Detection of block DCT-based Steganography in gray-scale images

Constantine Manikopoulos, Yun-Qing Shi, Sui Song, Zheng Zhang, Zhicheng Ni, Dekun Zou
New Jersey Center for Wireless Networking and Internet Security (NJWINS)
Department of Electrical & Computer Engineering (ECE)
New Jersey Institute of Technology (NJIT); University Heights, Newark, NJ 07102
manikopoulos@njit.edu, shi@adm.njit.edu, sxs1148@njit.edu, zxz9622@oak.njit.edu, zn2@njit.edu,dz6@njit.edu

Abstract— **In this paper, a new approach, the steganography detection system (SDS), is proposed and applied to the detection of block DCT-based steganography in gray-scale images, segmented into 8x8 blocks. The differences in the coefficients of the block DCT transforms of the watermarked and unwatermarked images from the original are treated as features. SDS utilizes statistical preprocessing, over an observation region of each image, that generates feature vectors over the regions. These vectors are then fed into a simple neural network classifier. For the experiments conducted here, using 42 images, SDS achieves perfect detection rate with no misclassifications errors.**

Key words— steganalysis, data hiding, steganography, watermarking, statistical modeling, neural network classification.

I. INTRODUCTION

Recently, several steganography (data hiding, i.e., watermarking) techniques have been proposed and experimented with, as promising methods for copyright protection, authentication and other applications. These methods include algorithms based on LSB [1], DCT, block DCT [2], DWT, etc. One important requirement of data hiding is invisibility. There should be no visual artifacts that distinguish between the original image and the watermarked image. In light of this, given that an image may or may not be a stegoimage, there should be no visually trivial way to decide that issue. So then, could a third party design and utilize more sophisticated methods to identify the presence of steganography, for example, whether a watermark exists in the image at hand? And if it exists, a further question of interest may be posed: in which category does the data hiding method belong to?

In this paper, we respond to the first question, by proposing a new steganography detection technique for gray-scale images, based on statistical preprocessing and neural network classification. This technique was tested experimentally on a few dozen images that may or may not be watermarked, with excellent success. For this first experiment, the watermarking method used is based on the block DCT transform. This method was chosen because of its importance and prominence; the current image compression standard, JPEG, is itself based on the block DCT transform.

II. DATA GENERATION

The following is a brief description of the type of block DCT data hiding algorithm [2] that we employed. For a gray image, for example 'Lena' (512x512), we first segmented the image into non-overlapping 8 pixels x 8 pixels (8x8) blocks. This created a total of 64x64 blocks, altogether, for this image. Next, the DCT transform was applied on each block. The data, in the form of a random bit pattern of size N_D, here, were then embedded into the three largest AC coefficients, in the upper-left corner of the resulting matrix of transform coefficients (positions 2,9,10), according to the formula $V' = V + \alpha\omega_i$; here, $V(V')$ is the value of the AC coefficient before (after) embedding, $\omega_i \in \{-1,1\}$ represents the data at hand to be embedded and α is an adjustable weight given the value of either α=6, for the watermarked image, or α=0, for the unwatermarked image case. After that, the inverse DCT (IDCT) transform was applied on each block to generate the corresponding watermarked and unwatermarked images. This procedure was applied to 42 gray-level images, of different levels of complexity; of these, 14 are of 256x256, 25 of 512x512 and 3 of 1024x1024 resolution, respectively.

Thus, we have three versions of each of these images: the original, the watermarked, and the unwatermarked (but DCT-IDCT processed) images; the latter is essentially the original image but with some noise superimposed, due to the round-off errors associated with the computational processing of the applied DCT and IDCT transforms. The unwatermarked image provides here the reference of what effect watermarking of a message with zero embedded bits would create.

0-7803-7713-3/02/$17.00 © 2002 IEEE

Next, all three versions of the images were transformed with the same block DCT, separately, that is each image was first segmented into 8x8 blocks and then transformed by the DCT, block by block. Each block generates an 8x8 block of transform coefficients. We then subtracted the 8x8=64 transform coefficients of the original image from the corresponding coefficients of, on the one hand, the watermarked image, and, on the other hand, the unwatermarked image. Thus, for each block of an image, we generated 64 differences for the DCT transform coefficients of the watermarked vs. original as well as the unwatermarked vs. original comparisons, respectively. These differences, in each transform coefficient position, will be treated as features, in the statistical analysis that follows. So, the difference in position i of the 8x8 transform matrix, will correspond to feature X_i.

III. FEATURE VECTOR FORMATION

Without explicit statistical processing: In this manner, for each 8x8 block of every image, one value for each feature is generated for the watermarked, W, and the unwatermarked, U, versions, resulting in 64 values for each of the two. In principle, these U and W collections of values may be thought of as vectors of features, one vector for each block; these vectors could be used directly to train a classifier, for example, a neural network classifier, to distinguish between U and W. The authors believe that this approach would be successful for the task at hand, but not promising for other more difficult steganalysis challenges. A method, with more promise, that incorporates statistical processing explicitly, has been designed and is described below.

With explicit statistical processing: To incorporate statistical processing, statistical information needs to be collected for each feature. Thus, feature values associated with many blocks were collected for analysis. In this experiment, we aggregated feature values from 64 block regions; each region consists of an 8 block x 8 block rectangular section (or window) of observation; this amounts to 64 observation values for each feature. These 64 values for each feature X_i were organized into a probability density function (PDF), represented by a histogram, with values ranging from a minimum, $X_{i,\,min}$, to a maximum, $X_{i,\,max}$. A uniform partition into 64 equal size bins was employed between these two extremes. For an image of 512x512 resolution, there exist 64 regions (or windows) of observation, resulting into the creation of 64 different PDFs for the watermarked, termed W-PDFs, and an equal number of 64 PDFs, for the unwatermarked images, termed U-PDFs. After processing all 42 images, a total of 2592 U-type and 2592 W-type PDFs were generated, for each feature. Six example PDFs, for the 256x256 image resolution, are shown in Fig. 1 below. The PDFs for the 512x512 and 1024x1024 resolutions are similar. In each figure both the U and W-type PDFs are plotted for easy comparison. The character of most PDFs found is exemplified by those of parameters (features) 2, 16, 32 and 64, shown here. They show the U-type PDF consisting of a central peak at approximately the zero position, representing the round-off noise of the DCT-IDCT processing. The W-typed PDF is exemplified by two distinct peaks: for parameter 2, which carries most of the embedded information here, the peaks are located about 6 units away from zero, as they should, according to the embedding algorithm (similarly found in parameters 9 and 10); for parameters 16, 32 and 64, the twin peaks are closer to 0 and smaller, most

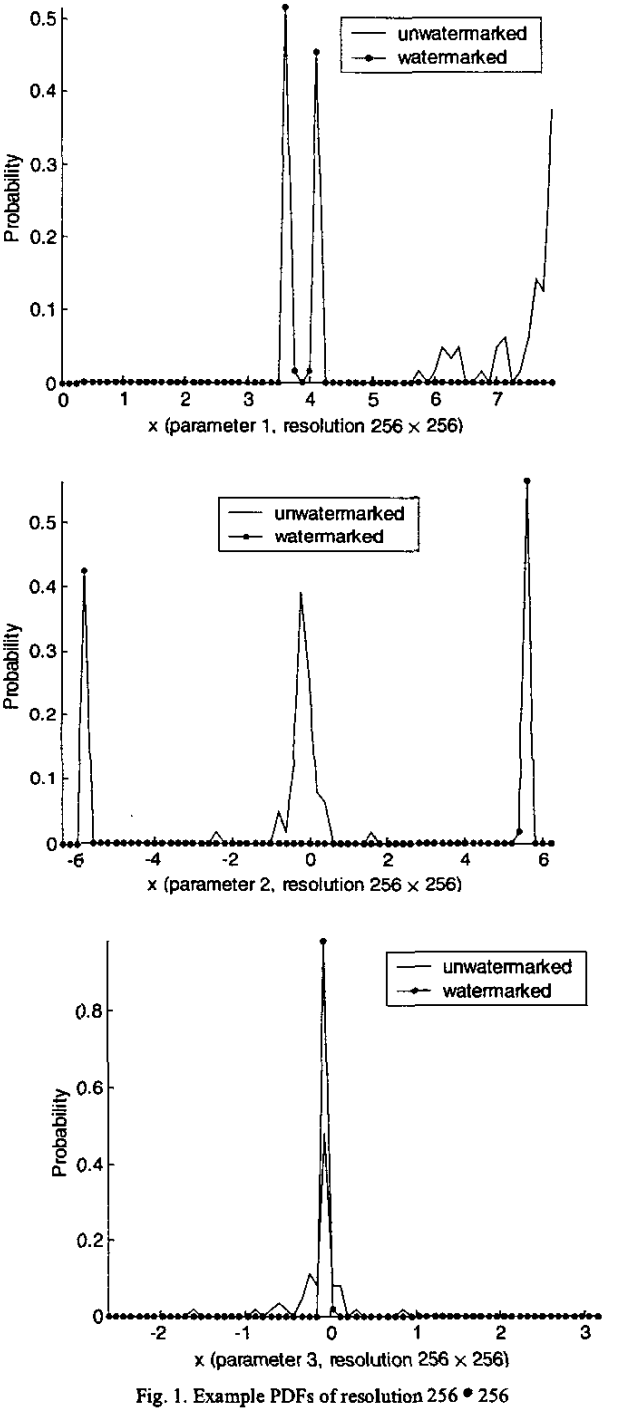
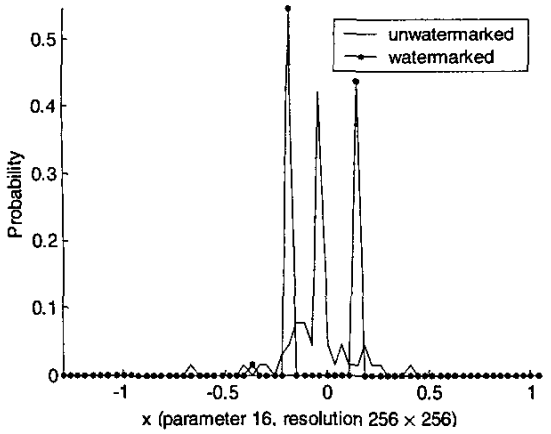
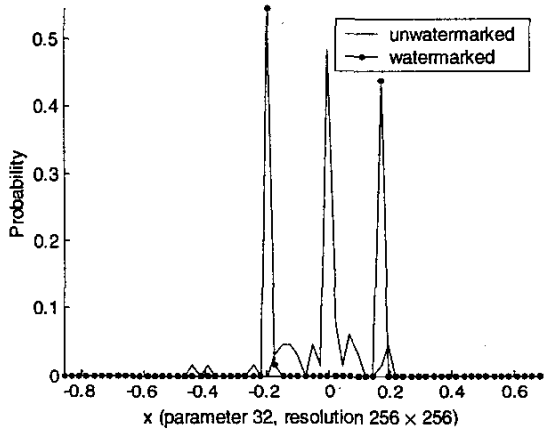


Fig. 1. Example PDFs of resolution 256 • 256





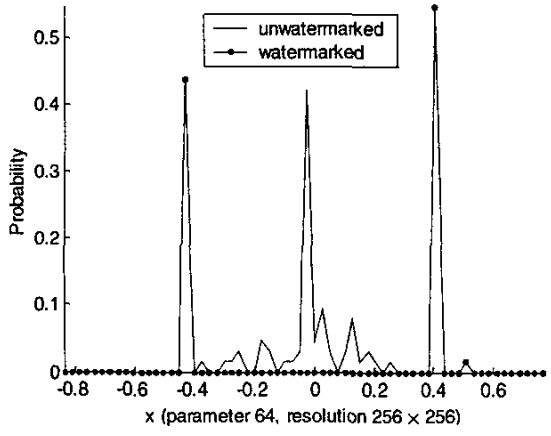


Fig. 1. Example PDFs of resolution 256 * 256 (cont'd)

likely representing some leakage of energy from the main peaks of parameter 2. Significantly, the tails of the U-type PDFs sometimes extend into the peaks of the W-type PDFs. This means that single parameter thresholding classification could be error prone. However, it is most significant that, in general, the U-type and the W-type PDFs are very different from each other. This means that statistical methods that capitalize on these differences can be very effective, especially, if many or all of the features were utilized in unison in arriving at the classification decision. This is exactly how the technique proposed here and the resulting tool, termed Steganography Detection System (SDS), achieves its high rate of success.

IV. STATISTICAL MODELING

Next, the average of all U-type-PDFs, for a particular feature, was computed, separately, for each feature and for each image resolution class. The resulting average PDF, U_{ave}, represents, statistically, the nominal unwatermarked feature, for that image resolution. Subsequently, each generated PDF, be it a U-type or a W-type, is statistically compared to U_{ave} by computing a similarity metric for it; this provides a single similarity scalar value, S_i, for each feature (or parameter) i, that may range from approximately –1 (representing a region very different than U) to approximately +1 (representing a region very similar to U). In other words, the status of the image over each observation window (region) is represented by 64 similarity values, according to how similar it is to the unwatermarked image.

Let S be the sample space of a random variable and events $E_1, E_2, \ldots, E_k$ a mutually exclusive partition of S. Assume p_i (p_i') is the expected reference probability (observed frequency) of the occurrence of the event E_i, during a given time interval or observation window. Many similarity metrics have been studied in previous work [3], however, the one used here, Q, is shown below:

$$Q = [0.5 * \sum_{i=1}^{k} |p_i' - p_i| + \max_{i=1}^{k} (|p_i' - p_i|)] \qquad (1)$$

The metric given in (1) above is a variant of the Kolmogorov-Smirnov (K-S) type of metrics that have the general advantage of being distribution independent [4].

Thus, feature vectors were generated, of another kind, each vector consisting of 64 similarity values, one for each feature, as components. These vectors statistically represent the windows (regions) of observation. There are evidently, two kinds of feature vectors, the unwatermarked U-type, labeled as +1, and the watermarked W-type, labeled as –1. These labeled U and W vectors can be used for training (67% of the total number) and validating (33% of the total number) the classifier. In this work, a neural network classifier was employed.

V. NEURAL NETWORK CLASSIFICATION

Neural network classifiers have been widely considered as an efficient approach to classify challenging patterns. However, here, statistical preprocessing has generated easily discernible and distinguishable U and W patterns. Thus, only

the simplest of the neural net classifiers, such as the perceptron, need be utilized. The perceptron [3], Fig. 2, is the simplest form of a neural network used for the classification of *linearly separable* patterns. The PDFs for parameter 2 indicate that this is indeed the case here, even if classification is based on that one parameter alone. However, in DCT-based image steganalysis, although some parameter may enable effective classification by itself, there is no a-priori guarantee that this will be the case; and even if it is, there appears to be no a-priory way to know which parameter will do that.

The perceptron consists of a single neuron with adjustable synapses and threshold. This is the smallest of the neural networks possible, for a given number of inputs. An advantage of the perceptron is that it is guaranteed to converge

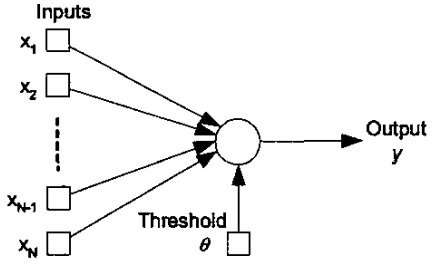


Fig. 2. Perceptron architecture

and should do so fairly quickly, that is within a small number of training epochs. In fact, the convergence of this perceptron, for the three image resolutions, 256x256, 512x512 and 1024x1024, is shown in Fig. 3, below. It is seen that for all three resolutions convergence is achieved within approximately 15 epochs, indeed a small number of epochs. The 512x512 resolution, which contains the largest number of training samples per epoch, clearly reaches lower MSR error levels sooner.

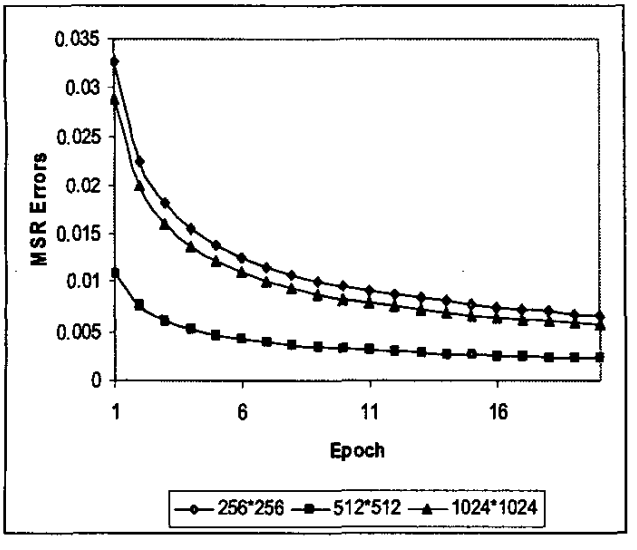


Fig. 3. The MSR vs. the number of training epochs

VI. CLASSIFICATION RESULTS

The MSR and misclassification rates of the classifiers are tabulated in Table I, below, at the end of the training period. It is seen that, for all resolutions, low MSRs are reached. The misclassification rate is the rate that unwatermarked data may be judged as watermarked (false positives) or watermarked data identified as unwatermarked (false negatives).

TABLE I. The classification results

Image Resolution	MSR Error	Mscl. Rate
256 * 256	0.00481	0
512 * 512	0.00172	0
1024 * 1024	0.00370	0

From the table above it may be concluded that the feature vectors, derived after statistical preprocessing, are easily identified by the classifier so that no misclassification errors occur.

VII. CONCLUSION

A new approach, the steganography detection system (SDS), is proposed and applied to the detection of block DCT-based stegoimages. SDS utilizes statistical preprocessing, over an observation region of the image, followed by neural network classification, to achieve perfect detection rate with no misclassifications, for the experiments conducted here, using 42 gray-scale images.

ACKNOWLEDGEMENT

Our research was partially supported by a Phase I and Phase II SBIR contract with the US Army. We would also like to thank OPNET Technologies, Inc.™, for providing partial support for the OPNET simulation software.

REFERENCES

[1] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," *In Proc. IEEE Int. Conference on Image Processing,* vol. 2, pp. 680-683, 1997.

[2] J. Huang, Y. Q. Shi, "An Adaptive image watermarking scheme based on visual masking," *Electronics Letters,* 1998, 34 (8), pp. 748-750.

[3] Z. Zhang and C. Manikopoulos, "Neural Networks in Statistical Anomaly Intrusion Detection," *Journal of Neural Network World,* Vol. 3, 2001, pp. 305-316.

[4] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, J. Ucles, "HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification," *CD-ROM Proceedings of the 2nd Annual IEEE Systems, Mans, Cybernetics Information Assurance Workshop,* West Point, NY, June 2001.