Digital Investigation 11 (2014) 67–77

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Blind Image Steganalysis of JPEG images using feature extraction through the process of dilation

Pritesh Pathak, S. Selvakumar*

Dept. of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli 620015, Tamil Nadu State, India

ARTICLE INFO

Article history:
Received 17 November 2013
Received in revised form 27 November 2013
Accepted 28 December 2013

Keywords:
Blind Image Steganalysis
Dilation
Steganography
Feature extraction
Frequency
Spatial
Wavelet

ABSTRACT

The detection of stego images, used as a carrier for secret messages for nefarious activities, forms the basis for Blind Image Steganalysis. The main issue in Blind Steganalysis is the non-availability of knowledge about the Steganographic technique applied to the image. Feature extraction approaches best suited for Blind Steganalysis, either dealt with only a few features or single domain of an image. Moreover, these approaches lead to low detection percentage. The main objective of this paper is to improve the detection percentage. In this paper, the focus is on Blind Steganalysis of JPEG images through the process of dilation that includes splitting of given image into RGB components followed by transformation of each component into three domains, viz., frequency, spatial, and wavelet. Extracted features from each domain are given to the Support Vector Machine (SVM) classifier that classified the image as steg or clean. The proposed process of dilation was tested by experiments with varying embedded text sizes and varying number of extracted features on the trained SVM classifier. Overall Success Rate (OSR) was chosen as the performance metric of the proposed solution and is found to be effective, compared with existing solutions, in detecting higher percentage of steg images.

1. Introduction

Steganography is the art of hiding a message in a carrier. Earlier this technique was used by kings for sending any private message by embedding it in the messenger's body parts. Today, this art of hiding has turned digital, hence the term digital image steganography. Various algorithms have been developed over the years for hiding the message into the digital image (The resultant image is then called as Steg Image.) This art has also become a challenge for the human as it could be used for illegal activities such as terrorism. Terrorists use this art for sending their messages to various parts of the world through internet without being noticed. Hence a dire need arises for a counter technique to detect such steg images which is known as Digital Image Steganalysis.

Digital Image Steganalysis is the technique only for the detection of any message in a digital image. Extraction of message is a part of Cryptanalysis. There are two types of Steganalysis: (a) Targeted or Specific and (b) Blind or Universal. Targeted Steganalysis refers to the technique of identifying the Steg image where the Steganography algorithm used for hiding the message is known, whereas, in case of Blind Steganalysis, the steganography algorithm is unknown. Hence it becomes most difficult to identify. JPEG images have been the most commonly exchanged image format over internet. This paper focuses on the Blind Image Steganalysis and proposes a technique for identification of any JPEG image as steg or clean image.

The rest of the paper is organized as follows: Section 2 discusses the existing solutions. Motivation is discussed in Section 3. Section 4 discusses the proposed technique,

* Corresponding author. Tel.: +91 431 250 3203.
E-mail addresses: priteshpathak15@gmail.com (P. Pathak), ssk@nitt.edu (S. Selvakumar).

1742-2876/$ – see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.diin.2013.12.002

experiments conducted, and their results. Finally, the paper is concluded in Section 5.

2. Existing solutions

2.1. Targeted Steganalysis

In Fridrich et al. (2000), LSB embedding is detected by the presence of many close pairs. Detection of gray scale steg images was proposed in Fridrich et al. (2001). Further, the message length was derived by forming three groups, viz., regular, singular, and unusable. Detection of audio steganography was proposed in Dumitrescu et al. (2003) based on some statistical measures of sample pairs that are highly sensitive to LSB embedding operations. Steganography algorithm, F5 (Westfeld, 2001), was attacked in Fridrich et al. (2003a) and message length was determined using distinguished statistical quantities, such as T, that correlate with the number of modified DCT coefficients. F5 with very low embedding in gray scale images was detected in Cai et al. (2005).

The detection of EzStego (Machado) steganography technique in palette images (GIF image), using pair analysis was done in Fridrich et al. (2003b).

All these algorithms assumed that the steganography algorithm was already known. The image format used in most of these techniques was bmp.

2.2. Blind Steganalysis

Detection of a steganography along with watermarking was done in Avcıbas et al. (2003) by identifying the image quality metrics with the help of Analysis of Variance (ANOVA) (Rencher, 1995) technique and building a feature set which is passed to multivariate regression classifier used to classify the images as steg and clean. Training and testing has been done on bmp images with known LSB steganography techniques such as Steganos (Steganos II Security Suite), Stools (Brown) and Jsteg (Korejwa). The steganalysis technique works only on LSB embedding steganography techniques.

In Shi et al. (2005), steganalysis technique was proposed in which features from gray scale bmp images were extracted using the moments of characteristic functions in subbands of the wavelet transformation of image which was then trained and tested using a neural network classifier. These images used for training and testing were embedded with five known steganography techniques, viz., non-blind SS (Cox et al., 1997), blind SS (Piva et al.), block SS (Huang and Shi, 1998), generic QIM (Chen and Wornell, 1998), and generic LSB. This work was extended in Zhang and Zhong (2009) which measured all the 78-dimensional features with the help of *F*-score feature selection method, selected one threshold value, and dropped those features which have *F*-scores below that value. Choosing of suitable threshold is a difficult task as the results may vary for different steganography algorithms.

A technique for detecting additive steganography or LSB matching (Holotyak et al., 2005) with features extracted from an estimated stego signal, obtained in wavelet domain, using model based approximation of stego image pdf was proposed in Mielikainen (2006). The features from gray scale images were then trained and tested with linear classifier.

The steganalysis methodology in Luo et al. (2011) provides a comparison between two most commonly used statistical features, viz., Characteristic Function (CF) and Probability Density Function (PDF) moments, in Blind Steganalysis and gives a theoretical and practical analysis on feature selection and extraction.

Though a very good effort has been made in this field of steganalysis, still there are some areas unexplored. The above algorithms, in spite of their advantages, have some flaws. The proposed algorithm is an attempt to cover the less explored area of combining RGB with feature extraction in three domains of JPEG image.

3. Motivation

Steganography is being used for communication and carrying out anti-social activities (News article). Therefore, in the interest of the Nation's Security, detection of hidden communication within any media transmission is of utmost significant and this motivated to take up further research in this field. The literature survey on steganography and steganalysis, confirmed the need for a Blind Steganalysis algorithm which can clearly distinguish between steg and clean image. Though many Blind Steganalysis algorithms have been devised, still there are some less explored areas. Several detection techniques which considered R, G, or B separately or feature extraction in different domains, viz., frequency, spatial, and wavelet, separately are existing. As most of the images used nowadays are color in nature, any attempt to hide any information may affect either one or more or all color information. Further, it may affect any feature(s) of any of the three domains, viz., frequency, spatial, and wavelet. Hence, there is an intuition that if Feature extraction combined with R, G, B color information is used; there is a possibility of improved performance of the steganographic detection techniques, which is the basis for proposition in this paper.

Earlier research work has focused only on DCT or spatial or wavelet domain for steganalysis of JPEG images. Also, the JPEG images used in most of the techniques were gray scale or prepared by the researchers themselves. In this paper, the images were used from Berkley's image dataset BSD300 (BSD300 Image Dataset) in its original format without any modifications and features from the three domains have been extracted and classified by the SVM classifier (Cristianini and Shawe-Taylor, 2000).

4. Proposed technique

4.1. Introduction

The concept of image calibration to obtain the statistics of the DCT coefficients has been proposed in (Fridrich, 2005). This technique has been used in our dilation process after decomposing the image into RGB components. The statistics in spatial, frequency, and wavelet domains are obtained and statistical feature values are calculated. These features are extracted from various sets of images, each set

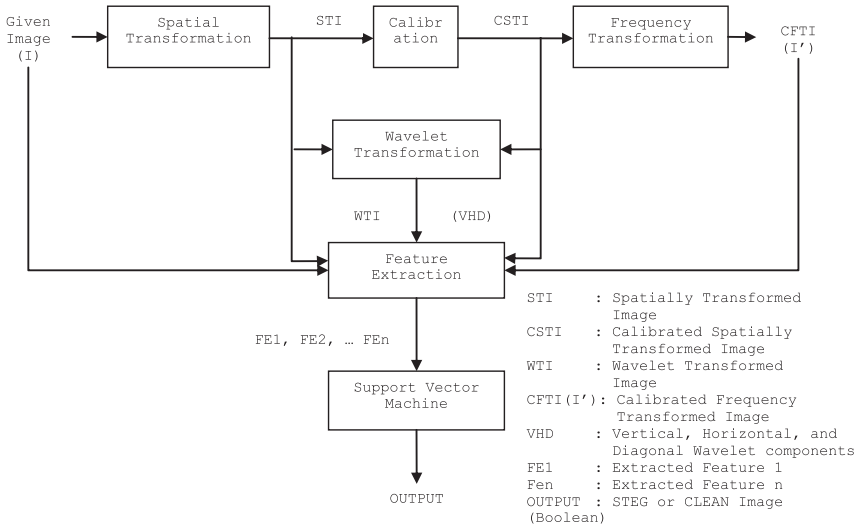

Fig. 1. Block schematic of proposed technique.

being prepared with well known steganography algorithms. Then all the features are put together for training in SVM classifier to get a trained model. Test images are compared with the trained SVM model and get classified as steg or clean image.

4.2. Block schematic of proposed technique

The block schematic of the proposed technique is given in Fig. 1. Let the JPEG image be denoted as I, spatially transformed image as STI, calibrated spatially transformed image as CSTI, calibrated JPEG image as CFTI (I′), wavelet transformed image as WTI, and the vertical, horizontal, and diagonal wavelet components as VHD.

Any given JPEG image has to be first split into three RGB components and then each component passes through the following feature extraction and classification algorithm using SVM:

Feature extraction and classification algorithm using SVM
Step-1: Divide the given JPEG image I into 8 × 8 DCT blocks
Step-2: Perform the spatial transformation over I to obtain STI
Step-3: Crop the image STI by 4 × 4 from all the sides to obtain the calibrated image CSTI
Step-4: Perform the 2-level wavelet transformation over STI and CSTI to obtain VHD at each level.
Step-5: Perform the frequency transformation on CSTI to obtain the image I′.
Step-6: Extract the frequency domain statistics from I and I′ as follows:

a. Find the mean, variance, skewness, and kurtosis of I and I′
b. Find the global histogram of AC coefficients of I
c. Find the histogram of AC coefficient differences between adjacent DCT blocks of I
d. Find the co-occurrence matrix of coefficients in the same location between I and I′
e. Find the co-occurrence matrix of coefficients at all locations along the diagonals of DCT blocks between I and I′
f. Find the global histogram of AC coefficients at all locations along the diagonals of DCT blocks of I
g. Find the histogram of adjacent pixel differences along the boundaries of DCT blocks

Step-7: Extract the spatial domain statistics from STI and CSTI as follows:

a. Find the mean and variance of STI and CSTI
b. Find the co-occurrence matrix of adjacent pixel differences in STI
c. Find the co-occurrence matrix of pixel values in same location in STI and CSTI
d. Find the co-occurrence matrix of adjacent pixel value differences in same location between STI and CSTI

Step-8: Extract the wavelet domain features from the VHD as follows:

a. Find the mean, variance, skewness and kurtosis of VHD of level-1
b. Find the mean, variance, skewness and kurtosis of VHD of level-1

Row/ Col	1	2	3	4	5	6	7	8
1	DC	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)
2	(2,1)	(2,2)	(2,3)	.	.	.	(2,7)	
3	(3,1)	.	.	.	.	(3,6)		
4	.	.	.	.	.			
5	.	.	.	.				
6	.	.	.					
7	.	.						
8	(8,1)							

Fig. 2. 8 × 8 DCT block.

Table 1
Frequency domain statistics.

Sr. no.	Statistics name
1.	Mean of DCT coefficients of I
2.	Variance of DCT coefficients of I
3.	Skewness of DCT coefficients of I
4.	Kurtosis of DCT coefficients of I
5.	Mean of DCT coefficients of I′
6.	Variance of DCT coefficients of I′
7.	Skewness of DCT coefficients of I′
8.	Kurtosis of DCT coefficients of I′
9.	Global histogram of AC coefficients of I
10.	Histogram of AC coefficient differences between adjacent DCT blocks of I
11.	Co-occurrence matrix of coefficients in same location between I and I′
12.	Co-occurrence matrix of coefficients in specific locations between I and I′
13–35.	Histograms of AC coefficients at locations along the diagonals of DCT blocks of I
36.	Histogram of adjacent pixel differences along the DCT block boundaries

Step-9: Calculate the features from the statistics obtained
Step-10: Insert these features in the trained SVM classifier
Step-11: Output the result (steg or clean Image)

4.3. Feature extraction module

The features are calculated from the three popular domains, spatial, frequency, and wavelet, for each image component. The statistics in each domain are obtained and then features are calculated from it.

The equations of Li et al. (2010) are used for computing the co-occurrence matrices and histograms for obtaining the statistics in spatial and frequency domain. Mean, variance, skewness, and kurtosis (Flannery et al., 1986–1992) are obtained using equations (1)–(4):

$$\text{Mean } M = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}F(i,j)}{M \times N} \tag{1}$$

$$\text{Variance } V = \frac{1}{MN-1}\sum_{i=1}^{M}\sum_{j=1}^{N}(F(i,j)-M)^2 \tag{2}$$

$$\text{Skewness } S = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\frac{F(i,j)-M}{\sqrt{V}}\right]^3 \tag{3}$$

Table 2
Spatial domain statistics.

Sr. no.	Statistics name
1.	Mean of pixel values of STI
2.	Variance of pixel values of STI
3.	Mean of pixel values of CSTI
4.	Variance of pixel values of CSTI
5.	Co-occurrence Matrix of adjacent pixel differences in STI
6.	Co-occurrence Matrix of pixel value in the same location of STI and CSTI
7.	Co-occurrence Matrix of adjacent pixel value difference in same location between STI and CSTI

$$\text{Kurtosis } K = \left\{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\frac{F(i,j)-M}{\sqrt{V}}\right]^4\right\} - 3 \tag{4}$$

where, F is the particular image statistics matrix and (M, N) gives the size of the matrix F

4.3.1. Frequency domain statistics

The statistics in Frequency domain or DCT domain are obtained by dividing DCT coefficient matrix of images I and I′ into 8 × 8 DCT blocks. A DCT block is filled along the diagonal and the values after the half of the center diagonal are null or zero. All the values above the center diagonal are considered. The locations other than the shaded part in Fig. 2 are to be considered as one of the statistics along with other statistics as described in Table 1.

A total of 36 × 3 = 108 statistics are obtained from frequency domain.

4.3.2. Spatial domain statistics

For extracting the statistics in spatial domain, the decomposed image pixel values in STI and CSTI are used. Table 2 shows the statistics obtained.

A total of 7 × 3 = 21 statistics has been obtained from Spatial domain.

4.3.3. Wavelet domain statistics

Two-level wavelet decomposition for each of the RGB image components is performed as shown in Fig. 3. V, H, and D are the vertical, horizontal, and diagonal wavelet components respectively. The first order statistics obtained from each wavelet component in each level of the wavelet decomposition are given in Table 3.

Finally, there are 4 (number of statistics) × 3 (number of wavelet components) × 2 (number. of levels) = 24 wavelet statistics from the wavelet decomposition of STI. Similarly from the calibrated image CSTI, the same 24 statistics are obtained. Totally, 24 + 24 = 48 wavelet statistics have been obtained. Considering the RGB color components a total of 48 × 3 = 144 wavelet statistics are used in this paper.

4.3.4. Calculation of features

Center of Mass (COM) is calculated as a feature using the equation (5) for statistics with histograms and co-occurrence matrices:

$$\text{COM}(I) = \frac{\sum_{i=1}^{\frac{M}{2}}\sum_{j=1}^{\frac{N}{2}}F(i,j) * \text{fft2}(F(i,j))}{\text{fft2}(F)} \tag{5}$$

where, F is the particular image statistics matrix, (M, N) gives the size of the matrix F, and fft2 gives the discrete fourier transform (DFT) for a two dimension vector.

For other statistics, the statistic obtained itself is taken as a feature.

A COM value should provide the uniform distribution of values over a particular matrix. When a particular image statistic value is multiplied by its Fourier transform and divided by overall Fourier transform, it yields a value which is uniformly spread over a matrix. Thus, this equation helps us to reduce the 2-dimensional matrix to a single value without disturbing its characteristic. As DFT is central symmetric, for a DFT sequence with length N, the value of COM needed to be calculated in the range [1, N/2]. Thus, a

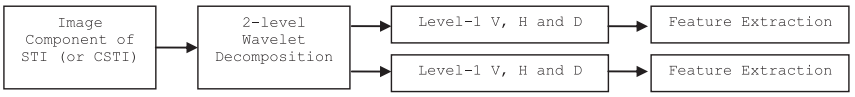

Fig. 3. Bock schematic of wavelet decomposition.

Table 3
Wavelet domain statistics.

Sr. no.	Statistics name
1.	Mean of coefficients of V (or H or D)
2.	Variance of coefficients of V (or H or D)
3.	Skewness of coefficients of V (or H or D)
4.	Kurtosis of coefficients of V (or H or D)

$103 \times 3 = 309$-dimensional feature vector has been formed in this paper.

4.4. Implementation

MATLAB R2010a from Mathworks (Matlab R2010a) has been used for all the feature extraction process. The following two toolboxes were integrated with matlab:

a) JPEG toolbox (JPEG toolbox): Useful in preprocessing of JPEG images
b) PCM (Parallel Computing Toolbox) (PCM toolbox): This tool was used for faster execution of statistics calculation functions in matlab. It creates worker threads which can execute multiple independent functions at the same time. 10 worker threads were used in our implementation in this paper.

After the feature extraction of JPEG images, SVM classifier (Support Vector Machine) has been used for training and testing of features. Also the final classification of test images, whether clean or steg is done by SVM. Fig. 4 shows the role of each of these tools in getting the output.

4.5. Experiments

4.5.1. Preparation of datasets

Berkley's image dataset BSD300 (BSD300 Image Dataset) which contains 300 JPEG images each of 481×381 resolution, with sizes varying from 23 KB to 111 KB, was used in this paper for preparing the training and testing datasets. Out of the 300 images, 200 were used for preparing training dataset and 100 were used for preparing testing dataset. Out of the 200 training images, 80 images were used as clean images and the rest 120 images were used for embedding text messages using the well known steganography algorithms. Similarly from the test images, out of 100 images, 50 images were used as clean and the rest 50 images were embedded with text messages. The well known steganography algorithms used for embedding text in training images were F5 (Westfeld, 2001), Outguess (Provos, 2001), StegHide (Hetzl, 2003), and Hide and Seek (Provos and Honeyman, 2003). For testing dataset images, steganography algorithms used for embedding text were: Invisible Secrets 4 (IS) (Invisible secrets 4), Dynamic Battle Steg (DBS), and Dynamic Filter First (DFF) (Sivasubramanian and Raju, 2013). The following varying sizes of text messages were used for the experiments conducted:

Experiment-1: With hidden messages of size M1 – 110–126 bytes
Experiment-2: With hidden messages of size M2 – 55–65 bytes
Experiment-3: With hidden messages of size M3 – 27–31 bytes

Ten different text messages were randomly embedded in the images in every experiment. The ratio of embedding comes out to be approximately 0.09% in all the embedded images in Experiment-3.

4.5.2. Experiments using Stego Tools

The steg images used in our experiment were prepared using the Stego Tools widely available on the internet. Table 4 gives the characteristics of each steganography algorithm used and their respective Stego Tools. For conducting experiments and for evaluating the performance of the proposed technique, 25 images were selected randomly out of the chosen 120 training images for embedding with text messages using the Stego Tools of well known steganography algorithms as shown in Fig. 5. For each steganography algorithm, this process was repeated. For the test images, out of the chosen 50 images, 25 images were embedded randomly with the available Stego Tools of the given algorithms as shown in Fig. 5. For each steganography algorithm, the process was repeated. The different characteristics of each steganography algorithm described in Table 4 make our training and testing dataset versatile in

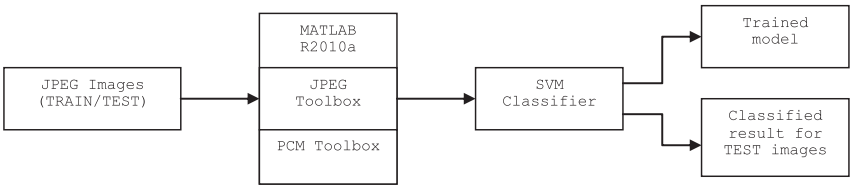

Fig. 4. Implementation of proposed technique.

Table 4
Characteristics of well known steganography algorithms and their respective Stego Tools.

Steganography algorithms	Characteristics	Stego Tools
F5 (Westfeld, 2001)	• Offers a large steganographic capacity • Implements matrix encoding improving the efficiency of embedding. • Employs permutative straddling to uniformly spread out the changes over the whole steganogram	F5-steganography (F5-steganography)
Outguess (Provos, 2001)	• Uses a pseudo-random number generator to select DCT coefficients at random • Allows the insertion of hidden information into redundant bits of data sources • Preserves statistics based on frequency counts • Can determine maximum message size than can be hidden	OutGuess 0.2 (Provos)
StegHide (Hetzl, 2003)	• The color-respectively sample-frequencies are not changed • Undetectable by color-frequency based statistical tests	StegHide 0.5.1 (Hetzl, 2003)
Hide and Seek (Provos and Honeyman, 2003)	• Randomly distributes the message across the image • Uses a password to generate a random seed, then uses this seed to pick the first position to hide in • It continues to randomly generate positions until it has finished hiding the message • More useful to figure out areas of the image where it is better to hide in	diit-1.5 (Digital Invisible Ink Toolkit)
Invisible Secrets (Invisible secrets 4)	• Encrypts and hides the message data on innocent surfaces of image. • Uses strong file encryption algorithms (like AES).	Invisible Secrets 4 (Invisible secrets 4)
DBS (Sivasubramanian and Raju, 2013)	• Message is hidden randomly in the best parts of the image. • Use of filter ensures message is hidden in least noticeable parts of image • Uses dynamic programming to make the hiding process faster and less memory intensive	diit-1.5 (Digital Invisible Ink Toolkit)
DFF (Sivasubramanian and Raju, 2013)	• Algorithm filters the image using one of the inbuilt filters and then hides in the highest filter values first. • Filters the most significant bits, and leaves the least significant bits to be changed • Uses dynamic programming to make the hiding process faster and less memory intensive	diit-1.5 (Digital Invisible Ink Toolkit)

nature, thus helping in making our evaluation technique more effective.

The experiments carried out along with their significance are discussed in the following Section 4.5.3.

4.5.3. Summary of experiments

Three experiments were conducted with the objective of finding the least embedding text size the proposed approach can detect in an image. The images in

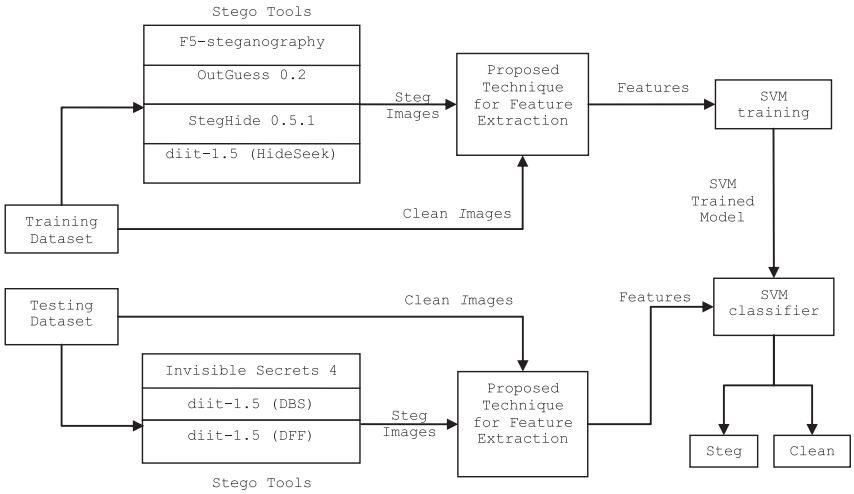

Fig. 5. Usage of Stego Tools in the experiments.

Table 5
Results for Experiment-1.

Domain name	True negatives (TN)			TP	False positives (FP)			FN
	Invisible secrets	DBS	DFF	Clean	Invisible secrets	DBS	DFF	Clean
Spatial	8	23	23	39	17	2	2	11
Frequency	22	20	20	43	3	5	5	7
Wavelet	19	19	23	42	6	6	2	8
Proposed	24	24	24	42	1	1	1	8

Table 6
Results for Experiment-2.

Domain name	True negatives (TN)			TP	False positives (FP)			FN
	Invisible secrets	DBS	DFF	Clean	Invisible secrets	DBS	DFF	Clean
Spatial	9	23	21	48	16	2	4	2
Frequency	21	23	23	37	4	2	2	13
Wavelet	18	20	22	42	7	5	3	8
Proposed	22	22	23	42	3	3	2	8

Table 7
Results for Experiment-3.

Domain name	True negatives (TN)			TP	False positives (FP)			FN
	Invisible secrets	DBS	DFF	Clean	Invisible secrets	DBS	DFF	Clean
Spatial	24	20	21	37	1	5	4	13
Frequency	23	20	19	37	2	5	6	13
Wavelet	24	19	18	42	1	6	7	8
Proposed	24	20	20	41	1	5	5	9

Table 8
Results for Experiment-4.1

Domain name	True negatives (TN)			TP	False positives (FP)			FN
	Invisible secrets	DBS	DFF	Clean	Invisible secrets	DBS	DFF	Clean
Spatial	24	20	21	37	1	5	4	13
Frequency	23	20	18	37	2	5	7	13
Wavelet	24	19	18	42	1	6	7	8
Proposed	24	19	20	41	1	6	5	9

Experiment-1 were embedded with text messages of sizes M1 varying from 110 to 126 bytes in training as well as in testing images. The results obtained are shown in Table 5. The message size M2 for embedding in Experiment-2 was approximately reduced to half of M1 equal to 55–65 bytes. The results obtained are shown in Table 6. The message size M3 for embedding in Experiment-3 was approximately reduced to one fourth of M1 equal to 27–31 bytes. The results obtained are shown in Table 7.

It was observed that the detection rate was reducing as the embedding text size was reduced. So, in order to increase the detection rate, one experiment, Experiment-4 was conducted with reduced features in frequency domain. The non-contributing elements for detecting were chosen as features for reduction. That is, the higher coefficients in DCT block of an image having more zeros were excluded.

Experiment-4.1 was conducted by reducing the statistics in Experiment-3 by excluding the histograms of all

Table 9
Results for Experiment-4.2

Domain name	True negatives (TN)			TP	False positives (FP)			FN
	Invisible secrets	DBS	DFF	Clean	Invisible secrets	DBS	DFF	Clean
Spatial	24	20	21	37	1	5	4	13
Frequency	24	23	20	35	1	2	5	15
Wavelet	24	19	18	42	1	6	7	8
Proposed	24	21	20	42	1	4	5	8

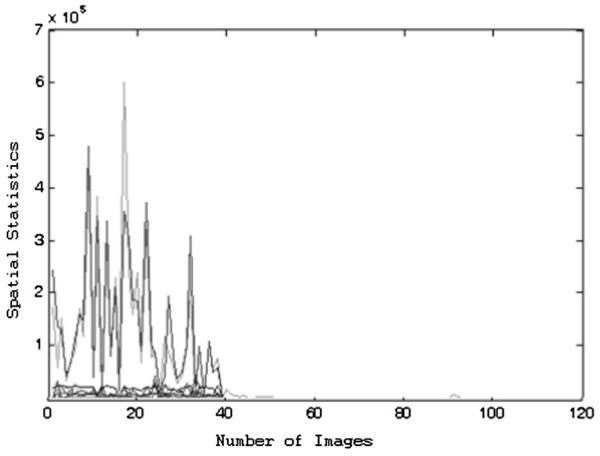

Fig. 6. Graph of spatial statistics versus number of images.

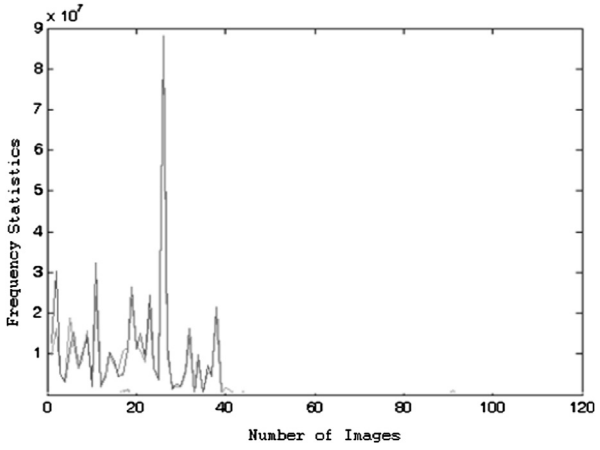

Fig. 7. Graph of frequency statistics versus number of images.

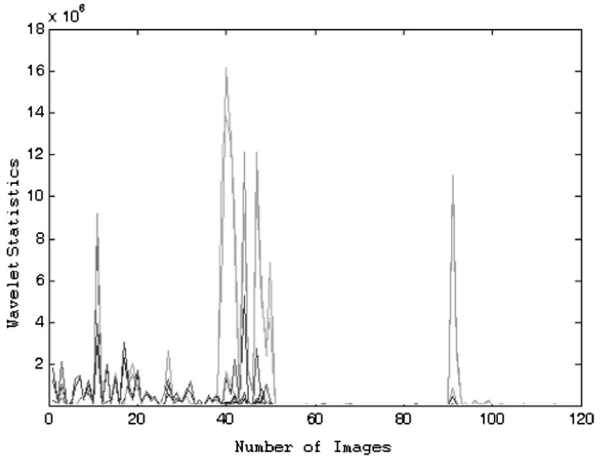

Fig. 8. Graph of wavelet statistics versus number of images.

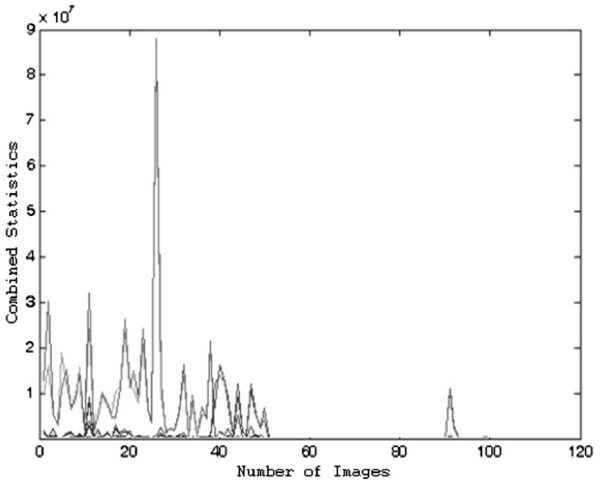

Fig. 9. Graph of combined statistics versus number of images.

location values along diagonal 7 and diagonal 8 and experimenting with the 32 × 3 frequency feature set. Results obtained are given in Table 8. Experiment-4.2 was conducted by reducing the statistics in Experiment-3 by excluding the histograms of all location values along diagonal 5 to diagonal 8 and experimenting with the 21 × 3 frequency feature set. Results obtained are shown in Table 9.

Tables 5–9 list the computed value for True Negatives (TN), True Positives (TP), False Positives (FP), and False Negatives (FN) for three different steganography techniques, viz., IS, DBS, and DFF. TN shows the number of steg images detected accurately and FP shows the number of undetected steg images. Columns TP and FN show the detection results for clean images. TP shows the number of clean images detected as clean images while FN shows the number of clean images detected as steg images. The first three rows list the detection results of the individual image domain while the last row lists the detection results of the proposed approach. On comparing the results of the proposed approach with the results of the individual domain, it can be observed that the FP is comparatively reduced.

4.6. Performance evaluation and analysis of results

4.6.1. Domain statistic versus number of images

Figs. 6–9 show the graphs of Statistics versus Number of images for Experiment-4.2. Images numbered from 1 to 50 are the clean images and from 51 to 120 are the steg images.

From Fig. 6, it can be seen that up to 50 images, clean images were detected correctly as clean images. Further, it

Table 10
OSR% for Experiment-1.

Domain name	Invisible secrets	DBS	DFF
Spatial	62.67	82.67	82.67
Frequency	86.67	84.00	84.00
Wavelet	81.33	81.33	86.67
Proposed	88.00	88.00	88.00

Table 11
OSR% for Experiment-2.

Domain name	Invisible secrets	DBS	DFF
Spatial	76.00	94.67	92.00
Frequency	77.33	80.00	80.00
Wavelet	80.00	82.67	85.33
Proposed	85.33	85.33	86.67

Table 12
OSR% for Experiment-3.

Domain name	Invisible secrets	DBS	DFF
Spatial	81.33	76.00	77.33
Frequency	80.00	76.00	74.67
Wavelet	88.00	81.33	80.00
Proposed	86.67	81.33	81.33

Table 13
OSR% for Experiment-4.1.

Domain name	Invisible secrets	DBS	DFF
Spatial	81.33	76.00	77.33
Frequency	80.00	76.00	73.33
Wavelet	88.00	81.33	80.00
Proposed	86.67	80.00	81.33

can also be seen that some of the clean images were detected as steg images. In Fig. 7, the steg images are correctly detected, but some clean images also behave as steg images. In Fig. 8, very few clean images were detected correctly. Finally, in Fig. 9, most of the clean and steg images were detected correctly.

4.6.2. Comparison of Overall Success Rate (OSR)

For each experiment, its Overall Success Rate (OSR) (Kaufmann, 2005), which is the ratio of number of correct classifications to the total number of classifications, is computed using equation (6) and is tabulated in Tables 10–14.

$$\mathrm{OSR} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{6}$$

where, TP – True Positives, TN – True Negatives, FP – False Positives, and FN – False Negatives.

In our experiments we have not only focused on the steg images but also on the clean images. It is highly possible that a clean image may be wrongly suspected for a steg image. To show that our proposed approach is more

Table 14
OSR% for Experiment-4.2.

Domain name	Invisible secrets	DBS	DFF
Spatial	81.33	76.00	77.33
Frequency	78.67	77.33	73.33
Wavelet	88.00	81.33	80.00
Proposed	88.00	84.00	82.67

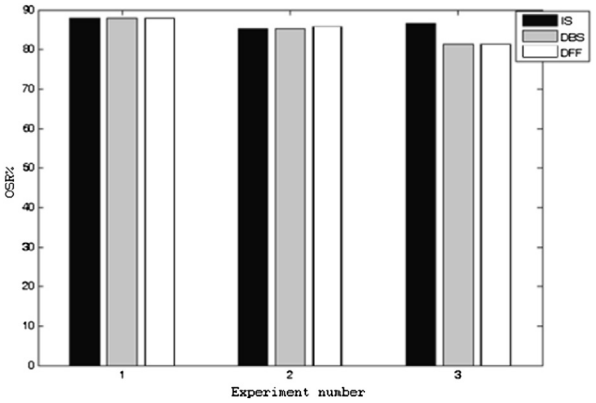

Fig. 10. Graph of OSR versus experiment number.

effective, we compared it with the individual domain results. The effectiveness of any steganography algorithm is judged by the OSR value. The graph of OSR for various experiments conducted is given in Fig. 10. Also the graph of OSR for the experiment with reduced feature set is given in Fig. 11. As can be seen from Tables 10–14, the OSR value of the proposed approach, compared to previous three domains individually, has been improved. Also the proposed approach has been able to detect the images with embedding ratio as low as 0.09% approximately (Table 12 and Fig. 10). The OSR percentage values in Table 14 and Fig. 11 show the slight improvement in detection of Steg images with low embedding on reducing the features in frequency domain.

4.6.3. Comparison of embedding message size

Algorithm-1 (Dumitrescu et al., 2003), Algorithm-2 (Shi et al., 2005), and Algorithm-3 (Xu et al., 2007) have been reported to detect the presence of hidden message up to 3%, 0.25%, and 5% of embedding message size respectively. We have conducted three experiments of varying embedding message size, viz., Experiment-1, Experiment-2, and Experiment-3 with approximately 0.5%, 0.3%, and 0.09% respectively. The graph comparing the embedding message size with the different existing algorithms and the proposed one in this paper are shown in Fig. 12.

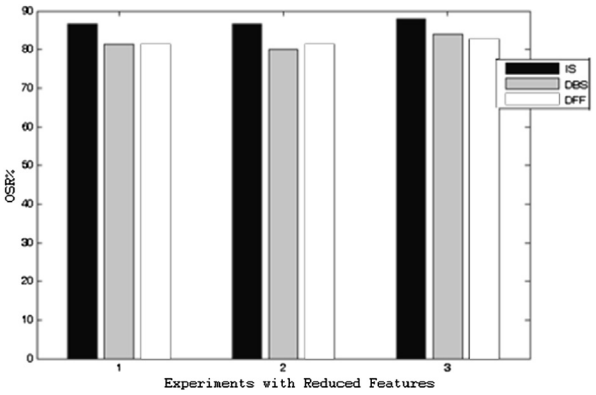

Fig. 11. Graph of OSR versus experiments with reduced features.

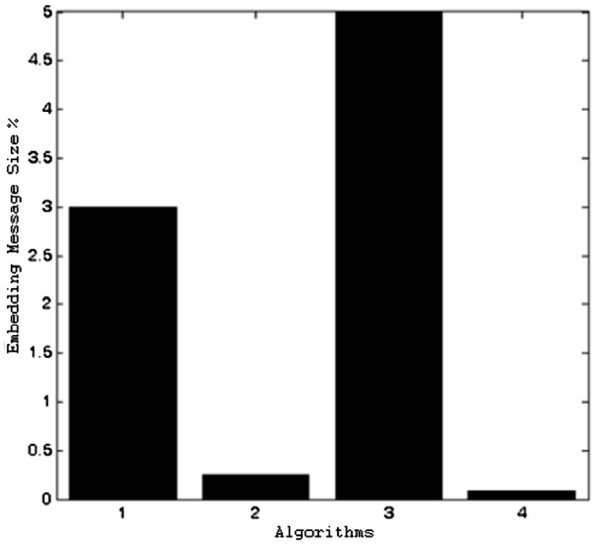

Fig. 12. Graph of embedding message size versus algorithms.

It is evident from Fig. 12 that the proposed solution in this paper, detects the steg images, with embedding ratio of as low as 0.09%, as steg images, which has not been reported so far.

5. Conclusion

The proposed problem of finding if something is hidden or not in a given image is a challenging one. A lot of research is carried out but still the existing steganalysis techniques are inadequate to detect the presence of hidden information. Such steganalysis technique either focused on gray scale images, single image domain, or lacked the proper blend of RGB domain with feature extraction process. In this research, the focus has been to improve the Overall Success Rate (OSR). Our proposed Blind Steganalysis technique makes an effective effort in detecting such images. The contributions made in this paper are, the use of dilation by decomposing image into RGB components and extracting features from spatial, frequency, and wavelet domain of each of these components. These contributions made our technique effective in terms of detecting the JPEG images with as low as 0.09% embedding and with 88% OSR value. Further, the clean images have been included in our experiment for testing which has been done and reported in very few of the existing researches. The experimental results confirm that the proposed technique is more robust in not detecting a clean image as steg image. Moreover, the JPEG images used in the experiment were without any format conversions. They were directly used from the JPEG images dataset. Thus, it is observed that given any random JPEG image from network for detection, the proposed technique gives effective results.

References

Avcıbas Ismail, Memon Nasir, Sankur Bülent. Steganalysis using image quality metrics. IEEE Trans Image Process February 2003;12(2):221–9.
Brown A. S-tools version 4.0. http://members.tripod.com/steganography/stego/s-tools4.html [accessed on 22.04.13].
BSDS300 Image Dataset. http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/ [accessed on 09.02.13].
Cai Hong, Agaian Sos S, Wang Yufeng. An effective algorithm for breaking F5. In: IEEE 7th workshop on multimedia signal processing Oct–Nov 2005. pp. 1–4.
Chen B, Wornell GW. Digital watermarking and information embedding using dither modulation. In: Proceedings of IEEE MMSP 1998. pp. 273–8.
Cox IJ, Kilian J, Leighton T, Shamoon T. Secure spread spectrum watermarking for multimedia. IEEE Trans Image Process 1997:1673–87.
Cristianini Nello, Shawe-Taylor John. An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press; 2000. ISBN: 0521780195 [chapter 6].
Digital Invisible Ink Toolkit, diit-1.5. https://code.google.com/p/f5-steganography/ [accessed on 15.04.13].
Dumitrescu Sorina, Wu Xiaolin, Wang Zhe. Detection of LSB steganography via sample pair analysis. IEEE Trans Signal Process July 2003;51(7):1995–2007.
F5-steganography. https://code.google.com/p/f5-steganography/ [accessed on 26.01.13].
Flannery Brian P, Teukolsky Saul A, Vetterling William T. Numerical recipes in Fortran 77: the art of scientific computing. ISBN 0-521-43064-X. Copyright (C) 1986–1992 by Cambridge University Press. Programs Copyright (C) 1986–1992 by Numerical Recipes Software. p. 604–7.
Fridrich Jessica. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In: Proceedings of the 6th international information hiding workshop May 23–25, 2005. pp. 67–81. Toronto, Ontario, Canada.
Fridrich Jiri, Du Rui, Long Meng. Steganalysis of LSB encoding in color images. In: IEEE international conference on multimedia and expo, vol. 3; 2000. pp. 1279–82. New York.
Fridrich Jessica, Goljan Miroslav, Du Rui. Detecting LSB steganography in color and GrayScale images. IEEE Multimed Oct–Dec 2001;8(4):22–8.
Fridrich Jessica, Goljan Miroslav, Hogea Dorin. Steganalysis of JPEG images: breaking the F5 algorithm. Springer linkIn Lecture notes in computer science, vol. 2578; 2003. pp. 310–23.
Fridrich Jessica, Goljan Miroslav, Soukal David. Higher-order statistical steganalysis of palette images. In: Proceedings of the SPIE 5020, security and watermarking of multimedia contents V 2003. pp. 178–90.
Hetzl S. StegHide 0.5.1. http://steghide.sourceforge.net; 2003 [accessed on 12.02.13].
Holotyak Taras, Fridrich Jessica, Voloshynovskiy Sviatoslav. Blind statistical steganalysis of additive steganography using wavelet higher order statistics. In: CMS'05 proceedings of the 9th IFIP TC-6 TC-11 international conference on communications and multimedia security 2005. pp. 273–4.
Huang J, Shi YQ. An adaptive image watermarking scheme based on visual masking. IEEE Electron Lett April 1998;34(8):748–50.
Invisible secrets 4. http://www.invisiblesecrets.com/ [accessed on 18.10.12].
JPEG toolbox. http://philsallee.com/jpegtbx/index.html [accessed on 03.10.12].
Kaufmann Morgan. Data mining, practical machine learning tools and techniques. 2nd ed.; 2005. pp. 161–3.
Korejwa J. Jsteg shell 2.0. http://www.tiac.net/users/korejwa/steg.htm.
Li Zhuo, Lu Kuijun, Zeng Xianting, Pan Xuezeng. A blind steganalytic scheme based on DCT and spatial domain for JPEG images. J Multimed June 2010;5(3):200–7.
Luo Xiangyang, Liu Fenlin, Lian Shiguo, Yang Chunfang, Gritzalis Stefanos. On the typical statistic features for image blind steganalysis. IEEE J Sel Areas Commun August 2011;29(7):1404–22.
Machado, R. http://www.fqa.com/ezstego/ [accessed on 07.02.13].
Matlab R2010a. http://www.mathworks.in/ [accessed on 15.07.12].
Mielikainen Jarno. LSB matching revisited. IEEE Signal Process Lett May 2006;13(5):285–7.
News article. http://www.zdnet.com/news/terrorists-and-steganography/116733.
PCM toolbox. http://www.mathworks.in/products/parallel-computing/ [accessed on 29.11.12].
Piva A, Barni M, Bartolini E, Cappellini V. DCT-based watermark recovering without resorting to the uncorrupted original image. In: Proceedings of the ICIP 97, vol. 1. p. 520.
Provos Niels. Defending against statistical steganalysis. In: Proceedings of 10th unisex security symposium August 2001. pp. 323–36. Washington, DC.
Provos N, OutGuess 0.2. http://www.outguess.org [accessed on 22.03.13].
Provos N, Honeyman P. Hide and seek: an introduction to steganography. IEEE Secur Priv May–June 2003:32–44.

Rencher AC. Methods of multivariate analysis. New York: John Wiley; 1995. p. 10 [chapter 6].

Shi Yun Q, Xuan Guorong, Zou Dekun, Gao Jianjiong, Yang Chengyun, Zhang Zhenping, et al. Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network. In: International conference on multimedia and expo 2005.

Sivasubramanian S, Raju Janardhana. Advanced embedding of information by secure key exchange via trusted third party using steganography. Int J Latest Res Sci Technol January–February, 2013;2(1):536–40.

Steganos II Security Suite. http://www.steganos.com/english/steganos/download.htm.

Support Vector Machine. http://www.mathworks.in/help/stats/support-vector-machines-svm.html [accessed on 30.12.12].

Westfeld Andreas. F5 – a steganographic algorithm. In: Proceedings of the 4th international workshop on information hiding. London, UK: Springer-Verlag; 2001. pp. 289–302.

Xu Bo, Wang Jiazhen, Liu Xiaqin, Zhang Zhe. Passive steganalysis using image quality metrics and multi-class support vector machine. In: IEEE third international conference on natural computation 2007. pp. 215–20.

Zhang Xue, Zhong Shang-Ping. Blind steganalysis method for bmp images based on statistical MWCF and f-score method. In: Proceedings of the 2009 international conference on wavelet analysis and pattern recognition 12–15 July 2009. pp. 442–7. Baoding.