Digital Watermarking of Color Images Using Support Vector Machines

Pao-Ta Yu
Department of Computer Science and Information Engineering, National Chung Cheng University
Email: csipty@ccunix.ccu.edu.tw

Hung-Hsu Tsai
Department of Information Management, National Huwei Institute of Technology
Email: thh@sunws.nhit.edu.tw

Duen-Wu Sun
Graduate Institute of Communication Engineering, National Chung Cheng University
Email: sdw90@comm.ccu.edu.tw

Abstract

In this paper, a novel digital watermarking technique based on support vector machines (SVMs) is proposed to protect copyrights for color images. The proposed technique can be called SVM-based Color Image Watermarking (SCIW). The SVM can be trained for an optimal hyperplane from the given training patterns composed of the preamble information and the corresponding watermarked color image. Due to the adaptability of SVMs, the SCIW method is developed to retrieve the user's signature. During the procedure of watermark retrieving, the user's signature can be retrieved directly from a watermarked color image without information on the corresponding original color image. Numerous experimental results are shown to prove that the SCIW method possesses robustness and generalization against common image manipulations.

Keywords: Digital watermarking; Support vector machines; Color images

一、 Introduction

In recent years, the transmission speed in data interchange on computer networking has been improved quite substantially. With the popularity of computers, the Internet especially for World Wide Web environment has prevailed over the whole world. Various formats of multimedia data can be rapidly distributed over the Internet as one pleases. Moreover, the DVD/CD recorders and the disk storages are much inexpensive with larger capacity and rapider speed. Consequently, the unauthorized or illegal duplication of multimedia data is encouraged easily by using the recorders and the facilities of computer networking. This results in the damage for the intelligent properties of multimedia data. Accordingly, the major concern is now the copyright assertion of multimedia data [1]. Digital watermarking is proposed to provide a protecting scheme of multimedia data so that users can directly access the protected multimedia data. Because color images are ubiquitous in various applications of multimedia systems and are utilized in MPEG video, the watermarking techniques of color images are still important for these motivations.

Kutter *et al.* utilized a pseudo-random number generator to locate the positions where the watermark is embedded [2]. The watermark can be retrieved using amplitude modulation. Yu *et al.* proposed a method based on neural networks, which can be trained to recover the watermark [3].

In Section 2, the concept of SVMs will be introduced. In Section 3, the embedding and retrieving algorithms of the SCIW method will be described. In Section 4, various experimental results will be displayed. In Section 5, at last, the conclusion of this paper will be stated.

二、 The Classification Mechanisms of SVMs

(一) SVMs with Linearly Separable Patterns

Let a training pattern set $\mathbf{F}$ including N training patterns be represented by

$$\mathbf{F} = \{(\mathbf{x}_i, d_i)\}_{i=1}^{N}, \quad (1)$$

where $\mathbf{x}_i \in \Re^n$ stands for an input vector and $d_i \in \{+1, -1\}$ is the desired output [4]. Suppose that the equation of a separable hyperplane H for $\mathbf{F}$ is represented by

$$\mathrm{H}: \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b = 0, \quad (2)$$

where $\mathbf{w}$ is an adjustable weight vector and b is a bias. Furthermore, a pair of hyperplanes H_1 and H_2, paralleling to the hyperplane H, can be defined by

$$\begin{cases} \mathrm{H}_1 : \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b = +1, \\ \mathrm{H}_2 : \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b = -1. \end{cases} \tag{3}$$

The purpose of using a support vector machine is to obtain an optimal hyperplane H for which the margin between H_1 and H_2 is maximized. The margin between H_1 and H_2 can be expressed by

$$r = \frac{1}{\|\mathbf{w}\|} + \frac{1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}. \tag{4}$$

This problem for solving out the optimal hyperplane can be substituted by using the Lagrange method as follows.

Minimize the cost function

$$C(\mathbf{w}) = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle, \tag{5}$$

subject to the constraints

$$d_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 \ge 0 \quad \text{for } i = 1, 2, 3, \cdots, N \cdot \tag{6}$$

Then, the Lagrangian function can be constructed by

$$J(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^{N} a_i [d_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1], \tag{7}$$

where a_i, for $i = 1, 2, 3, \cdots, N$, are called Lagrange multipliers. Let the gradients of $J(\mathbf{w}, b, \mathbf{a})$ with respect to $\mathbf{w}$ and b equal to zero. The conditions can be acquired as follows.

$$\frac{\partial J(\mathbf{w}, b, \mathbf{a})}{\partial \mathbf{w}} = \mathbf{0} \Rightarrow \mathbf{w} = \sum_{i=1}^{N} a_i d_i \mathbf{x}_i \tag{8}$$

$$\frac{\partial J(\mathbf{w}, b, \mathbf{a})}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} a_i d_i = 0 \tag{9}$$

From (7), (8), and (9), the dual problem can be stated as follows.

Maximize the Lagrangian function

$$J_D(\mathbf{a}) = \sum_{i=1}^{N} \mathbf{a}_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j d_i d_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle, \tag{10}$$

subject to the constraints

$$\begin{cases} \sum_{i=1}^{N} a_i d_i = 0, \\ a_i \ge 0, \text{for } i = 1, 2, 3, \cdots, N. \end{cases} \tag{11}$$

By the Karush-Kuhn-Tucker complementary conditions [5], the condition can be formulated by

$$a_i [d_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1] = 0, \text{ for } i = 1, 2, 3, \cdots, N \tag{12}$$

Note that the input vectors $\mathbf{x}_i$ in $\{x_1, x_2, x_3, \cdots, x_N\}$ satisfied in (12) are called support vectors on the hyperplane H_1 or H_2. To identify this, the letter $\mathbf{x}$ will be replaced by $\mathbf{s}$. Then the support vectors can be denoted by $\mathbf{s}_k$, for $k = 1, 2, 3, \cdots, N_S$, where N_S is the number of support vectors. Once the optimal hyperplane $\mathrm{H} : \mathbf{w}_O \cdot \mathbf{x} + b_O = 0$ has been determined, the decision function f can be defined by

$$f(\mathbf{x}) = \mathrm{sgn}(\sum_{i=1}^{N} a_{Oi} d_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b_O). \tag{13}$$

(二) SVMs with Linearly Non-Separable Patterns

To overcome this non-separable case, the slack variables x_i, for $i = 1, 2, 3, \cdots, N$, can be recommended as to loosen constraints. The optimal hyperplane H has to be subject to the constraints

$$d_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 + x_i \ge 0 \quad i = 1, 2, 3, \cdots, N, \tag{14}$$

and, the cost function $C(\mathbf{w})$ to be minimized is also changed from (5) to

$$C(\mathbf{w}) = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle + C\sum_{i=1}^{N} x_i \,. \tag{15}$$

The Lagrangian function can be constructed by

$$\begin{aligned} J(\mathbf{w}, b, \mathbf{?}, \mathbf{a}, \text{ß}) &= \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle + C\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} a_i [d_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 + x_i] - \sum_{i=1}^{N} b_i x_i \\ &= \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle + \sum_{i=1}^{N} (C - a_i - b_i) x_i - \sum_{i=1}^{N} a_i [d_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1], \end{aligned} \tag{16}$$

where **a** and **ß** are Lagrange multipliers. Then, the dual problem can be reformulated as follows.

Maximize the Lagrangian function

$$, \quad J_D(\mathbf{a}) = \sum_{i=1}^{N} \mathbf{a}_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j d_i d_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \tag{17}$$

subject to the constraints

$$\begin{cases} \sum_{i=1}^{N} a_i d_i = 0, \\ 0 \le a_i \le C, \text{ for } i = 1, 2, 3, \cdots, N. \end{cases} \tag{18}$$

The only difference between the non-separable and separable condition is the constraints of the Lagrange multipliers a_i. The constraints become severer with the user-defined upper boundary C. Let the gradients of (16) with respect to **?** equal to zero. The equation can be reformulated as

$$C - a_i - b_i = 0 \text{ or } C = a_i + b_i. \tag{19}$$

Because of $b_i \ge 0$ and $a_i \ge 0$, for $i = 1, 2, 3, \cdots, N$, the constraints $0 \le a_i \le C$ can be yielded.

(三) Nonlinear SVMs with Kernel Function

Nonlinear SVMs are proposed to correctly classify the training patterns. An important procedure in nonlinear SVMs is to select a nonlinear transformation (mapping) Φ from an input vector $\mathbf{x}_i \in \Re^n$ into a higher dimensional Euclidean space (feature space). Then, the Lagrangian function can be reformulated as follows:

$$J_D(\mathbf{a}) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j d_i d_j \langle \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \rangle \tag{20}$$

$$= \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{21}$$

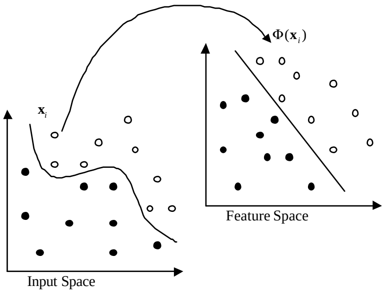


Figure 1: A feature mapping from the input space to feature space.

Note that the training algorithm with respect to $\mathbf{a}$ depends on the form of the dot product $\langle \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)\rangle$ in highly dimensional feature space $\mathbf{E}$. Let a kernel function $?(\mathbf{x}_i,\mathbf{x}_j)$, shown in (21), be equivalent to $\langle \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)\rangle$. The requirement in the training procedure would only be the kernel function $?$ without knowing the explicit function of Φ. The decision function f can be reformulated by

$$f(\mathbf{x}) = \mathrm{sgn}\Big(\sum_{k=1}^{N_S} a_{Ok} d_k\, ?\,(\mathbf{s}_k,\mathbf{x}) + b_{\Phi O}\Big) . \qquad (22)$$

The testing patterns $\mathbf{x} \in \Re^n$ then can be classified by f in feature space.

There are three common inner-product kernels for the types of SVMs summarized in Table 1 [6]. Note that, the characteristics of kernel functions have to satisfy the conditions in Mercer's theorem [7].

Type of support vector machine	Inner product kernel $K(\mathbf{x}_i,\mathbf{x})$
Polynomial learning machine	$(\langle \mathbf{x}_i\cdot\mathbf{x}\rangle + 1)^P$
Radial-basis function network	$\exp\left(\frac{-1}{2s^2}\|\mathbf{x}_i-\mathbf{x}\|^2\right)$
Two-layer perception	$\tanh(b_0\langle \mathbf{x}_i\cdot\mathbf{x}\rangle + b_1)$

Table 1: A summary of inner-product kernels.

三、 The SCIW Method

For convenience, let $O=[o_p]_{L\times K}$ denote an original RGB color image with width K and height L, where $o_p=(R_p,G_p,B_p)$ stands for a RGB vector for each component $R_p,G_p,B_p \in \{0,1,2,\cdots,255\}$. The subscript $p=(x_1,x_2)$ for R_p, G_p or B_p denotes the pixel position within the color image O, where $x_1 \in \{0,1,2,\cdots,K-1\}$ and $x_2 \in \{0,1,2,\cdots,L-1\}$.

According to Kutter et al.'s algorithm [2], a new method called SVM-based color image watermarking (SCIW) can implement the protection of copyright assertions depending on two main procedures: the embedding algorithm and the retrieval algorithm. These procedures will be introduced in the following sections in turn.

(一) Design of a Watermark

0	0	1	…	1	1	0	1	1	…	0	1

Preamble Information　Owner's Signature

Figure 2: A binary sequence of a watermark.

A watermark W is composed of two binary sequences which is expressed by:

$$\begin{aligned} W &= DS = d_0d_1d_2\cdots d_i\cdots d_{N-1}s_0s_1s_2\cdots s_{M-1} \\ &= w_0w_1w_2\cdots w_j\cdots w_{N+M-1}, \end{aligned} \qquad (23)$$

where $d_i = w_i$, for $i=0,1,2,\cdots,N-1$, and $s_l = w_{N+l}$, for $l=0,1,2,\cdots,M-1$. The first sequence $D=d_0d_1d_2\cdots d_{N-1}$ denotes the preamble information with length N, where $d_i \in \{0,1\}$. Note that, the elements in the preamble information D are taken as the desired outputs of the set of training patterns in (1). Moreover, $S=s_0s_1s_2\cdots s_{M-1}$ represents the owner's digital signature with length M, where $s_0,s_1,s_2,\cdots,s_{M-1}\in\{0,1\}$.

(二) The Embedding Algorithm for SCIW

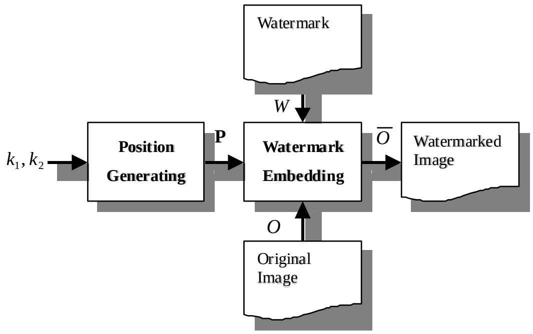


Figure 3: The procedural of the embedding algorithm.

The embedding procedure comprises two major stages: the position generating and the watermark embedding. The algorithm is displayed in Figure 3.

(1) Pseudo-random Positions

For the sake of the security, a pseudo-random number generator (PRNG) is employed to encrypt the information about these positions. In this paper, a quadratic residue generator proposed by Blum et al. [8] is chosen. The generator is also called the Blum-Blum-Shub (BBS) generator. Let a sequence of positions $\mathbf{P}$ for watermark embedding be defined by

$$\mathbf{P}=(p_0, p_1, p_2, \cdots, p_j, \cdots, p_{N+M-1}), \qquad (24)$$

where $p_j=(x_{1j}, x_{2j})$ is the *j*-th pseudo-random position within O. All of the pseudo-random positions p_j can be generated by the following steps.

Step 1: Compute the beginning pseudo-random numbers X_{10} and X_{20} as follows:

$$X_{10}=k_1^2 \bmod R, \text{ and } X_{20}=k_2^2 \bmod R,$$

where *R* is constant. In this case, *R* is equal to the width/length of images.

Step 2: Count the succeeding pseudo-random numbers X_{1j} and X_{2j}, for $j=1, 2, 3, \cdots, N+M-1,$ as follows:

$$X_{1j}=X_{1,j-1}^2 \bmod R, \text{ and } X_{2j}=X_{2,j-1}^2 \bmod R.$$

Step 3: Calculate the pseudo-random positions $p_j=(x_{1j}, x_{2j}),$ for $j=0, 1, 2, \cdots, N+M-1,$ as follows:

$$x_{1j}=X_{1j} \bmod R, \text{ and } x_{2j}=X_{2j} \bmod R. \qquad (25)$$

(2) Watermark Embedding

The watermark W can be embedded into the original color image with the procedure described in Figure 3. The embedding algorithm can be summarized as follows.

Step 1: Generate a pseudo-random position $p_j=(x_{1j}, x_{2j})$ within the original image O, according to (25).

Step 2: Compute the luminance L_{p_j} of o_{p_j} at the position p_j by using the following formula:

$$L_{p_j}=0.299R_{p_j}+0.587G_{p_j}+0.114B_{p_j}. \qquad (26)$$

Step 3: Embed w_j into o_{p_j} by modifying the blue-component B_{p_j} in proportion to the luminance L_{p_j}. That is,

$$\overline{B}_{p_j} \leftarrow B_{p_j}+a(2w_j-1)L_{p_j}, \qquad (27)$$

where a is a positive constant that determines the watermark strength.

Step 4: Repeat Step 1 to Step 3 until all bits w_j, for $j=0, 1, 2, \cdots, N+M-1$, in W are embedded.

After the procedure of watermark embedding stated above, the corresponding watermarked image, denoted as $\overline{O}=[\overline{o}_p]_{L\times K}$ can be obtained, where $\overline{o}_p=(\overline{R}_p, \overline{G}_p, \overline{B}_p)$ and $\overline{R}_p, \overline{G}_p, \overline{B}_p \in \{0, 1, 2, \cdots, 255\}$.

(三)The Retrieval Algorithm for SCIW

The retrieval algorithm comprises three major stages: the position generating, the SVM training, and the watermark retrieving, displayed in Figure 4.

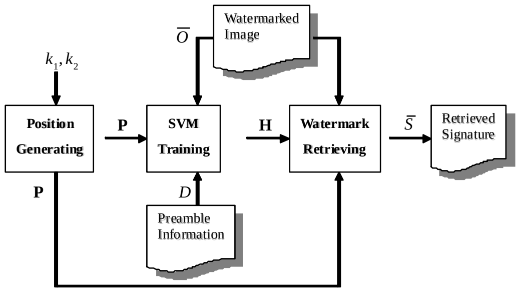


Figure 4: The procedural of the retrieval algorithm.

(1) SVM Training

A set of the training patterns can be constructed by using $\overline{O}$ and D. An optimal hyperplane H can be obtained by training an SVM over a set of training patterns. H can be represented by

$$\mathrm{H}: \mathbf{a}_O \mathbf{V}_i+b_O=0, \qquad (28)$$

where $\mathbf{a}_O$ is an optimal weight vector and b_O is an optimal bias. Let the set of training patterns $\mathbf{F}$ be defined by

$$\mathbf{F}=\{TR_i\}=\{(\mathbf{V}_i, d_i)\}_{i=0}^{N-1}, \qquad (29)$$

where TR_i denotes the i-th training pattern, $\mathbf{V}_i$ denotes an input vector and $\overline{d}_i=d_i$ represents a desired output. In this paper, $\mathbf{V}_i$ can be defined by

$$\mathbf{V}_i=(d^1_{(x_{1i}, x_{2i})}, d^2_{(x_{1i}, x_{2i})}, d^3_{(x_{1i}, x_{2i})})^T, \qquad (30)$$

where $d^k_{p_i=(x_{1i}, x_{2i})}$ is the difference between the value of the blue-component at the central pixel $\overline{B}_{p_i=(x_{1i}, x_{2i})}$ and the predictive-function value $\hat{B}^k_{p_i=(x_{1i}, x_{2i})}$ calculated from the corresponding neighborhood for each k. The first predictive-function value $\hat{B}^1_{p_i=(x_{1i}, x_{2i})}$ is the average of the eight values of the blue-component surrounding the central pixel $\overline{o}_{p_i=(x_{1i},x_{2i})}$ shown in Figure 5, and can be formulated by

$$\hat{B}^1_{p_i=(x_{1i},x_{2i})} = \frac{1}{8}\left(\sum_{k=-1}^{1}\sum_{l=-1}^{1}\overline{B}_{(x_{1i}+l,\,x_{2i}+k)} - \overline{B}_{(x_{1i},\,x_{2i})}\right). \quad (31)$$

$d^1_{p_i=(x_{1i},x_{2i})}$ then can be calculated by

$$d^1_{p_i=(x_{1i},x_{2i})} = \overline{B}_{p_i=(x_{1i},x_{2i})} - \hat{B}^1_{p_i=(x_{1i},x_{2i})}. \quad (32)$$

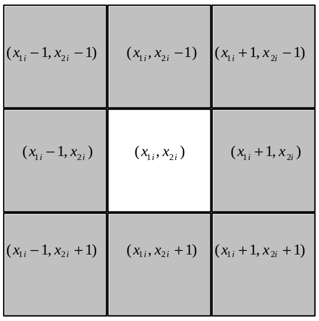


Figure 5: The spanning area with a circle shape to compute $\hat{B}^1_{p_i=(x_{1i},x_{2i})}$.

The second predictive-function value $\hat{B}^2_{p_i=(x_{1i},x_{2i})}$ is shown in Figure 6 by using the cross-shaped neighborhood. $\hat{B}_{p_j}$ can be formulated by

$$\hat{B}^2_{p_i=(x_{1i},x_{2i})} = \frac{1}{4c}\left(\sum_{k=-c}^{c}\overline{B}_{(x_{1i}+k,\,x_{2i})} + \sum_{k=-c}^{c}\overline{B}_{(x_{1i},\,x_{2i}+k)} - 2\overline{B}_{(x_{1i},\,x_{2i})}\right), \quad (33)$$

where c is the size of the sliding window for this cross-shaped neighborhood. The difference $d^2_{p_i=(x_{1i},x_{2i})}$ then can be computed by

$$d^2_{p_i=(x_{1i},x_{2i})} = \overline{B}_{p_i=(x_{1i},x_{2i})} - \hat{B}^2_{p_i=(x_{1i},x_{2i})}. \quad (34)$$

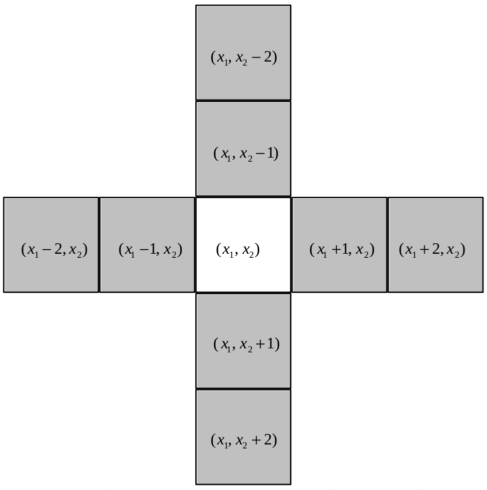


Figure 6: The spanning area with cross shape to compute $\hat{B}^2_{p_i=(x_{1i},x_{2i})}$

Similarly, the last predictive-function value $\hat{B}^3_{p_i=(x_{1i},x_{2i})}$ is the average of the values of the blue-component with another cross-shaped neighborhood shown in Figure 7. Therefore, $\hat{B}^3_{p_i=(x_{1i},x_{2i})}$ can be calculated by

$$\hat{B}^3_{p_i=(x_{1i},x_{2i})} = \frac{1}{4c}\left(\sum_{k=-c}^{c}\overline{B}_{(x_{1i}+k,\,x_{2i}+k)} + \sum_{k=-c}^{c}\overline{B}_{(x_{1i}-k,\,x_{2i}+k)} - 2\overline{B}_{(x_{1i},\,x_{2i})}\right). \quad (35)$$

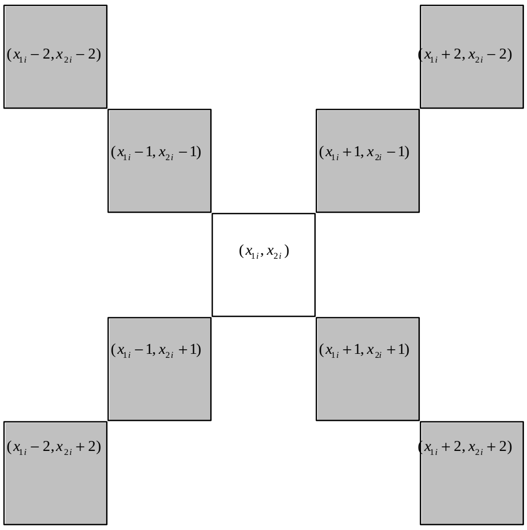


Figure 7: The spanning area with another cross shape to compute $\hat{B}^3_{p_i=(x_{1i},x_{2i})}$.

The difference $d^3_{p_i=(x_{1i},x_{2i})}$ between $\overline{B}_{p_i=(x_{1i},x_{2i})}$ and $\hat{B}^3_{p_i=(x_{1i},x_{2i})}$ can be calculated by

$$d^3_{p_i=(x_{1i},x_{2i})} = \overline{B}_{p_i=(x_{1i},x_{2i})} - \hat{B}^3_{p_i=(x_{1i},x_{2i})}. \quad (36)$$

Consequently, the i-th training pattern TR_i in $\mathbf{F}$ can be rewritten by

$$TR_i = \{((d^1_{(x_{1i},x_{2i})}, d^2_{(x_{1i},x_{2i})}, d^3_{(x_{1i},x_{2i})})^T, d_i)\}. \quad (37)$$

In order to training a SVM, the preamble information D with length N has to be known in the retrieval algorithm, and $d_i \in \{0,1\}$ is generated by PRNG with a seed Λ.

All the training patterns TR_i, $i \in \{0,1,2,\cdots,N-1\}$, will be fed into a support vector machine. Once the training procedure for an SVM over $\mathbf{F}$ has been completed, an optimal hyperplane H with the corresponding optimal weight vector $\mathbf{a}_O$ and the corresponding optimal bias b_O can be determined. The procedure of SVM training can be summarized in the following.

Step1: Compute the pseudo-random position $p_i = (x_{1i}, x_{2i})$ within the watermarked image $\overline{O}$, for the known bits $\overline{d}_i = d_i \in \{0,1\}$, according to (25) in the procedure of position generating. Note that the secret keys k_1 and k_2, and a seed Λ

must the same in the embedding algorithm.

Step2: Calculate the RGB vector $\bar{o}_{p_i}$ at position p_i and at the corresponding neighborhoods shown in Figure 5, 6, and 7, respectively.

Step3: Compute the corresponding predictive-function value $\hat{B}^k_{p_i}$ according to (31), (33), and (35) for computing $d^k_{p_i}$, for $k = 1, 2, 3.$

Step4: Calculate the differences $d^k_{p_i}$, for $k = 1, 2, 3,$ according to (32), (34), and (36) for computing $\mathbf{V}_i$.

Step5: Construct $TR_i = (\mathbf{V}_i, d_i)$ with the corresponding desired output $\bar{d}_i = d_i$.

Step6: Repeat Step 1 to Step 5 until all training patterns TR_i, for $i = 0, 1, 2, \cdots, N-1$, are constructed to form $\mathbf{F}$ in (29).

Step7: Determine the optimal hyperplane H using a SVM.

Figure 8 shows the input vectors $\mathbf{V}_i$ in three-dimensional input space. Note that, “o” denotes the positive training patterns and “x” denotes the negative training patterns.

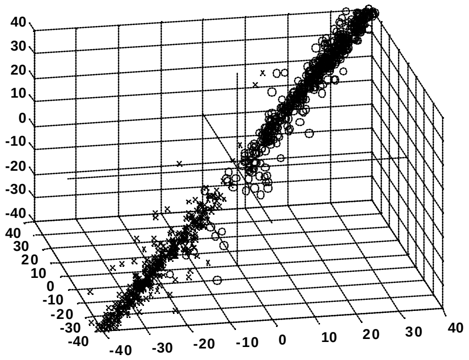


Figure 8: The three-dimensional illustration for the input vectors in input space.

(2) Watermark Retrieving

After the procedure of SVM training, the SVM then will be used to retrieve the owner’s signature S. Similar to Section 3.3.1, the pseudo-random positions $p_j = (x_{1j}, x_{2j}),$ for $j = N, N+1, N+2, \cdots, N+M-1,$ can be continuously computed from the procedure of position generating. Then, the differences $d^k_{p_j}$ at the position p_j for all k can be determined by (31) - (36) to compute $\mathbf{V}_j$. Let $\bar{S} = \bar{s}_0\bar{s}_1\bar{s}_2\cdots\bar{s}_{M-1}$ be the retrieved signature. When an optimal hyperplane H is acquired by using an SVM, a decision function f, defined in (22), can be employed to determine the $\bar{s}_{j-N}$ by

$$\bar{s}_{j-N} = \begin{cases} 1, & \text{if } f(\mathbf{V}_j) = 1, \\ 0, & \text{if } f(\mathbf{V}_j) = -1, \end{cases} \tag{38}$$

where $j = N, N+1, N+2, \cdots, N+M-1$. Finally, the procedure of watermark retrieving can be summarized in the following.

Step1: Compute the succeeding pseudo-random positions $p_j = (x_{1j}, x_{2j})$ within the watermarked image $\bar{O}$, according to (25) in Section 3.2.1.

Step2: Calculate the RGB vector $\bar{o}_{p_j}$ at position p_j and at the corresponding neighborhoods shown in Figure 5, 6, and 7, respectively.

Step3: Compute the corresponding predictive-function value $\hat{B}^k_{p_j}$ according to (31), (33) and (35) for computing $d^k_{p_j}$, for $k = 1, 2, 3.$

Step4: Calculate the differences $d^k_{p_j}$, for $k = 1, 2, 3,$ according to (32), (34) and (36) for computing $\mathbf{V}_j$.

Step5: Determine the value of the bit $\bar{s}_{j-N}$ from (38).

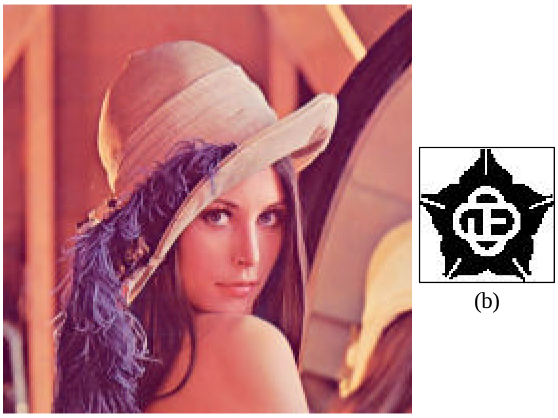
Step6: Repeat Step 1 to Step 5 until all bits $\bar{s}_{j-N}$, for $j = N, N+1, N+2, \cdots, M-1$, in $\bar{S}$ are retrieved.

四、 Experimental Results

The owner’s signature S used in this experiment is the school emblem of CCU in Taiwan shown in Figure 9 (b). The emblem is represented by a 64×64 binary image. Then the owner’s signature S is a bit sequence with length 4096. The preamble information D decided by the owner is a bit sequence with length 1024. Therefore, the embedded watermark W, formed by (23), can be a bit sequence with length 5120 in this experiment.

For the sake of evaluating the performance of various watermarking techniques, some criteria are used to measure the quality of an image and a signature. In this experiment, mean square error (MSE) and peek signal-to-noise ratio (PSNR) are selected to measure the difference between an original image O and a watermarked image $\overline{O}$. The value of MSE between O and $\overline{O}$ can be calculated by

$$\mathrm{MSE}=\frac{\sum_{i=0}^{K-1}\sum_{j=0}^{L-1}\left[(R_{ij}-\overline{R}_{ij})^2+(G_{ij}-\overline{G}_{ij})^2+(B_{ij}-\overline{B}_{ij})^2\right]}{3\times K\times L}, \qquad (39)$$



(a)

(b)

Figure 9: (a) The original “Lenna” color image with size 512×512 .(b) The school emblem of CCU.

where K is the width of a color image and L is the height of a color image. The value of PSNR between O and $\overline{O}$ can be calculated by

$$\mathrm{PSNR}=10\log_{10}\frac{255\times255}{\mathrm{MSE}}. \qquad (40)$$

Mean absolute error (MAE) is selected to measure the difference between an original signature S and a retrieved signature $\overline{S}$. The value of MAE between S and $\overline{S}$ can be computed by

$$\mathrm{MAE}=\frac{\sum_{k=0}^{M-1}\left|s_k-\overline{s}_k\right|}{M}, \qquad (41)$$

where M is the length of the signature. The other necessary parameters involved in this experiment contain the watermark strength $a=0.2$, the size of the cross-shaped neighborhoods $c=2$, and only one copy to a watermark. The MAE values of three methods between the original signature and each of the retrieved signatures are displayed in Table 2.

五、 Conclusion

A novel method for the SCIW technique has been developed in this paper by using a SVM and three image features for a copyright assertion of color images. The SVM is employed to train an optimal hyperplane H for classification by memorizing the relationship between D and $\overline{O}$. By using this kind of training patterns, the performance of the SVM-training can be greatly improved.

Experimental results have been shown to prove that the performance for the SCIW method is better than that for the Kutter *et al.*’s method or that for the Yu *et al.*’s method. Consequently, the optimal hyperplane H trained by an adaptive SVM can correctly retrieve the user’s signature, if the set of training patterns $\mathbf{F}$ is composed of those suitable image features with linear separability after attacks.

Attacks	MSE	PSNR	MAE of the Retrieved Signature		
			Kutter’s method	Yu’s method	SCIW
Attack-free	4.5752	41.5267	0.07543945	0.03125000	0.02197266
JPEG	6.2916	40.1432	0.08203125	0.03222656	0.02343750
Blurring	10.2339	38.0304	0.08715820	0.04956055	0.03076172

Table 2: The experimental results.

六、 Acknowledgments

This paper was in part supported by the National Science Council, R.O.C., under Grant NSC 91-2213-E-150-006.

七、 References

[1] S. C. Cheung and H. Curreem, “Rights protection for digital contents redistribution over the Internet,” *IEEE Proceedings of the 26th Annual International Computer Software and Applications Conference*, pp. 105-110, 26-29 Aug. 2002.

[2] M. Kutter, F. D. Jordan, and F. Bossen, “Digital signature of color images using amplitude modulation,” *SPIE Proceedings of the International Conference on Storage and Retrieval for Image and Video Databases V*, Vol. 3022, pp. 518-526, 13-14 Feb. 1997.

[3] H. H. Tsai, J. S. Cheng, and P. T. Yu, “Audio watermarking based on HAS and neural networks in DCT domain,” *EURASIP Journal on Applied Signal Processing*, Vol. 2003, No. 3, pp. 252-263, 13 Aug. 2002.

[4] J. Chen and L. Jiao, “Classification mechanism of support vector machines,” *IEEE Proceedings of the 5th International Conference on Signal Processing*, Vol. 3, pp. 1556-1559, 21-25 Aug. 2000.

[5] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, pp. 121-167, 1998.

[6] S. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Second Edition, Prentice-Hall, Inc., New Jersey, USA, 1999.

[7] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, United Kingdom, 2000.

[8] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on Computing*, Vol. 15, pp. 364-383, May 1986.