A Modified Fuzzy Learning Vector Quantization Algorithm for Image Compression

N.Krishnan, A. Dyana and R.K.Selvakumar
Centre for Information Technology and Engineering, Manonmaniam Sundaranar University,
Tirunelveli-627012, INDIA. email:krishnan17563@yahoo.com

Abstract *This paper presents an efficient method to compress images using vector quantization. The limitations faced by the Fuzzy Algorithms for Learning Vector Quantization (FALVQ) and various other clustering algorithms in lossy image compression system are the time complexity and the blocking artifacts in the reconstructed image. The proposed algorithm has eliminated these limitations by a good initialization of codebook in FALVQ and also improves the performance over various other methods. The initialization of codebooks is performed based on the Euclidean distance of the input vectors from a reference vector thus providing knowledge about the input vectors during the initialization itself. Experiments were carried out using the test image Lena and we found that the proposed algorithm outperforms the standard FALVQ and the FALVQ combined with wavelet.*

***Keywords*---Learning vector quantization, Fuzzy Algorithms for LVQ, Clustering algorithms, codebook initialization, Blocking artifacts, wavelet.**

I. INTRODUCTION

Image compression reduces the amount of data required to represent a digital image thereby addresses the problem of minimizing the storage and transmission of images. Vector quantization remains as an important processing stage in lossy image compression since several years. Shannon's theory has proved that vector quantization is better than scalar quantization.

In Vector Quantization, the input data are clubbed together in groups called vectors, and processed to give the output. This clubbing of data and treating them as a single unit, increases the optimality of the vector quantizer, but at the cost of increased computational complexity. VQ can also be seen as a mapping from an n-dimensional Euclidean space into a finite set of prototypes. In clustering algorithms, compression is achieved by forming vectors from a training sequence, grouping similar vectors into clusters, and assigning each cluster with a single representative vector. Incoming data can then be compressed by replacing vectors with the nearest cluster representative referenced by a simple cluster index. In VQ terminology, the clusters are usually referred to as cells and their representative as codevectors. The list of all cluster representatives forms a codebook.

The most prevalent technique for codebook design is the generalized Lloyd algorithm (GLA) or c-means algorithm which was further modified [7]. GLA is also referred to as LBG algorithm [9]. Kohonen initiated the study of prototype generation algorithm called Learning Vector Quantization (LVQ) and he also introduced the concept of Self Organizing Feature Maps (SOFM) in 1989. Bezdek et al proposed a batch learning scheme, known as Fuzzy Learning Vector Quantization (FALVQ) in 1994[4]. Modifications on learning vector quantization also developed [5] [6]. In 1996, Karayiannis and Pin.I.Pai presented a general formulation the LVQ problem and proposed a framework for the development of a broad variety of FALVQ algorithms [2] and also combined the FALVQ with wavelet [3].

The main drawbacks of the LVQ and other clustering methods such as c-means are Initialization of codebook, Time Complexity, Blocking artifacts This paper aims at overcoming the above mentioned problems of clustering techniques and testing the algorithm in Lossy Image compression system based on Fuzzy algorithms for Learning Vector Quantization.

II. FUZZY LEARNING VECTOR QUANTIZATION

In Learning Vector Quantization, clusters sub structure hidden in the unlabeled p-dimensional data is discovered. LVQ can be performed through an unsupervised learning process using a competitive neural network whose weight vectors represent the prototype. Let X={x1, x2…xn} subset of R^p denote the samples at hand and the use c to denote the number of nodes (and clusters in X), in the competitive layer. The salient features of LVQ model are contained in Figure 1.

The network consists of an input layer and an output layer. Each node in the input layer is connected directly to the cells in the output layer. A prototype vector is associated with each cell in the output layer .The input layer of an LVQ network is connected directly to the output layer. Each node in the output layer has a weight factor (or prototype) attached to it. The prototype V= {v1,v2,………vc} are essentially a network array of (unknown) cluster centers $y_i \in R^p$ for each cluster i . In this context the word learning refers to finding values for the $\{v_{ij}\}$. When an input vector z is submitted to this network, distances are computed between each v_r and z. The output nodes compete a (minimum distance) winner node, say v_i is found; and it is then updated using one of the several update rules. Consider the set of samples x $\in X$ from an n-dimensional Euclidean space and let f(x) be the probability density function of $x \in R^n$. LVQ is frequently based on the minimization of the functional

$$L(v_r, r=1,2,..c) = \iiint_{R^n} \sum_{r=1}^{c} \| x - v_r \|^2 f(x)dx \qquad (1)$$

which represents the expectation of the loss function

$$L_x = L_x(v_r, r = 1,2....c)$$
$$= \sum_{r=1}^{c} u_r \| x - v_r \|^2 \qquad (2)$$

$u_r = u_r(x)$, r = 1,2,…c is a set of weights. A fuzzy algorithm for LVQ (FALVQ) was proposed by Karayiannis [1]. FALVQ was developed by interpreting the weights $u_r = u_r(x)$, r=1,2,….c as membership functions which regulate

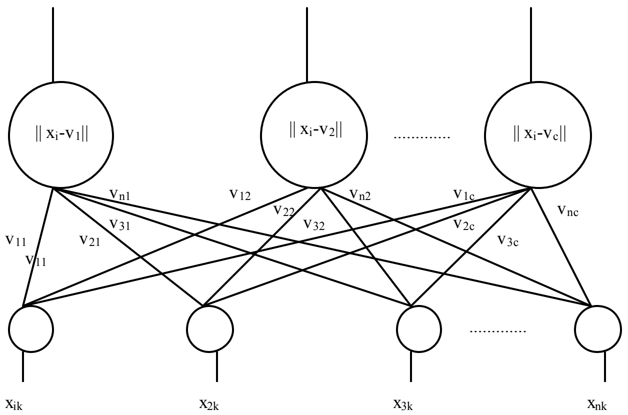

Figure 1. The LVQ Competitive network

the competition between the prototypes, v_r, r = 1,2,……c for the input x. The specific form of membership functions determine the strength of attraction between each input and the prototypes during the learning process. Assuming that v_i is the winning prototype corresponding to the input vector x, that is the closest prototype to x in the Euclidean distance sense, the membership u_r, r = 1,2….c can be of the form

$$u_r = u_{ir} = \begin{cases} 1, & if \quad r = i \\ u\left(\|x - v_i\|^2 \big/ \|x - v_r\|^2 \right), & if \quad r \neq i \end{cases} \qquad (3)$$

If x is the input vector, the winning prototype v_i can be updated by

$$\Delta v_i = \eta(x - v_i)(1 + \sum_{r \neq i}^{c} w_{ir}) \qquad (4)$$

where

$$w_{ir} = u'\left(\| x - v_i \|^2 \big/ \| x - v_r \|^2 \right) \qquad (5)$$

Each non winning prototype can be updated by

$$\Delta v_i = \eta(x - v_i) n_{ij} \qquad (6)$$

where

$$n_{ij} = u_{ij} - \left(\| x - v_i \|^2 \big/ \| x - v_j \|^2 \right) w_{ij} \qquad (7)$$

The FALVQ algorithms [1] can be summarized as follows

Step 1: Select c; Fix h_0; N; set it = 0;
Randomly generate an initial codebook
$v_0 = \{ v_{1,0}, v_{2,0}, \ldots\ldots\ldots\ldots v_{c,0} \}$

Step 2: Calculate $h = h_0(1 - it/N)$;

Step 3: Set it = it+1;

Step 4: For each input vector x

a) Find i such that $\| x - v_{i,it-1} \|^2 < \| x - v_{j,it-1} \|^2 \ \forall j \neq i$

b) Calculate $u_{ir,it}$ by (2)

c) Calculate $w_{ir,it}$ by (5) where $r \neq i$

d) Calculate

$$n_{ir,it} = u_{ir,it} - \left(\|x - v_i\|^2 \big/ \|x - v_j\|^2 \right) w_{ir,it} \quad , \forall r \neq i \qquad (8)$$

(e) Update v_{it} by

$$v_{i,it} = v_{i,it-1} + \eta(x - v_{i,it-1})\left(1 + \sum_{r \neq i} w_{ir,it}\right) \qquad (9)$$

(f) Update v_j not equal to v_i by

$$v_{j,it} = v_{j,it-1} + \eta(x - v_{i,it-1}) n_{ij,it} \qquad (10)$$

Step 5: If it < N, the go to step 2

Three families of FALVQ were developed by selecting the form of the interference function that determines the effect of the nonwinning prototypes on the attraction between the winning prototype and the input of the network.

FALVQ with wavelet decomposition was developed [3]. Wavelet based subband decomposition of an image can be interpreted as an image filtering process. The codebook is designed for each subband using FALVQ algorithm except the upper left subband at resolution level 2 which contains background information.

III. THE PROPOSED CODEBOOK INITIALIZATION

Initialization of codebook strongly affects the clustering procedure. Various techniques proposed thus far are

1. Selection of the first M vectors from the training set
2. Selection of every N/Mth vector from the training set.
3. Random selection of M vectors from the training set
4. Splitting technique with orderly preference
5. Splitting technique with preference based on cell population
6, Splitting technique with preference based on cell distortion
7, Merging technique (similar to the pairwise nearest neighbour method)
8, Using uniform quantizer

When different data is used for initialization, different results are produced. When sufficient effort is taken to initialize the codebook rather than modifying the clustering procedures, better results are achieved. The proposed algorithm uses the clustering procedure FALVQ discussed in section II. Initialization of codebook is done with respect to the training set. And thus FALVQ's task is made easier to train the incoming training vectors. The method is as follows.

For each input vector, the Euclidean distance of the vector from the origin is found out. If the dimension of the vector is k, ad x is the input vector, then $\| x - z \|^2$ is the Euclidean distance where $x = [x_1\ x_2\ x_3 \ldots\ldots\ldots x_k]$ and $z = [0_1\ 0_2\ 0_3 \ldots\ldots\ldots 0_k]$. A set of n distances is obtained where n is the number of training data. The input vectors are sorted based on their distances in ascending or descending order. Now the order of arrangement of vectors is such that their distances from origin increases as their location index increases, if sorted in ascending order. The sorted data is divided into c parts where c is the number of clusters or the size of codebook. So, every part contains n/c vectors. For every part, the mean or centroid is calculated by the formula

$$\hat{y}_{id} = \frac{\sum_{i=1}^{n} u_{ij} x_{id}}{\sum_{i=1}^{n} u_{ij}} \tag{11}$$

$\forall$ j=1,2,…….c ; j → no.of parts
$\forall$ d=1,2,…….k ; d→ dimensions

$$u_{ij} = \begin{cases} 1, if & x_i \in P_j \\ 0, & otherwise \end{cases}$$

where P_j represents the j^{th} cluster

Thus calculated mean vector is the representative of the divided parts, in other words clusters. This constitutes the initial codebook.

The algorithm is explained as follows

Step 1: For every input vector of k dimension calculate Euclidean distance from the origin.
Step 2: Sort the input vectors based on their distances
Step 3: Divide the sorted vectors into c clusters taking every n/c vectors where n is the total number of vectors.
Step 4: For every cluster, find the mean or centroid
Step 5: Thus calculated mean vectors form the codebook.

IV. THE PROPOSED SYSTEM AND ITS FEATURES

The following steps summarize the proposed image compression system. The input image is divided into n*n blocks and this is fed into the proposed initialization algorithm. This constructs a initial codebook.

The training vectors formed by taking n*n blocks are given to Fuzzy Learning Vector Quantization Algorithm which trains network using the initialized codebook. Finally it generates a codebook with desired number of codevectors.

The input image is then coded with the codebook. Each n*n block is replaced by the index of nearest codevector in the codebook. The compressed image when reconstructed, each entry in the compressed image serves as the index of codebook and then replaced by the corresponding codevector of the index. This is a simple table-look up procedure.

The clustering algorithms like fcm and FALVQ shows blocking effects due to the improper choice of prototypes.

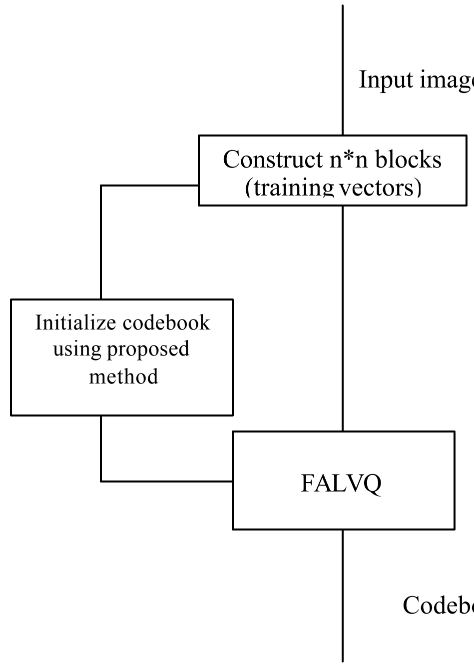

Figure 2. Codebook design in the proposed system

When the same clustering algorithm FALVQ when initialized with the proposed method discussed in the previous session, the blockiness is removed. This shows that the clustering reaches optimal solution when proper initialization is made. When random initialization is made, it reaches only local optimum solution. Since the sorted input vectors are divided into clusters with equal number of vectors in each cluster, the result is less than optimal. This codebook when refined further with the Fuzzy Algorithms for Learning Vector Quantization provides a optimal result.

Time complexity is also reduced. The time taken by the FALVQ to learn from the input is drastically reduced by giving a proper choice of initial codebook. Time is reduced about 50 percent and even more by the proposed method and the distortion is also lesser for the same compression ratio.

Considering c-means algorithm, the classification is made depending upon the nearest neighbour condition that is taking Euclidean distance and making new partition based on the distance. This takes number of iterations to converge if initialized with random code. The proposed algorithm initializes with vectors that are reached by fcm after number of iterations.

Considering wavelet combined with clustering algorithms, [8] Wavelet decomposes the image into various bands in different resolutions. Each decomposed band is vector quantized using any clustering algorithm. The blocking effect is removed due to the feature of wavelet and not due to the clustering algorithms. So the clustering algorithms, has yet to be refined so that it makes a good clustering. The proposed method has the essence of the features of wavelets when used with image compression especially at high bit rates.

Multiresolution codebooks are designed when wavelet was used with FALVQ which takes higher compression rate to yield a low distortion. Without any multiresolution codebooks, and thereby without wavelets, the features of wavelet decomposition in image compression with FALVQ are captured by the proposed algorithm. Since the approximated subband in the upper left corner is not quantized in the FALVQ with wavelet, only the smoothing effect is maintained. The details are lost to a considerable amount. But the proposed algorithm does not suffer from these types of effects.

V. EXPERIMENTAL RESULTS

The Performance was calculated using the Peak Signal to Noise Ratio which is defined by the following formula.

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right) \qquad (12)$$

Where MSE represents Mean Square Error given by

$$\mathrm{MSE} = \frac{1}{(k*k)}\left(\sum_{i=0}^{k-1}\sum_{i=0}^{k-1}\| x_{i,j} - y_{i,j}\|^2\right) \qquad (13)$$

where k*k is the dimension of the input image, x represents the original image and y represents the reconstructed image.

The compression rate is calculated by the number of bits required to represent a pixel. If c is the size of codebook or the number of clusters, and s *s is the size of block then

$$\textit{Compression} \quad \textit{rate} = \frac{\log_2 c}{s*s}\textit{bits / pixel}$$

The experiments were carried out using the software Matlab. The Lena image was used as the test image. The Lena image was divided into n*n blocks and given to the system. The following settings were used for the entire experiment unless otherwise mentioned.

Learning Rate	:0.001
FALVQ family	:1
Parameter for FALVQ1	:1

The first set of experiments was conducted between Fuzzy c-means, FALVQ and the proposed algorithms for different compression rates as shown in Table 2.

Table 1. Comparison between Fcm, FALVQ, proposed

Compression Rate (b/p)	PSNR Fcm	PSNR (FAL VQ)	PSNR (Proposed)
.125	20.35	23.09	25.44
.5	22.98	25	27.23
.5625	23.02	27.6	28.31

The same procedure was tried out for color images, processing the red, green and blue components separately. The following table shows the result for Test image Lena in color.

Table 2. Comparison for color image Lena

Compression Rate (b/p)	PSNR (FAL VQ)	PSNR (Proposed)
.5	40.11	42.61
0.125	37.71	40.88

The next experiment was conducted using wavelet transform. First the image was decomposed using wavelet. Multiresolution codebook was designed using FALVQ and the proposed procedure. In this case also, the proposed method outperforms the FALVQ. The following settings were used to carry out the experiment.

No. of iterations	: 50
Size of codebook	: 256
Size of block	: 4*4

Test Image:	Lena	Bird
PSNR using FALVQ with wavelet	: 25.06	31.54
PSNR using proposed with wavelet	: 25.72	32.09

Convergence characteristics for FALVQ and proposed was depicted by the following graph (Figure 3) for every 10 iterations

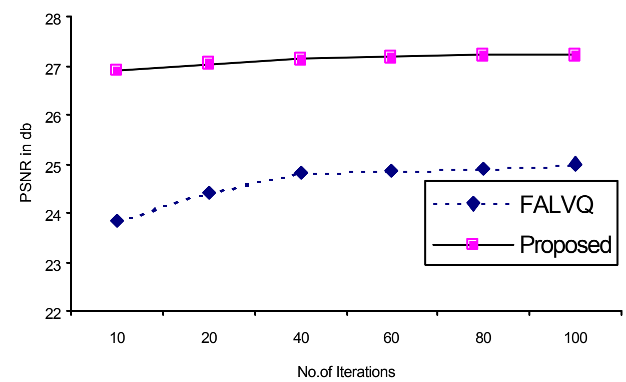

Figure 3. Convergence characteristics of FALVQ and proposed

This shows that the proposed algorithm has better performance from the initial stage itself and continues to outperform the FALVQ with random initialization as the number of iteration increases. Even after 100 iterations the FALVQ with random initialization could not cope up with the FALVQ with proposed algorithm.

(a)

(b)

(c)

(d)

(e)

Figure 4. (a) Original Lena image, (b) Reconstructed image with FALVQ combined with wavelet, (c) Reconstructed image with the proposed system, (d) Reconstructed image with FAVLVQ(random code initialization), (e) Reconstructed image with proposed system.

The next experiment illustrates the visual difference between FALVQ with wavelet and FALVQ with the proposed method.

Figure 4 (a) shows the original Lena image and figure 4(b) and 4(c) shows the reconstructed images with FALVQ with wavelet and FALVQ with proposed method respectively. The settings are as follows

No. of Iterations : 50
No. of Clusters : 256

Figure 4(d) and Figure 4(e) gives the visual difference between FALVQ with random initialization and proposed respectively. The following settings are used

No. of iterations: 100
No. of clusters: 512

Analyzing the first set of experiments the proposed algorithm outperforms FALVQ with random initialization. Not only in the distortion measure it performs well, but also the blocking effect is removed in this experiment and also in subsequent experiments. This also shows the outperforming nature of FALVQ when compared to Fuzzy c -means. This depicts the efficiency of the neural network architecture when compared to the centroid method. When tried with color a component, the same performance is achieved as like for monochrome images.

Considering compression with wavelet the proposed algorithm with wavelet outperforms FALVQ with wavelet, but by a small difference. This depicts the feature of wavelets. Wavelet naturally removes the block artifacts and gives high performance.

So, from these results, the proposed algorithm has gained the features of wavelet even without using them. So in image compression based on vector quantization, the wavelet decomposition can be bypassed to achieve the same result and even better result using clustering algorithms.

In Figure 4(b), though the reconstructed image using wavelet appears to be smoothened, when compared with original image, the reconstructed image with proposed (figure 4 (c)) has greater detail than the other one. This can be noticed in the hat and the hair of the lena image, while in figure 4(b) these detail have been smoothened. Since the bands other than the first band (approximation) image has been vector quantized, the details of the image is not clear. Though the blocking effect has been removed, due to the bad clustering of the subbands that is now the blocking effect has affected the knowledge of the detail which were represented by the subbands.

VI. CONCLUSION

The clustering algorithms when used for image compression with vector quantization, the initialization of codebook remained a problem. There was time complexity and blocking effects. A solution or remedy to the problem of initialization in clustering algorithms has been provided in this paper.

Experiment has been carried out with fuzzy algorithms for learning vector quantization with a different approach in the initialization of codebooks. Experimental results show superior performance when compared with other methods. The blocking effects in the reconstructed image have been removed. And also the time complexity has been reduced. An analysis of the result has been presented.

The initialization technique proposed in this paper can be extended to c-means algorithm. A slight modification of the proposed algorithm could be used to initialize fuzzy c-means by initializing the membership values for the input training set with respect to each cluster based on the statistics of the input training set. And this could be compared with the standard LBG algorithm.

REFERENCES

[1] Karayiannis, N.B., and Pin-Pai, J., “Fuzzy algorithms for Learning Vector Quantization”, IEEE Transactions on Neural Networks, Vol.7, No.5, Page 1196-1211, Sept. 1996.

[2] Karayiannis, N.B., “A Methodology for constructing fuzzy algorithm for Learning Vector Quantization”, IEEE Transactions on Neutral Networks, Vol.8, No.3, Page 505- 518, May 1997.

[3] Karayiannis, N.B., Pai, P.I., and Zervos, N., “ Image compression based on Fuzzy Algorithms for learning Vector Quantization and wavelet Image Decomposition”, IEEE Transactions on Image Processing, Vol.1, No.8, Page 1223, Aug.1998.

[4] Pal,N.R., Bezdek, J.C., and Tsao, E.C.K.,“Generalized clustering networks Kohonen’s Self Organizing Scheme”, IEEE Transactions on Neutral networks, Vol.4, No.4, Page 549-557, July 1994.

[5] Karayiannis N.B., “An Axiomatic Approach to Soft Learning Vector Quantization and Clustering”, IEEE Transactions on Neural Networks, Vol. 10, No. 5, September 1999, Page No 1153-1165

[6] Karayiannis N.B., “Soft Learning Vector Quantization and Clustering Algorithms Based on Ordered Weighted Aggregation Operators”, IEEE Transactions on Neural Networks, Vol. 11, No. 5, September 2000,Page No. 1093-1105.

[7] Kolen, J.F., Hutcheson, T., “Reducing the Time Complexity of the Fuzzy C- Means algorithm”, IEEE Transactions on Fuzzy systems, Vol.10, No.2, April 2002.

[8] Wang J.H., Gou K.J., “Image compression using wavelet transform and Self development Neural networks”, IEEE 1998.

[9] Yoseph, L., Andress, B., and Gray, R.M., “An algorithm for Vector Quantizer Design”, IEEE Transactions on Communications, Vol com-28, No.1, Page 84-9, Jan.1980.