

Measures for Classification and Detection in Steganalysis

Sujit Prakash Gujar
Dept of CSA, IISc.
Bangalore, India.
sujit@csa.iisc.ernet.in

C E Veni Madhavan
Dept of CSA, IISc.
Bangalore, India.
cevmm@csa.iisc.ernet.in

Abstract—*Still and multi-media images are subject to transformations for compression, steganographic embedding and digital watermarking. We propose new measures and techniques for detection and analysis of steganographic embedded content. We show that both statistical and pattern classification techniques using our proposed measures provide reasonable discrimination schemes for detecting embeddings of different levels. Our measures are based on a few statistical properties of bit strings and wavelet coefficients of image pixels.*

Keywords : *Steganography, Steganalysis, SVM, Wavelets*

I. INTRODUCTION

Steganography is the art and science of secret communication, aiming to conceal the existence of the communication. *Steganalysis* is the art of seeing unseen. With advent of computers, hiding information inside digital carriers, especially multi media files like audio (.wav) files, images(.bmp, .pnm, .jpg), is becoming popular ([1], [2]). Digital images are most common sources for hiding message. The process of hiding information is called embedding. Least Significant Bit (LSB) embedding is the most widely used steganographic technique. In LSB embedding, the LSB of uncompressed images are replaced with the message bits. The amount of embedding (the number of bits embedded) referred to as *level*, is given as a percentage of the total number of pixels.

Some of the powerful methods for the analysis of steganographic images are [3], [4], [5]. We propose new measures and techniques for detection and analysis of steganographic embedded content. We show that both statistical and pattern classification techniques using our proposed measures provide reasonable discrimination schemes for detecting embeddings of different levels. Our measures are based on a few statistical properties of bit strings and wavelet coefficients of image pixels.

In Section II, we explain our approach towards classification of given data based on a feature vector consisting of statistical measures and using Support Vector Machine (SVM) tools. In Section III we propose the use of wavelet transforms for steganalysis. Our results presented in Section II and III show the efficacy of our measures in discriminating different levels of embedding. We conclude with our plans for improved and finer steganalysis in section IV

II. CLASSIFICATION BASED ON STATISTICAL MEASURES AND SVM

A. Classification of different types of files

Image steganography is a kind of transformation of a *cover image* and embedded data. As a first step we establish the power of our feature vector of measures based on statistical properties of bit strings in discriminating a variety of standard file types. Then we explore the possibility of discriminating images with different levels of embeddings. Once the level of embedding is determined to reasonable accuracy, we can proceed to the next step of *location* of embedded bits by other statistical and combinatorial techniques.

We use a statistical feature space. We propose a vector of statistical measures [6] for this purpose. Our feature vector $\mu \in \mathbb{R}^9$ consists of nine statistical measures. We consider a bit string S of size $32 * n$ bits as concatenation of n 32 bit words, S_i $i = 1, \dots, n$. We define the measures $\mu(S_i) = \langle \mu_1(S_i), \dots, \mu_9(S_i) \rangle$ for the words S_i and define the measure for entire string S , namely $\mu(S)$ as a weighted sum of the measures $\mu(S_i)$. The measures are as follows.

μ_1 : *Weighted sum of the of k -gram frequencies.* Let $f(k, j)$ denote the overlapping frequency of the k -gram binary pattern of the integer j in S_i . For example $f(4, 3) =$ number of occurrences of the pattern $\langle 0011 \rangle$ in S_i . For a 32 bit word W , we define

$$\mu_1(W) = \sum_{k=1}^4 (\max_j (f(k, j)) - \min_j (f(k, j))) 2^{4(k-1)}$$

We expect the measure μ_1 to be smaller for random strings as compared to non-random strings.

μ_2 : *Weighted sum of run lengths.* Let the vector $\langle l_1, l_2, \dots \rangle$ denote the sequence of run lengths of 0's and 1's in a 32 bit word W . Then we define,

$$\mu_2(W) = \sum 2^{c_i l_i}$$

where c_i are specifically chosen weights. We set $c_i = 1 \forall i$, without loss of generality. For random strings, we expect the

measure μ_2 to be smaller compared to non-random strings, since one expects very few long runs.

μ_3 : *Weighted sum of byte-wise hamming weight transition.* Let $W = \langle b_0, b_1, b_2, b_3 \rangle$, where b_i 's are the bytes of the 32 bit word. Let $\#1(b)$ denote the number of 1's in a byte b . Then we define,

$$\mu_3(W) = 2^{\#1(b_0)} + 2^{\#1(b_0 \oplus b_1)} + 2^{\#1(b_1 \oplus b_2)} + 2^{\#1(b_2 \oplus b_3)}$$

For random strings, we expect μ_3 to be higher than for non-random strings. It is also possible to define the measure μ_3 with respect to overlapping bytes in a word, to measure the smoothness/suddenness of transitions.

μ_4 : *Fourier transform of the autocorrelation function of the sequence bits in W .* Let $W = \langle a_0, \dots, a_{31} \rangle$ be a 32 bit word. The autocorrelation function $A(W)$ is the sequence $A(W) = \langle c_0, \dots, c_{31} \rangle$ where $c_i = \sum_{j=0}^{31-i} a_j \cdot a_{j+i} \pmod{32}$, $i = 0, \dots, 31$. The discrete Fourier transform $F(A(W))$ is given by the sequence $F(A(W)) = \langle f_0, \dots, f_{31} \rangle$; where $f_k = \sum_{j=0}^{31} c_j \omega^{jk \pmod{32}}$ $k = 0, \dots, 31$. Here ω is a 32^{nd} root of unity. Finally, the measure $\mu_4(W)$ is a root mean square average of F and is given by,

$$\mu_4(W) = \left(\sum_{j=0}^{31} |f_j|^2 \right)^{1/2}$$

For random strings, we expect μ_4 to be smaller than for non-random strings.

μ_5 : *Weighted Hadamard transform.* Using an 8×8 Hadamard matrix (H) and the operation $y = Hx$, where x is 8×1 bit vector, we get measure μ_5 . x is single data byte. When the Hadamard transform is applied on image data, x is taken as the bit string corresponding to a pixel value.

$\mu_6, \mu_7, \mu_8, \mu_9$: These measures are based on the weighted entropy measures $-\sum p_i \log p_i$ where p_i 's are probabilities of non-overlapping occurrences of 1,2,3,4 grams in string S .

Thus given a file S of some data, we compute the feature vector $\mu(S)$ as capturing the statistical characteristics of the bit string corresponding to S . We note that the statistical properties such as k -gram frequencies, run lengths, auto-correlation and entropy together are powerful features that discriminate a wide variety of non-random data. In the following we demonstrate this by classification based on our feature vector.

SVM (*Support Vector Machine*) is a powerful tool for pattern classification. With introduction of kernel tricks in SVM, it has become a very popular in machine learning community. In some cases, the given data is not directly classifiable. Such cases can be solved by transforming the given data to higher dimensional space in such a way that in transformed domain, the classification is much easier. Kernel tricks help this without actually transforming features to

TABLE I
CONFUSION MATRIX FOR DATA CLASSIFICATION

	jpeg	bmp/pnm	zip	gz	txt	ps	pdf	c
jpeg	0.9	0.05	0.05	0.0	0.0	0.0	0.0	0.0
bmp/pnm	0.0	0.9	0.0	0.0	0.0	0.0	0.05	0.05
zip	0.0	0.0	0.6	0.35	0.0	0.0	0.05	0.0
gz	0.0	0.0	0.1	0.9	0.0	0.0	0.0	0.0
txt	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
ps	0.0	0.0	0.0	0.0	0.05	0.95	0.0	0.0
pdf	0.0	0.0	0.6	0.05	0.0	0.05	0.3	0.0
c	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.95

higher dimensional space.

We use the feature vector μ defined above. For training of SVM, we measure statistics on 2000 words (8000bytes) of 30 different files to get 30 different values μ for each class. For testing, we measure statistics on 2000 words of 20 different files from each class. Though we have used measures calculated on 2000 words, our experiments shows that even 400 words are sufficient for testing a data for classification. The SVM tool is obtained from <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>. We used the most widely used 'Gaussian kernel' for SVM. For avoiding some features dominating the classification, we scale each measure to zero mean, unit variance. We studied the following 8 different classes:

1. jpeg
2. bmp/pnm
3. zip files
4. gz files
5. text files
6. ps files
7. pdf files
8. c files.

We present in Table I our classification results in the form of confusion matrix. The i_j^{th} entry is the probability of a test data belonging to class i and being classified as class j . We see from the Table that in all but two of the eight cases, the classification accuracy is near 1. We used a total of 180 files for testing and achieved overall accuracy of 82.22%.

B. Analysis of LSB planes from Stegoed and non-Stegoed Images

In above experiments, we measured statistics on the whole sequence of bits of the given data. An embedding operation is performed on LSB of an image. So to detect perturbation due to steganographic operation, we measure statistics only of LSB of images. In this direction, we first consider only two classes : one is LSB obtained from *non-stegoed image* and the other is LSB obtained from *images with 50% embedding*. In our experiments we use a random embedding instead of using any particular steganographic tool. We are conducting separate studies on different types of tools. The feature vector μ defined above is computed on LSB of 30 images from both classes. (total of 180 = 30*3(colors/images)*2 classes). Out of these, 150 were used for training SVM and 30 for testing. Thus we have two classes :

1. LSB plane of non-Stegoed image.
2. LSB plane of stegoed image.

We present the results in a confusion matrix form in Table II

TABLE II
CONFUSION MATRIX FOR 2 CATEGORY LSB CLASSIFICATION

	non-Stego	Stegoed Image
non-Stego	0.67	0.33
Stegoed Image	0.0	1.0

The overall accuracy is 85%. We next consider a 4 category classification problem.

The different classes are :

1. LSB plane of non-Stegoed image.
2. LSB plane of 25% stegoed image.
3. LSB plane of 50% Stegoed image.
4. LSB plane of 75% stegoed image.

The confusion matrix for this experiment is in Table III,

TABLE III
CONFUSION MATRIX FOR 4 CATEGORY LSB CLASSIFICATION

0.6	0.0	0.33	0.07
0.0	0.6	0.27	0.13
0.0	0.0	0.4	0.6
0.0	0.0	0.0	1.0

The overall efficiency is 65%. Thus, this experiment alone is not sufficient for detection of levels of embedding. Hence we take another alternative approach.

III. ANALYSIS OF IMAGES USING WAVELET TRANSFORMS

Our feature vector μ considers a linear sequence of bits as input. However, image properties are in general captured more accurately by two dimensional transforms. Our goal is to classify images accurately under different levels of embedding. The approaches in Section II-A and II-B serve as good handles in this direction. To further enhance our understanding of the effects of embedding, we study the behavior of wavelet coefficients. Here, a basic assumption is that the steganographic algorithm is known.

We consider the 2nd level LL sub-band coefficients, since most of the energy gets concentrated in this sub-band. The 2nd level LL coefficients of 4 * 4 image will be if image is

$$\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} \text{ wavelet } 2^{nd} \text{ level LL coefficient is}$$

$$\frac{a + b + c + d + e + f + g + h + i + l + m + n + o + p}{4}$$

(Note : 2nd Level LL sub-band size is $\frac{1}{4}$ th of the original image size in both directions.)

For our experiments, we use 10 images which do not contain any hidden information.

Call the set of these images $I = \{I_j : j = 0, 1, 2, \dots, 9\}$

The set of images of interest is,

$\mathbf{T} = \{I_{jk} : j = 0, 1, 2, \dots, 9; k = 0, 10, 20, 30, \dots, 100\}$. I_{jk} refers to the j^{th} image in I with $k\%$ embedding. Given an image I_{jk} , the goal of the steganalyst is to find k without having access to I . Call the given image as a start image S_k . We perform additional embedding on this start image. We refer to this kind of embedding as *forced embedding*. Let S_{ki} denote a start image S_k with ($k\%$) embedding and a forced embedding $i\%$. Let,

'a' denote the array of 2nd level LL sub-band coefficients of S_k

'e' denote the array of 2nd level LL sub-band coefficients of S_{ki} and

'c' denote the number of entries that are different in the arrays a and e.

$\eta = \frac{c * 500}{\text{image size in pixels}}$. We have chosen the factor 500 to normalize the quantity η to be near 100 for the size of image being considered (800 * 600).

Let η_{ki} be the average value of η over different images $\in I$ with $k\%$ initial embedding and $i\%$ forced embedding.

We use the stego algorithm Hide4PGP in our experiments. In our experiments we use $i = 10, \dots, 100$. $k = 0, 10, 20, 30, 40, 50$. We plot η_{ki} vs. i for various k as shown in Fig. 1.

For a particular forced embedding say i , it can be observed that η_{ki} decreases as k increases.

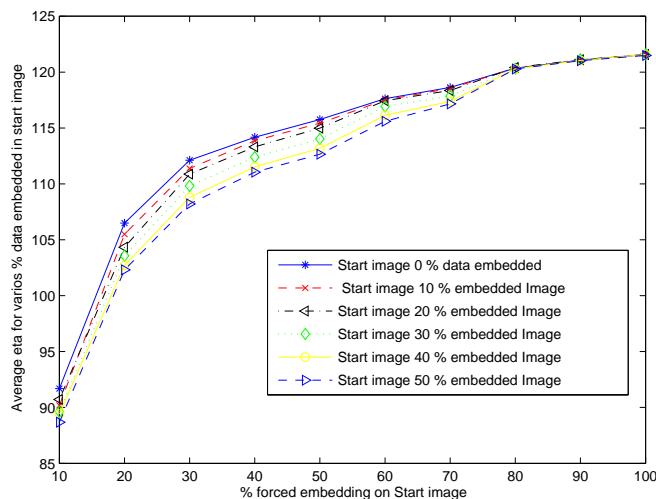
Encouraged by this monotonic trend, we now look closely at the variations in measure η at a fixed forced embedding of $i = 20\%$, with respect to k on different start images. The results are shown in Fig. 2.

The continuous line shows the average value, η_{k20} vs k . The other curves show the η values for the individual images. These also show the monotonic decreasing trend around the average value. We note that such trends are quite significant especially at low levels of 20% embedding. Thus, this serves as a first indicator for detecting approximately the amount of embedding (even at low levels) in any given image.

In our lab we have built a tool called CSA-Tool for simulating the behavior of S-Tool [1]. It is quite difficult to conduct a large number data generation experiments under various parameter choices using a public domain tool as we don't get appropriate handles into the source code. We have taken care to incorporate our own functions for encryption, randomized location generation and embedding analogous to the steps performed by S-Tools. Hence, the statistical characteristics of our tools would closely represent those of S-Tools.

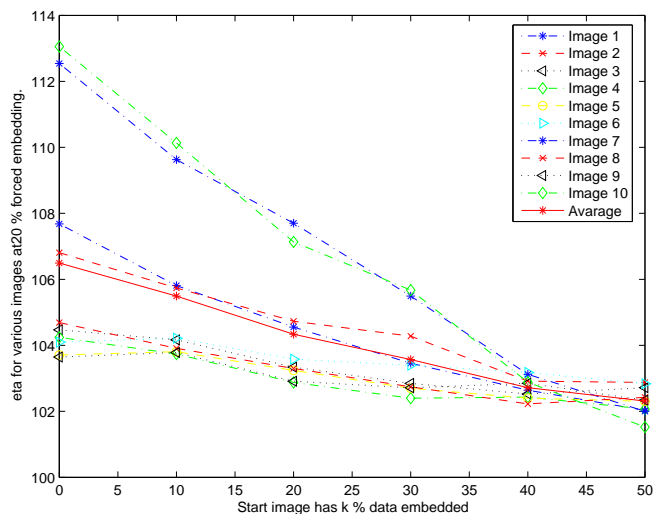
We performed similar experiments as detailed above using the CSA tool. Fig. 3 and Fig. 4 show the results. We note that the results are along the same trends as the Hide4PGP. However, the separations in Fig. 3 are smaller than in Fig. 1 and fluctuations in Fig. 4 are more than in Fig. 2. A reason for this is that CSA Tool (and S-Tools) employ more sound random generators for choosing the LSB for embedding than the tool Hide4PGP.

Fig. 1.



Graph of η_{ki} vs 'i' for various 'k' Hide4PGP

Fig. 2.



Graph of η vs 'k' for various at fixed forced embedding 20% for various images Hide4PGP

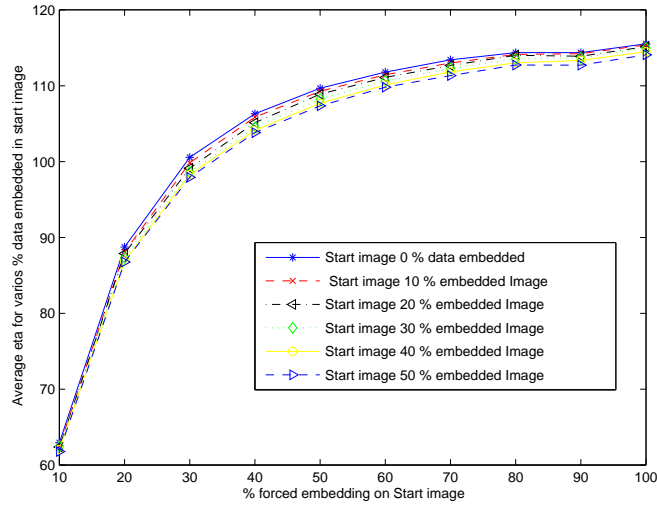
IV. CONCLUSION

We discussed two of our approaches towards analysis of stego images for detection of levels of embedding. Our approach of using wavelet coefficient perturbations holds promise. We also would consider a modified wavelet coefficient based measure that takes into account the numerical changes in the pixel values introduced by embedding. We plan to use this measure in addition to the statistical measures to arrive at finer detection.

REFERENCES

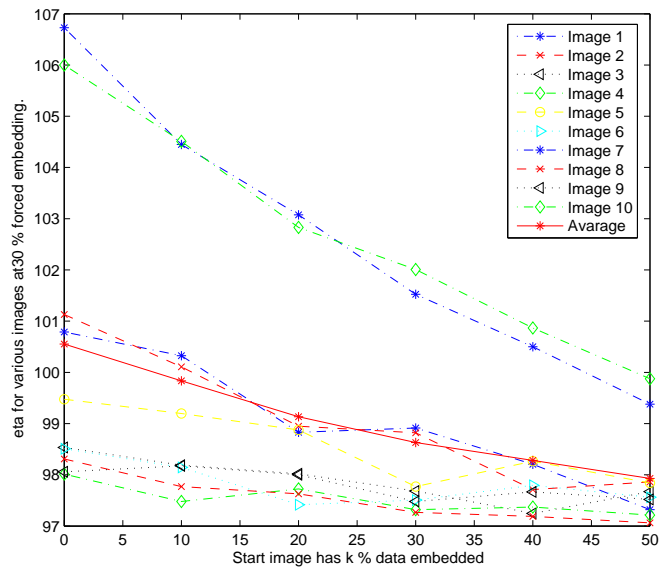
- [1] Andy Brown, S - Tools V 4.0, <ftp://ftp.demon.net/pub/mirrors/crypto/idea/s-tools4.zip>
- [2] Farid H, Detecting Steganographic Message in Digital Images, Report TR2001-412, Dartmouth College, Hanover, NH, 2001
- [3] Fridrich J, Du R, Meng L, Steganalysis of LSB encoding in color image, Proceedings of IEEE International conference on Multimedia an Expo, July 30- August 2, 2000, New York City, NY.
- [4] Fridrich J, Du R, Meng L, Reliable Detection of LSB Steganography in Color and Grayscale Image, Magazine of IEEE multimedia, Special Issue on security, October November issue 2001, pp 22- 28
- [5] J. Fridrich, R. Du, and L. Meng, Steganalysis of LSB Encoding in Color Images, ICME 2000, New York City, July 31-August 2, New York

Fig. 3.



Graph of η_{ki} vs 'i' for various 'k' CSA Tool

Fig. 4.



Graph of η vs 'k' for various at fixed forced embedding 30% for various images CSA Tool

[6] Veni Madhavan C E, Statistical Techniques for Cryptanalysis and Steganalysis, Workshop on Steganography, C-Dac Kolkatta, Oct 2004.