

A Technique for Image Compression by using GSOM Algorithm

Aslam Khan
M.Tech (IT) Technocrats
Inst. of Tech
Bhopal (MP) India

Sanjay K Mishra
Asst.Prof, Dept.of IT
Bhopal (MP) India

Shiv K. Sahu, Ph.D
Assoc.Prof.& Head, Dept.
of IT
Technocrats Inst.Of Tech
Bhopal (MP) India

Amit Mishra
Asst.Prof, Dept of IT
Technocrats Inst.of Tech
Bhopal (MP) India

ABSTRACT

In this research work a technique is proposed for image compression which is based on Neural Network. Here image compression is done by using Growing Self Organizing Map algorithm (GSOM). This proposed work may be compression up to 90% of the source files.

Keywords

Neural Network, Image Compression.

1. INTRODUCTION

With the rapid development of digital technology in consumer electronics, device demand to preserve raw image data for further editing or repeated compression is increasing.[14] In the context of image (picture) processing, compression schemes are aimed to reduce the transmission rate for images while maintaining a good level of visual quality. Compressing an image is significantly different than compressing raw binary data. Common purpose compression programs can be used to compress images, but the result is not better.

Image compression is a problem of reducing the amount of data required to represent a digital image. It is a process intended to give in a compact representation of an image, thus by reducing the image storage transfer requirements. The reduction in image size allows more images to be stored in a given amount of disk or space of memory. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. Compression is achieved by the removal of one or more of the three basic data redundancies

1. Coding Redundancy
2. Inter pixel Redundancy
3. Psycho visual Redundancy

Coding redundancy is present when less than optimal code words are used. Inter pixel redundancy results from correlations between the pixels of a picture. Psycho visual redundancy is due to data that is ignored by the human visual system (visually non essential information).[14] Image compression techniques decrease the number of bits required to represent an image by taking advantage of these redundancies.

2. RELATED WORK FOR DATA COMPRESSION

2.1 Huffman Coding

Huffman coding is a commonly method for compressing data with variable-length code. Given a set of data symbols and their frequencies of occurrence the method constructs a set of

variable-length code-words with the shortest average length and assigns them to the word. Huffman coding serves as the basis for several applications implemented on popular platforms. A few programs use just the Huffman method, while others use it as one step in a multistage compression process. The main disadvantages of Huffman Coding are

- 1.The Compression ratio is low at higher size of image file
- 2.The efficiency is lowest
- 3.The redundancy is not reduced

In Huffman coding, the binary strings or codes in the encoded data are all different lengths. This makes it problem for decoding software to determine when it has reached the last bit of data and if the encoded data is distorted. In other words it contains spurious bits or has bits missing it will be decoded incorrectly and the output will be nonsense.

2.2. Shannon-Fano Method

Shannon–Fano coding is a technique for constructing a prefix code based on a set of symbols and their probabilities (estimated or measured). It is sub optimal in the sense that it does not achieve the lowest possible expected code word length like Huffman coding; however unlike Huffman coding, it does assurance that all code word lengths are within a bit of their theoretical ideal [15].

In Shannon–Fano coding, the symbols are arranged in order from most probable to least probable, and then separated into two sets whose total probabilities are as close as possible to being equal

The main disadvantage of Shannon-Fano Coding is following

- 1) In Shannon-Fano coding, we cannot be sure about the codes generated. There may be two distinct codes for the same symbol depending on the way we build our tree.
- 2) Also, here we have no unique code that is a code might be a prefix for another code. So in case of errors or crush during data transfer, we have to start from the beginning.
- 3) Shannon-Fano coding does not assurance optimal codes. Hence, Shannon-Fano coding is not very efficient

2.3. Arithmetic Coding

Arithmetic coding [4] bypasses the idea of replacing input symbols with a single floating point output number. More bits are considered necessary in the o/p number for longer, complex messages. This conception has been known for a few time, but only newly were practical methods found to

implement arithmetic coding on computers with fixed sized-registers. The output from an arithmetic coding method is a single number less than 1 and greater than or equal to zero. The single number can be individually decoded to create the exact stream of symbols that went into structure. Arithmetic coding seems more complex than Huffman coding, but the size of the program compulsory to apply it, is not radically different. Runtime presentation is radically slower than Huffman coding. If performance and squeezing the end bit out of the coder is important, arithmetic coding will always give as a better performance than Huffman coding. But watchful optimization is needed to get performance up to acceptable levels.

There are a few disadvantages of arithmetic coding. One is that the total codeword must be received to start decoding the symbols, and if there is a consider bit in the codeword, the entire message could become consider. Another is that there is a limit to the precision of the number which can be encoded, therefore limiting the number of symbols to encode within a codeword

2.4 LZ-77

Another technique for data compression is LZ-77 encoding [7]. This technique is a simple, clever, and useful approach to compress text files. This technique are exploits the fact that words and phrases within a text stream are likely to be repetitive. When they do again, they can be coded as a pointer to an earlier incidence, with the pointer accompanied by the number of characters to be matched. This technique is useful for compression text file because it able to reduce the file size and increase the compression ratio after compression. However, it is not useful for all image file format such bmp, gif, and tiff. nearly that, this technique will take some minutes to compress a data. Sometimes, the extendedly processing time will cause the missing of some characters.

The main disadvantage of LZ-77 is

1. This algorithm is time consuming
2. LZ77 is the limited size. When the data size is high for compression then mostly data reduced or corrupt during compression

2.5 LZW Method

The most popular technique for data compression is Lempel Ziv Welch (LZW) [8]. LZW is a general compression algorithm capable of working on almost any type of data. It is normally fast in both compressing and decompressing data and does not require the use of floating point operation. LZW technique also has been applied for text file. This technique is very capable to compress image file such tiff and gif. However, this technique not capable for compress text file because it require many bits and data dictionary.

The main disadvantage of LZW-77 is

1. This technique is most expensive other compression technique
2. Mostly content are lost during compression step

2.6 Run Length Method

One of the techniques for data compression is Run-length-encoding, which is few time knows as Run-length-limiting (RLL) [5, 6]. Run length encoding is very helpful for solid black image (picture) This technique can be useful to compress text especially for text file and to find the repeating string of characters. This compress software will scan through the file to find the repeating string of characters, and store them using escape character (ASCII 27) followed by the

character and a binary count of he number of items it is repeated.

The main disadvantage of Run length algorithm is

1. First problem with this technique is the output file is bigger if the decompressed input file includes lot of escape characters.
2. Second problem is that a single byte cannot specify run length greater than 256.
3. they disconnect the outer error-correcting code from the bit-by-bit likelihoods that come out of the channel

So, we apply the proposed approach such as GSOM Algorithm that can remove the above disadvantage of various traditional algorithms.GSOM Provide optimal value of compression ratio of image file. It require no more time for compression.

3. NEURAL NETWORK BASED METHOD FOR IMAGE COMPRESSION

Artificial Neural Networks have been applied to many problems [3][11], and have demonstrated their superiority over classical methods when dealing with noisy or shortened data. One such application is for data compress. Neural networks seem to be well suitable to this particular function, as they have a capacity to preprocess input patterns to produce simpler patterns with fewer components[16],[9]. This compressed information (stored in a hidden layer) preserves the full information obtained from the external environment. The compressed features may be then exit the network into the external environment in their original uncompressed form. The main algorithms that will be discussed in ensuing sections are the Back propagation algorithm and the Kohonen self-organizing maps.

3.1 Back propagation Neural Network

The Back propagation (BP) algorithm [12] has been one of the most successful neural network algorithms applied to the problem of data compression [10]. The data compression problem in the case of the BP algorithm is posed as an encoder problem. The data or image to be compressed passes through the input layer of the network, and then sub-sequently through a very small number of hidden neurons. The hidden layer that the compressed features of the image are stored, therefore the smaller number of hidden neurons, the high compression ratio. The output layer sub-sequently output the decompressed image to the external environment. It is a expected the input and output data are the same or very close. If the image or image file to be compressed is very large, this may be few times cause complexity in training, as the input to the network becomes very large. That case of large images, they may be broken down into minor, sub-images [9]. Each sub-image may then be used to train an individual ANN.

The main disadvantage of Back propagation algorithm is

1. In this technique, the error rate is high during image compression
2. It take more time for image compression and decompression
3. It is expensive technique
4. This technique is apply only for specific image format such as Bitmap ,Joint Photographic Experts Group ,GIF File.

So, we apply the proposed approach such as GSOM Algorithm that will remove the above disadvantage and

improve the compression ratio with quality and provide better result compared to traditional compression algorithm.

4. PROPOSED TECHNIQUES FOR IMAGE COMPRESSION

4.1 Growing Self Organizing Map Algorithm

A growing self-organizing map (GSOM) is a growing variant of the popular self-organizing map (SOM). The GSOM was developed to address the issue of identifying a suitable map size in the SOM. It starts with a minimal number of nodes (usually 4) and grows new nodes on the boundary based on a heuristic. By using the value called (SF) Spread Factor, data analyst has the ability to control the growth of the GSOM.

All the starting nodes of the GSOM are boundary nodes, that is each node has the freedom to grow in its own direction at the beginning. New Nodes are grown from the boundary nodes. An once a node is selected for growing all its free neighboring positions will be grown new nodes. In GSOM, input vectors are organized into categories depending on their similarity to each other. For data compression, the image file or data is broken down into smaller vectors for use as input. For Euclidean distance each input vector presented, to all the output nodes are computed. The weights of the node with the bare minimum distance, all along with the neighboring nodes are adjusted. This ensures that is the output of these nodes is slightly enhanced. This process is repeated until some condition for extinction is reached. After a sufficient number of input vectors have been presented, every output node becomes sensitive to a group of similar input vectors, and may therefore be used to represent character of the input data. This means that for a very high numbers of input vectors passed into the network, (uncompressed data or image), the compressed form will be the data exiting from the output nodes of the network (considerably smaller number). This compressed data may be further decompressed by another network. We take 50 neuron as a one input hidden layer and one output layer we take learning rate 0.5. the compression and decompression figure of GSOM Algorithm are following

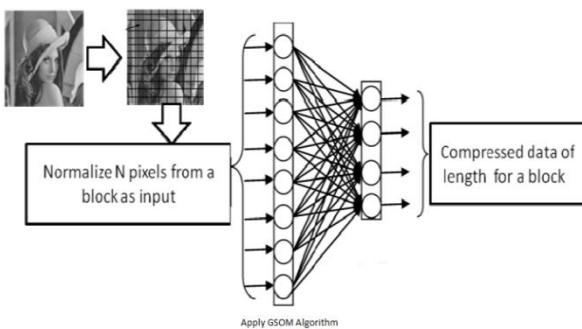


Fig.1: GSOM Architecture compression

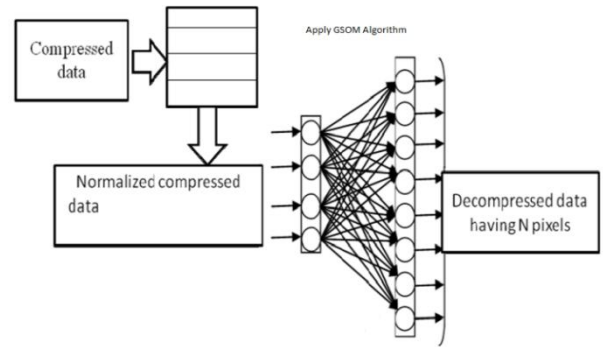


Fig.2: GSOM Architecture decompression

4.2 The Learning Algorithm of the GSOM

The GSOM process is as follows:

1. Initialization phase:

1. Initialize the weight vectors of the starting nodes (usually four) with random numbers between 0 and 1.
2. Calculate the growth threshold (GT) for the given data set of dimension D according to the spread factor (SF) using the formula $GT = -D \times \ln(SF)$

2. Growing Phase:

1. Present input to the network.
2. Determine the weight vector that is closest to the input vector mapped to the current feature map (winner), using Euclidean distance. This step can be summarized

as: find q such that $|v - w_q| \leq |v - w_q| \forall q \in \mathbb{N}$ where v, w are the input and weight vectors respectively, q is vector position for nodes and \mathbb{N} is the set of natural numbers.

3. The weight vector adaptation is applied only to the neighborhood of the winner and the winner itself. The neighbor is a set of neurons around the winner, but in the Growing self organization map the starting neighborhood selected for weight adaptation is smaller compared to the SOM (localized weight adaptation). The amount of adaptation (learning rate) is also reduced exponentially over the iterations. Even with the neighborhood, value (weights) that are closer to the winner are adapted more than those further away. The weight adaptation may be explain by

$$w_j(k+1) = \begin{cases} w_j(k) & \text{if } j \notin N_{k+1} \\ w_j(k) + LR(k) \times (x_k - w_j(k)) & \text{if } j \in N_{k+1} \end{cases}$$

where the Learning Rate $LR(k)$, $k \in \mathbb{N}$ is a sequence of positive parameters converging to zero as $k \rightarrow \infty$. $w_j(k)$, $w_j(k+1)$ are the weight vectors of the node j before and after the adaptation and N_{k+1} is the neighborhood of the winning neuron at the $(k+1)$ th iteration. The decreasing value of $LR(k)$ in the GSOM depends on the number of nodes existing in the map at time k .

4. Increase the error value of the winner (error value is the difference between the input vector and the weight vectors).
5. When $TE_i > GT$ (where TE_i is the total error of node i and GT is a boundary node) growth threshold. Distribute weights to neighbors if i is a non-boundary node.

6. Initialize the new node weight vectors to match the neighboring node weights.
7. Initialize the learning rate (LR) to its starting value.
8. Repeat steps 2 – 7 until all inputs have been presented and node growth is reduced to a minimum level.

3.Smoothing phase:

Reduce learning rate and fix a small starting neighborhood. Find winner and get used to the weights of the winner and neighbors in the same way as in growing phase.

4.2.1 Encoding

The trained network is now ready to be used for image compression which, is achieved by dividing the input images into blocks after that scaling and applying each block to the input of Input Layer (IL) then the output of Hidden layer HL is quantized and entropy coded to represent the compressed image. Entropy coding is tradition (lossless) compression that will further squeeze the image; for instance, Huffman coding code can be used here. Fig. (3) Show the encoding steps.

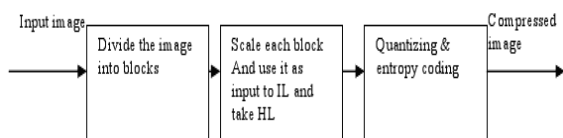


Fig.3: Encoding process of image compression

4.2.2 Decoding:

To decompress the image; first decode the entropy coding then apply it to the output of the hidden layer and get the output of the compressed Image. Fig.(4) show the decoder block diagram.

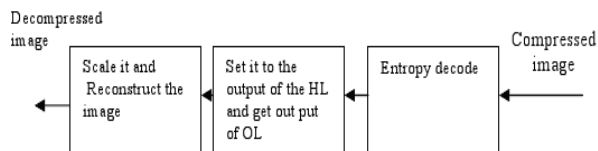


Fig.4: Decoding process of image compression

5. RESULT ANALYSIS

In order to test the compression software, several files of different formats were run through the scheme. This system uses three metrics such as compression ratios of various transmission time. Compression ratios are defined as [9].

$$Z = \frac{\text{Compressed Length (in bytes)} \times 100}{\text{Total Length (in bytes)}}$$

$$\text{Compression Ratio} = 100 - Z$$

Table 1: show the compression ratio of various file such as BMP, JPG, TIFE

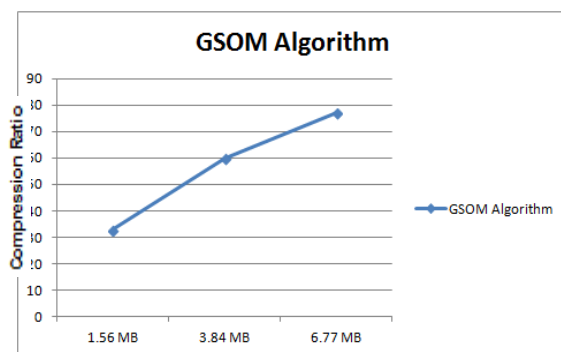
Table1: compression ratio of image file

BMP File Compression			
File Size	Original byte	GSOM Algorithm	Compression Ratio
1.56 MB	1.644,945	1.15 MB	33%
3.84 MB	4.029,002	1.54 MB	60%
6.77 MB	7.7101,066	3.41 MB	77%

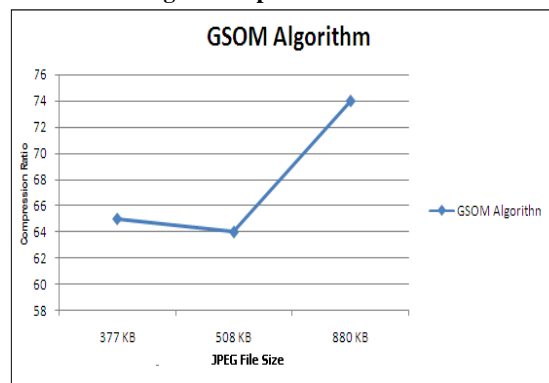
JPEG File compression			
File Size	Original byte	GSOM Algorithm	Compression Ratio
377 KB	386,098	323 KB	65%
508 KB	520.4	377 KB	64%
880 KB	901,213	508 KB	75%

TIFF File Compression			
File Size	Original byte	GSOM Algorithm	Compression Ratio
1.20 MB	1,269,350	926 KB	88%
2.17 MB	2,277,422	1.19 MB	89%
3.66 MB	3,846,382	2.15 MB	75%

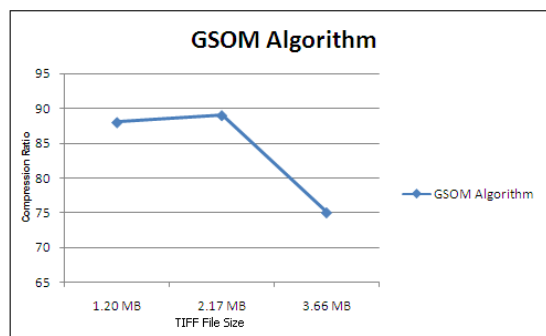
The graph for different compression ratio is



BMP File Size
Fig.5: Compression in BMP



JPEG File Size
Fig.6: Compression in JPG



TIFF File Size
Fig.7: Compression in TIFF

Identifying files as belonging to particular classes offers a quicker way of encoding and decoding the files. The GSOM Algorithm offers us a way of encoding knowledge as a set of training examples rather than by a set of rules and is an effective technique for problem domains where there are many rules or the rules cannot be easily devised. This research also shows that conventional computing and artificial neural networks are not in conflict with each other but each can be exploited for the advantages they offer.

6. FUTURE ENHANCEMENT

1. GSOM algorithm for data compression being a wide field which is rapidly finding use in many applied fields and technologies

2. GSOM has some limitation. They can not compress higher size of audio and video file. So To Improve the Compression Ratio of higher size of audio and video file in future enhancement

7. CONCLUSION

This work is focused on the data compression by using Growing Self Organizing Map algorithm. The Proposed algorithm is save more memory space and in case of web applications, transfer of images and download rapidly . The propose approach (GSOM) are used to compressed the various image file such as BMP,JPG,TIFE etc and compression ratio is batter then other traditional algorithm such as Huffman ,LZW, Arithmetic etc.The proposed Algorithm has been trained and tested for the analysis of different type of images. It has been practical that the convergence time for the training of GSOM Algorithm is very faster. Different attributes of compression such as compression ratio are calculated. It has been observed that there is implication change in compression ratio from 33% to 90%.The Proposed technique of initialization of weights exhibits fast rate of convergence and using the trained weight sets, good quality of regenerated images are available at the receiving end.

8. REFERENCES

- [1] K.Siva Nagi Reddy, Dr. B.R.Vikram, L. Koteswara Rao"Image Compression and reconstruction using a new approach by artificial neural network", International journal of image processing ", volume -6, Issue-2, 2012, pp:68-85.
- [2] R. C.Gonzales, R. E. Woods, Digital Image Processing, Second Edition, Prentice-Hall, 2002.
- [3] Veisi H., Jamzad M., Image Compression Using Neural Networks, Image Processing and Machine Vision Conference (MVIP), Tehran, Iran, 2005.
- [4] Missing Data Estimation Using Principle Component Analysis and Auto associative Neural Networks. Jaisheel Mistry, Fulufhelo V. Nelwamondo, Tshilidzi Marwala. 3, s.l. : iisc.org, Journal of Systemics, Cybernatics and Informatics, Vol. 7, pp. 72-79, 2009.
- [5] B. Verma, M. Blumenstein, S. Kulkarni, "A new compression technique using an artificial neural network". Faculty of Information and Communication Technology, Griffith University, Australia, 2004.
- [6] M. Egmont-Petersen, D. de Ridder, H. Handes, "Image processing with neural networks- a review", Pattern Recognition , 35, pp. 2279- 2301, 2002.
- [7] J. Jiang, "Image compression with neural networks – A survey, Signal Processing": Image Communication, 14, pp. 737 – 760, 1999.
- [8] S. Anna Durai and E. Anna Saro "Image Compression with Back-Propagation Neural Network using Cumulative Distribution Function" World Academy of Science, Engineering and Technology 17, 2006.
- [9] Martin Riedmiller and Heinrich Braun "A Direct Adaptive Method for Faster Back propagation Learning: The RPROP Algorithm" Institute fur Logic, Komplexitat und Deduktionssysteme, University of Karlsruhe, W-7500 Karlsruhe, FRG, 1993.
- [10] Md. I. Bhuiyan, Md. K. Hassan, "Image compression with neural network using DC algorithm", Journal of signal processing, vol. 5, no. 6, pp. 445 -459, 2001
- [11] S.S Panda, M.S.R.S Prasad Ch.SKVR Naidu"Image compression using backpropagation neural network", International journal of engineering science and advance technology volume -2,issue-1,2012 pp:74-78 ISSN:2250-3676.
- [12] Neelmani chhedaiya, prof. Vishal Moyal "Implementation of Backpropagation algorithm in Verilog",International Journal Computer Technology and application volume-3,issue-1,2012, pp:340-343:ISSN:2229-6093.
- [13] Anuj Sharma & Mahendra Pratap Panigraphy" neural network and image compression", VSRD-International journal of computer science and information technology",volume-2,Issue-9,2012,pp:746-755.
- [14] P.Nirupama 1 Senior Assistant Professor in CSE Dept, BVRIT 2 . Associate Professor in SE Dept "Analysis of Various Image Compression Techniques" . vol,2no.4, May 2012. ARPN Journal of science and Technology
- [15] Mamta Sharma S.L. Bawa D.A.V. college Compression Using Huffman Coding IJCSN International Journal of Computer Science and Network Security, VO .10 No.5, May 2010.
- [16] Sangeeta Mishra TCET, MumbaiThakur Village, Kandivli(E)-101 "Image Compression Using Neural Network" International Conference & workshop on Recent Trends in Technology, (TCET) 2012 Proceedings published in International Journal of Computer Applications(IJCA)