Optics Communications 285 (2012) 4252–4261

Contents lists available at SciVerse ScienceDirect

Optics Communications

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

A pattern recognition system for JPEG steganography detection

C.L. Philip Chen [a], Mei-Ching Chen [b], Sos Agaian [c], Yicong Zhou [a,*], Anuradha Roy [d], Benjamin M. Rodriguez [e]

[a] Department of Computer and Information Science, University of Macau, Macau, People's Republic of China
[b] Montgomery College, Rockville, MD 20850, USA
[c] Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249, USA
[d] Department of Management Science and Statistics, University of Texas at San Antonio, San Antonio, TX 78249, USA
[e] Johns Hopkins University Applied Physics Laboratory, Laurel, MD 20723, USA

ARTICLE INFO

Article history:
Received 11 March 2012
Received in revised form
18 June 2012
Accepted 19 June 2012
Available online 30 June 2012

Keywords:
Feature generation
Feature ranking and selection
Pattern classification
Pattern recognition system
Steganography detection

ABSTRACT

This paper builds up a pattern recognition system to detect anomalies in JPEG images, especially steganographic content. The system consists of feature generation, feature ranking and selection, feature extraction, and pattern classification. These processes tend to capture image characteristics, reduce the problem dimensionality, eliminate the noise inferences between features, and further improve classification accuracies on clean and steganography JPEG images. Based on the discussion and analysis of six popular JPEG steganography methods, the entire recognition system results in higher classification accuracies between clean and steganography classes compared to merely using individual feature subset for JPEG steganography detection. The strength of feature combination and preprocessing has been integrated even when a small amount of information is embedded. The work demonstrated in this paper is extensible and can be improved by integrating various new and current techniques.

1. Introduction

Due to an enormous need on digital data communication, information transmission, and storage, information security in digital files is a serious issue [1]. Encryption has been developed to keep the confidentiality of messages by making them unreadable [2]. Digital steganography has been one of the main tools for protecting data. A cover signal chosen to be a cover media imperceptibly hides secret information with the use of steganography methods [3,4]. The secret message is known as steganographic content, as well as stego in short. Signals containing the concealed information are stored and/or transmitted through public channels preventing hidden data from being observed. On the other hand, Cyber crime may use steganography as a tool to conceal potential evidence inside of another file, making evidence virtually unobtainable [1,5]. This has been an issue since evidences first indicated steganography is used for covert communication [6–8]. Steganalysis against steganography tends to discover whether a given signal potentially contains a secret, determines possible steganography method(s) utilized, and/or further extracts pertinent data [9].

Among all digital files, numerous devices generate JPEG images due to the capability of compression and compatibility. A large number of JPEG steganography methods are also available online for free usage [10]. A diverse of JPEG steganography methods applying different embedding techniques results in various changes to natural image characteristics [11]. This has spawned significant research in the area of JPEG image steganalysis. For image steganography detection, prior arts have researched and developed image features for distinguishing clean and stego JPEG images [12–17]. However, as more features being generated, the dimensionality of problem space increases. Although some have tended to solve this problem within the stage of feature generation, the resulting feature information may contain inferences or correlation within, which may degrade the classification performance.

A typical pattern recognition (PR) system includes several components: feature generation, feature ranking and selection, feature extraction, as well as classification [18–20]. There have been many pattern recognition applications developed in various fields, such as bioinformatics and signal detection/prediction systems. This paper constructs a pattern recognition system for JPEG steganography detection. The first part of the recognition system is to develop a set of image features capable of distinguishing clean images from stego images. Utilizing a combination of existing features is also a way to capture the strength from various techniques. For clustering steganography images of different embedding methods with clean

[*] Corresponding author. Tel.: +853 83978458; fax: +853 28838314.
E-mail address: yicongzhou@umac.mo (Y. Zhou).

0030-4018/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.optcom.2012.06.049

images, a subset of features may be selected differently in order to have better separability. The criterion for feature selection based on a separability ranking measure. The selected features are further extracted/transformed into a set of principle components which are mutually exclusive. Then a neural network classifier is utilized for classification.

Clean JPEG images along with six JPEG stego image sets are analyzed for demonstration, including F5 [21], JPEG-JSteg v1 [22], JPEG-JSteg v4 [23], Model-based v1.2 [24], Outguess v0.2 [25], and Steghide v0.5.1 [26]. The results show that the proposed system improves the separability between clean and each of the stego image set with three embedding file sizes and three stego methods. With the entire feature combination and preprocessing system, the procedures utilizing current feature generation methods not only reduce the cost of developing new algorithms but also increase the classification accuracies (CA) with smaller problem dimensionality. The system is extensible and can be further improved with the integration of newly developed and currently existing techniques in each pattern recognition stage.

The paper is organized as follows. An overview of JPEG steganography and detection is given in Section 2. Section 3 discusses and analyzes six popular JPEG steganography methods: F5, JPEG-JSteg v1, JPEG-JSteg v4, Model-based v1.2, Outguess v0.2, and Steghide v0.5.1. Section 4 presents the pattern recognition system for JPEG steganography detection. Section 5 describes, demonstrates, and compares the experimental results of the presented system with three benchmark methods [13,15,16], ensuing the conclusion and discussion of possible future work in Section 6.

2. Related work

This section briefly describes steganography on JPEG images and then provides a short survey on existing techniques used for JPEG image steganography detection.

2.1. JPEG steganography

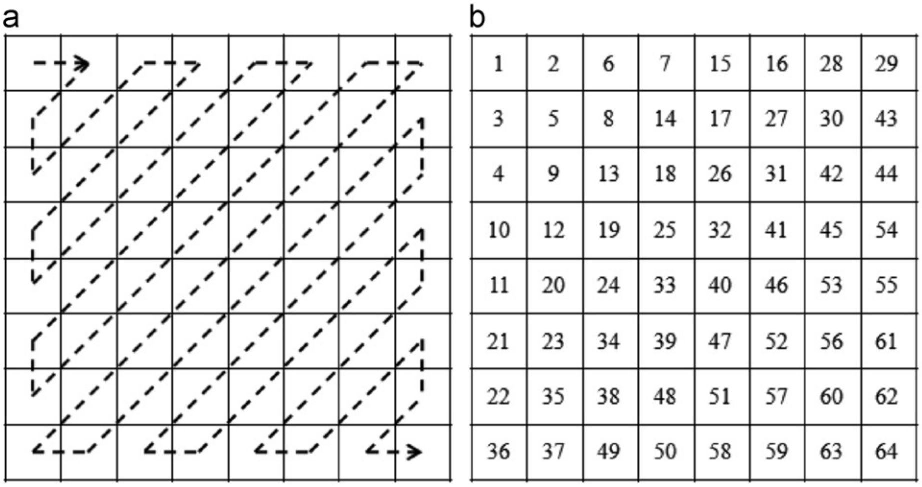
JPEG, developed in 1992 by the Joint Photographic Experts Group, is a standard format for lossy compression based on discrete cosine transforms (DCT) [27]. The JPEG image compression format inherits the characteristics from DCT in 8 × 8 image blocks. One of the characteristics of applying DCTs yields the resulting coefficients in frequency order from low to high as a zigzag scan, as shown in Fig. 1 with an 8 × 8 block. The JPEG coefficient locations in Fig. 1(b) are specified as the arrow flows in Fig. 1(a). The coefficient at location 1 in Fig. 1(b) is the DC coefficient, while the others are called AC coefficients. The energy after the transforms concentrates on lower frequency coefficients, resulting in larger coefficient values comparing to those located in the higher frequency area.

JPEG steganography usually applies various embedding techniques on JPEG coefficients. The embedding algorithms for hiding secret alter the coefficients in a way that an image is imperceptibly changed. JPEG-JSteg, the first stego method on JPEG images, was introduced by Upham [22]. The stego message is embedded in the least significant bits of JPEG coefficients sequentially or randomly [23]. This results in changing the characteristics of histograms (or the first order statistics) of coefficient values and leads to a simple observation of alteration.

More embedding techniques intend to maintain the natural histogram of the coefficients. F5 stego method [21] was developed as a challenge to the steganalysis community. It makes use of matrix embedding techniques. In the F5 algorithm, k message bits are XORed with n hashed coefficients to determine if absolute value of coefficients should be decremented or keep unchanged. Outguess stego method [25] modifies the LSB of the DCT coefficients by statistically checking the original image. It manipulates nearby DCT blocks to maintain a uniform-like distribution of the coefficient histogram. Based on statistical modeling and information theory, Model-based stego method [24] tends to avoid first order statistical attacks while achieving a higher stego message capacity than previous methods. Steghide [26] also has the aim of resisting first order statistical tests in addition to the compression and encryption of stego data. Note that Outguess v0.2 and Steghide v0.5.1 do not embed stego data if a cover does not have enough capacity while other methods hide stego message to its maximum capability of a cover. Analysis of these stego methods will be shown in Section 3.

2.2. JPEG steganography detection

Steganalysis tools, based on functionalities, are categorized into targeted steganalysis and blind steganalysis [11,28,29]. The former type focuses on a certain steganography tool while the latter one is independent to the methods being used. To solve the image steganalysis problem, there are mainly two approaches. One is visual analysis which examines the images perceptually. The other approach, statistical analysis, analyzes the images by applying statistical tests in order to find anomalies within images,



1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

Fig. 1. An 8 × 8 JPEG coefficient block: (a) coefficient zigzagging; and (b) coefficient locations.

such as chi-square tests, blockiness analysis, pixel comparisons, etc [13,30]. In addition, commercial tools have been developed for steganalysis purpose, such as Stego Suite [31], StegAlyzer [32], etc.

For JPEG steganography detection, the characteristics of JPEG coefficients described in the previous section are important. Due to a large amount of sources generating JPEG images and online freeware generating stego files with JPEG images, it is necessary to properly detect various JPEG stego methods. In current JPEG steganalysis methods [13–17,33], generating image features plays an important role for determining if an image contains stego messages or not. Classifiers such as support vector machines are then applied to the features to make detection decision. The presence of noises in features increases the difficulty of a detection system for detecting stego messages embedded by various methods [34]. To address this problem while achieving higher classification accuracies, a specific pattern recognition system for steganalysis will be proposed by combining various feature sets including both existing and new ones. This ensures the system making full use of all developed features. Moreover, the proposed system has the ability to update itself by including newly generated features.

3. Analysis on JPEG steganography methods

For a clean JPEG image, the histogram of JPEG AC coefficient values approximately appear to have a Laplacian distribution with zero mean [35,36]. Fig. 2(a) is an averaged histogram on JPEG AC coefficient values from a thousand clean JPEG images randomly selected from [37]. Fig. 2(b) focuses on coefficient values from −15 to 15. Apparently, the average number of the coefficient value 0 is much higher than other coefficient values. This property leads to JPEG compression.

Fig. 3 shows averaged histograms of AC coefficient values from a thousand randomly selected JPEG images from [37] incorporated with various JPEG stego tools with a stego message file of size 1.04 KB in a zoomed-in scale as the same as in Fig. 2(b). As

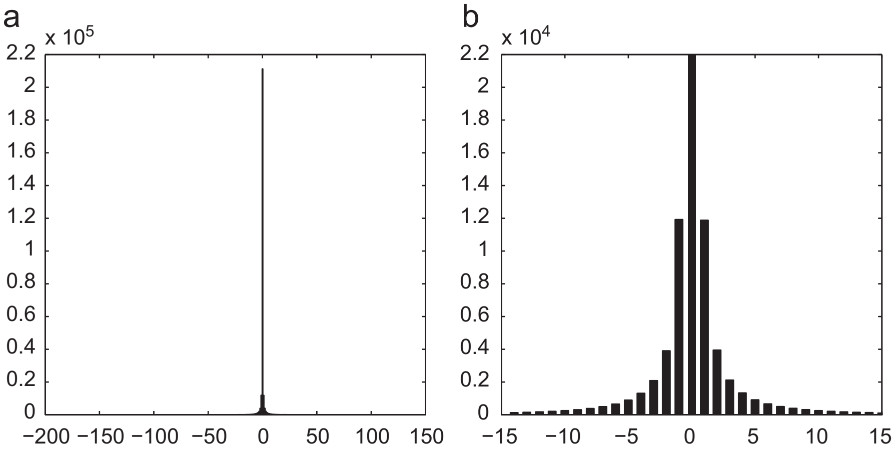


Fig. 2. An averaged histogram of coefficient values from 1000 JPEG images: (a) averaged histogram and (b) zoomed-in histogram.

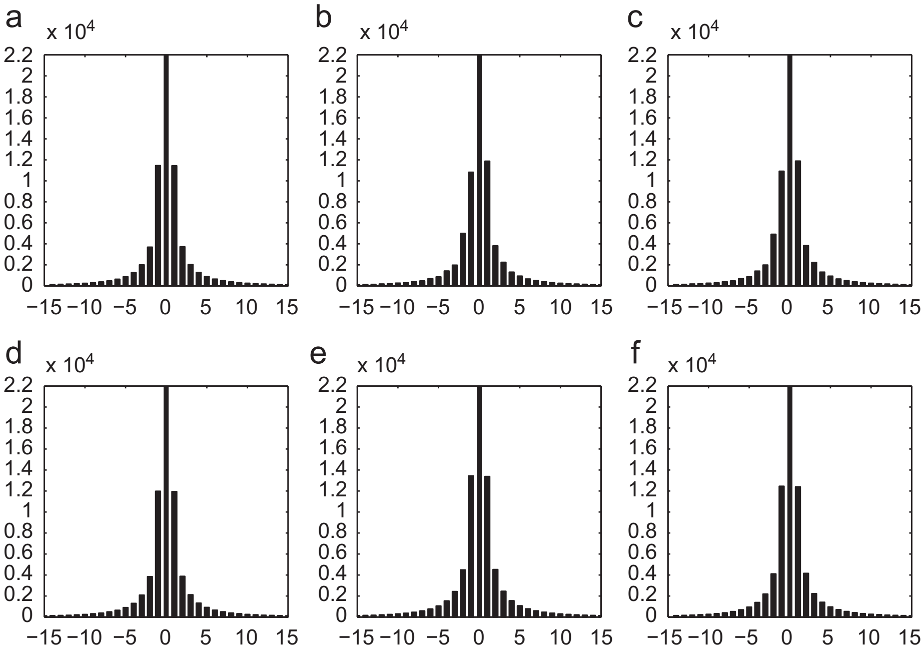


Fig. 3. Zoomed-in averaged histograms of coefficient values from 1000 JPEG images with various steganography tools: (a) F5; (b) JPEG-JSteg v1; (c) JPEG-JSteg v4; (d) Model-based v1.2; (e) Outguess v0.2; and (f) Steghide v0.5.1.

can be seen, the changes made on coefficient value histograms of both JPEG-JSteg v1 and JPEG-JSteg v4 are obvious. JPEG stego methods as F5, Model-based v1.2, Outguess v0.2, and Steghide v0.5.1 shown in Fig. 3(a), (d), (e), and (f) intend to manipulate the coefficients in a way that minimizes the impact of the first-order statistics of images. This increases the difficulty of statics analysis in steganalysis. In each of the following experimental demonstration, seven stego message files with different sizes are utilized, including size approximations of 0.1 KB, 0.2 KB, 0.3 KB, 0.4 KB, 0.5 KB, 1.0 KB, and 1.5 KB.

3.1. Root mean squared error (RMSE) measurements

Eq. (1) is the formula of root mean squared error in two dimensions. Given two matrices **X** and **Y** of the same size $M \times N$, the RMSE between the two is calculated as

$$\mathrm{RMSE}(\mathbf{X},\mathbf{Y}) = \sqrt{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(\mathbf{X}[m,n]-\mathbf{Y}[m,n])^2}, \tag{1}$$

where $m = 1, 2, \ldots, M$ and $n = 1, 2, \ldots, N$.

3.1.1. RMSE in spatial domain

This analysis computes the averaged RMSE between 1000 randomly selected clean and stego images of the six stego methods described in the previous section along with seven stego message file sizes in the spatial domain.

As can be seen in Fig. 4(a), JPEG-JSteg v4 is an exception among all, resulting in higher error measurements. Fig. 4(b) zooms in the five stego methods tangled at the bottom in Fig. 4(a). Note that Outguess v0.2 and Steghide v0.5.1 do not embed stego data if a cover does not have enough capacity. In other words, not all 1000 images in random selection are feasible for analysis using Outguess v0.2 and Steghide v0.5.1. This results in [38] by considering stego message as noises.

3.1.2. RMSE in JPEG domain

Fig. 5 shows the analysis of the averaged RMSE between 1000 randomly selected clean and stego images of the six stego methods described in the previous section along with seven stego message file sizes in the JPEG domain.

Note that Outguess v0.2 and Steghide v0.5.1 have lower RMSE when stego message file size is larger than 1 KB since there are more cover images in the 1000 images not feasible for embedding due to the capacity of each image. For stego message file size less than 0.5 KB, F5 has the lowest RMSE while JPEG-JSteg v1 has the highest RMSE most of the times.

3.2. Number of changes on JPEG coefficient locations

This analysis demonstrates how coefficient locations affect the number of changes on JPEG coefficient values between clean and stego images. Referring to Fig. 1, Fig. 6 shows the averaged number of changes on each coefficient location denoted from 1 to 64 with the six stego methods along with 1000 randomly selected images. Please note that all the six images in Fig. 6 have the same scale.

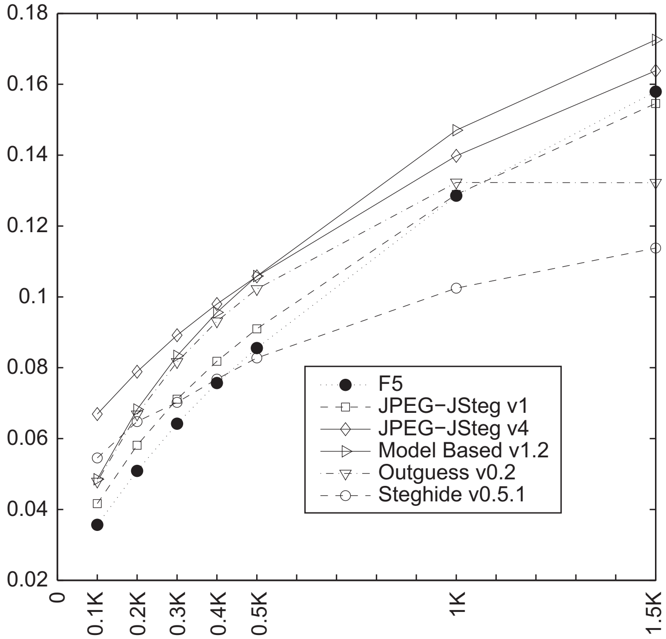


Fig. 5. An averaged RMSE of JPEG coefficient values between 1000 clean and stego JPEG images.

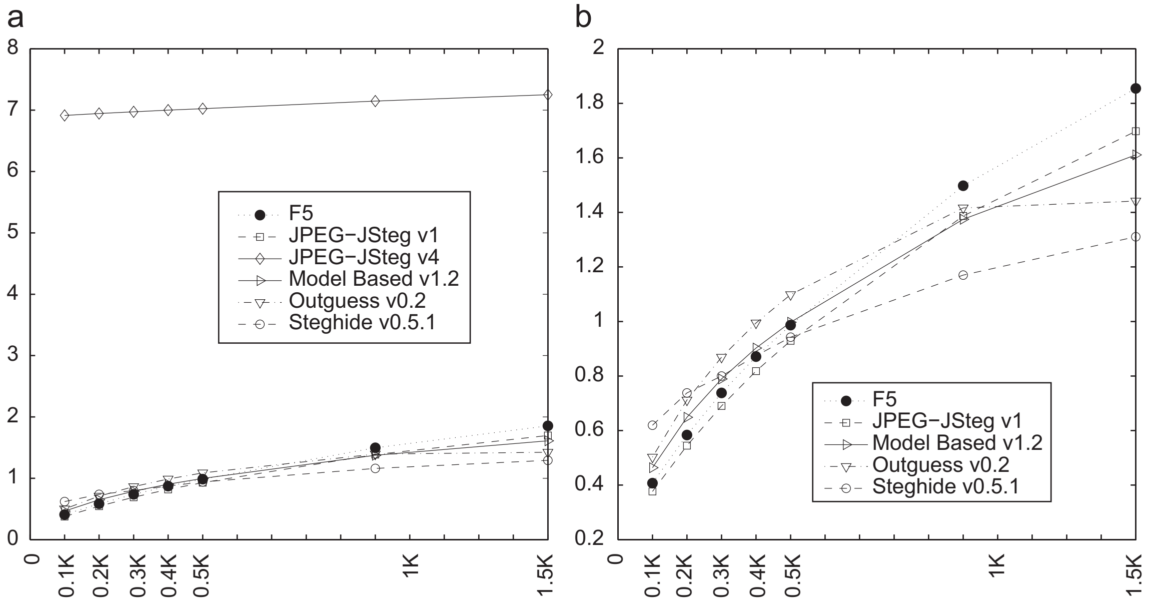


Fig. 4. An averaged RMSE of pixel values between 1000 clean and stego JPEG images: (a) averaged RMSE and (b) zoomed-in averaged RMSE excluding JPEG-JSteg v4.

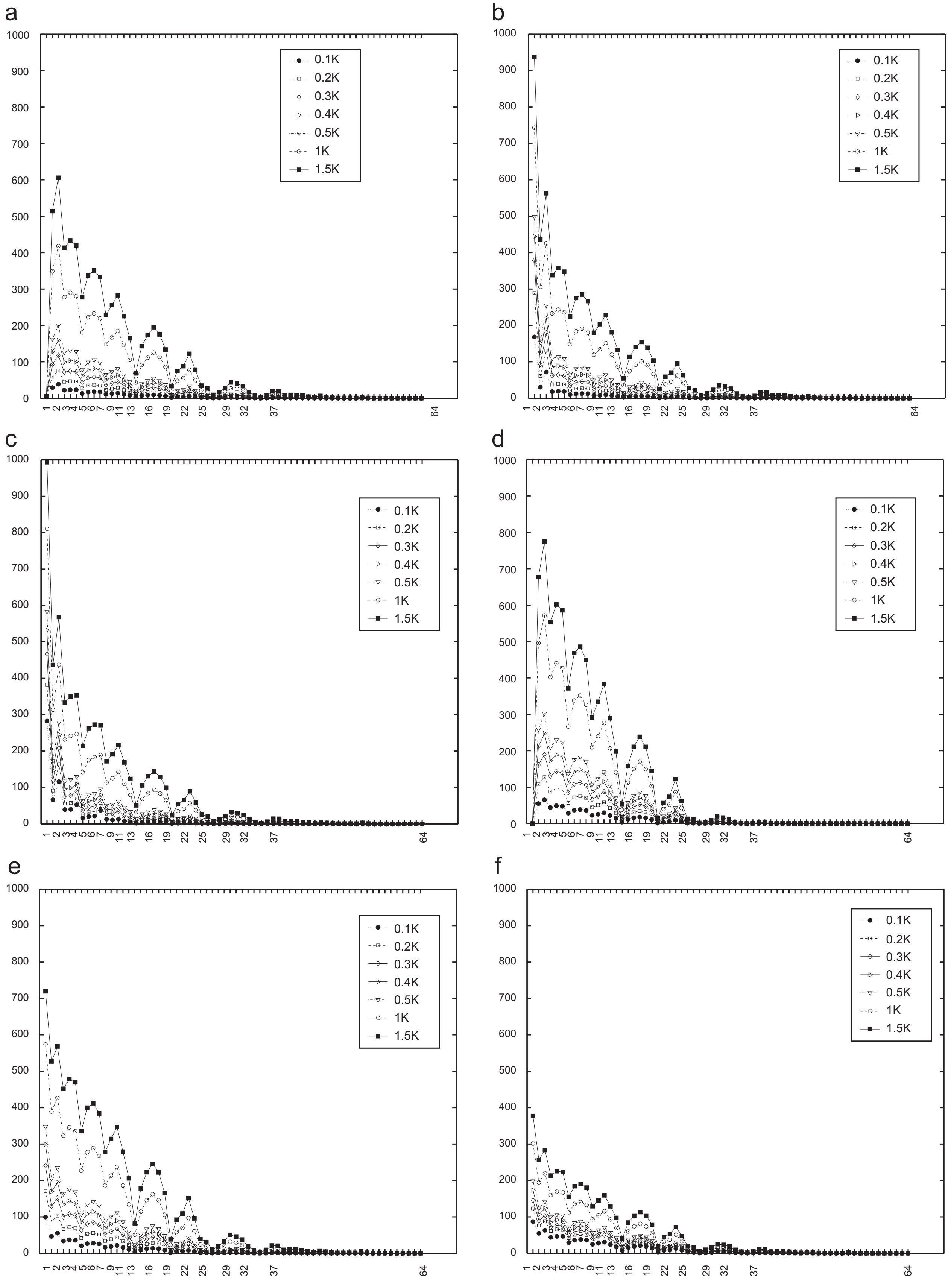


Fig. 6. An averaged number of changes on coefficient locations between 1000 clean and stego JPEG images: (a) F5; (b) JPEG-JSteg v1; (c) JPEG-JSteg v4; (d) Model-based v1.2; (e) Outguess v0.2; and (f) Steghide v0.5.1.

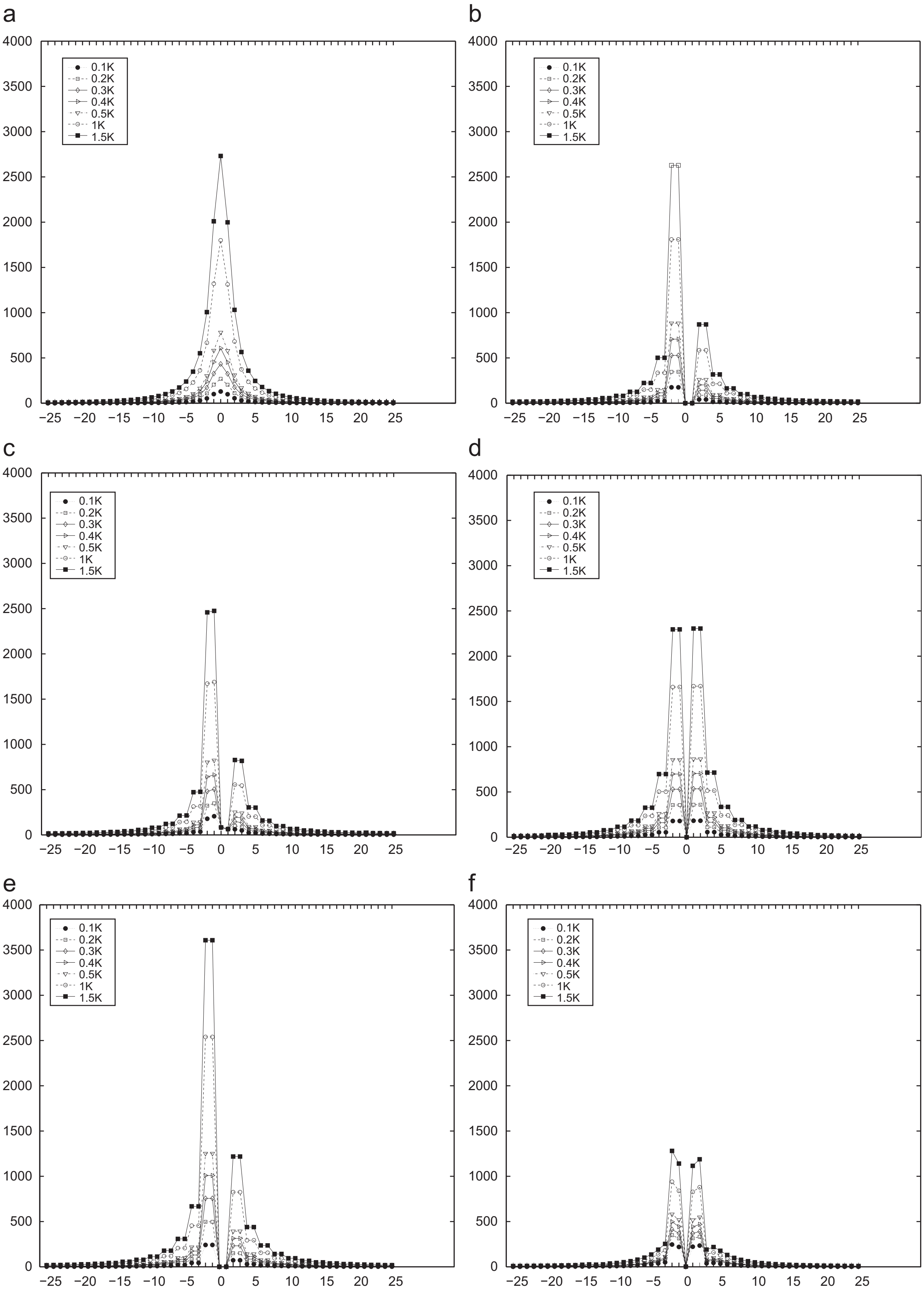


Fig. 7. An averaged number of changes on clean coefficient values from −25 to 25 between 1000 clean and stego JPEG images: (a) F5; (b) JPEG-JSteg v1; (c) JPEG-JSteg v4; (d) Model-based v1.2; (e) Outguess v0.2; and (f) Steghide v0.5.1.

As can be seen in Fig. 6(a) and (d), F5 and Model-based v1.2 does not touch DC coefficients located at coefficient location 1. With a feasible set of 1000 images, Steghide v0.5.1 has the minimum effect on manipulating the coefficients at each location. In addition, except for the DC coefficients and the first few AC coefficients, the peaks and the valleys are located at the same location indices, such as 9, 13, 19, 25 are peaks and 7, 11, 16, 22, 29 are valleys, for every stego method. Note that the valleys are the location indices close to the edges of the coefficient block in Fig. 1(b) and the indices of the peaks have more concentration to the diagonal locations.

3.3. Number of changes on JPEG coefficient values

With 1000 clean and stego JPEG images, this analysis illustrates the averaged number of changes on clean JPEG coefficient values between -25 and 25. See Fig. 7. For instance, for a clean coefficient value -20 in a certain image, how many of them will be changed in stego image due to the stego method used. Among the six stego methods in Fig. 7, it can be seen that only F5 and JPEG-JSteg v4 alter the coefficient value 0. Especially in F5, the number of value 0 increases after the embedding process as shown in Fig. 7(a). JPEG-JSteg v1 and Outguess v0.2 do not touch coefficient value 1 as well. These methods change coefficients within the smaller absolute values, such as 2, 3, 4. This results in [15] considering the absolute threshold value as 4 in both [15] and [33].

4. Steganography detection system

A steganography detection system developed in [34] has shown that feature generation along with preprocessing for steganography detection is vital. This section designs a pattern recognition system with a set of features combining three feature subsets, feature ranking and selection, as well as classification for JPEG steganography detection.

4.1. Feature generation

For image steganalysis problem, in order to find out if there are anomalies in an image, one way is to approximate the characteristics of an image, such as image pixel values or coefficient values in transform domains. Fig. 8 shows a generic procedure of the technique.

Prior arts have developed assorted feature sets, such as [13,15,38]. Each feature set provides different separable strength on clean and stego images from various JPEG stego methods. It, however, only offers a detection system the limited ability to detect a small set of image steganography algorithms. To overcome this problem, the feature generation stage in the presented system combines three feature subsets listed as follows, resulting in a 370 dimensional feature vector for each image.

1. Fridrich generates a set of features 23 features based on image calibration [13].
2. Shi et al. develops a set of 324 features viewing the differences in the JPEG 2-D array with Markov random process [15].
3. Chen et al. exploits the concept of considering stego message as noises, estimating image pixel values by alpha-trimmed mean filtering techniques along with statistical measures as in [13]. A total of 23 features is generated [38].

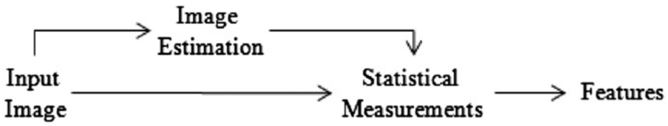


Fig. 8. A generic image estimation feature generation method.

This combined feature set offers the presented system the stronger capability of detecting a larger number of JPEG embedding methods that embed within the header, DCT coefficients and the footer of JPEG images. Moreover, this feature set can be expanded by incorporating current and/or new image feature subsets.

4.2. Feature multivariate analysis

4.2.1. Feature ranking and selection

Among raw features, some of them may have the ability to separate the input data into different classes, others may be not. Additionally, a specific feature may show different separability strength for various categories. The purpose of the feature selection is to make sure the data to be properly classified [39].

The equal covariance discriminant ratio (ECDR), also known as the Fisher's discriminant ratio, is an efficient approach for dimensionality reduction of the data space in statistical pattern recognition [40]. As an improved version of the ECDR, a generalized equal covariance discriminant ratio (GECDR) is used as the ranking tool in the proposed detection system. It can efficiently quantify the reparability of individual features utilizing trimming outliers [34]. The features are ranked based on the class separability by measuring the class discrimination of the individual features. For a one-dimensional, two-class problem, the GECDR is defined by

$$FDR_{\alpha_{ij}} = \frac{(\mu_{\alpha_i} - \mu_{\alpha_j})^2}{\sigma^2_{\alpha_i} + \sigma^2_{\alpha_j}}, \tag{2}$$

where μ_{α_i} and μ_{α_j} are the alpha-trimmed means [41], and $\sigma^2_{\alpha_i}$ and $\sigma^2_{\alpha_j}$ are the deviations corresponding to the individual feature under investigation for the two classes.

Using the GECDR in Eq. (2) leads to an increase in differences between the means while less variance within two classes respectively, obtaining a higher ranking value. In other words, the high-ranked features have higher possibilities to distinguish between classes. Based on the ranking, a certain number of top-ranked features are selected in this preprocessing stage. The meaning is that these selected features are expected to have higher class separability while the features not selected have little ability to distinguish between classes.

4.2.2. Feature extraction

The goal of feature extraction is to avoid information redundancies. This stage consists of mapping the current space onto a new space which is more suitable for the given task. The method exploited in the system is known as principle component analysis (PCA) [42], also known as the Karhunen–Loeve Transform (KLT).

Using PCA for feature extraction is to represent a new space in a way to extract mutually uncorrelated features from the current space. To achieve the minimum MSE, the eigenvalues and eigenvectors of the correlation matrix derived from the selected raw features are first calculated. The MSE is minimized if the eigenvectors corresponding to a certain number of largest eigenvalues are chosen. Therefore, in this stage, the extracted features are to separate the class potentially since the principle components are uncorrelated but keep all of the information from the selected raw features.

4.3. Classification and validation

4.3.1. Data standardization

Standardization allows for the centroid of the data to be moved to the origin and stretched or compressed according to the individual variances. Without preprocessing, a feature with a larger value can dominate the input effect and influence the model accuracy of neural network classifiers, fuzzy learning classifiers, or others. This, however, does not necessarily reflect the individual features respective significance in the design of the classifier model. In this stage, an alpha-trimmed mean normalization technique is used. Expanding from zero-mean normalization, the mean and standard deviation are calculated with alpha-trimmed mean. Using alpha-trimmed mean with the standardization allows the removal of outliers without additional outlier processing [34].

4.3.2. Neural network classifier

A neural network classifier is utilized here for nonlinear classification [43,44], as shown in Fig. 9.

The input of the trained neural network classifier are the standardized feature vector $\hat{\mathbf{f}}$ of length q. The output y is the classification result indicating which class the input $\hat{\mathbf{f}}$ belongs to. For each node in the first layer network, Gaussian kernels ϕ_i with centers and spreads are applied. A weighted linear mapping in the second layer makes the classification decision with a threshold b. The model equation is defined in Eq. (3).

$$y = b + \sum_{i=1}^{r} \omega_i \phi_i(\hat{\mathbf{f}}) \tag{3}$$

4.3.3. Cross validation

A k-fold cross validation, sometimes called rotation estimation, is applied in this stage. The technique separates the entire data set into k mutually exclusive subsets (folds) [45]. The folds are of approximately equal size. The inputs are trained on the selected training data and tested on the test data selection. The cross validation estimation of accuracy is the overall number of correct classifications divided by the number of instances in the data set. The accuracy estimate is the average accuracy for k mutually exclusive subsets.

Kohavi has shown through experimental results on artificial data and theoretical results in restricted settings, that selecting a good classifier from a set of classification model, 10-fold cross validation may be better than the more expensive leave-one-out cross validation [45]. In this paper, this procedure will be adopted and carried out for all experimental results.

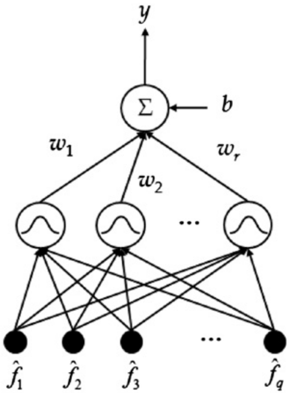


Fig. 9. A neural network model with Gaussian kernels.

5. Experimental results

The clean image data set used in the experiment is downloaded from [37], which contains 10,000 512 × 512 images. Two hundred images are randomly selected, converted from PGM uncompressed image file format into JPEG file format with a compression ratio of 75 for a fair comparison. Based on the analysis described in Section 3, it is apparent that among seven stego message files as the file size increases more alteration whether on JPEG coefficients or image pixel values is made. One can then assume that more changes lead to easier detection. Hence, in the demonstration, three sizes of stego message text files smaller than and equal to 1 KB are exploited, i.e., 0.2 KB, 0.3 KB, and 1 KB, for creating stego images. Additionally, as can be seen in Section 3, JPEG-JSteg v1 and JPEG-JSteg v4 result in higher error measurements, as well as Model-based v1.2 makes larger alteration based on coefficient location. Hence, three JPEG stego methods, F5 [21], Outguess v0.2 [25], and Steghide v0.5.1 [26], are demonstrated in this section due to harder detection intuitively.

Using 10-fold cross validation, Tables 1–3 show the classification accuracies of distinguishing the clean and three JPEG stego methods with the presented PR system. Results for comparison show the classification accuracies when a number of features, 5, 10, 15, or 20, is selected after ranking. Compared to using individual feature sets [13,15,38], a combination of these features tends to corporate the separation between classes and a selected subset of features are selected, extracted, and standardized in order to reduce the problem dimension as well as integrate the classification strength of each individual feature set. Feature combination shows better classification performance compared to other methods.

Table 1
Classification accuracy (%) using the PR system for F5.

Features	File size (KB)	Fridrich [13] (%)	Shi [15] (%)	Chen [38] (%)	Combination (%)
5	0.2	52.5	47.3	47.4	57.2
	0.3	49.9	45.5	46.3	43.1
	1.0	51.8	55.2	51.2	55.2
10	0.2	46.7	38.2	45.3	48.8
	0.3	54.6	51.4	50.6	55.6
	1.0	83.6	56.4	75.7	80.5
15	0.2	51.6	40.5	49.8	44.0
	0.3	73.6	43.5	59.3	61.3
	1.0	88.5	64.8	87.3	80.5
20	0.2	52.4	47.2	48.6	52.2
	0.3	64.8	48.6	59.9	62.5
	1.0	96.5	68.0	86.2	91.6

Table 2
Classification accuracy (%) using the PR system for Outguess v0.2.

Features	File size (KB)	Fridrich [13] (%)	Shi [15] (%)	Chen [38] (%)	Combination (%)
5	0.2	38.8	43.8	48.7	55.6
	0.3	59.5	50.6	47.0	51.8
	1.0	62.9	59.0	58.1	69.6
10	0.2	54.2	50.8	40.4	52.4
	0.3	72.0	58.0	54.9	63.0
	1.0	84.3	69.4	75.1	80.3
15	0.2	63.6	57.6	63.8	58.8
	0.3	69.7	65.3	66.9	72.4
	1.0	89.2	83.1	84.1	92.7
20	0.2	74.5	61.8	66.3	67.9
	0.3	81.3	63.5	77.1	81.9
	1.0	95.5	95.0	94.5	97.4

Table 3
Classification accuracy (%) using the PR system for Steghide v0.5.1.

Features	File size (KB)	Fridrich [13] (%)	Shi [15] (%)	Chen [38] (%)	Combination (%)
5	0.2	52.1	53.8	56.1	53.8
	0.3	50.6	48.5	48.1	48.5
	1.0	58.5	63.5	45.8	63.5
10	0.2	55.3	59.3	58.0	59.3
	0.3	60.6	63.0	58.0	63.0
	1.0	85.6	68.9	73.6	70.2
15	0.2	79.7	72.3	68.8	72.3
	0.3	77.8	64.7	63.0	64.7
	1.0	88.7	81.6	71.1	80.7
20	0.2	75.5	77.5	67.9	82.0
	0.3	75.9	76.1	77.5	79.5
	1.0	93.0	81.7	89.1	88.0

Table 4
Classification accuracy (%) using the PR system.

Stego methods	File size (KB)	CA ± std.(%)	Number of selected features
F5	0.2	66.0 ± 7.7	40
	0.3	72.6 ± 11.0	29
	1.0	93.7 ± 4.2	29
Outguess v0.2	0.2	93.6 ± 5.73	56
	0.3	95.4 ± 3.33	57
	1.0	99.5 ± 1.44	56
Steghide v0.5.1	0.2	93.6 ± 6.35	36
	0.3	93.5 ± 5.12	36
	1.0	94.4 ± 6.74	37

With the entire feature combination and preprocessing system for JPEG steganography detection, Table 4 illustrates the highest classification accuracies ± standard deviations obtained along with the number of features over all the 370 is selected. The system performs over 90% of classification accuracies on both Outguess v0.2 and Steghide v0.5.1 stego methods, even when the small amount of information is embedded. For F5, the system does not perform as good as the others when a 0.2 KB and a 0.3 KB stego message file is incorporated within the clean images. This indicates that the three feature sets for combination during the feature generation stage are not strong enough for classifying the clean and F5 stego images. Other techniques may be incorporated into the system for improvement.

6. Conclusion and discussion

This paper has presented a pattern recognition system for JPEG steganography detection, including feature generation, feature ranking and selection, feature extraction, and neural network classification. Six JPEG stego methods have been discussed and analyzed in Section 3. With these information, three feature sets are combined in the first stage. The advantage of exploiting existing feature generation methods is to decrease the cost of developing new algorithms. Feature ranking and selection then assist to reduce the problem space by determining the class separability of features. Feature extraction mutually excludes the selected raw features to avoid information redundancies. These multivariate preprocessing stages improve the performance on classifying the clean and the stego images. The results in Section 5 show that the PR system strengthens the capability of distinguishing the clean and F5, Outguess v0.2, and Steghide v0.5.1 stego images respectively even when a small amount of information is embedded. Apparently, other current and/or new techniques may be incorporated within the system to have better performance as well as classify various stego methods and anomalies.

References

[1] G.R. Gordon, C.D. Hosmer, C. Siedsma, D. Rebovich, Assessing technology, methods, and information for committing and combating cyber crime, 〈http://www.ncjrs.gov/pdffiles1/nij/grants/198421.pdf〉, 2003.

[2] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[3] D. Kahn, The history of steganography, in: The First International Workshop on Information Hiding, Lecture Notes in Computer Science, Springer Verlag, 1996, pp. 1–5.

[4] N.F. Johnson, S. Jajodia, IEEE Computer (1998) 26.

[5] M.B. Mukasey, J.L. Sedgwick, D.W. Hagy, Electronic Crime Scene Investigation: A Guide for First Responders, second ed., National Institute of Justice, 2008. 〈http://www.ncjrs.gov/pdffiles1/nij/219941.pdf〉.

[6] B.H. Astrowsky, Steganography: hidden images, a new challenge in the fight against child porn, in: Update, American Prosecutors Research Institute, vol. 13, 2000, 〈http://www.ndaa.org/publications/newsletters/update_volume_13_number_2_2000.html〉.

[7] F. Manjoo, The Case of the Missing Code, Salon Media Group Inc., 2002. 〈http://dir.salon.com/story/tech/feature/2002/07/17/steganography/〉.

[8] M. Conway, Knowledge, Technology and Policy 16 (2) (2003) 45.

[9] B.M. Rodriguez, G.L. Peterson, Multi-class classification fusion using boosting for identifying steganography methods, in: Proceedings of SPIE, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, vol. 6974, 2008, p. 697407.

[10] Stegoarchive.com, 〈http://home.comcast.net/∼ebm.md/stego.html〉.

[11] R. Chandramouli, M. Kharrazi, N. Memon, Image steganography and steganalysis: concepts and practice, in: Proceedings of the Second International Workshop on Digital Watermarking, Lecture Notes in Computer Science, vol. 2939, Springer Verlag, 2004, pp. 35–49.

[12] H. Farid, Detecting hidden messages using higher-order statistics, in: Proceedings of the International Conference on Image Processing, vol. 2, 2002, pp. 905–908.

[13] J. Fridrich, Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes, in: Proceedings of the Sixth International Workshop on Information Hiding, Lecture Notes in Computer Science, vol. 3200, Springer Verlag, 2004, pp. 67–81.

[14] S.S. Agaian, H. Cai, New multilevel DCT, feature vectors, and universal blind teganalysis, in: Proceedings of SPIE/IS&T Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII, vol. 5681, 2005, pp. 653–663.

[15] Y.Q. Shi, C. Chen, W. Chen, A Markov process based approach to effective attacking JPEG steganography, in: International Workshop on Information Hiding, Lecture Notes in Computer Science, vol. 4437, 2007, pp. 249–264.

[16] P.T., F.J., Merging markov and dct features for multi-class JPEG steganalysis, in: Proceedings of SPIE/IS&T Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505, 2007, pp. 650503.1–650503.13.

[17] F. Huang, B. Li, J. Huang, Universal JPEG steganalysis based on microscopic and macroscopic calibration, in: Proceedings of the Fifteenth IEEE International Conference on Image Processing, 2008, pp. 2068–2071.

[18] A.K. Jain, R.P. Duin, J. Mao, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4.

[19] R.O. Duda, P.E. Hart, D.G. Stork, Patter Classification, second ed., John Wiley & Sons, New York, 2001.

[20] S. Theodoridis, K. Koutroumbas, Pattern Recognition, fourth ed., Academic Press, 2009.

[21] A. Westfeld, F5—a steganographic algorithm high capacity despite better steganalysis, Lecture Notes in Computer Science, vol. 2137, 2001, pp. 289–302.

[22] N. Provos, P. Honneyman, IEEE Security and Privacy Magazine 1 (3) (2003) 32.

[23] D. Upham, JPEG-JSteg-v4, 〈http://www.nic.funet.fi/pub/crypt/steganography/jpeg-jsteg-v4.diff.gz〉, 1993.

[24] P. Sallee, Model-based steganography, in: International Workshop on Digital Matermarking, Lecture Notes in Computer Science, vol. 2939, Springer Verlag, 2004, pp. 154–167.

[25] N. Provos, Outguess 0.2, 〈http://www.outguess.org/〉, 2001.

[26] S. Hetzl, Steghide, 〈http://steghide.sourceforge.net/〉, 2003.

[27] Information technology – digital compression and coding of continuous-tone still images – requirements and guidelines, 〈http://www.w3.org/Graphics/JPEG/itu-t81.pdf 〉, 1992.

[28] J. Fridrich, M. Goljan, Practical steganalysis of digital images - state of the art, in: Security and Watermarking of Multimedia Contents IV, Proceedings of SPIE, vol. 4675, Springer Verlag, 2002, pp. 1–13.

[29] X.-Y. Luo, D.-S. Wang, P. Wang, F.-L. Liu, Signal Processing 88 (2008) 2138.

[30] A. Westfeld, A. Pfitzmann, Attacks on steganographic systems, in: Proceedings of the Third International Workshop on Information Hiding, Lecture Notes in Computer Science, vol. 1768, 2000, pp. 61–76.

[31] WetStone, Stego suite, 〈http://www.wetstonetech.com/cgi-bin/shop.cgi?view,1〉, 2009.

[32] Backbone, StegalyzerRTS, 〈http://www.sarc-wv.com/〉, 2009.

[33] C. Chen, Y.Q. Shi, JPEG image steganalysis utilizing both intrablock and interblock correlations, in: IEEE International Symposium on Circuits and Systems, 2008, pp. 3029–3032.

[34] M.C. Chen, S.S. Agaian, C.L.P. Chen, B.M. Rodriguez, Steganography detection using RBFNN, in: Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, vol. 7, 2008, pp. 3720–3725.

[35] R.C. Reininger, J.D. Gibson, IEEE Transactions on Communications 31 (6) (1983) 835.

[36] E.Y. Lam, J.W. Goodman, IEEE Transactions on Image Processing 9 (10) (2000) 1661.

[37] P. Bas, T. Furon, Break our watermarking system 2nd ed., 〈http://bows2.gipsa-lab.inpg.fr/〉, 2008.

[38] M.C. Chen, S.S. Agaian, C.L.P. Chen, B.M. Rodriguez, Alpha-trimmed image estimation for JPEG steganography detection, in: Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, 2009, pp. 4718–4722.

[39] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, 1990.

[40] A.R. Webb, Statistical Pattern in Recognition, second ed., John Wiley & Sons, Inc., xx, 2002.

[41] J.B. Bednar, T.L. Watt, IEEE Transactions on Acoustics, Speech, and Signal Processing 32 (1) (1984) 145.

[42] H. Hotelling, Journal of Educational Psychology 24 (1933) 417 498–520.

[43] S. Chen, C.F.N. Cowan, P.M. Grant, IEEE Transactions on Neural Networks 2 (2) (1991) 302.

[44] H. Demuth, M. Beale, M. Hagan, Raidal basis networks, neural network toolbox^{TM} 6 user's guide, The MathWorks Inc., 2009, 〈http://www.mathworks.com/access/helpdesk/help/pdf_doc/nnet/nnet.pdf〉.

[45] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, vol. 2, 1995, pp. 1137–1145.