StegTrack: Tracking images with hidden content

Veenu Bhasin
Department of Computer Science,
University of Delhi, Delhi-110007, India
91 11 2766 7059
vbhasin@cs.du.ac.in

Punam Bedi
Department of Computer Science,
University of Delhi, Delhi-110007, India
91 11 2766 7059
pbedi@ieee.org

Aakarshi Goel
Department of Computer Science,
University of Delhi, Delhi-110007, India
aakarshi92@gmail.com

Sukanya Gupta
Department of Computer Science,
University of Delhi, Delhi-110007, India
sukanya.mcs.du.2012@gmail.com

ABSTRACT
This paper presents the design and implementation of StegTrack, a novel proactive steganalysis tool. StegTrack is an antivirus like tool to track steganograms among images on a computer. To the best of our knowledge such a tool does not exist in literature. Once installed on a machine, StegTrack always remains active. It keeps track of user's entire file system and detects arrival of new images in the system. Every image entering the system is tested for steganography. Steganalysis, the process to detect the presence of the hidden data/message, has two major components feature extraction and classification. The StegTrack tool gives user the flexibility of choosing the feature extractor as well as classifier although a default is provided for both. The tool provides various feature extraction options like features based on Markov Model, co-occurrence matrix, neighboring joint density probabilities, Run-length matrix, SPAM and statistical features. The classifier opted as default in the StegTrack is ELM to provide multi-class classification results in real time. This proposed tool also provides a new feature - cleaning of stego-image, where image is rendered unfit for extracting hidden material from it. A prototype for the tool was implemented in MATLAB and Java.

Categories and Subject Descriptors
D.4.6 [Security and Privacy]: • Security and privacy~Systems security

Keywords
Steganalysis, steganography, feature extraction, ELM.

1. INTRODUCTION
Steganography, an information hiding technique, is used to hide data in the innocuous cover object [1]. Different digital media is used as cover objects, with images being a popular option as they are being uploaded and exchanged over internet more than the other digital media. Image steganography ensure that the data is hidden in images making it completely invisible to human visual system as well as other standard image viewers.

© 2015 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

WCI '15, August 10 - 13, 2015, Kochi, India
© 2015 ACM. ISBN 978-1-4503-3361-0/15/08…$15.00
DOI: http://dx.doi.org/10.1145/2791405.2791451

The steganograms can carry harmful payloads. Steganalysis, the counter-technique of steganography, tries to detect the presence of the hidden data/message. Blind Steganalysis is the process of discrimination between stego-objects and non-stego-objects without prior knowledge of steganography method employed to hide the data. Steganalysis techniques use various methods, predominantly statistical analysis, to determine if there is any hidden data. Information security research community is focusing on developing steganalysis techniques and lots of new techniques are being discussed, but no consolidated tool exist to the best of our knowledge that can help an individual in blocking the steganographic content to enter its computer. Such a tool will be for steganography, what antiviruses are for virus attacks. Like Antivirus is used to protect computer systems from the various virus attacks, a tool with an ability to detect and stop steganography will make the digital communication safer.

The detection tools for steganography that exist are StegSpy, StegDetect and Stegbreak. StegSpy [2] uses signature analysis to find hidden data and works for steganography techniques that leave some signature in the image file, like Hiderman and InvisibleSecrets. StegDetect developed by Neil Provos perform steganalysis of JPEG images, although he was unable to detect a single image with hidden data using this software from the millions of JPEG images from sites like eBay and USENET [3]. Stegbreak detects steganograms created using steganography techniques JSteg-Shell, JPHide and OutGuess 0.13b by launching dictionary attack against them. All these tools work on a specific image file type for detecting some specific steganography techniques only and are not upgradable to detect new Steganographic techniques.

This paper proposes a novel proactive steganalytic tool - StegTrack - that finds presence of message/information hidden in digital images using various steganography schemes (e.g. LSB, F5, Invisible Secrets, OutGuess, Model-based steganography, JPHide, and JSteg). The tool, StegTrack is a memory resident program, which tracks the traffic of images on a system and performs steganalysis on this dynamic data i.e., the images. The user interface proposed is graphical. To the best of our knowledge such a proactive comprehensive tool, which performs blind steganalysis using various steganalytic features for a variety of image types, does not exist in literature. As a lot of research efforts by information security fraternity are being put towards steganalysis, new and better steganalysis techniques are being proposed for combating different steganography methods. This

proposed tool is also upgradable to include new steganalysis methods. The paper also proposes to add a novel feature to StegTrack tool which enables the tool to modify the image in such a way that Steganographic material in it cannot be extracted. This means, if a non-clean image file, i.e. a stego-image, enters the system, the user is informed and if the user still wants to keep the file, the image file is cleaned so that Steganographic material in it cannot be extracted. This feature is not available in any of the existing steganalysis tools as per our knowledge. StegTrack, apart from detecting stego-images, is also capable of detecting steganography tool used to embed data. For classification, Extreme Learning Machine (ELM) is provided as the default option. ELM being a very fast classifier suits well as the tool is expected to provide results in real time. Moreover, ELM is a multi-class classifier, thus apart from detecting stego-images the tool also reports the steganography method used to create this stego-image.

This tool is an attempt to provide information security community a tool against steganography that can potentially play quite a large role in protection against malware communication.

The rest of the paper is organized as follows. Section 2 gives a brief account of Image Steganalysis. Section 3 presents the framework for the proposed steganalytic tool StegTrack. The prototype built for the proposed tool is described in section 4 followed by the conclusion in section 5.

2. IMAGE STEGANALYSIS

Steganography hides the message in other information items so that the message as well as the presence of the message is concealed from the third party. Omnipresence of images in the modern communication increases the chances of images being used for steganography much more than any other media. Many different carrier image formats (e.g. JPEG, TIFF, BMP etc.) are available for image steganography.

Steganalysis is the art of attacking steganographic methods by detection or by modification or extraction of embedded data [1]. Blind Steganalysis refers to the process of discrimination between stego-objects and non-stego-objects without prior knowledge of the cover image or the method employed to hide the data. Steganographic procedures on images alter the statistical properties of the image, which can be used to detect presence of messages.

The process of image steganalysis techniques (Fig. 1) is like that of pattern-recognition where some features calculated from images are analyzed and decision is taken based on this analysis. These techniques use a multidimensional feature set. These features are analyzed by classifier. The features, extracted from the training images, are used to train a classifier, which is used to classify test images. There is abundance of steganalysis methods in literature for various formats of digital images, which mainly differ in the feature extraction process.

Avcibas et. al. [4] gave a Steganalysis technique where the classifier uses multivariate regression on the image quality metrics and the training of the classifier is done on the basis of an estimate of the original image. Hossein et. al. [5] have given spatial as well as frequency domain features for steganalysis of gray-scale images and used SVM for classification. Many authors have proposed Steganalysis based on gray-level co-occurrence matrix [6] [7] [8]. S. Lyu and H. Farid [9] proposed a steganalysis method using wavelets decomposition of colored images and have used first and higher order magnitude and phase statistics. G. Xuan et al [10] have described Steganalysis based on feature vector set constructed from main diagonal and upper four diagonals of co-occurrence matrix of gray-scale images and used Class-wise Non-Principal Components Analysis to classify images into cover and stego. G. Xuan et al [11] presented steganalysis based on the multiple features formed by statistical moments of wavelet characteristic functions and Bays Classifier. S. Ghanbari et al [12] presented steganalysis technique in which features, extracted from Gray-Level Co-occurrence Matrix (GLCM), are used for training four layers Multi-Layer Perceptron (MLP) neural network.

The feature extraction in case of JPEG images has been done from different domains [13] like spatial, DCT, DWT and DWT etc. Several studies have led to development of different feature extraction models for steganalysis such as statistical [14], histogram, co-occurrence matrix [6], gray level run length matrices [15], IQM [4], SPAM [16], CF moments [17], neighboring joint density probabilities (NJD) [18] and Markov model [19] [20] etc. Pevny and Fridrich [21] have merged the features from different models, i.e. DCT and Markov model, to improve accuracy of steganalysis method. Some authors [19] [22] have used calibrated features to get better detection results. H. Zong et al [23] described a blind JPEG steganalysis method based on the correlation of inter- and intra-wavelet sub bands in the wavelet domain and using back propagation neural network classifier.

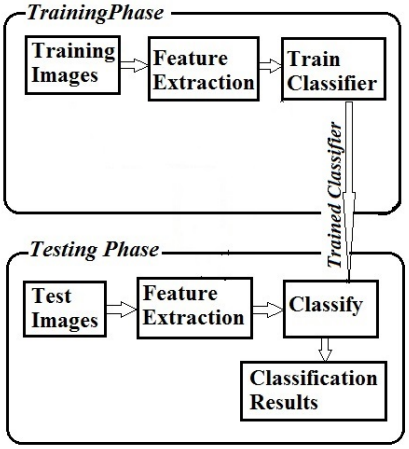

Figure 1 Image Steganalysis Process

Each of the above mentioned steganalysis methods work on given set of images, all being of same format and size. In spite of many steganalysis techniques available in literature, a concise GUI based tool, to safeguard from steganography is not available, which can serve as one stop solution against steganography content. In this paper we propose a novel steganalytic tool - StegTrack - which will find steganograms among the images on a system or being copied to a system, as antiviruses look for the presence of virus infected files. Feature extraction options provided by StegTrack includes statistical features, co-occurrence matrix based features, gray level run length matrices based features, SPAM features, neighboring joint density probabilities (NJD) and Markov model based features. The tool also includes option to clean a steganogram so that the Steganographic material cannot be extracted by a Steganographic tool.

3. FRAMEWORK FOR THE PROPOSED TOOL StegTrack

The proposed tool, StegTrack, looks for hidden content in digital images. Apart from checking and scanning the images on system for steganography contents, it also acts like a gateway, scanning image traffic on the system for steganograms in real time. Once installed, the tool remains in memory and does tracking and scanning of the images. Whenever an image enters system, features are extracted and then it is checked using trained classifier, specific to the type of this image being tested. The user will be warned if the image is a steganogram and the possible stego-tools which has been used to make this steganogram is displayed. The tool also gives an option to clean the image.

Various functional features of the proposed tool are as follows:

• *Installation*: The installation of the tool prepares it for tracking and scanning. All the drives present on the system and their directory tree are registered for checking any event of image transfer. The tool browses thru the directory tree of each drive and registers them in order to track them all for any image entering them.

The tool extracts features from the training images and trains the classifier. The extracted features and the trained classifier are stored for future use. This is repeated for different types of images like, BMP, JPEG, and TIFF etc.

• *Tracking*: The steganalysis tool StegTrack is a memory-resident tool that remains active to track the traffic of images on the system and scans this traffic in real time for steganograms. Any image entering into the system – copied from an external device or being downloaded from internet - is tracked and scanned.

• *Scanning*: As soon as an image enters the system, it is scanned by the tool. The thumbnail of the image being scanned is shown in order to give the user idea of the image. The tool also provides the user the option to specify a drive or a directory name, thereby allowing user to specify scanning of multiple images with one command.

StegTrack is proposed to check for digital images of any size and format (e.g. JPEG, BMP, GIF etc.). Depending on the format of the image being scanned it chooses appropriate steganalysis technique to be used. To achieve this various different steganalysis techniques have been brought together in order to make this choice appropriate. For example for a JPEG image, Markov based or NJD based features are chosen.

StegTrack gives detailed information about scanned images in real time. The various information shown include steganography algorithm used (if image is a steganogram) and processing time, with other useful information as image size and location, image type and file name.

• *Cleaning*: Once an image is identified as steganogram, the tool tries to clean the image making it impossible to retrieve the message at a later point in time.

The proposed tool aspires to run on multiple platforms.

3.1 Architecture of the proposed tool

The proposed tool has several components, as shown in Fig. 2, each having its own specialty as well as the functionality attached to it. The most important component is the Engine, which binds all components together and also controls them. The next component is the User Interface (GUI) which has the functionality of enabling the graphical interface. The other components are Tracker, Feature Extraction and Classifier. The tool also provides provision for cleaning an image steganogram. The tool has a data Repository containing training image sets (consisting of stego and non-stego images) for different image formats. In rest of this section the details of all the components are given.

• *Engine*: This is the central component of the tool that binds and controls all the other components of the tool. This component initializes installation and controls tracking, scanning and cleaning of the images after the tool is installed. During installation it activates tracking of the system drives and initiates the steganalysis process by extracting features from training data and training the classifier. All the extracted features and trained classifier are saved for future use.

When Tracker detects an image entering the system, Engine invokes the relevant feature extraction procedure (for extracting the new image's features) and uses trained classifier to identify this image as potential steganogram or an innocent image.

When any change is made in the contents of Repository, the Engine repeats the feature extraction and classifier training cycle.

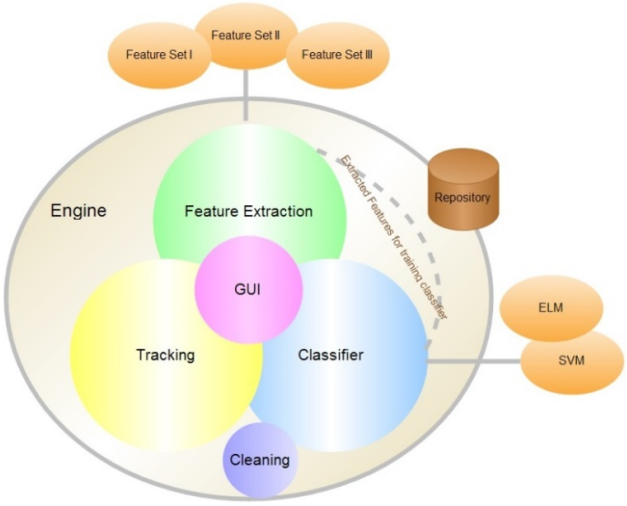

Figure 2 Architecture of the Proposed Tool

• *User Interface*: A graphical user interface (GUI) is proposed for the system. The tool's GUI provides following options:

- Start the tool
  - Install and make the tool ready for use.
- Scan a disk/drive for steganograms
  - By default the whole system will be scanned and watched for steganograms, a provision for scanning a pen-drive or other drives is also available
  - The tool also gives an option to check a directory for potential stego material
- Choose an image type
  - Training images will be provided for each type, in the Repository, but a provision for user to include more images will also be given
- Choose the steganalytic feature set
  - The tool includes several steganalytic feature sets extraction techniques and an option for user-defined technique is given

- Choose the classifier (SVM or ELM), default being ELM
- Cleaning the steganogram
- Stop the tool

The entry of an image into system and its classification result are reported to the user. The GUI provides this information in real time. For the image that have been scanned, the tool reports time spent in processing, image type, file name, image size and location along with a thumbnail of the image. If the image is found to be steganogram the possible steganography algorithm used is also shown.

• *Tracker*: This component is responsible for tracking the image traffic on system. It registers all the drives of the system and detects any external drive if attached. The tracker component uses file system interface of the operating system and has file type scanner and memory scanner. It registers all the drives, with their entire directory structure, for keeping a watch for any file entering the system. In the event of any file entering in any of the directories on system, Tracker checks the file type of the entered file and if it is an image file then the Engine is informed. If an external drive is attached to the system, Tracker scans this drive for any image and informs the Engine accordingly.

• *Feature Extraction*: This component is responsible for extracting the relevant features of the image. During installation of the tool, the features extraction is done for various image types (as different file type may need steganalysis based on different feature sets) using the training image data set provided with the tool.

The features generated from the training set are used to train the classifier. When an image needs to be tested, features are extracted from that image and the trained classifier is used for classifying this image as stego or non-stego image. The training sets of images are provided in Repository, the user can add images to this directory. In this event, the Engine invokes the feature extraction processes so that feature extraction of the new training set happens.

The Feature Extraction component provides several extraction procedures for feature sets corresponding to various steganalysis methods. The feature extraction options available are for statistical features, co-occurrence matrix based features, gray level run length matrices based features, SPAM features, neighboring joint density probabilities (NJD) and Markov model based features etc. The user also has an option for specifying and adding a user-defined steganalytic feature extraction procedure, thereby making this tool upgradable with respect to new steganalysis techniques.

• *Classifier*: The steganalysis method of the proposed tool is based on supervised learning pattern-classification technique, where a classifier is trained, using feature extracted from the training dataset of images, before actually classifying the images. The tool provides an option to choose among the two classifiers: Support Vector Machine (SVM) and Extreme Learning Machine (ELM). SVM [24] is a binary classifier, whereas ELM [25] can perform multi-classification. Thus the choice of classifier also amounts to picking up one among binary classification and multi-classification. In the latter case, apart from classifying images as non-stego or stego, they are also categorized into classes which correspond to different steganography methods. Also ELM is a fast classifier [26], hence provides scan results in real time.

Based on the chosen option, the selected classifier is trained on the training image datasets provided in Repository of the tool, during installation of the tool. Using this trained classifier along with the feature vector of the image, that needs to be tested, the image is classified as either steganogram or Non-steganogram. The possible steganography technique used to create a steganogram may also be reported by the tool.

• *Cleaning*: This component is used if the user opts to keep the images identified as steganograms and asks for cleaning images of potentially harmful steganographical content without attempting to extract the hidden data. This novel feature has been added in the proposed tool. The process does not attempt to get the cover image; instead it tries to destroy the effect of steganography rendering it useless. Once an image is identified as steganogram and user has chosen to clean it, the tool tries to clean the image such that it might become impossible to retrieve the message at a later point in time. The two methodologies used for cleaning of the stego-image are:

- Resizing - Resize the image to half by replacing four adjacent pixels by one pixel which is equivalent to the average of these four pixels
- Smoothing - Replace each pixel value with the mean value of its neighbors and that of itself. A pixel value which is unrepresentative of its surroundings (i.e., might have been introduced by the steganography methods), is eliminated by this process. In this proposed tool, the value of a pixel, A(i,j) is computed using values of the eight neighboring pixels and itself as follows:

$$\text{Mean}(i,j) = \tfrac{1}{8}[A(i-1,j-1) + A(i-1,j) + A(i-1,j+1) + A(i,j-1) + A(i,j+1) + A(i+1,j-1) + A(i+1,j) + A(i+1,j+1)] \quad (1)$$

$$A(i,j) = (1/2)\,[A(i,j) + \text{Mean}(i,j)] \quad (2)$$

By default, Smoothing is used for cleaning the stego-images, but user is provided an option to choose between Resizing and Smoothing.

4. PROTOTYPE FOR StegTrack (Stego-Tracker)

For testing the utility and feasibility of the proposed StegTrack tool, a prototype was implemented for JPEG steganalysis in Java and MATLAB. This prototype tool developed for JPEG steganalysis was called Stego-Tracker. Four screen shots of Stego-Tracker are given in Fig.3 - Fig.6.

Stego-Tracker, a prototype of StegTrack, is a software application built to carry out steganalysis process for JPEG images. It is a memory-resident program. After initialization (Fig. 4), it keeps track of user's entire file system for detecting arrival of new JPEG image in the system. It then extracts specified features for each new image, store those features and label them as feature vectors.

Stego-Tracker gives the user an option to extract features of his choice (by providing his own MATLAB code). The different feature sets tested included SPAM [16], Neighboring Joint Density Probabilities (NJD) based [18], Markov model based [19] [20]. It also lets user choose among the two classifiers: SVM and ELM (Fig.5). Based on it, the selected classifier is trained on the default image datasets (consisting of stego and non-stego images) provided by the software. Using this trained classifier along with the feature vector (of the detected JPEG image) the image is

tested and classified as either steganogram or non-steganogram. The result is reported to the user as in Fig.6.

Testing for image files from different sources was performed. Images on the computer were classified into stego and non-stego images using this tool. The image files being downloaded or being copied from a pen drive were successfully detected and then checked for steganograms among them.

Figure 3 Prototype for StegTrack (Stego-Tracker)

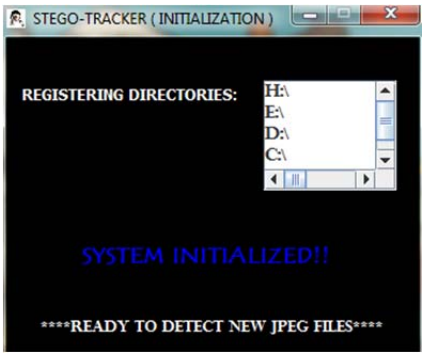

Figure 4 Stego-Tracker Initialization

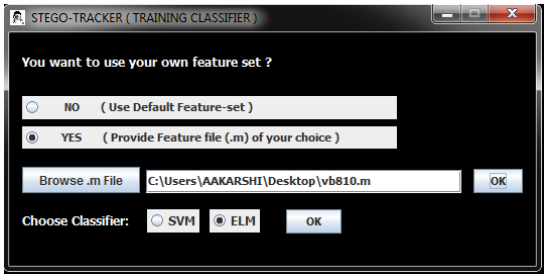

Figure 5 Various Options available in Stego-Tracker

Figure 6 Reporting of a steganogram by Stego-Tracker

Both the cleaning methods were successfully used to clean steganograms. Fig. 7 and Fig. 8 show a stego-image before and after smoothing respectively. The stego-image was created using OutGuess and message extracted was same as the message embedded while creating this stego-image, but the message extracted from the image got after smoothing was empty.

The feature extraction process, training of the classifier and the cleaning is carried out in MATLAB, whereas the GUI is developed in Java. Stego-Tracker uses Java inbuilt classes to register and keep a watch on the entire system. MATLAB code is integrated with Java code using an API called MatlabControl.

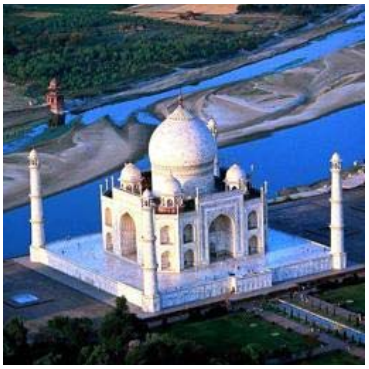

Figure 7 Stego-Image

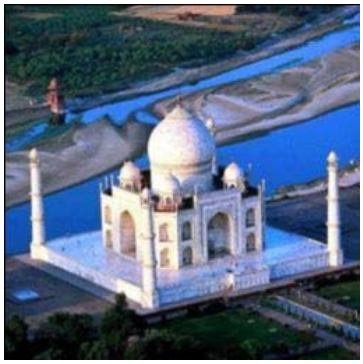

Figure 8 Stego-Image after cleaning (smoothing)

5. CONCLUSION

A comprehensive steganalytic tool, StegTrack that provides the user a proactive defense against Steganographic material has been presented in this paper. Once installed this novel tool remains active and monitors the computer's file system and keeps track of the image traffic in order to catch any steganogram that might have entered the system. The tool consists of various components having separate functionality and working seamlessly with each other. The proposed tool – StegTrack - strives to provide a safeguard against steganography. StegTrack uses ELM as classifier and is capable of detecting steganograms as well as the steganography tool used in real time. The tool introduces a novel concept of cleaning of the stego-images which means hidden message becomes irretrievable after cleaning of stego-image if user wants to keep the image. A prototype of the tool was developed, specifically for JPEG images. This prototype was developed using MATLAB and Java.

6. ACKNOWLEDGEMENT

The authors duly acknowledge the University of Delhi for the support in work on this paper under the research grant number RC/2014/6820.

7. REFERENCES

1 Katzenbeisser, Stefan and Petitcolas, Fabien A. P. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, 2000.

2 Raggo, Michael T and Hosmer, Chet. *Data Hiding: Exposing Concealed Data in Multimedia, Operating Systems, Mobile Devices and Network Protocols*. Newnes, 2012.

3 Provos, Neil and Honeyman, P. *Detecting Steganographic Content on the Internet*. CITI, University of Michigan, San Diego, 2001.

4 Avcıbas, Ismail , Memon, Nasir, and Sankur, Bülent. Steganalysis Using Image Quality Metrics. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 12, 2 (2003), 221-229.

5 Malekmohamadi, H. and Ghaemmaghami, S. Steganalysis Of Lsb Based Image Steganography Using Spatial And Frequency Domain Features. In *IEEE international conference on Multimedia and Expo* (2009), 1740-1743.

6 Abolghasemi, M., Aghainia, H., Faez, K., and Mehrabi, M. A. LSB data hiding detection based on gray level co-occurrence matrix. In *International symposium on Telecommunications* (Iran, Tehran 2008), 656-659.

7 Kekre, H. K., Athawale, A. A., and Patki, S. A. Steganalysis of LSB Embedded Images Using Gray Level Co-Occurrence Matrix. *International Journal of Image Processing*, 1, 1 (2011), 36-45.

8 Aghainia, H., Abolghasemi, M., Faez, K., and Mehrabi, M. A. Steganalysis of LSB Matching Based on Co-Occurrence Matrix and Removing MSB Planes. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (2008), IEEE Computer Society Washington, 1527-1530.

9 Lyu, Siwei and Farid, H. Steganalysis Using Higher-Order Image Statistics. *IEEE Transactions on Information Forensics and Security*, 1, 1 (March 2006), 111 - 119.

10 Xuan, Guorong, Shi, Yun Q., Huang, Cong, Fu, Dongdong, Zhu, Xiuming, Chai, Peiqi, and Gao, Jianjiong. Steganalysis Using High-Dimensional Features Derived from Co-Occurrence Matrix and Class-wise Non-Principal Components Analysis (CNPCA). In *International Workshop on Digital Watermarking* (Jeju Island, Korea 2006), Lecture Notes in Computer Science.

11 Xuan, Guorong, Shi, Yun Q., Gao, J. et al. Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions. In *7th international conference on Information Hiding* (Barcelona, Spain 2005), Lecture Notes in Computer Science, Springer, 262-277.

12 Ghanbari, S., Keshtegary, M., and Ghanbari, N. New Steganalysis Method using Glcm and Neural Network. *International Journal of Computer Applications*, 42, 7 (2012), 45-50.

13 Kharrazi, Mehdi, Sencar, H. T., and Memon, Nasir. Benchmarking steganographic and steganalysis techniques. *Electronic Imaging 2005* (March 2005), 252-263.

14 Fridrich, Jessica. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In *6th Information Hiding Workshop* (Toronto, ON, Canada 2004).

15 Seyedhosseini, Mojtaba and Ghaemmaghami, Shahrokh. Detection of LSB Replacement and LSB Matching Steganography Using Gray Level Run Length Matrix. In *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (Kyoto, Japan 2009), IEEE, 787-790.

16 Pevný, Tomáš, Bas, Patrick, and Fridrich, Jessica. Steganalysis by Subtractive Pixel Adjacency Matrix. *IEEE Transactions on Information Forensics and Security*, 5, 2 (June 2010), 215-224.

17 Hui, Li, Ziwen, Sun, and Zhiping, Z. An Image Steganalysis Method Based on Characteristic Function Moments and PCA. In *Proc. of the 30th Chinese Control Conference* (Yantai, China 2011).

18 Liu, Qingzhong, Sung, Andrew H., and Qiao, Mengyu. Neighboring joint density-based JPEG steganalysis. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2, 2 (Feb 2011), 16:1-16:16.

19 Bhasin, Veenu and Bedi, Punam. Steganalysis for JPEG Images Using Extreme Learning Machine. In *IEEE International Conference on Systems, Man, and Cybernetics* (Manchester, UK 2013), IEEE, 1361-1366.

20 Chen, Chunhua and Shi, Yun Q. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *IEEE ISCAS, International Symposium on Circuits and Systems* (Seattle, Washington, USA 2008), 3029–3032.

21 Fridrich, Jessica and Pevny, Tomas. Merging Markov and DCT features for multiclass JPEG steganalysis. In *Proc. SPIE Electronic Imaging,Security, Steganography, and Watermarking of Multimedia Contents IX* (San Jose, CA 2007), 650503-1-650503-13.

22 Fridrich, Jessica and Kodovsky, J. Calibration revisited. In *11th ACM Multimedia & Security Workshop* (Princeton, NJ 2009), 63–74.

23 Zong, H., Liu, F., and uo, Xi. Blind image steganalysis based on wavelet coefficient correlation. *Digital Investigation*, 9, 1 (June 2012), 58-68.

24 Jakkula, V. *Tutorial on Support Vector Machine (SVM)*. School of EECS, Washington State University, Pullman, 2006.

25 Huang, Guang-Bin, Zhu, Qin-Yu, and Siew, Chee-Kheong. Extreme learning machine: a new learning scheme of feedforward neural networks. In *International joint conference on neural networks (IJCNN)* (Budapest, Hungary 2004), 985–990.

26 Bhasin, Veenu and Bedi, Punam. Multi-class JPEG Steganalysis Using Extreme Learning Machine. In *ICACCI-WCI 2013 - International Symposium on Women in Computing and Informatics* (Mysore, India 2013), IEEE, 1948-1952.