Scientific Research and Essays Vol. 5(16), pp. 2288-2294, 18 August, 2010
Available online at http://www.academicjournals.org/SRE
ISSN 1992-2248 ©2010 Academic Journals

Full Length Research Paper

A digital robust image watermarking against desynchronization attacks

Oğuz Fındık*, İsmail Babaoğlu and Erkan Ülker

Department of Computer Engineering, Selcuk University, Konya, Turkey.

Accepted 29 April, 2010

In this paper, a robust image watermarking technique using support vector regression (SVR) and particle swarm optimization is introduced to protect intellectual property rights of the gray images in discrete cosine transform domain against a variety of desynchronization attacks. After the division of the original image to 8 × 8 non-overlapping blocks, frequency coefficients of each block are found using discrete cosine transform. Positions of the inputs and output, among the low frequency coefficients which have the significant characteristics of the image, which are used to train SVR are obtained by using particle swarm optimization technique. After SVR is trained using the obtained positions of the inputs and output, watermark embedding and extracting processes are implemented using the trained SVR. Experiments implemented using the optimized coefficients selected among low frequency coefficients show that our watermarking technique has better watermark extracting success after the desynchronization attacks.

Key words: Robust image watermarking, support vector regression, particle swarm optimization.

INTRODUCTION

With the increasingly usage of digital media, the protection of intellectual property rights problem has become even more important. Digital watermarking is one of the most important techniques that are used to protect property rights for digital media like image, video and text etc. (Bhatnagar et al., 2009; Podilchuk et al., 2001). According to the applied domains, digital water-marking can be separated into two groups as spatial and frequency domain watermarking. In spatial domain watermarking, watermark is embedded using the image's pixel values. Watermark embedding and extracting is easy, although this technique is not robust after image processing attacks as compared with the frequency domain watermarking (Wang et al., 2008; Aslantas, 2008; Podilchuk et al., 2001). On the other hand, in frequency domain watermarking, watermark is embedded using the frequency coefficients of the image. Watermark embed-ding and extracting is complex but this technique is robust against image processing attacks Aslantas et al., 2009; Shih et al., 2005). Lu et al., (2006) have introduced a robust image watermarking technique against desynchronization attacks using artificial neural network in discrete cosine transform (DCT) domain. Low DCT frequency coefficients have been selected to obtain the embedded watermark better after the attacks.

Shen et al. (2005) have suggested a color image watermarking technique utilizing support vector regression as a machine learning technique in spatial domain. After the blue channel of the original image has been divided into 3 × 3 non-overlapping blocks, pixel locations have been selected for every sub blocks. Pixel value which the watermark would be embedded has been selected as the output value and the rest of the pixel values have been selected as the input values. Considering these values, support vector regression (SVR) has been trained and the process of watermark embedding has been performed.

Usman et al. (2010) have utilized genetic algorithm to obtain the frequency coefficients to which the watermark would be added on DCT domain. They have increased the success of the watermark extraction process by implementing BCH coding method on their algorithm.

Lin et al. (2010) have proposed a novel watermarking algorithm in DCT domain. The authors have proposed their method against JPEG compression attacks, and they adjust the DCT low-frequency coefficients by the concept of mathematical remainder.

*Corresponding author. E-mail: oguzf@selcuk.edu.tr. Tel: +90 332 223 2128. Fax: +90 332 241 06 35.

Wang et al. (2008) have proposed a new and robust image watermarking technique using SVR and local image characteristic against desynchronization attacks. Training samples have been obtained by calculating sum of the some randomly selected pixel values and the variances with their neighbors. Embedded watermark has been obtained by using the trained SVR. Image processing attacks can be categorized into common signal processing, such as JPEG compression, blurring, sharpening, noise addition, etc and desynchronization attacks such as rotation, scaling, etc. The aim of this study is to increase the extracting success of watermark against desynchronization attacks in the DCT domain. SVR and low frequency coefficients (AC coefficients) in DCT are used for both watermark embedding and watermark extracting. In order to increase extracting success of watermark from the watermarked image, the positions of the AC coefficients which will be used in watermarking process are found by optimizing with particle swarm optimization (PSO). The results of this study have been compared with Lu et al. (2006) study in which artificial neural network and fixed AC coefficients were used against desynchronization attacks.

MATERIALS AND METHODS

Support vector regression

Support vector machine (SVM) is supervised and unsupervised classification algorithm based on statistical learning theory. SVM were firstly introduced by Vapnik (1995). As shown in basic concept depicted in (Figure 1), SVM is a method utilized to classify samples belonging to two classes. Assuming that circles (○) belong to a class and squares (□) belong to the other class, the main purpose of SVM is to find the optimum hyperplane within hyperplanes which separate classes from each other. Optimum hyperplane is the hyperplane that maximizes the distance to the samples of both classes.

Support vector regression (SVR) can be expressed as a submodel of SVM in regression learning area. Assuming that inputs are accepted as x_i and outputs are accepted as y_i, the purpose of SVR training stage is to find optimum $f(x)$ regression function. That the difference between output value and expected value is smaller than ε is aimed at the optimum $f(x)$ function and $f(x)$ function can be expressed as follows:

$$f(x) = \omega.x + b \tag{1}$$

Where
w ∈ x
ab ∈ R.

Taking an optimization problem, finding the optimum f(x) function can be transformed as follows:

$$\min \frac{1}{2}\| \omega \|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*),$$
$$s.t.\begin{cases} y_i - \omega.x_i - b \le \varepsilon + \xi_i, \\ \omega.x_i + b - y_i \le \varepsilon + \xi_i, \end{cases} \quad i = 1,......., n \tag{2}$$

Where
$\xi_i \ge 0$ and $\xi_i^* \ge 0$ = positive slack variables denotes the regression error
C = penalty parameter which controls the trade-off between errors of the SVM on training data and margin maximization.

After kernel substitution, the dual objective function is:

$$J(\alpha,\hat{\alpha}) = \sum_{i=1}^{m} y_i(\alpha_i - \hat{\alpha}_i) - \varepsilon\sum_{i=1}^{m}(\alpha_i + \hat{\alpha}_i) - \frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j)K(x_i, x_j) \tag{3}$$

Which is maximized subject to

$$\sum_{i=1}^{m}\alpha_i = \sum_{i=1}^{m}\hat{\alpha}_i \;\; and \;\; 0 \le \alpha_i \le C,\, 0 \le \hat{\alpha}_i \le C. \tag{4}$$

Where
$i = 1,..., m$; α_i
$\hat{\alpha}_i$ =Lagrange multiplying factors
C = penalty parameter, the regression function is:

$$f(x) = (w.x) + b = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)(x_i - x) + b^* \tag{5}$$

where
α_i and α_i^* = only few non-zeros which corresponding samples are namely called support vectors
b^* = a scalar which determines the position of the separating hyperplane (Wang et al., (2008).

More detailed explanations and formulations of SVR can be reached from Tsai et al. (2007), Çomak et al. (2008) and Chen et al. (2007).

Particle swarm optimization

Particle swarm optimization developed by inspiring from bird flocking and fish schooling is an optimization technique introduced by Kenedy and Eberhart (1995). It was firstly used for optimization of the weights in the back propagation algorithm. Because it is a fast and effective algorithm, it have been used in many application fields till today (Sun, 2009).

Each randomly created initial solution within the solution space is called as a particle and the set composed of the particles having position X and velocity V is called as the swarm. The success of each particle is determined by a fitness function going to be used in the application. The best solutions obtained from each particle are saved as *localbest (i,j)* and the best solution of the swarm is saved as *globalbest*. The velocity and position of the particles are calculated as follows:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1R_1(p_{best_{i,j}} - x_{i,j}(t)) + c_2R_2(g_{best_{i,j}} - x_{i,j}(t)) \tag{6}$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \tag{7}$$

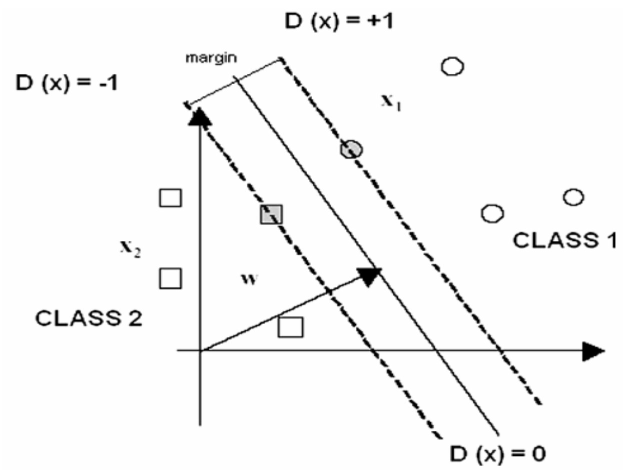

Figure 1. The basic concept of SVM.

where
i = the index of the particle
j = index of the position in particle
t = iteration number
$v_{i,j}(t)$ = velocity of the *i*th particle in the swarm on *j*th index of the position in the particle
$x_{i,j}(t)$ = position. R_1 and R_2 are the random numbers uniformly distributed between 0 and 1
c_1 and c_2 = acceleration numbers
w = inertia weight.

The PSO algorithm pseudo code is illustrated briefly as follows:

1. Initialization
For each particle
  Create a particle randomly within the *j*-dimensional solution space
  Evaluate the particle using fitness function
  Update *localbest* and *globalbest*
End

2. Overall optimization process
  During the iteration process or until the sopping criteria reached
  Update the inertial weight
For each particle
  Update the particles using Equations (6) and (7)
  Update *localbest* and *globalbest* if necessary
  End
End

Inertial weight is embedded to the original PSO algorithm by Shi and Eberhart (1998a, b). It is used to avoid PSO's attachment to local or global minimum. In other words, it is used to provide the balance between local and global search area. While a higher inertial weight value increases the search ability within the global area, a smaller inertial weight value increases the search ability within the local area (Shi, 2004). In this study, inertial weight value is used having a linear decrease from high to low using the following formula:

$$w = \frac{t_{\max} - t}{t_{\max}} \tag{8}$$

Where

w = inertial weight
t =current iteration number
$t_{\max}$ = maximum iteration number.

It can be seen from the equation that the inertial weight w will be [0,99]. The constants c_1 and c_2 are the acceleration numbers.

These constants are learning factors which control the influence of $p_{best_{i,j}}$ and $g_{best_{i,j}}$ on the search process. Earlier introduced studies on particle swarm optimization in literature are showed that acceleration constants c_1 and c_2 are both equal to 2 for almost all applications. Thus, c_1 and c_2 are used being equal to 2 in this study.

Proposed algorithm

In the proposed method, watermark embedding and extracting processes are carried out using the DCT coefficients obtained from grayscale images by SVR method. To extract the watermark most successfully, the optimum positions of DCT coefficients which are used to embed and extract the watermark is found using PSO.

Watermark embedding

An original gray image I can be defined by

$$I = [I_p]_{mxn} \tag{9}$$

Where mxn is the size of the image, I_p stands for the pixel located at position $p_{(i,j)}$ over the image I. Here, $i \in \{0,1,2,...,m-1\}$ and $j \in \{0,1,2,...,n-1\}$. Similar to representation of the original image, binary watermark image W can be represented by:

$$W = [W_k]_{pxq} \tag{10}$$

Where pxq is the size of the watermark, W_k stands for the pixel located at position $k_{(i,j)}$ over the watermark image W. Here, $i \in \{0,1,2,..., p-1\}$ and $j \in \{0,1,2,...,q-1\}$. Watermark image W can be transferred into pxq sized bit sequence as

$$W = w_0, w_1, w_2, ..., w_{pxq-1} \tag{11}$$

To transfer Original gray image into DCT coefficients, I is divided into 8 × 8 non-overlapping sub-images given as:

$$I = \bigcup_{m=1}^{m/8} \bigcup_{n=1}^{n/8} I_{(m,n)} \tag{12}$$

$$D = DCT\,(I_{(m.n)}) \tag{13}$$

DC	1	5	6	14	15	27	28
2	4	7	13	16	26	29	
3	8	12	17	25	30		
9	11	18	24	31			
10	19	23	32				
20	22	33					
21	34						
35							

Figure 2. Index of the AC components.

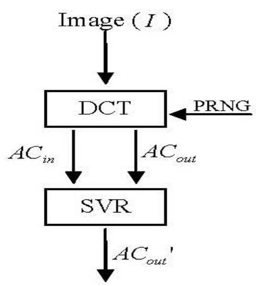

Figure 3. SVR training.

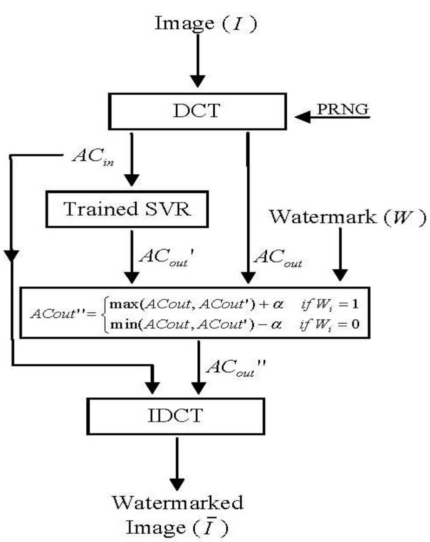

Figure 4. Watermark embedding.

Where

D is the transferred form of DCT coefficients.

Embedding process is implemented by the help of a network produced by using SVR. The *n*-dimensional input vector which is used in this process is selected from the AC coefficients given in (Figure 2). One of the AC coefficients which is not included in the input vector is selected as the output value. To provide the security of the embedded watermark, the blocks to which the watermark is going to be added are selected using Rabin's public key;

$$AC_{out} = \eta,\ \eta \in \{AC_1, AC_2, .., AC_{35}\}\ \&\ \eta \notin AC_{in}, \tag{15}$$

cryptosystem (PRNG) (Rabin, 1978). SVR is trained using the input vector and output value obtained from the selected blocks. Index of the AC components in *D* can be illustrated in (Figure 2). AC_{in} and AC_{out} used as the input vector and output value can be defined as follows:

$$AC_{in} = \{\delta(1), \delta(2), \ldots, \delta(n)\},\ \delta \in \{AC_1, AC_2, .., AC_{35}\},\ 1<n<=35, \tag{14}$$

Where

AC_{in} stands for the input vector

AC_{out} stands for the output which will be used to train the SVR.

SVR is trained using input vector and expected output value. An output value is obtained using the input vectors and the trained SVR. Watermark embedding process is implemented by using the output value obtained from the trained SVR and the expected output value which is used for training SVR and it is formulized as follows:

$$ACout''=\begin{cases}\max(ACout, ACout')+\alpha & if\ W_i=1\\ \min(ACout, ACout')-\alpha & if\ W_i=0\end{cases} \tag{16}$$

where

AC_{out} = the expected AC coefficient value used as output,

AC_{out}' = the AC coefficient obtained using the trained SVR,

AC_{out}'' = the AC coefficient for watermark embedded image

α = the positive constant that determines the watermark strength.

After watermark embedding, watermarked image is obtained by implementing inverse DCT; $\bar{I} = IDCT(I_{(m.n)})$ (17). SVR training and watermark embedding stages of the proposed watermarking algorithm are shown in (Figures 3 and 4).

Watermark extracting

After the embedding procedure, the watermark can be acquired from corresponding watermarked image denoted as:

$$\bar{I} = [\bar{I}_p]_{mxn} \tag{18}$$

Where

mxn = the size of the watermarked image, $\bar{I}_p$ stands for the pixel located at position $p_{(i,j)}$ over the watermarked image $\bar{I}$.

Here, $i \in \{0,1,2,\ldots,m-1\}$ and $j \in \{0,1,2,\ldots,n-1\}$.

Watermarked image which is divided into 8 × 8 non-overlapping blocks can be shown as:

$$\bar{I} = \bigcup_{m=1}^{m/8}\bigcup_{n=1}^{n/8}\bar{I}_{(m,n)} \tag{19}$$

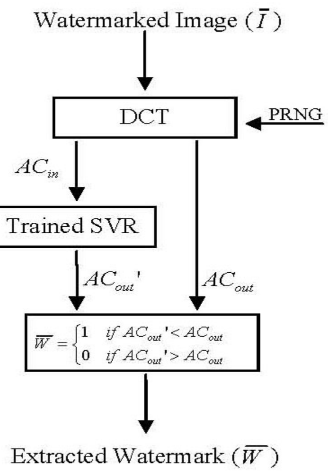

Figure 5. Watermark extracting.

Afterwards, each sub-image is transferred into DCT domain as:

$$\overline{D} = DCT(\bar{I}_{(m.n)}) \tag{20}$$

Where

$\overline{D}$ = the form of transformation of watermarked image to DCT coefficients.

Similar as in the embedding process, AC_{in} vector and AC_{out} value are obtained from the coefficients $\overline{D}$. AC_{out}' output value is obtained using the obtained AC_{in} vector and the trained SVR. Using the expected AC_{out} output value and AC_{out}' output value, watermark is obtained utilizing the following formula:

$$\overline{W} = \begin{cases} 1 & if\ AC_{out}' < AC_{out} \\ 0 & if\ AC_{out}' > AC_{out} \end{cases} \tag{21}$$

Where

AC_{out}' = output value obtained using SVR

AC_{out} = the AC coefficient previously used to train SVR.

Watermark is obtained by repeating this process for the whole blocks which are intended to be watermark added. Watermark extraction stage of the proposed watermarking algorithm can be diagrammatized in (Figure 5).

PSO optimization

Low frequency DCT coefficients are the most important frequency coefficients of a DCT transformed image. These frequency coefficients are used in the most of the watermarking studies implemented in DCT domain (Hwang et al., 2000). But, it is observed that the change of these coefficients may occur after desynchronization attacks. Optimizing positions of AC_{in} vector and AC_{out} value through low-frequency coefficients for the purpose of the most successful watermark extracting after desynchronization attacks was aimed for the whole images. In another word, the positions of AC_{in} vector and AC_{out} value which were found by PSO indicate same locations for whole images which the watermark will be embedded. In this study, two PSO is used one within the other. The first PSO optimizes AC_{in} vector and the second PSO optimizes AC_{out} value according to obtained AC_{in} vector optimized by first PSO. The fitness function which is used to update the particles for both PSO processes maximizes the average of the watermark extracting successes after desynchronization attacks. The fitness function can be formulized as follows:

$$BCR(W,\overline{W}) \equiv \left(1 - \frac{\sum_{i=1}^{pxq}\left(W_i \oplus \overline{W}_i\right)}{pxq}\right) x100\%\ , \tag{22}$$

$$Fit = \frac{1}{A\,x\,N}\sum_{j=1}^{N}\sum_{i=1}^{A} BCR_{i,j}\ , \tag{23}$$

Where

Bit correct ratio (BCR) = the success of the extraction of the watermark process

W = the digital signature

$\overline{W}$ = the extracted signature

pxq = the length of the signature,

Fit = the fitness function

A = the number of the attacks

N = the number of the images.

The generalized form of the proposed watermarking algorithm which obtains the optimum positions of the AC_{in} vector and AC_{out} value utilizing PSO is shown in (Figure 6).

RESULTS AND DISCUSSION

The peak signal to noise ratio (PSNR) and the BCR values are used to qualify the watermarked image and the extracted watermark respectively. In watermarking concept the PSNR is calculated as follows:

$$MSE(I,\bar{I}) = \frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(Ip_{(i,j)} - \bar{I}p_{(i,j)}\right)^2$$

$$PSNR(I,\bar{I}) = 10 \times \log\left(\frac{255^2}{MSE(I,\bar{I})}\right) \tag{24}$$

Where

MSE = the mean squared error

I = the original image

$\bar{I}$ = the watermarked image.

The binary digital signature and the original images Lena

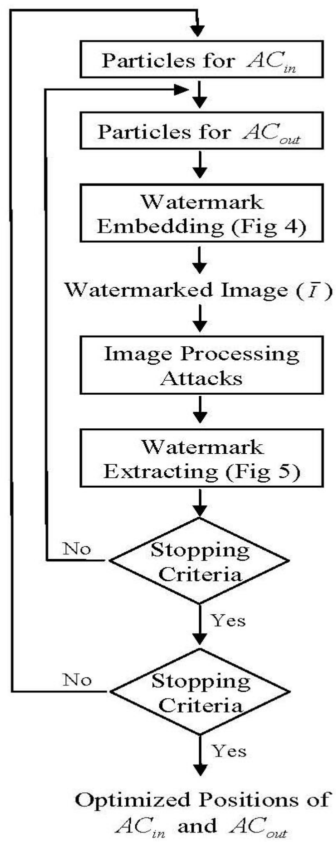

Figure 6. PSO optimization for AC_{in} vector and AC_{out} value.

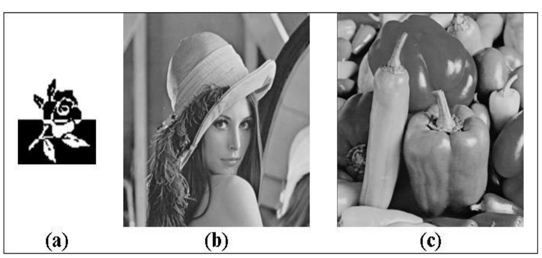

Figure 7. (a) The binary digital signature rabbit, (b) the original Lena image, (c) the original Peppers image.

and Peppers are given in the (Figure 7). The binary digital signature is used with size 32 × 32 and both the original images Lena and Peppers are used with the size 512 × 512. In embedding process, the strength parameter α is taken as 20. In this study, both PSO's, one used for obtaining the optimum AC_{in} vector and the other is user for obtaining the optimum AC_{out} value, are examined for 300, 250, 200, 150, 100, 50 and 25 iterations. For each iteration the sizes of the particles are also examined for

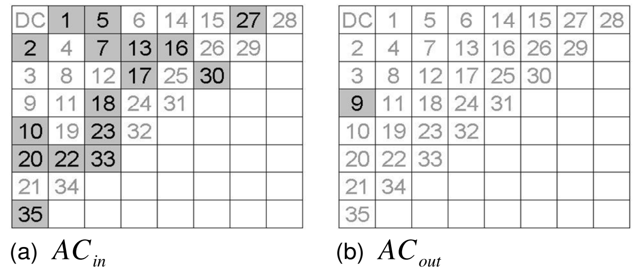

Figure 8. (a) Optimum AC_{in} values and (b) AC_{out} value.

5, 10, 20, 30, 40 and 50. The inertia weight (w) is used according to the Equation 8 and c_1 and c_2 are used being both equal to 2 mentioned above for both PSO's. LIBSVM package (Chang, 2001) is used to train SVR in this study. To find the best trained SVR, 0.5, 0.001 and 1 values have been computed for nu, ε and C training parameters respectively. The divided DCT coefficients are scanned in Zig-Zag order. First the low frequency part, finally the high frequency part. Attacks like jpeg attacks modify the high frequency coefficients of the image. To obtain embedded watermark the most successfully, low frequency DCT coefficients between AC_1 and AC_{35} are used in this study. The optimum AC_{in} vector and AC_{out} value is obtained from the proposed algorithm are given as follows and illustrated in (Figure 8);

AC_{in} = [1,2,5,7,10,13,16,17,18,20,22,23,27,28,30,32,35];

AC_{out} = [9]

Results of this study are compared with Lu et al. (2006) study results. The same desynchronization image attacks are used for the comparison. These attacks are rotation 10° and 45°, resizing 60% and 130%, rotation 20° and resizing 80%, JPEG (80%) and rotation 60° and JPEG (80%) and resizing 120%. The comparison table is given in (Table 1) and watermarked images and extracted watermarks are given in (Figure 9). Table 1 shows the BCR comparison results of the SVR model using fixed AC positions, the proposed method and the Lu et al. (2006) method and fixed AC positions denote the positions of the AC coefficients used in the study. As a statistical method, SVR is successful on making a linear relation between input vector and output value. When BCR results which are obtained from SVR model using fixed AC positions compared to Lu et al. (2006) BCR results, it can be seen that BCR results which are obtained from SVR model using fixed AC positions are 3 - 10% better than Lu et al. (2006) BCR results. In addition to this, BCR results which are obtained from proposed method are 1 - 2% better than BCR results which are obtained from SVR model using fixed AC positions. By

Table 1. Results of the study.

Attacks	Lena			Pepper		
	SVR model using fixed AC positions	Proposed Method	Lu's Method	SVR model using fixed AC positions	Proposed Method	Lu's Method
Rotation 10°	95.89	96.29	92.01	95.48	96.38	91.08
Rotation 45°	90.03	91.30	93.38	88.42	90.23	91.95
Resizing 60%	94.62	95.11	89.17	90.55	91.40	87.40
Resizing 130%	97.12	98.33	93.09	97.21	98.24	90.83
Rotation 20° and Resizing 80%	91.87	93.94	87.73	91.94	93.35	87.01
JPEG (80%) and Rotation 60°	89.60	90.42	88.54	89.40	90.33	85.93
JPEG (80%) and Resizing 120%	99.05	99.12	89.83	99.02	99.02	87.72

optimizing the AC positions to which the watermark would be embedded using PSO, the success of the watermark extraction process has increased as expected. In watermarking concept on DCT domain, the AC positions can be also implemented by using genetic algorithms or other evolutionary algorithms as a future work.

Conclusion

In this study, a robust image watermarking scheme is suggested in DCT domain using SVR classification method for grayscale images. The increment on the success of the watermarking process is observed when SVR classification method is used on DCT domain. At the same time, the results have become better by optimizing the AC positions which is used in the watermarking process. The result of this study is compared with the results of Lu et al.'s study (2006) and it's shown that this study has better results. As a result, the SVR classification method used on DCT domain increase the success of the watermarking process and also it is shown that the success of the watermarking process using the PSO optimization method has been increased.

REFERENCES

Aslantas V (2008). A singular-value decomposition-based image watermarking using genetic algorithm. Int. J. Elect. Commun., (AEÜ). 62: 386 – 394.
Aslantas V, Ozer S, Ozturk S (2009). Improving the performance of DCT-based fragile watermarking using intelligent optimization algorithms. Opt. Commun. 282: 2806–2817.
Bhatnagar G, Raman B (2009). A new robust reference watermarking scheme based on DWT-SVD. Comput. Standards Interfaces, 31: 1002–1013.
Chang CC, Lin CJ (2001). LIBSVM: a library for support vector machines, Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm.
Chen KY, Wang CH (2007). A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan. Expert Syst. Appl. 32(1): 254–264.
Çomak E, Arslan A (2008). A new training method for support vector machines: Clustering k-NN support vector machines. Expert Syst Appl. 35(3): 564-568.
Hwang MS, Chang CC, Hwang KF (2000). Digital watermarking of images using neural networks. J. Electron Imaging. 9(4): 548–555.
Kennedy J, Eberhart RC (1995). Particle swarm optimization. In: Proc. IEEE Int. Conf. Neural Networks. 4: 1942-1948.
Lin SD, Shie S-T, Guo JY (2010). Improving the robustness of DCT-based image watermarking against JPEG compression. Computer Standards & Interfaces. 32: 54-60.
Lu W, Lu H, Chung FL (2006). Robust Image Watermarking Using RBF Neural Network. Lect Notes Comput Sci. 3972: 623–628.
Podilchuk CI, Delp EJ (2001). Digital watermarking: Algorithms and applications. IEEE Signal Process Mag.18: 33-46.
Rabin MO (1978). Digital signatures in Foundations of Secure Communication. Academic Press, New York, pp. 155–168.
Shen R, Fu Y, Lu H (2005). A novel image watermarking scheme based on support vector regression. J. Syst. Software. 78: 1–8.
Shi Y (2004). Particle Swarm Optimization. IEEE Neural Networks Society. pp. 8-13.
Shi Y, Eberhart RC (1998). Parameter selection in particle swarm optimization. In: Proceedings. Annual Conferen. Evolut. Comput. pp. 591-600.
Shi Y, Eberhart RC (1998). A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation; Piscataway, NJ: IEEE Press. pp. 69-73.
Shih FY, Wu YT (2005). Enhancement of image watermark retrieval based on genetic algorithms. J. Visual Commun. Image Represent. 16: 115–133.
Sun TH (2009). Applying particle swarm optimization algorithm to roundness measurement. Expert Syst. Appl., 36: 3428-3438.
Tsai HH, Sun DW (2007). Color image watermark extraction based on support vector machines. Inf. Sci., 177: 550-569.
Usman I, Khan A (2010). BCH coding and intelligent watermark embedding: Employing both frequency and strength selection. Appl. Soft. Comput. 10: 332-343.
Vapnik V (1995). The nature of statistical learning theory. New York: Springer.
Wang XY, Yang HY, Cui CY (2008). An SVM-based robust digital image watermarking against desynchronization attacks. Signal Processing. 88: 2193– 2205.
Wang XY, Cui CY (2008). A novel image watermarking scheme against desynchronization attacks by SVR revision. J. Visual Commun. Image Represent. 19: 334–342.