International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Implementation and Testing of Image Compression using Neural network

Swarn Kiran Kumari
Department of Computer Science & Engg
BBD University, Lucknow, U.P.
swrnkrnkumari@gmail.com

Abstract-- In this paper using neural network to survey and to develop a comprehensive lossless image compression method to achieve a improved compression ratio over other conventional methods used like using PCA algorithm, without retaining the image quality. In this paper a literature survey has been carried out to find and efficient multi-layered neural network. MATLAB software along with its Neural Network and Image Processing toolbox will be used to implement the given technique. The MATLAB software provides various easy to use and readily available built in functions for realizing Neural Network algorithms in quick time. An extensive study of this will be required as well.

Keyword-- Compression, Image Processing, Neural Network, and PCA Algorithm.

1. Introduction

Compression offers a means to reduce the cost of storage and increase the speed of transmission. The images are very large in size and require lot of storage space. Image compression can be lossless and lossy depending on whether all the information is retained or some of it is discarded during the compression process. In lossless compression, the recovered data is identical to the original, whereas in the case of lossy compression the recovered data is a close replica of the original with minimal loss of data. Lossy compression is used for signals like speech, natural images, etc., where as the lossless compression can be used for text and medical type images[1]. Apart from the existing technology on image compression represented by series of JPEG, MPEG and H.26x standards, new technology such as neural networks and genetic algorithms are being developed to explore the future of image coding. Successful applications of neural networks to basic propagation algorithm have now become well established and other aspects of neural network involvement in this technology.

Using neural network to survey and to develop a comprehensive lossless image compression method to achieve a improved compression ratio over other conventional methods used like using PCA algorithm, without retaining the image quality.

1. To develop a lossless image compression technique using neural network.
2. To design and implement image compression using Neural network to achieve better SNR and compression levels.

Neural networks are inherent adaptive systems; they are suitable for handling non-stationarity in image data. The greatest potential of neural networks is the high speed processing that is provided through massively parallel VLSI implementations. The choice to build a neural network in digital hardware comes from several advantages that are typical for digital systems:

1. Low sensitivity to electric noise and temperature.
2. Weight storage is no problem.
3. The availability of user-configurable, digital field programmable gate arrays, which can be used for experiments.
4. Well-understood design principles that have led to new, powerful tools for digital design

In this paper a literature survey has been carried out to find and efficient multi-layered neural network. MATLAB software along with its Neural Network and Image Processing toolbox will be used to implement the given technique. The MATLAB software provides various easy to use and readily available built in functions for realizing Neural Network algorithms in quick time. An extensive study of this will be required as well.

2. Related Work:

The usability and utility of the power of neural network for image compression lies on the following three important aspects:

(a) Selection of efficient multi layered network
(b) Selection of training methods
(c) Test vector.

Thus the present study will have to closely monitor all these three aspects while choosing the optimum Neural Network technique for realization keeping in mind the nature of the input data type i.e. image.

Paper Name	Research Topic	Method	Result
Sadashivappal , Mahesh Jayakar, K.V.S Anand	Performance of different wavelets using SPIHT	R,G and B component of color image are converted	Results are analyzed using PSNR and HVS property.

ISSN: 2454-6844

Babu'' Color Image Compression using SPIHT Algorithm ''	algorithm for compressing color image.	to YcbCr before wavelet transform is applied .Y is luminance component: Cb and Cr are chrominance components of the image.	
Prachi Tripathi'' Image Compression Enhacement using Bipolar Coding with LM Algorithm in Artificial Neural Network	Defines the image compression is to reduce irrelevance image data	In this paper the Lossless method of Image Compresssion using Bipolar Coding Technique with LM algorithm in Artificial Neural Network is proposed by the author	With compression it is possible to reduce file size to 10 percent from the original without noticeable loss in quality.
M. Venkata Subbarao'' Hybrid Image Compression using DWT and Neural Networks	Hybrid Image Compression using DWT and Neural Networks	In this paper DWT and Neural network in Back Propagation methods is used for compression of images	The disadvantages of Joint Photographic Expert Group (JPEG) have overcome in Neural Network based Hybrid image compression .
R,Vanaja,N.Lakshmi Prabha ,DR.N Stalin "Efficient Architecture for SPIHT Algorithm in Image Compression"	A throughput efficient image compression using ,,Set Partitioning in Hierarchial Trees"(SPIHT) algorithm for compression of images	SPIHT use inherent redundancy among wavelet coefficients and suited for both gray and colored Image.	Comparison of SPIHT in both the arithmetic coder and pipelined architecture was enumerated in this paper.
Farnoosh Negaahban, Mohammad Ali Shafieian and Mohammad Rahmanian" Various Novel Wavelet- Based Image Compression Algorithms Using a Neural Networks as a Predictor.	A novel technique in image compression with different algorithms by using the transform of wavelet accompanied by neural networks as a predictor.	This Paper consists of four novel algorithms for image compression. The details sub bands in different low levels of image wavelet decomposition are used as training data for neural network.	Image compression as well as comparing them with each other and well – known jpeg and jpeg 2000 methods

3. Proposed Image Compression Using Neural Network

A two layer feed-forward neural network and the Levenberg Marquardt algorithm[17] was considered. Image coding using a feed forward neural network consists of the following steps:

An image, F, is divided into rxc blocks of pixels. Each block is then scanned to form a input vector x (n) of size p=rxc.

It is assumed that the hidden layer of the layer network consists of L neurons each with P synapses, and it is characterized by an appropriately selected weight matrix $\mathbf{W_h}$.

All N blocks of the original image is passed through the hidden layer to obtain the hidden signals, h(n), which represent encoded input image blocks, x(n) If L<P such coding delivers image compression.

It is assumed that the output layer consists of m=p=rxc neurons, each with L synapses. Let W_y be an appropriately selected output weight matrix. All N hidden vector h(n), representing an encoded image H, are passed through the output layer to obtain the output signal, y(n). The output signals are reassembled into p=rxc image blocks to obtain a reconstructed image, F_r.

There are two error matrices that are used to compare the various image compression techniques. They are Mean Square Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image whereas PSNR is the measure of the peak error.

$$MSE = {}^{I}\!/_{MN} \sum_{y=1}^{m} \sum_{x=1}^{n} [I(x,y) - I'(x,y)]^2$$

The quality of image coding is typically assessed by the Peak signal-to-noise ratio (PSNR) defined as

$$PSNR = 20 \log_{10} [255/\mathrm{sqrt}(MSE)]$$

Training is conducted for a representative class of images using the Levenberg Marquardt algorithm.

Once the weight matrices have been appropriately selected, any image can be quickly encoded using the W_h matrix, and then decoded (reconstructed) using the W_y matrix.

A. Levenberg Marquardt Algorithm

The Levenberg Marquardt algorithm is a variation of Newton's method that was designed for minimizing functions that are sums of squares of other nonlinear functions. This is very well suited to neural network training where the performance index is the mean squared error[17][18].

B. Basic Algorithm

Consider the form of Newton's method where the performance index is sum of squares. The Newton's method for optimizing a performance index F(x) is

$$X_{k+1}= X_k - A_k^{-1} g_k,$$

Where $Ak = \nabla^2 F(x)$ and $g_k = \nabla F(x)$;

It is assume d that F (x) is a sum of squares function:

$$F(x)=\sum_{r=1}^{n} v_i^2(x)=V^T(x)v(x)$$

Then the j[th] element of the gradient will would be

$$[\nabla F(x)]_j = \delta F(x)/\delta S_j = 2\sum_{i=1}^{n} V_i(x)\delta v_i(x)/\delta x_j$$

The gradient can be written in matrix form:

$$\nabla F(x) = 2J^T(x)\, v(x)$$

where J(x) is the Jacobian matrix.

Next the Hessian matrix is considered. The k.j element of Hessian matrix would be :

$$\left[\nabla^2 F(x)\right]_{kj} = \delta^2 F(x)/\delta x_k\, \delta x_j$$

The Hessian matrix can then be expressed in matrix form:

$$\nabla^2 F(x) = 2J^T(x)J(x) + 2S(x)$$

Where

$$S(x)=\sum_{i=1}^{n} V_i(x).\nabla^2 v_i(x)$$

Assuming that S(x) is small, the Hessian matrix is approximated as

$$\nabla^2 F(x) \equiv 2J^T(x)J(x)$$

Substituting the values of ∇2 F(x) & ∇ F(x), we obtain the Gauss-Newton method:

$$X_{k+1} = X_k - [J^T(X_k) J(X_k)]^{-1} J^T(X_k) V(X_k)$$

One problem with the Gauss-Newton over the standard Newton's method is that the matrix $H=J^TJ$ may not be invertible. This can be overcome by using the following modification to the approximate Hessian matrix:

$$G = H + \mu I.$$

This leads to Levenberg –Marquardt algorithm

$$X_{k+1} = X_k - [J^T(X_k) J(X_k)+\mu_{kI}]^{-1} J^T(X_k) V(X_k)$$

Or

$$\Delta X_k = - [J^T(X_k) J(X_k)+\mu_{kI}]^{-1} J^T(X_k) V(X_k)$$

this algorithm has the very useful feature that as μk is increased it approaches the steepest descent algorithm with small learning rate[17].

The iterations of the Levenberg- Marquardt back propagation algorithm (LMBP) can be summarized as follows:

Present all inputs to the network and compute the corresponding network outputs and the errors $e_q = t_q - a^M{}_q$. Compute the sum of squared errors over all inputs. F(x).

$$F(x) = \sum e_q^T e_q = \sum\sum(e_{j.q})^2 = \sum(v_i)^2$$

Compute the Jacobian matrix. Calculate the sensitivities with the recurrence relation. Augment the individual matrices into the Margquardt sensitivities.

Obtain ΔX_k.

Recompute the sum of squared errors using xk $+ \Delta X_k$. If this new sum of squares is smaller than that computed in step 1 then divide μ by v,

let $X_{k+1} = X_k + \Delta X_k$ and go back to step 1. if the sum of squares is not reduced, then multiply μ by v and go back to step 3.

4. Training Procedure

During training procedure data from a representative image or a class of images is encoded into a structure of the hidden and output weight matrices.

It is assumed that an image, F, used in training of size Rx C and consists of rxc blocks.

1. The first step is to convert a block matrix F into a matrix X of size P x N containing training vectors, x(n), formed from image blocks.

That is:

$$P= r.c \text{ and } p.N = R.C$$

2. The target data is made equal to the data, that is:

$$D=X$$

3. The network is then trained until the mean squared error, MSE, is sufficiently small.

The matrices W^h and W^y will be subsequently used in the image encoding and decoding steps.

5. Image Encoding

The hidden-half of the two-layer network is used to encode images. The Encoding procedure can be described as follows[12]:

$$F \rightarrow X,\ H= (W^h.\ X)$$

Where X is an encoded image of F.

6. Image Decoding

International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

The image is decoded (reconstructed) using the output-half the two-layer network. The decoding procedure is described as follows:

$$Y = (W^{y}.\ H),\ Y \rightarrow F$$

These steps were performed using MATLAB (Matrix laboratory). The compression so obtained was though offline learning. In the off-line learning methods, once the systems enters into the operation mode, its weights are fixed and do not change any more.

Algorithms-

1)Read image and select a sub matrix of image of size 64*64
2)Convert the sub matrix into column vector of 4*4 block
3)Jenerate a new nueral network with input as minmax values of column vector of submatrix (the set parameter of nueral network with following values)

a) Activation function ten segminde and leanear function
b)training function trainlm
C)learning rate alfa(a)=.8 momentum coeficient mc=.2 and gaol and show five itreation 1e-5
d)Training of the network using submatrix of image

4)simmulation of train network to obtain compressed block
5)read complite image
6)convert image matrix into column vector produced compressed image using trained network for any given image
7)Calculate mean squire error and psnr
8)Save compressed image
9)end

7. Result and simulation

The above algorithms have been tested using MATLAB R2010a version. MATLAB has dedicated functions package for Neural Network, thus the algorithms has been implemented using this toolbox functions.

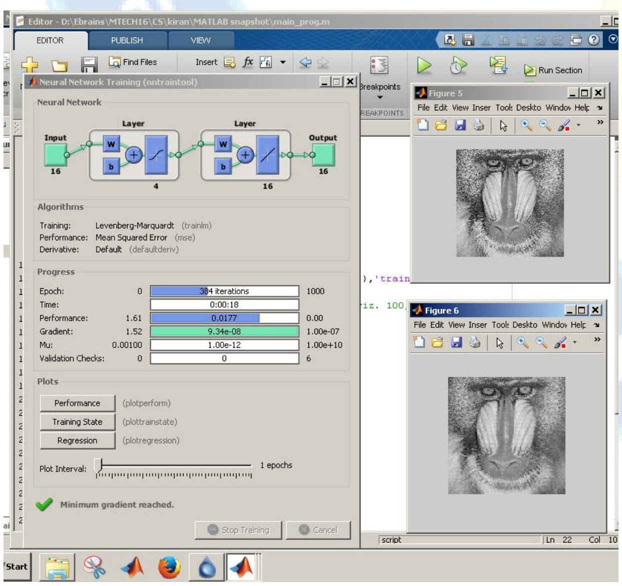
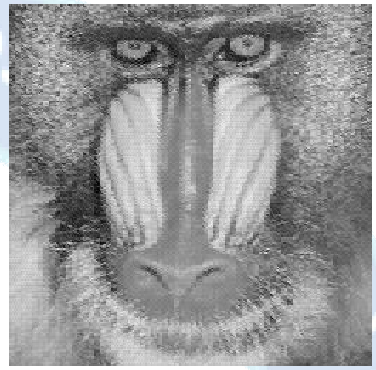
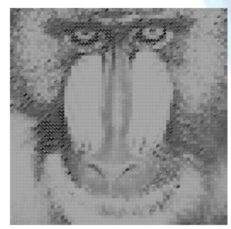


Fig 1. Simulation Results

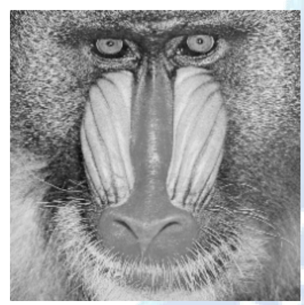
Different test images were taken for the experiment. Each image was of 256x256 pixels size in png format. The three algorithms viz. gradient descent, Lavenberg-Marquardt and quasi newton have been applied and the results and output compressed image are derived as shown below:



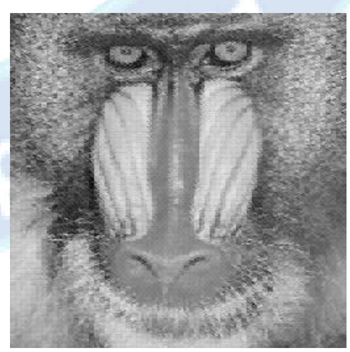
Input Baboon



Output with GDA, MSE 11.4999, PSNR 85.8542



Output with LM, MSE .0016, PSNR 174.3917



Output with BFGS, MSE .0144, PSNR 152.7077

Fig 2. Different Output Results

International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

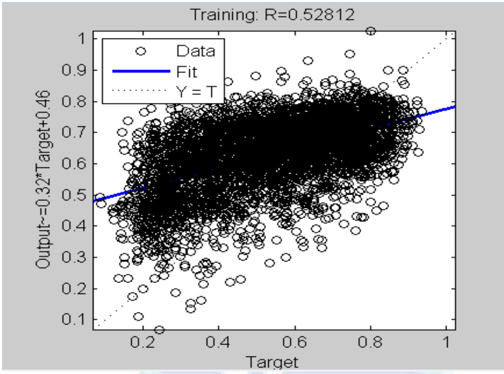


Fig. 3. Regression plot GDA Algorithm

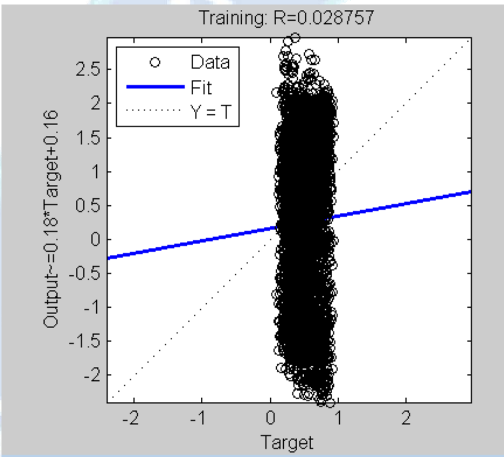


Fig. 4. Regression Plot BFGS Algorithm

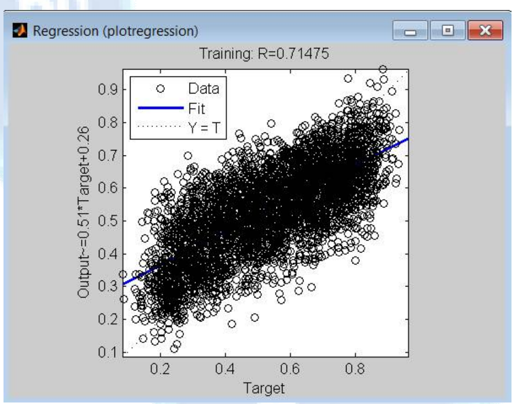


Fig. 5. Regression plot LM Algorithm

Table 1. PSNR(db) and MSE(db) values(LM algorithms), (a=.8,u=.2)

LM algo		
Epochs	MSE	PSNR
100	0.001845	173.22
200	0.0015958	174.67
300	0.0017852	173.55
400	0.0017646	173.67
500	0.0016474	174.35
600	0.0016487	174.35
700	0.0016785	174.17
800	0.0016388	174.41
900	0.0016823	174.14
1000	0.0016412	174.39

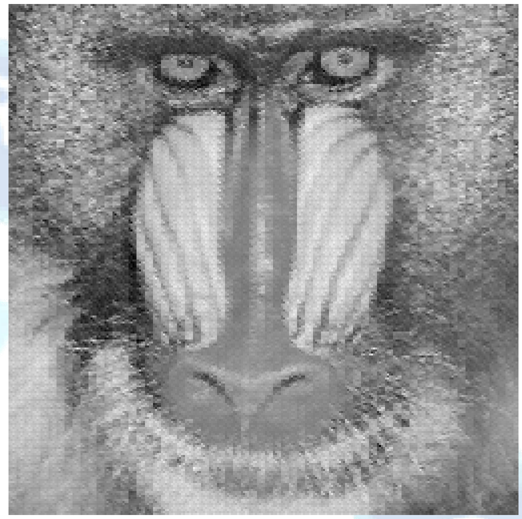


Fig. 6. Reconstructed Image for LM Algo

Table 2. PSNR(db) and MSE(db) values(GDA algorithms), (a=.8,u=.2)

GDA algo		
Epochs	MSE	PSNR
100	116.13	62.721
200	84.944	65.849
300	33.638	75.112
400	10.927	86.357
500	30.963	75.941
600	8.2707	89.141
700	2.2699	102.07
800	71.887	67.518
900	2.9796	99.351
1000	0.00010546	201.84

Fig. 7. Reconstructed Image for GDA Algo

ISSN: 2454-6844

Table 3. PSNR(db) and MSE(db) values(BFG algorithms) (a=.8,u=.2)

BFG algo		
Epochs	**MSE**	**PSNR**
100	1.9415	103.63
200	0.2415	124.48
300	0.070144	136.84
400	0.3425	120.98
500	0.37621	120.04
600	0.14529	129.56
700	0.029672	145.44
800	0.30028	122.3
900	0.0048308	163.6
1000	0.015107	152.19

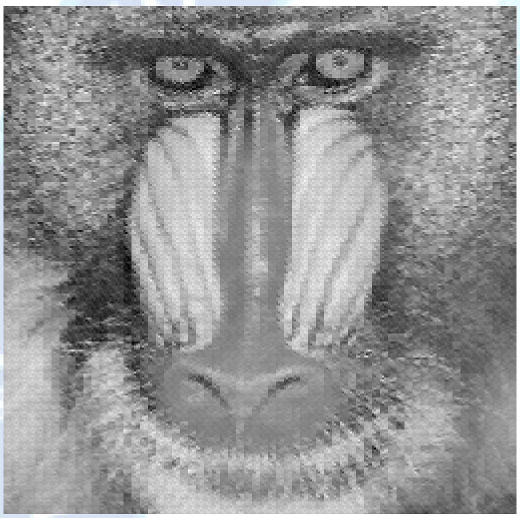


Fig. 8. Reconstructed Image for BFG Algo

8. Conclusion:

This research work takes investigates into novice prospective of image compression using Neural Network Architectures. The various different kinds of training algorithms were applied on a set of test images and there results were compared on various performance parameters viz. MSE, PSNR, Regression plots as well as the quality of the output image. LM Algorithm and BFGS Algorithm gave good results in terms of image quality and PSNR but the time taken in implementation of BFGS Algorithm was considerably less than the LM Algorithm. Thus the possibilities of using this training method and Neural Network are immense as the size of most of the images have been reduced to less than half. The author can clearly visualize the importance of this technique in the future of Image Processing on various other aspects apart from Image Compression like Image Segmentation denoising etc.

References:

[1] E. Watanabe and K. Mori, “Lossy Image Compression Using a Modular Structured Neural Network,” Proceed- ings of IEEE Signal Processing Society Workshop, Wash- ington DC, 2001, pp. 403-412.

[2] M. J. Weinberger, G. Seroussi and G. Sapiro, “The LOCO-I Lossless Image Compression Algorithm: Principles and Sta- ndardization into JPEG-LS,” IEEE Transaction on Image Processing, Vol. 9, No. 8, 2000, pp. 1309-1324.

[3] V. H. Gaidhane, Y. V. Hote and V. Singh, “A New Ap- proach for Estimation of Eigenvalues of Images,” Inter- national Journal of Computer Applications, Vol. 26, No. 9, 2011, pp. 1-6.

[4] S.-G. Miaou and C.-L. Lin, “A Quality-on-Demand Algo- rithm for Wavelet-Based Compression of Electrocardio- gram Signals,” IEEE Transaction on Biomedical Engi- neering, Vol. 49, No. 3, 2002, pp. 233-239.

[5] S. N. Sivanandam, S. Sumathi and S. N. Deepa, “Introduction to Neural Network Using MATLAB 6.0,” 2nd dition, Tata Mc-Graw Hill Publication, Boston, 2008.

[6] http://en.wikipedia.org/wiki/Neuralnetwork.

[7] R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, Reading. MA: Addison Wesley, 2004.

[8] A. Rahman, Chowdhury Mofizur Rahman, “A New Approach for Compressing Color Images using Neural Network”, Proceedings of International Conference on Computational Intelligence for Modeling,Control and Automation – CIMCA 2003 ,Vienna, Austria, 2003.

[9] R. C. Gonzalez, R. E. Woods and S. L. Eddins, “Digital Image Processing Using MATLAB,” Pearson Edition, Dorling Kindersley, London, 2003.

[10] J.-X. Mi and D.-S. Huang, “Image Compression Using Principal Component Analysis Neural Network,” 8th IEEE International Conference on Control, Automation, Robotics and Vision, Kunming, 6-9 December 2004, pp. 698-701.

[11] S.-T. Bow, B. T. Bow and S. T. Bow, “Pattern Recogni- tion and Image Processing,” Revised and Expanded, 2nd Edition, CRC Press, Boca Raton, 2002.

[12] M. Nixon and A. Aguado, “Feature Extraction & Image Processing,” 2nd Edition, Academic Press, Cambridge, 2008, pp. 385-398.

[13] S. N. Sivanandam, S. Sumathi and S. N. Deepa, “Introdu- ction to Neural Network Using MATLAB 6.0,” 2nd Edi- tion, Tata Mc-Graw Hill Publication, Boston, 2008.

[14] A. Laha, N. R. Pal and B. Chanda, “Design of Vector Quantizer for Image Compression Using Self Organizing Feature Map and Surface Fitting,” IEEE Transactions on Image Processing, Vol. 13, No. 10, October 2004, pp. 1291-1303. doi:10.1109/TIP.2004.833107

[15] G. Qiu, T. J. Terrell and M. R. Varley, “Improved Image Compression Using Back Propagation Networks,” In: P. J. G. Lisbao and M. J. Taylor, Eds., Proceeding of the Workshop on Neural Network Applications and Tools, IEEE Computer Society Press, Washington DC, 1994, pp. 73-81.

[16] A. Laha, N. R. Pal and B. Chanda, “Design of Vector Quantizer for Image Compression Using Self Organizing Feature Map and Surface Fitting,” IEEE Transactions on Image Processing, Vol. 13, No. 10, October 2004, pp. 1291-1303. doi:10.1109/TIP.2004.833107

[17] V. Singh, I. Gupta and H. O. Gupta, "ANN Based Estimator for Distillation Using Levenberg-Marquardt Approach," Engineering Applications of Artificial Intelli- gence, Vol. 20, No. 2, 2007, pp. 249-259. doi:10.1016/j.engappai.2006.06.017

[18] M. I. A. Lourakis, "A Brief Description of the Levenberg- Marquardt Algorithm Implemented by Levmar," Founda- tion of Research and Technology, Vol. 4, 2005, pp. 1-6.