The ISC Int'l Journal of Information Security
ISeCure

January 2009, Volume 1, Number 1 (pp. 17–26)
http://www.isecure-journal.org

Steganalysis of Embedding in Difference of Image Pixel Pairs by Neural Network

Vajiheh Sabeti [a,*], Shadrokh Samavi [a], Mojtaba Mahdavi [a], Shahram Shirani [b]

[a] *Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran*
[b] *McMaster University, Hamilton, Canada*

ARTICLE INFO

Article history:
Received: 14 August 2008
Revised: 6 December 2008
Accepted: 16 December 2008
Published Online: 28 January 2009

Keywords:
Steganography, Steganalysis, Pixel-Value Differencing, Neural Net, Perceptrons

ABSTRACT

In this paper a steganalysis method is proposed for pixel value differencing method. This steganographic method, which has been immune against conventional attacks, performs the embedding in the difference of the values of pixel pairs. Therefore, the histogram of the differences of an embedded image is different as compared with a cover image. A number of characteristics are identified in the difference histogram that show meaningful alterations when an image is embedded. Five distinct multilayer perceptrons neural networks are trained to detect different levels of embedding. Every image is fed in to all networks and a voting system categorizes the image as stego or cover. The implementation results indicate an 88.6% success in correct categorization of the test images.

1 Introduction

Steganography is an art of sending a secret message under the camouflage of a carrier content. The carrier content appears to have normal ("innocent") meanings. The goal of steganography is to mask the very presence of communication, making the true message not discernible to the observer [1]. Steganography is different from classical encryption, which seeks to conceal the content of secret messages. Steganography is about hiding the very existence of the secret messages [2].

Three different aspects in information hiding systems contend with each other: capacity, security, and robustness. Steganographic method strives for high security as well as capacity, and robustness usually is not of main concern [3]. Digital watermarking is a technique which allows an individual to add hidden copyright notices or other verification messages to digital audio, video, or image signals and documents. Such hidden message is a group of bits describing information pertaining to the signal or to the author of the signal (name, place, etc.). Hence, the main difference between steganography and watermarking is the information that has to be secured. In a steganography technique the embedded information is of much importance where as in watermarking the cover image is important. Also, steganography is different from classical encryption, which seeks to conceal the content of secret messages. Steganography is about hiding the very existence of the secret messages [2]. While different communication mediums, such as text, audio and video, can be used for steganography, embedding data in images have been the subject of many studies. Basic elements of steganography in images are shown in Figure 1 [4]. The carrier image in steganography is called the "cover image" and the image which has the embedded data is called the "stego image". The

* Corresponding author.

Email addresses: vajiheh.sabeti@ec.iut.ac.ir (V. Sabeti), samavi96@cc.iut.ac.ir (S. Samavi), mahdavi@ec.iut.ac.ir (M. Mahdavi), shirani@mcmaster.ca (S. Shirani).

ISSN: 2008-2045 © 2009 ISC. All rights reserved.

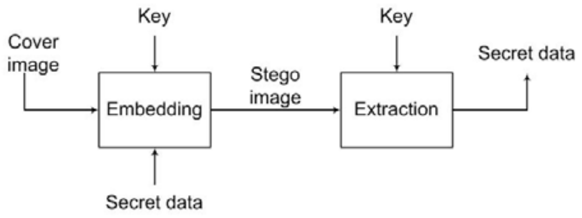


Figure 1. Typical elements of any steganographic method.

embedding process is usually controlled using a secret key shared between the communicating parties. This key ensures that only recipients who know the corresponding key will be able to extract the message from a stego image.

With an increase in popularity of steganographic methods, a new field of steganalysis is established which is amid at distinguishing the presence of steganography. A successful attack is one which detects the presence of hidden data in a stego image. Most steganalytic methods use changes in the statistical characteristics of an image to detect the presence of embedded data. There are two approaches to the problem of steganalysis, one is to come up with a steganalysis method specific to a particular steganographic algorithm (special attacks). The other is developing techniques which are independent of the steganographic algorithm to be analyzed (blind attacks) [5].

There are two kinds of image steganography techniques: spatial-domain and transform domain based methods. Spatial-domain based methods embed messages directly in the intensity of pixels of images [6, 7]. For transform domain based techniques, images are first transformed to another domain (such as the frequency domain), and messages are then embedded in transform coefficients. DCT transformation is the most important transform domain that is used in these methods. Steganography techniques such as Jsteg, Outguess and F5 use this domain for data embedding [8, 9].

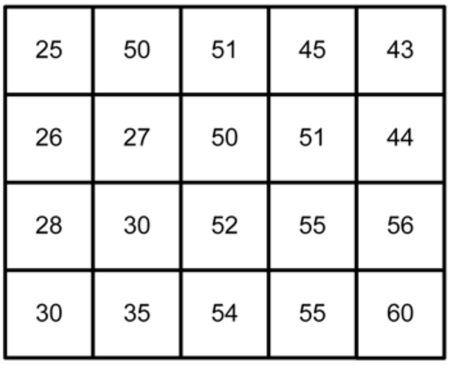
A popular digital steganography technique is the so-called least significant bit (LSB) replacement. With the LSB replacement technique, the two parties in communication share a private secret key that creates a random sequence of samples of a digital signal. The encrypted secret message is embedded in the LSB's of those samples of the sequence. This digital steganography technique takes the advantage of random noise present in the acquired images [1]. Many reliable steganalytic methods have been devised for LSB flipping technique. The production of Pair of Value (PoV) in the histogram of a stego image is the main weak point of the LSB flipping. The presence of PoV has allowed many steganalysis methods, such as x^2 and RS, to successfully attack LSB flipping [10, 11].

Wu and Tsai [12] presented a steganographic method based on Pixel_Value Differencing (PVD). They divide the cover image into a number of non-overlapping two-pixel blocks. Each block is categorized according to the difference of the gray values of the two pixels in the block. A small difference value indicates that the block is in a smooth area and a large one indicates that it is in an edged area. The pixels in edged areas may tolerate larger changes of pixel values than those in the smooth areas. Therefore, it is possible to embed more data in edged areas than in the smooth areas. All possible difference values are classified into a number of ranges. The number of bits that are to be embedded in a pixel pair depends on the width of the range that the difference value belongs to. PVD is immune against the attacks that scrutinize changes in spatial domain [11] or changes in the histogram [10].

In [13] another method based on PVD is proposed which tries to increase the embedding capacity of PVD. In this version the image blocks are categorized based on the calculated pixel differences into two groups of smooth and edge. A block with difference less than a threshold is a smooth block; otherwise, it is an edge block. LSB embedding is used for smooth regions and PVD embedding for edged areas. We refer to this method as the enhanced PVD approach as opposed to the basic PVD approach discussed in [10]. Sabeti et. al [14] have successfully attacked basic PVD and the enhanced version of PVD.

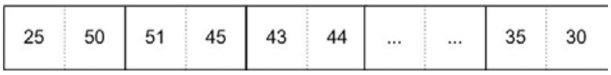
In a paper by Zhang [15], PVD was successfully attacked. This was done by analysis of the histogram of the stego-image. Zhang has also proposed a modified version of PVD, where the shortcomings of basic PVD are alleviated. In this method, instead of fixed ranges used in the original PVD method, variable ranges are used. We refer to this process as the modified PVD (MPVD). So far, no attack has been devised against MPVD.

Generally, blind steganalytic methods use classifiers. These classifiers are trained using a set of cover and stego images obtained from a number of steganographic methods. Classification is based on alterations in a number of characteristics of the natural images due to the embedding process. A number of researchers have used classifiers for steganalysis [16, 17, 18, 19, 20, 21]. These techniques have used numerous feature sets and classification methods such as neural networks, support vector machines and Fisher Linear Discriminator. The goal of these methods is to discover a group of embedding algorithms. There are also methods that use classifiers,



25	50	51	45	43
26	27	50	51	44
28	30	52	55	56
30	35	54	55	60

Figure 2. Example of an image.



25	50	51	45	43	44	...	...	35	30

Figure 3. Vectorized image

and are devised to attack a specific steganographic method [22].

In this paper, we propose a steganalysis method which attacks and successfully identifies the existence of MPVD embedding. In this attack, we have used neural networks for classification purposes. The organization of the paper is as follows. In section 2, PVD and MPVD algorithms are reviewed. Statistical changes that occur in an image due to the MPVD embedding are analyzed in section 3. The proposed steganalysis is detailed in section 4. The implementation results of the proposed method are presented in section 5. Conclusions appear in section 6.

2 Pixel-Value Differencing Steganography

The cover images used in the PVD method are supposed to be 256 gray-valued ones. A difference value d is computed from every non-overlapping block of two consecutive pixels, say p_i and p_{i+1} of a given cover image. Partitioning the cover image into two-pixel blocks runs through all the rows of each image in a zigzag manner. In Figure 2, an example of an image is shown. The two-pixel blocks that are constructed by zigzag scanning of the example image is shown in Figure 3.

Assume that the gray values of p_i and p_{i+1} are g_i and g_{i+1}, respectively; then d is computed as $g_{i+1}-g_i$, which may be a number ranging from -255 to 255. A block with d close to 0 is considered to be an extremely smooth block, whereas a block with d close to -255 or 255 is considered as a sharply edged block. The method only considers the absolute values of d (0 through 255) and classifies them into a number of contiguous ranges, such as R_k where $k = 1, 2, ..., q$. The lower and upper bound values of R_k are denoted by l_k and u_k, respectively, where l_1 is 0 and u_q is 255. The width of $R_k(w_k)$ is $u_k - l_k + 1$. In PVD method, the width of each interval is taken to be a power of 2. A practical set of intervals may be:
$[0, 7], [8, 15], [16, 31], [32, 63], [64, 127], [128, 255]$

Every bit in the bit stream should be embedded into the two-pixel blocks of the cover image. Given a two-pixel block B with gray value difference d belonging to kth range, then n bits can be embedded in this block, and can be calculated by $n = \log(u_k + l_k + 1)$ which is an integer. A sub-stream S with n bits is selected next from the secret message for embedding in B. A new difference d' then is computed by:

$$d' = \begin{cases} l_k + b, & \text{if } d \geq 0; \\ -(l_k + b), & \text{if } d < 0. \end{cases} \tag{1}$$

where b is the value of the sub-stream S. Because the value b is in the range from 0 to $u_k - l_k$, the value of d' is in the range from l_k to u_k. If we replace d with d', the resulting changes are presumably unnoticeable to the observer. Then b can be embedded into pixels p_i and p_{i+1} in a manner that the new pixel values produce a difference of d'. The new gray values (g'_i, g'_{i+1}) are obtained for the pixels in the corresponding two-pixel block (p'_i, p'_{i+1}) of the stego-image. The embedding process is finished when all the bits of the secret message are embedded. The calculation for computing (g'_i, g'_{i+1}) from the original gray values (g_i, g_{i+1}) of the pixel pair is based on a function $f((g_i, g_{i+1}), m)$, which is defined to be

$$(g'_i, g'_{i+1}) = f((g_i, g_{i+1}), m) = \\ \begin{cases} (g_i - \lceil m/2 \rceil, g_{i+1} + \lfloor m/2 \rfloor), & \text{if } d \textit{ is odd}; \\ (g_i - \lfloor m/2 \rfloor, g_{i+1} + \lceil m/2 \rceil), & \text{if } d \textit{ is even}. \end{cases} \tag{2}$$

where m is $d' - d$. Obviously embedding is only considered for pixels whose new pixel values would fall in the range of [0,255].

In order to increase the security of the simple PVD method, another group of researchers proposed a modified PVD method [15]. In this method, instead of fixed ranges used in the original PVD method, variable ranges are used. In other words, the ranges corresponding to different blocks are differently defined according to a secret key $\beta \in [0, 1]$. This parameter is selected pseudo-randomly for each block, then lower and upper bound of ranges are calculated by:

$$\begin{aligned} l'_k &= l_k + \lfloor \beta . w_k \rfloor \\ u'_k &= u_k + \lfloor \beta . w_{k+1} \rfloor \end{aligned} \tag{3}$$

where k is a range index. If $l'_k \leq |d| \leq u'_k (k > 1)$, a total of $\log_2(w_k)$ secret bits are embedded into the corresponding block. Convert the secret bits into a decimal value b, and calculate

$$d' = \begin{cases} \underset{l'_k \le e \le u'_k,\quad \bmod (e, w_k) = b}{\operatorname{argmin}} (|e-d|) \quad , & \text{if } d > 0; \\ -[\underset{l'_k \le e \le u'_k,\quad \bmod (e, w_k) = -b}{\operatorname{argmin}} (|e-d|)\quad], & \text{if } d < 0. \end{cases} \tag{4}$$

In other words, d' is the value that is closest to d among all values in the same range having a residue $b \mod w_k$. If $0 \le |d| \le u'_0$, calculate d' from the decimal value b representing $\log_2(w_0)$ secret bits, as follows:

$$d' = \underset{-u'_0 \le e \le u'_0,\quad \bmod (e, w_0) = b}{\operatorname{argmin}} (|e-d|) \tag{5}$$

Then we modify the two pixels using equation 2. As in the simple PVD method, the larger the values of d, the more the secret bits are embedded.

On the extraction side, b can be restored simply as follows:

$$b = \begin{cases} \bmod (d', w_0), & \text{if } 0 \le |d'| \le u'_0; \\ \bmod (d', w_k), & \text{if } l'_k \le |d'| \le u'_k (k > 0). \end{cases} \tag{6}$$

3 MPVD Statistical Alteration of Image

Despite the claims of the designers of the basic PVD algorithm about the resilience of the method against known attacks, two steganalysis methods, so far, have been presented for it [14,15]. In both attacks an analysis of the histogram has been employed. Since the embedding in PVD is not done directly in the pixels of the image, the histogram of the image does not show any pronounced modifications. PVD embeds the data in the difference of pixels hence scrutiny of the pixel difference histogram (**PD** histogram) is the main tool in our analysis. As an example, Figure 4 shows the PD histogram of Lena after 100% embedding using basic PVD. The deformities in the histogram in terms of grouping of certain bins are apparent.

In the basic PVD, the lengths of different intervals, w_k, are fixed. The new pixel difference, d', is computed in such a manner that the data can be extracted from $d' \mod w_k$. Since the length of an interval has to be a power of 2, the number of bits of d' is dependent on w_k. Embedding is done in the least significant bits of the pixel difference of a block. In other word, basic PVD is very similar to LSB flipping method. This explains the groupings of the PD histogram bins, which are similar to the pairing phenomenon that happens in the LSB flipping method. The only difference is that in the LSB flipping, there is pair-wise flatness among certain histogram bins, and in PVD there is group-wise flatness among adjacent bins. This effect has been used for an effective attack which employs x^2 tools [14].

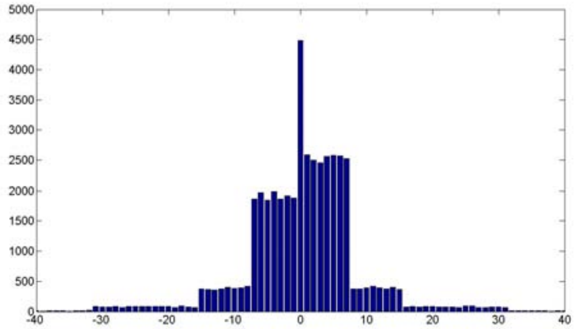
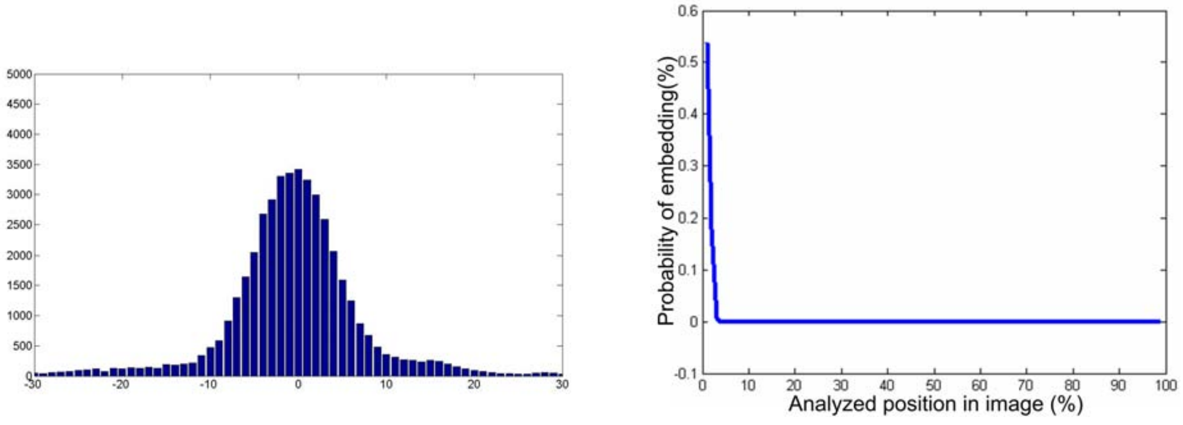


Figure 4. Pixel difference (PD) histogram of Lena with 100% PVD embedding.

In [15] after presenting a successful attack to the basic PVD, a modified version of PVD (MPVD) is introduced. It is claimed that the bin grouping as happens in the basic PVD does not occur in the MPVD. The only difference is that in MPVD, the lengths of the ranges are randomly selected and are not necessarily powers of 2. Embedding in MPVD is not in the LSBs of pixel differences and hence the grouping (pairing) effect does not occur. Therefore, MPVD is secure when attacked by [14]. The steganalysis's result is shown in Figure 5 for Lena image. Figure 5(a) shows the PD histogram of Lena after 100% embedding using basic MPVD and 5(b) shows result of Sabeti *et al.* [14] attack on this image. Probability of embedding is zero, so this attack is unsuccessful in MPVD embedding detection.

In [15] it is shown that although the proposed histogram analysis is successful in attacking the simple PVD, MPVD defeats this steganalysis method. Therefore, MPVD is secure against the attacks of [14] and [15], which are specifically designed for PVD based methods. On the other hand, there are several blind attacks designed to detect a host of different steganography methods. Many of the parameters that are used in the blind methods are extracted from the image histogram. MPVD has the advantage of not producing perceptible effects on the image histogram. Hence, it is expected that the blind attacks not defeat MPVD embedding. Even if a blind method could detect MPVD, its overall accuracy is much lower than that of a specially designed MPVD steganalysis [5]. This is why we attempt to propose a special steganalysis method which successfully identifies the existence of MPVD embedding.



(a)

(b)

Figure 5. (a) PD histogram after MPVD embedding in Lena, (b) result of [14] attack on Lena image.

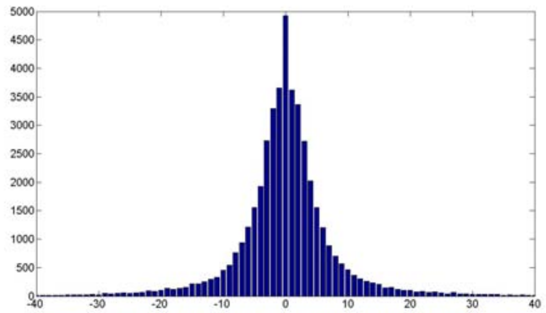


Figure 6. PD histogram of Lena: a Gaussian distribution.

It has been observed that the PD histogram for regular images with no embedding has a Gaussian distribution [23]. Figure 6 shows the Gaussian shape of the PD histogram of the Lena image. It is expected that the PD histograms for stego images would be different than usual images and would have distributions different than Gaussian. To distinguish the dissimilarity between the histograms, we need a criterion. For images using a number of tests, the mentioned dissimilarities can be parameterized and a threshold can be obtained to separate cover and stego images. In most cases, finding the appropriate threshold is not possible. Using learning methods based on neural nets is a possible solution. A neural net is trained using the characteristics of a set of cover and stego images. After the training phase, the neural net is fed with the characteristics extracted from images of a test set, and can distinguish between stego and cover images.

4 Proposed Steganalysis

The first step is to find a set of image characteristics that alter due to embedding. Based on the behavior of MPVD, it is expected that the PD histogram can provide the required characteristics. We need to search for characteristics that drastically alter when cover is embedded to form a stego image. These characteristics should not vary when a stego image is reembedded. Therefore, the overview of the attack is as shown in Figure 7.

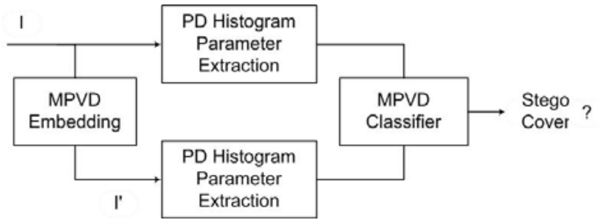
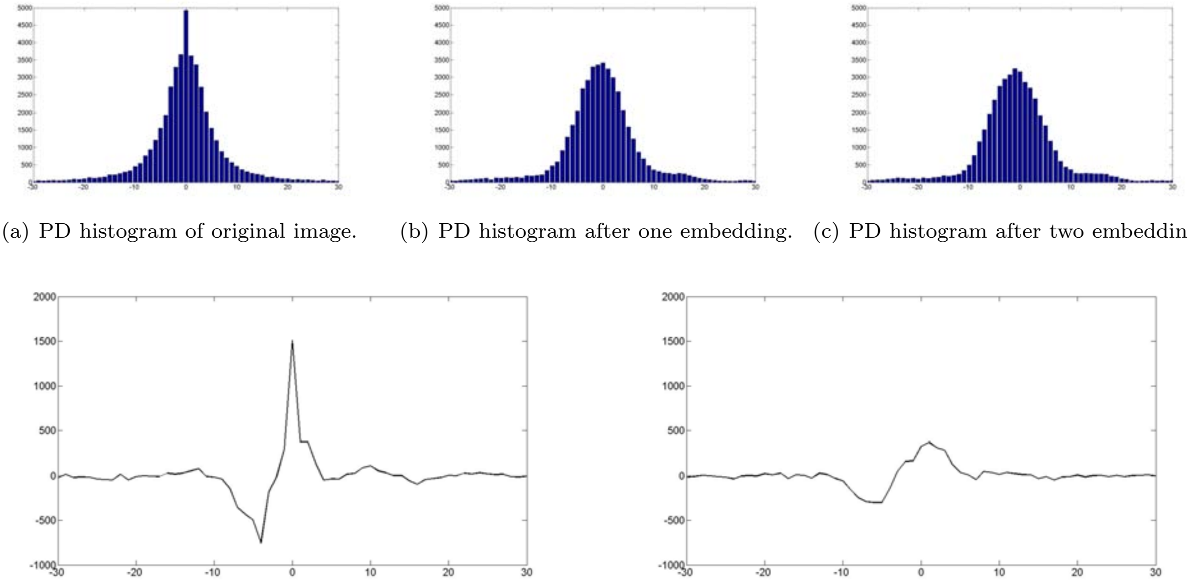

I
PD Histogram Parameter Extraction
MPVD Embedding
MPVD Classifier
Stego Cover ?
PD Histogram Parameter Extraction
I'

Figure 7. Block diagram of proposed steganalysis.

Diagram of Figure 7 can be explained when considering that a suspected image I is under analysis. An appropriate feature is extracted from the PD histogram. This is done by the block that is called *PD Histogram parameter Extraction*. The features under consideration will be explained shortly. The suspected image also goes through an embedding process, where with MPVD algorithm, 100% of its capacity is used for embedding to generate image I'. Also extracted are characteristics from the PD histogram of I'. Now we need to define the type of characteristics and the structure of the classifier. A study of the PD histogram of a number of images after one and two embeddings is needed to reveal the required characteristics. Figures 8 and 9 show two examples from our studies. What are shown are the original PD histograms and PD histograms after one and two embeddings in the Lena and Boat images. Also, the differences between the PD histograms after each embedding are shown.

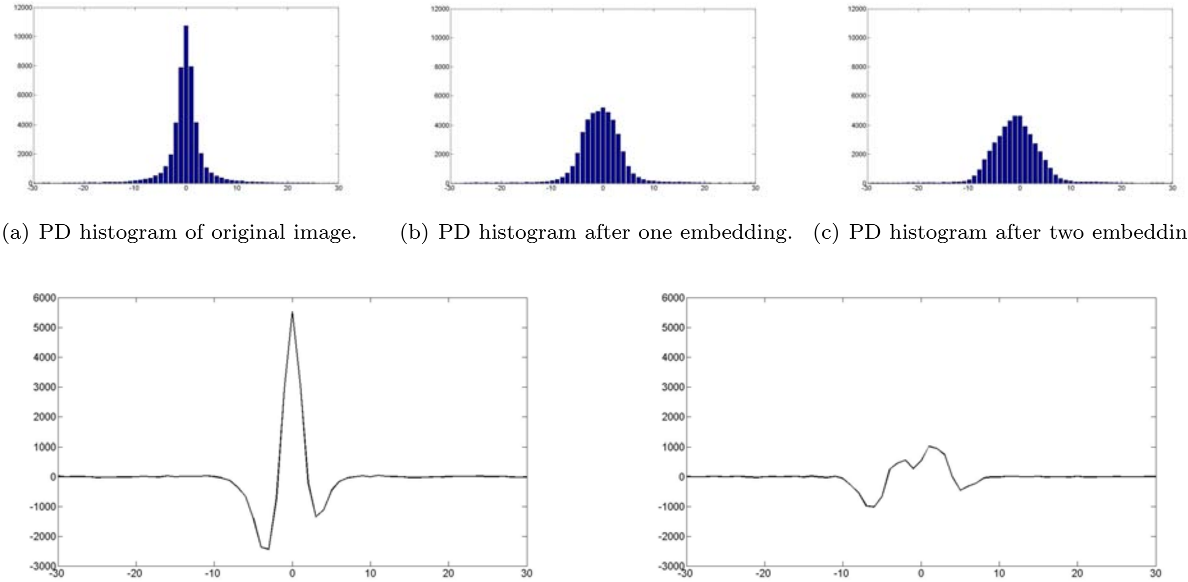
As it was expected, PD histograms have a Gaussian distribution before embedding is performed. Even



(a) PD histogram of original image. (b) PD histogram after one embedding. (c) PD histogram after two embeddings.

(d) Difference between PD histograms of (a) and (b). (e) Difference between PD histograms of (b) and (c).

Figure 8. Results from testing with Lena image.



(a) PD histogram of original image. (b) PD histogram after one embedding. (c) PD histogram after two embeddings.

(d) Difference between PD histograms of (a) and (b). (e) Difference between PD histograms of (b) and (c).

Figure 9. Results from testing with Boat image.

though after MPVD embedding, there are no grouping and deformities occurring in the PD histogram, there are apparent dissimilarities in the distribution of these histograms. On the other hand, if we perform the embedding two times on a single image, the PD histogram after the first time of embedding and the PD histogram after the second time of full capacity embedding have the same type of distributions. Hence, we need a parameter that shows the difference between the PD histogram of a cover image as well as a stego image, and also show the similarity between the PD histogram of a stego image and a stego image which is re-embedded within MPVD.

Based on the above discussions, we can suggest a parameter based on the values of the three central bins of the PD histogram. Obviously, on such basis, a number of different formulations can be suggested, of which we only consider one.

Suppose that d_0, d_1 and d_{-1} are respectively the values of '0 '1', and '-1' bins of the PD histogram. Parameter Q can now be formulated as:

$$Q = \frac{d_0 - \frac{d_1 + d_{-1}}{2}}{d_0 + d_1 + d_{-1}}$$

In computing the value of Q, the difference between the value of '0' bin with the average of the other two bins is of interest. This value, which can vary for different images, is normalized with the sum of the three mentioned bins. For images with no embedding it is expected that Q is a large number. For stego images, Q should have a small value. The value of Q is computed for a given image I, and is labeled as QI. Then the image I is embedded using MPVD to produce image I'. The subsequent value of Q extracted from image I' is labeled as QI'.

In Table 1, we have a list of the parameters used for training the neural network. The i^{th} bin of the PD histogram of images I and I' respectively have the values d_i and d'_i. As mentioned before, we need to look at bins -2 to 2. The differences between two neighboring bins in the mentioned range are also of interest. The rest of the bins of the PD histogram, outside of the region of interest, are not considered. Beside the values of the bins of the PD histogram, we are also considering the values of QI, QI' and their ratios.

The classifier that was employed is shown in Figure 10. In our proposed method, artificial neural network (NN) [24] is used for the classification purposes. Multi Layer Perceptron (MLP) is a layered feed forward network typically trained with static backpropagation. It is expected that the powerful learning capability of the NN will outperform the linear classifiers [20].

The purposed classifier structure consists of five MLP networks. Each MLP network is dedicated to distinguish a certain amount of embedding. For example, the MLP network that is labeled as NN20 is trained to detect images with 20% embedding. By x% embedding we mean that x percent of the image blocks is selected randomly and then the embedding process is done for these blocks. For example 20% embedding in MPVD means that 20% of the 2-pixel blocks are selected for embedding. But how many bits are embedded in each block depends on the differences of the grayscale intensities of the pixel pair. If the difference is high then MPVD embeds more bits. This network is trained to produce positive values for cover images, and negative values for images with 20% embedding. Similarly, other networks labeled as NN40, NN60, NN80 and N100 respectively detect 40% to 100% embedding. All of the networks shown in Figure 10 are MLPs which differ from one another by their hidden layers and the number of epochs required for their training. Table 2 shows the common features of the 5 networks of Figure 10. Each network has 21 processing elements (PEs) at its input, 8 PEs in its hidden layers and one PE at its output. The specific characteristics of each of these networks are shown in Table 3.

Table 1. Image feature set.

Features	Quantity
$d_i(-2 \le i \le 2)$	5
$d_{i+1} - d_i(-2 \le i \le 1)$	4
$d'_i(-2 \le i \le 2)$	5
$d'_{i+1} - d'_i(-2 \le i \le 1)$	4
QI	1
QI'	1
QI/QI'	1

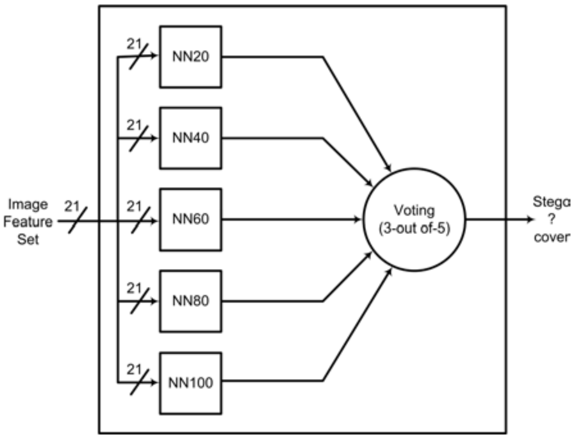


Figure 10. Structure of the suggested classifier.

To differentiate between a cover and a stego image,

Table 2. Common specifications of employed neural nets.

Attribute	**Value**
Neural model	Multilayer Perceptrons (MLP)
Learning algorithm	Back Propagation
Input PEs	21
Hidden PEs	8
Hidden transfer function	Hyperbolic tangent
Output PEs	1
Output transfer function	Linear Hyperbolic tangent

Table 3. Specifications of individual neural nets.

Network	**Hidden Layers**	**Epochs**
NN20	3	60000
NN40	2	50000
NN60	2	30000
NN80	1	50000
NN100	1	30000

the extracted image feature set is applied to the inputs of the five networks of the classifier. A voting process is conducted among the outputs of the networks. At least three positive outputs are required to declare an image as stego. Otherwise, the image is considered as cover. In this case zero is selected as threshold, but this threshold could be modified. Each network is trained separately with a specific amount of embedding. The trained networks react differently when they are exposed to test images that contain amounts of embedding that are not trained for. Hence, the voting process alleviated this problem.

5 Implementation Results

In this section, we present the results obtained from implementation of the suggested steganalysis. The first step in implementation of the method was to extract the feature set using Matlab 7.1. The second step was the implementation of the neural networks using NeuroSolution 5.05. To train the classifier, we need extracted features both from stego and cover images. The train and test sets consisted of 200 images. Images of this data set are selected randomly from The Internet. Hence the images possessed a wide variety of visual characteristics. These images have different sizes and types (JPG, BMP, *etc.*). Each image was embedded with 6 different levels of embedding. This means that 1200 cases were used for training and testing the classifier.

Table 4. Results from testing each neural net.

Network	**True Positive (TP)**	**False Positive (FP)**
NN20	43	12
NN40	47	10
NN60	46	3
NN80	49	1
NN100	48	0

Table 5. Results of testing the proposed classifier.

Embedding Percent of Test Images	**Cover**	**Stego**	**Accuracy**
0	48	2	96%
20	40	10	80%
40	15	35	70%
60	4	46	92%
80	2	48	96%
100	1	49	98%

Each network was individually trained with feature sets extracted from 150 cover and 150 stego images embedded with the required percentage of capacity. The trained classifier was then fed within the 50 test images. Each test image was once used as the cover and five times, at different levels of embedding, was used as a stego image. Table 4 shows the results obtained from each trained network when tested with cover images and stego images embedded with amounts of data that the network was trained for. These results were obtained for positive voter outputs indicating stego-images; negative outputs showed cover images ($threshold = 0$). The true positive (TP) results are those stego images that were correctly identified and hence, false positive (FP) are those cover images that are identified as stego. We need higher values for TP and lower values for FP.

The proposed classifier was eventually tested with the entire mentioned test set, which contained 50 cover, and 50 differently embedded percentages. Table 5 shows the results from these tests. In these results images with 20 percent embedding or less are considered as cover images and the threshold is taken to be zero.

Let us assume that the number of cover images that are tested is N, and the number of stego images is P. If TN is the number of cover images that are correctly identified (true negative), then we can define the accuracy of the classifier as the follows:

$$Accuracy = \frac{(TP+TN)}{(P+N)}$$

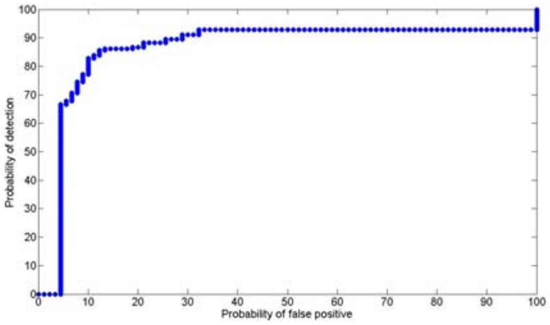


Figure 11. ROC curve for true-positive vs. false-positive cases.

The results shown in Table 5 show the high accuracy of the proposed method when the amount of embedding is above 50%. Cover images were identified as containing no embedding with high accuracy. Overall, according to Table 5, we tested 200 stego and 100 cover images. Stego-images were produced by 40, 60, 80 and 100 percent embeddings. On the other hand, cover images had 0 and 20 percent embeddings. The results that are shown in Table 5 indicate that 88 images were correctly identified as cover, and 178 images as stego. Hence, with $P = 200$, $N = 100$, $TP = 178$, and $TN = 88$ we can calculate the overall accuracy of the classifier as 88.6%.

In Figure 11, the obtained receiver operating characteristic (ROC) curve is illustrated, which shows how the true-positive and false-positive cases vary as the output threshold changes. In [25], it is argued that a reasonable one-dimensional measure of the performance is the false positive rate when the true-positive rate is 50%. Figure 11 shows that when the true-positive rate is 50%, the false-positive rate is 4%. This rate indicates the high accuracy of our method.

6 Conclusion

The steganographic method that was attacked in this paper, MPVD, embeds the secret data within the differences of pairs of pixels. This is unlike the methods which embed their messages into the LSB of pixels. Therefore, conventional attacks that are based on analysis of the histogram of the image are not capable of detecting the MPVD embedding.

In this paper we proposed that rather than analyzing the histogram of an image, we should consider the histogram of the differences of pixel pairs (PD histogram). It was shown that the PD histogram of a cover image is different than that of a stego image. The difference between these two histograms is characterized with a feature set of 21 elements. Using a classifier with 5 neural networks, we were able to distinguish stego images from the covers with an overall accuracy of 88.6%. This accuracy is true when images with embedding more than 20% are tested. With changing threshold, when the false negative rate is 50%, false positive rate is 4%. This shows that designed classifier is very successful in MPVD detection.

References

[1] S. Dumitrescu, X. Wu, and Z. Wang. Detection of LSB Steganography via Sample Pair Analysis. *IEEE Transactions on Signal Processing*, 51(7):1995–2007, 2003.

[2] N. Hopper. *Toward a Theory of Steganography*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2004.

[3] N. Provos and P. Honeyman. Hide and Seek: An Introduction to Steganography. *IEEE Security & Privacy*, pages 32–44, 2003.

[4] B.Pitzmann. Information Hiding Terminology. In *Information Hiding*, volume 1174 of *Lecture Notes in Computer Science*, pages 347–350, Cambridge, U.K., 1996. Springer.

[5] M. Kharrazi, H.T. Sencar, and N. Memon. Image Steganography: Concepts and Practice. *WSPC/ Lecture Notes Series*, 2004.

[6] A. Westfeld. F5 a Steganographic Algorithm: High Capacity Despite Better Steganalysis. In *Proceedings of the 4th International Information Hiding Workshop*, volume 2137, pages 289–302. Springer-Verlag, 2001.

[7] M. Swanson, M. Kobayashi, and A. Tewfik. Multimedia Data Embedding and Watermarking Technologies. *Proceedings of the IEEE*, 86(6):1064–1087, 1998.

[8] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure Spread Spectrum Watermarking for Multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[9] N. Provos. Defending against statistical steganalysis. In *the 10th USENIX Security Symposium*, pages 323–335, 2001.

[10] A. Westfeld and A. Pfitzman. Attacks on Steganographic Systems. In *Proceedings of the 3rd International Information Hiding Workshop*, pages 61–76. Springer-Verlag, 1999.

[11] J. Fridrich, M. Goljan, and R. Du. Reliable Detection of LSB Steganography in Grayscale and Color Images. In *Proceedings of the ACM Special Session on Multimedia Security and Watermarking*, pages 27–30, 2001.

[12] D.C. Wu and W.H. Tsai. A Steganographic Method for Images by Pixel-Value Differencing. *Pattern Recognition Letters*, 24:1613–1626, 2003.

[13] H.C. Wu, N.I. Wu, C.S. Tsai, and M.S. Hwang. Image Steganographic Scheme Based on Pixel-Value Differencing and LSB Replacement Methods. *IEEE Proceedings Vision, Image and Signal Processing*, 152(5):611–615, 2005.

[14] V. Sabeti, S. Samavi, M. Mahdavi, and S. Shirani. Steganalysis of Pixel-Value Differencing Steganographic Method. In *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 292–295, 2007.

[15] X. Zhang and S. Wang. Vulnerability of Pixel-Value Differencing Steganography to Histogram Analysis and

Modification for Enhanced Security. *Pattern Recognition Letters*, (3):331–339, 2004.

[16] H. Farid. Detecting Steganographic Message in Digital Images. Technical report, Dartmouth College, 2001.

[17] J. Fridrich. Feature-based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes. In *Proceedings of the 6th International Workshop on Information Hiding*, volume 3200 of *LNCS*, pages 67–81, 2004.

[18] T. Holotyak, J. Fridrich, and S. Voloshynovskiy. Blind Statistical Steganalysis of Additive Steganography Using Wavelet Higher Order Statistics. In *Proceedings of the 9th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, volume 3677 of *LNCS*, pages 273–274, 2005.

[19] J. Fridrich and T. Pevny. MultiClass Blind Steganalysis for JPEG Images. In *SPIE Electronic Imaging, Security, Steganography and Watermarking of Multimedia Contents*, volume 8, pages 257–269, 2006.

[20] Y. Q. Shi, G. Xuan, D. Zou, and J. Gao. Steganalysis Based on Moments of Characteristic functions Using Wavelet Decomposition, Prediction-Error Image, and Neural Network. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 269–272, 2005.

[21] J. Fridrich and T. Pevny. Merging Markov and DCT Features for Multi-Class JPEG Steganalysis. In *the SPIE Electronic Imaging*, pages 3–4, Photonics West, 2007.

[22] Q. Liu, A.H. Sung, Z. Chen, and J. Xu. Feature Mining and Pattern Classification for Steganalysis of LSB Matching Steganography in Grayscale Images. *Pattern Recognition*, 41:56–66, 2008.

[23] T. Zhang and X. Ping. A New Approach to Reliable Detection of LSB Steganography in Natural Images. *Signal Processing*, 83:2085–2093, 2003.

[24] M.T. Hagan, H.B. Demuth, and M. Beale. *Neural Network Design.* PWS Publishing, Boston, 1996.

[25] A.D. Ker. Quantitative Evaluation of Pairs and RS Steganalysis. *SPIE Security, Steganography, Watermarking Multimedia Contents*, 5306:83–97, 2004.

Vajiheh Sabeti received her BS degree in Computer Engineering from Isfahan University of Technology (IUT) in 2005 ranking first in her graduating class. She continued her education at the Master's level in the field of Computer Architecture at IUT. She ranked first when she received her MS degree in 2007. She is now a Ph.D. candidate at the department of Electrical and Computer Engineering of IUT. Mrs. Sabeti's research interests are in the fields of data hiding and soft computing.

Shadrokh Samavi is a professor of computer engineering at the Electrical and Computer Engineering Department, Isfahan University of Technology, Iran. He completed a B.S. degree in Industrial Technology (1980) and received a B.S. degree in Electrical Engineering (1982) at California State University, a M.S. degree (1985) in Computer Engineering at the University of Memphis and a Ph.D. degree (1989) in Electrical Engineering at Mississippi State University, U.S.A. Professor Samavi's research interests are in the areas of image processing and hardware implementation and optimization of image processing algorithms. He is also interested in steganography and related subjects. Dr. Samavi is a Registered Professional Engineer (PE), USA. He is also a member of IEEE and a member of Eta Kappa Nu and Tau Beta Pi honor societies.

Mojtaba Mahdavi received his BS degree in Computer Engineering in 1999, and his MSc degree in Computer Architecture in 2002, both from the Electrical and Computer Engineering department of Isfahan University of Technology (IUT), Isfahan, Iran. From Autumn of 2004 he has been a PhD candidate in the field of Computer Engineering at the ECE department of IUT. Mr. Mahdavi's main research area is in the field of data hiding and steganography.

Shahram Shirani is an associate professor at the department of Electrical and Computer Engineering of McMaster University, Canada. He received his BS degree in Electrical Engineering from Isfahan University of Technology in 1989. In 1994 he finished his Master's program in Medical Engineering at Amirkabir University of Technology. Dr. Shirani received his Ph.D. in Electrical Engineering from University of British Colombia, Canada, in 2000. His research interests are biomedical image processing, image compression, and data hiding in multimedia. Dr. Shirani is registered professional engineer in Ontario, Canada, and is a member of IEEE.