International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011
ISSN: 1793-8201

Image Medical Compression by A new Architecture Optimization Model for the Kohonen Networks

M. Ettaouil, Y. Ghanou, K. El Moutaouakil and M. Lazaar

Abstract— **This paper presents a novel lossy compression scheme for medical images by a new architecture Optimization model for the Self-Organized Map (OSOM). Both neural networks for lossy compression scheme are comparatively examined: Kohonen map and OSOM. This new approach based on genetic algorithms to determine the optimal parameters of neural networks. In the compression process of the proposed method, the medical image is decomposed into blocks of 4×4 pixels. The numerical results assess the effectiveness of the theoretical results shown in this paper, and the advantages of the new modeling.**

Index Terms — **Image medical compression, Vector Quantization, Codebook, Self-Organized Map, Genetic algorithms.**

I. INTRODUCTION

UNcompressed multimedia data require considerable storage capacity and transmission bandwidth. Thus, image compression is a very important factor for better utilization of network bandwidth and computer storage. The compression process is usually lossy and is based on redundancy and irrelevancy reduction, which are inherent in the image domain. In the medical field, using radiographic images, ultrasound, MRI (Magnetic Resonance Imaging), ... poses a great problem of storage and archiving. To overcome these problems, compression of these images is an operation necessary and imperative. The main purpose of image compression is to reduce the amount of bits needed to describe them while keeping an acceptable visual appearance of the reconstructed images. The compression of images has been performed by several techniques among the best known: the JPEG is a lossy method standardized by ISO in August 1990, these methods perform compression by performing a scalar quantization (SQ) on the values obtained after processing. The disadvantage of the scalar quantization is that it does not exploit the spatial correlation between different pixels of the image. Another more interesting way to achieve compression coding is not the values individually one after the other, but to encode a set of values simultaneously. This procedure is called vector quantization (VQ) [12], the VQ has been successfully used for encoding the voice signal and for compressing still images. Approaches using artificial neural networks for intelligent processing of data seem to be very promising, this is mainly due to their structures, providing opportunities for parallel computation [8], [9] and the use of the learning process allows the network to adapt the data to be processed. New techniques based on neural networks as compression tools have been proposed by Jiang [14], Robert [18] and Stanley and al [19] . The Kohonen network is a particular neural network; it can be used as a vector quantizer for images. The results obtained by the Kohonen algorithm are acceptable, but the quality of compression depends on the optimal parameters of the Kohonen networks. This article discusses a new method of compression of medical images using a new model for optimizing the architectures of the Kohonen networks and genetic algorithms.

This paper is organized as follows: In section II the vector quantization by Kohonen networks is described. In section III, a new model for optimizing the architectures of the Kohonen networks is presented. Experimental study and discussion of the results are presented in the last section.

II. VECTOR QUANTIZATION BY KOHONEN NETWORKS

Vector Quantization has been observed as an efficient technique for image compression [11]. VQ compression system contains two components: VQ encoder and decoder. The principle of the VQ techniques is simple. At first, the image is splitted into square blocks $X = \{x^1, x^2, x^3, ..., x^n\}$ of $\tau \times \tau$ pixels, for example 4×4 ; each block is considered as a vector in a 16-dimensional space, respectively, a clustering algorithm, for example LBG [16], is used to generate a codebook $C = \{Y_1, Y_2, Y_3, ..., Y_N\}$ for the given set of image blocks. Second, a limited number (N) vectors (codewords) in this space is selected in order to approximate as much as possible the distribution of the initial vectors extracted from the image; in other words, more codewords will be placed in the region of the space where there are more points in the initial distribution (image), and vice-versa. Third, each vector from the original image is replaced by the nearest codeword (usually according to a second-order distance measure). Finally, in a transmission scheme, the index of the codeword is transmitted instead of the codeword it self.

Many authors used the Kohonen's algorithm [17] or Self-Organized feature Map (KSOM) [3] to achieve the vector quantization process of image compression. Kohonen's algorithm is a reliable and efficient way to achieve VQ, and has shown to be usually faster than other algorithm and to avoid the problem of "dead units" that can arise for example with the LBG algorithm.

Kohonen's algorithm has however another important property

Mohamed Ettaouil is with the Faculty of Science and Technology of Fez , University Sidi Mohammed ben Abdellah, City Fez, MOROCCO, email: mohamedettaouil@yahoo.fr

Youssef Ghranou is with the Faculty of Science and Technology of Fez , University Sidi Mohammed ben Abdellah, City Fez, MOROCCO, email: youssefghanou@yahoo.fr

Karim El Moutaouakil is with the Faculty of Science and Technology of Fez , University Sidi Mohammed ben Abdellah, City Fez, MOROCCO, email: Karimmoutaouakil@yahoo.fr

Mohamed Lazaar is with the Faculty of Science and Technology of Fez , University Sidi Mohammed ben Abdellah, City Fez, MOROCCO, email: mohamedlazaar1@yahoo.fr

besides vector quantization: it realizes a mapping between an input and an output space that preserves topology; in other words, if vectors are near from each other in the input space, their projection in the output space will be close too.
In the proposed compression scheme, we will use a two dimensional Kohonen map corresponding to a grid of codewords (instead of a one-dimensional table in standard VQ), as the projection of an initial space including all vectors coming from blocks of the initial image.

SOM training

The SOM consists of a regular, usually two-dimentional (2-D), grid of map units. Each unit is represented by a prototype vector $w^i = (w^{i,0}, w^{i,1}, w^{i,2}, ..., w^{i,d-1})$, where d is input vector dimension. The units are connected to adjacent ones by neighborhood relation. The number of map units, which typically varies from a few dozen up to several thousand, determines the accuracy and generalization capability of the SOM. During training, the SOM forms an elastic net that folds onto the "cloud" formed by the input data. Data points lying near each other in the input space are mapped onto nearby map units. Thus, the SOM can be interpreted as a topology preserving mapping from input space onto the 2-D grid of map units (Fig. 1).

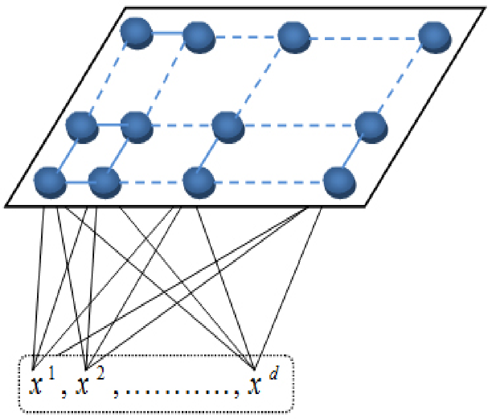

Fig. 1. Graphical presentation of a SOM

The SOM is trained iteratively. At each training step, a sample vector x is randomly chosen from the input data set. Distances between x and all the prototype vectors are computed. The best-matching unit (BMU), which is denoted here by g, is the map unit with prototype closest to x.

$$g(t) = arg \min_{i}^{N} \|x(t) - w^i\| \quad (1)$$

N is the number of map units. Next, the prototype vectors are updated. The BMU and its topological neighbors are moved closer to the input vector in the input space. The update rule for the weight vector, for all $i \in V_g(t)$, is :

$$w^i(t+1) = w^i(t) + \alpha(t)\beta_{g,i}(t)\|x - w^i(t)\| \quad (2)$$

where

- t : time;
- $\alpha(t)$: adaptation coefficient;
- $h_{g,i}$: neighborhood kernel centered on the winner unit:

$$h_{g,i}(t) = \exp(-\frac{\|r_g - r_i\|}{2\sigma^2(t)}) \quad (3)$$

- $V_g(t)$: specifies the neighborhood of the winner neuron in the map units.

where r_g and r_i are positions of neurons g and i on the SOM grid. Both $\alpha(t)$ and $\sigma(t)$ decrease monotonically with time. There is also a batch version of the algorithm where the adaptation coefficient is not used [15].
In the case of a discrete data set and fixed neighborhood kernel, the absolute error function E which is a quantitative criterion of the quality of our data modeling. This function distance between each stimulus and the nearest neuron. It is defined as follows:

$$E = \sum_{q=1}^{n} \min_{i \in \{1,..,N\}} \|x^q - w^i\| \quad (4)$$

where n is the number of stimuli, N the number of neurons, the neuron index i and the stimulus q .

Kohonen algorithm

1) Initialize the weights W with small values;
2) Fixed t_{max} and V_0;
3) $t = 1$;
4) repeat until $t \leq t_{max}$;
   4-1 Select a random entry $x(t)$ from the training set;
   4-2 Determine the winner neuron $g(t)$ using equation(1);
   4-3 Update the weights using equation (2);
   4-4 $t = t + 1$.

III. A NEW MODELING OF NEURAL ARCHITECTURE OPTIMIZATION

The choose of the initials weights and the neurons number has an impact into the convergence of the training methods and the stabilization of the network after the training stage. In this section, we propose a new modelization of neural architecture optimization problem of self maps as an optimization problem with a polynomial mixed objective and the linear constraints. For modeling, the problem of neural architecture optimization, we have needed to define some parameters as follows:

Notations

- n : Observations number of data base;
- N : Optimal number of artificial neurons in the topology map of Kohonen;
- N_{min}: Minimal number of artificial neurons in the topology map of Kohonen;
- N_{max}: Maximal number of artificial neurons in the topology map of Kohonen;
- $X = \{x^1, ..., x^n\}$: Training base where $x^k = (x^k_1, ..., x^k_d)$ for $k = 1, ..., n$;
- $u_{i,j}$: The binary variable for i = 1,...,n and j = 1,...,N, $u_{i,j} = 1$ if the i^{th} example is assigned to j^{th} neuron, and $u_{i,j} = 0$ else;

- v_j : The binary variable for $j = 1, ..., N_{max}$, v_j takes 0 if the j^{th} neuron is deleted, otherwise $v_j = 1$ if j^{th} neuron is used.

Objective function

The objective function of the mathematical programming model is the summation of the distances between the inputs vector and the correspond weights for the neurons used:

$$E(U, V, W) = \sum_{i=1}^{n} \sum_{j=1}^{N_{max}} v_j u_{i,j} \|x^i - w^j\|^2 \qquad (5)$$

Where $U = (u_{i,j})_{i,j}, \quad V = (v_j)_j, \quad W = (w^j)_j$.

If $u_{i,j} = 1$ then the i^{th} example x^i is assigned to the j^{th} neuron, and the corresponding error $\|x^i - w^j\|$ has been calculated on the objective function E.

Assignment constraints

The following constraints guarantee the assignment for each example to one neuron:

$$\sum_{j=1}^{N_{max}} u_{i,j} = 1 \qquad for\ i = 1, ..., n \qquad (6)$$

Equilibrium constraint

However, in order to guarantee that the auxiliary variable $u_{i,j}$ takes the value 1 when the neuron j has been used, another quadratic constraint must be added:

$$\sum_{j=1}^{N_{max}} (1 - v_j) \sum_{1=1}^{n} u_{i,j} = 0 \qquad (7)$$

Optimization Model

We can eliminate some neurons from the map without loss the quality of this map, we can use the variables (v_j) and we introduce the following optimization model (P):

$$(P) \begin{cases} min \sum_{i=1}^{n} \sum_{j=1}^{N_{max}} v_j u_{i,j} \|x^i - w^j\|^2 \\ \sum_{j=1}^{N_{max}} u_{i,j} = 1 \\ \sum_{j=1}^{N_{max}} (1 - v_j) \sum_{i=1}^{n} u_{i,j} = 0 \\ u_{i,j} \in \{0, 1\} \\ v_j \in \{0, 1\} \\ w^j \in IR^p \\ \forall\ i = 1, ..., n\ and\ j = 1, ..., N_{max} \end{cases}$$

Let (V^*, U^*, W^*) be an optimal solution of the problem (P). The optimal number of neurons in the Kohonen map can be computing by the following expression:

$$N = \sum_{j=1}^{N_{max}} v_j^* \qquad (8)$$

Generally, the natural number $N = \sum_{j=1}^{N_{max}} v_j^*$ tend to N_{max}; as consequence, the size of the Kohonen map, associated with a given problem, may not be reduced. In this case, the performance of the Kohonen algorithm may not be improved too. To overcome this problem, we propose to add the term: $\lambda \sum_{j=1}^{N_{max}} v_j$ to objective function of the optimization model (P), where the real λ is a control parameter which permits to make balance between the both of the terms in the energy function. Finally, we obtain the optimization model (PP) in which the number of neurons is penalized:

$$(PP) \begin{cases} min \sum_{i=1}^{n} \sum_{j=1}^{N_{max}} v_j u_{i,j} \|x^i - w^j\|^2 + \lambda \sum_{j=1}^{N_{max}} v_j \\ Subjuct\ to: \\ \quad \sum_{j=1}^{N_{max}} u_{i,j} = 1 \\ \quad \sum_{j=1}^{N_{max}} (1 - v_j) \sum_{i=1}^{n} u_{i,j} = 0 \\ \quad u_{i,j} \in \{0, 1\} \\ \quad v_j \in \{0, 1\} \\ \quad w^j \in IR^p \\ \quad \forall\ i = 1, ..., n\ and\ j = 1, ..., N_{max} \end{cases}$$

Several algorithms have been proposed to solve convex Mixed Integer Non Linear Problem: Branch-and-Bound , Outer Approximation, LP/NLP-based Branch-and-Bound and Branch-and-Cut [2], [7], [10].

IV. Genetic algorithm

Genetic algorithm is proposed to solve this problem. Genetic Algorithm belongs to a class of probabilistic methods called ”evolutionary algorithms” based on the principles selection, Crossover and mutation. GA was introduced by J. HOLLAND [13] that it's based on natural evolution theory of Darwin. Each solution represents an individual who is coded in one or several chromosomes. These chromosomes represent the problem's variables. First, an initial population composed by a fix number of individuals is generated, then, operators of reproduction are applied to a number of individuals selected switch their fitness. This procedure is repeated until the maximums number of iterations is attained. GA has been applied in a large number of optimization's problems in several domains [6], [5], telecommunication, routing, scheduling, and it proves it's efficiently to obtain good solutions [4]. We have formulated the problem as a non linear program with mixed variables.

Genetic Algorithm's framework:

1) Coding individuals;
2) Generate the initial population ;
3) repeat ;
   3-1 Evaluation individuals;
   3-2 Selection of individuals;
   3-3 Crossover ;
   3-4 Mutation;
   3-5 Reproduction;
4) Until attaining the criteria.

Coding

In our application, we have encoded an individual by 3 chromosomes, the first one represents the matrix of weights ”W”, the second represents the vector ”U” , and the 3^{th} represents the vector C which is an array of decision variables who takes 1 if the neuron is activated and 0 other.

Initial Population

Individuals of the initial population are randomly generated, where $u_{i,j}$ and v_j takes the value 0 or 1, and the matrix of weights takes random values in space IR.
There are other ways to generate the initial population like applying others heuristics, this is the case when the research space is constrained what is not our case.

Evaluating individuals

In this step, each individual is assigned a numerical value called fitness which corresponds to its performance; it depends essentially on the value of objective function in this individual. An individual who has a great fitness is the one who is the most adapted to the problem.
The fitness suggested in our work is the following function:

$$Fitness(i) = \frac{1}{1 + objective(i)} \tag{9}$$

Minimize the value of the objective function "objective" is equivalent to maximize the value of the fitness function.

Selection

The selection method used in this paper is the Roulette Wheel Selection (RWS) which is a proportional method of selection. In this method, individuals are selected according to their fitness.
The principle of the RWS method can be summarized in the following schema:

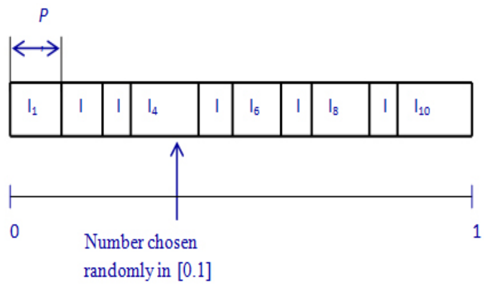

Fig. 2. Selection RWS

where

$$P_i = \frac{f_i}{\sum_{i=1}^{n} f_i} \tag{10}$$

Crossover

The crossover is very important in the algorithm, in this step, new individuals called children are created by individuals selected from the population called parents. Children are constructed as follows: We fix a point of crossover, the parents are cuted switch this point, the first part of parent 1 and the second of parent 2 go to child 1 and the rest go to child 2. In the crossover that we adopted, we choose 3 different crossover points, the first for the matrix of weights, the second is for vector U and the last is for vector C.

Mutation

The role of mutation is to keep the diversity of solutions in order to avoid local optimums. It corresponds on changing the values of one (or several) value (s) of the individuals who are (or were) (s) chosen randomly.

V. Experimental study and discussion of the results

To solve this optimization model, we propose a method using the genetic algorithm [1], [7]. The most theoretical and algorithmical results permit to determine the optimal number of artificial neurons, and to obtain the matrix of weights. This algorithm is tested to realize the training stage. We have trained several heads and chests with different number of iterations. We present in the TABLE I the mean of the remaining neurons associated with the size 324, 400 and 625 neurons. For example for a map of 625 neurons, the mean of remaining neurons is approximately 361 neurons; this mean the percentage of the reduced neurons is approximatively 40%. The percentage of the reduced neurons increases with the size of the initial map.

TABLE I
Results table of Kohonen architecture optimization

Pm	Pc	N_{min}	N_{max}	N
0.2	0.7	256	324	262
0.3	0.8	256	324	286
0.3	0.8	256	400	293
0.2	0.8	256	400	272
0.2	0.7	256	625	422
0.2	0.8	256	625	331

Where:
Pm: Probability of mutation
Pc: Probability of crosser
We have tested the both proposed compression approaches and the original SOM compression technique on medical images. These images of 256 x 256 pixels in size are used, each pixel is coded on 8 bits; see the Fig. 3.
The images are cuted into square blocks of size 4x4 which constitute the input vectors of SOM. After normalization, the blocks are used for generating the codebook by Kohonen algorithm and according to the rules of the Kohonen algorithm. Performances of the above algorithms are evaluated in terms of bit rate (bits per pixel) and Peak Signal-to-Noise Ratio ($PSNR$) is given by:

$$PSNR = 10 \log_{10}\left(\frac{255^2}{MSE}\right)(endb) \tag{11}$$

where Mean Squared Error (MSE) is defined as follows:

$$MSE = \frac{1}{T} \sum_{i=1}^{T} (\widehat{x}_i - x_i)^2 \tag{12}$$

where $\widehat{x}_i$ and x_i denote,respectively, the original and the encoded pixel values and T is the total number of pixels in an image. A topological map square $R \times R$ neurons was used.

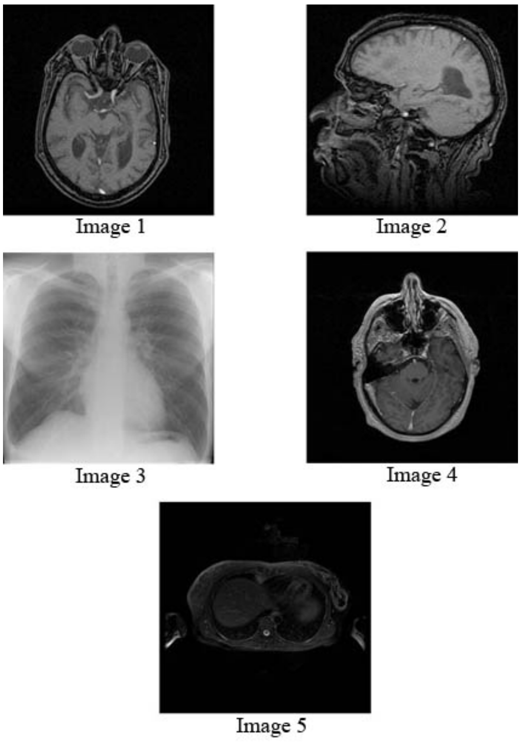

Fig. 3. Original images

To compress on image, we dived it into blocks 4×4 pixels. Numerical results obtained by applying the proposed method (OSOM) and the classical one (SOM) to several images are, respectively, presented in the TABLE III and TABLE II. These tables list, respectively, the test images, the number of iterations, Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio ($PSNR$) associated with tree maps of different size (324, 400 and 625).

TABLE II
NUMERICAL RESULTS OBTAINED BY SOM

		$N_{max} = 324$		$N_{max} = 400$		$N_{max} = 625$	
	N.It	MSE	PSNR	MSE	PSNR	MSE	PSNR
Image 1	100	52.28	30.95	47.99	31.32	48.74	31.25
	200	49.51	31.18	46.85	31.42	47.69	31.35
	1000	47.22	31.38	46.16	31.49	47.16	31.40
Image 2	100	62.31	30.18	62.48	30.17	55.90	30.66
	200	57.96	30.5	61.27	30.26	57.81	30.51
	1000	53.24	30.87	60.43	30.32	57.10	30.56
Image 3	100	28.16	33.63	20.16	35.08	21.14	34.88
	200	25.44	34.08	18.02	35.57	17.62	35.67
	1000	22.73	34.56	15.45	36.24	13.22	36.92
Image 4	100	59.33	30.4	58.31	30.47	53.86	30.82
	200	54.21	30.79	54.74	30.75	50.02	31.14
	1000	48.56	31.27	50.12	31.13	46.73	31.43
Image 5	100	28.42	33.59	25.8	34.01	19.22	35.29
	200	25.53	34.06	22.14	34.68	17.10	35.80
	1000	21.69	34.77	20.54	35.00	15.67	36.18

Legend of the table II

- N.It : Number of Iteration;
- MSE : Mean Squared Error;
- PSNR : Peak Signal-to-Noise Ratio
- N_{max} : Maximal number of artificial neurons in the topology map of Kohonen.

Recall that the proposed method contained in additional phase; this phase consists of solving the proposed model in order to remove the unnecessary neurons from the initial map. For example a map of 400 neurons we obtain a map of 279. Fig. 4 shows PSNR comparison of image 1 between both

TABLE III
NUMERICAL RESULTS OBTAINED BY THE NEW APPROACH (OSOM)

		$N_{max} = 324 \longrightarrow N = 272$		$N_{max} = 400 \longrightarrow N = 289$		$N_{max} = 625 \longrightarrow N = 361$	
	N.It	MSE	PSNR	MSE	PSNR	MSE	PSNR
Image 1	100	57.28	30.55	52.01	30.97	51.00	31.05
	200	55.65	30.67	50.57	31.09	49.88	31.15
	1000	51.43	31.02	49.86	31.15	49.71	31.17
Image 2	100	73.6	29.46	70.71	29.64	73.09	29.49
	200	68.35	29.78	68.62	29.77	66.71	29.89
	1000	63.71	30.09	67.57	29.83	64.74	30.02
Image 3	100	30.85	33.23	22.8	34.55	23.04	34.51
	200	28.43	33.59	20.43	35.03	20.47	35.02
	1000	25.63	34.04	17.11	35.8	15.2	36.31
Image 4	100	64.15	30.06	63.88	30.08	59.01	30.42
	200	61.77	30.22	61.22	30.26	58.11	30.49
	1000	60.41	30.32	59.79	30.36	55.31	30.70
Image 5	100	31.67	33.12	29.47	33.44	21.28	34.85
	200	29.88	33.38	25.13	34.13	19.66	35.19
	1000	26.81	33.85	22.73	34.56	18.26	35.51

Legend of the table III

- N : Optimal number of artificial neurons in the topology map of Kohonen.

approaches SOM and proposed OSOM, for example with compression ratio 93.25 and OSOM with compression ratio 93.61, where PSNR is for the decoded image 1. We can see that the PSNR very close between the classical method and the proposed approach. But proposed method can reduce the number of neurons and time training.

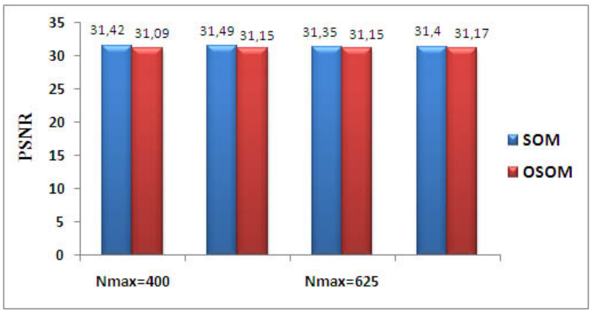

Fig. 4. Comparison between SOM and OSOM in term of PSNR associated with Image1

Fig. 5 (resp. Fig. 6) shows the five reconstructed images (a, b, c, d and e) obtained by the new architecture Optimization model for the Self-Organized Map (OSOM). Our approach permits to construct maps of, approximatively, 361 (resp. 279) neurons from initial maps of 625 (resp. 400) neurons.

A simple comparison between both the reconstructed images, via our approach, and the original images shows that the proposed method it is enable to construct images of good visual quality.

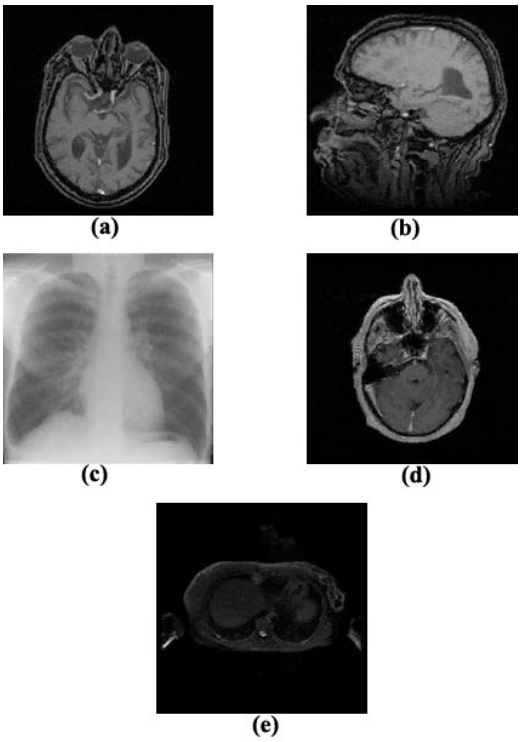

Fig. 5. Reconstructed images by the OSOM : $N_{max} = 600$, $N = 361$

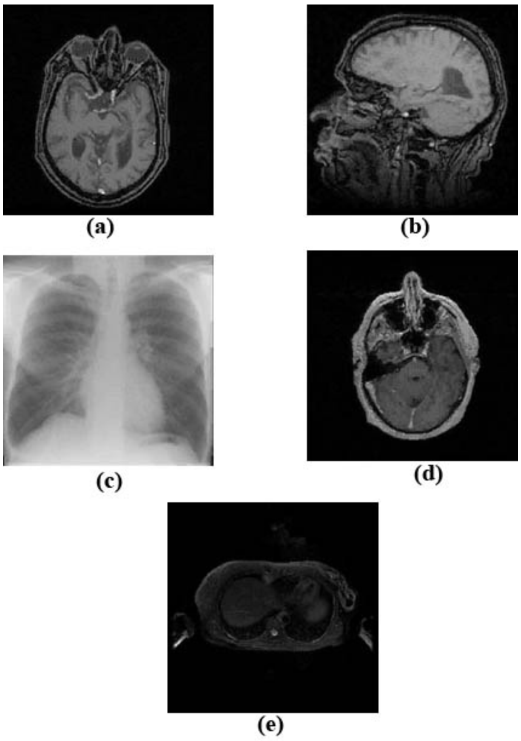

Fig. 6. Reconstructed images by the OSOM : $N_{max} = 400$, $N = 289$

Discussions

From a numerical point of view, our method gives better image quality and low time learning, coding and decoding of the image, so reducing the size of the data dictionary (codebook).
The proposed method reduce the size of map without losing the quality of the compressed images, this mean that the proposed approach makes a balance between the size of map and the quality of learning.
It should be noted that the number N_{max} tend to one value from the set $\{250, ..., 370\}$ in all tests. The necessary number of neurons for the compression of images under study is approximatively 260 neurons.
We can see that the PSNR very close between the classical method and the proposed approach.
It should be noted that the percentage of the reduced neurons grows with the size of the initial map; In fact:

$$Red\%(324) < Red\%(400) < Red\%(625)$$

Where $Red\%(x)$ presents the percentage of the reduced neurons associated with an initial map of size x. It is important to point out the reduction of only one neuron permits the omission of d connections, this imply the reduction of $1000 * d$ instructions if the number of iterations is 1000, where d presents the input vector dimension. Moreover, if we remove one neuron, we gain $4 * d$ bytes; recall that float is encoded on 4 bytes in java programming.

VI. Conclusion

Digital medical imaging technologies have become increasingly important in medical practice and health care for providing assistant tools for diagnosis, treatment, and surgery. Due to the volume of medical images is huge and has grown rapidly, especially on MRI (Magnetic Resonance Imaging) and CT (Computer Tomography), the compression technique must be applied. In this paper, we have presented a new modeling for the Self Organizing Maps (SOM) architecture optimization problem for medical image compression. The GA is especially appropriate to obtain the optimization solution of the complex nonlinear problem. This method is tested to determine the optimal number of artificial neurons in the map and the most favorable weights matrix. From a numerical point of view, our method gives better image quality and low time learning, coding and decoding of the image, so reducing the size of the data dictionary (codebook). The presented method considers the compression techniques on 2-D images. Our concept can be extended to 3-D images. The research issues such as efficiency of the compression rate in 3-D watermarking will be studied in the future.

Acknowledgment

The authors would like to thank the anonymous reviewers for their useful suggestions and comments on the paper.

REFERENCES

[1] M. K. Apalak, M. Yildirim, R Ekici, ”Layer optimization for maximum fundamental frequency of laminated composite plates for different edge conditions”, composites science and technology (68), pp. 537-550, 2008.
[2] Benders J. F. ”Partitioning procedures for solving mixed-variables programming problems”, Numer. Math.4 :238, 1962.
[3] O. T.-C. Chen, B. J. Sheu, and W.-C. Fang, ”Image compression using self-organization networks”, IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 480-489, Oct. 1994.
[4] A. Cichoki, R. Unberhuen, ”Neural Network for Optimization and signal processing”, John Wiley and Sons, New York, 1993.
[5] G. Dreyfus, J. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S.Thiria, L. Hrault; ”Rseaux de neurones Mthodologie et applications”, Eyrolles, 2ime dition, 2003.
[6] J Dréo, A. Ptrowski, P. Siarry, . Taillard, ”Mta-heuristiques pour l’optimisation difficile”, ditions Eyrolles, 2003.
[7] M. Ettaouil, ”Contribution l’étude des problèmes de satisfaction de contraintes et la programmation quadratiques en nombre entiers, allocation statique de tâches dans les systèmes informatiques distribués”, Thèse d’état, Université Sidi Mohammed ben Abdellah, F.S.T. de Fès, 1999.
[8] M. Ettaouil, K. Elmoutaouakil and Y. Ghanou, ”The Continuous Hopfield Networks (CHN) for the Placement of the Electronic Circuits Problem”, Wseas Transactions On Computer, Issue 12, Volume 8, 2009.
[9] M. Ettaouil and Y. Ghanou, ”Neural architectures optimization and Genetic algorithms”, Wseas Transactions On Computer, Issue 3, Volume 8, 526-537, 2009.
[10] M. Ettaouil and C. Loqman, ”Constraint satisfaction problem solved by semi definiterelaxation”, Wseas Transactions On Computer, Issue 7, Volume 7, 951-961, 2008.
[11] M. Goldberg, P. R. Boucher and S. Shlien, ”Image Compression using adaptive vector quantization”, IEEE Transactions on Communication, Vol. 34, No. 2, pp. 180-187, 1986.
[12] R. M. Gray, ”Vector quantization”, IEEE Acoustics, speech and Signal Processing Magazine, pp. 4-29, 1984.
[13] Holland J. ”Adaptation in natural and artificial systems”. Ann Arbor, MI: University of Michigan Press, 1992.
[14] J.Jiang, ” Image compression with neural networks A survey”, Signal processing :image communication n14, pp 737-760, 1999.
[15] T. Kohonen, ”Self-Organizing Maps”. Berlin/Heidelberg, Germany: Springer, vol. 30, 1995.
[16] Y. Linde, A. Buzo and R. M. Gray, ”An algorithm for vector quantizer design”, IEEE Transactions on Communication, Vol. 28, No. 1, pp. 84 - 95, 1980.
[17] N. Nasrabadi and Y. Feng, ”Vector quantization of images based upon the Kohonen self-organizing feature maps”, in IEEE Int. Conf. Neural Networks, San Diego, CA, vol. 1, pp. 101-108, 1988.
[18] Robert, Haykin, ”Neural network approaches to images compression ”, in .Proceedings of the IEEE, vol 82, n 2, 1995.
[19] Stanley, C. and al. , ” Competitive learning algorithms for vector quantization ”, in Neural networks, vol 3, pp. 277-290, 1990.