IMPLEMENTATION OF WEIGHTED CENTROID NEURAL NETWORK FOR EDGE PRESERVING IMAGE COMPRESSION

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Amarender Reddy Depa

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

November 2017

Fargo, North Dakota

North Dakota State University
Graduate School

Title

IMPLEMENTATION OF WEIGHTED CENTROID NEURAL NETWORK FOR EDGE PRESERVING IMAGE COMPRESSION

By

Amarender Reddy Depa

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Simone Ludwig
Chair

Dr. Anne Denton

Dr. Ying Huang

Approved:

11/17/2017 — Date

Dr. Kendall Nygard — Department Chair

ABSTRACT

Image compression is a type of data compression applied to images. The objective of image compression is to reduce the cost for storage or transmission. Image compression is associated with removing redundant information of image data. Image storage is required for several purposes like document, medical images, etc. In this paper, an edge preserving image compression algorithm based on an unsupervised competitive neural network called weighted centroid neural network (WCNN), is implemented and compared to the other algorithms. The WCNN algorithm allots more representative vectors from the edges of the image than the interior of the image thus helping in better edge preservation of the reconstructed image. After experimenting with the cluster count it is evident that with the increase in the number of cluster the quality of the picture is improved, which is the expected behavior as more clusters leads to more representational vectors.

ACKNOWLEDGEMENTS

I would take this opportunity to thank my advisor, Dr. Simone Ludwig, who has given me valuable support, encouragement and advice without which this work would not have been completed. I am thankful to the members of the committee, Dr. Anne Denton and Dr. Ying Huang, for their support. I would also like to thank my parents for their valuable support and constant encouragement.

TABLE OF CONTENTS

ABSTRACT iii

ACKNOWLEDGEMENTS iv

LIST OF TABLES vi

LIST OF FIGURES vii

INTRODUCTION 1

RELATED WORK 5

APPROACH 7

RESULTS AND DISCUSSION 15

CONCLUSION AND FUTURE WORK 32

REFERENCES 34

LIST OF TABLES

Table Page

1. Codebook Size and bpp.. 13

LIST OF FIGURES

Figure Page

1. Compression/Decompression Process.. 2

2. Transform Based Image Compression.. 3

3. Training Images. 11

4. Codebook with Number of Clusters 16 and 32.. 12

5. Codebook with Number of Clusters 64 and 128.. 12

6. Codebook with Number of Clusters 256. 12

7. Test Images. 13

8. Cameraman Original Image.. 18

9. Cameraman Decompressed with 256 Clusters. 18

10. Cameraman Decompressed with 128 Clusters. 19

11. Cameraman Decompressed with 64 Clusters. 19

12. Cameraman Decompressed with 32 Clusters. 20

13. Cameraman Decompressed with 16 Clusters. 20

14. Bird Original Image. 21

15. Bird Decompressed with 256 Clusters.. 21

16. Bird Decompressed with 128 Clusters.. 22

17. Bird Decompressed with 64 Clusters.. 22

18. Bird Decompressed with 32 Clusters.. 23

19. Bird Decompressed with 16 Clusters.. 23

20. Airplane Original Image. 24

21. Airplane Decompressed with 256 Clusters.. 24

22. Airplane Decompressed with 128 Clusters.. 25

23. Airplane Decompressed with 64 Clusters.. 25

24. Airplane Decompressed with 32 Clusters. 26

25. Airplane Decompressed with 16 Clusters. 26

26. Couple Original Image. 27

27. Couple Decompressed with 256 Clusters. 27

28. Couple Decompressed with 128 Clusters. 28

29. Couple Decompressed with 64 Clusters. 28

30. Couple Decompressed with 32 Clusters. 29

31. Couple Decompressed with 16 Clusters. 29

32. Comparison of PSNR Values for Different Codebook Sizes. 30

33. Comparison of SSIM Values for Different Codebook Sizes. 30

INTRODUCTION

A digital image is an ordered array of pixels conveying visual information for human or machine analysis. Due to the advent of mobile phones and other multimedia image capturing devices together with the effect of social media, the number of images captured for personal viewing has increased exponentially. Apart from a that huge number of images are captured for commercial and research purposes. Images captured for medical diagnosis and satellite images are examples for the later type. The revolution in Artificial Intelligence and Computer Vision fields has made cameras and images an integral part of many applications. Despite the advancement in memory and communication technologies, the storage and transmission of such huge amount of data is still a challenge, and image compression algorithms ensures efficient storage and faster transmission. The original image pixel array consumes a huge amount of data. By applying a compression algorithm, a smaller sized (compressed) image is obtained. Thus, this smaller sized image can be stored in less space or transmitted faster. The stored/received smaller image is then reconstructed using an associated decompression algorithm (Figure 1).

Lossless compression is a class of data compression algorithms that allow the original data to be perfectly reconstructed from the compressed data. By contrast, lossy compression permits reconstruction only of an approximation of the original data, though this usually improves the compression rates and therefore reduces the file sizes.

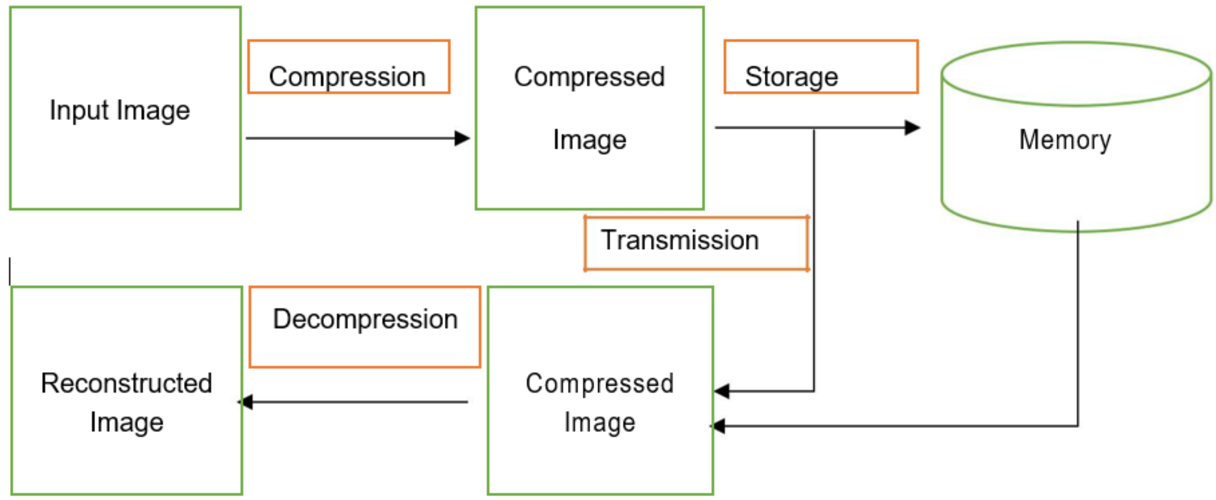

Figure 1. Compression/Decompression Process.

Basically, there are two types of image compression schemes. Lossy compression and lossless compression. In a lossless compression scheme both the input and reconstructed images are exactly the same. But in lossy compression they are different, though the difference is not easily perceivable. Lossless compression schemes can achieve only a limited amount of compression; hence they are rarely used. Critical medical images or satellite images obtained using expensive cameras are compressed using lossless techniques. Most general-purpose image compressions use lossy schemes. There is a trade-off between the amount of compression achieved and the reconstructed image quality while using lossy compression methods. There are many different types of lossy compression algorithms. All of them make use of the redundancy present in the data. In addition to the statistical redundancy image compression algorithms exploit the spatial redundancies of the image data. Smooth regions in images have similar pixel values and human eyes cannot perceive slight changes in the image pixel values. These two properties are exploited to achieve lossy compression. Some lossy compression algorithms are based on predictive methods. But a majority of lossy compression algorithms are based on image transforms. In transform-based methods (Figure 2) the image is transformed from the spatial domain to the transform domain. That is the image is represented by transform domain coefficients instead of spatial domain pixels. The transform domain coefficients have a simple representation and are used to identify significant and insignificant coefficients. Coefficients whose loss will not affect the visual perception of the image are removed by a simple thresholding operation, thus, compression is achieved. Further compression in the coefficients can be achieved by applying lossless data compression methods. The Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are the most commonly used techniques for image compression [1,2]. DCT is used in the JPEG compression algorithm and DWT is used in the JPEG2000 algorithm

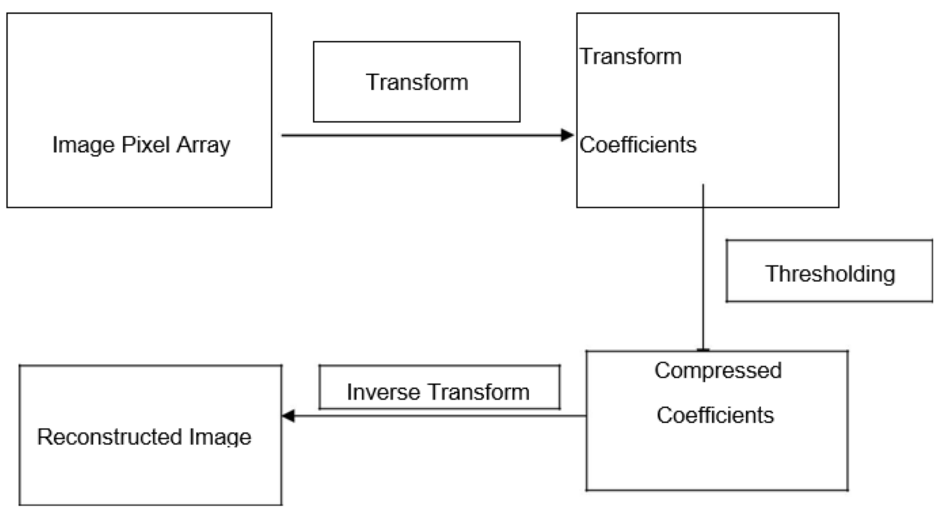

Figure 2. Transform Based Image Compression.

In the proposed implementation, the compression is based on the Vector quantization algorithm [3]. Quantization, in general, maps a set of infinite values to a finite set of values. There are two major types of quantization techniques they are scalar quantization and vector quantization. In scalar quantization algorithms, a scalar value is mapped from an infinite set to a finite set. Likewise, in vector quantization a vector is mapped from an infinite set to a finite set of vectors. Vector quantization gives better compression performance than scalar quantization. But the computational burden and memory resources required for a vector quantizer is greater than the resources required by a scalar quantizer. While using vector quantization algorithms for image compression the image is divided into blocks and each image block forms a vector. Thus, while applying vector quantization to an image the input to the quantizer is a vector of particular length. But the values of the vector can be anything as it depends upon the image. Thus, the input vectors can have an infinite number of possible values.

There is a need for image compression due to various factors like full size images cannot be viewed or are not supported so the compressed images are used, saving memory as the size of the compressed image is less than the actual size of the image. In this project, we have taken several images, in which original images were converted into compressed images using the various compressing methods and then decompressed. A comparison of the output has been done for different cluster count and achieving interesting results that are correlated to the cluster count.

RELATED WORK

There are many ways that were proposed to tackle the idea of better edge preservation during the reconstruction of images in the past [3,4]. The idea to have a better edge preservation is based on a simple concept of classifying vectors from two regions, edge and non-edge, and by allocating more representational vectors to the edge region.

The approach of classifying the training vectors into edge and non-edge types and a separate codebook, called the sub codebook for the vectors of each class to the classified vector quantizer (CVQ) has been introduced [5]. To achieve an optimal distortion different classes of edges/shades are assigned by using the criterion of the equal average MSE distortion measure in CVQ.

One more concept on the idea of reducing the edge degradation during reconstruction of an image is proposed by introducing the concept called adaptive self-organization map (ASOM) [6]. In this approach, the variance of image block is used as an activity measure, possibility of edge presence. The measured activity of a block is, then, used as a weighting factor of the block, which is finally used as a learning gain of the block data for Kohonen's self-organization map (SOM) [7]. It is simple and effective in reducing edge degradation in VQ.

In images, edges contain significant information about the information an image is conveying and the human eye understands the major information of an image by its edges. These factors make it important to preserve the edges during image compression. Edge preserving image compression algorithm based on an unsupervised competitive neural network, called weighted centroid neural network (WCNN), is implemented in this paper. The proposed WCNN calculates the probability density function of the edge strength from the image data and utilizes it to assign the number of code vectors. In WCNN the algorithm as compared to the other algorithms allots more representative vectors from the edges of the image than the interior of the image thus helping in better edge preservation in reconstructed image. After experimenting with the number of clusters it was evident that with increase in number of cluster the quality of the picture was improved which is the expected behavior as more clusters leads to more representational vectors.

APPROACH

Vector quantization (VQ) is a coding technique that has been widely used in signal coding applications given its attractive features [8].

Favorable circumstances of VQ are applications that require a low bit rate for transmission of the fact that VQ can use high compression ratios with relatively small block sizes while this is troublesome in other compression strategies, for example, change coding. Alongside the high compression proportion of VQ, another preferred standpoint utilizing VQ is that the decoder is moderately easy to actualize. At the point when connected to image compression, this property of VQ is particularly vital in different decoder cases, for example, computerized TV flag transmission, video conferencing, or transmission of remote sensing pictures from satellites. Late advance in multimedia frameworks makes picture compression with VQ considerably more attractive.

The problem with early picture coding strategies that utilize VQ are that at low bit rates for the most part experience ill effects of edge corruption since edges cannot be replicated well by a small-sized codebook [9]– [10]. Only a little segment of codes in a codebook will be allocated to the edge information in those early picture coding calculations with ordinary mutilation measures, for example, the mean squared error (MSE) since that edges have just little populaces in a picture. Since, the edges incorporate the most critical information of data of a picture, their corruption ought to be more irritating to the human eyes than the debasements in the non-edge part of a picture. In this sense, the MSE is not a good measure to the extent that edge coding is concerned.

The CNN (centroid neural network) finds the centroid of data in relating clusters at every introduction of the information vector [11,12]. Rather than computing the centroids of the grouped information for each datum introduction, the CNN algorithm refreshes their weights just when the status of the yield neuron for the exhibiting information has changed: that is, the weights of the winner neuron in the present epoch for the information change just when the winner neuron did not win the information in the past introduction and the weights of the winner neuron in the past epoch for the information change just when the neuron does not win the information in the present epoch. We call the former one a "winner neuron" and the latter one a "loser neuron."

When CNN is compared with the conventional competitive learning algorithm, the CNN creates equivalent outcomes with less computational effort. That is, CNN requires neither a predetermined schedule for learning pick up nor an aggregate number of cycles for clustering and it converges stably to suboptimal solutions while conventional algorithms may give unstable results depending on the initial learning gain and the total number of iterations.

Since a definitive judge of the nature of recreated pictures are human onlookers and there are no target measures which intently reflect the execution of the human watcher [13], the optimality conditions for picture coding is elusive. However, the WCNN described in this section can be an approach for edge preserving image compression [14]. In this paper, the author proposed a new unsupervised competitive neural network, called weighted centroid neural network (WCNN), to reduce edge degradation significantly in image while minimizing the computational complexity.

WCNN deals with vector quantization codebook generation using a weight centroid neural network. The algorithm takes two variables as inputs “dataVectors”, which is the image data vectors and “nClusters”, which is the cluster count, which indicates the codebook size and determines the compression ratio, and the output is the codebook. In a vector quantization based compression scheme, an optimal codebook is the one that gives minimum distortion in the reconstructed image. In the WCNN algorithm, the optimality condition is changed, and the optimal codebook is the one that gives minimum distortion in the edge regions of the reconstructed image. This minimum distortion is incorporated in the WCNN algorithm in the following way. The centroid calculation of each cluster in WCNN is weighted, the weights will be updated in such a way that the eventual codebook will introduce minimum distortion in the reconstructed image. Updating the weights procedure utilizes the probability density function of edge information from the training data. Thus, the resulting codebook generally preserves edges in the reconstructed images. The algorithm proceeds as follows:

1. Find the centroid of the data vectors (by calculating the mean of the data vectors).

2. Initialize a neural network with 2 neurons. Choose the initial weights of the neurons around the centroid value. The weight matrix is declared for the maximum number of neurons, but only the first two neurons are initialized. A neuron is specified by the row index of the weight matrix and its weight values are the column elements.

3. Start the training for the zeroth epoch.

4. Present the first data vector to the neural network.

5. Find the winning neuron by calculating the distance between the data vector and the neuron weights. Update its weights.

6. If this is not the zeroth epoch find the loser neuron and update its weights.

7. Accumulate the distance values for the winning neurons.

8. Repeat steps 5 to 7 for all other data vectors.

9. If this is not the zeroth epoch and if there is no loser neuron, proceed to the next step, otherwise repeat steps 4 to 8; after incrementing the epoch count by 1.

10. If the number of neurons is less than the required cluster count, add a new neuron by splitting the most erroneous existing neuron and repeat steps 4 to 9.

WCNN does not requires a predetermined schedule for the learning gain or a total number of iterations. We wanted to extend the work beyond implementation of the algorithm to see the impact of the input parameters on the image compression process. The input parameters in the WCNN algorithm are cluster count and the number of data vectors.

WCNN is used to generate the codebooks. The codebooks are the major components in vector quantization based compression. In VQ based compression techniques, the selection of block size plays an important role. A smaller block size will increase the codebook generation time and reduce the compression ratio. But on the positive side it will improve the reconstructed image quality. In the same way, larger block sizes will reduce codebook generation time and increase compression but at the cost of poor reconstructed image quality. So, the best solution is to select an optimal block size that gives good compression with an acceptable amount of loss in the reconstructed image. The paper explains the concept that 3*3 and 5*5 images produce similar results that are produced with 4*4 image, but the computational complexity increases as we increase the block size so the paper considers 4*4 as the default block size, which strikes the balance between the computational complexity and the compression results.

The other parameter this algorithm considers is cluster count. A bigger codebook will give many representative vectors; hence the quality of the reconstructed image will be high. But on the other hand, the codebook generation process will take more time and the compression ratio will get reduced. When we increase the cluster count, the size of the codebook increases as it will accommodate more representative vectors and that is observed in the results with increase in measures like PSNR, bits per pixel, and SSIM. The algorithm is coded using MATLAB. A set of six training images are used for generating the codebook using WCNN. The training images are given in Figure 3.

Figure 3. Training Images.

The codebooks are generated by the WCNN algorithm for 16, 32, 64, 128 and 256 clusters as shown in Figures 4-6.

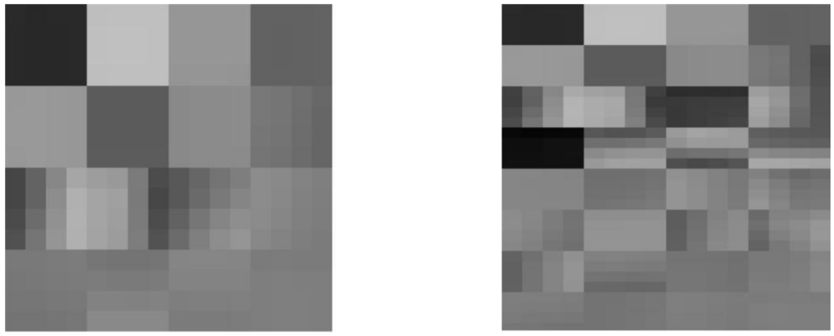

Figure 4. Codebook with Number of Clusters 16 and 32.

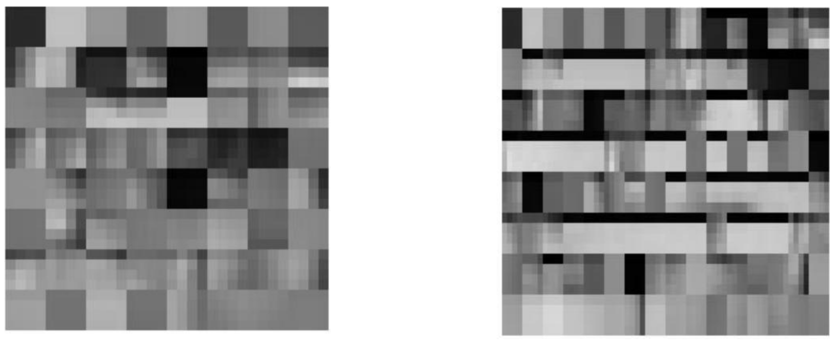

Figure 5. Codebook with Number of Clusters 64 and 128.

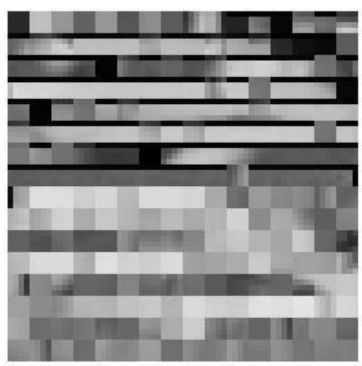

Figure 6. Codebook with Number of Clusters 256.

The six training images are partitioned into 4x4 blocks. Thus, each block gives a data vector of length 16. A total of 84992 data vectors are obtained from the six images. The compression achieved by the vector quantization using a sized codebook is always the same, irrespective of the image size. The compression ratio that can be achieved by the five codebooks described above are given in Table 1.

Table 1. Codebook Size and bpp.

Codebook Size	**Bit Rate**
16	0.25bpp
32	0.3125bpp
64	0.375bpp
128	0.4375bpp
256	0.5bpp

Here the measure used for measuring the compression efficiency is in bits per pixel (bpp), which indicates the number of bits required to code a single pixel of the image. For example, when the codebook size is 16, the bitrate is 0.25 bpp. That is a pixel can be coded using one fourth of a bit or alternatively the information about four pixels can be stored using a single pixel. The lower the bitrate value, the higher the compression is. As described previously in a lossy compression scheme, there is always a trade-off between the compression performance and the reconstructed image quality. Hence a set of test images is compressed and decompressed using vector quantization with the generated codebooks. The test images are different from the training images. The reconstruction image quality is studied in terms of peak signal to noise ratio (PSNR) [15] and structural similarity index measure (SSIM) [16].

The two test images being used are the Lena and Airplane images (shown in Figure 7).

Figure 7. Test Images.

The Peak Signal to Noise Ratio (PSNR) metric compares the original and reconstructed image. It is the ratio of the energy of the maximum pixel to the mean square error between the original and reconstructed images. So, when the difference between the original image and the reconstructed image is less, then the PSNR will be high. Hence a high PSNR value indicates reconstructed image quality is high and vice versa. PSNR in decibels is given by the following formula PSNR in db = 10 log (P2/ MSE), where P is the maximum pixel value and MSE is the mean square error between the original and reconstructed images. In addition to better reconstruction image quality, the proposed algorithm also preserves the edge information in the original image. This is validated by using the structural similarity index measure (SSIM). SSIM metric is based on user perception. It measures the changes in image information and structures by comparing different blocks of images. Generally, an Image is split it to windows of size 8X8 and the luminance, contrast and structure information are compared over different windows. SSIM takes values between -1 to 1. A SSIM value of 1 indicates both the images are identical.

RESULTS AND DISCUSSION

We have generated various codebooks with different cluster count using the WCNN algorithm to observe the impact of the cluster count on the overall image clarity during the reconstruction of the image that is decompressed.

The proposed algorithm is applied to the test and training images and the results are as follows. Here the properties of images used are they should have no discrete pixels and there should be no over lapping blocks. As we know the desirable condition for image compression is where we have a high compression ratio and a high PSNR. If we include Structural Similarity Index as our measure for the image compression, then the desired output should be a high value of SSIM which represents the similarity of the reconstructed image compared to the original image. We know that low bits per pixel leads to high compression ratio. So, the ideal condition is a low bpp which means a high CR (compression ratio).

In this experiment if we observe with the increase in cluster count as we go along from 16 to 256 we can observe the improvement in PSNR and SSIM because as we know more clusters means they will accommodate more representative vectors so during the image reconstruction there will be better matching of the vectors, thus the better image is reconstructed.

Figure 8 is the original camera image, Figure 14 is the is the original bird image, Figure 20 is the original airplane image, Figure 26 is the original couple image that are used to create the codebook.

In cameraman images, Figure 9, the image reconstructed with the codebook of 256 clusters the PSNR and SSIM values are 79.42 and 0.89, respectively, in Figure 10, reconstructed with the codebook of 128 clusters the PSNR and SSIM values are 78.06 and 0.83, respectively, in Figure 11, the image reconstructed with the codebook of 64 clusters the PSNR and SSIM values are 75.83 and 0.83, respectively, in Figure 12, reconstructed with the codebook of 32 clusters the PSNR and SSIM values are 75.46 and 0.79, respectively, in Figure 13, reconstructed with the codebook of 16 clusters the PSNR and SSIM values are 73.69 and 0.70, respectively. It is evident that cluster count impact PSNR and SSIM.

In bird images, Figure 15, the image reconstructed with the codebook of 256 clusters the PSNR and SSIM values are 86.17 and 0.91, respectively, in Figure 16, reconstructed with the codebook of 128 clusters the PSNR and SSIM values are 83.84 and 0.87, respectively, in Figure 17, the image reconstructed with the codebook of 64 clusters the PSNR and SSIM values are 83.85 and 0.86, respectively, in Figure 18, reconstructed with the codebook of 32 clusters the PSNR and SSIM values are 82.38 and 0.85, respectively, in Figure 19, reconstructed with the codebook of 16 clusters the PSNR and SSIM values are 80.94 and 0.83, respectively. It is evident that cluster count impact PSNR and SSIM.

In airplane images, Figure 21, the image reconstructed with the codebook of 256 clusters the PSNR and SSIM values are 80.55 and 0.93 , respectively, in Figure 22, reconstructed with the codebook of 128 clusters the PSNR and SSIM values are 78.59 and 0.82, respectively, in Figure 23, the image reconstructed with the codebook of 64 clusters the PSNR and SSIM values are 76.22 and 0.85, respectively, in Figure 24, reconstructed with the codebook of 32 clusters the PSNR and SSIM values are 76.03 and 0.83, respectively, in Figure 25, reconstructed with the codebook of 16 clusters the PSNR and SSIM values are 75.75 and 0.82, respectively. It is evident that cluster count impact PSNR and SSIM.

In couple images, Figure 27, the image reconstructed with the codebook of 256 clusters the PSNR and SSIM values are 79.87 and 0.87, respectively, in Figure 28, reconstructed with the codebook of 128 clusters the PSNR and SSIM values are 79.04 and 0.83, respectively, in Figure

29, the image reconstructed with the codebook of 64 clusters the PSNR and SSIM values are 78.87 and 0.82, respectively, in Figure 30, reconstructed with the codebook of 32 clusters the PSNR and SSIM values are 78.54 and 0.80, respectively, in Figure 32, reconstructed with the codebook of 16 clusters the PSNR and SSIM values are 77.66 and 0.75, respectively. It is evident that cluster count impact PSNR and SSIM.

If we observe the trend of PSNR and SSIM in Figure 32 and Figure 33 respectively, we can say there is a linear relation between the cluster count and the values of PSNR and SSIM with a few exceptions depending on the nature of the image.

From the above observations, we can say that there has been a constant improvement in the quality of the reconstructed image.

Another way we can see the effect of the cluster count apart from the improvement in PSNR and SSIM is that with the increase in cluster count the size of the corresponding codebook generated also increases.

But the increasing cluster count adds to the computational complexity demanding more resources during the computation and with increasing cluster count the codebook size increases, so the optimum cluster count cannot be generalized since it depends on the nature of the image and their application.

The main spirit of WCNN is not having a better PSNR ratio since, there are other algorithms like SOM and ASOM which provide better PSNR ratios but where WCNN stands out is its reconstruction of image with minimal edge degradation when compared to other algorithms.

Figure 8. Cameraman Original Image.

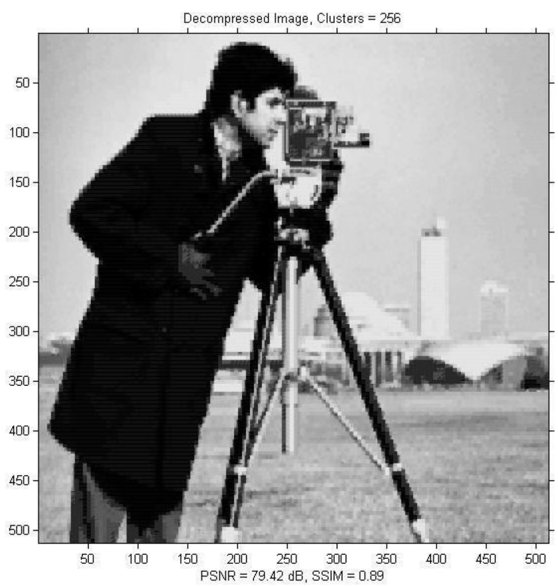

Figure 9. Cameraman Decompressed with 256 Clusters.

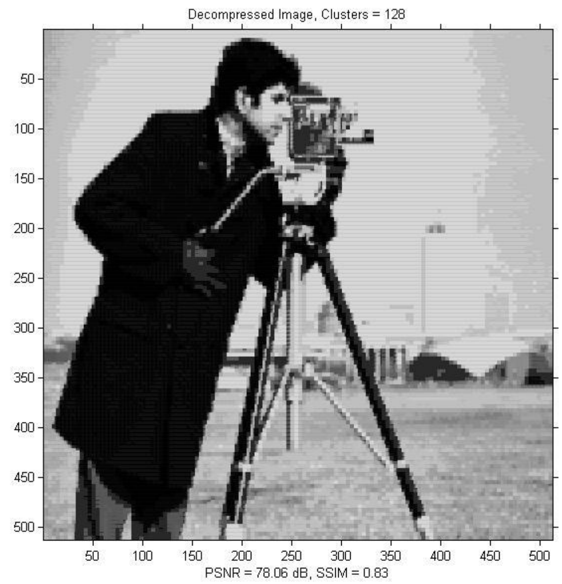

Figure 10. Cameraman Decompressed with 128 Clusters.

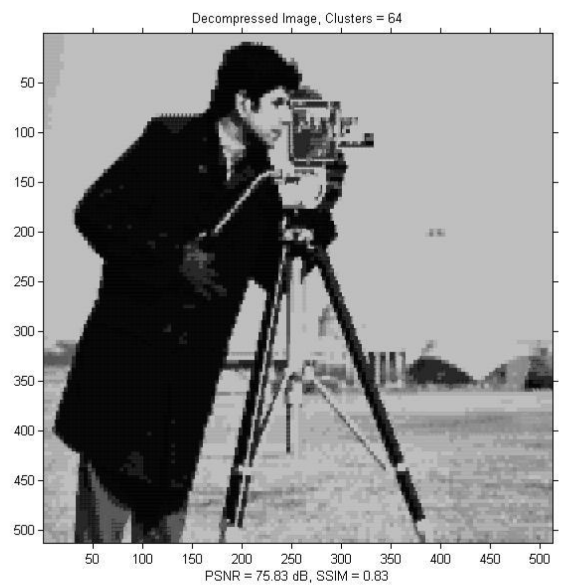

Figure 11. Cameraman Decompressed with 64 Clusters.

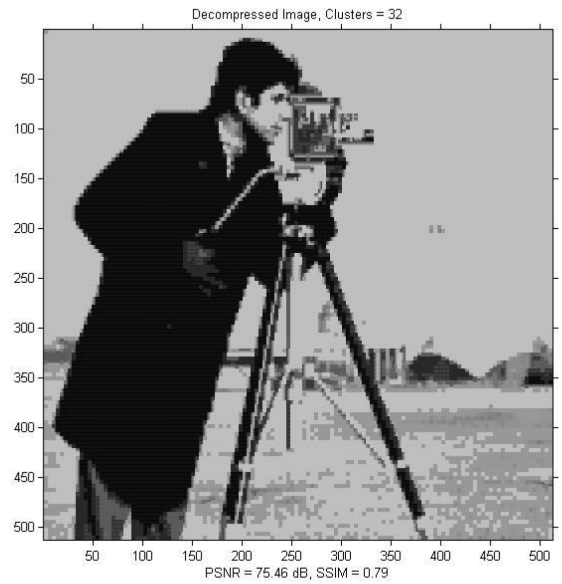

Figure 12. Cameraman Decompressed with 32 Clusters.

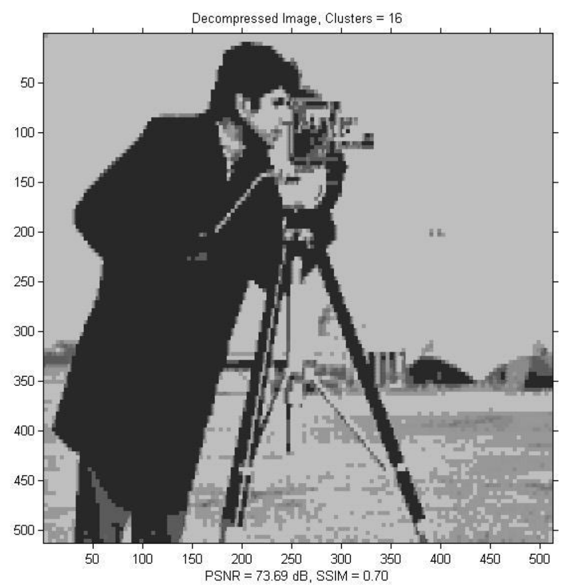

Figure 13. Cameraman Decompressed with 16 Clusters.

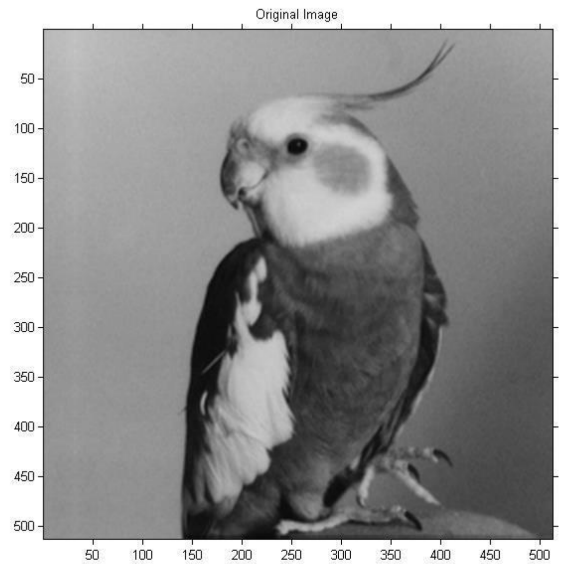

Figure 14. Bird Original Image.

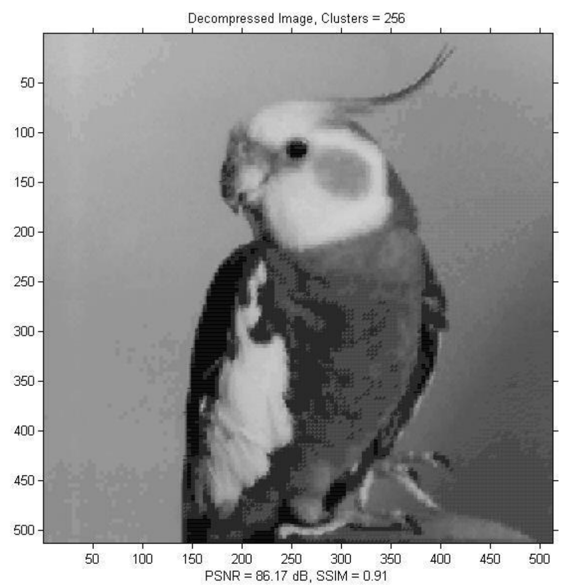

Figure 15. Bird Decompressed with 256 Clusters.

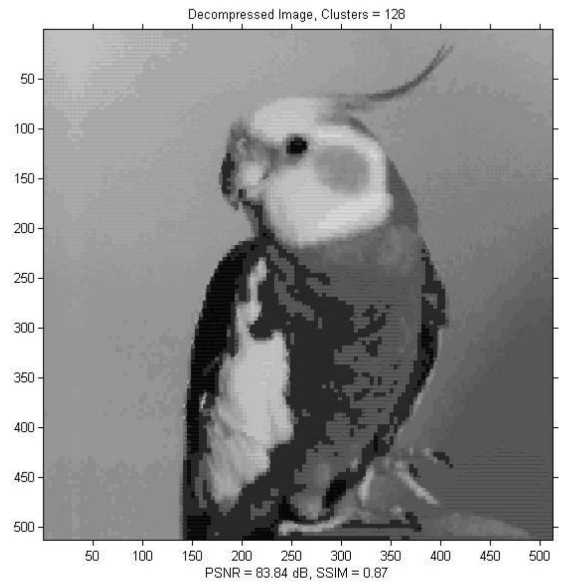

Figure 16. Bird Decompressed with 128 Clusters.

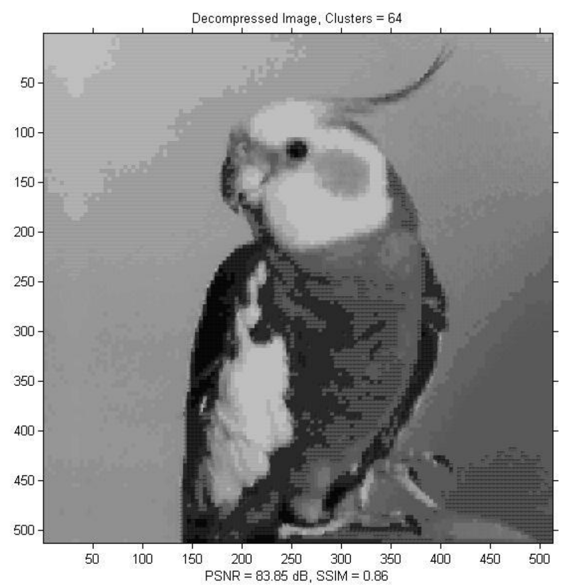

Figure 17. Bird Decompressed with 64 Clusters.

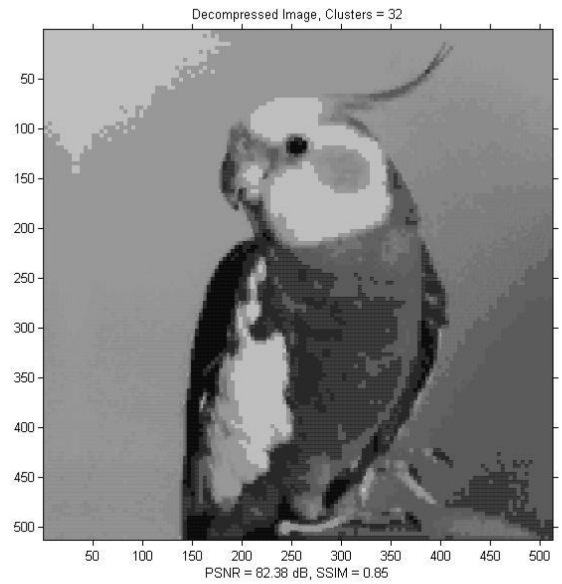

Figure 18. Bird Decompressed with 32 Clusters.

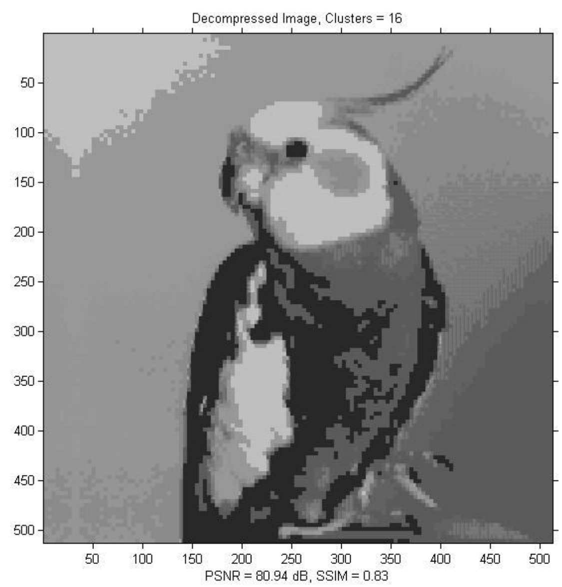

Figure 19. Bird Decompressed with 16 Clusters.

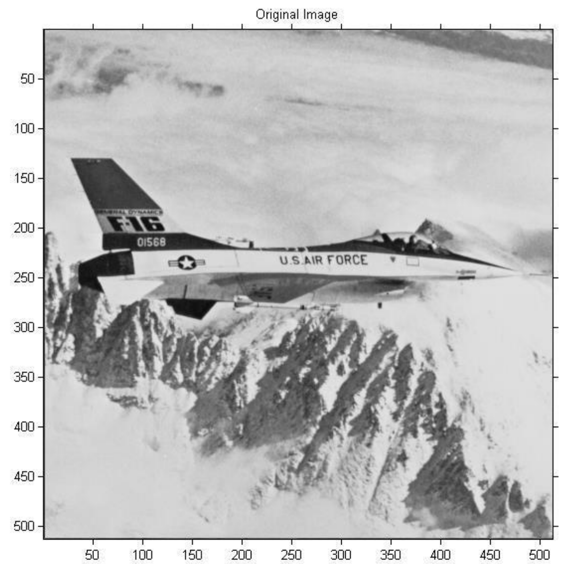

Figure 20. Airplane Original Image.

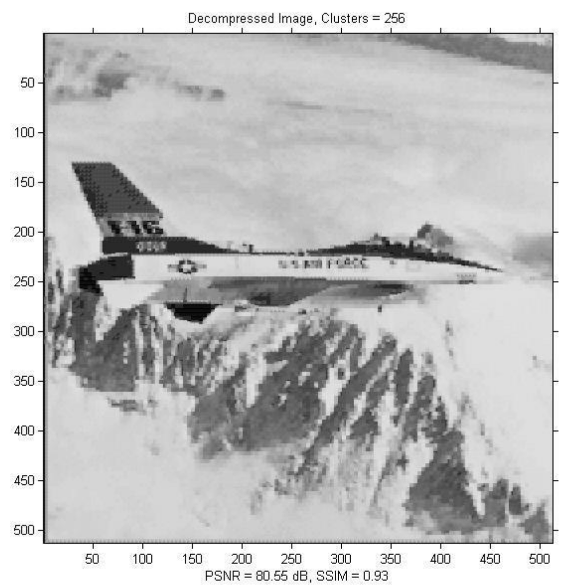

Figure 21. Airplane Decompressed with 256 Clusters.

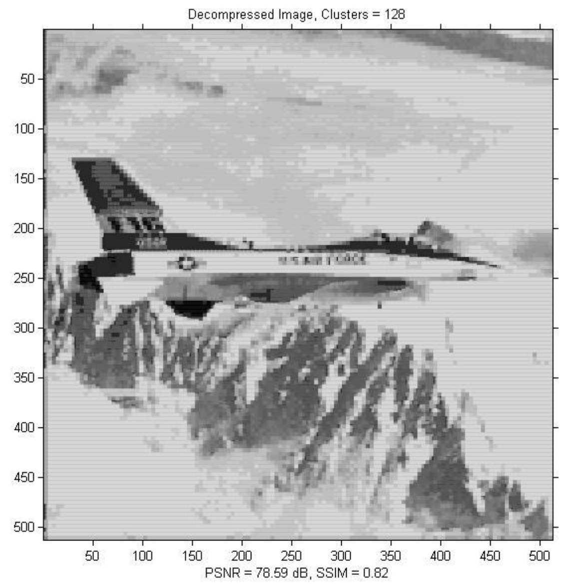

Figure 22. Airplane Decompressed with 128 Clusters.

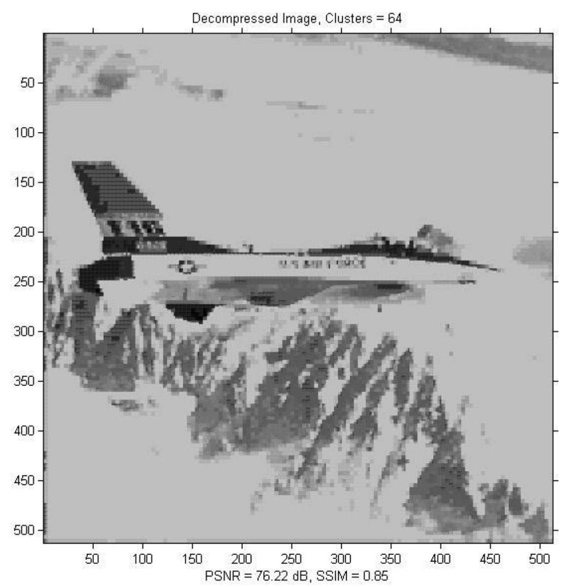

Figure 23. Airplane Decompressed with 64 Clusters.

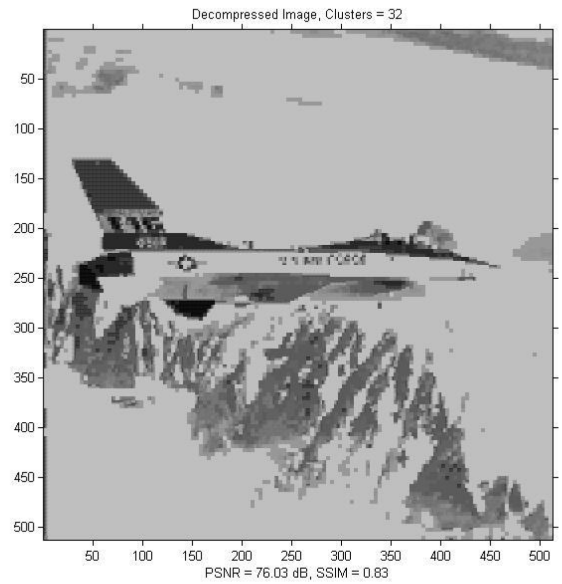

Figure 24. Airplane Decompressed with 32 Clusters.

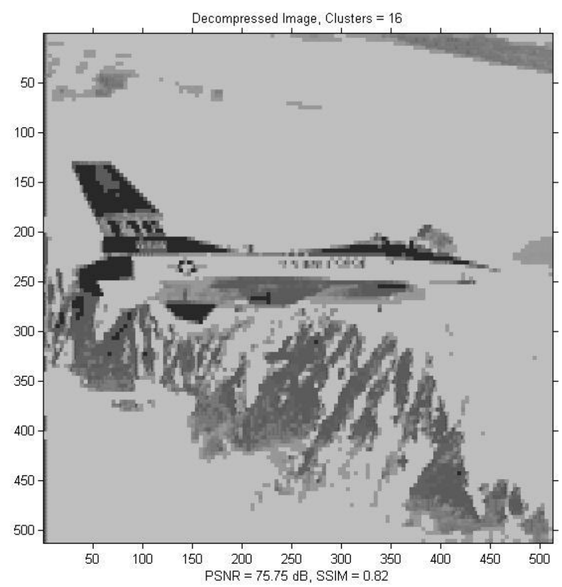

Figure 25. Airplane Decompressed with 16 Clusters.

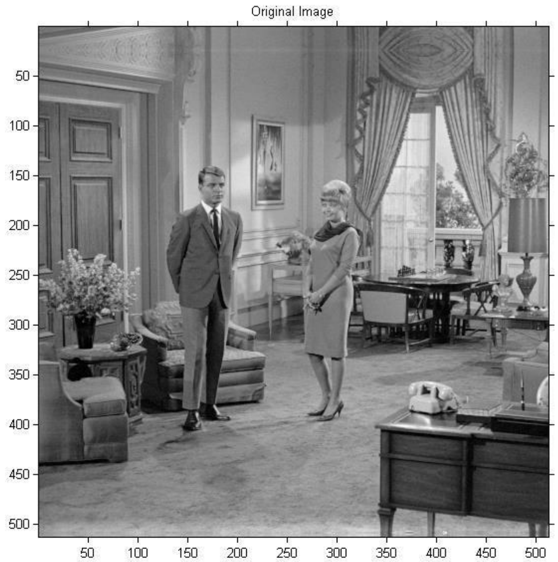

Figure 26. Couple Original Image.

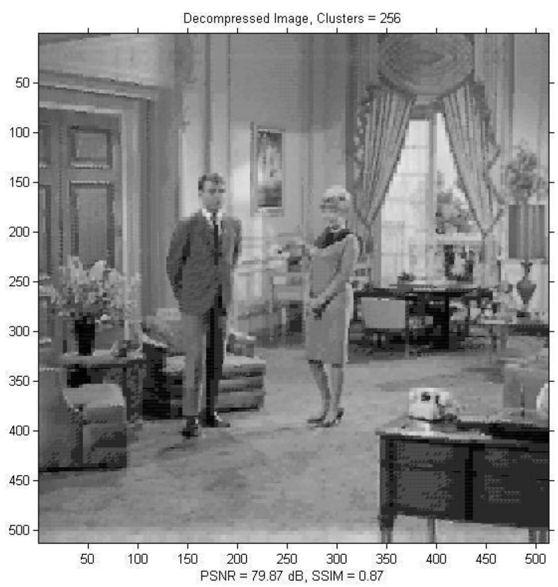

Figure 27. Couple Decompressed with 256 Clusters.

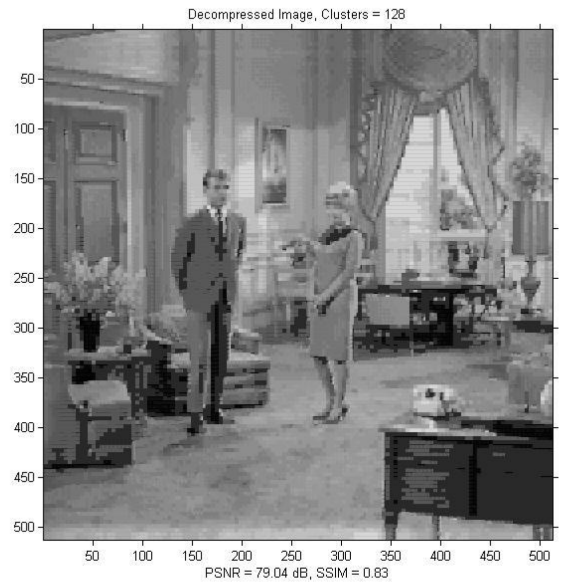

Figure 28. Couple Decompressed with 128 Clusters.

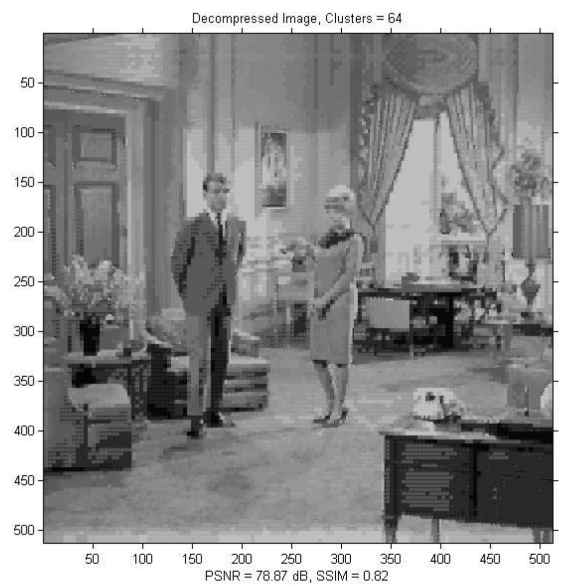

Figure 29. Couple Decompressed with 64 Clusters.

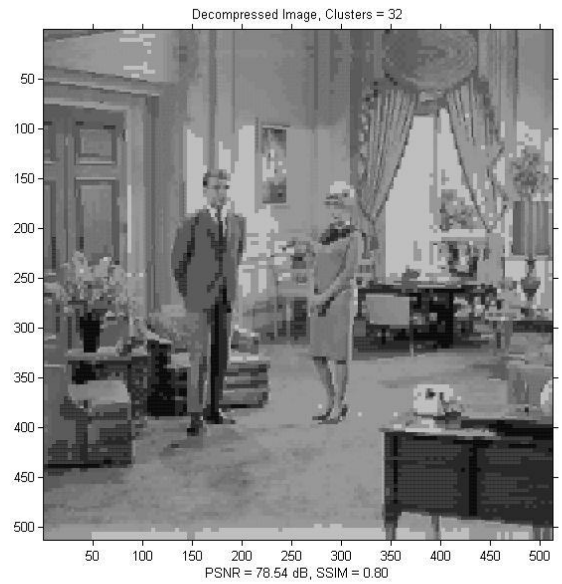

Figure 30. Couple Decompressed with 32 Clusters.

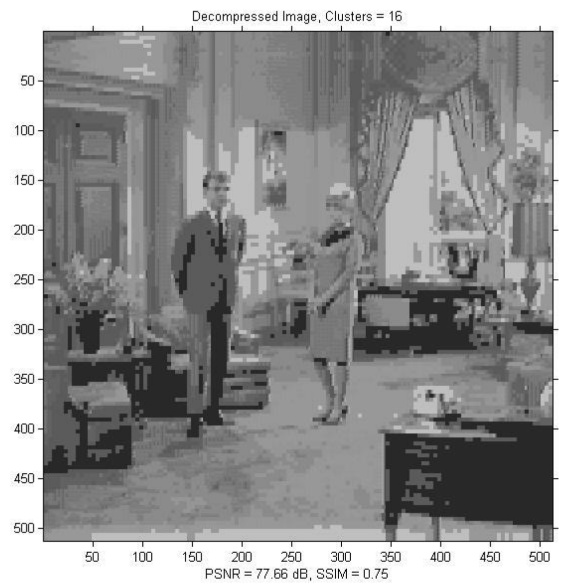

Figure 31. Couple Decompressed with 16 Clusters.

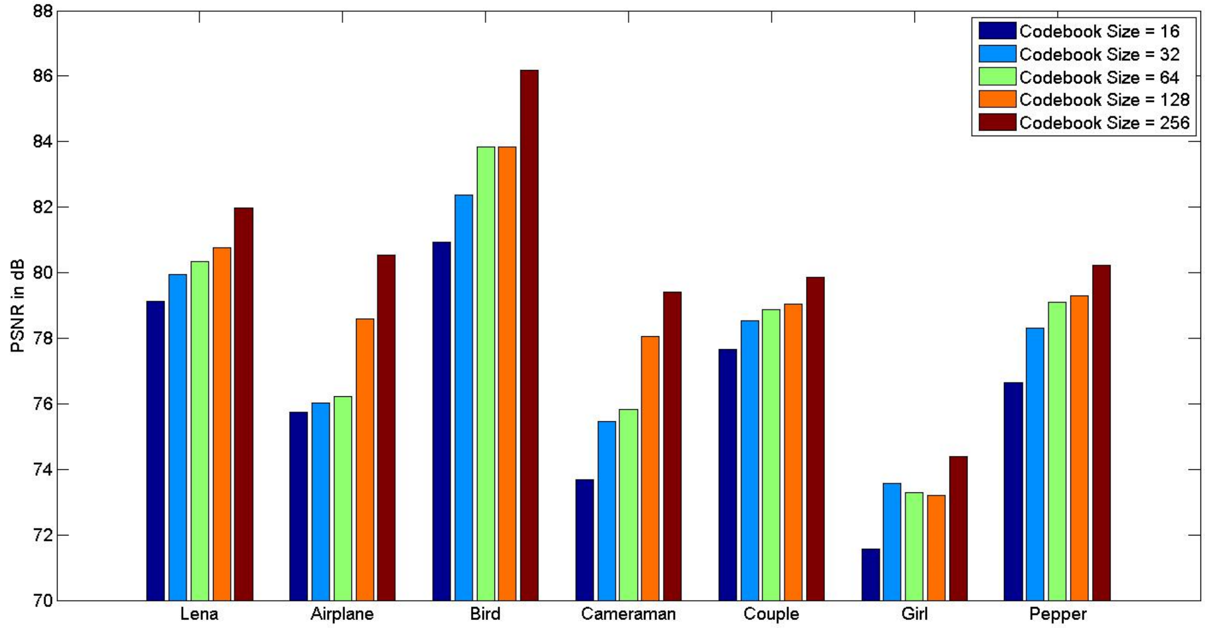

Figure 32. Comparison of PSNR Values for Different Codebook Sizes.

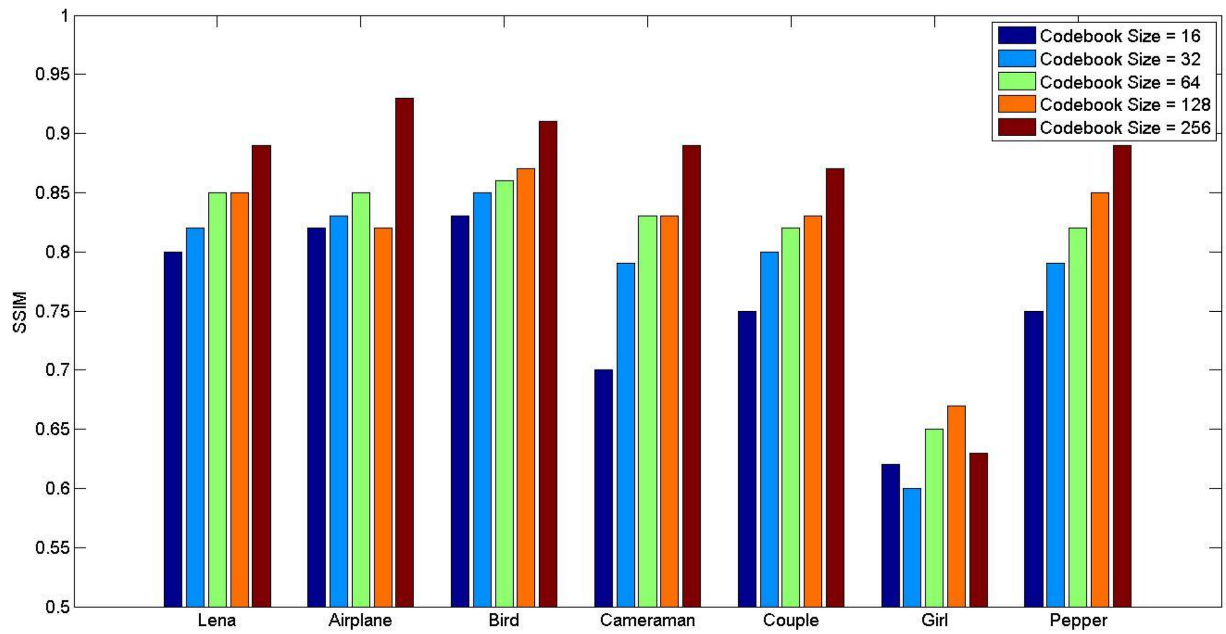

Figure 33. Comparison of SSIM Values for Different Codebook Sizes.

The PSNR and SSIM values for all the test and training images are presented in Figure 32 and Figure 33, respectively.

From the above figures, it is apparent that the quality of the reconstructed image improves with the increase in codebook size that is an increase in the cluster count in each codebook.

CONCLUSION AND FUTURE WORK

In this paper, a review of different methods of picture compression was done. The image compression can be lossy and lossless in nature. The advance made in the field of image information pressure in the past two decades is vast. Different quantization strategies, entropy coding and numerical change are presented for image compression. The thought behind each new strategy is the better execution contrast compared to past techniques. Image compression utilized at various pictures like medicinal pictures, common pictures, simulated pictures and satellite pictures, etc. fundamentally data compression is most relevant when we have to transmit or store a colossal measure of information since the idea of pictures are changing from two-dimensional pictures to three-dimensional pictures. The present information pressure techniques may be far from a definitive breaking point. Fascinating issues like getting precise models of pictures, ideal portrayals of such models, and quickly figuring such ideal portrayals are the great difficulties confronting the information compression group. Picture coding considering models of human observation, adaptability, heartiness, blunder flexibility, and 36 multifaceted nature are a couple of the many difficulties in picture coding to be completely settled and may influence image data compression execution in the years to come.

In this paper, a neural-network algorithm for edge preserving image compression was implemented. A simulated neural network is utilized for picture compression. Pictures were given as the input to the system, and different compressed pictures were created, contingent upon the variety of neural network parameters and the cluster count.

The WCNN algorithm allots more representative vectors from the edges of the image than the interior of the image thus helping in better edge preservation of the reconstructed image. After experimenting with the cluster count it is evident that with the increase in the number of cluster the quality of the picture is improved, which is the expected behavior as more clusters leads to more representational vectors. The increase in codebook sizes with the increase in the cluster count reiterate the fact that there is an increase in representational vectors which results in improved image quality.

The future work could be reached out to the arrangement of uses to cover much more regions of image and video processing, even for the captcha age.

REFERENCES

[1] Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. IEEE transactions on Computers, 100(1), 90-93.

[2] Burrus, C. S., Gopinath, R. A., & Guo, H. (1997). Introduction to wavelets and wavelet transforms: a primer.
Retrieved from “http://www2.ee.ufpe.br/codec/burrus.pdf”

[3] Gersho, A., & Ramamurthi, B. (1982, May). Image coding using vector quantization. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82. (Vol. 7, pp. 428-431). IEEE.

[4] Baker, R. L., & Gray, R. M. (1982, October). Image compression using non-adaptive spatial vector quantization. In Conf. Rec. 16th Asilomar Conf. Circuits Systems and Computers (pp. 55-61).

[5] Ramamurthi, B., & Gersho, A. (1986). Classified vector quantization of images. IEEE Transactions on communications, 34(11), 1105-1115.

[6] Kim, Y. K., & Ra, J. B. (1995). Adaptive learning method in self-organizing map for edge preserving vector quantization. IEEE Transactions on Neural Networks, 6(1), 278-280.

[7] Kohonen, T. (2012). Self-organization and associative memory (Vol. 8). Springer Science & Business Media.

[8] Linde, Y., Buzo, A., & Gray, R. (1980). An algorithm for vector quantizer design. IEEE Transactions on communications, 28(1), 84-95.

[9] Yamada, Y., Fujita, K., & Tazaki, S. (1979). A consideration on the block quantization of the image. J. IECE Japan, 9-12.

[10] Ramamurthi, B., & Gersho, A. (1984, March). Image vector quantization with a perceptually-based cell classifier. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84. (Vol. 9, pp. 698-701). IEEE.

[11] Park, D. C. (1997, June). Development of a neural network algorithm for unsupervised competitive learning. In Neural Networks, 1997., International Conference on (Vol. 3, pp. 1989-1993). IEEE.

[12] Park, D. C. (2000). Centroid neural network for unsupervised competitive learning. IEEE Transactions on Neural Networks, 11(2), 520-528.

[13] Limb, J. O. (1979). Distortion criteria of the human viewer. IEEE Transactions on systems, Man, and Cybernetics, 9(12), 778-793.

[14] Park, D. C., & Woo, Y. J. (2001). Weighted centroid neural network for edge preserving image compression. IEEE transactions on neural networks, 12(5), 1134-1146.

[15] Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. Electronics letters, 44(13), 800-801.

[16] Hore, A., & Ziou, D. (2010, August). Image quality metrics: PSNR vs. SSIM. In Pattern recognition (icpr), 2010 20th international conference on (pp. 2366-2369). IEEE.