EURASIP Journal on Applied Signal Processing 2005:17, 2749–2757

Image Steganalysis with Binary Similarity Measures

İsmail Avcıbaş
Department of Electronics Engineering, Uludağ University, 16059 Bursa, Turkey
Email: avcibas@uludag.edu.tr

Mehdi Kharrazi
Department of Electrical and Computer Engineering, Polytechnic University, Brooklyn, NY 11201, USA
Email: mehdi@isis.poly.edu

Nasir Memon
Department of Computer and Information Science, Polytechnic University, Brooklyn, NY 11201, USA
Email: memon@poly.edu.tr

Bülent Sankur
Department of Electrical and Electronics Engineering, Boğaziçi University, 34342 İstanbul, Turkey
Email: bulent.sankur@boun.edu.tr

Received 14 March 2004; Revised 10 May 2005; Recommended for Publication by Mauro Barni

We present a novel technique for steganalysis of images that have been subjected to embedding by steganographic algorithms. The seventh and eighth bit planes in an image are used for the computation of several binary similarity measures. The basic idea is that the correlation between the bit planes as well as the binary texture characteristics within the bit planes will differ between a stego image and a cover image. These telltale marks are used to construct a classifier that can distinguish between stego and cover images. We also provide experimental results using some of the latest steganographic algorithms. The proposed scheme is found to have complementary performance vis-à-vis Farid's scheme in that they outperform each other in alternate embedding techniques.

Keywords and phrases: steganography, steganalysis, universal steganalysis.

1. INTRODUCTION

Steganography refers to the science of "invisible" communication, where communication between two parties is undetectable by an eavesdropper. This is quite different from cryptography, where the goal is to make the content of the communications inaccessible to an eavesdropper. In contrast, steganographic techniques strive to hide the very presence of the message or communication itself from an observer. The subject is best explained in terms of the prisoner's problem [2], where Alice and Bob are two inmates who wish to communicate in order to hatch an escape plan. However, all communication between them is examined by the warden, Wendy, who will put them in solitary confinement at the slightest suspicion of covert communication. Specifically, in the general model for steganography we have Alice wishing to send a secret message m to Bob. In order to do so, she "embeds" m into a *cover object* c, to obtain the *stego object* s. The stego object is then sent through the public channel.

In a *pure steganography* framework, the technique for embedding the message is unknown to Wendy and shared as a secret between Alice and Bob. However, it is generally not considered as good practice to rely on the secrecy of the algorithm itself. In *private key steganography* Alice and Bob share a secret key, which is used to embed the message. The secret key, for example, can be a password used to seed a pseudorandom number generator to select pixel locations in an image cover object for embedding the secret message (possibly encrypted). Wendy has no knowledge about the secret key that Alice and Bob share, although she is aware of the algorithm that they could be employing for embedding messages. In *public key steganography*, Alice and Bob have private-public key pairs and know each other's public key.

As stated above, the goal of steganography is to communicate securely in a completely undetectable manner, such that an adversary should not be able to differentiate in any sense between *cover objects* (objects not containing any secret

message) and *stego objects* (objects containing a secret message). In this context, *steganalysis* is the set of techniques that try to defeat the very purpose of steganography, by detecting the presence of hidden communication. Thus steganalysis aims to distinguish between cover objects and stego objects. The art of steganalysis is becoming increasingly more important in computer forensics, for screening and tracking documents that are suspect of criminal activities, and for information security to prevent leakage of unauthorized data. Conversely, steganalysis can be used to assess the weaknesses of steganographic algorithms.

In the past few years we have witnessed a great expansion in fields of steganography and steganalysis. Several new steganography methods are being proposed each year, most of which are followed by new and improved steganalysis techniques for their detection. The steganalysis techniques proposed in the literature could be categorized into two groups. First we have *technique-specific* steganalysis methods, which attack a specific embedding algorithm, such as the approach proposed in [3]. The second type of techniques is blind to the embedding method and could be used with any embedding algorithm. They are called *universal steganalysis* techniques. It is the second category that we will be looking at in this paper. For a review on current steganography and steganalysis techniques the reader is referred to [3, 4, 5, 6, 7, 8].

As demonstrated previously in [9, 10], the embedding process on a document leaves statistical artifacts, which could be used to distinguish between stego and cover versions. The argument that watermarking and steganography leave telltale effects is common to all the steganalytical methods. For example, Harmsen and Pearlman [5] assume that steganography affects the histograms of the images, which they measure via the center of gravity of the characteristic function of the RGB probability density functions (pdf). Farid assumes that correlation across wavelets bands is affected [1], while Avcibas et al. demonstrate that image quality metrics are perturbed [9]. Fridrich et al. assume that histogram of DCT coefficients are modified, as in [3, 6], and that the lossless compression capacity of the LSB plane is diminished, as in [8].

In this paper, in order to capture these statistical artifacts and hence to determine the presence of hidden messages, we propose a set of binary similarity measures between successive bit planes. The basic idea is that, the correlation between the bit planes as well as the binary texture characteristics within the bit planes will differ between a stego image and a cover image. The seventh and eighth bit planes, and possibly others, are used to calculate these binary similarity measures. The proposed technique does not need a reference image and it works with both spatial and transform-domain embedding. The method is similar to that in [1, 9], in that it exploits intrinsic statistical properties of images to reveal the presence of steganographic content.

The rest of this paper is organized as follows. In Section 2 we review binary similarity measures. In Section 3 we describe our steganalysis technique. In Section 4 we give simulation results and conclude with a brief discussion in Section 5.

2. SIMILARITY MEASURES ON BINARY IMAGES

In the proposed steganalysis scheme we investigate statistical features extracted from the lower-order bit planes of images for the presence of hidden messages. Since each bit plane is also a binary image, we start by considering similarity measures between two binary images. We assume that any steganographic manipulation on an image will alter the patterns in the neighborhood of a bit in its bit plane as well as across the bit planes. In other words, the planar-quantal bit patterns will be affected. An evidence of such telltale effect can be found in the probability of bit transitions.

One might argue if straightforward bit plane correlations cannot be used for the steganalysis purpose. However, the evidence of any change is too weak if we measure only bit correlations across bit planes. In this study, we have found that it is more relevant to make comparisons based on *binary texture statistics*. Let $\mathrm{x}_i = \{x_{i,k} \mid,\ k = 1, \ldots, K\}$ be the sequences of bits representing the K neighborhood pixels ($K = 4$ and includes N, W, S, and E neighbors), where the index i runs over all the image pixels. We assume images of size $M \times N$. Let us define the 5-point stencil function $\chi_{r,s}$ as follows:

$$\chi_{r,s} = \begin{cases} 1 & \text{if } x_r = 0 \text{ and } x_s = 0, \\ 2 & \text{if } x_r = 0 \text{ and } x_s = 1, \\ 3 & \text{if } x_r = 1 \text{ and } x_s = 0, \\ 4 & \text{if } x_r = 1 \text{ and } x_s = 1, \end{cases} \tag{1}$$

based upon which we now define the agreement variable for the pixel x_i as $\alpha_i^j = \sum_{k=1}^{K} \delta(\chi_{i,k}, j)$, $j = 1, \ldots, 4$, $K = 4$, where $\delta(m, n)$ is the Kronecker delta function, which is defined as $\delta(m, n) = \begin{Bmatrix} 1 & m = n \\ 0 & m \neq n \end{Bmatrix}$. Obviously the α_i^j functions denote the central pixel-neighbor pixel transition types. The accumulated agreements are defined as

$$\begin{aligned} a &= \frac{1}{MN} \sum_i \alpha_i^1, \qquad b = \frac{1}{MN} \sum_i \alpha_i^2, \\ c &= \frac{1}{MN} \sum_i \alpha_i^3, \qquad d = \frac{1}{MN} \sum_i \alpha_i^4. \end{aligned} \tag{2}$$

These four variables $\{a, b, c, d\}$ can be interpreted as the one-step co-occurrence values of a binary image. Using the above definitions, several binary image similarity measures can be defined as shown in Table 1. A good review of similarity measures can be found in [11]. Almost all the measures in Table 1 have an intuitive interpretation; for example, the fifth measure dm_5, Sokal and Sneath's similarity measure 4, yields the conditional probability that LSBs of seventh bit plane is in the same state (1 or 0) given the state of the LSBs in the eighth bit plane. The measure is an average over both states acting as predictors and it has a range of 0 to 1. In this table, the measures dm_1 to dm_{10} are obtained for seventh and eighth bit planes of the image, separately. These measures form an adaptation of the classical binary string similarity measures, such as in Sokal and Sneath [12].

Table 1: Binary similarity measures.

Similarity measure	Description	Similarity measure	Description
Sokal and Sneath similarity measure 1	$dm_1 = m_1^{7\text{th}} - m_1^{8\text{th}}$, where $m_1 = \frac{2(a+d)}{2(a+d)+b+c}$	Variance dissimilarity measure	$dm_{10} = m_{10}^{7\text{th}} - m_{10}^{8\text{th}}$, where $m_{10} = \frac{b+c}{4(a+b+c+d)}$
Sokal and Sneath similarity measure 2	$dm_2 = m_2^{7\text{th}} - m_2^{8\text{th}}$, where $m_2 = \frac{a}{a+2(b+c)}$	Binary minimum histogram difference	$dm_{11} = \sum_{n=1}^{4} \min(p_n^7, p_n^8)$
Kulczynski similarity measure 1	$dm_3 = m_3^{7\text{th}} - m_3^{8\text{th}}$, where $m_3 = \frac{a}{b+c}$	Binary absolute histogram difference	$dm_{12} = \sum_{n=1}^{4} \lvert p_n^7 - p_n^8 \rvert$
Sokal and Sneath similarity measure 3	$dm_4 = m_4^{7\text{th}} - m_4^{8\text{th}}$, where $m_4 = \frac{a+d}{b+c}$	Binary mutual entropy	$dm_{13} = -\sum_{n=1}^{4} p_n^7 \log p_n^8$
Sokal and Sneath similarity measure 4	$dm_5 = m_5^{7\text{th}} - m_5^{8\text{th}}$, where $m_5 = \frac{a/(a+b)+a/(a+c)+d/(b+d)+d/(c+d)}{4}$	Binary Kullback-Leibler distance	$dm_{14} = -\sum_{n=1}^{4} p_n^7 \log \frac{p_n^7}{p_n^8}$
Sokal and Sneath similarity measure 5	$dm_6 = m_6^{7\text{th}} - m_6^{8\text{th}}$, where $m_6 = \frac{ad}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	Ojala minimum histogram difference	$dm_{15} = \sum_{n=0}^{511} \min(S_n^7, S_n^8)$
Ochiai similarity measure	$dm_7 = m_7^{7\text{th}} - m_7^{8\text{th}}$, where $m_7 = \sqrt{\left(\frac{a}{a+b}\right)\left(\frac{a}{a+c}\right)}$	Ojala absolute histogram difference	$dm_{16} = \sum_{n=0}^{511} \lvert S_n^7 - S_n^8 \rvert$
Binary Lance and Williams nonmetric dissimilarity measure	$dm_8 = m_8^{7\text{th}} - m_8^{8\text{th}}$, where $m_8 = \frac{b+c}{2a+b+c}$	Ojala mutual entropy	$dm_{17} = -\sum_{n=0}^{511} S_n^7 \log S_n^8$
Pattern difference	$dm_9 = m_9^{7\text{th}} - m_9^{8\text{th}}$, where $m_9 = \frac{bc}{(a+b+c+d)^2}$	Ojala Kullback-Leibler distance	$dm_{18} = -\sum_{n=0}^{511} S_n^7 \log \frac{S_n^7}{S_n^8}$

There are three categories of similarity measures derived from these scores.

(i) The first group consists of the computed similarity differences, $dm_i = m_i^{7\text{th}} - m_i^{8\text{th}}$, $i = 1, \ldots, 10$, across the 7th and 8th bit planes.

(ii) The second group consists of histogram and entropic features. We first normalize the histograms of the agreement scores for the bit planes (indicated by the superscript b):

$$p_j^b = \frac{\sum_i \alpha_i^j}{\sum_i \sum_j \alpha_i^j}, \quad b = 7, 8. \tag{3}$$

Based on these normalized *four-bin* histograms, we define the minimum histogram difference dm_{11} and the absolute histogram difference measure dm_{12}, binary mutual entropy dm_{13}, and binary Kullback Leibler distance dm_{14}, as also given in Table 1.

(iii) The third set of measures $dm_{15}, \ldots, dm_{18}$ are somewhat different in that we use the neighborhood-weighting mask proposed by Ojala [13]. For each binary image we obtain a 512-bin histogram based on the weighted neighborhood, where the score is given by $S = \sum_{i=0}^{7} x_i 2^i$ by weighting the eight-directional neighbors as shown in Figure 1. Defining S_n^7 as the count of the nth histogram bin in the 7th bit plane and S_n^8 the corresponding one in the 8th plane, after normalizing these 512-*bin* histograms, we can define Ojala minimum histogram difference dm_{15} and Ojala absolute histogram difference measure dm_{16}, Ojala mutual entropy dm_{17}, and Ojala Kullback-Leibler distance dm_{18} as given in Table 1.

1	2	4
128	256	8
64	32	16

Figure 1: The weighting pattern of the neighbors in the computation of Ojala score. For example, the score becomes $S = 2+4+8 = 14$ in the example where E, N, NE bits are 1 and all other bits are 0.

In Figure 2 we show how Stools algorithm modifies the LSB and 7-8 bit plane correlations in terms of the Ojala Kullback-Leibler distance (measure dm_{18} in Table 1) as a function of embedded message size. In the active warden case where message has to be embedded robustly, deeper bit plane

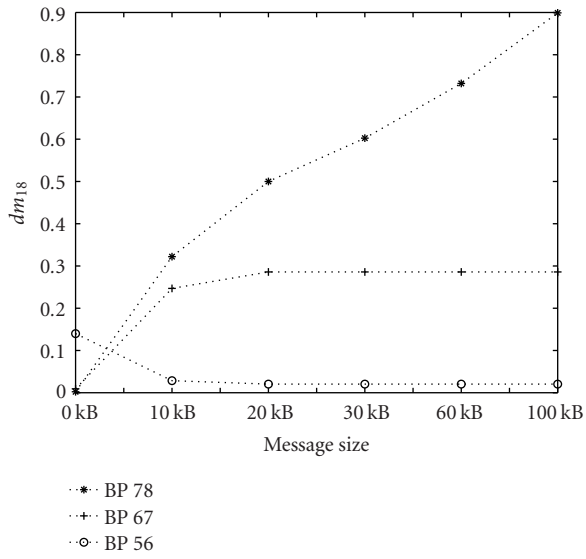


Figure 2: Variation of the Ojala Kullback-Leibler distance as a function of embedded message size in the Stools steganographic method. (Legend: BP 78 means bit planes 7 and 8, etc.)

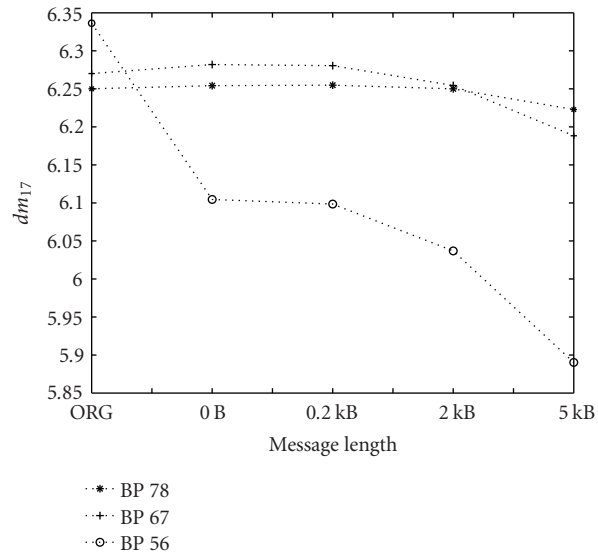


Figure 4: Variation of the bit plane correlations, measured with the measure dm_{17}, the Ojala entropy, as a function of embedded message length for F5 algorithm.

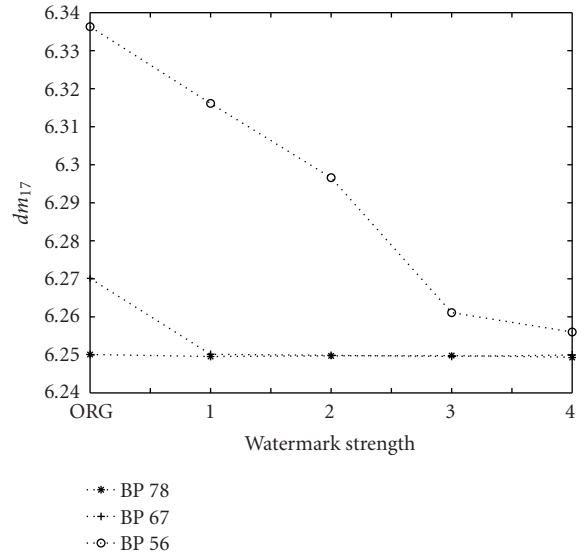


Figure 3: Variation of the bit plane correlations, measured with the Ojala entropy measure dm_{18}, as a function of embedding strength in Digimarc [16] algorithm.

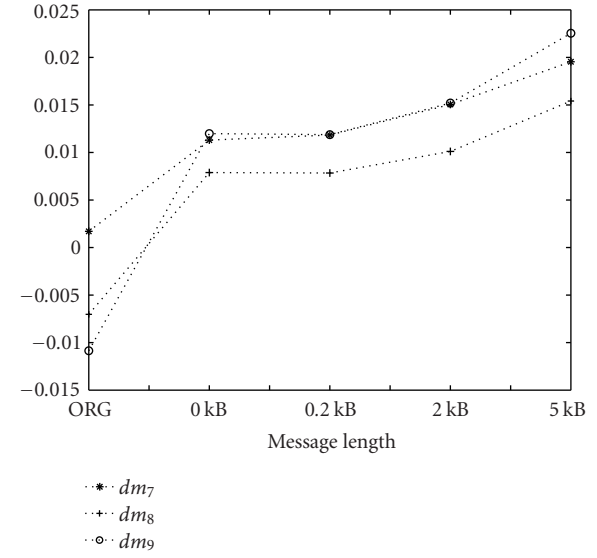


Figure 5: Variation of dm_7, dm_8, dm_9 measures across 7-8 bit planes as a function of embedded message length for F5 algorithm.

correlations (5-6 bit planes) should be taken into account. In Digimarc [14] example, as shown in Figure 3, we see the same monotonic trend in the Ojala entropy measure (dm_{18} in Table 1) as a function of watermark strength. In Figure 4, the effects of the F5 [15] embedding algorithm on the 5-6 bit planes (dm_{17}, the Ojala entropy in Table 1) are illustrated. Finally, variations of dm_7, dm_8, dm_9 measures in Table 1 across 7-8 bit planes as a function of embedded message length for F5 [15] algorithm are shown in Figure 5.

3. STEGANALYSIS TECHNIQUE BASED ON BINARY MEASURES

We hypothesize that binary similarity measures between bit planes will differ in their patterns between clean and stego images, that is, the statistics will be modified as a consequence of message embedding. This is the basis of our steganalyzer that aims to classify images as stego and cover images. In fact, embedding information in any bit plane

modifies the correlation between that plane and its contiguous neighbors. For example, for LSB steganography, one expects a decreased similarity between the seventh and the eighth bit planes of the image as compared to its unmarked version, due to randomization of the eighth plane. Hence, similarity measures between these two LSBs should yield higher scores in a clean image as compared to a stego image, as the embedding process destroys the preponderance of bit-pair matches. Note that the same procedure generalizes quite easily to detect messages in any other bit plane. Furthermore, our results indicate that we can even build steganalyzer for non-LSB embedding techniques like the recent F5 algorithm [15]. This is because a technique like F5 (and many other robust watermarking techniques, which can be used for steganography in an active warden framework [2]) results in the modification of the correlation between bit planes.

Classifier design

We have used support vector machines (SVM) classifier [16]. In support vector machine (SVM) [17], the underlying idea rests on the minimization of the training set error, or the maximization of the summed distances between the separating hyperplane and the subset of closest data points (the support vectors). For the training feature sets $(\mathbf{m}_i, y_i)$, $i = 1, \ldots, N$, $y_i \in [-1, 1]$, the feature vector $\mathbf{m}$ lies on a hyperplane given by $\mathbf{w}^T\mathbf{m} + b = 0$, where $\mathbf{w}$ is the normal to the hyperplane. A set of feature vectors is said to be optimally separated if no errors occur and the distance between the closest vectors to the hyperplane is maximal. The distance $d(\mathbf{w}, b; \mathbf{m})$ of a feature vector $\mathbf{m}$ from the hyperplane $(\mathbf{w}, b)$ is $d(\mathbf{w}, b; \mathbf{m}) = |\mathbf{w}^T\mathbf{m} + b|/\|\mathbf{w}\|$. The optimal hyperplane is obtained by maximizing this margin. There are a number of available implementations of SVM. We have used the freely available Libsvm [18] package.

The classifier training and testing procedures are as follows.

(1) An equal number of marked (with varying message lengths) and unmarked images are randomly chosen for the design step. Since the number of embeddable images varies as a function of both the message size and the steganographic algorithm, the number of stego images with a given message length used in the marked category of the training set is determined with their empirical statistics in mind. For example, given a specific method, if there are twice as many images with 1% message lengths as compared to 5% messages, then the training set will contain twice as many images with 1% message as images with 5% messages.

(2) The trained classifier is then tested against the remaining set of unseen unmarked and marked images which consists of images with 1% embedding, 5% embedding, ..., denoted respectively as 1P, 5P, ...

(3) The above procedure is repeated 5 times, resulting in 5 different classifiers and the average of the classifier performances is computed.

Table 2: The number of images in the database given the message length and the embedding type. The embedded message size is equal to 384, 1920, 3840, and 5760 bytes, respectively, for bit/pixel ratios from 1% to 15%.

P (bits/pixel)	LSB	LSB +/−	F5	Outguess−	Outguess+
1P (384B)	1800	1800	1798	1797	1800
5P (1920B)	1800	1800	1637	1516	1644
10P (3840B)	1800	1800	838	557	623
15P (5760B)	1800	1800	224	110	102

4. SIMULATION RESULTS

4.1. Experimental setup

An initial database consisting of 1800 natural images was used [19]. The images were converted to grayscale and the borders around them were cropped, resulting in images of size 640×480 pixels, after which they were recompressed with a quality factor of 75. This database was augmented with the stego versions of these images using 5 different embedding techniques, and different message lengths were employed. Since the actual steganographic capacity of a given image is dependent on the content of the image as well as the embedding technique used, we used a variety of message lengths to create our dataset.

4.2. Embedding methods and message lengths

The embedding methods were chosen on the basis of most current steganographic algorithms available. The embedding algorithms used in our experiments were LSB, LSB +/−, OutGuess−, OutGuess+, and F5. The LSB and LSB +/− techniques operate in the spatial domain, but with LSB the least-significant bit of each pixel value is flipped, whereas with LSB +/− the pixel values are incremented or decremented by 1. The second set of techniques which include OutGuess [20], with + and − flags, and F5 [15] operates in the JPEG domain by modifying the least significant bit of DCT coefficients.

There are different approaches in the literature in choosing the length of the stego message being embedded. Farid [1] chooses $n \times n$ pixels from the central region of a randomly chosen image. Another approach is to take constant length messages, say at 100, 500, or 1000 bits. But in both approaches the actual message size has no proportionality with the actual image size. In order to avoid these problems, we have used the following approach in defining the message length: we assumed that p bits could be embedded in each pixel value, regardless of the depth of the pixels, that is, 8 or 24 bit/pixel, where p is a fraction $0 < p < 1$. Thus the message length consists of a percentage point of the total number of pixels, and the length is independent of the type of image format, bmp or jpeg, but proportional to the size of the image. Furthermore, since the sizes of all images in our experiments were equal, the actual message lengths were also constant. Thus we considered four message lengths, 1%, 5%, 10%, and 15%, respectively denoted by the symbols 1P, 5P, 10P, and 15P. Table 2 below shows the arrangement in the

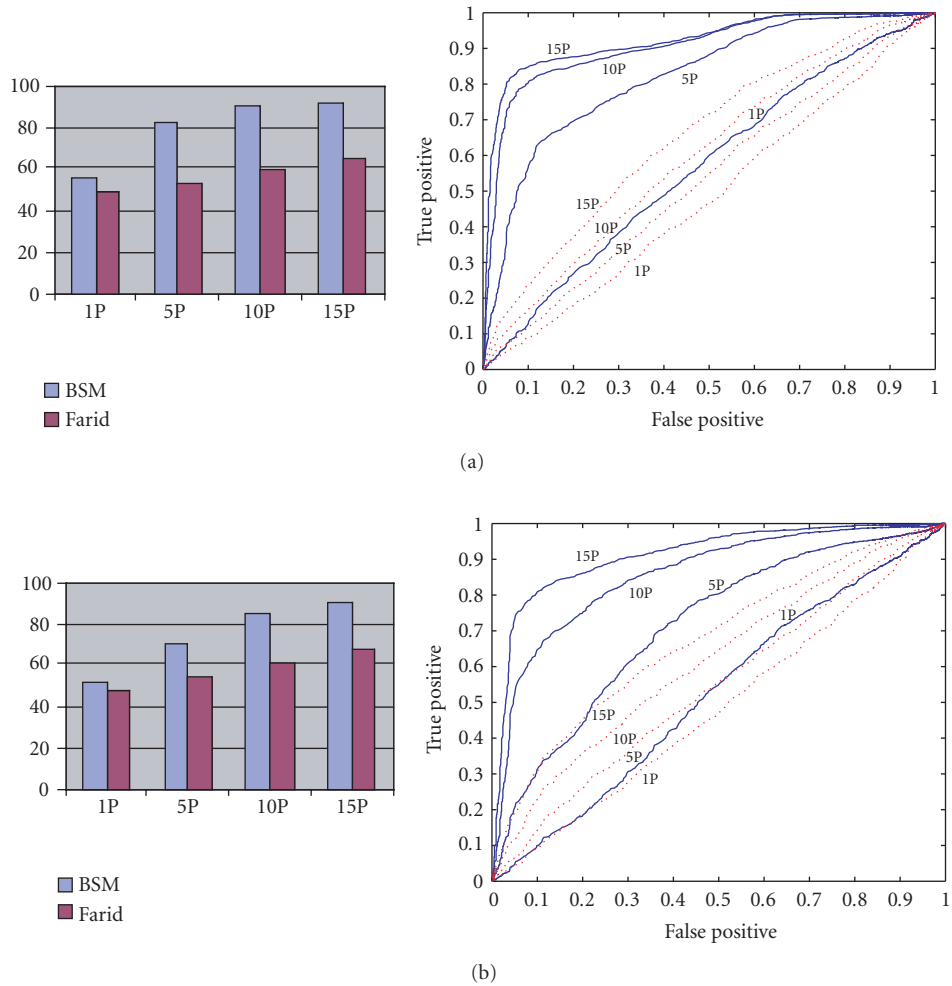


Figure 6: Comparison results, in the ROC plots; the solid lines are from BSM whereas the dashed lines represent Farid's technique. (a) LSB and (b) LSB +/−.

database. One can notice that as the message length grows, the number of images that can be accommodated decreases; in fact, with methods such as Outguess this decrease is quite sharp.

4.3. Classification results

Statistics from the original unmarked images as well as the stego images were obtained by computing the binary similarity measures, introduced in Section 2. Thus a vector of length 18 was obtained for each image. These vectors were then used to train and test the classifier, where we used 720 marked and 720 unmarked images in the training process. The classifier was trained with all embedding percentages from 1% to 15%. For example in the Outguess−, the classifier was presented with 720 unmarked and 720 marked images, where the marked images consisted of 318 images with 1P message, 298 with 5P, and 104 with 10P. Images with 15P messages were not used in this case since very few of them are available.

Table 3: Classification results using SVM.

	Outguess−	Outguess+	F5	LSB	LSB +/−
Accuracy: 1P	50	50.52	48.80	56.67	52.26
Accuracy: 5P	59.49	61.62	52.18	82.02	71.27
Accuracy: 10P	78.03	80.07	65.21	90.42	85.61
Accuracy: 15P	—	—	—	92.17	91.06

In Table 3, we give the test stage classification accuracy. Here accuracy is defined as the area under the ROC curve obtained from the classifier, where the ROC curves are obtained by first designing a classifier and then testing the data unseen to the classifier against the trained classifier at the same time moving the separating hyperplane. As the separating hyperplane is moved, the false alarm rate changes, and we get the corresponding detection rate. Also the obtained ROC curves for each embedding technique could be seen in Figures 6 and 7.

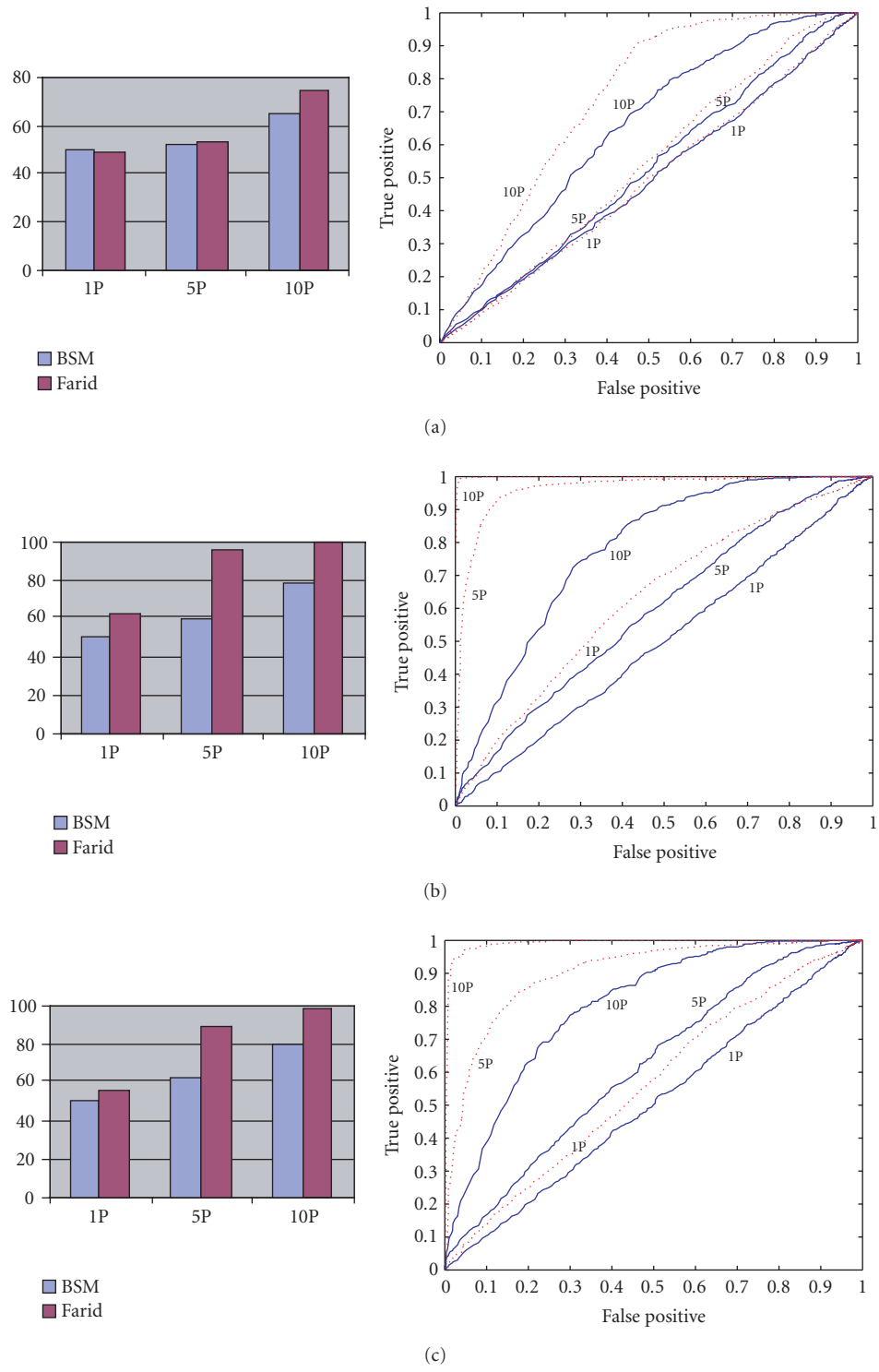


Figure 7: Comparison results, in the ROC plots; the solid lines are from BSM whereas the dashed lines represent Farid's technique. (a) F5, (b) Outguess −, and (c) Outguess +.

The most closely related publication to our work is by Farid [1] in which higher-order statistics of wavelet components are used for detecting hidden messages. But due to the fact that the results are presented differently a direct comparison was not possible. So in order to make a fair comparison between the two proposed techniques, we have used the publicly available matlab script from the website (www.cs.dartmouth.edu/~farid/) to calculate the proposed

feature set, and then used our design and testing process on the obtained features. The results in Figures 6 and 7 show that the two methods are close competitors in that the proposed BSM method proves superior for the LSB and LSB+/− embedding techniques while Farid's method proves superior in the case of the F5 and Outguess+/− techniques.

5. CONCLUSIONS

In this paper, we have addressed the problem of steganalysis of images. We have developed a technique for discriminating between cover images and stego images obtained from various steganographic methods. Our approach is based on the hypothesis that steganographic schemes leave telltale evidence between bit planes of lower significance, which in turn can be exploited for detection. The steganalyzer has been instrumented with binary image similarity measures and a classifier. Simulation results with commercially available steganographic techniques indicate that the proposed steganalyzer is effective in classifying stego and cover images.

Although tests have been run on LSB-based steganography, initial results have shown that it can easily generalize to the active warden case by taking deeper bit plane correlations into account. For example as in [9] we are able to detect Digimarc [14] when the measures are computed for higher-significance bit planes.

After this proof-of-concept design, the stegoanalyzer can be improved by judicious selection of the feature set in Table 1, for example, via SFFS (sequential floating feature search) algorithm. For the non-LSB techniques both image quality metrics [9] and binary similarity measures can be used jointly. Finally, given the fact that our algorithm and that of Farid [1] outperform each other for different steganographic methods and that neither one is uniformly superior to the other one, their complementary role should be exploited, for example, in a decision fusion scheme.

ACKNOWLEDGMENTS

This work was supported by AFRL Grant no. F30602-03-C-0091 and TÜBİTAK-NSF Project no. 102E018. We would also like to thank Mr. Mike Sosonkin for his help in coding parts of the proposed steganalysis technique.

REFERENCES

[1] H. Farid, "Detecting hidden messages using higher-order statistical models," in *Proc. IEEE International Conference on Image Processing (ICIP '02)*, vol. 2, pp. 905–908, Rochester, NY, USA, September 2002.

[2] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Proc. Advances in Cryptology (CRYPTO '83)*, pp. 51–67, Santa Barbara, Calif, USA, August 1983.

[3] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: breaking the F5 algorithm," in *Proc. 5th International Workshop on Information Hiding (IH '02)*, pp. 310–323, Noordwijkerhout, the Netherlands, October 2002.

[4] M. Kharrazi, H. T. Sencar, and N. Memon, *Image Steganography: Concepts and Practice*, Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore, Singapore, Republic of Singapore, 2004.

[5] J. J. Harmsen and W. A. Pearlman, "Steganalysis of additive-noise modelable information hiding," in *Security and Watermarking of Multimedia Contents V*, vol. 5020 of *Proceedings of SPIE*, pp. 131–142, Santa Clara, Calif, USA, January 2003.

[6] J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on JPEG compatibility," in *Multimedia Systems and Applications IV*, vol. 4518 of *Proceedings of SPIE*, pp. 275–280, Denver, Colo, USA, August 2001, Special Session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding.

[7] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color and gray-scale images," *IEEE Multimedia*, vol. 8, no. 4, pp. 22–28, 2001, Special Issue on Security.

[8] K. Sullivan, U. Madhow, S. Chandrasekaran, and B. S. Manjunath, "Steganalysis of spread spectrum data hiding exploiting cover memory," in *Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681 of *Proceedings of SPIE*, pp. 38–46, San Jose, Calif, USA, January 2005.

[9] İ. Avcıbaş, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," *IEEE Trans. Image Processing*, vol. 12, no. 2, pp. 221–229, 2003.

[10] H. Ozer, İ. Avcıbaş, B. Sankur, and N. Memon, "Steganalysis of audio based on audio quality metrics," in *Security and Watermarking of Multimedia Contents V*, vol. 5020 of *Proceedings of SPIE*, pp. 55–66, Santa Clara, Calif, USA, January 2003.

[11] V. Batagelj and M. Bren, "Comparing resemblance measures," in *Proc. International Meeting on Distance Analysis (DISTANCIA '92)*, Rennes, France, June 1992.

[12] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy. The Principles and Practice of Numerical Classification*, W. H. Freeman, San Francisco, Calif, USA, 1973.

[13] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[14] PictureMarc, Embed Watermark, v 1.00.45, Digimarc Corporation.

[15] A. Westfeld, "F5—a steganographic algorithm: high capacity despite better steganalysis," in *Proc. 4th International Workshop on Information Hiding (IH '01)*, vol. 2137 of *Lecture Notes in Computer Science*, pp. 289–302, Springer, Pittsburgh, Pa, USA, April 2001, Release 12 was used in experiments.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, New York, NY, USA, 2001.

[17] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

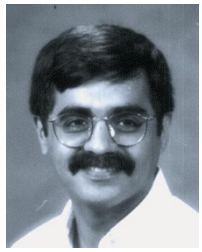
[18] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[19] Images were obtained from: http://philip.greenspun.com/.

[20] http://www.outguess.org/.

İsmail Avcıbaş received the B.S. and M.S. degrees in electronics engineering from Uludağ University, Bursa, Turkey, in 1992 and 1994, respectively, and the Ph.D. degree in electrical and electronics engineering from Boğaziçi University, İstanbul, Turkey, in 2001. He received a scholarship from The Scientific Council of Turkey, TUBITAK, BDP Program, and did research on image compression and steganalysis in the Department of Computer and Information Science, Polytechnic University, Brooklyn, NY, in 1999–2000. He is currently an Assistant Professor at the Department of Electronics Engineering, Uludağ University, Bursa, Turkey. His current research interests are in signal processing, data compression, steganalysis of audio-visual data, and pattern recognition.

Mehdi Kharrazi received his B.E. degree in electrical engineering from City College of New York. He received his M.S. degree in electrical engineering from the Department of Electrical and Computer Engineering at Polytechnic University, Brooklyn, NY, in May 2002, and is currently pursuing his Ph.D. His current research interests include multimedia and computer security.



Nasir Memon is a Professor in the Computer Science Department at Polytechnic University, New York. Professor Memon's research interests include data compression, computer and network security, and multimedia communication, computing, and security. He was an Associate Editor for the IEEE Transactions on Image Processing from 1999–2002. He is currently an Associate Editor for the ACM Multimedia Systems Journal, the IEEE Transactions on Information Forensics and Security, and the Journal of Electronic Imaging.

Bülent Sankur received his B.S. degree in electrical engineering from Robert College, Istanbul, and completed his M.S. and Ph.D. degrees at Rensselaer Polytechnic Institute, NY, USA. He has been teaching at Boğaziçi (Bosphorus) University in the Department of Electrical and Electronics Engineering. His research interests are in the areas of digital signal processing, image and video compression, biometry, cognition, and multimedia systems. Dr. Sankur has held visiting positions at the University of Ottawa, Technical University of Delft, and École Nationale Supérieure des Télécommunications, Paris. He was the Chairman of ICT '96: International Conference on Telecommunications and EUSIPCO '05: The European Conference on Signal Processing as well as Technical Chairman of ICASSP '00.