PERCEIVING DIGITAL WATERMARK DETECTION AS IMAGE CLASSIFICATION PROBLEM

P. Then[a] and Y.C. Wang[b]

[a] School of Information Technology and Multimedia,Swinburne University of Technology (Sarawak)
pthen@swinburne.edu.my
Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak
ycwang@fit.unimas.my

Abstract - **Digital watermark detection is treated as classification problem of image processing. For image classification that searches for a butterfly, an image can be classified as positive class that is a butterfly and negative class that is not a butterfly. Similarly, the watermarked and unwatermarked images are perceived as positive and negative class respectively. Hence, Support Vector Machine (SVM) is used as the classifier of watermarked and unwatermarked digital image due to its ability of separating both linearly and non-linearly separable data. Hyperplanes of various detectors are briefly elaborated to show how SVM's hyperplane is suitable for Stirmark attacked watermarked image. Cox's spread spectrum watermarking scheme is used to embed the watermark into digital images. Then, Support Vector Machine is trained with both the watermarked and unwatermarked images. Training SVM eliminates the use of watermark during the detection process. Receiver Operating Characteristics (ROC) graphs are plotted to assess the false positive and false negative probability of both the correlation detector of the watermarking schemes and SVM classifier. Both watermarked and unwatermarked images are later attacked under Stirmark, and then tested on the correlation detector and SVM classifier. Remedies are suggested to preprocess the training data. The optimal setting of SVM parameters is also investigated and determined besides preprocessing. The preprocessing and optimal parameters setting enable the trained SVM to achieve substantially better results than those resulting from the correlation detector.**

Keywords: **Support Vector Machine, Digital Watermark, Receiver Operating Characteristics, Stirmark attacks.**

1. INTRODUCTION

Support Vector Machine (SVM), a universal classification algorithm developed by Vapnik and his colleagues, has been used successfully for many classification tasks (Clark et al. 2004; Vapnik 1995,1998). In this paper we investigate the application of SVM in digital image watermark detection. We chose to look at classifying watermarked and unwatermarked images similar to the classification tasks in image processing. SVM classifiers are therefore developed to classify watermarked and unwatermarked images.

Digital watermark is embedded into digital images as bits of information such as copyright and authorship. These bits of information are embedded into the images by satisfying a series of properties. The properties are effectiveness, fidelity, data pay-load, blind or informed detection, false positive and false negative probability and robustness (Cox et al. 2002). In this paper, we are focusing on the blind detection and investigating the effects that Stirmark attacks can impact on the SVM classifier. The false positive and false negative probability of correlation detector and SVM classifier are compared to study their detection accuracy. The false positive probability is also cross compared with and without Stirmark attacks. Another important property of digital watermark embedding algorithm is the fidelity of the watermarked images. Fidelity refers to the perceptual similarity of the original and watermarked images. Another property that is closely related to fidelity is payload of the watermark. Data payload refers to the number of bits a watermark encodes within a unit of time or within an image. Digital watermark is embedded into the image with maximum payload of the watermark while maintaining the perceptual similarity of the unwatermarked and watermarked images. Higher payload probably downgrades the image fidelity. Lower payload which is preferable to maintain high fidelity might not be able to represent sufficient information in the watermark. Generally, watermark with higher payload is more robust to attacks. In this paper, the payload of the watermark is set consistent at 1000 bits. The watermarking algorithm used throughout all experiments in this paper is Cox's spread spectrum (Cox et al. 1997). We use Lena's image in all experiments.

Neural networks were introduced into watermarking by Yu et al. (2001) that used neural networks to make the watermark detection more robust against common attacks. Picard and Robert (2001) proposed Multilayers Neural Networks architectures to build public detection functions in public key watermarking system. Shen et al. (2003), and Yu and Sattar (2002) used Independent Component Analysis method for blind watermark extraction. Zhang et al. (2003) used Radial Basis Function Neural Networks to find the maximum watermark embedding strength according to the frequency component feature of the cover image. Similarly, Lou et

al. (2003) and Davis and Najarian (2001) used neural networks based on Human Visual Model to estimate the maximum watermark embedding strength to eliminate human intervention. Shieh et al. (2004) later used genetic algorithms to determine the best watermark embedding positions in block-based DCT domain watermarking. Fu et al. (2004) used Support Vector Machine for watermark detection and extraction.

Various watermark detector fundamentals are analysed to investigate the behaviour of the hyperplanes. These hyperplanes are actually the decision boundary of the detector that determines existence and non-existence of watermark in a cover image. The linearly and non-linearly separable data of the Stirmark attacked cover image is analysed. Studies of the hyperplane intuited SVM as the watermark detector. Necessary pre-processing is run to determine the right behaviour of the training set while finding the optimal parameters for SVM. Experiment results showed promising detector ROC from SVM classifier compared to correlation detector. Throughout this paper, SVM classifier and detector are used interchangeably to show the terminology familiarity before and after empirical experiments.

Section 2 begins with the analysis of various digital watermark detectors and its behaviour to malicious attacks. Section 3 further analysed the nonlinearity of the Stirmark attacked images from the hyperplane perspective. Section 4 shows the capability of SVM classifier that has high detection accuracy on nonlinear dataset that is the Stirmark attacked images. Remedies are implemented to pre-process the training data in Section 5. Various detector kernels are compared in literature and with empirical experiments in Section 6. In Section 7, the watermark detection accuracy of the correlation detector and SVM classifier are analysed based on ROC curves that take into account both false positive and false negative probabilities. Improvement from SVM classifier over correlation detector is analysed in Section 8. Lastly, Section 9 concludes the performance of the SVM classifier that improves the robustness of the chosen Cox's watermarking scheme to Stirmark attacks. SVM classifier's performance compared to correlation detector is overwhelming with its blind detection compared to non-blind correlation detector.

2. DETECTOR ANALYSIS AND IMPACTS OF ATTACKS

Kalker (1998) has shown that typical watermark detection scheme can be modelled by a black box D that takes as input a vector $x \in X \subset R^N$ (large N and R is real number) and returns a binary decision $b \in \{+1,-1\}$. Decision b=+1 is interpreted as x is watermarked and b=-1 is interpreted as x is not watermarked. The black box D

is characterized by a 5-tuple $\{w, A, B, h, g\}$ where $A \le B$ are positive real numbers and where $w \in V^N$ for some small and symmetrical subset $V \subset R$. The function h is some hash functions on the space of inputs X that returns uniformly distributed values on the interval [0,1]. The function g is a monotonically increasing self map of the interval [0,1]. The decision $b = D(x)$ is obtained by correlating x with w and comparing the correlation value d with the thresholds A and B. The decision d is computed as

$$d = \frac{\sum_{i=0}^{N-1} w_i x_i}{\sigma_x \sqrt{N}} \tag{1}$$

Kalker (1998) has shown that the d-distribution is close to a normal distribution N(0,1) for a large N if x and w are independent. If $|d| \le A$, x is judged to be unwatermarked and the value $b = -1$ is output. If $|b| \ge B$, x is judged to be watermarked and the value b=+1 is returned. For d between A and B a value b is returned subjected to the dependency such that the probability of a return value -1 is close to 1 for d close to A. Similarly the probability of a return value +1 is close to 1 for d close to B. In short, the return value b is computed as

$$b = \begin{cases} -1 & \text{if} \quad |d| \le A \\ +1 & \text{if} \quad |d| \ge B \\ -1 & \text{if} \quad A < d < B \quad \text{and} \quad h(x) > g(\tilde{d}) \\ +1 & \text{if} \quad A < d < B \quad \text{and} \quad h(x) \le g(\tilde{d}) \end{cases} \tag{2}$$

where $\tilde{d} = (d - A)/(B - A)$. This probability P is depicted in Figure 1.

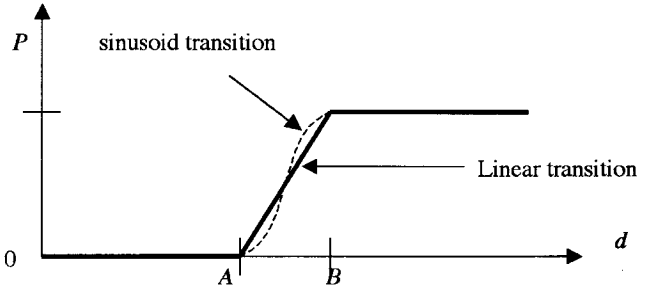


Figure 1 The probability of return value b as function of the correlation value d. The linear and sinusoid transitions are shown (linear-solid line,transition-dashed line)

Multiple of the watermark w are then added to the original content

$$y = x + \lambda w \tag{3}$$

If λw is small compared to x, then the correlation value d that is obtained from D consists of two terms:

$$d \approx d_x + \lambda \frac{\sqrt{N}}{\sigma_x} \tag{4}$$

D will decide y is watermarked when λ is chosen large enough. λ is the embedding strength of the watermarking scheme that determine the detectability of y.

Mansour and Tewfik (2002) have reviewed the various types of public watermark detector extensively. They have covered correlation based detectors, generalized exponential family detector, asymmetric detector, multibit embedding and quantization based detector. Mansour and Tewfik (2002) showed that all detectors share the common feature that is the decision boundary is parametric. This means the decision boundary can be fully specified by a finite set of parameters and generally, the detectors can be formulated as a binary hypothesis test. Let I as the original (unwatermarked) signal, and S as the signal under investigation by the watermark detector, WM as the watermark, and I_W as the watermarked signal, M and N respectively as number of row and column of the signal, L (=MxN) as signal length, wm_n as samples of the watermark indexed by n, a binary hypothesis test can be formulated as

$$\begin{aligned} &\text{H}_0: Iw = I \\ &\text{H}_1: Iw = I + WM \end{aligned} \tag{5}$$

assuming the detector removed the signal mean prior to detection. The optimal detector depends on the assumed underlying probability density function. It becomes a correlation detector for Gaussian distribution and if watermarks of equal weights are used.

Correlation detector is the most common and optimal detector for the class of additive watermark (Kalker 1998). Hence, the watermarking schemes that use correlation detector will be analysed in more details.

When the correlation is performed in the signal domain, the log-likelihood test statistic is reduced after removing the common terms to

$$l(S) = S^* WM = (1/L)\sum_n s_n wm_n \tag{6}$$

where "*" denotes the conjugate transpose of the matrix. H_1 is determined if $l(S) > \lambda$ and H_0 is decided otherwise, where λ is the detection threshold.

Geometrically, Figure 2 shows that the watermark detector involves a decision boundary that is a hyperplane in the multidimensional space S^L. The hyperplane requires L distinct points on it to be completely specified. Least square minimization can be applied to estimate the hyperplane given sufficient points on the boundary. The detector can be viewed as a black box (Kalker 1998; Linnartz et al. 1998). Slight changes can be made to the watermarked signal until reaching a point that the detector is not able to detect the watermark (Kalker 1998; Mansour and Tewfik 2002; Linnartz et al. 1998). The possible changes are normal image manipulation such as JPEG compression or intent modification like geometrical distortion. The minimal changes required to render the watermark undetectable is following the minimum normal projection as labelled in Figure 2. There are three solid arrows showing the other projections that are caused by the changes. The projections as long as crossing the decision boundary will render the watermark undetectable. The direction of the solid line arrow is showing projection from watermarked plane H_1 to unwatermarked plane H_0.

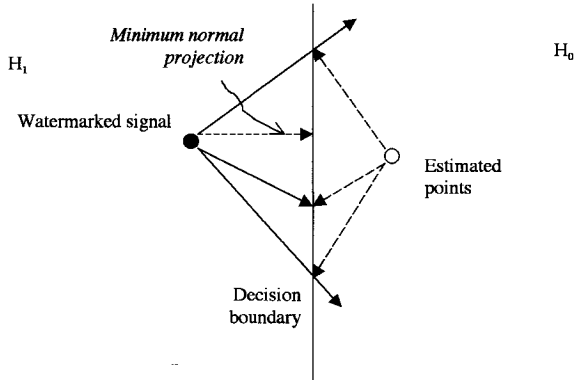


Figure 2 Projection caused by changes on image and its decision boundary of hypothesis test Mansour and Tewfik 2002)

The works from Kalker (1998) and Mansour-Tewfik (2002) showed similar intuition of binary decision in term of decision boundary and hypothesis testing. In general, the detectors are treated as binary decision maker. Their works are different in determining the decision boundary where Kalker (1998)'s detector considers a fuzzy area that is $A<d<B$ as shown in Equation 2. A gradient function $g(d)$ is used to determine the boundary of a vector x that falls in watermarked and unwatermarked hyperplane. In pattern recognition research domain, the fuzzy area in Kalker (1998)'s detector is analogous to the misclassification that is caused by outliners. These outliners are root from the nature of the dataset distribution. The

distribution of the data can be either linearly or unlinearly separable. Unlinearly separable data causes misclassification of the dataset.

Figure 3 and Figure 4 show linearly and unlinearly separable dataset. Considering binary classification, ring circle represented class A and cross circle represents class B. In Figure 3, it is visually clear that class A and B are well-separated by the hyperplane *T* where ring circles fall on the right side of *T* and cross circles fall on the left side of *T*. In Figure 4, there are some outliners that fall on the wrong side of *T*. There are some ring-circles falls on the left side of *T* whilst some cross-circles fall on the right side of *T*.

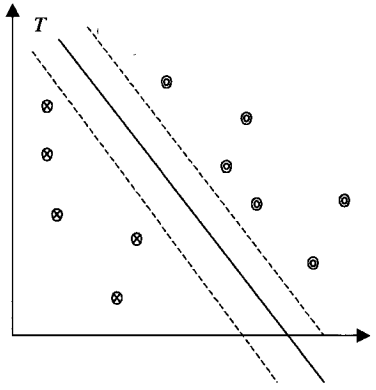


Figure 3 Linearly separable dataset (A ring circle represents class A, cross circle represents class B)

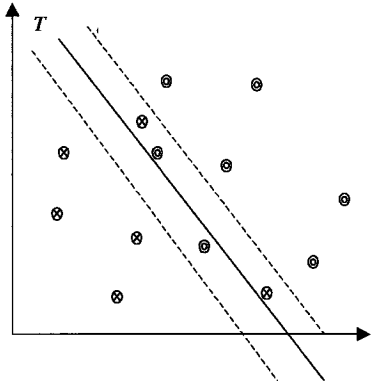


Figure 4 Unlinearly separable dataset (A ring circle represents class A, a cross circle represents class B)

The misclassification caused by outliners in pattern recognition and image processing domain is very much alike to digital watermarking. The idea of perceiving watermark detection as image classification and pattern recognition can be further enhanced by observing the behaviour of digital watermark detector under attacks (Then and Wang 2005, 2006).

False alarm from the digital watermark detector is inevitable (Cox et al. 2002). This false alarm comes in the form of false positive and false negative. For false positive detection, the detector detects an unwatermarked cover data as watermarked while for false negative detection, the detector detects watermarked cover data as unwatermarked. This false detection can be described as type-A and type-B errors in statistical terms.

The false detection rates increase after the cover data is attacked. The specific attack that solely challenged the robustness of the watermarking scheme is Stirmark attack. It is also a widely accepted benchmarking attack to evaluate the robustness of digital watermarking schemes (Petitcolas and Anderson 1999; Petitcolas et al. 1998). In general, the attacks increase the false positive and false negative probabilities. Hence, by perceiving watermark detection as pattern recognition, the issue of outliners in misclassification can be treated similar to the false detection in digital watermarking. From pattern recognition and image classification perspective, the main reason of false detection is due to the nonlinearly separable data. The nonlinearity of the data is worsened after the cover data is attacked. This leads to the investigation of nonlinearity of the cover data.

3. NONLINEARITY OF STIRMARK ATTACKED IMAGES

Webb (2004) has addressed the decision boundary of nonlinearly separable data in statistical pattern recognition context. Radial basis function (RBF) network and SVM are the most suitable kernel classifier for nonlinear separable data (Webb 2004). These methods are developed primarily in the neural networks and machine learning, and are flexible models for nonlinear separable data discriminant analysis (Webb 2004).

RBF network implements a mapping function, $F(R^n) \rightarrow R$ according to

$$F(x) = \sum_{i=1}^{n} w_i \phi(\| x - x_i \|) \tag{7}$$

where R is real number, n is the dimension of R, $x \in R^n$ is the input vector, $\{\phi(\| x - x_i \|) | i = 1,2,....,n\}$ is a set of n arbitrary nonlinear function from R^+ to R, known as RBF, $\|.\|$ denotes norm that is usually Euclidean distance, $x_i \in R^n$ $(1 \leq i \leq n)$ are the RBF centres, and n is the number of centres. Choice of unlinear function $\phi(.)$ can be:

(i) Gaussian function, $\phi(x) = \exp\left(-\frac{\| x - x_i \|^2}{\text{cov}^2}\right)$, where x_i is centre of cluster i and cov is the covariance of the cluster i.

(ii) Thin-plate-spline function, $\phi(x) = \left(\frac{\| x - x_i \|^2}{\text{cov}}\right)^2 \log\left(\frac{\| x - x_i \|^2}{\text{cov}}\right)$

(iii) Multiquadric function, $\phi(x) = \sqrt{\| x - x_i \|^2 + \text{cov}^2}$

(iv) Inverse multiquadric function, $\phi(x) = \frac{1}{\sqrt{\| x - x_i \|^2 + \text{cov}^2}}$

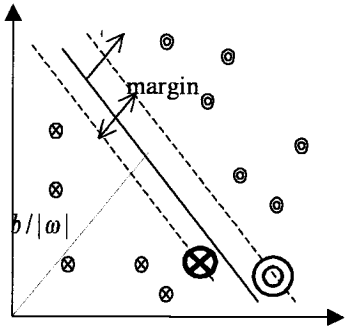
Gaussian function is most commonly used in the neural network community (Chen et al. 1999). Furthermore, theoretical analysis and practical experiments suggest that the choice of the nonlinearity $\phi(.)$ is not crucial to the performance of the RBF network. To select the suitable set of RBF centres, x_i is the crucial factor to the performance of the RBF network. Chen et al. (1999) proposed an Orthogonal Least Squares learning algorithm that operates in forward regression procedure to select RBF centres, x_i, from the data points.

Comparative study reviewing several approaches to RBF training show that SVM learning approach is often superior on a classification task compared to the standard two-stage learning of RBFs (Webb 2004; Schwenker et al. 2001). Besides, works from Massachusset Institute of Technology (MIT) has also undermined the suitability of RBF compared to Support vector machine in classifying data (Schőlkpf et al. 1996). The applicability of the studies by Schwenker et al. (2001) and Schőlkpf et al. (1996) on digital watermarking is investigated in this paper. Schőlkpf (1996)'s works have tested SVM with Gaussian Kernel, classical RBF machine and hybrid of them on recognizing the US postal service database of handwritten digits. Their results showed that SVM approach is not only theoretically sound but superior in the practical application. Further justification of choosing the right kernel and its associated parameters are discussed in Section 6. Thus, it is worthwhile to give a reasonable background of SVM prior to its role as digital watermark detector.

4. SVM CLASSIFIER

Support Vector Machines are machine learning tool for performing classification and detection tasks. They have been applied to a wide range of real-world problems such as face recognition, image retrieval, and text categorizations. SVM classifier is trained using a set of labelled training examples. The training set consists of l training examples, with each example described as h-dimensional vector. Each example is labelled as belonging to one of two classes, $y \in \{1, -1\}$, described as the positive class and the negative class. Hence, each example is represented as $\{x_i, y_i\}$, $i=1,\ldots,l$, $y_i \in \{1, -1\}$ $x_i \in \boldsymbol{R}^h$ where $\boldsymbol{R}$ is real number. After the training, the resulted SVM is able to classify unseen instances, x, into a class based on the examples learnt from the training set. Typically, SVM outputs zero when classification failed.

In the simplest form, SVMs are hyperplanes that are separating training data by maximum margin. This margin is defined as the distance between the closest training examples in the positive and negative classes to the separating hyperplanes (Burges 1998). The training examples that determine the margin are known as "support vectors". All vectors lying on one side of the hyperplane are labelled as 1, and vectors lying on another side of the hyperplane are labelled as –1.



___	Separating hyperplane
◎	Examples from class 1
⊗	Examples from class 2
◎	Support vectors in class 1
⊗	Support vectors in class 2

Figure 5. Simple SVM classification of two classes in a two-dimensional input space

There is tradeoff on finding the optimal hyperplane decision boundary when some examples unavoidably fall on the wrong side of the boundary. Hence the decision boundary have to be found by maximizing the margin between the two classes while minimizing the penalty associated with the misclassifications in the training set. The equation of a decision surface in the form of a hyperplane that does the separation is

$$(\omega \bullet x) + b = 0 \tag{8}$$

where x is an input vector, ω is an adjustable weight vector, and b is a bias. The distance from the origin to the optimal hyperplane is given by $b/|\omega|$, where $|\omega|$ is the Euclidean norm of ω. A scaling of b and ω leaves this distance unchanged that maintain the condition in Equation (11). Supposed all training examples satisfy the following constraints as shown as dotted lines respectively in Figure 5:

$$(\omega \cdot x_i) + b \geq 1 \text{ for } y_i = +1 \tag{9}$$

$$(\omega \cdot x_i) + b \leq 1 \text{ for } y_i = -1 \tag{10}$$

Equations (9) and (10) form a set of inequalities:

$$y_i(\omega \cdot x_i + b) - 1 \geq 0 \quad \forall i \tag{11}$$

By introducing positive Lagrange multipliers α_i, $i=1,\ldots,l$, the optimal hyperplane defined by equation (8) is found by solving the following quadratic programming problem, where $\alpha_i, i = 1,\ldots,l$ are the Lagrange multipliers:

$$\min_{\omega,b,\alpha} \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{l} \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^{l} \alpha_i \tag{12}$$

5. TRAINING SET PREPROCESSING

The performance of SVM classifier is dependent on its training and therefore it is important to make sure the training dataset consists of positive and negative classes. The generalization performance of SVM classifier is highly dependent on the linear separation of positive and negative classes in the input space. The images used in training and testing had dimensions of 256 x 256 pixels. We found that using the training set by including every pixel from an image of these dimensions produced SVM that can still maintain its high generalization ability when the watermarked images are attacked under Stirmark.

Samples from watermarked and unwatermarked images are chosen to train and test SVM. The image Lena as in Figure 6 is used throughout the experiments.

Figure 6. Lena

Lena image is pre-processed by incrementing the amplitude of the image pixels up to as much as 220. The watermarked images are denoted as I_i^{wm} and the unwatermarked images are denoted as I_i^o where i represents the increment value of the image pixels. The diagrammatic view of the training scheme is shown in Figure 7 where $\{I_i^{wm} \cup I_i^o, i = 1,2,3,.....n\}$ represents all adjacent training dataset from $i=1$ to $i=n$ are fed into the training. Few examples of I_i^{wm} are arbitrarily selected and shown in Figure 8.

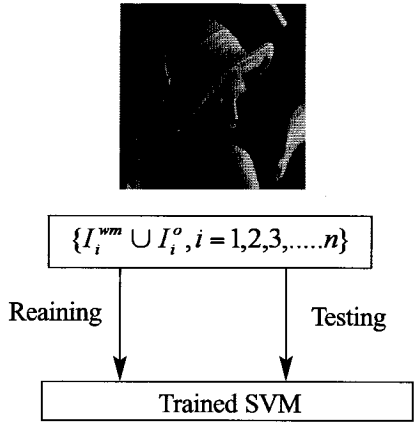


Figure 7. Diagrammatic view of training scheme

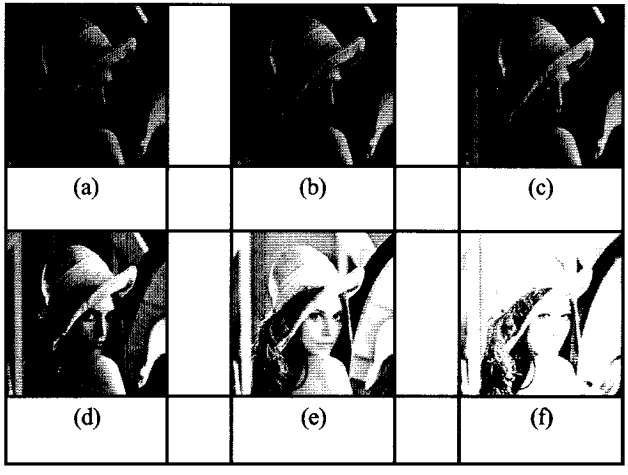


Figure 8. Resultant Images from Pixel Increments. (a) +1, (b) +10, (c) +20, (d) +50, (e) +100, and (f) +150

This experiment takes all the watermarked and unwatermarked images to train the SVM. There are 220 watermarked images and 220 unwatermarked images being fed into the training. The same training images are tested on the trained networks. This yields zero error from the training sets.

The trained SVM is tested on the training images. Then, the trained SVM is cross-tested with separate set of images besides the ones used in training. As depicted in Table 1. The results are measured in terms of false positive and false negative probability.

Table 1. Results of network testing

Training Images	Testing Images	False positive	False negative
Lena	Lena	0	0
	Baboon	0	0
	Peppers	0	0

The outputs of the SVM trained with Lena when is tested with Baboon and Peppers are shown in Figure 9 and Figure 10. All the outputs fall under 0 that indicate negative classification. In other words, SVM trained with Lena will not be able to recognise Baboon and Peppers regardless of whether they are watermarked or unwatermarked.

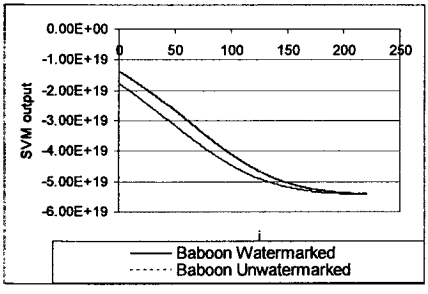


Figure 9. SVM trained with Lena and tested with Baboon

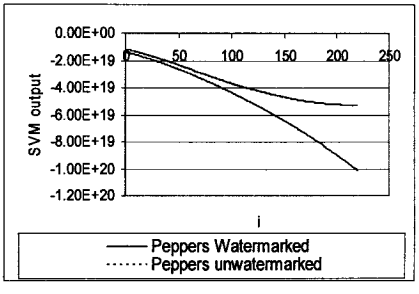


Figure 10. SVM trained with Lena and tested with Peppers

Subsequent experiment trained SVM using three training image sets. Then, the trained SVM is tested with all three image sets. Its result is shown in Table 2.

Table 2 Results of detection rate of network trained using three image sets

Training Images	Testing Images	False positive	False negative
Lena, baboon, Peppers	Lena	0	0
	Baboon	0	0
	Peppers	0	0

This training scheme aims to determine the baseline performance of SVM on detecting watermarked and unwatermarked training images. Independently, SVM trained with separate set of training images has zero false positive and false negative. SVM trained with all the three training sets again detect all the training images without false alarm. This training scheme has verified the expected baseline results.

6. TRAINING PARAMETERS

Fu et al. (2004) used the blue component of colour image to train the SVM using linear, polynomial and RBF kernels. The training scheme is represented in Figure 11.

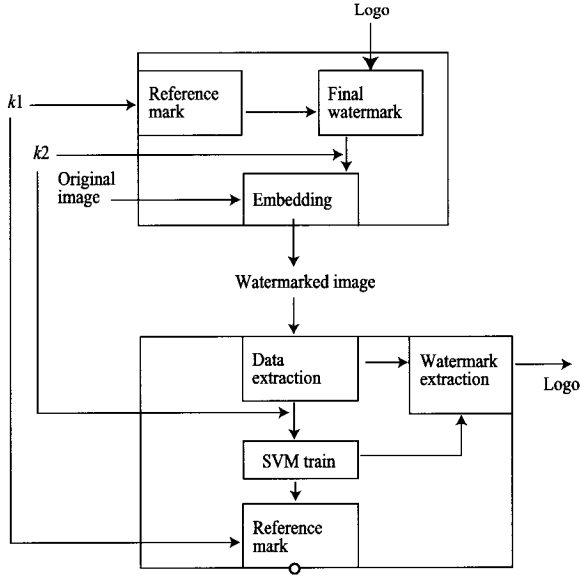


Figure 11. Diagrammatic representation of Fu et al. (2004)'s watermarking scheme

The training dataset comprises of two features

i) at pixel $p_t=(i,j)$ selected randomly by key k_2, the difference between the intensity of the blue component value of the central pixel and that of the others within the slide window with size c, $d_{i,j}$, where $d_{ij} = B'_{ij} - \bar{B}'_{ij}$
where $\bar{B}'_{ij} = \left(\sum_{r=-c}^{c} B'_{i+r} + \sum_{r=-c}^{c} B'_{i,j+r} - 2B'_{ij}\right)/(4c)$

ii) reference mark at pixel (i,j), r_t. This reference mark is embedded based on key k_1.

One of the row of the training feature vector, $d_{i,j}$ is defined as $D_t=\{d_{i,j-2}, d_{i,j-1}, d_{i,j}, d_{i,j+1}, d_{i,j+2}, d_{i-2,j}, d_{i-1,j}, d_{i,j}, d_{i+1,j}, d_{i+2,j}\}$. Thus the training dataset can be defined as $\{D_t,r_t\}_{t=1,....K}$.

Fu et al. (2004) tested their algorithm on RGB images with various content complexities including Lena, Peppers, and Baboon. The experimental results shown in their paper is based on the RGB Lena. The signature used in the experiments is a binary logo image with size 32 x 25 i.e. "CHN". This signature is reshaped into line ordered watermark sequence S by row major fashion and modulated into PN sequence.

One of the important parameters used in the experiments is the SVM kernels including linear, polynomial and RBF. The best suitable one is necessary to produce the lowest Bit Error Rate (BER). From Fu et al. (2004)'s experiments on these three kernels, the RBF is the best, defeating linear and polynomial. The kernel parameter σ of the RBF kernel from 1 to 40 are tested and the best BERs are obtained from 5 to 10.

Series of experiments show the effects of the setting different values for the exponent, e, of Polynomial kernel and the gamma, σ of RBF kernel. The summarized parameters setting and its corresponding ROC curves are shown in Table 3.

Figure 12a-k show the ROC graphs of 4 sets of testing images i.e. Set 1-Stirmark attacked watermarked Lena, Set 2-Stirmark attacked unwatermarked Lena, Set 3-Stirmark attacked pixel-incremented watermarked Lena, and Set 4 - Stirmark attacked pixel-incremented unwatermarked Lena. The training dataset are 50 pixel-incremented watermarked Lena and 50 pixel-incremented unwatermarked Lena for all the settings.

The ROC curves resulted from testing image set 1 and 2 are depicted in solid line while the ROC curves resulted from testing image set 3 and 4 are shown in dotted line. The unclassified result means that the trained SVM yields zero during testing. Zero means the SVM failed to classify the dataset to neither positive nor negative class.

Table 3. Parameters Setting and ROC Curves

Kernel	Parameters	ROC Curves
Linear	Nil	Figure 12a
Polynomial	e=1	Figure 12b
	e=2	Figure 12c
	e=3	Unclassified
RBF	σ =1	Figure 12d
	σ =2	Figure 12e
	σ =5	Figure 12f
	σ =10	Figure 12g
	σ =15	Figure 12h
	σ =20	Figure 12i
	σ =25	Figure 12j
	σ =30	Figure 12k

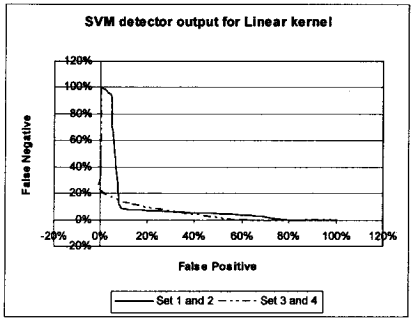


Figure 12a

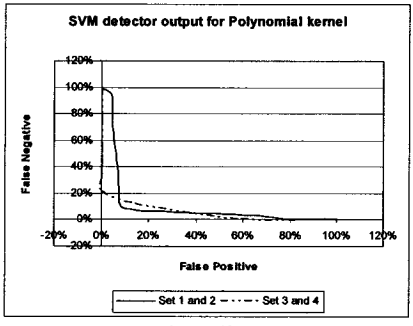


Figure 12b

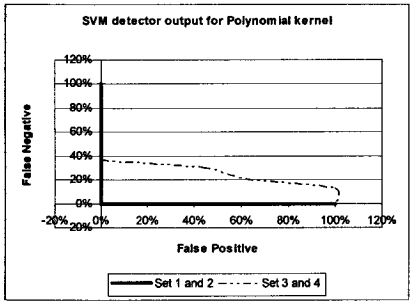


Figure 12c

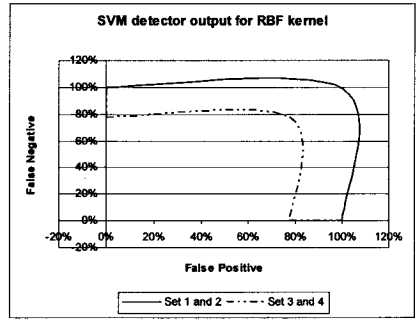


Figure 12f

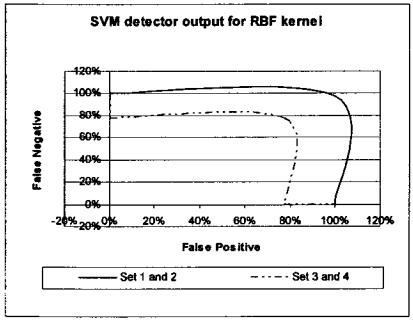


Figure 12d

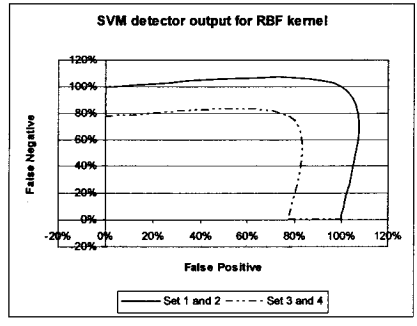


Figure 12g

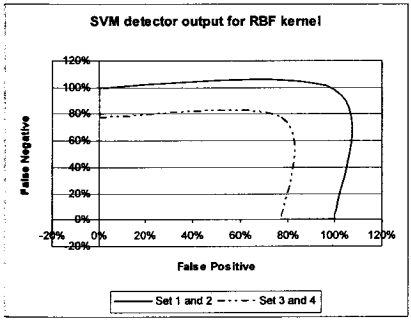


Figure 12e

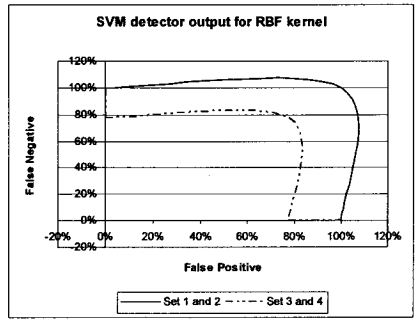


Figure 12h

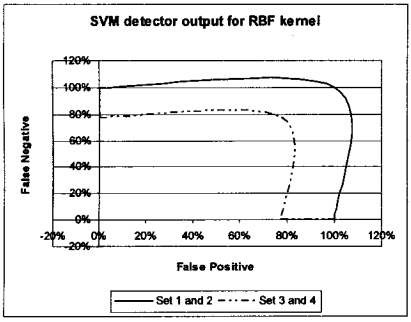


Figure 12i

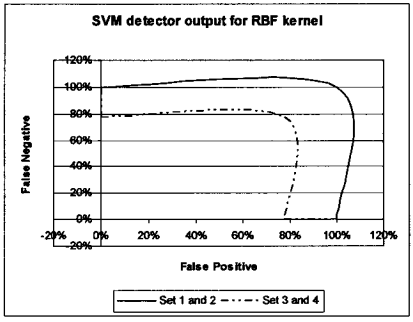


Figure 12j

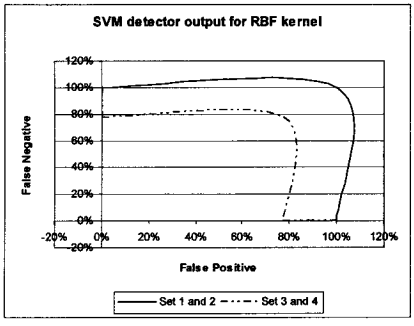


Figure 12k

From Figures 12a-k, it is obvious that the parameter settings for linear, polynomial, and RBF kernel do not show acceptable ROC curves except Figures 12a-c. With Polynomial Kernel and exponent e=2, the optimal detection accuracy is achieved with the lowest false positive and false negative probabilities are achieved from ROC curves as in Figure 12c. This observation contradicts Fu et al. (2004)'s result that shows optimal detection from RBF kernel.

7. PERFORMANCE OF CORRELATION AND SVM DETECTOR

Both correlation and SVM detectors are further tested on the images after Stirmark attack (Kutter and Petitcolas 1999). The series of Stirmark attack are launched on both original images and watermarked images to determine the ROC of correlation and SVM detectors.

The types of Stirmark attacks launched on the images are Affine transformation, Convolution filter, JPEG compression, Latest Small Random Distortion, Median cut, Noise addition, strength of fake watermark, Self Similarities, Lines removal, rescaling, rotation, rotation and cropping, rotation and scaling, Small Random Distortion. In general, Stirmark generates 106 attacked versions of a vulnerable image after the attacking process. This range of Stirmark

attacks is far more extended compared to Fu et al. (2004) and its impact on the detectors is studied based on the false alarm probability produced by the detectors.

Cox et al. (1997)'s spread spectrum is used in all experiments due to its implementation simplicity. When inserting the watermark, *WM*, into the image, *I*, to obtain I_i^{wm}, a scaling parameter α which determines the extent to which *WM* alters I is set, so that $I_i^{wm} = I_i(1+\alpha WM)$. The scaling parameter α is set consistent at 0.1 in this paper. A 1000-bit watermark, *WM*, is generated randomly based on Gaussian distribution. Then, the watermarked images are attacked. The existence of the embedded watermark is determined using Cox's correlation detector and SVM classifier.

From the Stirmark attacked images, the detection results of Cox's spread spectrum are plotted as Frequency Distribution of correlation coefficients as in Figure 13, False Positive vs False Negative graph is shown in Figure 14 and its ROC is shown in Figure 15. The same watermarked and non-watermarked images are used to train SVM. The detection results from the Stirmark attacked images expressed in frequency distribution, false positive vs false negative, and ROC are depicted in Figure 16-18 respectively.

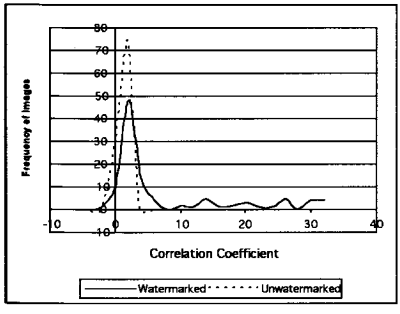


Figure 13. Frequency distribution of Cox's detector

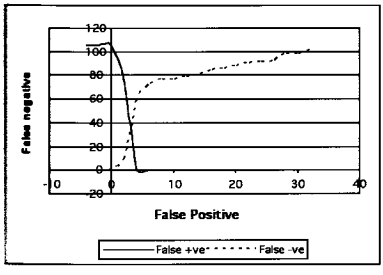


Figure 14. False positive vs false negative of Cox detector

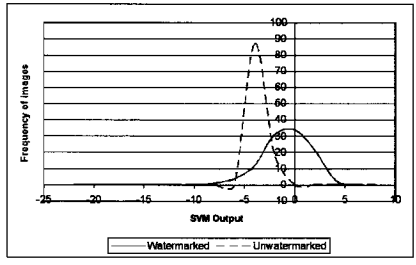


Figure 16. Frequency distribution of SVM detector

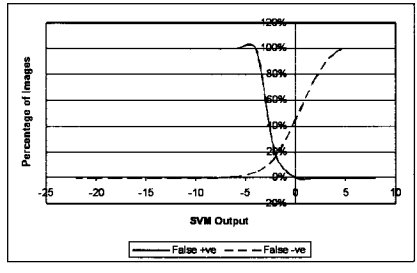


Figure 17. False positive vs false negative of SVM detector

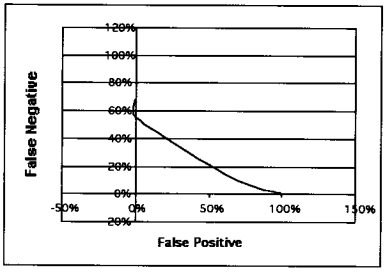


Figure 15. ROC of Cox's detector

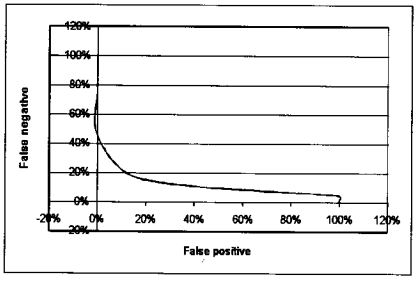


Figure 18. ROC of SVM detector

8. COMPARISON OF ROC

The relative performance of Cox's correlation detector and SVM classifier can be clearly assessed by comparing their ROC graphs from Figure 15 and Figure 18. ROC shows the tradeoff of the false positive probability and false negative probability of the detector. In all watermarking detection system, higher false positive can be achieved with lower false negative, and vice versa. The performance of the watermarking system can only be interpreted by considering both false positive and false negative probabilities. The ROC graphs of Cox's correlation and SVM detectors are compared as in Figure 19.

Diagonal of the graph shows the smallest false positive and false negative probability that one watermarking system could possibly achieve simultaneously. The relative performance of both detectors can be compared based on the points x and y that falls on the diagonal of the graphs. Point x shows 18% false positive and negative probability for SVM while point y approximates 35% as lowest false positive and false negative probability for correlation detector. Obviously, SVM detector outperformed the correlation detector.

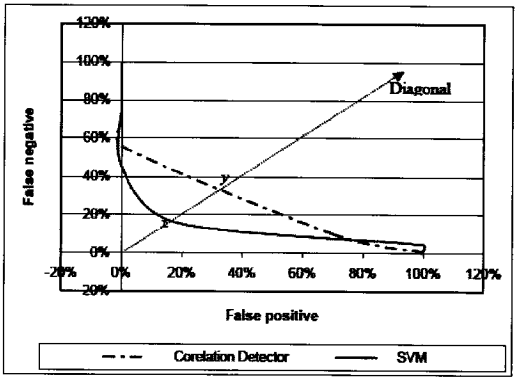


Figure 19. ROC curves of Cox and SVM detectors

9. CONCLUSION

With preprocessing of the training set and extensive experiments on the various settings of SVM parameters, we found the optimal setting of SVM used for blind watermark detection. This setting is however very different from Fu et al. (2004)'s RBF kernel setting. The optimal setting lies on the Polynomial kernel with exponent d=2. Perceiving watermark detection as image classification poses challenges of maintaining two important features of watermark detector that are robust to attack and blind detection ability. This perception led to the analysis of various watermark detectors in term of their hyperplanes. From the findings of this paper, SVM detector has higher robustness as it survives Stirmark attacks better than the correlation detector according to the ROC curves. SVM detector eliminates the use of the original work during detection as this shows its blind detection ability. The SVM detector clearly outperformed the correlation detector in the Cox's spread spectrum watermarking system used in this paper.

10. REFERENCES

Burges, C. J. C., 1998. A Tutorial on Support Vector Machines for Pattern Recognition, Kluwer Academic Publishers, Boston.

Chen, S., Cowan, F.N., and Grant,P.M., 1999. Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks,IEEE Trans. On Neural Network, 2,2, pp.302-309.

Clark,D., Albrecht, D.W., and Tischer, P. 2004. An Investigation into Applying Support Vector Machines to Pixel Classification in Image Processing, 17th Australian Joint Conference on Artificial Intelligence, LNCS 3339 , pp140-151.

Cox, I.J., Kilian, J. , Leighton, T., and Shamoon, T., 1997. Secure Spread Spectrum Watermarking for Multimedia, IEEE Trans. Image Processing, Vol. 6, No. 12, pp. 1673-1687.

Cox, I.J., Miller, M.L., and Bloom, J. 2002. *Digital Watermarking,* Morgan Kaufmann.

Davis, K.J., and Najarian, K.,2001. Maximizing Strength of Digital Watermarks using Neural Networks, IJCNN'01,vol.4,pp. 2893-2898.

Fu, Y. , Shen, R., and Lu, H., 2004. Watermarking Scheme based on Support Vector Machine for Colour Images, IEE Electronic Letters,40,16.

Joachims, T., 1999. Making large-Scale SVM Learning Practical, Advances in Kernel Methods - Support Vector Learning, In B. Schölkopf and C. Burges and A. Smola eds. MIT-Press.

Kalker. T.. 1998. A Security Risk for Publicly Available Watermark Detectors, Benelux Information Theory Symposium, May 98, Veldhoven, The Netherlands.

Kutter, M., and Petitcolas, F.A.P., 1999. A Fair Benchmark for Image Watermarking Systems, Security and Watermarking of Multimedia Contents, In Proc. Of SPIE, 3657, USA.

Linnartz, J-P., and van Dijk,M., 1998. Analysis of the Sensitivity Attack against Electronic Watermarks in Images. Information Hiding 1998, LNCS 1525, pp. 258-272.

Lou, D.C., Liu, J.L., and Hu, M.C.,2003. Adaptive Digital Watermarking using Neural Network Technique, IEEE 37th Annual 2003 Intl Carnahan Conference on Security Technology, pp.325-332.

Mansour,M.F. and Tewfik,A.H., 2002. Improving the Security of Watermark Public Detectors, In Proceedings of DSP 2002,vol.1,pp.59-66

Mei, S.H., Li,R.H., Dang,H.M., and Wang, Y.K., 2002. Decision of image watermarking strength based on artificial neural-networks, ICONIP 2002,vol.5,pp. 2430-2434.

Petitcolas, F.A.P., and Anderson,R.J., 1999. Evaluation of Copyright Marking Systems. In Proceedings of IEEE Multimedia Systems, vol.1, pp. 574-579.

Petitcolas, F.A.P., Anderson, R.J., and Kuhn, M.G., 1998. Attacks on Copyright Marking Systems, Second workshop on Information Hiding, LNCS 1525, pp.218-238.

Picard, J., and Robert, A., 2001. Neural Networks Functions for Public Key Watermarking, IH 2001, LNCS 2137, pp. 142-156.

Sch_lkpf, B., Sung, K., Burges, C., Girosi, F., Niyogi, P., Poggio, T., and Vapnik,V., 1996. Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers. MIT, AI Lab and Center for Biological and Computational Learning Department of Brain and Cognitive Sciences,A.I. Memo No. 1599, C.B.CL. Paper No. 142.

Schwenker, F., Kestler,H.A., and Palm, G.,2001. Three Learning Phases for Radial-basis-function Networks, Neural Networks, 14, pp.439-458.

Shen, M.F., Huang, J., and Beadle, P.J., 2003. Application of ICA to the digital image watermarking, IEEE Intl Conf. on Neural Networks and Signal Processing, vol.2 pp. 1485-1488.

Shieh, C.S., Huang, H.C., and Wang, F.H., 2004. Genetic Watermarking based on Transform-Domain Technique, Pattern Recognition, vol. 37, 3, Elsevier, pp. 555-565.

Then,H.H. Patrick, and Wang, Y.C., 2006. Support Vector Machine as Digital Image Watermark Detector, Real-Time Image Processing 2006. Edited by Kehtarnavaz, Nasser; Laplante, Philip A. Proceedings of the SPIE, Volume 6964, pp. 478-489.

Then,H.H. Patrick, and Wang,Y.C., 2005. Perceiving Digital Watermark Detection as Image Classification Problem using Support Vector Machine. In Proceedings of CITA05, pp.198-206.

Vapnik, V., 1995. The Nature of Statistical Learning Theory, Springer-Verlag New York, USA.

Vapnik, V., 1998. Statistical Learning Theory, John Wiley, New York, USA.

Webb, A.,2004. *Statistical Pattern Recognition,* 2nd Ed. Wiley.

Yu, D., and Sattar, F., 2002. A New Blind Watermarking Technique Based on Independent Component Analysis, IWDW 2002, LNCS2613, pp. 51-63.

Yu, P.T., Tsai, H.-H., and Lin, J.-S., 2001. Digital Watermarking based on Neural Networks for Color Images, Signal Processing, vol 81,3, Elsevier, pp. 663-671.

Zhang, Z.M., Li, R-Y., and Wang, L.,2003. Adaptive Watermark Scheme with RBF Neural Networks, IEEE Int. Conf. Neural Networks & Signal Processing, Nanjing, China, pp.1517-1520.