

EFFICIENT IMAGE COMPRESSION USING ARTIFICIAL NEURAL NETWORK

Priyali Verma

Department of Electronics and Communication
Jaypee University of Engineering & Technology,
Guna, Madhya Pradesh, India

Email: priyaliverma20@yahoo.co.in

ABSTRACT: In this paper, image compression has been done using Artificial Neural Network (ANN). Need of image compression and advantages of implementing it using ANN have been discussed. An image [256X256] has been considered for compression and its performance parameters like PSNR, SSIM, MSE, RMSE, compression ratio, computational complexity, training and testing time have been evaluated. Block size of image is varied from [4X4] to [8X8] and its effect on reconstructed image quality is observed. Effects of changing the number of hidden neurons and changing the number of epochs during training are also analysed for the reconstructed image. ANN based image compression is highly beneficial for the applications where same image has to be sent repeatedly.

KEYWORDS: Image Compression, ANN, Training of ANN, Feed Forward Back Propagation Algorithm.

INTRODUCTION

Image and video applications have been increased dramatically in recent years. For its efficient storage and transmission, image compression is done. It is a compact representation of image in which we represent an image in lesser number of bits. It reduces irrelevant and redundant information from large sized image for speedy transmission. Image compression can be classified as lossless and lossy compression. In lossless compression, the quality of image is preserved but low compression ratio is achieved. Lossless compression finds applications in Medical Imaging. On the other hand, lossy compressions provide good compression ratio at the cost of image quality. It is used for digital and video applications. Image compression reduces statistical redundancy of the image and exploits perceptual irrelevancy. It reduces transmission time and storage requirement with increase in transmission rate [1].

An Artificial Neural Network (ANN) is a powerful tool able to capture and represent complex input output relationships. It is an information processing paradigm that attempts to imitate the way a human brain works. It stores experimental knowledge and has learning capabilities. ANN can be used for image compression because it has lower computational complexity as compared to

Swapna Kadian

Department of Computer Science Engineering
Jaypee University of Engineering & Technology,
Guna, Madhya Pradesh, India

Email: swapnakadian@gmail.com

conventional compression techniques like DCT. It requires negligible time for image compression and reconstruction once an image is trained. The ANN model used for training can then be generalized for similar kind of images. It also incorporates encryption as the weights should be known to the receiver prior to reconstruction [2].

The compression using ANN is advantageous in applications where same image has to be transmitted repeatedly as for website images. For those images, there is no need for complex compression and decompression algorithms to be implemented each time they are sent. Once those images get trained by the network, extremely less time and complexity is involved for its compression and reconstruction.

Figure 1 shows the model of ANN for image compression. There are x input neurons and y output neurons which are same in number. However, the number of hidden neurons are h where $h < x$. Each neuron of input layer is connected to every other neuron of hidden layer by weights v [$x \times h$] for compression. Each neuron of hidden layer is connected to every other neuron of output layer by weights w [$h \times y$] for reconstruction.

ANN MODEL FOR IMAGE COMPRESSION

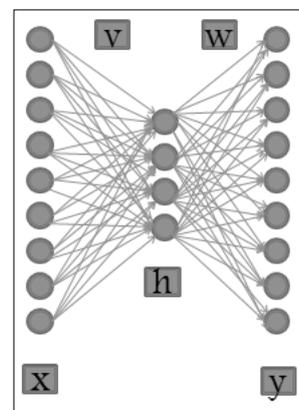


Fig.1. Design of ANN model for image compression

A well-known Lena (bmp format) gray scale image [256X256] has been used to demonstrate the compression technique where each pixel is represented by 8 bits. The image is divided into blocks of [4X4] giving 16 pixels in each block which are to be fed in the network at once. Hence, x and y are taken as 16. Each block is fed one after another until all 4096 patterns are fed to the network which corresponds to 1 epoch.

MATHEMATICAL FORMULATION

For evaluation of quality of reconstructed image and to check the efficiency of compression algorithm, we calculate some parameters.

- Compression Ratio (CR)

It gives the efficiency of compression algorithm used for image compression.

$$CR = (M * N * Q) / T_B$$

Where [MXN] gives the size of original image and Q is the number of bits to represent its one pixel. T_B is the total number of bits required to represent the compressed image given by-

$$T_B = \frac{M * N}{m * n} * h * Q_c + (m * n) * h * r(b)$$

Where, [mXn] is the size of the compressed image, h is the number of hidden neurons, Q_c is the number of bits required to represent one pixel of the compressed image and r(b) is the total number of bits required to store the weight parameters between hidden and output neurons which is generally equal to 32 bits.

- Mean Squared Error (MSE)

MSE quantifies the difference between values implied by estimator and true value of quantity being estimated.

$$MSE = \frac{1}{M * N} \sum_{m=0}^{m-1} \sum_{n=0}^{n-1} [f(m, n) - g(m, n)]^2$$

Where, [MXN] represents the size of the image, f(m,n) and g(m,n) are the original and reconstructed images respectively.

- Root Mean Squared Error (RMSE)

RMSE represents sample standard deviation of difference between predicted and observed values.

$$RMSE = \sqrt{\frac{1}{M * N} \sum_{m=0}^{m-1} \sum_{n=0}^{n-1} [f(m, n) - g(m, n)]^2}$$

- Peak Signal to Noise Ratio (PSNR)

PSNR measures the quality of reconstruction of lossy codecs. It gives the ratio of original image and noise or error introduced by compression. Since, many signals have wide dynamic range, PSNR is usually expressed in dB.

$$PSNR = 10 \log_{10} \left[\frac{255^2}{\frac{1}{M * N} \sum_{m=0}^{m-1} \sum_{n=0}^{n-1} [f(m, n) - g(m, n)]^2} \right] dB$$

- Structural Similarity (SSIM)

It is a method of measuring similarity between reconstructed and original uncompressed image. SSIM considers image degradation as perceived change in structural information.

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Where, μ_x , μ_y are the average of x and y respectively. σ_x^2 , σ_y^2 are the variance of x and y and σ_{xy} is the covariance of x and y. $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ where k_1 is 0.01, k_2 is 0.03 and L is the dynamic range of pixel values which is 255.

- Computational Complexity

The input to each neuron in the output layer is given by

$$y_{-ink} = b_2 + \sum_{j=1}^h z_j w_{jk}$$

Where, b_2 is the bias of hidden layer neuron, z_j is the output of hidden layer, w is the weight matrix of size [jXk] where j and k are the number of hidden and output neurons respectively. It involves (j-1) additions and j multiplications. The output of each neuron of output layer is computed by applying sigmoid function to its input.

$$Y_k = \frac{1}{1 + e^{-(y_{-ink})}}$$

It involves one addition and one multiplication.

TRAINING OF ANN

ANN uses a gradient decent method based delta learning rule called back propagation of errors. The algorithm involves following steps:

- Initialization of weights: If the initial weights are too large, input signals to each neuron falls in saturation region and derivative of sigmoid function approaches zero. If weights are chosen too small, input signals attain values close to zero which will lead to extremely slow learning. Hence, we usually keep the values of v and w in the range $[-0.5$ to $0.5]$.
- Feed Forward Network: The inputs propagate in forward direction from input to hidden and from hidden to output layer.
- Back propagation of Error: The output of the output layer is compared with the actual input and error is computed. This error flows from output to hidden layer updating w and from hidden to input layer updating v .
- Updating weights: Weight is updated for multiple epochs until mean squared error goes below the threshold value or number of epoch get equal to a defined value (550 in this case).

This algorithm tends to minimize the total squared error and aims to train the network to achieve a balance between the ability to respond correctly to input pattern used for training and for inputs that are similar. Learning rate (α) is added to ANN which avoids major disruption in the direction of learning when unusual pair of training pattern is presented. Momentum factor (mf) accelerates the convergence of algorithm. It should be chosen carefully as sometimes mf updates weights in the direction of increasing error.

MATLAB has been used to perform experiments. Training of ANN for Lena image with different values of α and mf is carried out. From *Figure 2*, it is observed that a smooth decreasing curve for error is obtained for $\alpha=0.06$ and $mf=0.22$.

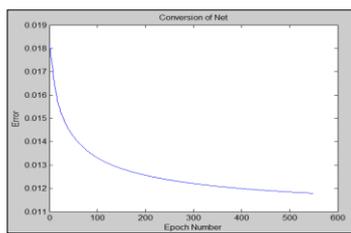


Fig.2. Training of ANN for 550 epochs

RESULTS AND DISCUSSIONS

For an image of 256×256 pixels, ANN configuration is 16:10:16 where number of input and output neurons are 16 and hidden neurons are 10. We train the network for 100 epochs. *Table 1* shows different performance parameters for 3 images (Lena, Cameraman and Penguin).

Table 1: Performance parameters for 3 images

Image	PSNR (dB)	TIME (sec) Training Test	RMSE	CR	SSIM
Lena	26.87	81.42 0.22	11.56	1.57	0.89
Cameraman	25.11	80.12 0.23	11.20	1.57	0.91
Penguin	24.91	81.55 0.21	10.30	1.57	0.85

Figure 3 shows the original and reconstructed



image after compression. It can be observed that the reconstructed images have acceptable quality.

Fig.3. The original images (top) and reconstructed images (bottom) for Lena, Cameraman and Penguin respectively.



Table 2 gives the comparison of $[256 \times 256]$ image of Lena when block size is $[4 \times 4]$ (ANN configuration is 16:10:16) and when block size is $[8 \times 8]$ (ANN configuration is 64:50:64). It is observed that even when higher CR is achieved for image with block size $[8 \times 8]$, PSNR is poor.

Table 2: Performance parameters for different block sizes of image

Parameters	4X4 Block Size, 16:10:16	8X8 Block Size, 64:10:64
Patterns	4096	1024
MSE	287.77	1078.46
RMSE	16.96	32.84
PSNR	23.54	17.80
SSIM	0.8011	0.4781
Training Time (sec)	8.70	14.72
Testing Time (sec)	0.23	0.22
CR	1.57	5.12
Complexity	Equal	Equal

Figure 4 gives the original and reconstructed image when block size is $[4 \times 4]$ and $[8 \times 8]$.



Fig.4. The original image, reconstructed image when block size is [4X4] and reconstructed image when block size is [8X8] with 10 hidden neurons respectively.

It is observed that quality of image in the case of [8X8] block size is very poor. Hence, we increase the number of hidden neurons for this image to improve its quality. However, the CR will be reduced.

Table 3 calculates various performance parameters of reconstructed image. For the image having [8X8] blocks; as the number of hidden neurons increases, PSNR improves from 17.80 dB to 25.67 dB, SSIM increase from 0.4781 to 0.86. However, due to increase in the number of hidden neurons, complexity of the network increases and CR falls from 5.12 to 1.025.

Table 3: Performance parameters for different number of hidden neurons for 100 epochs

Parameters	4X4 Block Size, 16:10:16	8X8 Block Size, 64:50:64
Patterns	4096	1024
MSE	129.91	175.89
RMSE	11.39	13.26
PSNR	26.99	25.67
SSIM	0.8811	0.86
Training Time (sec)	77.86	140.82
Testing Time (sec)	0.23	0.475
CR	1.57	1.025
Complexity	Low	High

Figure 5 shows the original and reconstructed image for different block sizes. As the number of hidden neurons increase, quality of reconstructed image having [8X8] block improves.



Fig.5. The original image, reconstructed image when block size is [4X4] and reconstructed image when block size is [8X8] with 50 hidden neurons respectively.

Table 4 gives the performance parameters of image when block size is [4X4] and [8X8] where training is done for only 10 epochs. It is observed that if training is done for less number of epochs we may not get good PSNR value for the reconstructed image.

Figure 6 shows the original and reconstructed image for different block sizes when training is done for 10 epochs.

Table 4: Performance parameters for different number of hidden neurons for 10 epochs

Parameters	4X4 Block Size, 16:10:16	8X8 Block Size, 64:50:64
Patterns	4096	1024
MSE	287.76	175.89
RMSE	16.96	13.26
PSNR	23.54	25.67
SSIM	0.8011	0.86
Training Time (sec)	8.7025	140.82
Testing Time (sec)	0.23	0.475
CR	1.57	1.025
Complexity	Low	High



Fig.6. The original image, reconstructed image when block size is [4X4] and reconstructed image when block size is [8X8] respectively for 10 epochs.

CONCLUSION

In this paper, ANN based image compression scheme is emphasized. Experiments are performed for the images with different block sizes. As block size increase, we have to increase the number of hidden layers for better perception of reconstructed image which in turn increases complexity and training time of the network. Results are better for the image with block size [4X4] providing a good PSNR, CR and SSIM with less RMSE, low complexity and negligible testing time for compression and reconstruction.

We conclude that ANN based image compression is suitable for the applications where same image has to be sent repeatedly. It gives better performance in terms of PSNR, time and complexity as compared to other techniques.

REFERENCES

-
- 1) **S Jayaraman, S Esakkirajan, T Veerakumar**, “*Digital Image Processing*”, Tata Mc Graw Hill Educaation Private Limited, 2009
 - 2) **S N Sivanandam, S Sumathi, S N Deepa**, “*Introduction to Neural Networks using MATLAB 6.0*”, Tata Mc Graw Hill Educaation Private Limited, 2006
 - 3) **Ivan Vilovic**, “*An Experience in Image Compression Using Neural Networks*”, 48th International Symposium dated 7-9 June 2006, Zadar, Croatia, Page 95-98, 2006
 - 4) **K.S. NG & L.M. Cheng**, “*Artificial Neural Network for Discrete Cosine Transform & Image Compression*”, 0-8186-7898-4197 \$10.00 0 1997 IEEE, Page 675-678, 1997
 - 5) **O. Abdel-Wahhab & M. M. Fahmy**, “*Image Compression using Multilayer Neural Networks*”, *IEE Proc.-Vi.y. Image Sixnal Process.*, Vol. 144, No. 5, October 1997. Page 307-312, October 1997
 - 6) **A. A. Somaie, M. B. Raid & M. A. El-Bahtity**, “*Satellite Image Compression using a Bottleneck Network*”, 0-7803-7000-7/011\$10.0002 001 IEEE. Page No 683-687, 2001