World Applied Sciences Journal 22 (11): 1572-1580, 2013
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2013.22.11.2828

A New Robust Blind Watermarking Method Based on Neural Networks in Wavelet Transform Domain

M. Vafaei, H. Mahdavi-Nasab and H. Pourghassem

Department of Electrical Engineering, Najafabad Branch, Islamic Azad University, Isfahan, Iran

Abstract: In this paper, a blind watermarking method based on neural networks in discrete wavelet transform domain is proposed. Robustness and imperceptibility are main contradictory requirements of a watermark. In the proposed method, better compromises are achieved using artificial neural networks to adjust the watermark strength. A binary image is used as the watermark and embedded repetitively into the selected wavelet coefficients, which also improves the watermark robustness. Experimental results demonstrate that the proposed scheme has a simultaneous good imperceptibility and high robustness against several types of attacks, such as Gaussian and salt and pepper noise addition, cropping, mean and median filtering and JPEG compression.

Key words: Blind watermarking・Discrete wavelet transform・Neural network

INTRODUCTION

With modern developments of information technology, widespread communication networks and digital multimedia, effective ways are needed to protect the security of digital data. Digital watermarking is one of the proposed solutions for copyright protection in multimedia (such as image, sound and video), in which a specified hidden signal (watermark) is embedded in digital data [1]. Robustness and imperceptibility are two basic requirements of digital image watermarking that are contradictory.

Many digital watermarking algorithms have been proposed in spatial and transform domains. The techniques in spatial domain still show relatively low capacity and are not robust enough to lossy image compression and other image processing operations [2, 3]. On the other hand, frequency domain techniques can embed more bits as watermark and are more robust to attacks. Transforms such as discrete Fourier transform [4], Discrete Cosine Transform (DCT) [5] and Discrete Wavelet Transform (DWT) [6, 7] are generally used for watermarking in the frequency domain. Watermark embedding can generally be classified into two categories [8]: Spread Spectrum based [9] and quantization based watermarking [7, 10, 11]. The spread spectrum methods add a pseudorandom pattern into host image to embed a watermark. This watermark can be detected by correlating with the same pattern or by applying other statistics to the watermarked image. In quantization watermarking a set of features extracted from the host images are quantized so that each watermark bit is represented by a quantized feature value. This technique improves the robustness to JPEG compression and other typical attacks.

In [7], the host image was decomposed using 3-level DWT. The watermark was embedded into components of the third decomposition layer of the DWT of an image. Every seven non-overlap wavelet coefficients of the host image were grouped into a block. The differences between local maximum and local second maximum values were modified to the watermark bit. In [11], to achieve the secrecy of watermark, variable block size was used for embedding a watermark bit using different sub-bands. The difference between our work and related works lie in the more elaborate selection of wavelet coefficients for watermark embedding, the block selection process and Artificial Neural Network (ANN) inputs.

In recent years, neural networks pave the way for the further development of watermarking techniques by imitating the learning ability of brain. Neural networks are applied either to improve watermark extraction or to determine the strength of watermark [12-14].

Corresponding Author: Mohamad Vafaei, Department of Electrical Engineering, Najafabad Branch, Islamic Azad University, Isfahan, Iran.

As two examples, Mei proposed a method for deciding the watermark strength using DCT coefficients [13] and Davis proposed a method to implement an automated system of creating maximum-strength watermarks [14].

In this paper, a blind watermarking method based on Feed forward Neural Networks (FNN) is proposed. We embed the watermark in the components of the third decomposition layer of the DWT of an image.

The scientific contributions of this paper can be summarized as follows:

- The quantization watermark embedding is used to improve the robustness to JPEG compression and other typical attacks
- Artificial Neural Networks (ANN) are used to balance the two contradictory requirements (robustness and imperceptibility) and to detect the watermark strength automatically.
- The watermark is embedded repetitively into the selected wavelet coefficients. A result of the redundancy is improved the watermark robustness.

MATERIALS AND METHODS

Discrete Wavelet Transform (DWT): The basic idea of DWT is to split a signal into two parts, usually high and low frequency bands. The edge components of the signal are largely confined in the high frequency. The low frequency part is split again into two parts of high and low frequencies. This process is continued until the signal has been entirely decomposed or stopped before by the application at hand.

Original signal x[n] can be decomposed recursively for j=J+1, J,…, J_0 as:

$$f_{j-1}^{low}(k) = \sum_n h_{n-2k} \cdot f_j(n) \tag{1}$$

$$f_{j-1}^{high}(k) = \sum_n g_{n-2k} \cdot f_j(n) \tag{2}$$

where $h(n)$ and $g(n)$ are low and high pass filters, respectively. The coefficients $f_{J_0}^{low}(k), f_{J_0}^{high}(k), f_{J_0+1}^{high}(k), \ldots, f_J^{high}(k)$ are called DWT of x[n]. Where $f_{J+1}^{low}(k) = x[k]$, J+1 and J_0 are the high and low resolution level index, respectively; $f_{j0}^{low}(k)$ is the lowest resolution part of x[n] (approximation) and $f_j^{high}(k)$, $J_0 \leq j \leq J$ are the details of x[n] at various bands of frequencies. Furthermore, the original signal can be reconstructed using the inverse DWT (IDWT):

$$f_j^{low}(n) = \sum_k (h_{n-2k} \cdot f_{j-1}^{low}(k) + g_{n-2k} \cdot f_{j-1}^{high}(k)) \tag{3}$$

To ensure the above IDWT and DWT relationship, the following orthogonality condition on the filters *H(w)* and *G(w)* must hold (Figure 1):

$$|H(w)|^2 + |G(w)|^2 = 1 \tag{4}$$

where,

$$H(w) = \sum_k h(k)e^{-jkw}, G(w) = \sum_k g(k)e^{-jkw} \tag{5}$$

The DWT and IDWT for a two dimensional signal x(m, n) can be similarly defined by implementing one dimensional DWT and IDWT for each dimension m and n separately. An image can be decomposed through a pyramid structure with various band information such as low-low (LL), high-low (HL), low-high (LH) and high-high (HH) frequency bands. An example of three levels decomposition is shown in Figure 2 [15, 16].

Feed forward Neural Network (FNN): ANN is a powerful tool that provides an optimization procedure using high-speed computation. A general classification of ANN is illustrated in Figure 3:

A typical FNN architecture with three layers is depicted in Figure 4. This consists of an input, a hidden and an output layer. The number of nodes in the input and the output layers depend on the number of input and output variables, respectively. Each node is fully connected to its adjacent layers. Two nodes of each adjacent layer are directly connected to one another, which is called a link. Each link has a weighting value to represent the relational degree between two nodes.

Back Propagation Neural Network (BPNN) is the most widely used among FNNs. It is a supervised learning neural network. It uses steepest descent method to approximate arbitrary non-linear relations between input and output. Many different variations of BPNN have been proposed including gradient descent with momentum, adaptive learning rate, resilient BP, conjugate gradient, quasi-Newton and Levenburg-Marquardt (LM) algorithms. The LM algorithm is used to increase the training speed and make the training avoid getting into local minimum. It acts as a compromise between the steepest-descent method with stable but slow convergence and the Gauss-Newton method with opposite characteristics [19].

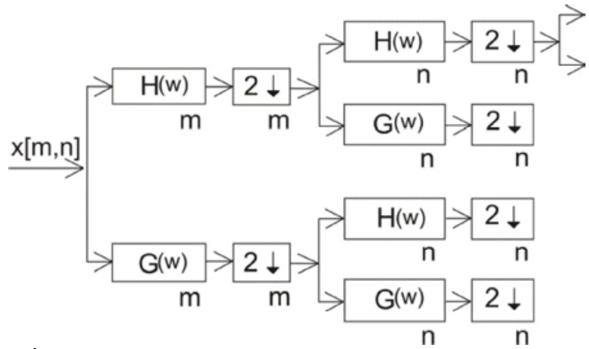

Fig. 1: DWT decompostion of an image

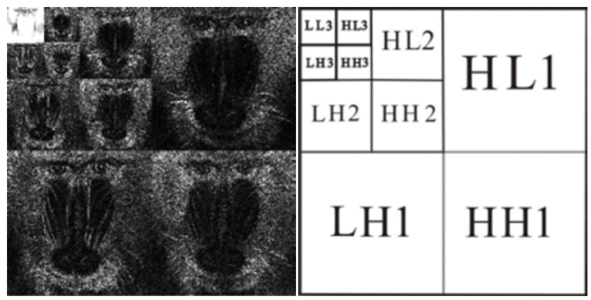

Fig. 2: The pyramidal three level decomposition of an image

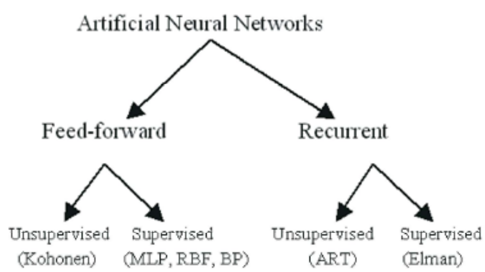

Fig. 3: General classes of ANN [17]

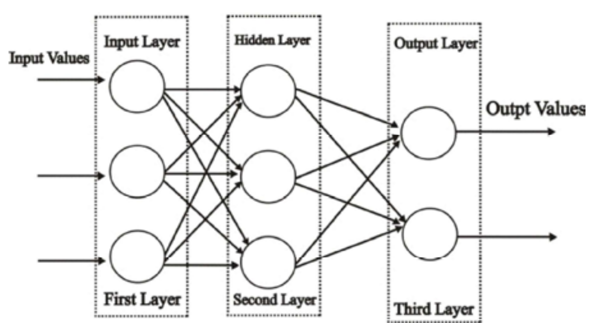

Fig. 4: Typical FNN architecture [18]

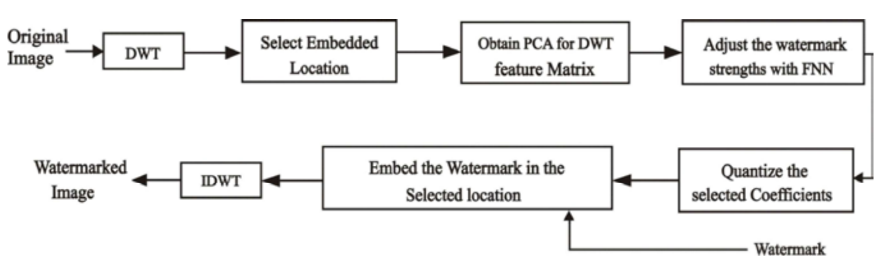

Fig. 5: Block diagram of the proposed watermarking embedding scheme

Table 1: Notations and acronyms

α	Watermark Strength
w_j	Watermark bit
σ	Watermark Extraction Function
BPNN	Back Propagation Neural Network
CDF	Cohen Daubechies Feauveau
DWT	Discrete Wavelet Transform
FNN	Feed-forward Neural Network
IDWT	Inverse Discrete Wavelet Transform
LM	Levenburg Marquardt
MSE	Mean Square error
NA	Not Available
NC	Normalized Correlation
PCA	Principal Component Analysis
PSNR	Peak Signal to Noise Ratio
QF	Quality Factor
$\oplus$	XOR Operator
$\overline{\oplus}$	XNOR Operator
$\lfloor . \rfloor$	Lower Integer Truncation

Principal Component Analysis (PCA): PCA is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of correlated variables into a set of values of less correlated variables called principal components. PCA plots the data into a new coordinate system where the data with maximum covariance are plotted together and is known as the first principal component. Similarly, there are second and third principal components and so on. The first principal component has the maximum energy concentration [20, 21].

The Proposed Method

Watermark Embedding: Here, the construction of watermark and the whole process of embedding the watermark into the host image are described. The watermark is embedded by changing the value of wavelet coefficients; then neural networks are used to automatically control and create the maximum image-adaptive watermark strength. Figure 5 demonstrates the block diagram of the proposed embedding method.

Table 1 shows the notations and acronyms used in the proposed method.

Wavelet Transformation of Images: Due to the linear phase and favorable signal reconstruction of CDF-9/7 [22], this biorthogonal wavelet was used for analyzing the host image.

Sub-Band Selection: Host image is decomposed using 3-level DWT. Increasing the number of levels improves the robustness. But it can reduce the payload capacity. As justified earlier, LL3 sub-band cannot be used for embedding as it contains important low frequency information and any minor change in this band coefficients leads to major perceptual distortion in an image. HH1~HH3 sub-bands are not suitable for embedding as they are very susceptible to compression. The proposed method utilizes LH3 and HL3 bands for embedding the watermark as shown in Figure 2.

The Embedding Algorithm: In the proposed method, the differences of wavelet coefficients are quantized according to the watermark bit. This is similar to the work presented in [7, 11], but in the proposed method the second and third maximum values are modified. This way, the host image seems to be less manipulated and better preserved.

Here, the HL3 sub-band is subdivided into non-overlapping small blocks along the columns from top to bottom and then left to right. Block size is a trade-off between capacity and robustness. Considering blocks of smaller size will increase the capacity at the cost of robustness. Block size of 4×1 appear to satisfy the requirements for host images of size 512×512. Since the host image is considered 512×512, the HL3 sub-band size will be 64×64 and one bit of watermark is embedded in 4 pixels of each small block as visualized in Figure 6. Therefore the watermark size can be 32×32 bits at maximum; which is taken here for maximum performance. LH3 sub-band includes the horizontal details of the third level of wavelet decomposition, therefore it is subdivided into 1×4 blocks and one bit of watermark is embedded in each block (Figure 6). The positive values of 4 coefficients are sorted in increasing order in each block. Considering C_i, i=1...4 as these sorted values, the distance between C_1 and C_4 is defined as [23]:

$$\Delta = \frac{C_4 - C_1}{2} \times \alpha \quad (6)$$

where α is the watermark strength. The distance of C_2 and C_3 is quantized according to Δ, defined as:

$$Distance = \frac{C_3 - C_2}{\Delta} \quad (7)$$

The *Distance* is modified to its closest even or odd integer according to the value of current watermark bit as:

$$Distance^{new} = \begin{cases} \lfloor Distance \rfloor + (1 \overline{\oplus} w_j) if \lfloor Distance \rfloor is\ even \\ \lfloor Distance \rfloor + (1 \overline{\oplus} w_j) if \lfloor Distance \rfloor is\ old \end{cases} \quad (8)$$

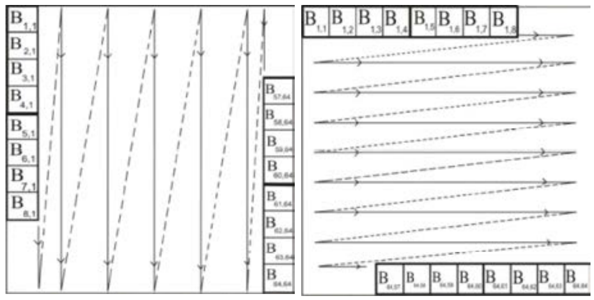

Fig. 6: Location of the 4 coefficients in HL3 and LH3 respectively

This way $Distance^{new}$ is the closest even or odd integer to *Distance* and w_j is the watermark bit that we are going to embed in the 4 coefficients.

A zero watermark bit (w_j) results to an even and a one bit toan odd value of $Distance^{new}$.

To change the value of C_3 $-C_2$, both C_2 and $C_3$are modified by the same value for keeping constant the sum of 4 coefficients, thus C_2 and C_3 are changed as:

$$C_2' = C_2 - \frac{Distance^{new} - Distance}{2} \times \Delta \tag{9}$$

$$C_3' = C_3 + \frac{Distance^{new} - Distance}{2} \times \Delta \tag{10}$$

Selecting higher amounts of α causes an increase in watermark robustness against attacks. On the contrary, when α decreases, image quality will be more likely preserved. However, the watermark is more vulnerable to noises because it is weaker. Therefore, there is a tradeoff for α between the robustness and imperceptibility.

Feature Extraction by PCA: As mentioned above, a watermark bit is embedded into 4 coefficients of small blocks at each sub-band of the third decomposition layer. The mean values of each small block are calculated and stored in row vectors of sizes 1×1024 for each sub-band and defined as *Vectordata*. The feature extraction procedure is illustrated in Figure 7.

The PCA is used to reduce the dimension of the DWT. Since the main features locate at the first rows or columns, we select only the first forty rows of the PCA matrix. The PCA factors of each image are obtained by multiplying the *MFpca* in the *Vectordata*.

Training the Neural Network: Function approximation is one of the important applications of FNN. Here, the FNNs with three layers are used to adjust the watermark strengths (α) automatically and to obtain the most acceptable amount for it. The resulting PCA factors of each image are considered as the input to the FNN.

$$\begin{aligned}
&Mvectors = mean(Vectordata)\\
&AVGmatrix(i,:) = Mvectors - Vectordata(i,:)\\
&CVData = C\,\mathrm{ov}(AVGmatrix)\\
&Fpca = Pca\,\mathrm{cov}(CVData)\\
&MFpca = Fpca(:,1:40)'
\end{aligned}$$

Fig. 7: Feature extraction procedure

The maximum watermark strengths of each image are considered as the training targets, which are obtained by exhaustive subjective experiments with many different values of α. This expects that the image quality does not drop, while getting a robust watermark.

The training samples set of 20 different grayscale images of size 512×512 in 'gif' formats such as "Elaine'', "Boat'', "Couple'', "Cat'', "Mount hood'', "House'', "Man'', "Harbour'', "Car'', "Bridge'' are used.

Before training, all the inputs and targets have to be scaled so that they always fall within a specified range. In the case of inputs, they are normalized in order to be in the range [-1, 1] according to equation (11) and the outputs are normalized to fall in the range [0, 1] according to equation (12) [24].

$$\overline{Input}_i = \frac{2Input_i - Input_{\max} - Input_{\min}}{Input_{\max} - Input_{\min}} \tag{11}$$

$$\overline{\mathrm{Target}}_i = \frac{\mathrm{Target}_i - \mathrm{Target}_{\min}}{\mathrm{Target}_{\max} - \mathrm{Target}_{\min}} \tag{12}$$

To obtain the watermarked image the IDWT is carried out on the analysed sub-bands.

Watermark Extraction Algorithm: The block diagram of the proposed watermark extraction is shown in Figure 8. Here, the same calculations as in watermark embedding stage are carried out for Δ and *Distance* from HL3 and LH3 sub-bands and the closest integer to *Distance* is calculated. With regard to being even or odd, the one or zero of the intended bit is specified. Since each bit of the watermark is embedded in 2 different locations of these

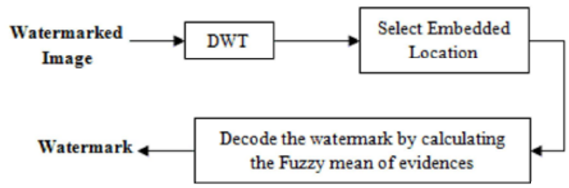

Fig. 8: Block diagram of the proposed watermark extraction method

sub-bands, also two amounts of *Distance* are obtained for each bit of watermark. Suppose there are 2 different quantities $Distance_i$, i=1, 2 for each bit of watermark. An appropriate method should be adopted in order to make decision between these 2 quantities. We use fuzzy mean [25] of $Distance_1$ and $Distance_2$.

The $Belief_i$ value is calculated as:

$$Belief_i = 1 - 2 \times | Distance - round(Distance_i)| \quad i=1, 2 \quad (13)$$

where $Belief_i$ represents the fuzzy membership of each evidence. The vote of each evidence is calculated as:

$$Vote_i = \begin{cases} -1 \ \text{if} \ round \ (Distance_i) \ \text{is even} \\ +1 \ \text{if} \ round \ (Distance_i) \ \text{is odd} \end{cases} \quad (14)$$

where *round* is used for rounding the nearest integer of its argument.

The decision criteria is computed as:

$$\sigma = \sum_{i=1}^{2} (Belief_i \times Vote_i) \quad (15)$$

Ultimately the value of *wj* bit is specified according to equation 16.

$$w_j = \begin{cases} 0 & \text{if}\, \sigma < 0 \\ 1 & \text{if}\, \sigma > 0 \end{cases} \quad (16)$$

RESULTS AND DISCUSSION

The proposed watermarking method is implemented in MATLAB. Four gray scale images, ‘‘Lena’’, ‘‘Baboon’’, ‘‘Airplane’’, ‘‘Barbara’’ are used as test images. All test images are of sizes 512×512 and the watermark is a 32×32 binary image. Table 2 depicts the watermark and watermarked images of the original images.

The evaluation of the watermarked image quality is based on *PSNR*; given as:

Table 2: Original, watermark and watermarked test images

		Watermark Image		
Image Size	Original Image	Image	Size	Watermarked Image
512×512			32×32	
				
				
				

$$PSNR = 10\log_{10} \frac{255^2}{MSE} \quad (17)$$

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{i=M} \sum_{j=1}^{j=N} [I(i,j) - IW(i,j)]^2 \quad (18)$$

where *M* and *N* are the rows and columns of host image, *I(i, j)* and *IW (i, j)* represent the original and watermarked images.

Since the watermark is embedded into 2 sub-bands of HL3, LH3, we must use an NN for each sub-band and calculate the average of NN outputs to determine the watermark strength. Each sub-band estimator is a three-layer FNN. The numbers of neurons in hidden and output layers are 40 and 1 respectively. 20 samples are used for training each sub-band estimator. The FNN has 40-40-1 structure to estimate the watermark strength A Levenberg-Marquardt training algorithm was selected for training the network until the MSE is less than 5e-5. The transfer function of the hidden layer is considered sigmoid and a linear transfer is used for the output layer. Table 3 shows amounts of watermark strength (α) for the four different test images with their respective PSNR values.

We tested the watermarked images under the attacks of JPEG compression, median filter, mean filter, noise addition and cropping.

The quality of watermark extracted from embedded image is measured by the NC. The NC between the embedded watermark $W(i,j)$ and the extracted watermark $\hat{W}(i,j)$ is defined as:

Table 3: Watermark strength with their respective PSNR Values

Image	α	PSNR(db)
Lena	0.43	48.25
Baboon	0.76	43.55
Airplane	0.57	45.26
Barbara	0.46	47.73

Table 4: NC values against JPEG compression (%)

	Image			
Quality Factor	Lena	Baboon	Airplane	Barbara
JPEG QF=40	0.96	0.98	0.93	0.97
JPEG QF=60	0.99	1	0.95	0.99
JPEG QF=90	1	1	0.98	1
JPEG QF=100	1	1	0.99	1

Table 5: NC values between the original and extracted watermarks after median filter attack

	Median filter	
Image	(3×3)	(5×5)
Lena	1	0.87
Baboon	0.95	0.72
Airplane	0.96	0.78
Barbara	0.99	0.85

Table 6: Comparing watermark NC values of the proposed method with the methods presented in [7, 11] for various attacks

Attack	Ref. [7] (PSNR=44.25)	Ref. [11] (PSNR=42.02)	Proposed method (PSNR=48.25)
Median filter (3×3)	0.88	0.90	1
Median filter (5×5)	0.74	0.76	0.87
Mean filter (3×3)	0.91	0.95	0.98
JPEG(QF=10)	0.41	0.34	0.36
JPEG(QF=20)	0.68	0.67	0.67
JPEG(QF=40)	0.95	0.90	0.96
JPEG(QF=60)	0.99	0.95	0.99
JPEG(QF=90)	1	0.99	1
JPEG(QF=100)	1	1	1
Gaussian Noise	NA	0.81	0.92
Salt-pepper	NA	NA	0.89

$$NC = \frac{\sum_i \sum_j W(i,j).\hat{W}(i,j)}{\sum_i \sum_j [W(i,j)]^2} \quad (19)$$

The robustness against cropping and JPEG compression with different quality factors are shown in Figure 9 and Table 4 respectively. Table 5 shows the robustness of the proposed method under the median filter attack.

Finally the proposed method is compared with the methods presented in [7, 11] for the same conditions, using the Lena image. As shown in Table 6, our method's PSNR is higher than related works and is more robust in resisting several attacks, especially filtering attacks such as median and mean. Also, the improved performance against different noise types is noticeable.

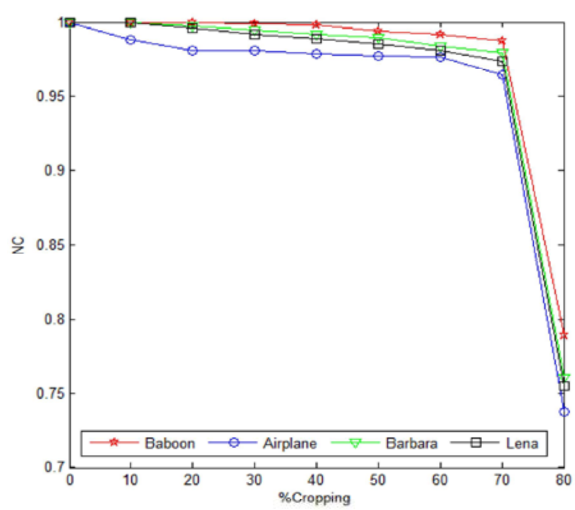

Fig. 9: NC values against cropping attack (%)

CONCLUSION

A blind digital watermarking algorithm based on feed-forward neural networks was presented. The host image was decomposed into wavelet domain and watermark bits embedded in the appropriately selected sub-band coefficients. It is shown that the neural networks can satisfactorily maximize the watermark strength using proper trainings; in addition to being adaptive based on the knowledge of the block features. The simulation results illustrate that the proposed method has good imperceptibility and high robustness simultaneously to different types of attacks such as cropping, filtering and noise addition.

REFERENCES

1. Cox, I. and M. Miller, 1997. A Review of Watermarking and the Importance of Perceptual Modeling. Proceedings of the SPIE/IST& T Conference on Human Vision and Electronic Imaging II, San Jose, pp: 92-99.
2. Hsieh, M.S., D.C. Tseng and Y.H. Huang, 2001. Hiding Digital Watermarks Using Multiresolution Wavelet Transform. IEEE Transactions on Industrial Electronics, 48(5): 875-882.
3. Mukherjee, D.P., S. Maitra and S.T. Acton, 2004. Spatial Domain Digital Watermarking of Multimedia Objects for Buyer Authentication. IEEE Transactions on Multimedia, 6(1): 1-15.
4. Doncel, V.R., N. Nikolaidis and L. Pitas, 2007. An Optimal Detector Structure for the Fourier Descriptors Domain Watermarking of 2D Vector Graphics. IEEE Transactions on Visualization and Computer Graghics, 13(5): 851-863.
5. Cox, I.J., J. Kilian, F.T. Leighton, T. Shamoon, 1997. Secure Spread Spectrum Watermarking for Multimedia. IEEE Transactions on Image Processing, 6(12): 1673-1687.
6. Bao, P. and X. Ma, 2005. Image Adaptive Watermarking Using Wavelet Domain Singular Value Decomposition. IEEE Transactions on Circuits and Systems for Video Technology, 15(1): 96-102.
7. Lin, W.H., S.J. Horng, T.W. Kao, P. Fan, C.L. Lee and Y. Pan, 2008. An Efficient Watermarking Method Based on Significant Difference of Wavelet Coefficient Quantization. IEEE Transactions on Multimedia, 10(5): 746-757.
8. Wu, M. and B. Liu, 2003. Data Hiding in Image and Video. I. Fundamental Issues and Solutions. IEEE Transactions on Image Processing, 12(6): 685-695.
9. Mairgiotis, A.K., N.P. Galatsanos and Y.Yang, 2008. New Additive Watermark Detectors Based on A Hierarchical Spatially Adaptive Image Model. IEEE Transactions on Information Forensics and Security, 3(1): 29-37.
10. Xie, L. and G.R. Arche, 2001. A Class of Authentication Digital Watermarks for Secure Multimedia Communication. IEEE Transaction on Image Processing, 10(11): 1754-1764.
11. Lin, W.H., Y.R. Wang, S.J. Horng, T.W. Kao and Y. Pan, 2009. A Blind Watermarking Method Using Maximum Wavelet Coefficient Quantization. International Journal of Expert Systems with Applications, Elsevier, 36(9): 11509-11516.
12. Yang, Q.T., T.G. Gao and L. Fan, 2010. A Novel Robust Watermarking Scheme Based on Neural Network. International Conference on Intelligent Computing and Integrated Systems. Guilin, pp: 71-75.
13. Mei, S.C., R.H. Li, H.M. Dang and Y.K. Wang, 2002. Decision of Image Watermarking Strength Based on Artificial Neural-networks. In Proceedings of the 9th International Conference on Neural Information Processing, pp: 2430-2434.
14. Davis, K.J. and K. Najarian, 2001. Maximizing Strength of Digital Watermarks Using Neural Networks. International Joint Conference on Neural Networks. Washington DC, pp: 2893-2898.
15. Xia, X., C. Boncelet and G. Arce, 1998. Wavelet Transform Based Watermark for Digital Images. Optics Express, 3(12): 497-511.
16. Shensa, M.J., 1992. The Discrete Wavelet Transform: Wedding the A Trous and Mallat Algorithms. IEEE Transactions on Signal Processing, 40(10): 2464-2482.
17. Qianhui, Y. and K. Wang, 2009. An Improved Watermarking Method Based on Neural Network for Color Image. International Conference on Mechatronics and Automation. Changchun, pp: 3113-3117.
18. Hong, S., W. Zhang, W. Feng and H. Yang, 2008. Blind Watermarking Scheme Based on Neural Network. In Proceedings of the 7th World Congress on Intelligent Control and Automation. Chongqing, pp: 5985-5989.
19. Hagan, M.T. and M.B. Menhaj, 1994. Training Feedforward Networks with the Marquardt Algorithm. IEEE Transactions on Neural Networks, 5(6): 989-993.
20. Gonzalez, R.C., R.E. Woods and S.L. Eddins, 2004. Digital Image Processing Using Matlab. Pearson Prentice Hall. New Jersey.

21. Yavuz, E. and Z. Telatar, 2007. Digital Watermarking With PCA Based Reference Images. Lecture Notes in Computer Science, Springer-Verlag. Berlin Heidelberg, pp: 1014-1023.
22. Cohen, A., I. Daubechies and J.-C. Feauveau, 1992. Biorthogonal Base of Compactly Supported Wavelets. Communications on Pure and Applied Mathematics, 45(5): 485-560.
23. Xie, L. and G.R. Arce, 1998. Joint Wavelet Compression and Authentication Watermarking. International Conference on Image Processing, pp: 427-431.
24. Lai, F., C.Z. Yong and T. Long, 2005. Novel Perceptual Modeling Watermarking with MLF Neural Networks. International Journal of Information Technology, 1(1): 40-43.
25. Kang, M., L.T. Ho, Y. Kim, C.H. Kim and J.M. Kim, 2011. Image Watermarking Using A Dynamically Weighted Fuzzy C-means Algorithm. Optical Engineering, 50(10): 1-9.