edges exist in natural images, they are by and large rare, and the chances that natural step edges coincide with the block borders are very small. In block-based image-coding schemes, individual blocks are quantized independently, this can result in abrupt changes in pixel intensities in the block borders, hence causing blocking artifacts. Fig. 1 shows a typical situation that may cause blocking

Manuscript received March 14, 1997; revised August 16, 2000. This paper was recommended by Associate Editor T. Chen.

The author is with the School of Computing and Information Technology, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, U.K. (e-mail: g.qiu@cs.nott.ac.uk).

Publisher Item Identifier S 1051-8215(00)10633-0.

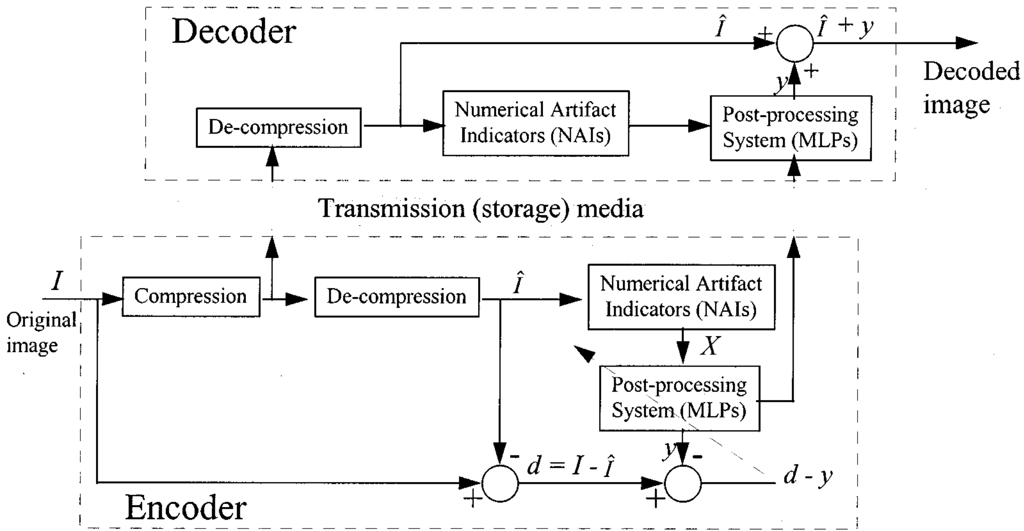

Fig. 2. A new technique for blocking-artifact removal based on adaptive learning.

TABLE I
PSNR OF JPEG COMPRESSED IMAGES, QUALITY FACTOR $q = 15$

Images	bit rate (bpp)	PSNR (dB)
Boats	0.30	31.02
F16	0.27	32.09
Girl	0.29	32.59
Lena	0.27	31.94
Peppers	0.24	32.31

TABLE II
PSNR IMPROVEMENTS FOR A SYSTEM TRAINED ON THE BOATS IMAGE

Images	PSNR (dB)	Improvement
Boats	31.51	0.49
F16	32.70	0.61
Girl	33.29	0.70
Lena	32.51	0.57
Peppers	32.79	0.48

effects. Bear in mind that our motivation is to construct an adaptive learning system to remove these abrupt changes; it is therefore necessary to represent the existence of such an artifact numerically. Given the nature of the problem, precise measurement of the artifact is almost impossible to define. On the other hand, there is also no need for a numerically precise measurement because the way in which the human visual system responds to visual signals is imprecise. For example, two images having different pixel distributions can have no difference in visual appearance (this is the fundamental fact that makes image compression possible). We therefore set out to find numerical artifact indicators (NAIs) that will give good indication of the existence and the strength of the artifacts.

Only measuring the changes between border pixels may not give us sufficient information to indicate the existence of blocking effects because it may be caused by fast-moving signals. One way of measuring the existence of blocking effects is to measure the changes in pixel intensities in a neighborhood of the border area.

The objective is to restore the pixels in the border areas that cause the blocking noise. This noise can be directly measured by calculating the difference between the original signal and the quantized signal in these areas. Our idea is to find a function of the NAIs that measures the coding errors in the border areas. The explicit form of the relationship between the NAIs and the coding error is not known, but we have data available; this gives rise to a classical application scenario where neural network are well suited.

B. MLP for Blocking-Artifact Removal

Inspired by the success that the MLP neural networks have had in a variety of applications in the fields of signal processing and pattern recognition, we develop in this work a new technique based on the MLP neural networks for blocking-effect removal in block encoded images. The idea is to extract relevant information from the compressed image as input to the neural network. The network will try to learn to reconstruct the original image. The schematic of the framework is illustrated in Fig. 2.

In the encoder, the image is compressed and decompressed by the standard image-compression algorithms such as JPEG. From the decompressed image, features indicating the existence of blocking effects, the NAIs, are extracted and fed to the MLP network as its input. The MLP will try to produce an output approximating the difference between the original image and the decompressed image. To train the MLP network, an appropriate supervised learning algorithm, such as the backpropagation algorithm, will be used and the difference between the original and the decompressed image will be used as the desired output

TABLE III
PSNR IMPROVEMENTS FOR A SYSTEM TRAINED ON THE F16 IMAGE

Images	PSNR (dB)	Improvement
Boats	31.50	0.48
F16	32.68	0.59
Girl	33.30	0.71
Lena	32.48	0.54
Peppers	32.78	0.47

TABLE IV
PSNR IMPROVEMENTS FOR A SYSTEM TRAINED ON THE GIRL IMAGE

Images	PSNR (dB)	Improvement
Boats	31.46	0.44
F16	32.46	0.37
Girl	33.28	0.69
Lena	32.48	0.54
Peppers	32.77	0.46

TABLE V
PSNR IMPROVEMENTS FOR A SYSTEM TRAINED ON THE LENA IMAGE

Images	PSNR (dB)	Improvement
Boats	31.51	0.49
F16	32.67	0.58
Girl	33.32	0.73
Lena	32.53	0.59
Peppers	32.82	0.51

(teacher). After the training is complete, the weights of the MLP network along with the compressed image data will be stored or transmitted.

In the decoder, the compressed data is first decompressed, the same set of blocking-effect features are extracted and fed to the MLP network, the output of the MLP network is added to the decompressed image to form the final decoded image.

In the basic system, each time an image is compressed, an associated post processing system (MLP network) needs to be trained. We shall show empirically that the networks can be trained "off line," i.e., networks trained on one image will work well on other images at a similar bit rate.

C. Implementation

Because blocking effects are caused by the abrupt changes in block intensities between the neighboring blocks, the inter-

TABLE VI
PSNR IMPROVEMENTS FOR A SYSTEM TRAINED ON THE PEPPERS IMAGE

Images	PSNR (dB)	Improvement
Boats	31.53	0.53
F16	32.71	0.62
Girl	33.35	0.76
Lena	32.53	0.59
Peppers	32.86	0.55

TABLE VII
PSNR (dB) IMPROVEMENTS OF LENA IMAGE AT DIFFERENT BIT RATES

Training Quality	Testing Quality		
	Q = 15	Q = 45	Q = 75
Q = 15	0.59	-0.03	-0.69
Q = 45	0.39	0.30	0.04
Q = 75	0.23	0.18	0.15

block slope, i.e., the difference in pixel intensity between the adjacent blocks contains useful information which will indicate the existence of blocking effects. In this scheme, we use this information as the NAIs and input to the MLP network. The desired output for the network is set to be the difference between the original (uncompressed) and the compressed image. It is appropriate at this point to stress that use of the pixel intensity of the compressed image as input and the original (uncompressed) image pixel intensity as desired output, which may be the most obvious choice, does not work well. The primary reason is that the absolute intensity values contain no information about the blocking effects.

Let $I_{i,j}$ be an $M \times N$ original image, $\hat{I}_{i,j}$ the reconstructed image of $I_{i,j}$ after compression, $i = 0, \ldots, M-1$, $j = 0, \ldots, N-1$. Assuming the image is coded using square block size of $b \times b$. A three-layer MLP neural network with three input and two output units is constructed to process the border pixels. The number of hidden units are decided through experiment. In extensive simulations we have performed, it was found that no more than four hidden units are required.

To process the horizontal block border pixels, the 3-D input vectors to the network, $X(i, j) = (x_1(i, j), x_2(i, j), x_3(i, j))$ are formed as

$$\begin{cases} x_1(i, j) = \hat{I}_{bi-1, j} - \hat{I}_{bi-2, j} \\ x_2(i, j) = \hat{I}_{bi, j} - \hat{I}_{bi-1, j} \\ x_3(i, j) = \hat{I}_{bi+1, j} - \hat{I}_{bi, j}, \end{cases}$$
$$i = 1, 2, \ldots, \frac{M}{b} - 1, \quad j = 0, 2, \ldots, N-1. \quad (1)$$

(a)

(c)

(b)

(d)

Fig. 3. Illustration of blocking-effect reduction. (a) Original Lena image. (b) JPEG compressed, 0.27 bpp. (c) Neural network processed image of (b) with its own training data. (d) Neural network processed image of (b) with training data from Boats image.

The corresponding *desired* output vectors, $Y_d(i, j) = (y_{1d}(i, j), y_{2d}(i, j))$ are formed as

$$\begin{cases} y_{1d}(i, j) = I_{bi-1, j} - \hat{I}_{bi-1, j} \\ y_{2d}(i, j) = I_{bi, j} - \hat{I}_{bi, j}, \end{cases}$$
$$i = 1, 2, \ldots, \frac{M}{b} - 1, \quad j = 0, 2, \ldots, N - 1. \tag{2}$$

Using the pair of training samples in (1) and (2) to train the network until it converges. Once the network is trained, its weights are saved. The border pixels of $\hat{I}_{i, j}$ are modified as follows to form a new image $\tilde{I}_{i, j}$:

$$\begin{cases} \tilde{I}_{bi-1, j} = y_1(i, j) + \hat{I}_{bi-1, j} \\ \tilde{I}_{bi, j} = y_2(i, j) + \hat{I}_{bi, j}, \end{cases}$$
$$i = 1, 2, \ldots, \frac{M}{b} - 1, \quad j = 0, 2, \ldots, N - 1 \tag{3}$$

where $y_1(i, j)$, $y_2(i, j)$ are the neural network's outputs.

The vertical block borders are processed in a similar manner. Please note that a single network is used to process both horizontal and vertical borders.

III. Experimental Results

We have performed extensive simulations using the new technique to process JPEG compressed images, some of the results are presented here. In the results presented, the images are compressed by the independent JPEG group's software by setting the quality factor to various values with coding optimization. Peak signal-to-noise ratio (PSNR) of the whole image calculated (4) is used to measure the performance

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right) \text{ dB}. \tag{4}$$

In all cases, the size of the MLP network used have three inpust, four hidden, and two output units. A single MLP network was used to process both horizontal and vertical border pixels. In the training stage, the inputs and desired outputs were formed according to (1) and (2) (the inputs and outputs for the vertical borders are formed in a similar manner). The networks were trained using the backpropagation algorithm [8], the training rate used was fixed to 0.05 (the momentum term was not used). It was found that the networks converged quite fast, in the results presented, all networks were trained for ten epoches.

Five well-known images are used in the experiment; the bit rate and PSNR of these images when compressed using JPEG software at a quality factor of 15 are listed in Table I. For each image, one network was trained and tested on itself and other four images. Results are listed in Table II–VI. It is seen for the same quality level that the networks generalized quite well from one image to the other. Visual qualities of the images have also improved accordingly. An example is shown in Fig. 3.

We have also experimented the network's generalization capability at different bit rates. In Table VI, we show the PNSR performance of Lena image at three different quality levels. It is seen that a network trained on a low-quality image would not work on images of higher quality, while on the other hand, a network trained on a high-quality image did work on low-quality images. The improvement is dependent on the similarity of the image quality.

These results are comparable to those published in the literature [5]–[7]. For example, at a bit rate of 0.3 bpp, the adaptive post-processor of [5] achieve a PSNR improvement of 0.5 dB (from 32.8 to 33.3 dB) on the Lena image, which was shown to be better than the methods of [3] and [4]. Generally speaking, the lower the bit rate (the poorer the compressed image quality), the larger the improvement. Notice in [5] and our current implementation, coding optimization is used, which can reduce the bit rate significantly at the same level of quality as compared to compression without optimization. At a bit rate of 0.27 bpp, our PSNR improvement ranges from 0.54 to 0.59 dB.

IV. Concluding Remarks

A new technique based on the MLP neural network has been developed for removing blocking effects in block-coded images. Despite its simplicity, the method is quite effective. Simulation results show the new technique is able to improve the quality of JPEG-compressed images, both subjectively and objectively. The networks used are quite small and computationally efficient; one can easily envisage the scheme being incorporated into JPEG-compression software, which may be valuable in low-bit-rate compression. It is also shown the new method is comparable to state-of-the-art technology.

References

[1] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Compression Standard*. New York: Van Nostrand Reinhold, 1992.

[2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205–220, 1992.

[3] M. Liou, "Overview of the p $\times$ 64 kbit/s video coding standard," *Commun. ACM*, vol. 34, pp. 46–58, 1991.

[4] B. Ramamurthi and A. Gersho, "Nonlinear space-invariant post-processing of block coded images," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-34, pp. 553–559, 1986.

[5] C. J. Kuo and R. J. Hsieh, "Adaptive post-processor for block encoded images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 298–304, 1995.

[6] J. Luo, C. W. Chen, K. J. Parker, and T. S. Huang, "Artifact reduction in low bit rate DCT-based image compression," *IEEE Trans. Image Processing*, vol. 5, pp. 1363–1368, 1996.

[7] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Projection-based spatially adaptive reconstruction of block transform compressed images," *IEEE Trans. Image Processing*, vol. 4, pp. 896–908, 1995.

[8] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, pp. 4–22, Apr. 1987.

[9] R. Hecht-Nielsen, *Neurocomputing*. Reading, MA: Addison-Wesley, 1991.

[10] S. Haykin, *Neural Networks, A Comprehensive Foundation*. New York: Macmillan, 1994.

Guoping Qiu (S'91–M'93) received the B.Sc. degree in electronic measurement and instrumentation from the University of Electronic Science and Technology of China in July 1984, and the Ph.D. degree in electrical and electronic engineering from the University of Central Lancashire, Preston, U.K., in December 1993.

Between 1987–1990, he was a Postgraduate Student with the Radar Research Laboratory, Beijing Institute of Technology, Beijing, China, studying and performing research in the area of digital signal processing. From October 1993 to September 1999, he was a Lecturer at the School of Mathematics and Computing, the University of Derby, U.K. Since September 1999, he has been a Lecturer (Assistant Professor) at the School of Computing, the University of Leeds, Leeds, U.K., where he teaches computer science courses and performs research in various areas of image processing and computer vision. His areas of research include color imaging, image coding/compression, image enhancement, (color) image database, (color) image representation/coding for (visual) content-based indexing and retrieval, computer vision/image processing for industrial inspection, neural networks and pattern recognition for visual information processing, WWW-based visual informatics, and human vision aspect of visual information processing.