Chapter 13

DETECTING STEGANOGRAPHY USING MULTI-CLASS CLASSIFICATION

Benjamin Rodriguez and Gilbert Peterson

Abstract When a digital forensics investigator suspects that steganography has been used to hide data in an image, he must not only determine that the image contains embedded information but also identify the method used for embedding. The determination of the embedding method – or stego fingerprint – is critical to extracting the hidden information. This paper focuses on identifying stego fingerprints in JPEG images. The steganography tools targeted are F5, JSteg, Model-Based Embedding, OutGuess and StegHide. Each of these tools embeds data in a dramatically different way and, therefore, presents a different challenge to extracting the hidden information. The embedding methods are distinguished using features developed from sets of stego images that are used to train a multi-class support vector machine (SVM) classifier. For new images, the image features are calculated and evaluated based on their associated label to the most similar class, i.e., clean or embedding method feature space. The SVM results demonstrate that, in the worst case, embedding methods can be distinguished with 87% reliability.

Keywords: Steganalysis, multi-class classification, support vector machine

1. Introduction

Steganography is a data hiding and transmission technique that attempts to conceal and prevent the detection of the true content of a message. The steganographic process uses a cover object – often an image – to conceal the message ("stego data"). An embedding algorithm combines a cover image and the stego data to produce a stego image, which is an image that contains the hidden message. Steganalysis, the process of breaking steganography, involves examining a set of cover objects to determine if steganography was used, identifying the fingerprint of the embedding algorithm, and then extracting the embedded content.

Several methods are available for detecting hidden information in images, but the embedding algorithm must be known for any of these methods to be effective. Unfortunately, such steganography fingerprinting is a major challenge as there are more than 250 steganography programs available [16]. To address this issue, it is necessary to develop detection methods that use a combination of features to identify the class or type of embedding method.

This paper presents a multi-class classification method that focuses on classifying unseen instances to their specific embedding method (class). The method categorizes JPEG stego images based on feature classification in which instances are associated with exactly one element of the label set. The multilevel energy band features presented in this paper are used with the multi-class support vector machine (SVM) classification technique. The features are generated from higher order statistics of the multilevel energy bands of the discrete cosine transform (DCT).

The test results are based on an image database of 1,000 high-quality JPEG images taken with a Nikon Coolpix 5. The stego images were created using five steganography tools (F5, JSteg, Model-Based Embedding, OutGuess and StegHide). Each of these tools embeds data using a different technique, with the exception of OutGuess and StegHide that embed similarly but use different randomization techniques. The results demonstrate that, in the worst case, embedding methods can be distinguished with 87% reliability.

The next section discusses embedding methods and multi-class classifiers. Section 3 describes the multilevel energy feature generation technique. This is followed by a description of the multi-class SVM classification method in Section 4. Section 5 presents the results of the SVM classifier using multilevel energy features. This paper ends with concluding remarks and a discussion of future work.

2. Related Work

Each embedding method leaves a fingerprint on the stego image representative of the algorithm used to create the image. Our approach is to use multi-class classifiers to detect specific classes of embedding methods using stego fingerprints. This section discusses the main JPEG image data embedding methods used by steganography tools and the primary multi-class classification methods.

2.1 Embedding Methods

Digital images are often used to hide stego data because numerous redundant portions within the images can be altered without affecting the quality as observed by the human eye [16]. This paper examines five prominent tools for embedding data in JPEG images: F5, JSteg, Model-Based Embedding, OutGuess and StegHide.

The JPEG image format is currently the most prevalent image storage format [16]. The vast number of JPEG images available on the Internet makes them ideal cover images for hiding secret data. A JPEG embedding process embeds the data in discrete cosine transform (DCT) coefficients. First, the DCT coefficients of an image are computed; the coefficients of an 8×8 block of image pixels $f(x, y)$ are denoted by $F(u, v)$. The coefficients are divided by the quantization matrix, which quantizes the coefficients for compression. After this process, most JPEG embedding methods use the least significant bits (LSBs) of the quantized DCT coefficients. Redundant bits are used to embed the hidden message so that the embedding has no effect on the binary encoder. While the embedding does not affect the compression process, modifying a single DCT coefficient affects all 64 pixels in the 8×8 image block.

F5 [18] was developed as a challenge to the steganalysis community. This method exploits the JPEG compression algorithm by decrementing the absolute values of the DCT coefficients in a process known as matrix encoding. An estimated embedding capacity is computed based on the total number of DCT coefficients. A recursive algorithm is then used to match the bits of the message in a hash function to determine the encoding, stopping when one of the coefficients is reduced to zero. An F5 embedding is identifiable by the unnatural coefficient histograms produced by the embedding technique.

Model-Based Embedding [14] fits the coefficient histogram into an exponential model using maximum likelihood. This method addresses the limitations of other embedding methods; it can successfully hide large messages so that they are undetectable by certain statistical analyses and it can achieve maximum capacity. Model-Based Embedding is accomplished by identifying the ideal embedding structure based on the statistical model of the DCT coefficients of the original cover image, and ensuring that the statistical model is retained after the embedding. Although the embedding technique is similar to that used by F5, it does not produce unnatural histogram frequencies for adjacent DCT coefficients. This embedding technique is identified by combining several higher-order statistics.

The JSteg tool encodes messages in JPEG images by manipulating the LSBs of the quantified DCT coefficients. The message is formatted so that the first five bits of the frequency band coefficient indicate the length of the band (size of the embedded message), which is also referred to as the capacity of the block. The next set of bits indicates the bit length of the actual message. This message length indication scheme avoids generating large numbers of zeros that occur when short messages are embedded using a fixed bit length to indicate message size [10]. This type of embedding does not spread the encoded bits among the 14 coefficients; therefore, it can be identified using a first-order statistic (e.g., mean).

OutGuess [13] was designed to evade detection by statistical steganalysis techniques such as the chi-square statistical attack. The embedding technique modifies the LSBs of the DCT coefficients by statistically checking the original image DCT heuristics against the embedded image; it then manipulates nearby DCT blocks to maintain the original DCT histogram. The coefficients ($F(u,v) \notin [0,1]$) are selected using a pseudo-random number generator. The statistical correction method embeds hidden data within the coefficient LSBs while offsetting nearby LSB coefficients with minor bit changes to preserve the chi-square statistic.

StegHide [8] hides data in multiple types of image and audio files. In the case of JPEG images, the color representation sample frequencies are not changed, which makes this method robust to first-order statistical attacks. This robustness is the direct result of embedding stego data within the LSBs of DCT coefficients that have large variations with adjacent coefficients. However, this embedding technique can be detected using a higher-order statistic (e.g., energy).

Proper identification of the embedding technique is crucial to any attempt at extracting the hidden information. The five tools considered in this paper embed data into the quantized DCT coefficients of a JPEG image. Each DCT encoding introduces certain statistical irregularities that constitute a signature. The fundamental problem is to classify the signatures left by the tools.

2.2 Multi-Class Classification Methods

More than 250 tools are available for performing steganography on digital images [16]. Because of this, multi-class classification is an attractive technique for identifying the potential signatures of steganography embedding algorithms. This section describes two promising multi-class classification techniques.

In many multi-class classification methods, two-class classifiers are combined using the posterior probabilities of their outputs. The multi-class learning algorithm must create hyperplane boundaries in kernel space where each hyperplane depends on the margin of separation obtained at the support vector nodes. This is achieved by combining the two-class classification methods using voting and combinations of approximate posterior probabilities, where the use of posterior probabilities enhances the solution by eliminating ties [12]. Another approach is to use combinations of binary classification methods with a naive Bayes classifier, which generalizes to multiple classes [17].

Several multi-class SVM classifiers employ a winner-take-all approach, assigning the class labeling based on a majority vote for the class [7]. The winner-take-all approach uses multiple two-class SVM prototypes per class, separating one class identity from the others. The method combines multiple sets of support vectors to create a large decision boundary separating the desired classes. This is achieved using a constrained quadratic search to find locally-optimal solutions for non-convex objective functions. In this way, the winner-take-all strategy creates a set of linear functions, each of which provides an ordering of the classes for a sample, where the "winner" is the first class in the ordering.

Our approach uses a majority-vote-wins strategy. However, in order to perform classification, a suitable set of features is required. The following section describes the features used to perform steganalysis via multi-class classification.

3. Features

This section describes the DCT multilevel energy bands method for calculating the transform domain features from a JPEG image. The features are obtained by computing the DCT energy bands for each block of 8×8 coefficients.

The transform domain features presented in Figure 1 focus on the energy bands of the DCT coefficients. Figure 1(b) shows the representation of the energy bands after the DCT. The DCT used in JPEG compression does not generate the multilevel energy bands produced by wavelet decomposition. Moreover, the multilevel energy band representation in Figure 1(b) does not allow for the energy levels to be extracted based on the edges of the original image as shown in Figure 1(c). Instead, the DCT output is rearranged in a wavelet decomposition structure to show the energy bands. This structure is created using 8×8 pixel blocks, which are the same as those used during JPEG compression. For each 8×8 block, the DCT energy band decomposition of vertical, diagonal and horizontal edges are formed via zigzag (Figure 1(d)) and Peano scans (Figure 1(e). Rearranging the coefficients of the DCT splits the frequency spectrum into uniformly spaced bands containing vertical, horizontal and diagonal edges. The ideal representation of the energy bands is shown in Figure 1(f).

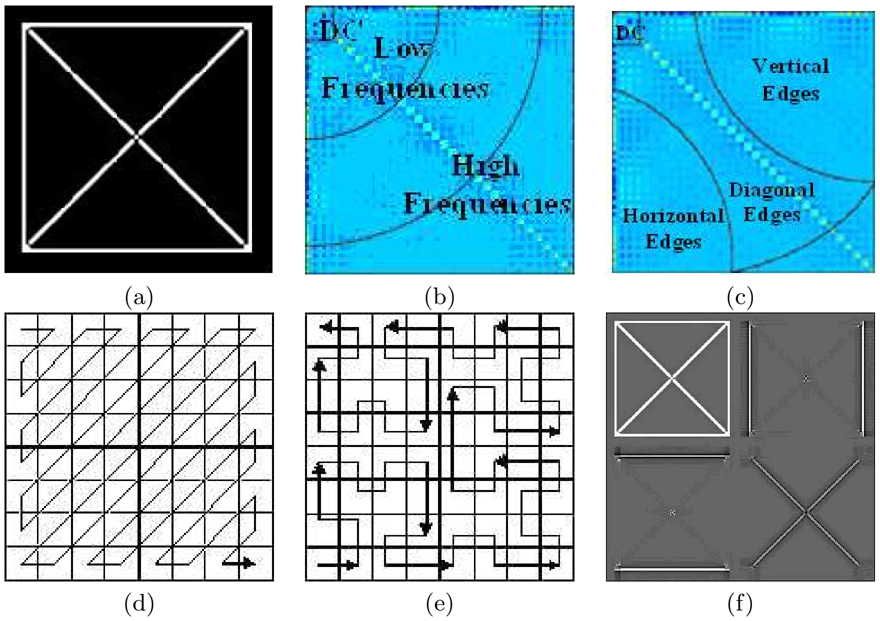

Figure 1. DCT multilevel energy bands: (a) Input image; (b) Energy band representation; (c) Extracted edges; (d) Vector with zigzag; (e) Peano scan matrix; (f) Level1 representation.

The structure presented captures the energy better than the normal DCT, and at least as well as wavelet decompositions used in image processing. The transformed coefficients are matched to higher level linear predicted neighboring coefficients, which result in an unstructured (non-Gaussian) distribution. Higher-order statistics are appropriate for measuring the coefficients for non-Gaussian processes.

The features are calculated using a mask, which, for a coefficient c, calculates the difference between c and its neighbors q, as shown in Figure 2. Similar methods have been used in pattern recognition [2] and steganography detection [1, 11] with wavelets. Higher-order statistics and predicted log errors are calculated across all of the mask values in order to create additional features.

Our classification methodology uses the features calculated from the DCT multilevel energy bands of JPEG images to separate the various embedding algorithms. This approach differs from other feature generation schemes (e.g., [1, 11]) that use different coefficients or wavelets. The most similar work to ours is that of Fridrich [6], which uses features that are specifically designed to distinguish classes of embedding algorithms, e.g., features that can distinguish an F5 embedding from an OutGuess embedding. Our features are developed for JPEG images, which makes

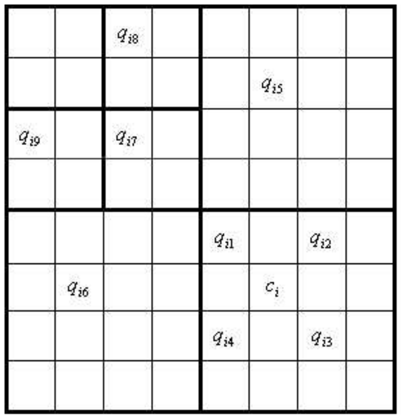

Figure 2. Target coefficients.

them applicable to the more general problem of anomaly detection as well.

4. SVM Multi-Class Classification

Multi-class classifiers are built on a winner-take-all set of class labels, each label representing one of the available classes. We employ a multi-class support vector machine (SVM) classifier, which separates classes by creating a hypersurface that maximizes the margins between all the classes.

SVMs have traditionally been applied to two-class or binary classification problems [15]. However, SVMs can be applied to multi-class classification. The techniques include: (i) the one-versus-all approach, which uses binary classifiers to encode and train the output labels; (ii) the one-versus-one approach, which uses a multi-class rule based on the majority-vote-wins approach; and (iii) training two-class classifiers and using voting and combinations of approximate posterior probabilities. Another approach to multi-class SVM classification is to train multi-class kernel-based predictors that use a compact quadratic optimization solution [5]. Our approach to SVM multi-class classification uses a one-versus-one majority-vote-wins strategy.

Figure 3 shows a multi-class SVM with support vectors (encapsulated in circles) and inter-class decision boundaries. SVM classification is performed by placing the classifying hyperplane, which separates the classes, in the center of the margin of separation. The margin of separation is calculated by locating the training points, x_i, that are closest to the opposing class and result in the largest margins of separation. Under this

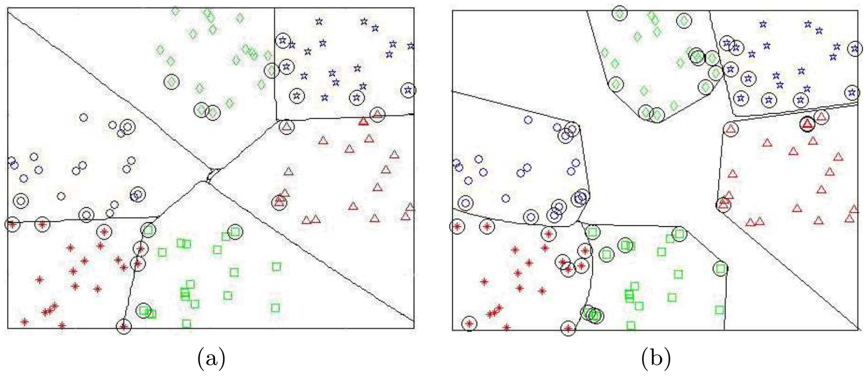

Figure 3. Multi-class SVM: (a) SVM without hyperplane; (b) SVM with hyperplane.

condition, the decision surface is referred to as the maximally separating hyperplane [3] (Figure 3(a)). The result shown in Figure 3(b), which is used to perform multi-class steganalysis, was obtained using the maximization of the margins of separation as the classification mechanism. This produces fewer false positives for each of the classes, but increases the number of anomalies when the data is unknown.

Our classification problem involves several aspects such as the amount of stego, the embedding method and the compression scaling factor. This leads to a complex multi-class classifier. In order to generate a multi-class classifier, a set of binary classifiers $g^1, ..., g^M$ is first constructed; next, each binary classifier is trained to separate one class from the rest; then, the classifiers are combined using a majority-vote-wins policy [3]. In the case of SVM, multi-class generalization involves a set of discriminant functions designed according to the maximal output. The majority voting strategy is used to implement the multi-class classifier [4]:

$$f(x) = \arg \max_{j=1,\ldots,M} g^j(x) \tag{1}$$

where $g^j(x) = \sum y_i \alpha_i^j K(x, x_i) + b^j$. The classification y_i provides the sign of the coefficients of x_i. The weight values α_i^j are proportional to the number of times the misclassification of x_i causes the weights to be updated. $K(x, x_i)$ is a radial basis kernel (RBF), and b^j is the bias vector. Each $g^j(x)$ value is a classifier's assignment for a sample, which may also be used to decide when a classification is too close to call and should be rejected.

The kernel function is used to transform input data sets in the feature space (which are not linearly separable) to the kernel feature space where the classification is performed. In general, the kernel function is explicitly specified (it implicitly defines the feature space). By defining the kernel function, the complexity of non-linear class separation is avoided not only when computing the inner product, but also when designing the learning machine. In this work, the training vectors x_i are mapped into a higher-dimensional space by the mapping function ϕ. The kernel function used is a Gaussian radial basis kernel function $< \phi(x)\dot{\phi}(x_i) >= K(x, x_i) = e^{|x-x_i|^2/\sigma^2}$[9].

To understand our approach, assume that the difference between the two largest $g^j(x)$ values is used as the measure of confidence in the classification of x. If the measure falls short of a threshold, the classifier rejects the pattern and does not assign it to a class (which produces an anomaly). The consequence is that a lower error rate is produced for the remaining patterns.

5. Results

This section presents the results based on testing data sets from the five tools (F5, JSteg, Model-Based Embedding, OutGuess, StegHide), and a clean data set. The experiments used a mixture of 1,000 (512×512 color JPEG) files comprising clean images and images created using the six embedding techniques described in Section 2.1. A total of 4,000 characters – equivalent to about one page of text – was embedded within each image file. The percentage of altered coefficients varied with the embedding method. The numbers of features used to represent the images were reduced from 120 to 40 features by eliminating features with similar correlations. The results in Table 1 were generated using five-fold cross validation where 80% of the data was used for training and 20% for testing; the test was conducted five times.

Table 1 shows the confusion matrix for the classification of the clean image set and five embedding method sets (F5, JSteg (JS), Model-Based Embedding (MB), OutGuess (OG) and StegHide (SH)). The matrix shows that the clean set is clearly separable from the remaining feature sets (Clean column and Clean row). In this multi-class classification, OutGuess and StegHide have the largest number of exemplars that are misclassified as each other. While these two methods are immune to statistical methods such as the chi-square test, they are vulnerable to higher-order statistics and transforms. These statistics, e.g., inertia, "compress" the energy bands of the DCT when the coefficients have not been modified and "expand" the energy bands when coefficients have been modified.

F5 and Model-Based Embedding also have mixed classification results. Therefore, we combined OutGuess and StegHide along with F5

Table 1. Confusion matrix for a six-class SVM classification.

	Actual					
Pred	Clean	F5	MB	OG	JS	SH
Clean	90.2 ± 4.5	3.4 ± 2.0	4.9 ± 2.2	1.4 ± 1.6	0.1 ± 0.0	0.0 ± 0.0
F5	4.2 ± 1.5	83.0 ± 5.4	6.7 ± 3.2	4.8 ± 1.1	0.2 ± 0.0	1.1 ± 0.9
MB	3.6 ± 2.3	16.8 ± 5.2	75.1 ± 9.1	2.4 ± 1.2	0.1 ± 0.0	2.0 ± 1.3
OG	0.4 ± 0.01	1.4 ± 1.6	0.4 ± 0.2	52.3 ± 12.2	6.6 ± 2.9	38.9 ± 7.6
JS	1.0 ± 0.5	3.4 ± 1.6	2.2 ± 2.0	6.8 ± 3.8	82.2 ± 5.8	4.4 ± 3.0
SH	0.6 ± 0.0	1.2 ± 0.7	1.7 ± 1.8	40.0 ± 7.0	7.1 ± 2.8	49.4 ± 10.9

and Model-Based Embedding to create a four-class classification (Table 2). Unlike OutGuess and StegHide, F5 and Model-Based Embedding produce DCT coefficients that are undetectable using sophisticated statistical measures. The features for F5 and OutGuess are not affected by the re-compression of the embedding. The statistical measures of inertia, energy and entropy show prominent features in the diagonal, vertical and horizontal energy bands, respectively. These findings stress the importance of separating the energy bands into the edge components and measuring each energy band with various statistics.

Table 2. Confusion matrix for a four-class SVM classification.

	Actual			
Pred	Clean	F5 & MB	OG & SH	JS
Clean	94.8 ± 3.3	2.4 ± 1.7	1.5 ± 0.4	1.3 ± 0.8
F5 & MB	4.5 ± 2.9	87.0 ± 7.6	6.5 ± 2.6	2.0 ± 1.8
OG & SH	3.2 ± 0.9	3.6 ± 2.0	90.7 ± 3.8	2.5 ± 2.2
JS	0.0 ± 0.0	4.0 ± 1.7	6.4 ± 2.4	89.6 ± 6.7

Because of similarities in the embedding techniques, the multi-class classifier was unable to identify the embedding methods in the six-class classification (Table 1). However, better classification results were obtained by recognizing similarities in the embedding techniques and performing a four-class classification (Table 2). The results in Table 2 show that combining OutGuess and StegHide produces a classification accuracy of 90.7% (up from roughly 50%). This allows the identification of the two embedding techniques. While the results for F5 and Model-Based Embedding are not as dramatic as those for OutGuess and StegHide, an increase in classification accuracy is achieved, which enables the two techniques to be distinguished from the other embedding techniques.

6. Conclusions

It is practically impossible to extract hidden data from a steganographic image without first identifying the embedding technique. The multi-class SVM-based technique presented in this paper can reliably determine if a JPEG image contains a hidden message and, if so, the type of lossy steganography that embedded the hidden data. The novel classification approach uses features constructed from DCT energy bands and engages a winner-take-all hierarchical classification structure.

Our future research will focus on two problems that must be solved after the embedding technique is identified. The first is to identify the algorithm that performed the embedding; this will require an additional multi-class classifier that is trained to recognize specific embedding algorithms. The second problem is to predict the amount of data embedded in an image and identify the image regions that contain the most embedded data; solving this problem would, of course, be crucial to developing techniques for extracting hidden messages.

Acknowledgements

This research was partially funded by the U.S. Air Force Research Laboratory, Information Directorate/Multi-Sensor Exploitation Branch, Rome, New York. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Air Force, U.S. Department of Defense or the U.S. Government.

References

[1] S. Agaian and H. Cai, Color wavelet based universal blind steganalysis, presented at the *International Workshop on Spectral Methods and Multirate Signal Processing*, 2004.

[2] R . Buccigrossi and E. Simoncelli, Image compression via joint statistical characterization in the wavelet domain, *IEEE Transactions on Image Processing*, vol. 8(12), pp. 1688–1701, 1999.

[3] C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, vol. 2(2), pp. 121–167, 1998.

[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, United Kingdom, 2000.

[5] J. Fridrich, G. Miroslav and H. Dorin, New methodology for breaking steganographic techniques for JPEGs, *Proceedings of the SPIE Symposium on Electronic Imaging*, pp. 143–155, 2003.

[6] J. Fridrich and T. Pevny, Determining the stego algorithm for JPEG images, *IEE Proceedings*, vol. 153(3), pp. 75–139, 2006.

[7] S. Har-Peled, D. Roth and D. Zimak, Constraint classification for multiclass classification and ranking, in *Advances in Neural Information Systems 15*, S. Becker, S. Thrun and K. Obermayer (Eds.), MIT Press, Cambridge, Massachusetts, pp. 785–792, 2003.

[8] S. Hetzl, StegHide (steghide.sourceforge.net).

[9] C. Hsu, C. Chang and C. Lin, A practical guide to support vector classification (www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf), 2005.

[10] T. Lane, P. Gladstone, L. Ortiz, L. Crocker, G. Weijers and other members of the Independent JPEG Group, JSteg (www.stegoarchive.com).

[11] S. Lyu and H. Farid, Steganalysis using color wavelet statistics and one-class support vector machines, *Proceedings of the SPIE Symposium on Electronic Imaging*, 2004.

[12] J. Platt, N. Cristianini and J. Shawe-Taylor, Large margin DAGs for multiclass classification, in *Advances in Neural Information Systems 12*, S. Solla, T. Leen and K. Muller (Eds.), MIT Press, Cambridge, Massachusetts, pp. 547–553, 2000.

[13] N. Provos, OutGuess (www.outguess.org).

[14] P. Sallee, Model-based steganography, *Proceedings of the Second International Workshop on Digital Watermarking*, pp. 154–167, 2003.

[15] B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, Massachusetts, 2002.

[16] StegoArchive.com (www.stegoarchive.com).

[17] A. Tewari and P. Bartlett, On the consistency of multiclass classification methods, *Proceedings of the Eighteenth Annual Conference on Learning Theory*, pp. 143–157, 2005.

[18] A. Westfeld, F5 – A steganographic algorithm, *Proceedings of the Fourth International Workshop on Information Hiding*, pp. 289–302, 2001.