RECONSTRUCTION OF GRAYSCALE IMAGES WITH ARTIFICIAL NEURAL NETWORKS AFTER THEIR COMPRESSION BY PIXEL ELIMINATION METHOD

Hafeez Ullah Jan¹, Dr. Gul Muhammad², Atif Jan³, Muhammad Aamir Aman⁴*

¹, ², ³ Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan.
⁴ Department of Electrical Engineering, Iqra National University, Peshawar, Pakistan.

Corresponding author: Muhammad Aamir Aman, E-mail: aamiraman@inu.edu.pk

https://doi.org/10.26782/jmcms.2020.03.00018

Abstract

Excessive use of electronic devices and image sharing applications in the modern world produce gigantic number of images. The huge image data demands to be handled properly to efficiently utilize the storage space and transmission bandwidth resources. Image compression techniques limit the storage size of the image for this purpose. With the passage of time compression techniques have enhanced to attain more compression and produce decompressed image of high quality. This study which is part of post graduate project suggests the use of neural network to reconstruct the gray scale images which are compressed by withdrawing the pixels from the image. MATLAB is used as programming tool to carry out the simulations. The results obtained are promising.

Keywords: Artificial Neural Network (ANN), Hidden Neurons, MATLAB, Image Compression, Reconstruction.

I. Introduction

The powerful potential of an image to deliver the multiplex ideas makes it preferred tool of communication. The use of images is very common in daily human life, ranging over wide span from diagnosing the diseases, satellite imaging, educational purposes, advertisements and identity recognition to capturing the memorable occasions.

Image is made up of rectangular array of pixels. Each pixel holds a particular location in 2-D plan and the value of the pixel at that point gives the gray level of the image [I]. In this research 8-bits grayscale images are used for which the pixel values range from 0-255. Image compression refers to process of decreasing the image size to
Artificial Neural Network

Artificial neural network processes the information just like a human brain does on much smaller scale. It is an information processing system that is made up of highly interconnected processing elements known as nodes (analogous to neurons in case of human nervous system). The most famous representation of a node is known as McCulloch-Pitts model [V].

Each node in the ANN has fixed number of inputs. The impact of an input on a node’s response depends on the connection’s strength between the input and the node. The connection is known as the weight. The inputs of a node are multiplied with their corresponding weights. The weighted inputs are summed together and the resultant summation is processed by the node according to its activation function to generate the output. The same process is graphically represented in the Fig. I.

![Fig.1: Representation of artificial neuron (node).](image)

Any differentiable function can serve as the activation function for processing nodes of neural network, the most common function is sigmoid [VI], which is given by

\[ f(x) = \frac{1}{1 + e^{-x}} \]  

(1)

II.a. Architecture

Archetypal design of a neural network consists of three entirely connected layers [VII] as shown in the Fig. II.
Input layer is responsible for communication with the environment outside and works as entry point for the data. The input layer configures itself according to the data, performs the necessary calculations and forwards the output to subsequent hidden layer.

The communication of hidden layer is limited within the network since it is placed between the input and output layers. This layer is responsible for most of the data processing. The number of nodes in hidden layer, their activation function and number of hidden layers is of particular importance when designing a neural network for a specific task. The calculation results of hidden layer are transmitted to output layer. After receiving the data from hidden layer, output layer processes the data and concludes the final result which is communicated for post processing outside the neural network if necessary.

II.b. Trainin

Since neural network mimics the animal nervous system it needs to be trained first to solve a problem. The neural network learns by example. During the training, the network is fed with the examples with known output. The weights are adjusted during the learning phase to give optimum output. Gradient descent based back propagation is used in this research for training purpose. In case of back propagation, the training data is made up of pair of input and output vectors. The basic steps [VIII] involved in back propagation learning algorithm are given below and graphically represented in Fig. III.

- Initially the connection weights are assigned random values. Smaller values are preferred.
- The neural network is furnished with the training inputs, the network process them layer by layer and produces output of every node. This is known as the forward pass.
- The error between the desired and actual values of the network’s output is computed by the following equation.
  \[ E = \frac{1}{2} (z' - z)^2 \]  

(2)

Where \( z \) and \( z' \) are the desired and actual values of output respectively.
III. Methodology

This research proposes asymmetric method for compressing the image and its reconstruction. The image is shrunk to achieve compression by eliminating some of its pixels and it is a form of lossy compression. ANN is used for the reconstruction of the missing pixels because of their strong ability to process erroneous and incomplete data. Various cases of compression are discussed below.

III.a. Case study 1

The image matrix is converted into vectors of four pixels row wise. The fourth pixel of each vector is eliminated for compression while the rest of the three pixels are retained. Compression ratio of 1.33 is obtained by this method. For reconstruction, neural network is trained to exploit interpixel redundancy and predict the eliminated pixel with three retained pixels. Once all the missing pixels are predicted, the image is reconstructed by arranging them back again. The image is divided in vectors of 10 pixels row wise for Case studies 2 to 7. For Case studies 2 to 6, the same neural network is used for reconstruction.

III.b. Case study 2

The sixth pixel of each vector is eliminated for compression ratio of 1.11. Another neural network is trained with five inputs. For this case the network uses the first five pixels of each vector to predict the eliminated sixth pixel. After the prediction of missing pixels, the vectors are arranged to reconstruct the image.

III.c. Case study 3

In addition to withdrawing the sixth pixel, seventh placed pixel is also eliminated from every vector to give compression ratio of 1.25. Sixth pixel is predicted after the other five pixels using the neural network.
predicted with the method illustrated in Case study 2. The newly known sixth pixel with four pixels places before it are used as inputs to predict the seventh pixel of each vector using the same neural network. The image is reconstructed after the missing pixels of each vector are predicted.

III.d. Case study 4

Eight pixel is also eliminated along with sixth and seventh pixels in this case to give compression ratio of 1.42. The sixth and seventh pixels are predicted as explained in Case studies 2 and 3. The eighth pixel is predicted by using recently known sixth and seventh pixels and the three pixels placed before them as inputs.

III.e. Case study 5

Same as Case study 4 but ninth pixel is also eliminated. This gives the compression ratio of 1.67. During reconstruction, sixth, seventh and eighth pixels, which are predicted as explained in the previous case, and two pixels placed before them are used to predict ninth pixel.

III.f. Case study 6

Later five pixels of each vector are eliminated to achieve compression ratio of 2. Same approach is used to predict the pixel which is discussed in previous cases.

III.g. Case study 7

The last placed pixel of each vector is eliminated to give compression ratio of 1.11. A new neural network is trained for this case with nine inputs to predict the eliminated tenth pixel.

Three networks are used in the study, Network 1 is the label given to the network which is trained for Case study 1, for Case studies 2 to 6 different network is trained which is labeled as Network 2, and another network labeled as Network 3 is trained for Case study 7. Network 1 takes three inputs, Network 2 takes five inputs and for Network 3, the number of inputs is nine. Since pixel values are used as inputs, they are normalized before providing them to the network. All of these networks try to predict the pixel next to the input pixels, so each network has single output. The basic three-layered architecture is used which means each of the network has only one hidden layer.

IV. Analysis of Results

This study uses feed forward neural for reconstruction. Grayscale version of images from UCID database is used to initially train the networks in MATLAB. The performance of the method is then checked on standard test images which are given shown in Fig.,
The basic training parameters for the networks are labeled in Table I. Typically, in MATLAB the networks can be trained based on different conditions. Since the prediction power of neural networks is exploited in this study, mean square error (MSE) is chosen as the performance function of the network. This means that during training algorithm tries to force the mean square error to minimum for optimum performance.

**Table 1: Network performance parameters**

| Network Performance Parameters | Network 1 | Network 2 | Network 3 |
|-------------------------------|-----------|-----------|-----------|
| Mean Absolute Error           | 0.0394    | 0.0379    | 0.0374    |
| Number of Epochs              | 59        | 139       | 78        |
| Mean Square Error             | 0.0058    | 0.0054    | 0.0053    |
| Training Time (s)             | 577       | 626       | 947       |
| Hidden Nodes                  | 2         | 3         | 5         |
The simulation results on the test images are obtained in terms of Root Mean Square Error (RMSE), Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM). RMSE is obtained by taking square root of mean square error which in turn is obtained by finding the mean of the squared error. For an $M \times N$ image, it is given by

$$RMSE = \sqrt{\left(\frac{1}{MN}\right) \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - I'(x,y))^2}$$ (3)

The units of PSNR is decibels (dB) and it gives the peak error. For 8-bit grayscale image it is given by

$$PSNR = 10 \times \log_{10}(255^2/MSE)$$ (4)

SSIM gives the similarity of two images. Both PSNR and SSIM are important image quality metrics.

### Table 2: Experimental results for Fingerprint

| Case study No. | RMSE | PSNR (dB) | SSIM |
|----------------|------|-----------|------|
| Case study 1   | 6.22 | 32.25     | 0.95 |
| Case study 2   | 2.84 | 39.07     | 0.99 |
| Case study 3   | 6.45 | 31.94     | 0.95 |
| Case study 4   | 10.01| 28.12     | 0.89 |
| Case study 5   | 13.44| 25.56     | 0.82 |
| Case study 6   | 39.54| 16.19     | 0.39 |
| Case study 7   | 2.83 | 39.09     | 0.99 |

### Table 3: Experimental results for Airplane

| Case study No. | RMSE | PSNR (dB) | SSIM |
|----------------|------|-----------|------|
| Case study 1   | 9.02 | 29.03     | 0.92 |
| Case study 2   | 5.70 | 33.01     | 0.97 |
| Case study 3   | 10.32| 27.85     | 0.91 |
| Case study 4   | 14.25| 25.06     | 0.85 |
| Case study 5   | 17.72| 23.16     | 0.79 |
| Case study 6   | 21.00| 21.69     | 0.73 |
| Case study 7   | 5.53 | 33.28     | 0.97 |
Table 4: Experimental results for Lena

| Case study No. | RMSE | PSNR (dB) | SSIM |
|---------------|------|-----------|------|
| Case study 1  | 9.00 | 29.05     | 0.92 |
| Case study 2  | 5.48 | 33.35     | 0.97 |
| Case study 3  | 10.49| 27.72     | 0.91 |
| Case study 4  | 14.97| 24.62     | 0.84 |
| Case study 5  | 19.31| 22.42     | 0.77 |
| Case study 6  | 23.34| 20.77     | 0.70 |
| Case study 7  | 5.50 | 33.32     | 0.97 |

Table 5: Experimental results for Baboon

| Case study No. | RMSE | PSNR (dB) | SSIM |
|---------------|------|-----------|------|
| Case study 1  | 15.25| 24.46     | 0.85 |
| Case study 2  | 9.96 | 28.17     | 0.93 |
| Case study 3  | 14.33| 25.01     | 0.86 |
| Case study 4  | 17.98| 23.03     | 0.78 |
| Case study 5  | 21.14| 21.63     | 0.70 |
| Case study 6  | 24.00| 20.53     | 0.62 |
| Case study 7  | 10.00| 28.13     | 0.93 |

The reconstructed Lena image from each Case study is displayed below in Fig. V.

The reconstructed Lena image from each Case study is displayed below in Fig. V.
7. Case study 7 (CR=1.11)

Fig.5: Reconstructed Lena images for each case study

For Case study 1, the reconstructed image quality for all the test images is acceptable. For case studies 2 and 3, there is no visible degradation and SSIM values are also high. From Case study 4 to Case study 6, the PSNR values go significantly low and the reconstructed image is clearly deteriorated. For Case study 7 the PSNR values are high, it is because nine pixels predict only one missing pixel.

The data used for training is very important to the network’s learning. When the network is trained for Case study 2 with the images from [IX] the PSNR value of the Lena image increases from 33.35 dB to 33.52dB. The learning ability of neural network is greatly affected by number of nodes in hidden layer. Table gives the network parameters which are trained with the dataset from [IX] and increased hidden layer nodes while all the other training parameters are kept to the default values. The networks are allowed to train until the best performance is achieved.

Table 6: Parameters of improved networks

| Network Parameters | Network 1 | Network 2 | Network 3 |
|--------------------|-----------|-----------|-----------|
| Number of Epochs   | 538       | 243       | 173       |
| MSE                | 0.0046    | 0.0044    | 0.0044    |
| Time(s)            | 851       | 555       | 368       |
| Hidden Nodes       | 4         | 10        | 9         |

It can be seen in the Table that the MSE values for the networks are better as compared to the MSE values obtained for the networks mentioned in the Table. MSE is chosen as the performance function for the networks so it is of much importance. Results obtained for Lena image from the networks mentioned in Table are shown in Table.

Table 7: Improved results of Lena

| Case study No. | RMSE  | PSNR (dB) | SSIM  |
|----------------|-------|-----------|-------|
| Case study 1   | 8.53  | 29.51     | 0.92  |
| Case study 2   | 5.19  | 33.83     | 0.97  |
| Case study 3   | 9.88  | 28.23     | 0.92  |
| Case study 4   | 14.33 | 25.01     | 0.85  |

Copyright reserved © J. Mech. Cont.& Math. Sci.
Hafeez Ullah Jan et al
Comparing the Table and Table it can be observed that the image quality has been increased by changing the training data and the dimension of hidden layer nodes. Although there isn’t much significant change in the SSIM of the reconstructed images, the PSNR values are improved a lot. The reconstructed Lena images are displayed below in Fig.

| Case study | SSIM | PSNR  | PSNR Improvement |
|------------|------|-------|------------------|
| Case study 5 | 18.72 | 22.69 | 0.77             |
| Case study 6 | 22.79 | 20.98 | 0.70             |
| Case study 7 | 5.48  | 33.36 | 0.97             |

Fig.6: Improved results shown for Lena

The reconstructed images from Case study 4 to Case study 6 in Fig. are clearly of better quality as compared to that in Fig.. The comparison of the PSNR values against compression ratio between the initial results and the improved results is shown in the Fig.
Fig.7: Improvement in the reconstructed image quality

From universal approximation theorem even one hidden layer in the neural network architecture makes it powerful enough to solve most of the problems. Same phenomenon is observed in this research as well. For Case study 2 and Lena image, the network trained with 2 hidden layers gives PSNR of 33.85 dB and when trained with 3 layers, PSNR of 33.88 dB is achieved. More hidden layers add complexity to the network without significance improvement in the performance. Stretching the hidden layer dimension and by wise selection of data for networks’ training, its performance can be enhanced.

By comparing the results to the standard image compression techniques from the literature, it can be discovered that this research yields less compression ratio. The PNSR values for the standard compression techniques are listed in the Table for compression ratio of 16:1. It can be observed that the PSNR values for Case studies 1 to 3 and Case study 7 are almost equally good as the PSNR results from Wavelet and JPEG compression. The PSNR values for Case study 4 are comparable with results from VQ (Vector Quantization) and FC (Fractal Compression). However, there is a huge different in compression ratio. The compression ratio is very high for standard compression techniques as compared to the compression ratio obtained in with proposed technique.

When compared with other neural network-based compression studies, the proposed work yields better values of PSNR as shown in the Table but compression ratio is low.
Table 8: Comparison of the proposed work with standard compression algorithms on images from Fig..

| Algorithm | Fingerprint | Airplane | Lena |
|-----------|-------------|----------|------|
| Wavelet   | 36.71       | 32.48    | 34.66|
| JPEG      | 34.27       | 30.39    | 31.73|
| VQ        | 28.26       | 26.76    | 29.28|
| FC        | 27.21       | 26.7     | 29.04|
| Case study 1 | 32.25       | 29.03    | 29.05|
| Case study 2 | 39.79       | 33.41    | 33.83|
| Case study 3 | 32.77       | 28.2     | 28.23|
| Case study 4 | 28.67       | 25.31    | 25.01|
| Case study 5 | 25.9        | 23.39    | 22.69|
| Case study 6 | 15.65       | 21.92    | 20.98|
| Case study 7 | 39.09       | 33.28    | 33.32|

Table 9: Proposed work vs. other algorithms based on ANN

| Test Image     | PSNR (dB) from other ANN based techniques | PSNR (dB) from proposed work |
|----------------|------------------------------------------|------------------------------|
| Cameraman (256×256) | 29.78                                    | 27.55                        |
| Lena (512×512)  | 28.91                                    | 32.70                        |
| Peppers (512×512)| 29.04                                    | 32.89                        |
| Boats (512×512) | 29.12                                    | 31.14                        |
| Baboon (512×512)| 29.68                                    | 27.77                        |

V. Conclusion

The large amount of image data produced in the modern era needs to be handled properly. Due to utmost need to save the storage and bandwidth resources, intelligent computational techniques have been emerged along with the traditional image compression algorithms. This study suggests withdrawing few pixels from the image for compression and predicting them at the receiver’s end with artificial neural networks to reconstruct the image. Since neural networks can deal with the missing data well, the proposed method can also be used to restore the image if some of the data is lost during the transmission. The study also illustrates that the neural network with only one hidden layer is a universal approximator and its prediction abilities can be used to predict the values of missing pixels. Varying compression ratio ranging from 1.11 to 2 can be achieved through the proposed method. However, the...
compression ratio obtained in this study is still very small. Improvements can be made for gaining high compression ratio and better image quality at high compression ratios. Genetic Algorithm (GA) which is based on human evolution or Generative Adversarial Network (GAN) which was invented in 2014 can be explored for this purpose.

References

I. Asuni, N., & Giachetti, A. (2013). Testimages: A large data archive for display and algorithm testing. Journal of Graphics Tools, 17(4), 113-125.

II. Drew, P. J., & Monson, J. R. (2000). Artificial neural networks. Surgery, 127(1), 3-11.

III. Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15), 2627-2636.

IV. Gonzales, R. C., & Woods, R. E. (2002). Digital image processing (Vol. 2). New Jersey: Prentice hall.

V. Jain, A. K. (1981). Image data compression: A review. Proceedings of the IEEE, 69(3), 349-389.

VI. Marsalli, M. (2006). McCulloch-pitts neurons. In The 2008 Annual Meeting of the consortium on cognitive science instruction (ccsi) (pp. 172-179).

VII. Nelson, M., & Gailly, J. L. (1996). The data compression book (pp. 457-458). New York: M & t Books.

VIII. Warner, B., & Misra, M. (1996). Understanding neural networks as statistical tools. The american statistician, 50(4), 284-293.

IX. Yang, M., & Bourbakis, N. (2005, August). An overview of lossless digital image compression techniques. In 48th Midwest Symposium on Circuits and Systems, 2005. (pp. 1099-1102). IEEE.