A Survey of image compression using neural network

Khushboomishra[1],Assistant Prof. M.Hasnine Mirja [2]
[1,2]Department of Electronics & Communication, oriental institute of science and technology , Bhopal
Email: [1]khushboo.mishra2389@gmail.com, [2]its hasninemirja@gmail.com

***Abstract* - This paper presents overview of neural network approaches to image compression as signal processing tools. Fractal image compression is a comparatively recent technique based on the representation of an image by a contractive transform, on the space of images, for which the fixed point is close to the original image .Image compression helps to reduce the storage space and transmission cost.**

Artificial Neural network (ANNs) is a training algorithm has used to compress the image. Artificial neural network is exceptionally Feed Forward Back propagation neural network (FFBPNN) in which Neural network has trained by Back propagation neural network algorithm for image compression and decompression. Many techniques and algorithms are used to train Artificial neural network by considering the different number of hidden neurons ,epoch and reconstructed image compared with original image. Conjugate Gradient Algorithm compressed the digital image TIF, JPEG,PNG and BMP with 64 input neurons,13 or 64 in hidden layer that are determining compression rate and 64 output neurons .Training algorithm and developed architecture provide better result.

***Keywords*—artificial neural network, Image Compression/Decompression, Back propagation Algorithm**

I. INTRODUCTION

Images require large amounts of memory for storage and the transmission of image from one computer to another over the web can be very time consuming. Image compression is a method through which we can reduce the storage space of image, videos which will be helpful to increase the storage and transmission process performance. For example a1024 x 1024 color image with 8 bit/pixel generates 25Mbits data, which without compression requires about 7 minutes of transmission time over 64kbps line. Approximately 224 GB is needed to store an uncompressed two-hour SD movie. Therefore, to fit such movie on a standard DVD-9, data must be compressed by a factor of approximately 26.3. Image Compression addresses the problem of reducing the amount of data required to represent an image as well as to maintain the visual quality of an image. By using compression techniques, it is possible to remove some of the redundant information contained in images, thus requiring less storage space and less time to transmit. The existing traditional techniques mainly are based on reducing redundancies in coding , inter pixel and psycho visual representation [1]. New soft computing technologies such as neural networks are being developed for image compression. Adaptive learning, self-organization, noise suppression, fault tolerance, and optimized approximations are some main reasons that encourage researchers to use artificial.

Recently there has been a tremendous growth of interest in the field of neural networks. Yet the wide range of applications and architectures which fall under this category make a specific definition of the term “neural network” difficult. A neural network can be defined as a “massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use” [5]. Generally, neural networks refer to a computational paradigm in which a large number of simple computational units, interconnected to form a network, perform complex computational tasks. There is an analogy between such a computational model and the functioning of complex neurobiological systems. Higher-order neurobiological systems, of which the human brain is the ultimate example, perform extremely complex tasks using a highly connected network of neurons in which each neuron performs a relatively simple information processing task.

Artificial neural networks have been applied to an image compression problem as they have the ability to process input pattern to produce simple patterns with fewer component [2]. The widely used Back-Propagation neural network has the merits of simple structure, stable state and easy hardware implementation, but its network training problems belong to high-dimensional optimization, which affects the algorithm precision because of its shortcomings of

long running time and local minimum value [2].

This model contrasts sharply with the classical serial computational model found in most general-purpose computers in use today. In the serial model, a highly complex processor perform computations in rigidly serial manner.In which the two models are programmed is also fundamentally different.In the classical model, explicit program steps or states provided to the processor which must account for all possible input states. . In contrast, neural networks are trained using examples of data which the networks will encounter. During training, the network forms an internal representation of the state space so that novel data presented later will be satisfactorily processed by the network.

In many application, neural network model may have a number of advantages over the serial model. Because of its parallel architecture neural network may break down some of the computational bottlenecks which limit the performance of serial machines. Since neural network are trained using example data,they can be made to adapt to changes in the input dat by allowing the training to continue during the processing of new data .another Advantage og training is that since data are presented individually, no overhead is required to store the entire training set .This particularly important when processing very large data sets, of which images are an example. The high degree of connectivity can allow neural networks to self-organize, which is an advantage when the structure of the data is not known beforehand. Finally, since there is an analogy between neural networks and neurobiological systems, existing biological networks could be used as models for designing artificial neural networks; it is hoped that some of the performance characteristics of the biological network will be inherited by the artificial network.

The sections of the paper are organized as follows.

II briefly reviews three of the major approaches to image compression: predictive coding, transform coding, and vector quantization. Section III discusses image compression techniques which use neural networks as nonlinear predictors. Transform coding methods using neural networks are presented in Section IV. These methods include linear principal components analysis (PCA) using Hebbian learning, auto associative coding, and adaptive coding. Section V discusses the application of the self-organizing feature map (SOFM) and its variations to vector quantization. Some of the issues relating to a comprehensive evaluation of the image compression tech-niques presented herein are discussed in Section VI. Finally, Section VII summarizes the important aspects of the methods presented and concludes the paper.

II. IMAGE COMPRESSION

The study of image compression methods has been an active area of research since the inception of digital image processing. Since images can be regarded as two-dimensional signals with the independent variables being the coordinates of a two-dimensional space, many digital compression techniques for one-dimensional signals can be extended to images with relative ease. As a result, a number of approaches to the problem are well established [6], [7], [8], [9], [10], [11], [12], [13], [14]. Most current approaches fall into one of three major categories: predictive coding, transform coding, or vector quantization. Alternatively, a combination of these techniques may be applied in a hybrid approach.

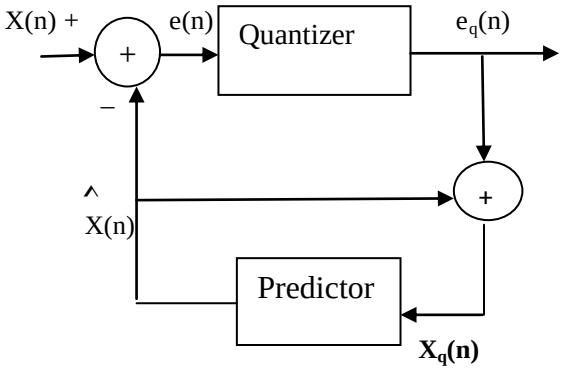

Figure1: Block diagram of a DPCM system

1. Predictive Coding

Typically, images exhibit a high degree of correlation among neighbouring samples. A high degree of correlation implies a high degree of redundancy in the raw data. Therefore, if the redundancy is removed by decor relating the data, a more efficient and hence compressed coding of the signal is possible. This can be accomplished through the use of predictive coding or differential pulse-code modulation (DPCM).

Figure 1 shows a block diagram for such a system. The predictor uses past samples $x(n-1); x(n-2); \ldots; x(n-p)$, or in the case of images, neighbouring pixels, to calculate an estimate, $\hat{x}(n)$, of the current sample. It is the difference

between the true value and the estimate, namely $e(n) = x(n) - \hat{x}(n)$, which is used for storage or transmission. As the accuracy of the predictor increases, the variance of the difference decreases resulting in a higher predictive gain and therefore a higher compression ratio.

The problem, of course, is how to design the predictor. One approach is to use a statistical model of the data to derive a function which relates the value of the neighbouring pixels to that of the current one in an optimal manner. An autoregressive model (AR) is one such model which has been successfully applied to images. For a pth order causal AR process, the nth values in the following manner:

$$X(n) = \sum_{j=1}^{p} w_j x(n-j) + e_n \qquad (1)$$

Where w_j is a set of AR coefficients, and e_n is a set of zero-mean independent and identically distributed (i.i.d.) random variables. In this case, the predicted value is a linear sum of the neighbouring samples (pixels) as shown by

$$X(n) = \sum_{j=1}^{p} w_j x(n-j) \qquad (2)$$

Equation 2 is the basis of linear predictive coding (LPC). To minimize the mean squared error $E[(\hat{x} - x)^2]$, the following relationship must be satisfied

$$R\,w = d \qquad (3)$$

Where, $[R]_{ij} = E[x(i)x(j)]$ is ijth element of the auto covariance matrix and, $d_j = E[\hat{x}(n)x(j)]$ is the jth element of the cross-covariance vector d. Knowing R and d, the unknown coefficients vector w can be computed, and the AR model (ie predicator) is thereby determined.

2. Transform Coding

Another approach to image compression is the use of transformations that operate on an image to produce a set of coefficients. A subset of these coefficients is chosen and quantized for transmission across a channel or for storage. The goal of this technique is to choose a transformation for which such a subset of coefficients is adequate to reconstruct an image with a minimum of discernible distortion.

A simple, yet powerful, class of transform coding techniques is linear block transform coding. An image is subdivided into non-overlapping blocks of $n \times n$ pixels which can be considered as N-dimensional vectors x with $N = n \times n$. A linear transformation, which can be written as an $M \times N$-dimensional matrix W with $M \times N$, is performed on each block with the M rows of W, w_i being the basis vectors of the transformation. The resulting M-dimensional coefficients vector y is calculated as

$$y = wx \qquad (4)$$

If the basis vectors w_i are orthonormal, that is If the basis vectors w_i are orthonormal, that is

$$w_i^T w_j = \begin{cases} 1; & i = j \\ 0; & i \neq j \end{cases} \qquad (5)$$

Then the inverse transformation is given by the transpose of the forward transformation matrix resulting in the reconstructed vector

$$X = W^T y \qquad (6)$$

The optimal linear transformation with respect to minimiz-ing the mean squared error (MSE) is the Karhunen-Loève transformation (KLT). The transformation matrix W consists of M rows of the eigenvectors corresponding to the M largest eigenvalues of the sample autocovariance matrix

$$\sum = E[xx^T] \qquad (7)$$

The KLT also produces uncorrelated coefficients and therefore results in the most efficient coding of the data since the redundancy due to the high degree of correlation between neighbouring pixels is removed. The KLT is related to principal components analysis (PCA), since the basis vectors are also the M principal components of the data. Because the KLT is an orthonormal transformation, its inverse is simply its transpose. A number of practical difficulties exist when trying to implements the above approach. The calculation of the estimate of the covariance of an image may be unwieldy and may require a large amount of memory. In addition, the solution of the Eigen-vectors and eigenvalues is computationally intensive. Finally, the calculation of the forward and inverse transforms is of order $O(MN)$ for each image block. Due to these difficulties, fixed basis transforms such as the discrete cosine transform DCT) [15], which can be computed in order $O(N \log n)$, are typically used when implementing block transform schemes The Joint Photographic Expert Group (JPEG) have adopted the linear block transform coding approach for its standard using the DCT as the transformation [16]

3 . Vector Quantization

The process of quantization maps a signal x(n) into a series of K discrete messages. For the kth message, there exists a pair of thresholds t_k and t_{k+1}, and an output value q_k such that $t_k < q_k \_ t_{k+1}$. For a given set of quantization values, the optimal thresholds are equidistant from the values. The concept of quantizing data can be extended from scalar or one-dimensional data to vector data of arbitrary dimension. Instead of output levels, vector quantization (VQ) employs a set of representation vectors (for the one-dimensional case) or matrices (for the two-dimensional case) [17], [18], [19], [20], [12]. The set is referred to as the "codebook" and the entries as "codeword". The thresholds are replaced by decision surfaces defined by a distance metric. Typically, Euclidean distance from the codeword is used. The advantage of vector quantization over scalar quantization is that the high degree of correlation between neighbouring pixels can be exploited. Even for a memory less system, the coding of vectors instead of scalars can theoretically improve performance.

In the coding phase, the image is subdivided into blocks, typically of a fixed size of n _ n pixels. For each block, the nearest codebook entry under the distance metric is found and the ordinal number of the entry is transmitted. On re-construction, the same codebook is used and a simple look-up operation is performed to produce the reconstructed image. The standard approach to calculate the codebook is by way of the Linde, Buzo and Gray (LBG) algorithm [17]. Initially, K codebook entries are set to random values. On each iteration, block in the input space is classified, based on its nearest codeword. Each codeword is then replaced by the mean of its resulting class. The iterations continue until a minimum acceptable error is achieved. This algorithm minimizes the mean squared error over the training set.

While the LBG algorithm converges to a local minimum, it is not guaranteed to reach the global minimum. In addition, the algorithm is very sensitive to the initial codebook. Furthermore, the algorithm is slow since it requires an exhaustive search through the entire codebook on each iteration.

With this brief review of conventional image compression techniques at hand, we are ready to consider the role of neural networks as an image compression tool.

III. NEURAL NETWORK

Artificial neural networks are relatively crude electronic networks of "neurons" based on the neural structure of the brain. They process records one at a time, and "learn" by comparing their classification of the record with the known actual classification of the record. The errors from the initial classification of the first record is fed back into the network, and used to modify the networks algorithm the second time around, and so on for many iterations.

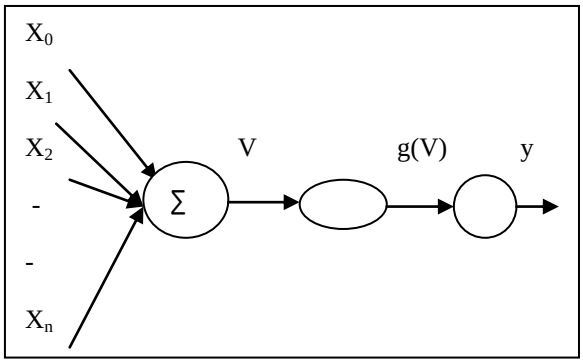

Figure 2: Neural Network

Roughly speaking, a neuron in an artificial neural network is :

1.A set of input values (xi) and associated weights (wi)

2.A function (g) that sums the weights and maps the results to an output

A neural network architecture as shown in **Figure 1** is used for solving the image compression problem. In this type of architecture, input from the large number of input neurons is fed to a less number of neurons in hidden layer, which is further fed to large number of neurons. The input layer is composed not of full neurons, but rather consists simply of the values in a data record, that constitutes inputs to the next layer of neurons. The next layer is called a hidden layer; there may be several hidden layers. The final layer is the output layer, where there is one node for each class. A single sweep forward through the network results in the assignment of a value to each output node, and the record is assigned to whichever class's node had the highest value.

1. Training an artificial neural network

In the training phase, the correct class for each record is known (this is termed supervised training), and the output nodes can therefore be assigned "correct" values -- "1" for the node corresponding to the correct class, and "0" for the others. (In practice it has been found better to use values of 0.9 and 0.1, respectively.) It is thus possible to compare the network's calculated values

for the output nodes to these "correct" values, and calculate an error term for each node (the "Delta" rule). These error terms are then used to adjust the weights in the hidden layers so that, hopefully, the next time around the output values will be closer to the "correct" values.

2. The Iterative learning process

A key feature of neural networks is an iterative learning process in which data cases (rows) are presented to the network one at a time, and the weights associated with the input values are adjusted each time. After all cases are presented, the process often starts over again. During this learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of input samples. Neural network learning is also referred to as "connectionist learning," due to connections between the units. Advantages of neural networks include their high tolerance to noisy data, as well as their ability to classify patterns on which they have not been trained. The most popular neural network algorithm is back-propagation algorithm proposed in the 1980's.

Once a network has been structured for a particular application, that network is ready to be trained. To start this process, the initial weights (described in the next section) are chosen randomly. Then the training, or learning, begins.

The network processes the records in the training data one at a time, using the weights and functions in the hidden layers, then compares the resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights for application to the next record to be processed. This process occurs over and over as the weights are continually tweaked. During the training of a network the same set of data is processed many times as the connection weights are continually refined.

Note that some networks never learn. This could be because the input data do not contain the specific information from which the desired output is derived. Networks also don't converge if there is not enough data to enable complete learning. Ideally, there should be enough data so that part of the data can be held back as a validation set.

3. Feed forward, Back-Propagation

Currently, this synergistically developed back-propagation architecture is the most popular, effective, and easy-to-learn model for complex, multi-layered networks. Its greatest strength is in non-linear solutions to ill-defined problems. The typical back-propagation network has an input layer, an output layer, and at least one hidden layer. There is no theoretical limit on the number of hidden layers but typically there are just one or two.

Some work has been done which indicates that a maximum of five layers (one input layer, three hidden layers and an output layer) are required to solve problems of any complexity. Each layer is fully connected to the succeeding layer. As noted above, the training process normally uses some variant of the Delta Rule, which starts with the calculated difference between the actual outputs and the desired outputs. Using this error, connection weights are increased in proportion to the error times a scaling factor for global accuracy. Doing this for an individual node means that the inputs, the output, and the desired output all have to be present at the same processing element. The complex part of this learning mechanism is for the system to determine which input contributed the most to an incorrect output and how does that element get changed to correct the error. An inactive node would not contribute to the error and would have no need to change its weights. To solve this problem, training inputs are applied to the input layer of the network, and desired outputs are compared at the output layer. During the learning process, a forward sweep is made through the network, and the output of each element is computed layer by layer. The difference between the output of the final layer and the desired output is back-propagated to the previous layer(s), usually modified by the derivative of the transfer function, and the connection weights are normally adjusted using the Delta Rule. This process proceeds for the previous layer(s) until the input layer is reached.

4. Structuring the network

The number of layers and the number of processing elements per layer are important decisions. These parameters to a feed forward, back-propagation topology are also the most ethereal - they are the "art" of the network designer. There is no quantifiable, best answer to the layout of the network for any particular application. There are only general rules picked up over time and followed by most researchers and engineers applying this architecture to their problems.

Rule One: As the complexity in the relationship between the input data and the desired output increases, the number of the processing elements in the hidden layer should also increase.

Rule Two: If the process being modelled is separable into multiple stages, then additional hidden layer(s) may be required. If the process is not separable into stages, then additional layers may simply enable memorization

of the training set, and not a true general solution effective with other data.

Rule Three: The amount of training data available sets an upper bound for the number of processing elements in the hidden layer(s). To calculate this upper bound.

IV. IMAGE COMPRESSION USING NEURAL NETWORK

Image compression is an important topic in the digital world, whether it can be commercial photography, industrial imagery, or video. A digital image bitmap can contain considerably large amounts of data causing exceptional overhead in both computational complexity as well as data processing. Compression is important to manage large amounts of data for network, Internet, or storage media.

Data compression itself is the process of reducing the amount of information into a smaller data set that can be used to represent, and reproduce the information. Types of image compression include loss less compression, and lossy compression techniques that are used to meet the needs of specific applications. JPEG compression can be used as a loss less or a lossy process depending on the requirements of the application both lossless and lossy compression techniques employ reduction of redundant data [5][6].

There are many coding technique which is used to neural network to compress the Image explained above. In Predictive coding technique optimal predictors based on a linear weighted sum of the neighbouring pixels are relatively easy to design using the statistics of the image. However, if a nonlinear model is more appropriate, the use of a linear predictor will clearly result in a suboptimal solution. While in Transform coding neural net work uses different algorithm. Linear PCA Generalized Hebbian Algorithm, Adaptive Principal Component Extraction and Robust Principal Components Estimation. In vector quantization uses self organizing map algorithm. The SOFM algorithm has a number of important properties which make it suitable for use as a codebook generator for vector quantization [5]:.

The Joint Photographic Experts Group produced the well-known image format JPEG, a widely used image format. JPEG provides solid baseline compression algorithm that can be modified numerous ways to fit any desired application. The JPEG specification was released initially in 1991, although it does not specify a particular implementation.

A three layered BPNN was designed for image compression based on the idea of encoder problem as shown in Fig.2.

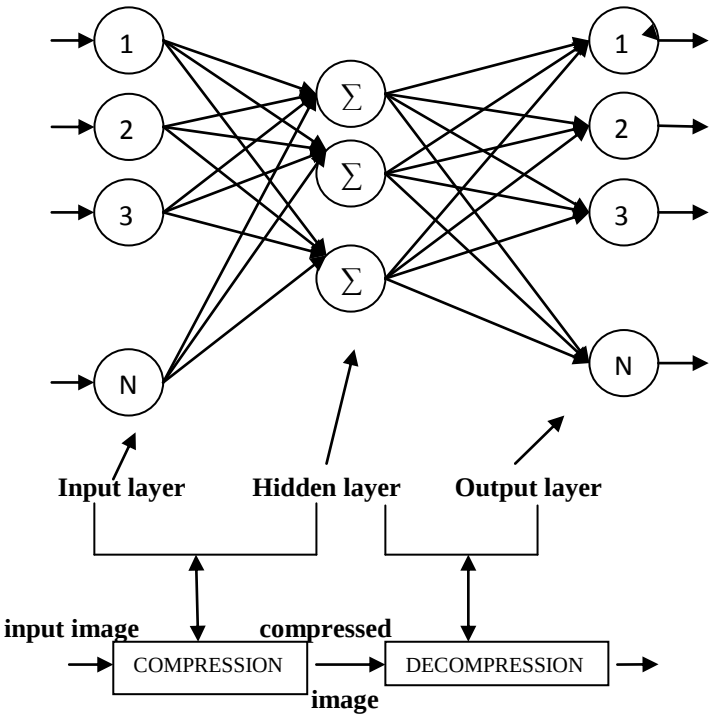

Figure 3: Neural Network Image Compression

This BPNN is fed by analog gray level of image as an input (values in range between 0 and 255) and produces an appropriate compressed code at outputs of hidden layer units. In the reconstruction process, this BPNN produce an analog gray level image by the outputs of output layer units. The input and output layers have Ni units each, and an intermediate layer with Nh units (Nh < Ni) that is the channel between the input and output layers.

In Image compression using artificial neural network, evaluated some parameter like peak-signal-to-noise ratio(PSNR) ,SNR, mean square error (MSE),CR .The quality of obtained image depend on these mathematical expression also. The compression-decompression error is evaluated by comparing the input image and decompressed image using normalized mean square error formula [7]:

$$E = (t - y_{in})^2$$

Where t is the target value and y_{in} is the input to the output.

Another approach to reduce the search time is to exploit. the organization of the codeword in the topologically ordered codebook. During the initial period of training in the SOFM algorithm, each input vector modifies a large number of

codeword due to the initial large neighbourhood function. Some researchers have proposed a training algorithm in which the neighbourhood may remain constant but the size of the network grows. In this approach, the size of the network doubles after a number of iterations which grows the network in a hierarchical fashion. The new codeword are inserted between the existing ones and initialized to the mean of their neighbours. This has the effect of drastically reducing the computational requirements during training while still producing a topologically ordered codebook. Luttrell [69] has successfully used this technique to generate VQ codebooks for SAR images.

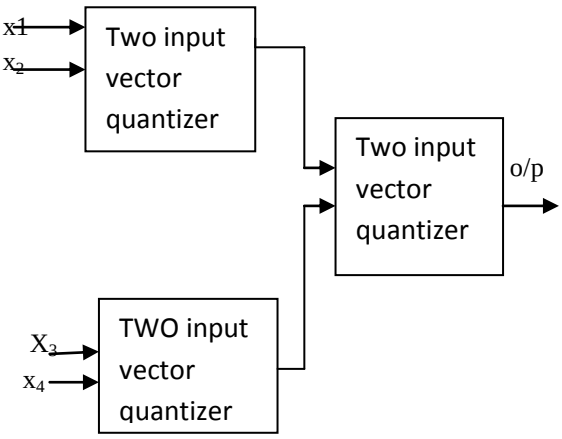

Figure 4: two stage hierarchal vector quantization using two input vector quantizer

PSNR is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. When comparing compression it is used as an approximation to human perception of reconstruction quality, therefore in some cases one reconstruction may appear to be closer to the original than another, even though it has a lower PSNR (a higher PSNR would normally indicate that the reconstruction is of higher quality). Mathematically PSNR is given by:

$$\text{PSNR} = 10\ 10\log_{10}\left(\frac{L^2}{MSE}\right)(db)$$

Where L is maximum value in pixel.

$$\text{Compressed image} = \frac{\text{Compressed image file size}}{\text{Uncompressed image file size}}$$

V. PROBLEM FORMULAION

While the scope of this paper has been to review the current research into applying neural network techniques to the task of image compression and summarize the various approaches, we have included, where possible, the claims as to the performance of the various techniques by their respective the distance to the codeword during training is weighted , there is still a need for an objective, unified evaluation of these new approaches. A common set of training images and common testing images from outside the training Such an evaluation must address the four issues of training, distortion, bit rate, and complexity.

1. Training

Because these techniques require training to calculate the parameters of the encoder, care must be taken in selecting the common set of training images. They must be representative of the class of images for which the network is to be used. For example, if natural images are to be processed, the training set should include a variety of naturally occurring scenes. On the other hand, if artificial images such as synthetic aperture radar (SAR) or magnetic resonance imaging (MRI) are used, then the training set must be representative of that class of images

2. Distortion

With lossy compression, the reconstructed image differs from the original. This difference may result in some visible distortion which may be measured in a number of ways.

a) Mean Squared Error (MSE):: A common measure of distortion is the mean squared error (MSE). Its wide-spread use is due mainly to its ease of calculation. Generally, an image with a high MSE has more visible distortion than that with a low MSE. In addition, the minimization of the MSE as an optimality criterion has formed the basis for the development of a large number of successful image compression algorithms.

b) Mean Opinion Score:: However, as many researchers have rightly pointed out, despite its popularity as a distortion measure, MSE can be a poor indicator of the subjective quality of reconstruction [14], [11]. As a result, perceptually based criteria may be more appropriate. One example is the *mean opinion*

score [11]. A number of subjects view an image and rate its quality on a five point scale of "bad," "poor," "fair," "good," or "excellent." The mean opinion score is simply the average rating assigned by all the subjects.

c) Weighted Mean Squared Error (WMSE):: An alternative to the MSE is the weighted mean squared error (WMSE). In this method, the error is weighted by a subjective function which measures the local visibility of distortion. The weighting function can be derived through subjective experiments or models of the human visual system (HVS). The visibility of error can depend on the modulation transfer function (MTF), (the spatial frequency response) of the HVS, the local contrast, and the local texture content [14].

d) End-use Measures:: If the end-use of a class of images is well defined, then the performance under the end-use can be measured. For example, if feature detection is performed on a remote sensing image, then receiver operating characteristics (ROC) curves can be employed [75]. ROC curves plot the relationship between the probability of detection (P_d) and the probability of false alarm (P_{fa}). As the distortion of the compression system increases, P_d will decrease for a given P_{fa}. For medical imaging, the effect of the distortion on the accuracy of diagnosis can be similarly measured [76].

3.Complexity

The complexity of a compression system is the computa-tional effort required for encoding and decoding images. A typical measure of complexity is the number of floating-point operations (flops) per pixel required to encode and decode an image. Associated with complexity is speed. Speed is a function of both complexity and implementation and may be substantially improved through the use of an efficient implementation of an image compression algorithm with a given computational complexity. For example, if a neural network were to be implemented in a parallel architecture, significant improvements in speed over a serial implementation would be realized due to the inherently parallel design of the neural network.

A) Encoding and Decoding:: In point-to-point transmission, an image is encoded once, transmitted, and then decoded once. In such an environment, the complexities of both the encoder and the decoder are equally important. However, this environment does not apply to many current uses of digital imaging. In a broadcast environment or a database retrieval environment, an image is encoded once and decoded *many* times. For these environments, it is the complexity of the decoder that is of greater importance

VI. CONCLUSION

In this paper, we have discussed various up-todate image compression neural networks which are classified into three different categories according to the nature of their applications and design. These include direct development of neural learning algorithms for image compression, neural network implementation of traditional image compression algorithms, and indirect applications of neural networks to assist with those existing image compression techniques.

With direct learning algorithm development, vector quantization neural networks and narrow channel neural networks stand out to be the most promising technique which have shown competitive performances and even improvements over those traditional algorithms. While conventional image compression technology is developed along the route of compacting information (transforms within different domains), then quantization and entropy coding, the principal components extraction based narrow-channel neural networks produce better approximation of extracting principal components from the input images, and VQ neural networks make a good replacement for quantization. Since vector quantization is included in many image compression algorithms such as those wavelets based variations, etc., many practical applications can be initiated in image processing and image coding related area. Theoretically, all the existing state-of-the-art image compression algorithms can be possibly implemented by extended neural network structures, such as wavelets, fractals and predictive coding described in this paper. One of the advantages of doing so is that implementation of various techniques can be standardized on dedicated hardware and architectures. Extensive evaluation and assessment for a wide range of different techniques and algorithms can be conveniently carried out on generalized

neural network structures. From this point of view,

image compression development can be made essentially as the design of learning algorithms for neural networks. People often get wrong impressions that neural networks are computing intensive and time consuming. The fact is the contrary.

For most neural networks discussed in the paper, the main bottleneck is the training or convergence of coupling weights. This stage, however, is only a preparation for the neural network. This will not affect the actual processing speed. In the case of vector quantization, for example, a set of pre-defined image samples is often used to train the neural network. This paper also include. At present, research in image compression neural networks is limited to the mode pioneered by conventional technology ,namely, information compacting(transforms)+quantization +entropy coding. Neural networks are only developed to target individual problems inside this mode [9,15]. In this way, every algorithm proposed can go through the same assessment with the same test data set. Further research can also be targeted to design neural networks capable of both information compacting. and quantizing. Hence the advantages of both techniques can be fully exploited. Therefore, future research work in image compression neural networks can be considered by designing more hidden layers to allow the neural networks go through more interactive training and sophisticated learning procedures. Accordingly, high performance compression algorithms may be developed and implemented in those neural networks. Within the infrastructure, dynamic connections of various neurons and non-linear transfer functions can also be considered and explored to improve their learning performances for those image patterns with drastically changed statistics.

References

[1] Han Feng, Man Tang, Jie Qi "A Back-Propagation Neural Network Based on a Hybrid Genetic Algorithm and Particle Swarm Optimization for Image Compression,"4th International Congress on Image and Signal Processing, 978-1-4244-9306-7/11 IEEE 2011.

[2] Jianji Wang, Nanning Zheng, "Novel Fractal Image compression scheme with block classification and storing Based on Pearson's Correlation Coefficient," IEEE Transactions on Image Processing , VOL. 22,no.9 September 2013

[3] W. C. Black and D Robert D. Dony, Neural Network Approaches to Image Compression, Proceedings of the IEEE, Vol.83, No.2, Feb1995.

[4] Dony.R.D. and Haykin.S., "Neural network approaches to image compression," Procceedings of the IEEE, 1995, vol. 83 ,pp. 288–303.

[5] Marin T Hagan. On Design of The Neural Network. Beijing: Michanical Industry Publishing House,2002

[6] R. C. Eberhart and J. Kennedy, "A new optimizer using particles swarm theory," in Proc. of Sixth Int.Symp. on Micro Machine and Human Science. Nagoya,Japan, 1995, pp. 39-43.

[7] O.Abdel-Wahhab and M.M.Fahmy, "Image compression using multi-layer neural networks," in Proc. of Vision, Image and Signal Processing. Vol. 144. No.5, October 1997

[8] M. F. Barnsley and A. E. Jacquin, "Application of recurrent iterated function systems to images," *Proc. SPIE*, vol. 1001, pp. 122–131, Nov. 1988.

[9] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Process.*, vol. 1, no. 1, pp. 18–30, Jan. 1992.

[10] M. Pi, M. K. Mandal, and A. Basu, "Image retrieval based on his-togram of fractal parameters," *IEEE Trans. Multimedia*, vol. 7, no. 4, pp. 597–605, Aug. 2005.

[11] J. H. Jeng, C. C. Tseng, and J. G. Hsieh, "Study on Huber fractal image compression," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 995–1003, May 2009.

[12] M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal image denoising," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1560–1578, Dec. 2003.

[13] M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal-wavelet image denoising revisited," *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2669–

[14] S.S.Wang and S. L. Tsai, "Automatic image authentication and recovery using fractal code embedding and image inpainting," *Pattern Recognit.*, vol. 41, no. 2, pp. 701–712, 2008.

[15] Hamdy S. Soliman, Mohammed Omari, "A Neural Networks Approach to Image Data Compression",

Applied Soft Computing,

