Artificial Neural Network Visual Model for Image Quality Enhancement

S. Chen[†], *Z He*[§] *and P.M. Grant*[‡]

[†] Department of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, U.K.

[§] LTX Corporation, Westwood, MA 02090, U.S.A.

[‡] Department of Electrical Engineering
University of Edinburgh, Edinburgh EH9 3JL, U.K.

Abstract

An artificial neural network visual model is developed, which extracts multi-scale edge features from the decompressed image and uses these visual features as input to estimate and compensate for the coding distortions. This provides a generic postprocessing technique that can be applied to all the main coding methods. Experimental results involving post-processing the JPEG and quadtree coding systems show that the proposed artificial neural network visual model significantly enhances the quality of reconstructed images, both in terms of the objective peak signal to noise ratio and subjective visual assessment.

1 Introduction

Image coding is always a trade-off between the coding bit rate and the coded image quality. Increasing coding bit rate can generally improve quality, but this is limited by channel bandwidth or storage capacity. Postprocessing offers an alternative to enhance decompressed image without increasing bit rate. Traditional postprocessing methods [1]–[9] employ filtering to smooth blocking artifacts, and are limited to fixed-block transform coding or vector quantization. Since filtering also causes oversmoothing on image edges, these methods are not appropriate for applications which require genuinely good image quality with minimum distortions. In addition, these existing methods cannot be applied to non-block or variable-block coding systems.

A recent work [10], specifically designed for the JPEG coding system, overcomes the oversmoothing problem of traditional postprocessing methods by directly compensating for coding distortions. It employs 64 neural networks, one for each spatial frequency component in JPEG transform coding, to estimate the quantization errors. Notice that in this method processing for image quality improvement takes place before decompression and, therefore, it alters the existing

JPEG coding system. The experimental results reported in [10] gave an average improvement of 0.63 dB in the peak signal to noise ratio (PSNR), which is better than those achieved by most of the traditional filtering approaches.

Motivated by the biological visual system [11]–[13], we develop a generic postprocessing technique based on an artificial neural network visual model (NNVM). This NNVM consists of a visual feature extractor to extract edge information from the decoded image, and a one-hidden-layer neural network to estimate the coding distortions using the visual features of the decoded image. Unlike the method proposed in [10], our method employs a single neural network to estimate coding distortions and image quality improvement takes place on the decompressed image. Therefore, an existing coding system need not be altered at all, and our method can be applied to all the main coding systems. We apply the proposed postprocessing technique to the JPEG [14] and quadtree (QT) [15] coders. The latter is a variable-block coding method, and existing postprocessing methods, including the method of [10], cannot be applied to this kind of coding systems. Our experimental results confirm that the NNVM achieves significant improvements on the quality of reconstructed images for both the JPEG and QT coding systems.

2 The NNVM for postprocessing

We adopt a generic approach depicted in Fig.1. In this model, the distortion image, which is defined as the difference between the original and decoded images, is estimated. The estimate is added to the decoded image to correct coding losses. The choice of this architecture has an obvious advantage, as the original coding system is not touched and postprocessing is an added component. Notice that, in the technique [10], distortion estimation takes place within the JPEG coding system before the decoding subsystem, which effectively changes the original coding system design. Obviously, relationship between the decoded and distortion images is highly complex. It is known that the main coding distortions are edge distortions, including blurred edges and blocking artifacts [16]. Blurred edges are the result of information losses during compression. For block-based coding methods, visual discontinuities between adjacent blocks result in exotic “edges” along block boundaries called blocking artifacts. These edge distortions are the main visual disturbances for human observers viewing images. Reducing these distortions can significantly improve quality of reproduced images.

We turn to biological visual systems for inspiration. It is known that feature detectors in visual cortex, called simple, complex and hypercomplex cells, respond to edges patterns of an image at different scales [11]–[13]. Simple cells have rectangular receptive fields and respond to bars or edges with specific orientations. By combining the responses of a number of simple cells, a complex cell generates its response over a larger receptive field. The effective stimuli for complex cells are still bars and edges. A hypercomplex cell generalizes the responses of a few complex cells and responds to bars and edges over a wider area of the visual field. A hypercomplex cell forms a basic information unit in perceiving a visual image. Picture perception is accomplished at a higher level of the central nerve system based on these basic information units. This motivates our model for recovering coding distortions shown in Fig. 2. A decoded image of size $N \times N$ is divided into blocks of size $n \times n$, and pixels of each block are fed into a visual feature extractor, which extracts edge features of the block. These edge features are fed into a one-hidden-layer neural network, which produces an estimate of the corresponding distortion image block. We will refer to $n \times n$ as the postprocessing block size.

Edge features are extracted as multi-scale first-order derivatives. To calculate derivatives for an $n \times n$ block in different scales, the block is recursively divided into 4 equal-size sub-blocks until the sub-block size is reduced to 2×2. For a generic sub-block X_s of size $n_s \times n_s$, a pair of horizontal and vertical derivatives (d_h, d_v) are calculated as:

$$d_h = \sum_{i=1}^{n_s} \sum_{j=1}^{\frac{n_s}{2}} X_s(i,j) - \sum_{i=1}^{n_s} \sum_{j=1+\frac{n_s}{2}}^{n_s} X_s(i,j), \quad d_v = \sum_{i=1}^{\frac{n_s}{2}} \sum_{j=1}^{n_s} X_s(i,j) - \sum_{i=1+\frac{n_s}{2}}^{n_s} \sum_{j=1}^{n_s} X_s(i,j)\,, \tag{1}$$

where $X_s(i,j)$ is the pixel value at position (i,j) in X_s. The multi-scale derivatives are normalized to the range $(-1,\ 1)$ and arranged in a vector form $\mathbf{d} = [d_1\ d_2 \cdots d_M]^T$. The total number of derivatives, M, is given by

$$M = 2 \sum_{i=1}^{\log_2 n} \left(\frac{n}{2^i}\right)^2 . \tag{2}$$

The hidden-layer outputs of the NNVM are given by

$$h_k = f\left(\sum_{l=1}^{M} V_{l,k} \cdot d_l + V_{0,k}\right), \quad 1 \le k \le H_n, \tag{3}$$

where H_n is the number of hidden neurons, and the outputs of the NNVM are given by

$$\hat{Y}(i,j) = \alpha \cdot f\left(\sum_{k=1}^{H_n} W_k(i,j) \cdot h_k + W_0(i,j)\right),\ 1 \le i,j \le n\,, \tag{4}$$

where α is a scaling factor. The activation function f is the usual bipolar sigmoid function. The total number of adjustable parameters, P_{NNVM}, for the NNVM is

$$P_{NNVM} = n \times n \times (H_n + 1) + H_n \times \left(1 + 2 \sum_{i=1}^{\log_2 n} \left(\frac{n}{2^i}\right)^2\right) . \tag{5}$$

The network weights $V_{l,k}$ and $W_k(i,j)$ are learnt using the backpropagation algorithm.

To collect training data for a coding system, an $N \times N$ training image is compressed and decompressed. The corresponding distortion image is then obtained. The decoded and distortion images are divided into $n \times n$ blocks. As an image can only provide $\frac{N\times N}{n\times n}$ pairs of training data, many images should be used to collect sufficient training data samples. The choice of the block size $n \times n$ has important influence on the complexity and performance of the model. Ideally, the block size should be as large as possible. However, too large a block size would make computation and storage impractical. For post-processing block-based coding systems, the postprocessing block size should be larger than the coding block size, so that blocking artifacts at coding block boundaries can be corrected. The number of the hidden-layer neurons, H_n, can be determined during training by starting with a small hidden layer and gradually increasing the hidden layer size until the performance stops improving. As training is done off-line, on-line postprocessing only involves passing a decoded image through the trained NNVM. Computational overhead of this postprocessing is very small in comparison with the coder complexity, and is typically less than 1% of the coder computational requirements.

3 Experimental results

The proposed postprocessing technique is generic and can be applied to all the main coding methods. A detailed study, including postprocessing of four different coders and performance comparison with existing methods, can be found in [16]. Here, we apply the NNVM to the JPEG [14] and QT [15] coders. QT coding has variable coding block sizes, and existing methods [1]–[10] are impractical for post-processing this kind of coder. We collected sixteen images of size 512×512 with 8 bits per pixel (bpp), namely “peppers”, “airplane”, “goldhill”, “lake”, “announcer”, “cornfield”, “windows”, “yacht”, “Lena”, “littlegirl”, “Zelda”, “boats”, “cablecar”, “hatgirl”, “kids” and “soccer”, from Internet archives for our experiment. The first eight images were used to provide training data, and the other images were used as test images.

An adequate postprocessing block size was 16×16, as the JPEG coding system has a standard 8×8 coding block size and majority of coding block sizes used by the QT coder were found to be 8×8 or smaller. Our experimental results suggested that $H_n = 40$ was sufficient, because performance improvement was leveled out for $H_n > 40$. Table 1 summarizes the JPEG coder PSNR values and the NNVM postprocessing gains for the eight testing images, respectively, given two different coding bit rates. Fig. 3 shows the face portions of JPEG coded and NNVM post-improved “Lena” images, for a visual evaluation. In the same way, the PSNR values of the QT coder and the postprocessing gains obtained by the NNVM are given in Table 2. Fig. 4 compares the face portion of QT coded “Lena” image with that of the NNVM improved image.

Performance comparison with the method of [10] is not straightforward, as the experimental conditions were not identical. In particular, PSNR values of the original JPEG coder used in

[10] was not provided. Notice that the better PSNR a coding system has, the more difficult to achieve postprocessing gain. Nevertheless, comparison can be made in terms of PSNR gain and our NNVM is better. From Table 2, it can be seen that the NNVM is very effective for enhancing the QT coder. It is worth emphasizing again that the existing postprocessing techniques cannot be applied to the QT coding system.

Coding	Coding bit rate=0.25 bpp		Coding bit rate=0.5 bpp	
image	Coding PSNR	NNVM gain	Coding PSNR	NNVM gain
Lena	28.85	0.81	31.67	0.71
Littlegirl	29.04	0.79	32.26	0.69
Zelda	29.98	0.80	33.22	0.55
Boats	28.23	0.81	31.01	0.77
Cablecar	27.84	0.79	30.96	0.71
Hatgirl	30.60	0.75	34.15	0.58
Kids	28.19	0.69	31.56	0.64
Soccer	27.34	0.73	30.76	0.68
Average	28.76	0.77	31.95	0.67

Table 1: PSNR values (dB) of JPEG coding and NNVM postprocessing gains (dB).

Coding	Coding bit rate=0.25 bpp		Coding bit rate=0.5 bpp	
image	Coding PSNR	NNVM gain	Coding PSNR	NNVM gain
Lena	29.66	0.91	32.39	0.84
Littlegirl	29.87	0.96	32.42	1.01
Zelda	31.38	1.00	33.90	0.89
Boats	28.72	0.85	31.80	0.77
Cablecar	27.83	0.91	30.59	0.93
Hatgirl	33.29	1.14	36.86	0.80
Kids	28.38	0.92	31.14	0.98
Soccer	25.83	1.11	28.32	1.23
Average	29.37	0.98	32.18	0.93

Table 2: PSNR values (dB) of QT coding and NNVM postprocessing gains (dB).

4 Conclusions

A generic postprocessing technique for image coder enhancement has been developed based on an NNVM. This NNVM is inspired by the mechanism of visual perception in visual cortex. Unlike traditional postprocessing methods which basically smooth blocking artifacts to achieve better viewing quality, the proposed technique corrects actual coding losses. As a result, our method is applicable to all the major coding methods while existing methods have limited applications. Experiments of applying the proposed technique to the JPEG and QT coders demonstrate that the NNVM achieves significant improvements on the quality of reconstructed images.

References

[1] H.C. Reeve and J.S. Lim, "Reduction of blocking effect in image coding," in *Proc. ICASSP* (Boston, USA), 1983, pp.1212–1215.

[2] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol.ASSP-34, pp.1258–1267, 1986.

[3] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.2, pp.91–95, 1992.

[4] H. Paek, J.W. Park and S.U. Lee, "Non-iterative post-processing technique for transform coded image sequence," in *Proc. ICIP*, 1995, pp.208–211.

[5] Y.K. Lai, J. Li and C.C.J. Kuo, "Image enhancement for low bit-rate JPEG and MPEG coding via postprocessing," in *Proc. SPIE*, Vol.2727, 1996, pp.1484–1494.

[6] C. Derviaux, F.X. Coudoux, M.G. Gazalet, P. Corlay and M. Gharbi, "A postprocessing technique for block effect elimination using a perceptual distortion measure," in *Proc. ICASSP*, 1997, pp.3001–3004.

[7] J.D. McDonnell, R.N. Shorten and A.D. Fagan, "Postprocessing of transform coded images via histogram-based edge classification," *J. Electronic Imaging*, Vol.6, pp.114–124, 1997.

[8] S.A. Martucci, "A new approach for reducing blockiness in DCT image coders," in *Proc. ICASSP'98* (Seattle, USA), May 12-15, 1998, Vol.V, pp.2549–2552.

[9] B. Jeon and J. Jeong, "Blocking artifacts reduction in image compression with block boundary discontinuity criterion," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.8, No.3, pp.345–357, 1998.

[10] J. Lazzaro and J. Wawrzynek, "JPEG quality transcoding using neural networks trained with a perceptual error measure," *Neural Computation*, Vol.11, pp.267–296, 1999.

[11] D.H. Hubel and T.N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiology*, Vol.160, pp.106–154, 1962.

[12] D.H. Hubel and T.N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *J. Physiology*, Vol.195, pp.215–243, 1968.

[13] E.R. Kandel, J.H. Schwartz and T.M. Jessell, *Principles of Neural Science*. Prentice-Hall, 3rd edition, 1991.

[14] W.B. Pennebaker and J.L. Mitchell, *JPEG Still Image Data Compression Standard.* New York: Van Nostrand Reinhold, 1993.

[15] E. Shusterman and M. Feder, "Image compression via improved quadtree decomposition algorithms," *IEEE Trans. Image Processing*, Vol.3, pp.207–215, 1994.

[16] Z. He, *A Generic Postprocessing Technique for Image Coding Applications.* PhD Thesis, University of Portsmouth, UK, January 1999.

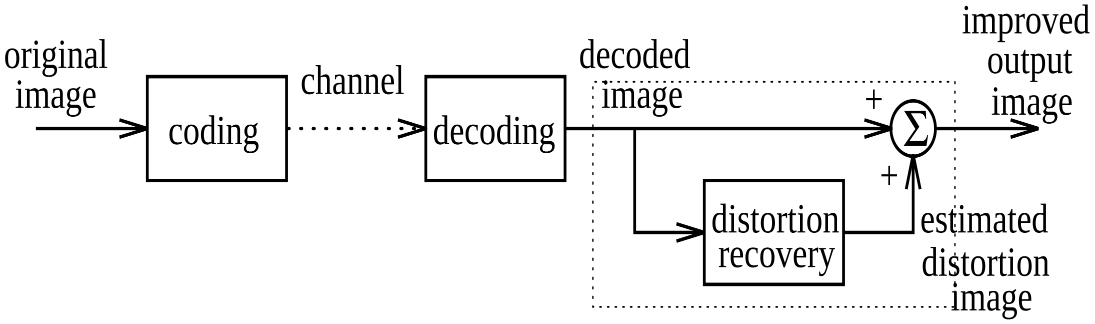


Figure 1: A generic postprocessing approach for image coding.

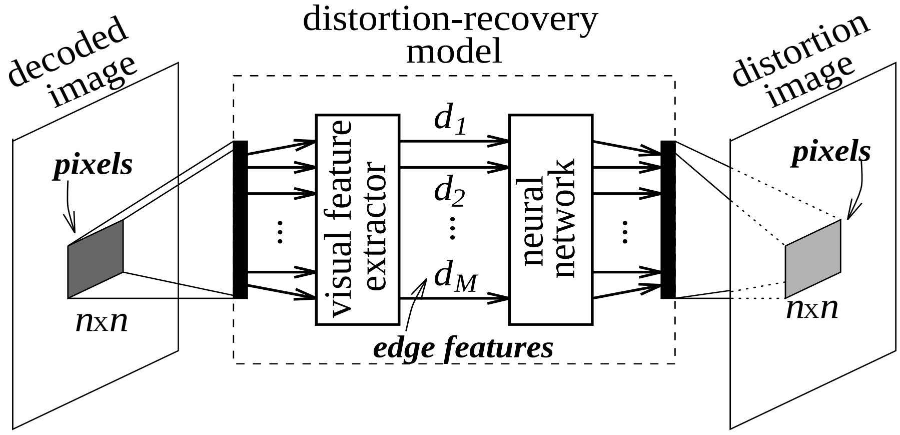
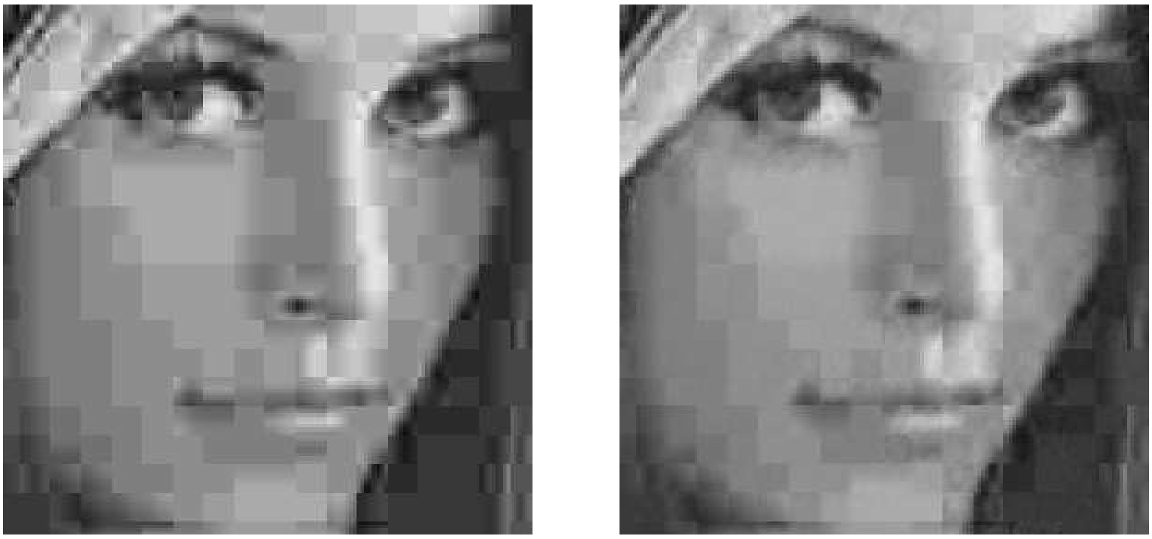


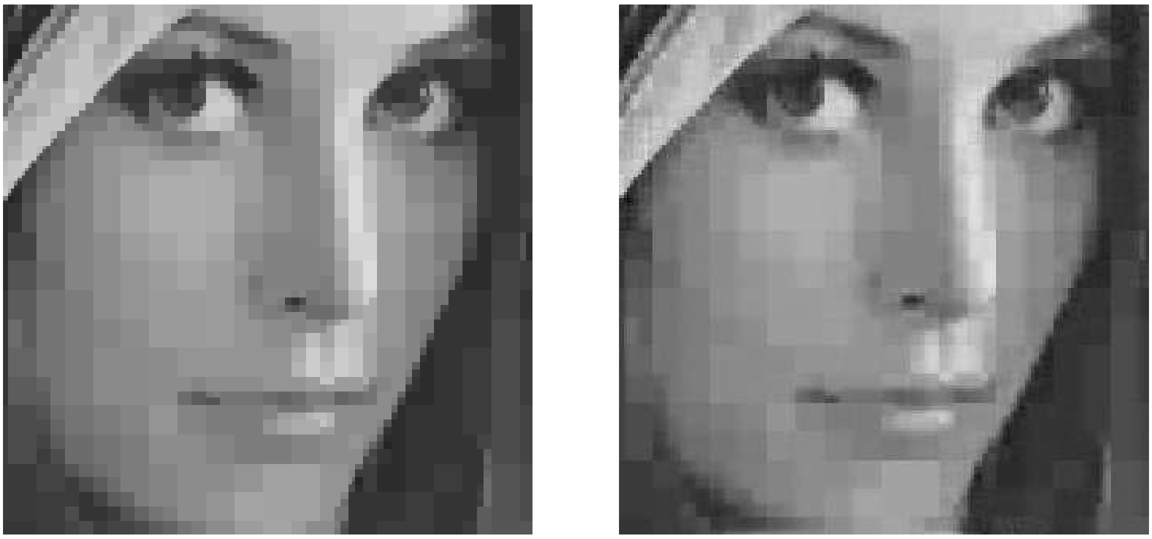
Figure 2: Schematic of the neural network visual model.



(a) JPEG coded (PSNR=28.85 dB)

(b) NNVM (PSNR gain=0.81 dB)

Figure 3: Face portions of the JPEG coded (bit rate=0.25 bpp) and NNVM post-improved images of "Lena."



(a) QT coded (PSNR=29.66 dB)

(b) NNVM (PSNR gain=0.91 dB)

Figure 4: Face portions of the QT coded (bit rate=0.25 bpp) and NNVM post-improved images of "Lena."