Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **2**, March **2017**

Detection System for Grey and Colour Images Based on Extracting Features of Difference Image and Renormalized Histogram

Ahd Aljarf, Saad Amin, John Filippas, James Shuttelworth

Computing department, Coventry University, UK
aljarfa@uni.coventry.ac.uk, s.amin@coventry.ac.uk, csy111@coventry.ac.uk, csx239@coventry.ac.uk

Received March, 2016; revised September, 2016

Abstract. *The literature has introduced many steganalysis methods intended to combat specific steganography techniques and to detect particular image formats. This paper proposes a detection system based on extracting histogram features. The features are extracted by exploiting the histogram of difference image, which is usually a generalised Gaussian distribution centred at 0. The histogram of difference image and the renormalized histogram are created for clean and stego images, therefore using the peak value and renormalized histogram as features for classification. To obtain the difference between neighbouring pixels, the difference images are computed for four directions (vertical, horizontal, diagonal, and anti-diagonal). The renormalized histogram of the difference image is created a number of times (n) for the four directions. This work implements two commonly-used steganography methods: the Least Significant Bit (LSB) and F5 algorithm to create a large database of stego images for system evaluation. Colour and grey images with different formats are chosen for training and testing the system. These formats are lossless and lossy compressions, with all features extracted from each colour channel (RGB) separately. The size of hidden files plays an important role in terms of detection. Therefore, to improve the proposed systems detection capacity, different sizes of hidden files have been considered. The proposed detection system was trained and tested to distinguish stego images from clean ones using the Discriminant Analysis (DA) classification method and Multilayer Perceptron neural network (MLP). The experimental results prove that the proposed system possesses reliable detection ability and accuracy. The chosen classification methods show dissimilar performance in terms of classifying grey and colour images. The system holds more generalisability than previous systems by covering different types of stego images, image formats and hidden file sizes. In addition, extensive experimental results show that the proposed steganalysis system outperforms some previous detection methods. .*

Keywords: Image Steganography, Images Steganalysis, Stegoimages, Security, Histogram Features.

1. **Introduction.** Steganography is the science of embedding secret data in an appropriate cover object. Derived from the Greek words *stegos* meaning 'cover' and *grafia* meaning 'writing', steganography is defined as 'covered writing' [1]. There are four types of cover objects: image, text, audio and video. Corresponding to these different cover objects are many types of steganography, such as image steganography and text-file steganography [1]. Images are ubiquitous on the Internet and can be used as carrier objects without raising much suspicion. As a result, images represent the most-commonly used cover to hide files. Steganography aims to hide files in cover objects, while steganalysis seeks to detect such hidden files. Steganalytic systems are used to determine whether or not an image

contains a hidden file by analysing various features of stego images (images containing hidden files) and clean images (images containing no hidden files). Generally speaking, steganalysis is performed in one of two ways: signature analysis and blind detection. In signature analysis, the steganographic hiding method is known, making detection easier. Embedding algorithms always leave a particular signature that can be tracked for detection [2, 3]. In contrast, the hiding method remains unknown in the blind detection technique. Although this detection technique is obviously the most commonly used, it is by far the most difficult to use [2, 3]. Blind steganalysis uses the following general structure: (1) stego signal estimation, (2) feature extraction and (3) classification. Researchers have introduced a wide variety of steganography methods and tools for hiding files in a cover object. These methods and tools support different image formats and use many steganography techniques to embed hidden files. Researchers have also developed many steganalysis methods to combat specific steganography techniques and detect particular image formats. However, no single steganalysis method or tool can detect all types of steganography and support all available image formats. Thus, there is a need to develop more robust steganalysis systems to deal with different image formats and to break various steganography methods [4, 5]. This paper proposes a detection system based on extracting histogram features, a system inspired by the features recently introduced for grey images by [6,7]. However, the proposed system differs in terms of the extracted histogram features used to train and test the system. All histogram features were improved to be extracted from the RGB channels of the colour images. Moreover, the image database used to train and test the system includes grey and colour images, lossless and lossy format. The stego images were created using the LSB and F5 steganography algorithm. The proposed system design works as a blind form of steganalysis. In this type of steganalysis, the system does not target specific steganography methods or specific image formats. Rather, it extracts many histogram features by exploiting the histogram of difference image, which is usually a generalized Gaussian distribution centred at 0. Detection using different types of image features is the most useful technique for blind steganalysis [7]. Therefore, the proposed detection system is trained to detect different types of stego images created by various steganography methods. The stepwise Discriminant Analysis (DA) and Multi-Layer Perceptron neural network (MLP) classification methods are used to train and test the system. Stepwise Discriminate Analysis (DA) attempts to find the best set of predictors, and it is often used in exploratory situations to identify variables from amongst a larger set that might be used later in a more rigorous, theoretically-driven study [8, 9]. However, the Multi-Layer Perceptron (MLP) represents a common example of the neural network's predictive application. MLP and RBF are supervised in the sense that the model-predicted results can be compared against the target variables' known values [10]. The classification methods used showed dissimilar performance in terms of classifying grey and colour images. The rest of the paper is organised as follows: section 2 discusses the extracted histogram features, while sections 3 and 4 explain the stages and architecture of the proposed system. Experimental results are presented in sections 5, 6 and 7. Section 8 concludes the paper and proposes directions for future research.

2. **Extracted Histogram Features.** The peak value and the renormalized histogram were used as features for classification [7]. The peak value of the histogram decreased after LSB embedding, while the renormalized histogram (the ratio of the histogram to the peak value) increased. In addition, the F5 algorithm does modify the histogram of Discrete Cosine Transform (DCT) coefficients, preserving some of its crucial characteristics, such as its monotonicity and monotonicity of increments [11]. The histogram of the cover image

can be calculated from the stego-image. Because F5 modifies the histogram in a well-defined manner, the number and the modified coefficients can be calculated by comparing the estimated histogram with the histogram of the stego and clean images [11]. The histogram features of the colour images were extracted from the three colour channels: red, green, blue (RGB). Each colour channel was treated and analysed separately [6, 7]. The features were extracted by exploiting the histogram of difference image, which is usually a generalised Gaussian distribution centred at 0. Compared with the histogram of the original image, the histogram of difference image can be more precisely characterised as a Generalised Gaussian Distribution (GGD) centred at 0. In this case, the 'renormalized histogram' is defined as the ratio of the histogram to the peak value; accordingly, the peak value of the renormalized histogram is an exact 1 [7]. The histogram of difference image and the renormalized histogram are created for clean and stego images, therefore, the peak value and the renormalized histogram are used as features for classification [7].

$$hs(k) = \alpha 4hc(k1) + 1 - \alpha 2hc(k) + \alpha 4hc(k+1) \tag{1}$$

where $\alpha \in (0,\ 1)$ is the embedding rate, hc and hs are the histograms of cover and stego images. The resulting stego image's histogram is a regularization of the cover images histogram.

Many notations are considered, let I be an image, and h be the (normalized) histograms of I:

$$\mathbf{h}\left(\mathbf{k}\right) = \frac{\#\left\{(\mathbf{i},\mathbf{j}) : \mathbf{I}\left(\mathbf{i},\mathbf{j}\right) = \mathbf{k}\right\}}{\mathbf{N}} \tag{2}$$

Where the symbol # denotes the cardinal number of a set, and N is the total number of image pixels. Iv is difference of neighbouring pixels in the vertical direction:

$$Iv(i,j) = I(i,j) - I(i+1,j) \tag{3}$$

The renormalized histogram of Iv is defined as h'v:

$$\mathbf{h'v}\left(\mathbf{k}\right) = \frac{\mathbf{hv}\left(\mathbf{k}\right)}{\mathbf{hv}\left(0\right)} = \frac{\#\left\{(\mathbf{i},\mathbf{j})\ : \mathbf{Iv}\left(\mathbf{i},\mathbf{j}\right) = \mathbf{k}\right\}}{\#\left\{(\mathbf{i},\mathbf{j})\ : \mathbf{Iv}\left(\mathbf{i},\mathbf{j}\right) = 0\right\}} \tag{4}$$

Where hv is the histogram of Iv. Ih is the difference of neighbouring pixels in the horizontal direction:

$$Ih(i,j) = I(i,j) - I(i,j+1) \tag{5}$$

$$h'h\ \ (k) = \frac{\mathbf{hh}\left(\mathbf{k}\right)}{\mathbf{hh}\left(0\right)} = \frac{\#\left\{(\mathbf{i},\mathbf{j})\ : \mathbf{Ih}\left(\mathbf{i},\mathbf{j}\right) = \mathbf{k}\right\}}{\#\left\{(\mathbf{i},\mathbf{j})\ : \mathbf{Ih}\left(\mathbf{i},\mathbf{j}\right) = 0\right\}} \tag{6}$$

Id is the difference of neighbouring pixels in the diagonal direction:

$$Id(i,j) = I(i,j) - I(i+1,j+1) \tag{7}$$

$$h'd\ \ (k) = \frac{\mathbf{hd}\left(\mathbf{k}\right)}{\mathbf{hd}\left(0\right)} = \frac{\#\left\{(\mathbf{i},\mathbf{j})\ : \mathbf{Id}\left(\mathbf{i},\mathbf{j}\right) = \mathbf{k}\right\}}{\#\left\{(\mathbf{i},\mathbf{j})\ : \mathbf{Id}\left(\mathbf{i},\mathbf{j}\right) = 0\right\}} \tag{8}$$

Ia is the difference of neighbouring pixels in the anti-diagonal direction:

$$Ia(i,j) = I(i,j+1) - I(i+1,j) \tag{9}$$

$$h'a\left(k\right) = \frac{\mathbf{ha}\left(\mathbf{k}\right)}{\mathbf{ha}\left(0\right)} = \frac{\#\left\{(\mathbf{i},\mathbf{j})\ : \mathbf{Ia}\left(\mathbf{i},\mathbf{j}\right) = \mathbf{k}\right\}}{\#\left\{(\mathbf{i},\mathbf{j})\ : \mathbf{Ia}\left(\mathbf{i},\mathbf{j}\right) = 0\right\}} \tag{10}$$

The difference of the neighbouring vertical pixel difference is extracted in many notations from the main four equations described above: *Ivv, Ivh, Ivd, Iva, Ihh, Ihd, Iha, Idd, Ida, Iaa*, as the following: *Ivv* represents the difference of neighbouring vertical pixel differences in the vertical direction:

$$Ivv(i,j) = Iv(i,j) - Iv(i,j+1) \tag{11}$$

Ivh indicates the difference of neighbouring vertical pixel differences in the horizontal direction:

$$Ivh(i,j) = Iv(i,j) - Iv(i+1,j) \tag{12}$$

Ivh represents the difference of neighbouring vertical pixel differences in the diagonal direction:

$$Ivd(i,j) = Iv(i,j) - Iv(i+1,j+1) \tag{13}$$

Ivh is the difference of neighbouring vertical pixel differences in the anti-diagonal direction:

$$Iva(i,j) = Iv(i,j+1) - Iv(i+1,j) \tag{14}$$

$$Ihh(i,j) = Ih(i,j) - Ih(i+1,j) \tag{15}$$

$$Ihd(i,j) = Ih(i,j) - Ih(i+1,j+1) \tag{16}$$

$$Iha(i,j) = Ih(i,j+1) - Ih(i+1,j) \tag{17}$$

$$Idd(i,j) = Id(i,j) - Id(i+1,j+1) \tag{18}$$

$$Ida(i,j) = Id(i,j+1) - Id(i+1,j) \tag{19}$$

However, the histogram of difference image can be more precisely characterised because neighbouring pixels are often highly correlated. Therefore, the features are derived from the difference images histogram instead of the histogram of the original image. Let *Ic* be a gray-scale clean image, *Is* be its stego image embedded by LSB or F5 algorithm with embedding rate α, and *hc* and *hs* be the histograms of *Ic* and *Is*.

According to equation (1), the following seems true:

$$hs = f\alpha hc \tag{20}$$

Where $f\alpha$ is the distribution of embedding noise:$f\alpha(0) = 1\alpha/2; f\alpha(1) = f\alpha(1) = \alpha/4$

Furthermore, *Ivc* and *Ivs* denote the vertical difference images of *Ic* and *Is*, and *hvc* and *hvs* represent the histograms of *Ivc* and *Ivs*. At this point, *hvs* is the convolution of *hvc* with a certain kernel function,

$$hvs = g\alpha hvc \tag{21}$$

Where $g\alpha = f\alpha f\alpha$.

Several reliable histogram features are obtained from this important property [7].

In addition, the renormalized histogram is symmetric to 0 and steep in shape. We then simply take.

$$\frac{\mathbf{h'v}\,(1) + \mathbf{h'v}\,(-1)}{2}\,, \ldots\ldots, \frac{\mathbf{h'v}\,(\mathbf{n}) + \mathbf{h'v}\,(-\mathbf{n})}{2} \tag{22}$$

to represent the statistics of the renormalized histogram, wherein ¿ 0 is a pre-selected integer. In summary, (n+1) reliable histogram features are selected for classification:

$$\mathbf{hv}\,(0)\,,\ \frac{\mathbf{hv}\,(1)+\mathbf{hv}\,(-1)}{2\mathbf{hv}\,(0)}\ ,\ldots.,\frac{\mathbf{hv}\,(\mathbf{n})+\mathbf{hv}\,(-\mathbf{n})}{2\mathbf{hv}\,(0)} \tag{23}$$

Where hv is the histogram of difference image. The following points summarise the features extraction process:

- First, compute the difference images for four directions (vertical, horizontal, diagonal and anti-diagonal) to obtain *Iv, Ih, Id* and *Ia*.
- Next, again calculate the difference images for each difference image to obtain *Is, t*, where *s, t* ∈ *v, h, d, a*. For instance, the image *Iv,h* here means the horizontal difference image of the vertical difference image *Iv*.
- Then, exclude identical images (for instance, *Iv,h = Ih,v*) and obtain 14 totally different images:
  - *Iv, Ih, Id, Ia*.
  - *Ivv, Ivh, Ivd, Iva, Ihh, Ihd, Iha, Idd, Ida, Iaa*.
- For each of these 14 images, compute the *(n + 1)* features according to Eq. (23). Thus, the total of the features is 14*(n + 1)*.
- For $n\_file = 1 : size(all\_files, 1)$ % for each file in the file list, do the following stat loop:
  $filename = all\_files(n_file).name$;
  $Im = imread(filename)$;
  $ncolors = size(Im, 3)$; % number of colour channels (=1 for indexed images and mono, = 3 for RGB images)
  if $ncolors > 1$
  for i=1:ncolors
  $[features] = hist\_lsb(Im(:, :, i), n)$;

3. **The Proposed System.** The developed detection system contains the following primary stages: creating stego images, detecting hidden files and using classification methods (see Figure 2). The system was developed to deal with colour (RGB) images, and it supports greyscale images as well.

At the first stage of the system, a large number of stego images were created using LSB and F5 steganography; three different image formats were used [12, 13].

The second stage represents the core of the proposed detection system. It includes the implementation processes, extracting 70 histogram features by exploiting the differences of adjacent pixels.

In the third stage, the system was trained and tested using the DA and MLP classification (prediction) methods.

The created database includes 2400 images categorised into grey and colour. Different image formats were used to improve the system's detection capacity. The system achieved dissimilar detection performance with respect to the various image types and steganography methods used. Therefore, training the system with a variety of image formats will increase its detection capacity and evaluate the system using (1) different image formats, (2) grey and colour images, (3) different sizes of hidden images and (4) varying steganography methods.

Image-compression techniques are extensively used in steganography. There are two types of image compression: lossy compression and lossless compression. Therefore, a variety of image formats from these two forms of compression were created for testing and training the proposed detection system [13, 14].

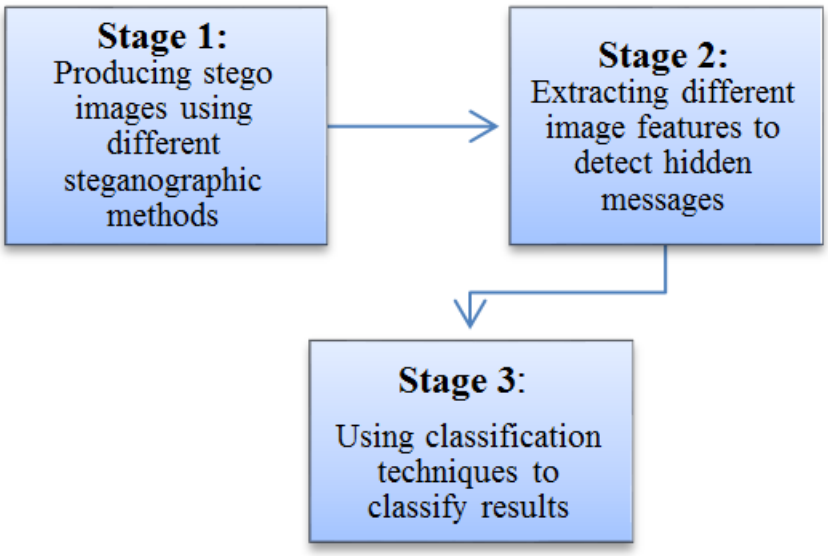

FIGURE 1. Main Stages of the Proposed System

FIGURE 2. Stage 1: Creating Stego Images

Lossless compression is a class of data-compression algorithms that allow the original data to be perfectly reconstructed from the compressed data. BMP was used as an example of the lossless format to train and test the system.

Lossy compression is a class of data-encoding methods that use inexact approximations for representing the encoded content [13, 14]. JPG images were used as an example of the lossy format to train and test the system.

3.1. **Stages of the Proposed System.** Figure 2 represents the process of stage 1: a collection of stego images was created, and the images were then used as applications to train and test the detection system.

Two different steganography methods were used to create the stego images. The first was the LSB method, implemented using S-Tools. This tool was used to create stego images from the BMP images. In addition, a developed LSB code was run using Paython to generate more stego images from the PNG images. Another code for LSB was implemented in order to enable changing of the hiding capacity. In LSB steganography, the pixels of the hidden file are embedded in the least significant bit of the clean image [12, 13].

The F5 algorithm was the second steganography method used; this method implements matrix encoding, which decreases the number of necessary changes [15]. Table 1 shows the image types, steganography methods and number of images used.

TABLE 1. Images Used and Created in the Database

Image Type	Number of Clean Images	Number of Stego Images	Range of Clean Images	Size of Hidden Files	Steganography Method Used
Grey	600	600	599 kb to 1 MB	83 kb, 152 kb (Hiding capacity: 10% & 25%)	LSB
Colour	600	600	599 kb to 1 MB	83 kb, 152 kb (Hiding capacity: 10% & 25%)	LSB
Grey	600	600	100 kb to 1 MB	83 kb, 480 kb (Hiding capacity: 10% & 25%)	F5 Algorithm
Colour	600	600	100 kb to 1 MB	83 kb, 480 (Hiding capacity: 10% & 25%) kb	F5 Algorithm

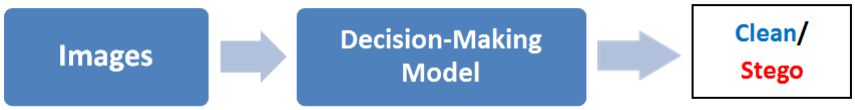

FIGURE 3. Architecture of the Proposed Detection System (Stages 2 and 3)

Stage 2 focused on detecting the hidden data using extracting histogram features from the clean and stego images. Finally, stage 3 utilised the chosen classification methods to distinguish the stego images from the clean ones.

4. **Architecture of The Proposed Detection System.** The proposed detection system was developed to work with colour images with 24-bit depths. The system primarily aims to distinguish stego images from clean ones. The main architecture of the system contains three elements: tested images, the decision-making model and the process of determining whether the tested images are clean or stego.

Figure 3 illustrates the real implementation of the proposed system within the decision-making model. The model includes two parts: training and testing phases. In the training phase, a large number of histogram features was extracted. The system was then trained using the DA and MLP classification methods, implemented using SPSS software.

Images in the created database were divided between the training and testing phases. As shown in figure 4, 70% of the images were used for training, and 30% of the images were used for the testing phase.

5. **Experimental Results.** The employed steganography methods represent an important factor in evaluating the results. Each used method showed dissimilar performance in terms of classifying the stego images from the clean ones. The colour of the images also represents an important consideration because the colours make a difference in data analysis. Additional experiments were conducted to test stego images containing different hidden files sizes, because detection ability relates to hidden file length. Clearly, the less information embedded into the cover image, the smaller the probability of introducing detectable artefacts through the embedding process. Each steganographic method has an upper bound on the maximal safe hidden file length that tells how many bits can be

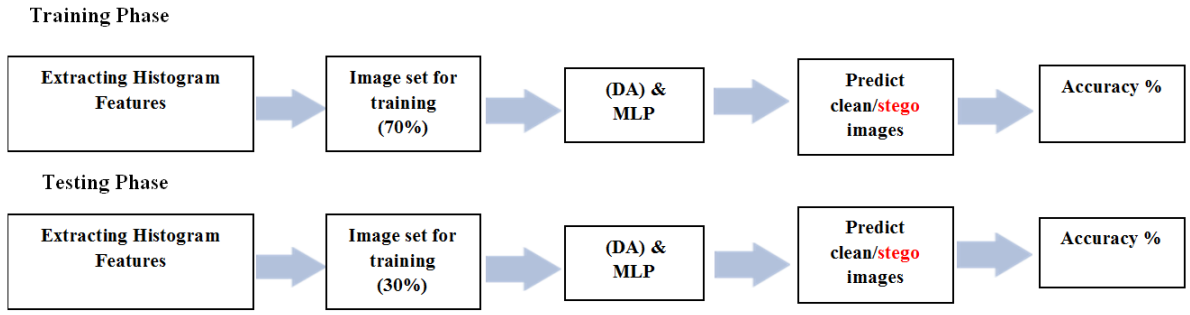

FIGURE 4. Procedures of the Training and Testing Phases

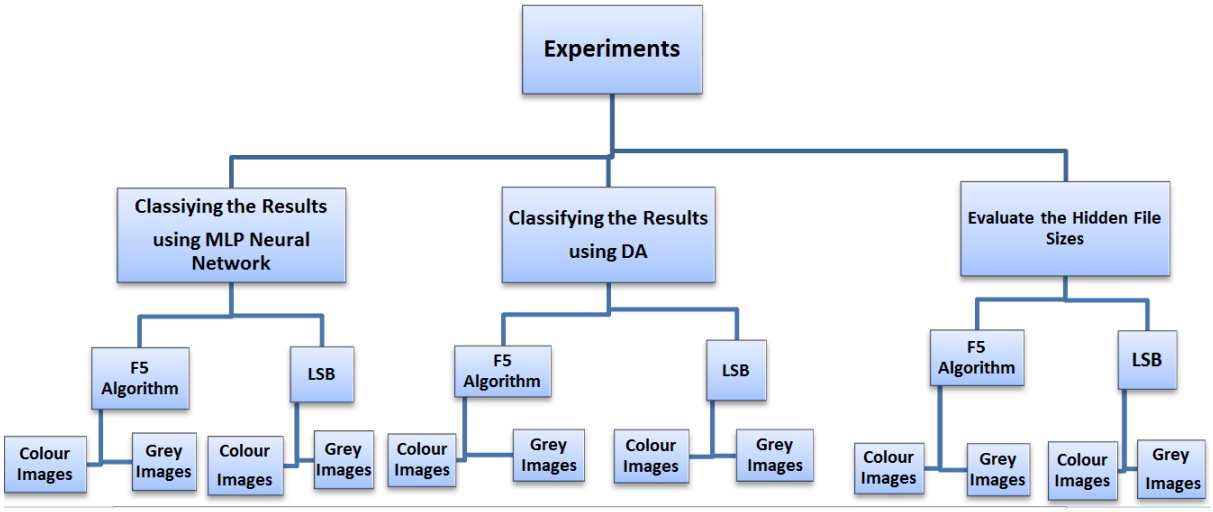

FIGURE 5. Methodology of the Experiments Conducted

safely embedded in a given image without introducing any statistically detectable artefacts. Determining this maximal steganographic capacity is a challenging task even for the simplest methods. Therefore, different capacities and hidden files were used to test the differences in the results and to train the system to deal with a variety of stego images.

Figure 5 shows the methodology of the conducted experiments. The experiments cover different areas in terms of detection reliability and accuracy. A value of 0 was assigned to all clean images, and a value of 1 was given to all stego images. The DA test grouped the images into two cases: clean and stego. During the training phase, all images were given either a 0 or 1 value according to their case.

5.1. **Evaluate Different Hidden File Sizes.** Different hidden-file sizes were tested, as the size of the hidden files plays an important role in terms of detection [17].

During the training phase, all images were given a value of either 0 or 1; a value of 0 was assigned to all clean images, and a value of 1 was given to all stego images. The DA classification test grouped the images into two cases: clean and stego.

As shown in table 2, the images database was divided into four main groups according to the steganography methods used to generate the stego images. Each of these groups has two sets; the first set includes the stego images with the small hidden file embedded into them. The second set includes stego images that have the large hidden file embedded into them. The grey and colour images also were separated to test the detection performance; this experiment uses the DA classifier. Regarding the colour images, each image has three

TABLE 2. Images Sets used to evaluate the differences of the Hidden Files Sizes

LSB Steganography (Grey Images)	**Image Set 1 (Small Hidden Files)**	**Image Set 2 (Large Hidden Files)**
LSB Steganography (Colour Images)	➢ 300 clean images were tested. ➢ 300 stego images were created using S-Tools. ➢ The hidden file size was 83 kb. ➢ The sizes of the clean images were from 599 kb to 1 MB. ➢ All images were grey.	➢ 300 clean images were tested. ➢ 300 stego images were created using S-Tools. ➢ The size of the hidden files was 400 kb in total. ➢ The range of the images was from 599 kb to 1 MB. ➢ All images were grey.
F5 Algorithm (Grey Images)	**Image Set 1 (Small Hidden Files)**	**Image Set 2 (Large Hidden Files)**
F5 Algorithm (Colour Images)	➢ 100 clean images were tested. ➢ 100 stego images were created using the F5 steganography algorithm. ➢ The hidden file size was 83 kb. ➢ The sizes of the clean images ranged from 300 kb to 1 MB.	➢ 100 clean images were tested. ➢ 100 stego images were created using the F5 steganography algorithm. ➢ The size of the hidden files was 480 kb in total. ➢ The images ranged from 300 kb to 1 MB.

TABLE 3. Classification Results of Grey and Colour Images (Small Hidden File)

Steganography Method	Cross-Validated %	Precise Prediction (Clean)	Error Prediction (Clean)	Precise Prediction (Stego)	Error Prediction (Stego)	Number of Features	Total Number of Images
LSB with Grey Images	84 %	86 %	13 %	82 %	17 %	70 Features	600
LSB with Colour Images	**85 %**	86 %	14 %	**85 %**	14 %	70 Features	600

colour channels: red, green and blue RGB. Each feature extracted from a single channel is considered a single feature.

5.2. **Analysis of the Results.** The percentages shown in table 3 represent the effectiveness of the small hidden files in terms of the results while using the LSB steganography with the grey and colour images. There is a noticeable improvement in correct predictions for the stego images, increasing from 82% to 85%. The percentage for correctly predicting the clean images is the same for grey and colour images. In addition, there is a slight difference in the cross-validated percentages, with the percentage increasing from 84% to 85%. The performance of predictions for the clean and stego images for both cases is thus similar so far.

On the other hand, table 4 illustrates the percentages of the large hidden files, displaying the effectiveness of using the LSB steganography with clean and stego images. There is a major difference between the correct percentages in predicting the clean images. The prediction's accuracy decreases from 86% with grey images to 79% with colour images. As such, the prediction's performance is better when predicting grey clean images. In addition, the prediction's accuracy noticeably increases from 79% with grey images to 90% with the colour images. The prediction's performance is higher in terms of predicting the colour stego images. Moreover, there is a slight difference in the cross-validated percentages for both cases; the percentage increased from 83% to 84%.

TABLE 4. Classification Results of the Grey and Colour Images (Large Hidden File)

Steganography Method	Cross-Validated %	Precise Prediction (Clean)	Error Prediction (Clean)	Precise Prediction (Stego)	Error Prediction (Stego)	Number of Features	Total Number of Images
LSB with Grey Images	83 %	**86 %**	13 %	79 %	20 %	70 Features	600
LSB with Colour Images	**84 %**	79 %	21 %	**90 %**	9 %	70 Features	600

TABLE 5. Classification Results of the Grey and Colour Images (Small Hidden File)

Steganography Method	Cross-Validated %	Precise Prediction (Clean)	Error Prediction (Clean)	Precise Prediction (Stego)	Error Prediction (Stego)	Number of Features	Total Number of Images
F5 Algorithm with Grey Images	**70.5 %**	**66 %**	34 %	75 %	25 %	70 Features	200
F5 Algorithm with Colour Images	60.5 %	54 %	46 %	67 %	33 %	70 Features	200

TABLE 6. Classification Results of the Grey and Colour Images (Large Hidden File)

Steganography Method	Cross-Validated %	Precise Prediction (Clean)	Error Prediction (Clean)	Precise Prediction (Stego)	Error Prediction (Stego)	Number of Features	Total Number of Images
F5 Algorithm with Grey Images	78 %	75 %	25 %	81 %	19 %	70 Features	200
F5 Algorithm with Colour Images	**86.5 %**	**84 %**	16 %	**89 %**	11 %	70 Features	200

As shown in table 5, there are many noticeable differences between the results of the grey and colour images. The cross-validated percentage of the colour images is 60.5%; however, it is higher for grey images, with a percentage of 70.5%. In addition, correct predictions for clean and the stego images are higher than for colour at 66% and 75%, respectively. Likewise, the error percentages are less for predicting the clean and stego images in grey than for the colour images.

On the other hand, there are obvious differences between the results shown in table 13 for the grey and colour images. The cross-validated percentage for the colour images is higher than for the grey images at 86.5%, compared to 78% for the grey images. Additionally, the correct percentages for predicting the clean and stego images in the colour case are higher than for the grey case at 75% and 89%, respectively. The error percentage is less for predicting the clean and stego images in colour than in grey.

Furthermore, as shown in table 6, the cross-validated percentage increases greatly when the hidden file becomes larger for both clean and stego images (grey and colour).

6. **Classifying the Clean and Stego Images Using DA.** In this part, the tested grey and colour images were divided into two groups depending on the steganography method used. Table 7 summarises the performance of the system for all cases grey images, colour images and the two steganography methods used. The cross-validated percent for the

TABLE 7. Comparison Results between the LSB Steganography and the F5 Algorithm

	Cross-Validated Of the Grouped Cases (Training) %	Accuracy of Prediction (Ungrouped Cases) (Testing) %	Number of Images Tested
LSB Steganography / Grey Images	78 %	75 %	600
LSB Steganography / Colour Images	**80 %**	74 %	600
F5 Algorithm / Grey Images	77%	**80%**	600
F5 Algorithm / Colour Images	**83.6 %**	76 %	600

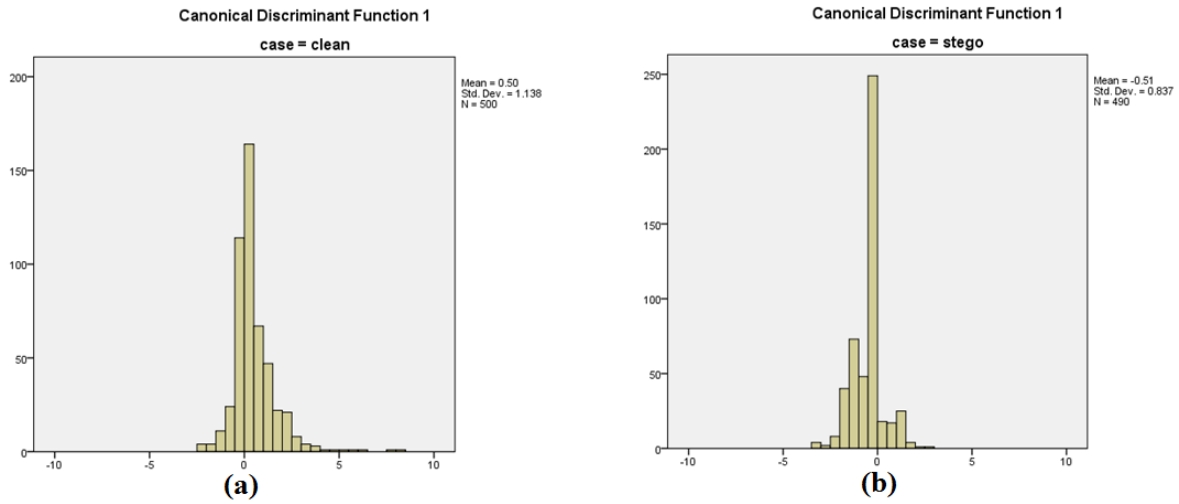

FIGURE 6. Histograms of the Discriminant Function Distribution for Clean Images and Stego Images (Grey Images, LSB)

colour images are higher than the grey ones when using LSB and F5 steganography algorithm. In addition, the highest accuracy is achieved when the F5 algorithm used with the grey images.

Figures 8, 9, 10 and 11 represent the effectiveness of the discriminant function for the four cases. The two histograms illustrate the distribution of the discriminant function scores from the clean and the stego cases. When the distributions do not overlap, this suggests that the function does discriminate well. Figure 8 shows the discriminant function of the clean and stego images is quite overlapping while using the LSB steganography with the grey images. Figure 8 (a) represents the clean images and the lines are shifted more toward the right. However, figure 8 (b) represents the stego images and it shifted more to the left. They are slightly overlapped from 0 to 3.

The discriminant function of the clean and stego images is quite overlapping while using the F5 steganography algorithm with the grey images as shown in figure 9. Figure 9 (a)

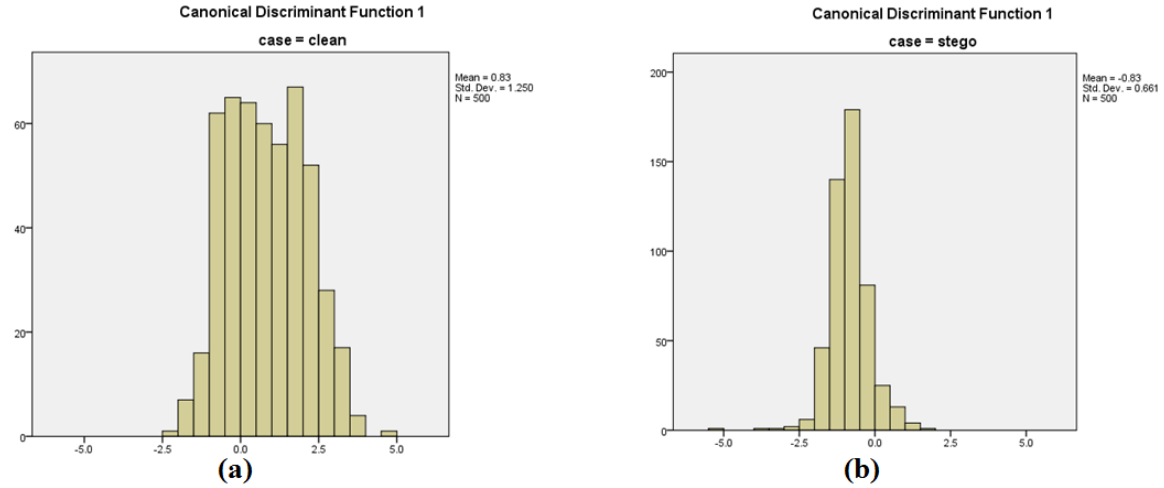

FIGURE 7. Histograms of the Discriminant Function Distribution for Clean and stego Images (Grey Images, F5 Algorithm)

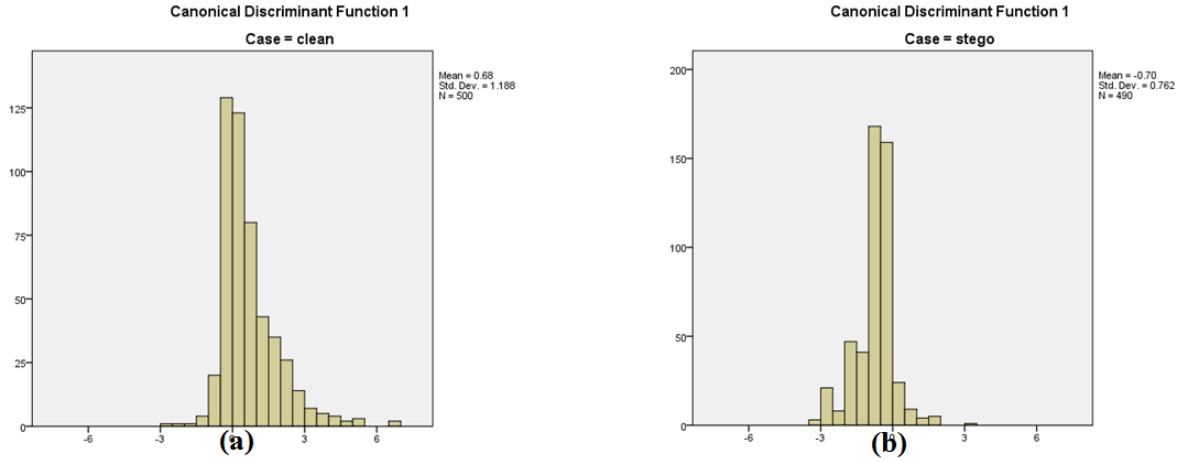

FIGURE 8. Histogram of the Discriminant Function Distribution for Colour Clean and stego Images (Colour Images, LSB)

represent the clean images and the lines are shifted further to the right. However, figure 9 (b) represents the stego images and it shifted more to the left. They are overlapped from -2.5 to 1.5. It seems that the function does not discriminant very well.

Though, the discriminant function of the clean and stego images is slightly overlapping while using the LSB steganography with the colour images as shown in figure 10. Figure 10 (a) represent the clean images and the lines are shifted further to the right. Figure 10 (b) represents the stego images and it shifted more to the left. They are slightly overlapped from 0 to 1.5; this reflects that the function does discriminate well.

Figure 11 shows the discriminant function of the clean and stego images do not overlapping while using the colour images created by F5 steganography algorithm. Figure 11 (a) represent the clean images and the lines are shifted further to the right. Figure 11 (b) represents the stego images and it shifted more to the left. They do not overlap from 0 to 1.5; this reflects that the function does discriminate well.

7. **Classifying the Clean and Stego Images using MLP.** The Multilayer Perceptron Neural Network (MLP) is used to validate the results and the accuracies achieved by the

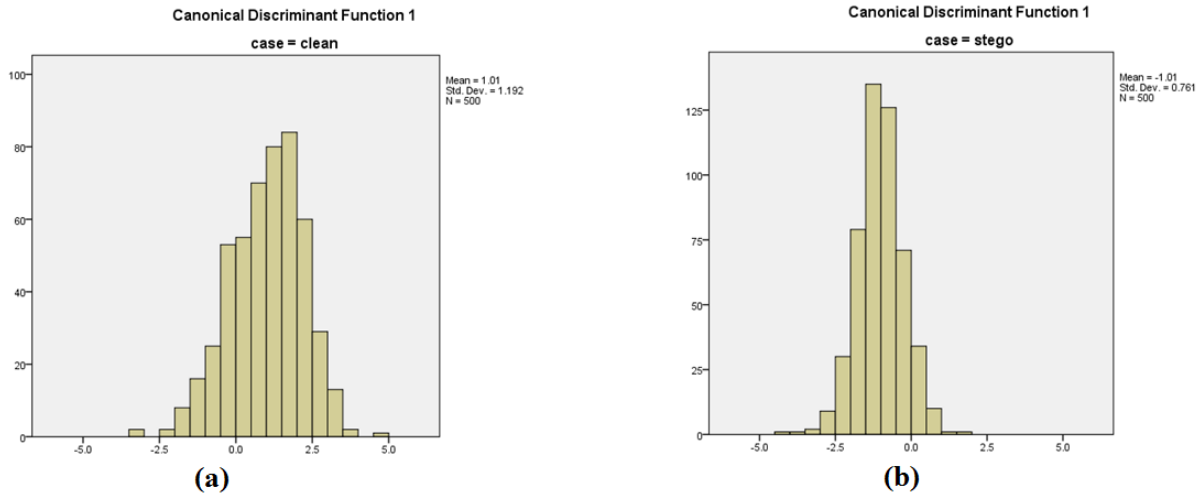

FIGURE 9. Histogram of the Discriminant Function Distribution for Clean and Stego Images (Colour Images, F5 Algorithm)

TABLE 8. The Percent of the Overall Accuracies for the Two Steganography Methods Used

Steganography Method	Images Types	Overall Accuracy Training	Overall Accuracy Testing	Number of Images
LSB	Grey	91%	89%	1200
LSB	Colour	100 %	99%	1200
F5 Algorithm	Grey	83%	81%	1200
F5 Algorithm	Colour	85%	83%	1200

system during the training and the testing phases. The MLP were run 12 times for accuracy; the test automatically divided the images into two sets. It used around 70% of the images for training and 30% for testing. The images databases were separated into two groups according to the steganography methods and to the colour (is it the colour). The first experiments conducted results are shown in table 8. The aim of these is to examine the performance of the detection system with respect to the steganography methods used to create stego images. Each of these steganography methods used different techniques for hiding the files. The experiments prove that the MLP achieved much higher performance in terms of classifying images created by LSB than the F5 steganography algorithm during the training and the testing phases. The MLP achieved outstanding performance in terms of classifying the images created by LSB steganography with the grey and the colour images.

The second experiments conducted are summarised in table 9, the purpose here is to examine the performance of the detection system with respect to colour channels of the images. The colour images are having different features in compare with the grey images. The colour images generally are more complicated according to analysing the three colour channels.

As shown in table 9, the MLP achieved higher accuracy during the training and the testing phases with the colour images more than the grey images.

TABLE 9. Overall Accuracies for the Grey and the Colour Images

Images Types	Overall Accuracy Training	Overall Accuracy Testing	Number of the Images
Grey	80%	76%	2400
Colour	86%	89%	2400

TABLE 10. Comparison of the AUC Values for using MLP with LSB and F5 Steganography

Steganography Method	Images Types	AUC	Features Number
LSB	Grey	.963	70 Features (n = 4)
LSB	Colour	**.998**	70 Features (n = 4)
F5 Algorithm	Grey	.916	70 Features (n = 4)
F5 Algorithm	Colour	.938	70 Features (n = 4)
LSB & F5 Algorithm	Grey	.890	70 Features (n = 4)
LSB & F5 Algorithm	Colour	**.988**	70 Features (n = 4)

However, the overall results are much higher when images are divided into four groups according to the steganography methods used to create the stego images.

7.1. **Area Under Curve.** In binary classification, often, the performance of classifiers is measured using the Area Under the Curve (AUC). Therefore, the area under curve AUC is also used to measure the performance of the network [18].

The AUC of a classification function f expresses the probability that a randomly selected positive example gets a higher score by f than a randomly selected negative example. This measure has proven to be highly useful for evaluating classifiers, especially when class distributions are heavily skewed.

Both accuracy and AUC are complementary measures to evaluate classifier performance, and AUC has the property of being insensitive to class distribution. Larger AUC value means better detection performance. In addition, an AUC value close to 1.0 indicates excellent discrimination, while a value close to 0.5 indicates poor discrimination.

Comparison of the AUC values for the system for images created by the LSB and F5 steganography algorithm is shown in table 10, all values show high discrimination performance up to 0.963 for gay images and 0,988 for Colour images.

Table 11 show the AUC values for the proposed system and for others pervious steganalysis methods. For the two different hiding capacities the proposed system outperforms all tested methods. In summary, it can be concluded that the proposed system is better than some of state-of-the-art algorithms.

8. **Conclusion.** A system for detecting colour and grey images was proposed in this paper. The proposed system targeted RGB images with 24-bit depth. Three different image formats were used to train and test the system. LSB and F5 steganography algorithm

TABLE 11. Comparisons of the AUC Values for using MLP to Classify the LSB and F5 Steganography

Method	AUC Hiding Capacity: 0.25	Images Type
Our method (n=4) (70 features)	**.963**	Grey
Our method (n=4) (70 features)	**.916**	Colour
Kaiwei et. al (n=4) (70 features)	.762	Grey
Gao et al. (50 features)	0.636	Grey
Xuan et al. (39 features)	0.626	Grey
Dong et al. (36 features)	0.618	Grey
Gul and Kurugollu (25 features)	0.68	Grey

were used to create the stego images for testing and evaluating the system. The created images database contained a large number of clean and stego images. Two groups of images were used colour and grey images.

The proposed detection system was evaluated according to four dimensions. First, the system was designed to implement blind steganalysis; thus, it supports different image formats; lossless and lossy image compressions. Second, the chosen images grouped according to whether colour or grey. Third, Different hidden-file sizes were used to evaluate the system's detection capacity. Several tests were performed to determine the effectiveness of different hidden-file sizes in terms of detection ability. The results show slight effect while changing the sizes of the hidden file for the grey and colour stego images created by LSB.

However, there are clear effect while changing the sizes of the hidden file for the grey and colour stego images created by F5 algorithm. The detection system achieved dissimilar performance in case of detecting the stego images created by the two different steganography methods; LSB and F5 algorithm.

Training and testing results show that the MLP classifier achieved greater accuracy than the DA classifier. The DA achieved high accuracy up to 83% cross-validation of the grey stego images created by F5 algorithm. In addition, MLP performed well for images created by LSB rather than F5 algorithms. It performed outstanding accuracy that reached 99% with the colour stego images created by LSB. Also, it performed high accuracy that reached 89% with grey stego images created by LSB. The MLP achieved similar performance to the DA in case of grey and colour stego images created by F5 algorithm. In the future work adding statistical features to the histogram features would be useful in order to make the proposed system more robust. This could potentially improve the detection performance in general.

Different types of images and steganography methods were used to test and train the proposed system. Therefore, adding different types of features to detection systems might help them detect a wider range of images and more types of steganography.

REFERENCES

[1] G. Huayong G, Mingshenge H, Qiana W (2011) ,Steganography And Steganalysis Based On Digital Image, *4th International Congress on Image and Signal Processing*, 2011.

[2] N. F. Johnson, S. Jajodia, Steganalysis: The investigation of hidden information, *IEEE Information Technology Conference*, 1998.

[3] X. Luo X, D. Wang, P. Wang P, F. Liu, A review on blind detection for image steganography, *Signal Processing*, vol. 88, no. 9, pp. 2138-2157, 2008

[4] J. Fridrich, and M. Long, Steganalysis of LSB Encoding in Color Images, *IEEE International Conference on Multimedia and Expo*, vol. 3, pp. 1279-1282, 2000.

[5] A. Hmood, B. Zaidan, and et al, An Overview on Hiding Information Technique in Images, *Journal of Applied Sciences*, vol. 10, no. 18, 2010.

[6] J. J. Harmsen and W. A. Pearlman, Steganalysis of additive noise modelable information hiding, *Security and Watermarking of Multimedia Contents V*, vol. 5020, pp. 131142, 2003.

[7] K. W. Cai, X. L. Li, T. Y. Zeng, B. Yang , X. Q. Lu, Reliable Histogram Features For Detecting LSB Matching, *17th IEEE International Conference on Image Processing (ICIP)*, 2010.

[8] G. McLachlan, Discriminant analysis and statistical pattern recognition, *John Wiley and Sons*, vol. 544, 2004.

[9] J. Pallant, SPSS survival manual: A step by step guide to data analysis using SPSS, *McGraw-Hill International*, 2010.

[10] O. Yen-Jen, Shien-Ching, and et al, Data Classification With Radial Basis Function Networks Based on a Novel Kernel Density Estimation Algorithm, *IEEE Transaction On Neural Networks*, vol. 16, no. 1, 2005.

[11] J. Fridrich, M. Goljan, and D. Hogea, Steganalysis of JPEG images: Breaking the F5 algorithm, *Information Hiding, Springer Berlin Heidelberg*, 2003.

[12] A. Ahd, A. Saad, and F. John, Creating Stego-Images through Hiding Single And Multiple Data Using Different Steganographic Tools.

[13] A. Ahd, A. Saad, F. John, J. Shuttelworth, Developing a detection system for grey and colour stego images, *International Journal of Modeling and Optimization*,vol. 3, no. 5, pp. 458-461, 2013.

[14] D. L. CurrieIII, C. E. Irvine, Surmounting The Effects Of Lossy Compression On Steganography, *19th National Information System Security Conference*, 1996.

[15] A. Westfeld, F5a steganographic algorithm, Ira S. Moskowitz, Information hiding, USA, *Springer Berlin Heidelberg*, pp. 289-302, 2001.

[16] J. Miano, Compressed image file formats: Jpeg, png, gif, xbm, bmp. Addison-Wesley Professional, 1999.

[17] V. L. Reddy, A. Subramanyam, and P. C. Reddy, Implementation of LSB steganography and its evaluation for various file formats, *Int. J. Advanced Networking and Applications*, vol. 2, no. 5, pp. 868-872, 2011.

[18] Fawcett, Tom, ROC graphs: Notes and practical considerations for researchers, *Machine learning*, vol. 31, no. 1, pp. 1-38, 2004.