



INAOE

Blind Steganalysis Method for Detection of Hidden Information in Images

by
Marisol Rodríguez Pérez

A dissertation submitted in partial
fulfillment of the requirements for the degree of

Master in Computer Science

at the

National Institute for Astrophysics, Optics and Electronics
September 2013
Tonantzintla, Puebla

Advisors:

Claudia Feregrino Uribe, PhD., INAOE
Jesús Ariel Carrasco Ochoa, PhD., INAOE

©INAOE 2013

All rights reserved

The author hereby grants to INAOE permission to reproduce and to
distribute copies of this thesis document in whole or in part



RESUMEN

Desde la antigüedad, la esteganografía ha sido utilizada para proteger información sensible de personas no autorizadas. Sin embargo, junto con la evolución de los medios digitales han surgido usos no deseados, como el terrorismo, la pornografía infantil, entre otros. Para contrarrestar los posibles efectos negativos, surge el esteganálisis. Existen dos enfoques principales de esteganálisis: específico y universal o ciego. Los métodos específicos requieren de un conocimiento previo del método esteganográfico analizado, mientras que los métodos ciegos no lo requieren. Debido a los altos requerimientos de aplicaciones reales, es necesario el desarrollo de métodos de esteganálisis cada vez más precisos que sean capaces de detectar información oculta de diversos métodos esteganográficos. Tomando esto en cuenta, proponemos un método ciego de esteganálisis para imágenes a color. El método propuesto se basa en el proceso estándar de esteganálisis, el cual consiste en la extracción de características y su posterior clasificación. Con el fin de que el método sea extensible, se utilizaron distintos extractores de características, así como un ensamble de clasificadores. Los experimentos realizados con diferentes tasas de inserción para distintos métodos esteganográficos, muestran una mejora de la tasa de detección sobre los métodos del estado del arte con un solo extractor de características y un solo clasificador, esto para F5, Spread Spectrum, LSBMR y EALSBMR. Para Steghide, JPHide y Model Based las tasas de detección apenas sobrepasaron el azar para tasas de inserción por debajo de 0.05bpp.

ABSTRACT

Since ancient times, steganography has been widely used to protect sensitive information against unauthorized people. However, with the evolution of digital media, unwanted uses of steganography, like terrorism, child pornography, among others, have been recognized. In this context, steganalysis arises as a countermeasure to the side effects of steganography. There are two main steganalysis approaches: specific and universal, also called blind. Specific methods require previous knowledge of the analyzed steganographic technique under analysis, while, universal methods do not. Due to the demanding requirements of real applications, it is necessary develop of even more accurate steganalysis methods capable to detect hidden information of diverse steganographic techniques. Taking this into account, we propose a universal steganalysis method specialized in color images. The proposed method is based on the standard steganalysis process, where a feature extractor and a classifier algorithm are used. To develop a flexible and scalable method, we use different feature extractors and a meta-classifier. The experiments were carried out for different embedding rates and steganographic methods. The results show that the proposed method outperforms the detection rate of state of the art methods with a single feature extraction and a single classifier, for F5, Spread Spectrum, Least Significant Bit Matching Revisited (LSBMR) and Edge Adaptive LSBMR. For Steghide, JPHide and Model Based, the detection rate was poor for embedding rates under 0.05bpp.

En memoria de mi padre.

AGRADECIMIENTOS

A mi esposo por alentarme a continuar mis estudios y acompañarme en el proceso. Gracias Iván por tu amor y tu apoyo.

A mi familia por su apoyo incondicional. A mi papá por sus consejos y su amor; en donde estés te dedico todos mis logros. A mi mamá por sus sacrificios y su amor absoluto. A mi hermanita por ser mi compañía y confidente.

A mis asesores Dra. Claudia Feregrino Uribe y Dr. Jesús Ariel Carrasco Ochoa por sus enseñanzas, así como por el apoyo, paciencia y tiempo invertidos en hacer este trabajo de investigación realidad.

A mis sinodales Dr. Rene Armando Cumplido Parra, Dra. Alicia Morales Reyes y Dr. Hugo Jair Escalante Balderas por sus atinados comentarios y observaciones.

A mis compañeros y amigos Ulises, Lindsey, Paco, Daniel, Metzli, Ricardo y Roberto por compartir esta experiencia.

Al Instituto Nacional de Astrofísica, Óptica y Electrónica por todas las atenciones y facilidades prestadas.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el financiamiento a través de la beca 322612.

TABLE OF CONTENTS

1 Introduction	1
1.1 Introduction	1
1.2 Motivation.....	3
1.3 Main Objective.....	4
1.4 Specific Objectives	4
1.5 Methodology	4
1.6 Thesis Organization	6
2 Background.....	7
2.1 Steganography.....	7
2.1.1 Steganography and Cryptography.....	8
2.1.2 Steganography and Information Hiding.....	9
2.1.3 Steganography Applications	10
2.2 Steganalysis	11
2.2.1 Steganalysis Categorization	12
2.2.2 Steganalysis Process	15
2.2.2.1 Feature Extraction	15
2.2.2.2 Classification	16
3 State of Art.....	18
3.1 Steganographic Methods.....	18
3.1.1 Least Significant Bit (LSB) Family	19
3.1.1.1 Steghide.....	19
3.1.1.2 JPHide and JPSeek	20

3.1.1.3 LSB Matching	20
3.1.1.4 LSB Matching Revisited	21
3.1.1.5 Edge Adaptive LSB Matching Revisited	21
3.1.2 Model-Based.....	21
3.1.3 F5 Steganography	22
3.1.4 Spread Spectrum	23
3.1.5 Other Steganographic Methods	24
3.2 Steganalysis Methods	25
3.2.1 Subtractive Pixel Adjacency Model (SPAM).....	26
3.2.2 Local Binary Pattern (LBP).....	26
3.2.3 Intrablock and Interblock Correlations (IIC)	28
3.2.4 Higher Order Statistics (HOS)	30
3.2.5 Other Steganalysis Methods.....	31
3.3 Summary and Discussion.....	33
4 Proposed Method.....	35
4.1 Proposed Method.....	35
4.2 Feature Extraction	37
4.1.1 Subtractive Pixel Adjacency Model	38
4.1.2 Local Binary Pattern	39
4.3 Classification.....	40
4.4 Chapter Summary	42
5 Experiments and Results	44
5.1 Experimental Setup.....	44
5.1.1 Dataset.....	44
5.1.2 Embedding Software.....	46

5.1.3 Classification	48
5.2 Results	49
5.3 Analysis and Discussion	52
5.4 Chapter Summary	55
6 Conclusions and Future Work.....	56
6.1 Contributions.....	56
6.2 Conclusions	56
6.3 Future Work.....	59
Bibliography	60

LIST OF FIGURES

Figure 1.1 Methodology	5
Figure 2.1 Steganographic traditional scenario.....	8
Figure 2.2 Passive warden scheme.....	11
Figure 2.3 Active warden scheme	12
Figure 2.4 Visual based steganalysis.....	13
Figure 2.5 Steganalysis categorization.....	14
Figure 2.6 Steganalysis process	15
Figure 3.1 F5 embedding process	23
Figure 3.2 Some steganalysis methods of the state of the art.....	25
Figure 3.3 Example of LBP value calculation	27
Figure 3.4 Interblock and intrablock correlation.....	29
Figure 3.5 Interblocking alignment.....	29
Figure 3.6 Multi-scale lowpass subband, horizontal, vertical and diagonal	31
Figure 4.1 Proposed method.....	37
Figure 4.2 SPAM process	39
Figure 4.3 LBP Process	40
Figure 4.4 Proposed classification method	42
Figure 5.1 Example of images from the dataset.....	45
Figure 5.2 Cover image (left) and Steghide embedded image (right).....	47
Figure 5.3 Pixels modified after embedding 0.005bpp with Steghide.....	48

Figure 5.4 Comparison between embedding rates detection of F5, SS, LSBMR and EALSBMR.....53

Figure 5.5 Comparison between embedding rates of Steghide, JPHide and Model Based.....54

LIST OF TABLES

Table 5.1 Review of the steganographic methods used in the experiments	46
Table 5.2 Detection rate results of second level classification for 0.005bpp embedding rate	50
Table 5.3 Detection rate results of joined feature space for 0.005bpp embedding rate	50
Table 5.4 Experiment detection rate results for 0.005bpp embedding rate	51
Table 5.5 Experiment detection rate results for 0.01bpp embedding rate.	51
Table 5.6 Experiment detection rate results for 0.05bpp embedding rate.	52

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Information privacy has always been an issue that concerns everyone. Throughout history, many techniques have been developed trying to protect sensitive information against unauthorized people. The art of hiding information without arousing suspicion is called **Steganography**. The term came from the Greek *steganos* meaning “covered” and *graphos* meaning “writing”. The first record about the term steganography was in the book *Steganographia* written by Johannes Trithemius in 1499. However, despite the title, the book was mainly about cryptography techniques and esoteric subjects (De Leeuw and Bergstra 2007).

The first documented steganographic technique was described by Herodotus, when people used to write in wax-covered tablets, a message could be unnoticed under the wax. Later, Aeneas Tacticus was responsible for providing a guide to securing military communications. He described several forms to hide physically a message, like women’s earrings or pigeons or in a letter as small holes over the paper hidden in the text. Another popular ancient technique called acrostic, consists in hiding a message in a specific spot of every word in a text, for example a poem.

Despite its simplicity, the use of acrostics has survived until modern wars. The invisible ink is another technique widely used, even in this time. Whether it be just for fun or for hiding a message in war times, invisible ink has evolved from natural substances to more complex chemicals. More recently, in 1870 during Franco-Prussian war Rene Dragon used photographic shrinking techniques in messages allowing pigeons to carry more information. This idea leads to the modern microdot that consists in images of the size of a printed period. The first detected microdot was taken from a German spy in 1941. As these examples there are more in history, however with the introduction of digital communications most of the previous techniques become obsolete and new forms appear, taking advantage of media data (Johnson, Duric, and Sushil 2001) (Cox et al. 2008).

Nowadays, steganography is mainly used as a form to protect confidential information. Applications like copyright protection, authentication, author identification and edition control, include steganographic methods. However, there are some hazardous applications as terrorism or child pornography. According to USA Today, in 2001 the government of the United States detected some messages hidden in images published in popular websites and even in pornographic ones (Maney 2001). More recently, CNN reported some documents about Al Qaeda plans. The information was confiscated by German authorities in 2011. The terrorist plan was found in diverse digital storage devices containing pornographic contents, which included more than 100 covered documents (Robertson, Cruickshank, and Lister 2012). Due to the increasing unwanted use of steganography, it becomes necessary the design of

methods capable of detecting possible hidden information. In this context, steganalysis is a set of techniques responsible to detect, extract or destroy covered information.

Depending on previous information about media steganographic content, steganalysis could be specific or blind. In specific steganalysis, previous knowledge becomes necessary about the steganographic method used to embed a message. In contrast, blind steganalysis, also called universal steganalysis, must be capable of detecting hidden information without any a priori knowledge of the content or the embedding method (Nissar and Mir 2010). This feature is especially useful when the information under analysis came from an unknown source.

Another issue to consider for selecting a steganalysis method is the media type. Thus analyzing images requires different tools to those used in text analysis. Because of the broad images use to cover information, the effort in this thesis is focused on detecting images with hidden data.

1.2 MOTIVATION

Due to the unwanted uses of steganography, it becomes necessary to take care of the possible side effects. Even more, because of the variety of steganographic methods, it is essential to have updated steganalysis tools, especially those which are independent of the steganographic method. This feature allows the steganalyzer to determine if an image contains a hidden message, without any previous information about the content or the embedding technique. In addition, steganalysis should be reliable for

different steganographic methods in order to allow taking appropriate countermeasures.

1.3 MAIN OBJECTIVE

To develop a reliable blind steganalysis method for color images, capable of detecting the presence of hidden information from diverse steganographic techniques, comparable to state of art.

1.4 SPECIFIC OBJECTIVES

- To integrate a dataset including diverse images embedded at different insertion rates, with different steganographic methods.
- To develop a reliable steganalysis method, using feature extraction and pattern recognition techniques.
- To evaluate the proposed method over the dataset.

1.5 METHODOLOGY

To accomplish the above objectives, the development of our steganalysis method has been planned in four stages, described in Figure 1.1.

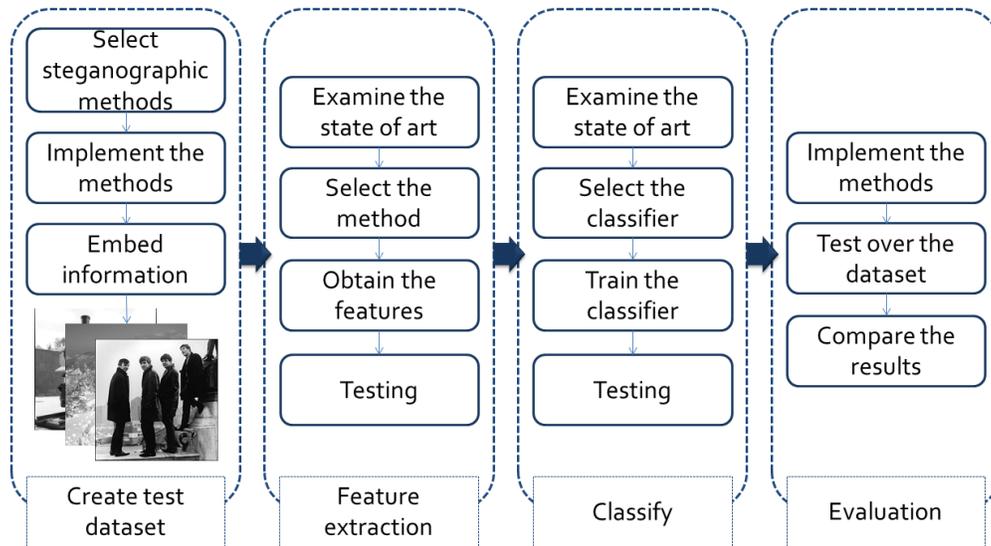


Figure 1.1 Methodology

Due to the lack of a public dataset containing images embedded with different steganographic methods, it is imperative to collect diverse images and embed them with various steganographic methods, considering different insertion rates. This step is essential, because the experimentation and evaluation depend on it.

About the method itself, after a careful analysis of the state of art, we will explore diverse feature sets used for detection. Thus, different feature extractors will be evaluated to determine the most suitable feature set.

Once the features have been chosen, it is necessary to evaluate different pattern recognition techniques. Hence, we are going to explore feature fusion and classifiers ensembles.

Finally the proposed method will be evaluated and compared against state of the art methods.

1.6 THESIS ORGANIZATION

The rest of the thesis is organized in six chapters.

Chapter 2 provides a review of the basic concepts about steganography and its relationship with other disciplines, such as cryptography and information hiding techniques in general, and its main applications. Similarly, we introduce elementary steganalysis concepts, its categorization and the main parts of the steganalysis process.

Chapter 3 includes a review of the steganographic methods state of art, as well as, principal works about steganalysis.

Chapter 4 describes the proposed method, including feature extraction and applied pattern recognition techniques.

Chapter 5 details the experimental setup, which includes dataset and embedding software used. We also describe experiments done for evaluating the proposed steganalysis method and their results. Finally, a results analysis and discussion is presented.

Chapter 6 contains this research conclusions and future work.

CHAPTER 2

BACKGROUND

In this chapter, we introduce steganography basic concepts and its applications. Also, we highlight the difference with cryptography and with watermarking. Additionally, we describe the basis for steganalysis, its categorization and main procedures.

2.1 STEGANOGRAPHY

Steganography is known as the art and science of concealed communication and it is one of the information hiding techniques besides watermarking and fingerprinting. In the hiding process, the traditional scenario (Figure 2.1) consists of three elements: a sender, a recipient and a public channel between them.

The communication is performed as follows. First the sender embeds a message m inside a cover object x using an optional key to provide more security, resulting in a stego object $x(m)$ which is sent through a public channel. On the other side, using the correspondent key, the recipient extracts the hidden message from the stego object (Böhme 2010)(Kharrazi, Sencar, and Memon 2004).

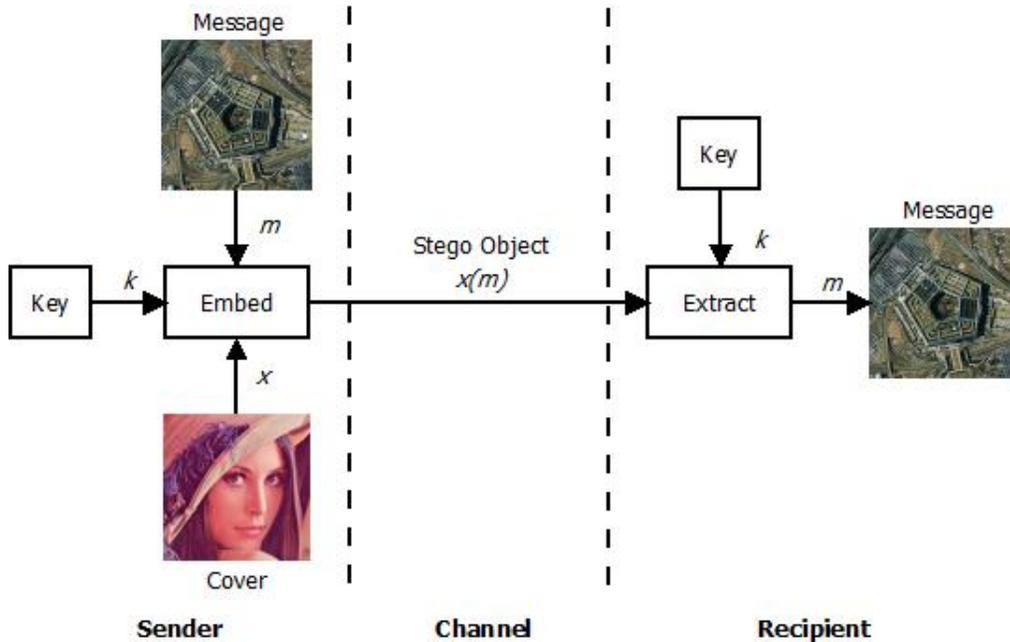


Figure 2.1 *Steganographic traditional scenario*

2.1.1 STEGANOGRAPHY AND CRYPTOGRAPHY

Although steganography came since ancient Greece, the very first work in the digital era was published in 1983 by Gustavus Simmons a cryptographer who introduced the idea behind digital steganography in his article “The Prisoners’ Problem and the Subliminal Channel”. Suppose there are two prisoners plotting an escape, however they are in separated cells, so, the only way to communicate is sending a letter through the warden. If the prisoners use a cryptographic technique in the writing, the warden would notice a suspicious activity and he would interrupt the communication. On the other hand, if the prisoners hide the message about the escape in an innocent message, the warden would not notice it and he would let it go (Simmons 1983).

The prisoners' problem describes the typical steganographic system: the warden is the channel and the prisoners are the sender and the recipient, respectively. The problem also illustrates the difference between cryptography and steganography. Cryptography looks for information confidentiality making communication incomprehensible for unauthorized people. However in some cases this could encourage information attacks. That is the reason why in countries where the cryptography is restricted, steganography has gained popularity.

2.1.2 STEGANOGRAPHY AND INFORMATION HIDING

Information hiding is a general area that includes different embedding techniques, such as steganography, watermarking and fingerprinting. Its main aim is to keep the presence of embedded information secret. To accomplish its aim information hiding techniques must consider three important aspects: capacity, robustness and security. Capacity is the amount of information that can be embedded in the cover object. In the images case, capacity can be measured as bits per pixel (bpp), meanwhile, the measure for video is bits per frame or it could also be bits per second same measures apply for audio. Robustness denotes the technique ability to resist several attacks. For example, in images the recovering process should be able to obtain the covered message even if the stego object suffered changes in contrast, brightness, size, rotation, cropping, among others. Robustness against attacks may differ among algorithmic techniques. Finally, security refers to the inability of detecting the existence of covered information for non-authorized people.

Additionally, information hiding techniques differ among them in some aspects. In steganography, the main aim is to embed a message in a not related cover object, considering high security and capacity. Meanwhile, watermarking strives for robustness and the embedded information is related to the cover object, through its timestamp, author, and/or checksum, among others. On the other hand, fingerprinting also strives for robustness, but hidden information is about the owner or the user of the cover object, making possible to trace any unauthorized transfer (Amin et al. 2003)(Rocha and Goldenstein 2008)(Cox et al. 2008).

2.1.3 STEGANOGRAPHY APPLICATIONS

Steganography applications are quite diverse; however they have been gathered according to their use, such as militia, dissidence or criminal purposes.

As we mentioned above, since ancient Greece to the Second World War, armies had used steganographic techniques to communicate sensitive information to their allies and troops. Nowadays, there is not more information about modern techniques or uses, due to security issues.

In the case of dissident uses, it is well known that in some countries the repression to their citizens includes digital media. To avoid the surveillance and accomplish their cause, in recent time, dissident groups have incorporated steganographic techniques for communicating with their members and mediator organizations, like Amnesty International(Cox et al. 2008).

However, criminal organizations like pedophiles and terrorists have also been interested in steganography capabilities. As a countermeasure to these unwanted uses, steganalysis emerges (Cox et al. 2008).

2.2 STEGANALYSIS

In the prisoners' problem, the warden could prevent the use of an embedded message inside the letter in two ways. He or she could modify the messages deliberately even if they are clean; this is called an *active* warden. Alternatively, he or she could just examine the message and try to determine if it contains a hidden message or not. In this aspect, steganalysis provides techniques required to detect hidden information. Figures 2.2 and 2.3 show schematic explanations of passive and active warden, respectively.

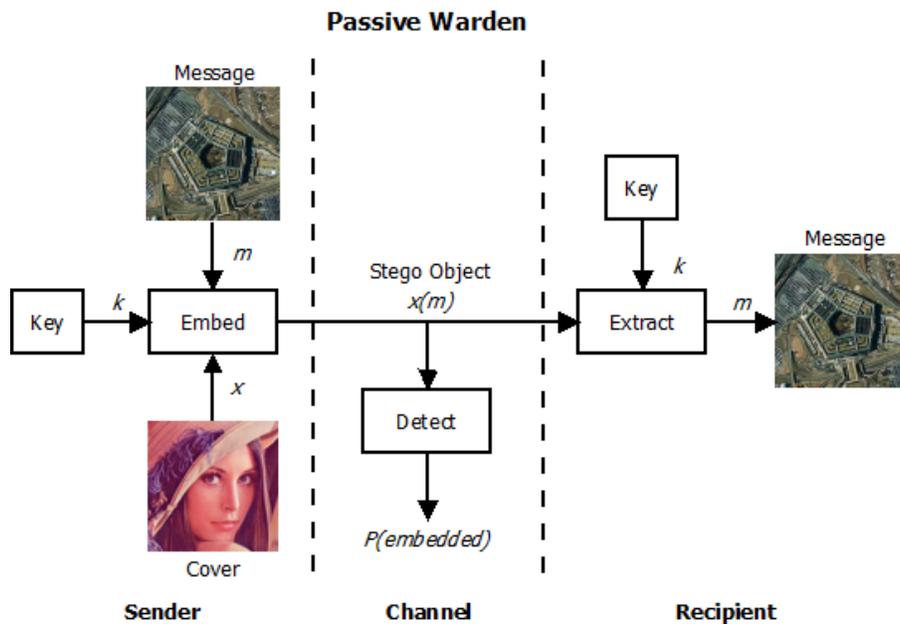


Figure 2.2 *Passive warden scheme*

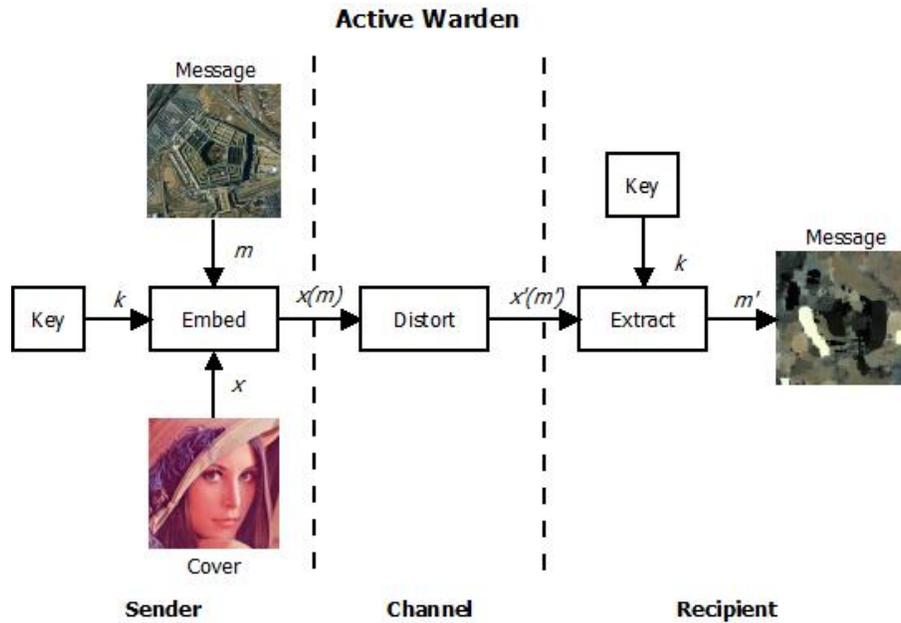


Figure 2.3 Active warden scheme

2.2.1 STEGANALYSIS CATEGORIZATION

There are three general types of steganalysis to determine if a media file contains covered information: visual or aural, structural or by signatures, and statistical (Nissar and Mir 2010)(Rocha and Goldenstein 2008).

- Visual or Aural: The content inspection is made by a human, looking for some anomaly. In images to facilitate the task, different bit planes are displayed separately. This is especially useful for spatial steganographic methods where the covert message is hidden in some specific bit planes, like Least Significant Bit (LSB). Figure 2.4 shows: a) the original greyscale image, b) the least significant bit

plane of the cover image and c) the least significant bit plane of a stego version. As we can appreciate, when an image has been manipulated, the graph of the LSB plane c) has a notable distortion compared with the clean one. However, for more advanced techniques or complex images, it is not possible to detect anomalies at plain sight.

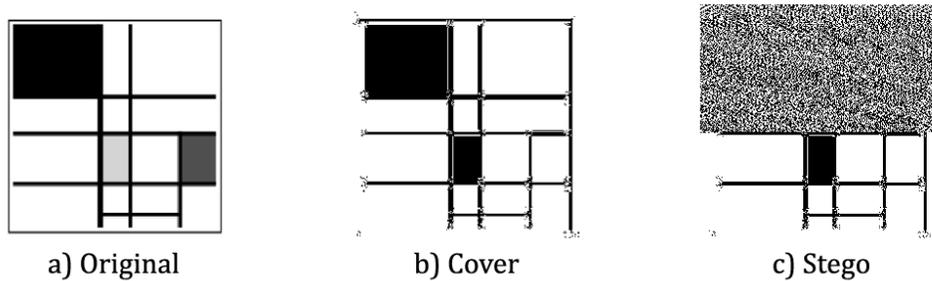


Figure 2.4 *Visual based steganalysis*

- Structural or by signature: In the inserting process, some steganographic techniques alter the properties of the media file. This modification may introduce some characteristic patterns, acting as a signature. A steganalyst will search for repetitive patterns in the file or in the structure of the media, depending on the used steganographic method.
- Statistical: Hiding information in images may lead to an alteration to their natural statistics. With statistical analysis it is possible to determine if an image has been altered. This is the most common steganalysis type, due to its capacity and sensitivity.

In general, there are two main approaches for steganalysis: specific and universal or blind, see Figure 2.5.

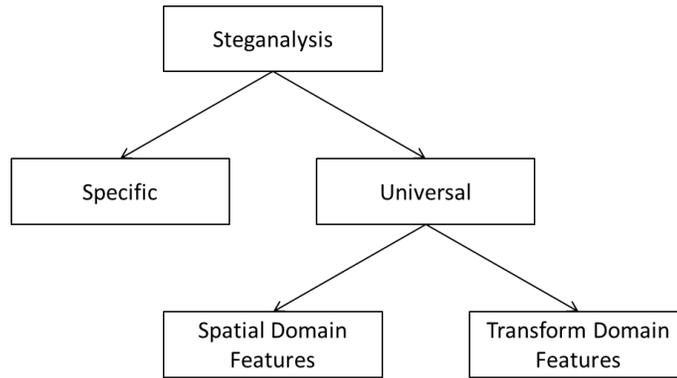


Figure 2.5 *Steganalysis categorization*

In a specific approach, the steganalyst knows the functioning and the properties of the steganographic technique used to embed the information. Usually, specific techniques look up for particular distortions. These steganalysis algorithms could be used with other steganographic methods; however, many times they cannot detect successfully the embedded message.

On the other hand, a universal approach, also called blind, must be capable of recognizing a stego image no matter which method was used for insertion. In practice, universal techniques provide a better tool for detection, but they are not reliable for every steganographic method, especially new ones (Kharrazi, Sencar, and Memon 2004)(Nissar and Mir 2010).

2.2.2 STEGANALYSIS PROCESS

Today, most of steganalysis methods use a standard process in order to detect hidden information (Figure 2.6). First, a feature extraction procedure is performed with the purpose of having essential information to determine if an image contains or not hidden data, but with manageable dimensionality. Second, resulting features vector is used as input for a classifier method, which after building a model should be capable to predict the image class (stego or cover). Feature extraction is detailed in the next section.

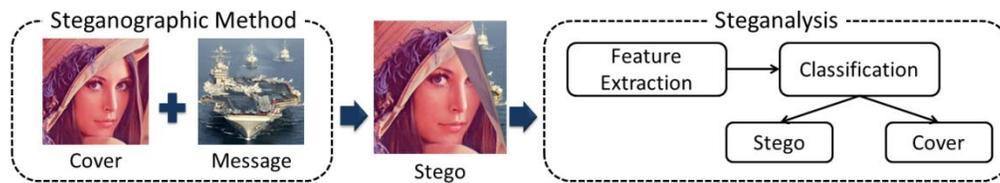


Figure 2.6 Steganalysis process

2.2.2.1 FEATURE EXTRACTION

In steganalysis, feature extraction process could be in the spatial or the transform domain.

Spatial features usually focus on pixel relationships. Diverse authors support the idea of natural images having certain relations within pixel neighborhoods and this relation is disrupted in an embedding process. For this reason, spatial features use models of pixel neighborhoods either based on textures, differences, transitions or interpolation, among others. However, calculating these models could be a difficult task due to

dimensionality. One way to reduce the amount of data consists in selecting pixels by steganalytic significance according to a previous analysis of hidden data behavior. The selection is ruled by a threshold which is specified by the feature extraction method, depending on tests result for diverse steganographic techniques. Another way to reduce dimensionality is using a small data representation, like histograms or statistical measures. Due to its simplicity, spatial features are popular for blind or universal steganalysis (Pevný, Bas, and Fridrich 2010)(Guan, Dong, and Tan 2011a)(Fridrich and Kodovsky 2012)(Lafferty and Ahmed 2004).

On the other hand, transform domain features change spatial information into wavelets, DCT, among others. To convert an image into a transform-domain form, it is divided in $N \times N$ blocks, where $N = 8$ most of the time. Then, each block is computed using a wavelet or DCT. Due to the resulting coefficients have the same dimensionality than original image; a final step is required to obtain a feature set. Here, some authors propose using statistical moments as feature sets (Shi et al. 2005)(Hui, Ziwen, and Zhiping 2011), transition probability between coefficients(Chen and Shi 2008), among other techniques.

Recently, to improve detection rate, various authors fuse both spatial and transform domain features, in order to take advantage of both types (Rodriguez, Peterson, and Bauer 2008).

2.2.2.2 CLASSIFICATION

Following the feature extraction process, it becomes necessary a classification procedure, in order to determine the image class (cover or

stego). However, classification task is usually left aside by steganalysis methods authors, which focus their efforts in the feature extraction process. For this reason, most steganalysis methods use Support Vector Machine (SVM) or Neural Networks as classifiers (Pevný, Bas, and Fridrich 2010), (Shi et al. 2005), (Lafferty and Ahmed 2004), (Guan, Dong, and Tan 2011a), (Hui, Ziwen, and Zhiping 2011), (Arivazhagan, Jebarani, and Shanmugaraj 2011), (Niimi and Noda 2011). However in recent years, certain authors started to improve the classification process using ensemble of classifiers (Bayram et al. 2010), (Kodovsky, Fridrich, and Holub 2012) or fusion of steganalysis systems (Rodriguez, Peterson, and Bauer 2008), (Sun, Liu, and Ji 2011).

CHAPTER 3

STATE OF ART

This chapter includes a review of some of the most used steganographic methods for images. Also, we include the state of art of steganalysis methods.

3.1 STEGANOGRAPHIC METHODS

Steganographic system design has evolved through the years in order to keep the embedded data unnoticeable. However, selecting a steganographic method depends on the end user requirements, such as capacity, security, complexity, among others. For this reason, there is not a unique steganographic method that can fulfill all the requirements.

Usually, the steganographic methods are classified according to the domain in which the data is embedded. In the spatial domain, the most popular method is Least Significant Bit (LSB), but its well-known weakness against visual and statistical attacks makes necessary to develop other methods.

Below, there is a further explanation of the most representative steganographic methods.

3.1.1 LEAST SIGNIFICANT BIT (LSB) FAMILY

Least Significant Bit (LSB) is the steganographic technique most widely used due to its simplicity. LSB takes advantage of the inability of the human eye to perceive small changes in the pixels of an image. The embedding process is carried out in the spatial domain, by replacing the least significant bit of selected pixels by message bits. The substitution could be either successive or pseudo-random. In the successive substitution, each pixel of the cover image is modified in the same order than the embedded bits. Meanwhile, pseudo-random substitution uses a key as seed for a pseudo-random number generator, where each number specifies a pixel to be modified. Despite this kind of embedding provides some security, in general, the LSB embedding could be easily destroyed, by almost any image modification (Rocha and Goldenstein 2008) (Chanu, Tuithung, and Manglem Singh 2012).

3.1.1.1 STEGHIDE

A well-known LSB implementation for images and audio is Steghide. It slightly modifies the original algorithm by adding a graph to reduce the amount of pixel modifications. Before embedding the message, it is encrypted and compressed to increase security. After that, a pseudo-random numeric sequence is produced from the passphrase as seed. This sequence belongs to the cover pixels, whose LSB will contain a bit of the message. To improve imperceptibility, the LSB that differs from the bit to embed is considered for exchanging for other LSB that matches with it. This is ruled by a graph where each vertex represents a change and each

edge is a possible exchange. Finally after the exchange, the remaining message bits are embedded replacing the corresponding LSB (Hetzl and Mutzel 2005) (Hetzl 2002).

3.1.1.2 JPHIDE AND JPSEEK

Another implementation based on LSB is JPHide and JPSeek, JPHide for embedding and JPSeek for extracting. Instead of modifying LSB pixels, JPHide uses non-zero quantized DCT coefficients. With the passphrase as seed, a pseudo-random number is initialized and used as sequence for insertion. Each message bit is embedded in the least significant bit of the selected non-zero quantized DCT coefficients. Additionally, JPHide permits the embedding in the second least significant bit (Li 2011)(Latham 1999).

3.1.1.3 LSB MATCHING

Through the years, LSB has evolved in several methods, developed in order to improve its imperceptibility. One of them is LSB matching, also called ± 1 embedding. This technique tries to prevent basic statistical steganalysis. In LSB substitution, odd values are decreased or kept unmodified, while even values are increased or kept unmodified. On the contrary, LSB matching randomizes the sign for each instance, so a half of will be increased by one and the other half will be decreased by one (Böhme 2010).

3.1.1.4 LSB MATCHING REVISITED

Another modification of the LSB method is LSB matching revisited (LSBMR). LSBMR uses pixels pairs as embedding unit, where each pixel contains a bit of the message. To embed a pair of bits, a binary function is used, such as increment or decrement. With this technique, the probability of modifications per pixel is 0.375 against 0.5 of LSB, for 1bpp embedding rate (Mielikainen 2006).

3.1.1.5 EDGE ADAPTIVE LSB MATCHING REVISITED

One of the most recent variants of LSB is the Edge Adaptive LSBMR. This technique uses the same concept of pixel pairs; however, the embedding process is carried out by regions. First, the image is divided in random size blocks. Later, a random rotation is applied to the block, in order to improve security. Once the image is divided into blocks, the pixel pairs in the threshold are considered as embedding units. Finally, a binary function is used for embedding (Luo, Huang, and Huang 2010).

3.1.2 MODEL-BASED

Most of steganalysis algorithms exploit the inability of steganographic methods to preserve the natural statistics of an image. For this reason, Sallee (Sallee 2004) proposed a model-based steganography algorithm, which preserves not only the distribution of an image, but the distribution of its coefficients as well.

Before embedding, the image is divided in two parts, x_α which will remain unaltered and x_β where message bits will be inserted. For JPG, x_α could be the most significant bits of the DCT coefficients and x_β the least significant bit. Then, using the conditional probability $P(x_\beta|x_\alpha)$, it is possible to estimate the distribution of the x_β values. Afterward, a x'_β is generated with the message bits using an entropy decoder according to the model $P(x_\beta|x_\alpha)$. Finally, the stego object is assembled with x_α and x'_β .

3.1.3 F5 STEGANOGRAPHY

F5 is a transform domain embedding algorithm for JPG, proposed by Westfeld (Westfeld 2001). The embedding process (Figure 3.1) is developed during JPG compression. First, the password initializes a pseudo-random generator, which is used for permuting DCT coefficients. Second, based on matrix encoding, message bits are inserted in the selected coefficients. To accomplish this, the coefficients are considered as a code word a with n changeable bits for k message bits of x . The amount of coefficients needed for embedding is equal to $2^k - 1$. Then, with k bits taken from a and using a hash function, the bits of x are inserted with the XOR operation, one by one. After each insertion, if the sum is not 0, then this result is the index of the coefficient that must be changed and its value is decremented; else the code word remains unaffected. Finally, the permutation is reverted and the JPG compression continues.

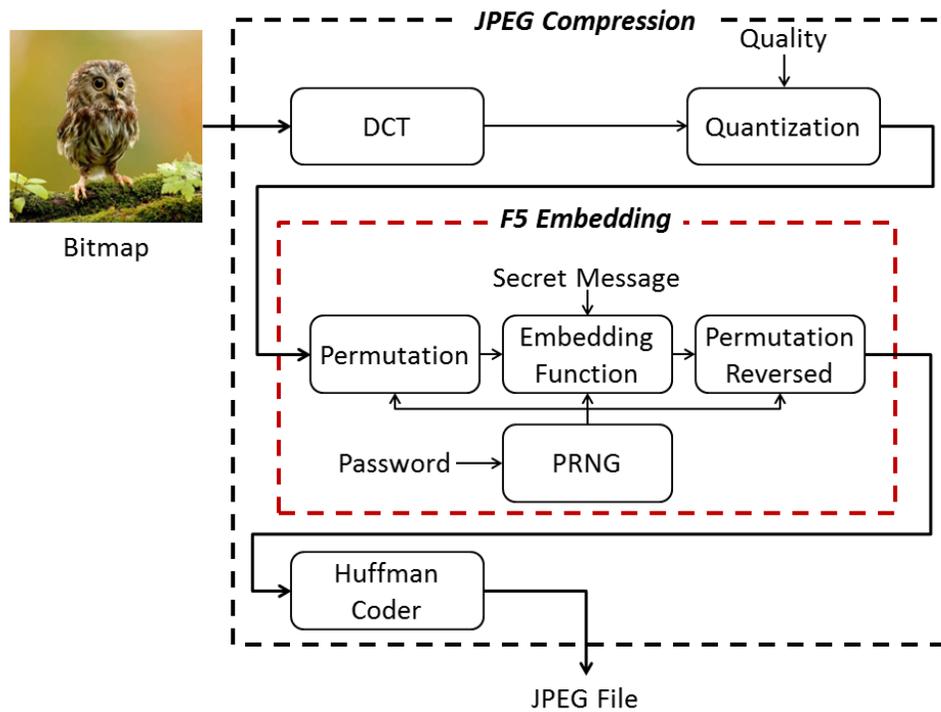


Figure 3.1 F5 embedding process

3.1.4 SPREAD SPECTRUM

Spread spectrum emerges for securing military communications in order to reduce signal jamming (an attempt to inhibit communication between two or more parts) and interruptions. An example of a spread spectrum technique in telecommunications is the frequency-hopping, where a message is divided and sent through different frequencies controlled by a key. In images, the first spread spectrum technique was proposed by Cox in 1997. Before insertion, the message is modulated as an independent and identically distributed Gaussian sequence, with $\mu = 0$ and $\sigma^2 = 1$. After, the resulting sequence is embedded in the most significant

coefficients of the DCT. The clean image is necessary to extract the message (Cox et al. 2008)(Maity et al. 2012).

3.1.5 OTHER STEGANOGRAPHIC METHODS

The Bit Plane Complexity Segmentation (BPCS), proposed by Kawaguchi and Eason in 1998 (Kawaguchi and Eason 1998), allows adaptive embedding in multiple bit planes, by searching for noise-like blocks.

In 2003, Fridrich and Goljan (Fridrich and Goljan 2003) developed the Stochastic Modulation Steganography, where the embedding data is inserted as a weak noise signal.

In 2005, Zhang and Wang (Zhang and Wang 2005) introduced the Multiple Base Notational System (MBNS), where the message bits are converted to symbols in a notational system with variable bases that depend on local variation.

About the transform domain, most of the methods are specialized for JPEG embedding, due to its popularity. Like Outguess, proposed by Niels Provos in 2001 (Provos 2001), where the message bits are embedded in the LSB of the quantized DCT (Discrete Cosine Transform) coefficients; after the insertion, the unmodified coefficients are corrected to maintain the statistics of the original image.

Another method for JPEG is Yet Another Steganographic Scheme (YASS), developed by Solanki, Sarkar and Manjunath in 2007 (Solanki, Sarkar, and Manjunath 2007). Before insertion, the image is divided in B-blocks larger

than 8x8. Inside each B-block an 8x8 H-block is randomly placed. Message bits are inserted in the DCT coefficients of each H-block.

3.2 STEGANALYSIS METHODS

As we mention in Chapter 2, standard steganalysis process consists in two main procedures: feature extraction and classification. However, most of the steganalysis methods focus their efforts in the feature extraction. For this reason, the following review of the state of art is mainly based on the feature extraction of each steganalysis method. Figure 3.2 shows some steganalysis methods described in this chapter.

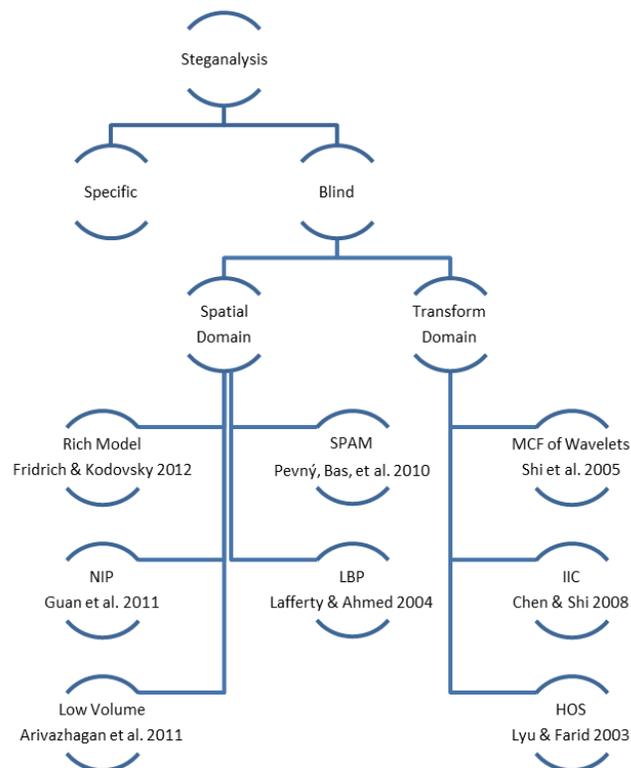


Figure 3.2 *Some steganalysis methods of the state of the art*

3.2.1 SUBTRACTIVE PIXEL ADJACENCY MODEL (SPAM)

SPAM (Pevný, Bas, and Fridrich 2010) is a feature extraction method for images, proposed by Pevny, Bras and Fridrich in 2011. It works in the spatial domain, where initially, the differences between the pixels in eight directions are calculated (\leftarrow , \rightarrow , \uparrow , \downarrow , \nwarrow , \nearrow , \swarrow , \searrow). For example, the horizontal differences are calculated by $H1_{i,j} = I_{i,j} - I_{i,j+1}$ and $H2_{i,j} = I_{i,j} - I_{i,j-1}$, where I is the image represented as a pixel values matrix, and $i \in \{1 \dots m\}, j \in \{1 \dots n\}$.

Subsequently, it is set a threshold $[-T, T]$ to every difference result in order to reduce dimensionality and processing time. Thus, transition probability matrices for every direction are calculated between difference result pairs for first order or triplets for second order. The authors propose $T = 4$ for first order and $T = 3$ for second order because they are more relevant for the steganalysis.

Finally, the average of the four horizontal and vertical matrices is calculated to obtain the first half of the features. The four diagonal matrices are averaged to complete the features.

3.2.2 LOCAL BINARY PATTERN (LBP)

In order to be unnoticed for the human eye, some steganographic methods use noise-like areas in the image for embedding, such as textures and edges. Taking into account this premise, the operator LBP is used as a feature extractor method based on texture modeling. Originally, LBP was proposed for measuring the texture of an image. LBP was first mentioned

by Harwood (Harwood et al. 1995) and formalized by Ojala (Ojala, Pietikäinen, and Harwood 1996). But, it was not until 2004, that Lafferty and Ahmed (Lafferty and Ahmed 2004) developed a feature extractor for steganalysis based on LBP.

The LBP process for an image I is as follows. For each pixel p a local binary pattern value is calculated, which combines the values of the eight pixels around p . Let x_i be a pixel in the 3×3 neighborhood, with $i \in \{1 \dots 8\}$, and $x'_i = 0$ if $x_i < p$, else $x'_i = 1$ if $x_i \geq p$. Then $p' = \sum x'_i * 2^i$. Figure 3.3 shows an example of LBP value calculation.

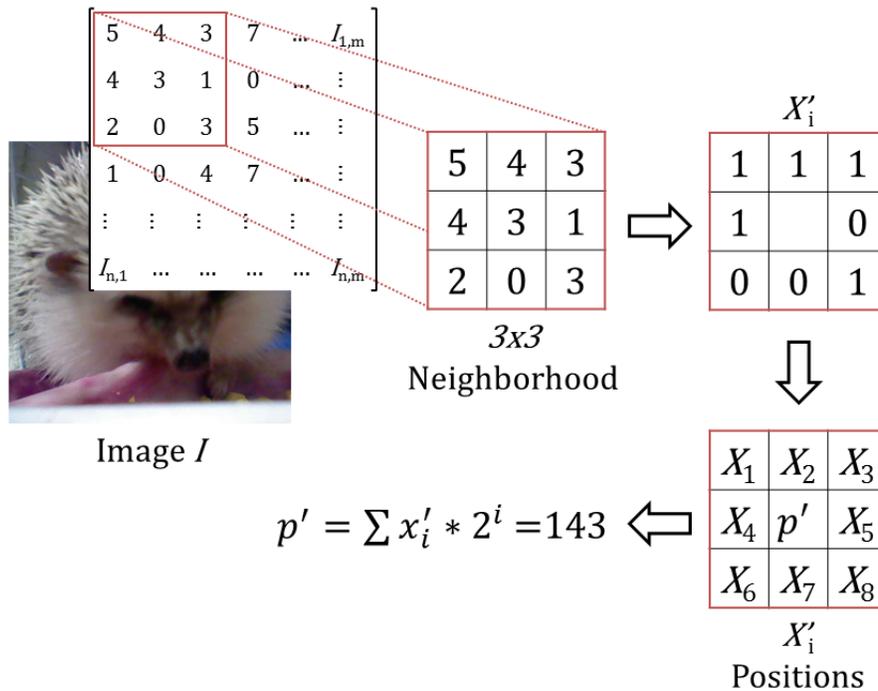


Figure 3.3 Example of LBP value calculation

Finally, the LBP values are represented as a 256-bin histogram. The features used in (Lafferty and Ahmed 2004) are the standard deviation, variance, and mean of the final histogram.

3.2.3 INTRABLOCK AND INTERBLOCK CORRELATIONS (IIC)

Natural images usually keep a correlation between the coefficients of a 8×8 DCT, both intrablock and interblock (Figure 3.4). In order to detect any irregularities in these correlations, in 2008, Chen and Shi (Chen and Shi 2008) proposed a feature extractor for steganalysis based on a markov process that takes into account relations between neighbors (intrablock) and frequency characteristics (interblock). To determine intrablock correlations, the DCT coefficients of an 8×8 block are used to generate four difference matrices: horizontal, vertical, main diagonal and minor diagonal. After, a transition probability matrix is calculated for each difference matrix. In order to reduce the complexity, a threshold t is established; any value larger than t , or smaller than $-t$, will be replaced by t or $-t$ respectively.

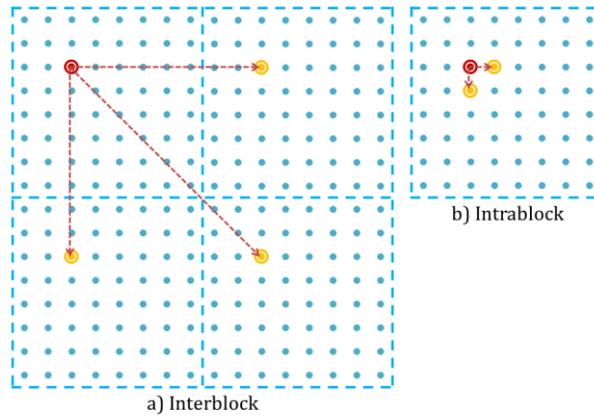


Figure 3.4 Interblock and intrablock correlation

- a) Interblock correlations between coefficients in the same position within 8x8 blocks.
- b) Intrablock correlations with the neighbor coefficients within an 8x8 block

Interblock correlations are computed between coefficients in the same position within the 8x8 blocks. First, for each position in the DCT coefficient (except the first one) an alignment is necessary (Figure 3.5). Then, the resulting matrices are processed as in the intrablock calculation.

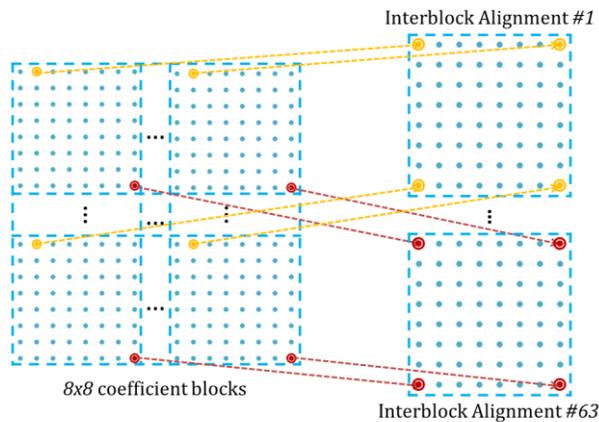


Figure 3.5 Interblocking alignment

3.2.4 HIGHER ORDER STATISTICS (HOS)

This feature extractor, proposed by Farid and Lyu in 2003 (Lyu and Farid 2003), tries to expose statistical distortions by the decomposition of the image in orientation and scale. The feature extraction is divided in two parts.

First, the image is decomposed using Quadrature Mirror Filters (QMF), which are formed by lowpass and highpass filters. The filters are applied along vertical, horizontal, and diagonal directions. In order to increase the detection rate, the features are calculated in different scales. These scales are obtained with a lowpass subband filter, which is recursively filtered along vertical, horizontal, and diagonal directions (Figure 3.6). For all the resulting subbands, the mean, variance, skewness and kurtosis are calculated.

Second, a linear error predictor is applied for vertical, horizontal and diagonal subbands in each scale, taking into account the neighbors values. For the resulting models the mean, variance, skewness and kurtosis are also calculated.

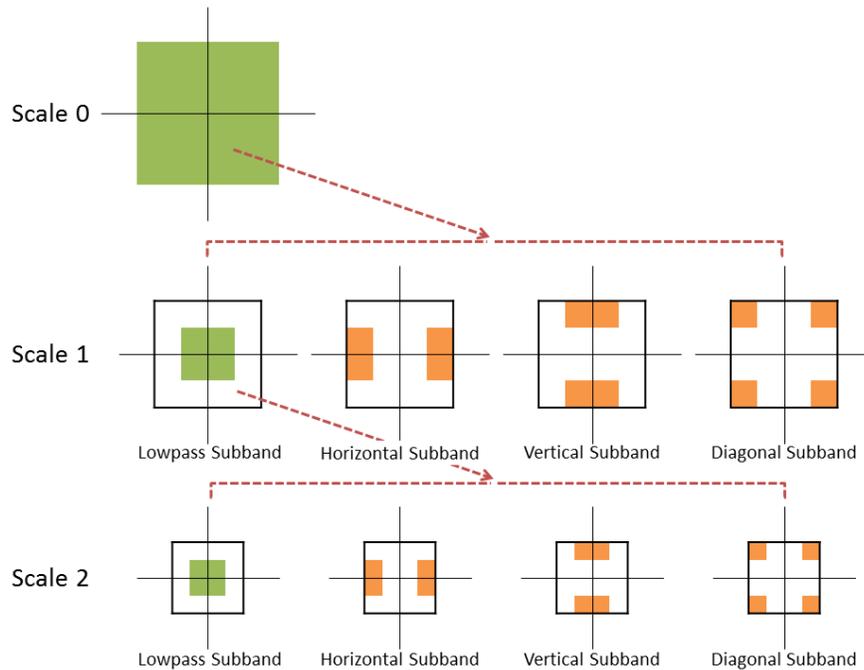


Figure 3.6 Multi-scale lowpass subband, horizontal, vertical and diagonal

3.2.5 OTHER STEGANALYSIS METHODS

One of the most recent methods in the spatial domain is the rich model proposed in 2012 by Fridrich and Kodovsky (Fridrich and Kodovsky 2012), where different pixel dependency sub models are used as features. Using diverse types of sub models makes it possible to capture different embedding artifacts; however, the dimensionality increases substantially. For classification, they use an ensemble of classifiers.

In 2010, Guan, Dong and Tan (Guan et al. 2011) proposed a spatial domain method called Neighborhood Information of Pixels (NIP), in which, the differences between neighbor pixels and the center of the

neighborhood are calculated and subsequently codified using invariant rotation. The result is processed as histogram, removing empty values.

In 2011, Arivazhagan, Jebarani and Shanmugaraj (Arivazhagan, Jebarani, and Shanmugaraj 2011) used 4x4 segments where pixel differences are calculated according to nine paths within the neighborhood. The results between -4 and 4 are placed within a co-occurrence matrix and are used as feature vectors.

In the transform domain, spatial data are usually changed by wavelets or DCT, For example, in 2005, Shi et al. (Shi et al. 2005), proposed the use of first, second and third order Haar wavelet, calculating the moments of each transform divided into four sub-bands. Finally, three statistic moments are calculated from each sub-band and used as features for a neural network.

Some authors complement the results of both domains using fusion of features or fusion of classifiers with different features. Rodríguez, Bauer and Peterson (Rodríguez, Peterson, and Bauer 2008) in 2008 fuse wavelet and cosine features with a Bayesian Model Averaging, which merges multi-class classifiers. In 2010, Bayram, Sencar and Memon (Bayram et al. 2010) ensemble different binary classifiers with AdaBoost; using Binary Similarity Measure (BSM), Wavelet Based Steganalysis (WBS), Feature Based steganalysis (FBS), Merged DCT and Markov Features (MRG) and Joint Density Features (JDS) as feature extractors. In 2011 Guan, Dong and Tan (Guan, Dong, and Tan 2011b) merged the results of feature extractors like Markov feature, PEV-247D and differential calibrated Markov feature. Afterwards features are fused by subspace method and classified with

gradient boosting. More recently, in 2012, Kodovsky and Fridrich (Kodovsky, Fridrich, and Holub 2012) used random forest as an ensemble of classifiers; to address the problems of dimensionality and number of instances of regular classifiers.

3.3 SUMMARY AND DISCUSSION

Since the steganography became a popular way to protect sensitive information against unauthorized people, the creation of steganographic methods has increased, leading to a great variety of them. With this availability of embedding methods, users are capable to find a method that fulfills their requirements, in capacity, robustness and security. In order to provide a general outlook of the recent steganographic development, in this Chapter, we include a review of the most representative steganographic methods.

Sadly, the unwanted uses of the steganography have also grown. To countermeasure its negative effects, steganalyzers have focused their efforts on developing new and better steganalysis methods. However, this has not been an easy task, due to the great variety of embedding techniques. In this context, steganographic methods development is divided in two main approaches: specific and universal.

Specific steganalysis requires previous knowledge of the steganographic method under analysis; this type of methods usually have good detection rate. On the other hand, universal or blind steganalysis works for a variety of steganographic methods, but frequently they have lower detection rates

than the specific ones. To accomplish their aim, universal methods typically center their design in the feature extraction process, leaving aside the classification procedure. Taking this opportunity into account, this research looks for an enhanced universal steganalysis method, improving both processes.

CHAPTER 4

PROPOSED METHOD

4.1 PROPOSED METHOD

The contribution to the state of art in this thesis consists of a blind steganalysis method for color images based on multiple feature extractors and a meta-classifier. The decision of developing a steganalysis method for color images was taken because most of the images on the Internet are in color or they could be easily transformed into a RGB image; additionally, most of the steganographic software use only color images in order to increase insertion capacity.

The proposed method was designed taking into account state of the art experience. Some authors (Rodriguez, Peterson, and Bauer 2008)(Bayram et al. 2010)(Guan, Dong, and Tan 2011b) recently started to combine feature sets in order to increase detection rate. This is because using different feature sets could complement each other, detecting more steganographic data. Besides, in order to improve detection rate and make the design scalable, it is proposed a meta-classifier rather than a simple classifier scheme.

The proposed method (Figure 4.1) consists of three stages: Feature Extraction, First Level Classification and Second Level Classification.

In the first stage, four feature sets are obtained from each image. Here, we use four previously proposed feature extractors with some modifications (detailed in section 4.2): Local Binary Pattern (LBP), Subtractive Pixel Adjacency Model (SPAM) (Pevný, Bas, and Fridrich 2010), Intrablock and Interblock Correlations (IIC) (Chen and Shi 2008), and Higher Order Statistics (HOS) (Lyu and Farid 2003). In section 4.2 the feature extraction process is detailed.

Next in the second stage, resulting feature sets from previous stage are used for supervised learning. Independently, each feature set is used for building two different binary classification models; one based on logistic regression and one based on random forest. The output of this stage is the predicted class (stego or cover image) of an image for the eight classifiers.

In the final stage, the resulting classes of the previous classifiers are used as features for logistic regression classification. Section 4.3 contains details of the classification process.

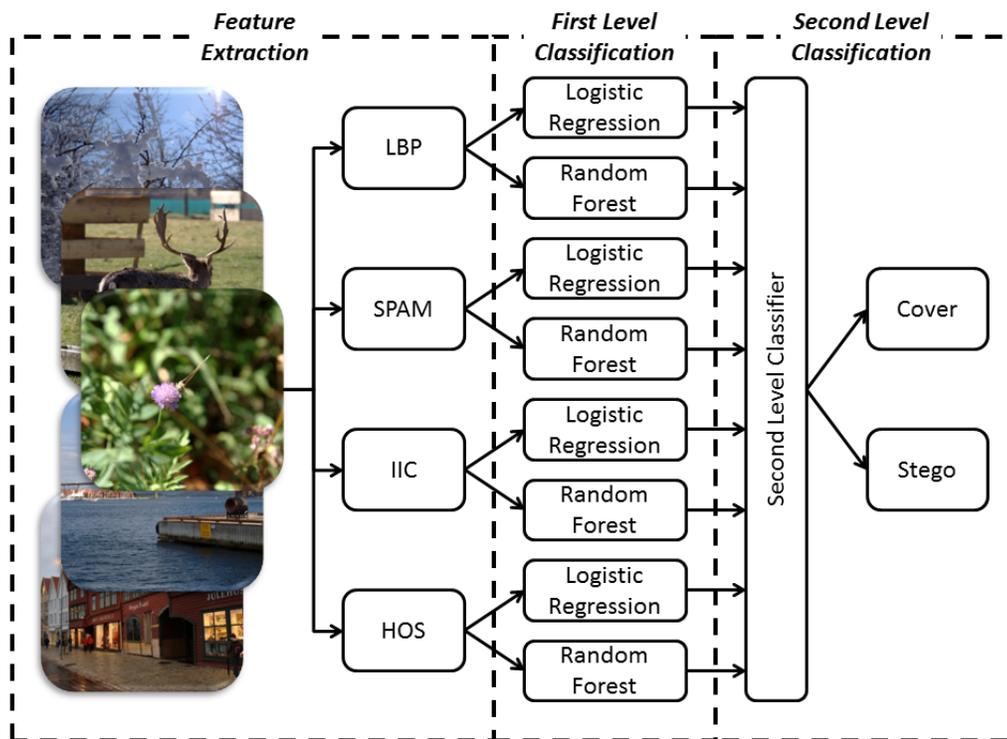


Figure 4.1 Proposed method

4.2 FEATURE EXTRACTION

In order to accomplish the objectives, we choose four feature extractors: Subtractive Pixel Adjacency Model (SPAM) (Pevný, Bas, and Fridrich 2010), Local Binary Pattern (LBP), Intrablock and Interblock Correlations (IIC) (Chen and Shi 2008), and Higher Order Statistics (HOS) (Lyu and Farid 2003). The algorithm selection was made based on diverse aspects. First, features should be extracted in different domains; thus, stego images that are not detected in the spatial domain could be recognized in the transform domain and vice versa. Second, dimensionality should be manageable. For example, high dimensionality of the rich model in (Fridrich and Kodovsky

2012) (34,761 features for the entire model) makes it impractical for a scenario with huge amount of images. Another desirable aspect is the algorithmic reproducibility or code availability; since, in some cases, authors omit relevant information, making impossible to reproduce the algorithm.

Below, we detail the modifications made to SPAM and LBP algorithms, with the purpose of improving LBP detection rate and making SPAM suitable for color images. For Intrablock and Interblock Correlations and Higher Order Statistics, we keep the original algorithm described in Chapter 3.

4.1.1 SUBTRACTIVE PIXEL ADJACENCY MODEL

For our method, we adapted the original second order SPAM algorithm to take into account the information of the RGB channels, in order to make it suitable for color images. First the differences along eight directions are calculated for each color channel. For transition probability calculation, the values of the differences within a threshold $T, -T$, where $T = 3$, are summarized in two different arrays; a frequency array from $-T$ to T containing the incidences and a co-occurrence array from $-T, -T, -T$ to T, T, T with the frequency of threshold values triplets. Later, the results for each channel are summed in a unique frequency and co-occurrence array. Next, the probability of each triplet is calculated. Finally, the features are calculated in two parts: the average of horizontal and vertical directions and the average of the four diagonals; resulting in a feature set with $(2T + 1)^3$ features. Figure 4.2 shows the SPAM process.

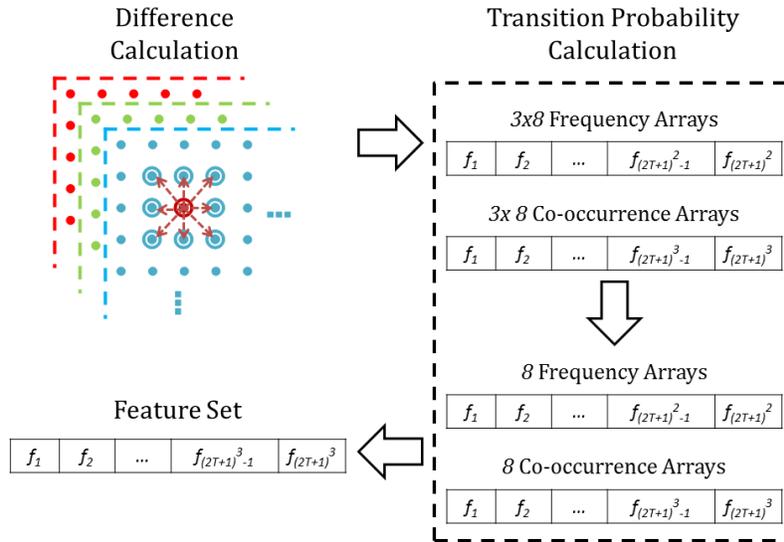


Figure 4.2 SPAM process

4.1.2 LOCAL BINARY PATTERN

The proposed change to the LBP algorithm is the final extraction of the feature set. After some tests, we found out that the statistics of the LBP values histogram, as the feature set proposed in (Lafferty and Ahmed 2004), produce lower detection rates than using the whole histogram.

The LBP algorithm used in our method is defined as follows (Figure 4.3). After LBP values calculation for each color channel, a global histogram is obtained. This histogram is used as feature set.

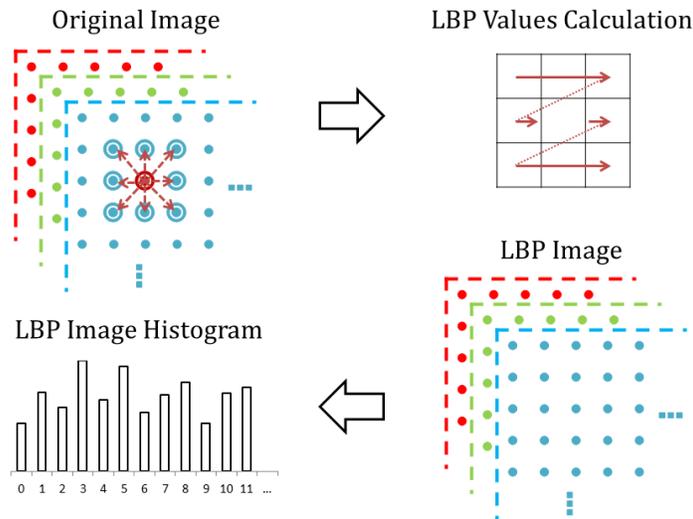


Figure 4.3 LBP Process

4.3 CLASSIFICATION

Most steganalysis methods in the state of the art usually focus their efforts on improving the feature extraction process, leaving aside the classification stage. Thus, classifiers like Support Vector Machines (SVM) or Neural Networks are commonly used. However, this may not provide the best detection rate. More recently, some authors have proposed the use of classifier ensembles to improve accuracy (Rodriguez, Peterson, and Bauer 2008)(Bayram et al. 2010)(Kodovsky, Fridrich, and Holub 2012). In this context, we propose a meta-classifier based on Logistic Regression and Random Forest. The selection of these classifiers was made based on accuracy and training time, due to the great amount of data to process. For instance, classifiers such as Multilayer Perceptron are reliable, but the training time makes them infeasible for our purpose. Thus, after some tests

Logistic Regression and Random Forest showed to fit best our problem, in time and accuracy.

Logistic regression is a probabilistic discriminative model that uses the conditional distribution between two variables $P(y|X)$ where X is the feature set and y is the class of the object. In binary problems, y could be 0 or 1, in our case, 1 = *stego* and 0 = *cover*. To predict the class of an object
(1)
A logistic function is given by:

$$P = \frac{1}{1 + e^{-\beta_0 + \beta_1 X}}$$

In binary problems, the probability of $y_i = 1$ or $y_i = 0$, in our case 1=*stego* and 0=*cover*, is calculated using the logistic function with X_i as the features of every image i . β values are obtained based on training data, commonly by maximum likelihood (Bishop 2006).

Alternatively, random forest is an ensemble classifier, composed by several decision trees. The training of a random forest is as follows. First, different random subsets are taken from the feature set. Then, for each feature subset a decision tree is built. The nodes of the decision tree are iteratively chosen from a small set of input variables; here, according to an objective function, the variable that provides the best split is set in the node. For testing, each instance is evaluated by all decision trees. The result could be an average or a voting of results from individual decision trees (Breiman 2001).

In our method, these classifiers are combined to build a robust classifier of two levels. Where the feature sets given by the four selected extractors

are used to build logistic regression and random forest classifiers. The resulting predictions for every instance are recorded in eight dimensional vectors. These vectors plus the real label are used to build a new classifier. Figure 4.4 shows the classification procedure proposed in this thesis.

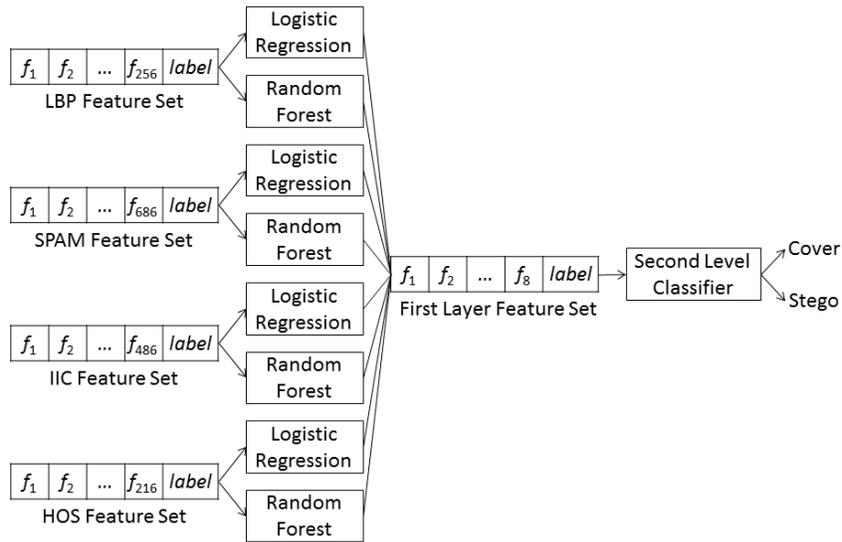


Figure 4.4 Proposed classification method

4.4 CHAPTER SUMMARY

This chapter details our steganalysis method for color images, which consists of three stages. For the first stage we selected four feature extractors: SPAM(Pevný, Bas, and Fridrich 2010), LBP(Lafferty and Ahmed 2004), IIC(Chen and Shi 2008) and HOS(Lyu and Farid 2003). The first two extractors were modified to improve their detection rate. In the second stage we used two well-known classifier algorithms: Logistic

Regression(Cessie and Houwelingen 1992) and Random Forest(Breiman 2001). Prediction results from these classifiers are the input for the last stage: a Logistic Regression classifier.

For the purpose of this thesis, the proposed method uses four feature extractors; however, this number could increase or decrease according to practical requirements. The flexibility of the method to add or to replace feature extractors is an attractive characteristic accomplished by the proposed classification process. This allows the proposed method to adapt to other steganographic methods, achieving universality.

CHAPTER 5

EXPERIMENTS AND RESULTS

In this chapter, we describe the dataset used for experiments, images type, the settings of the steganographic methods used for embedding, and the settings for classification. Also, we explain the experiments carried out to show the performance of the proposed method and the obtained results. Finally, there is an analysis and discussion of these results.

5.1 EXPERIMENTAL SETUP

5.1.1 DATASET

A difficulty for testing new steganalysis methods is the lack of a standard image dataset, restricting a fair comparison with the state of the art. Another problem about the selection of images is ensuring the total absence of a watermark or stego data. In this context, some authors of steganographic systems have published their datasets. Commonly datasets are from contests BOWS in 2006 (Break Our Watermarking System)(Barni, Voloshynovskiy, and Perez-Gonzalez 2005), BOWS2 in 2008 (Break Our Watermarking System 2)(Bas and Furon 2007) and BOSS in 2010 (Break Our Steganographic System)(Pevný, Filler, and Bas 2010) base. In this

thesis, we use images provided by authors of the BOSS Base, due to the availability of raw images directly from cameras. Figure 5.1 shows some examples of the dataset content.



Figure 5.1 *Example of images from the dataset*

The raw dataset contains 10,000 high resolution images from different cameras. These images were converted to 512x512 RGB JPEG without compression, using the convert command of the ImageMagick library in linux. For practical purposes, each image is labeled from 1 to 10,000. This allows generating a different secret message for each image, using their label as a key of a pseudo random number generator. Then, each image was embedded with 164(0.005bpp), 328(0.01bpp) and 1,638(0.05bpp) bytes. The steganographic methods used for embedding are: F5, Steghide, Jphide, Spread Spectrum, LSB Matching Revisited, EALSBMR and Model Based. The following section contains the details of the embedding software used.

5.1.2 EMBEDDING SOFTWARE

The selection of the steganographic methods used in the experiments was made based upon embedding software availability and serial embedding capacity, because of the amount of images. Another important aspect for consideration was the method popularity, either in spatial or transform domain. The selected methods were: F5, Steghide, Jphide, Spread Spectrum, LSB Matching Revisited, EALSBMR and Model Based. Table 5.1 shows a review of the steganographic methods used in the experiments; this includes the embedding domain, the changes distribution within the image, a brief description of each method and the implementation source. For random distribution, a key is used to initialize a pseudo random number generator.

Table 5.1 Review of the steganographic methods used in the experiments

Method	Domain	Distribution of Modified Pixels/Coeff.	Description	Implementation Source
F5	Transform	Random	Using matrix encoding, the message bits are inserted in the selected coefficients.	Code Google (Gaffga)
Steghide	Spatial	Random	It uses a graph to exchange matching pixel LSB and message bits, to reduce changes.	SourceForge (Hetzl 2002)
Jphide	Transform	Random	Message bits are inserted in the LSB of non-zero DCT coefficients.	Authors' web site (Latham 1999)
SS	Transform	i.i.d. Gaussian	The message is modulated as an i.i.d. Gaussian and inserted in the most significant DCT coefficients.	Hakki Caner Kirmizi (Kirmizi 2010)
LSBMR	Spatial	Random	Pixel pairs are used as embedding unit using increment or decrement.	Dr. Weiqi Luo, School of Software, Sun Yat-Sen University

Method	Domain	Distribution of Modified Pixels/Coeff.	Description	Implementation Source
EALSBMR	Spatial	Random	It is a LSBMR modification where pixel pairs are taken from random size blocks.	Dr. Weiqi Luo, School of Software, Sun Yat-Sen University
MB	Transform	Conditional Probability	It uses an entropy decoder with the model of the conditional probability of the image part to be modified given the rest of it.	Phil Salle web page (no longer available)

In order to avoid detecting JPEG compression instead of the embedded data itself, all algorithms maintains 100% quality. Additionally in order to standardize the embedding process, insertion was made without password, because some of the embedding software does not support it.

Figure 5.2 shows an example of cover image and a Steghide embedded image with 0.05bpp.

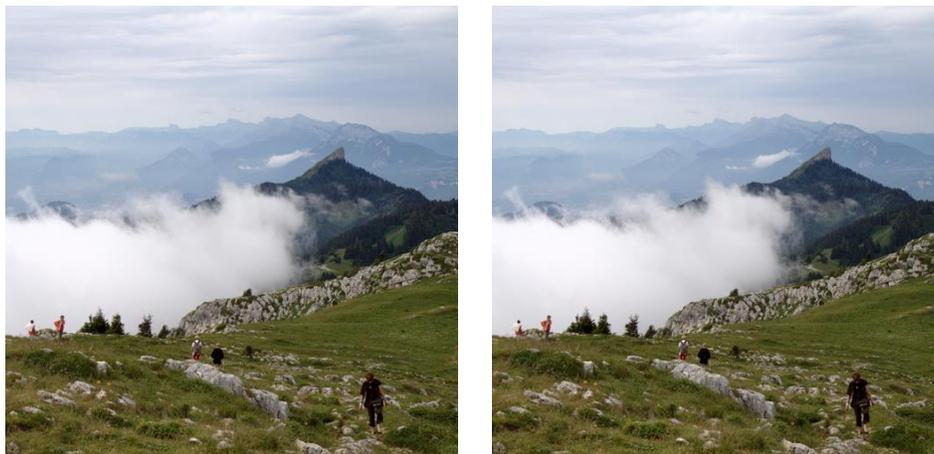


Figure 5.2 *Cover image (left) and Steghide embedded image (right)*

At first glance the above images may look the same, but the embedding process has modified some parts of them only detectable by a steganalysis system. Figure 5.3 shows an example of pixel modified after embedding

0.005bpp with Steghide; the image is the result of the absolute subtraction between cover and stego images. The white pixels are all the differences equal to zero.

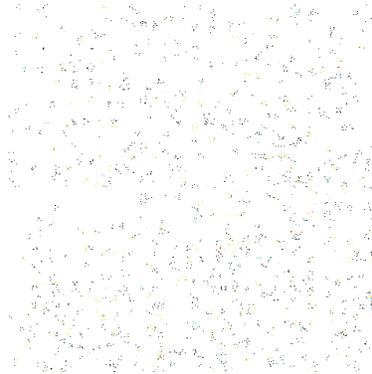


Figure 5.3 *Pixels modified after embedding 0.005bpp with Steghide*

5.1.3 CLASSIFICATION

To evaluate our method, we used the default configuration of the Logistic Regression and Random Forest implementations provided by Weka 3.6.6 (Hall et al. 2009).

The implementation of Logistic Regression in Weka is a multinomial logistic regression model with a ridge estimator algorithm based on Cessie and Houwelingen paper (Cessie and Houwelingen 1992), but with some modifications allowing the algorithm to handle instance weights (Xin).

On the other hand, the implementation of Random Forests is taken from Breiman in (Breiman 2001), without modification.

The experiments were made using cross validation with ten folds for each steganographic system and embedding rate separately. The images of

each fold were picked consecutively; that way, cover and stego of the same image would be together. The training set of the folds contained 8,000 cover images and 8,000 stego images, while the testing set contained 2,000 cover images and 2,000 stego images. For the second level classification stage, after all the results of the first level classification stage were completed the folds were created using the same distribution before.

The metric evaluation used is the detection rate, also known as accuracy given by the equation (2).

$$\frac{\textit{True Positive} + \textit{True Negative}}{\textit{Instance Total Number}} \quad (2)$$

5.2 RESULTS

Because the state of the art steganalysis methods were tested with different images, embedding rates and general parameters, it is difficult to directly compare among them. For this reason, we compare our method with LBP, SPAM, IIC and HOS using the same dataset and classifiers.

To support the test results showed in this section, we use the Wilcoxon statistical significance test, with a certainty of 95%. The results of the proposed method that showed a statistical significance over the other methods are represented as an asterisk next to the detection rate.

For evaluating which classifier was the most suitable for second level classification, results of first level classification using logistic regression and class label of every instance were classified with Voting, Random

Forest, SVM, Multilayer Perceptron and Logistic Regression. Table 5.2 show the results.

Table 5.2 Detection rate results of second level classification for 0.005bpp embedding rate

Embedding Method	Voting	Random Forest	SVM	Multilayer Perceptron	Logistic Regression
F5	98.63*	99.73	99.74	99.73	99.75
Steghide	51.14*	51.07*	52.79	50.04*	52.79
JPHide	50.8*	50.49	51.32	50.36*	51.19
SS	99.18*	99.96	99.93	99.95	99.94
LSBMR	98.4*	99.81	99.7*	99.82	99.80
EALSBMR	98.48*	99.87	99.82	99.86	99.84
MB	50.93*	51.05*	52.16	50.03*	52.16

Due to the detection rate from one classifier to another were almost the same and to standardize the following experiments, we use Logistic Regression as second level classifier.

In the first level classification stage we evaluated the possibility of joining the four features space into one. To test feasibility of using a joined feature space, we tested all the features with Logistic Regression, Random Forest, AdaBoost and Baggins. Table 5.3 shows the results including the results of the proposed method describes in chapter 4.

Table 5.3 Detection rate results of joined feature space for 0.005bpp embedding rate

Embedding Method	Join Logistic	Join RF	Join Baggins	Join AdaBoost	Proposed Method
F5	98.86*	97.2*	99.45*	97.83*	99.75
Steghide	53.08	50.09*	50.2*	50*	52.79
JPHide	52.43*	49.75*	50.2*	50.01*	51.19
SS	99.47*	96.63*	99.48*	97.46*	99.94
LSBMR	99.5*	97.08*	99.49*	97.57*	99.80
EALSBMR	99.5*	97.32*	99.5*	97.68*	99.84
MB	52.19	50.17*	49.88*	49.99*	52.16

Table 5.4, 5.5 and 5.6 show the detection rate percentage of the experiment results for 0.005bpp, 0.01bpp and 0.05bpp respectively. The evaluated embedding method is in the first column. The four next columns contain the obtained detection rate of LBP, SPAM, IIC and HOS using Logistic Regression, while the next four are the results using Random Forest. Penultimate column shows the detection rate average per steganographic method, while last row shows the detection rate average per steganalysis method. The last column is the detection rate of the proposed method. The higher detection rate of each row is in bold.

Table 5.4 Experiment detection rate results for 0.005bpp embedding rate

Embedding Method	Logistic Regression				Random Forest				Average	Proposed Method
	LBP	SPAM	IIC	HOS	LBP	SPAM	IIC	HOS		
F5	90.15*	96.44*	81.38*	99.73	81.64*	94.51*	62.54*	96.65*	87.88	99.75
Steghide	50.36*	50.79*	52.79	50.16*	49.97*	49.69*	50.16*	50.15*	50.51	52.79
Jphide	50.45*	51.40	50.40	50.15*	50.43*	50.29	49.97*	50.10*	50.40	51.19
SS	87.81*	97.20*	90.48*	99.74*	73.31*	92.33*	76.34*	96.45*	89.20	99.94
LSBMR	90.68*	96.46*	59.57*	99.68	82.72*	94.51*	52.27*	96.87*	84.09	99.80
EALSBMR	90.67*	96.66*	59.38*	99.65*	82.72*	94.83*	52.34*	96.88*	84.14	99.84
MB	50.23*	50.45*	52.16	49.99*	50.32*	49.84*	50.10*	49.98*	50.38	52.16
Average	72.91	77.06	63.74	78.44	67.30	75.14	56.25	76.73		79.35

Table 5.5 Experiment detection rate results for 0.01bpp embedding rate

Embedding Method	Logistic Regression				Random Forest				Average	Proposed Method
	LBP	SPAM	IIC	HOS	LBP	SPAM	IIC	HOS		
F5	90.14*	96.33*	82.34*	99.72*	81.53*	94.5*	63.32*	96.79*	88.08	99.84
Steghide	50.79*	51.43*	54.78	50.31*	50.33*	50.29*	50.29*	50.25*	51.06	54.78
Jphide	50.75*	52	50.55	50.35*	50.1*	50.17	49.98*	49.92*	50.48	52.02
SS	87.71*	97.13*	90.57*	99.8*	73.21*	92.06*	76.27*	96.67*	89.18	99.90
LSBMR	90.63*	96.41*	59.59*	99.72	82.71*	94.53*	52.57*	97.01*	84.14	99.86
EALSBMR	90.67*	96.42*	59.31*	99.71*	82.72*	94.68*	51.82*	96.68*	84.00	99.86
MB	50.42*	50.94*	53.94	50.26*	50.13*	49.73*	50.7*	50.2*	50.79	53.94
Average	73.01	77.24	64.44	78.55	67.24	75.14	56.42	76.79		80.03

Table 5.6 Experiment detection rate results for 0.05bpp embedding rate

Embedding Method	Logistic Regression				Random Forest				Average	Proposed Method
	LBP	SPAM	IIC	HOS	LBP	SPAM	IIC	HOS		
F5	90.11*	96.51*	85.95*	99.66*	80.91*	94.57*	68.91*	96.79*	89.17	99.85
Steghide	53.66*	56.86*	67.06	51.61*	51.02*	51.26*	53.24*	50.23*	54.36	67.06
Jphide	53.43*	57.17	69.45	51.23*	50.32*	51.26	53.54*	49.88*	54.53	69.45
SS	87.98*	97.09*	91.54*	99.78*	73.24*	92.42*	78.18*	97.12*	89.67	99.91
LSBMR	90.58*	96.53*	60.74*	99.68	82.49*	94.49*	53.29*	97.13*	84.36	99.86
EALSBMR	90.7*	96.66*	59.69*	99.73*	82.56*	94.83*	52.76*	96.66*	84.20	99.88
MB	52.8*	56.26*	67.93	51.4*	50.45*	50.74*	53.93*	49.81*	54.16	67.93
Average	74.18	79.58	71.77	79.01	67.28	75.65	59.12	76.80		86.27

In the next section, we analyze the obtained results.

5.3 ANALYSIS AND DISCUSSION

Experimental results of joined feature space in Table 5.3 show an improved detection rate with Logistic Regression than the proposed method for Steghide and MB, but they are not statistically significant. On the other hand, although detection rate improvement for JPHide is statistically significant, it is still low for binary classification. For F5, Spread Spectrum, LSBMR and EALSBMR cases, detection rate of the proposed method was statistically significant better than joined feature space. This detection enhancement also implies that meta-classification has better detection performance than standard classifier ensembles such as Baggins or AdaBoost.

Second level classification results (Table 5.2) show similar detection rates for Random Forest, SVM, Multilayer Perceptron and Logistic Regression, while voting has lower detection rates. In this case, Logistic

Regression was chosen for practicality, but any of the tested methods are suitable as second level classifier.

The experimental results for different embedding rates (Tables 5.4, 5.5 and 5.6) show an improvement of the detection rate for our method for F5, Spread Spectrum, LSBMR and EALSBMR. In particular, detection rate of these steganographic methods does not show significant difference among embedding rates, but it slightly increases with more hidden information, except of Spread Spectrum (Figure 5.4).

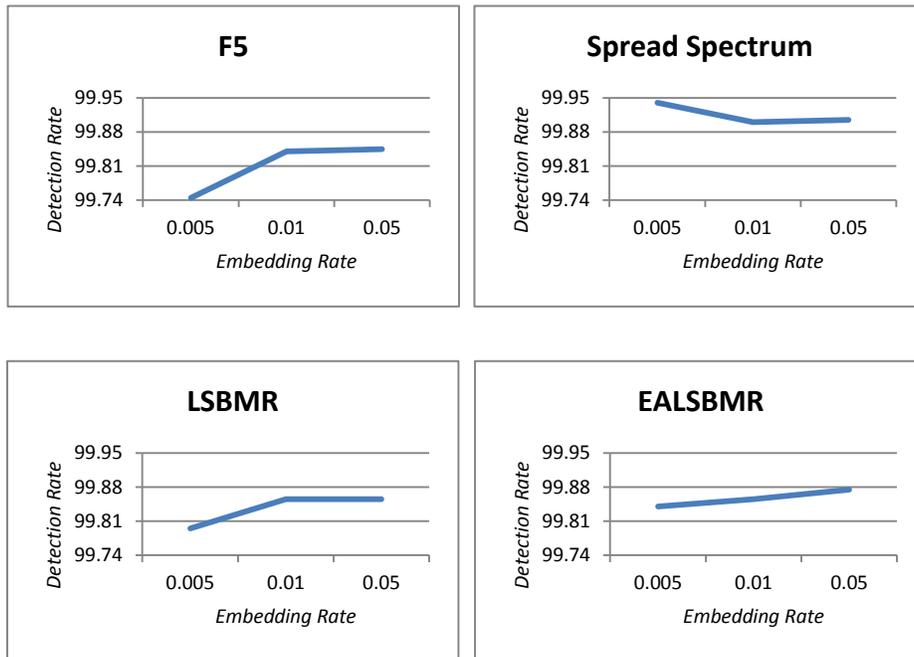


Figure 5.4 Comparison between embedding rates detection of F5, SS, LSBMR and EALSBMR

Steghide, JPHide and Model Based methods show in almost all experiments, equal or higher detection rate. This behavior is the result of poor detection capacity when comparing to the state of the art methods

with the proposed feature extractors, mainly because the amount of modifications in images were too small in comparison with F5, SS, LSBMR and EALSBMR. However, increasing the size of the secret message also increases the detection rate, as it is shown in Figure 5.5.

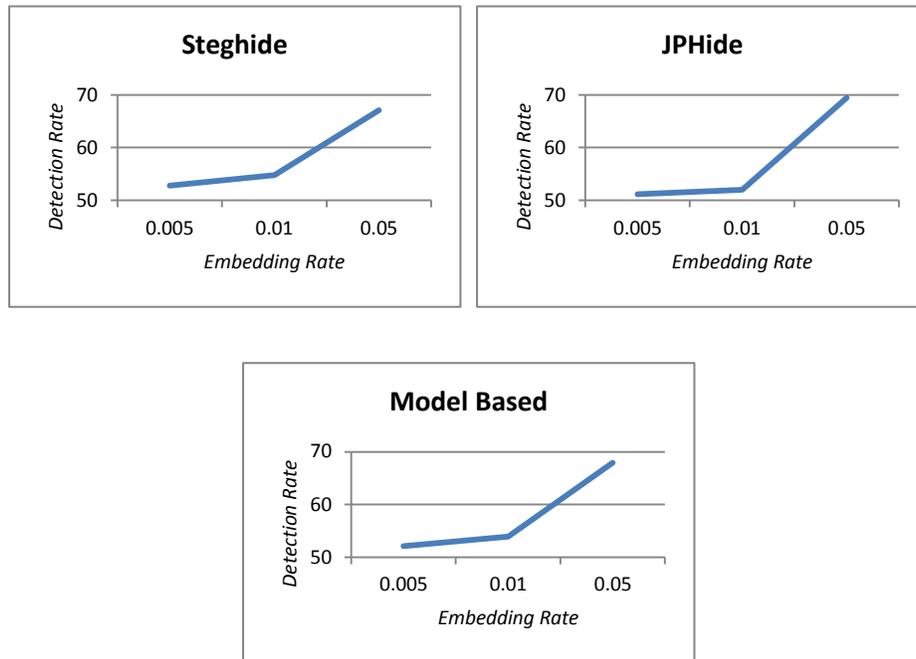


Figure 5.5 Comparison between embedding rates of Steghide, JPHide and Model Based

To sum up, column Average in Tables 5.4, 5.5 and 5.6 shows mean detection rate for each steganographic method; in every case, the proposed method outperform the average. Alternatively, row Average shows mean detection rate per steganalysis method. Here, the mean detection rate of the proposed method outperform the average of methods like HOS and IIC with the higher detection rate for F5, Spread Spectrum, LSBMR and EALSBMR, and for Steghide, JPHide and Model Based respectively, showing the proposed method robustness for different steganographic methods.

5.4 CHAPTER SUMMARY

In this chapter we described the experimental environment used to evaluate our method. Due to a lack of a standard dataset for testing steganalysis systems, we had to build one, based on the cover images provided by the BOSS base(Pevný, Filler, and Bas 2010). Then, using a collection of steganographic software, the cover images were embedded with different amount of bytes, corresponding to 0.005bpp, 0.01bpp and 0.05bpp. In order to make a fair comparison, the methods from the state of the art were tested with the same dataset and classifiers.

The experiment results demonstrated that our method improves the detection rate of the state of art methods for F5, Spread Spectrum, LSBMR and EALSBMR embedding. However for Steghide, JPHide and Model Based, the results of the state of the art and our method barely outperform a random decision, for the embedding rates 0.005bpp and 0.01bpp. For the 0.05bpp embedding rate, the detection rate increases equally for our method and the state of the art.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 CONTRIBUTIONS

According with the research objectives, this thesis accomplished the following.

- A dataset including seven different steganographic methods, with three insertion rates each one.
- A blind steganalysis method that outperforms the state of the art detection rate. Additionally, the proposed method is scalable in order to detect other steganographic methods due to a flexible design.

6.2 CONCLUSIONS

Nowadays there are many steganalysis methods claiming to be universal; however, the lack of standard datasets and settings makes it difficult to prove it. Another problem is the variety of steganographic methods. For example, a steganalysis method may be capable to detect embedding data from some steganographic techniques, but it may not detect the secret message from others. As it is showed in the experimental

results, the high order statistics (HOS) method proposed by Farid(Lyu and Farid 2003) has a good detection rate for F5, Spread Spectrum, LSBMR and EALSBMR, but it cannot detect Steghide, JPHide and Model Based. On the contrary, the IIC method proposed by Chen (Chen and Shi 2008) has a poor performance detecting F5, Spread Spectrum, LSBMR and EALSBMR, but it outperforms the other methods detecting Steghide, JPHide and Model Based. The proposal of this research is to archive universality using different feature extractors.

From experimental results we also notice another issue that affects universality for a steganalysis method: the embedding rate. Some steganographic methods like Steghide, JPHide and Model Based make reduced number of changes in the image than others; lowering chances of detection. The current proposed method is not capable to recognize them for embedding rates lower than 0.05. However, with the flexibility of the proposed method, it is possible to add new feature extractors in order of to recognize them. Something to consider while adding new feature extractors is the feasibility. In other words, more feature extractors should increment the processing time. Higher processing time will limit the use of the method in real applications, where real time is a requirement, for example in a network backbone.

Evaluation of the proposed method against the state of the art showed an improvement in detection rate of F5, Spread Spectrum, LSBMR and EALSBMR. For these steganographic methods, detection rate does not vary substantially from different embedding rates. However, the detection rate of Steghide, JPHide and Model Based barely outperforms a random

decision for 0.005bpp and 0.01bpp embedding rates. For 0.05bpp the detection rate increases, but it still is poor. Nevertheless, the proposed method outperforms detection rate average of different steganalysis methods, as it is showed in Tables 5.4, 5.5 and 5.6.

This research also evaluated joined feature space feasibility. The results showed that it improves detection rate for Steghide, JPHide and Model Based steganographic systems, however this entails two disadvantages. First, classifier training time increases substantially, because of dimensionality that goes from 686 at worst case to 1644 features. Second, characteristics as flexibility and scalability of proposed method are reduced, as a result of the training process. The proposed method is designed as modules, in other words, if a new feature set is included to the scheme, it would not be necessary to retrain both classifier levels, just the second one. But joined feature space requires retraining both classifier levels.

It was also tested two ensemble classifiers: Baggins and AdaBoost. Here, experimental results showed that the proposed method has better detection rate.

Tests for second level classifier were not conclusive, because excluding voting, classification results were similar among Random Forest, SVM, Multilayer Perceptron and Logistic Regression. For this reason either classifier could be used as second lever classifier.

6.3 FUTURE WORK

Although the proposed method showed an improvement in the detection rate for some steganographic methods, it is necessary to test even more of them. For this reason, it is recommendable to increase the current dataset with new stego images and different embedding rates. Also, future tests could include totally new datasets, in order to make cross-dataset tests. About steganographic methods with poor detection rate, it is possible to test higher embedding rates, in order to determine how much information is required to be inserted to allow detection.

If the proposed method does not recognize a steganographic method at any embedding rate; it is possible to include or replace one or more of the current feature extractors.

Other improvements to the proposed method could be multiclass response and feature selection. With multiclass response, the proposed method might determine which steganographic method is responsible for the content of the stego image; while with feature selection, besides a possible detection rate improvement, it may also reduce classifier training time.

BIBLIOGRAPHY

- Amin, Muhalim Mohamed, Subariah Ibrahim, Mazleena Salleh, and Mohd Rozi Katmin. 2003. "Information Hiding Using Steganography." *4th National Conference on Telecommunication Technology, 2003*.
- Arivazhagan, S., W. Sylvia Lilly Jebarani, and M. Shanmugaraj. 2011. "A Novel Approach to Low Volume Generic Steganalysis." *2011 21st International Conference on Systems Engineering (August)*: 153–158. doi:10.1109/ICSEng.2011.35.
- Barni, Mauro, Sviatoslav Voloshynovskiy, and Fernando Perez-Gonzalez. 2005. "Break Our Watermarking System." <http://lci.det.unifi.it/BOWS/>.
- Bas, Patrick, and Teddy Furon. 2007. "Break Our Watermarking System 2." <http://bows2.ec-lille.fr/>.
- Bayram, Sevinc, Ahmet Emir Dirik, Husrev Taha Sencar, and Nasir Memon. 2010. "An Ensemble of Classifiers Approach to Steganalysis." *2010 20th International Conference on Pattern Recognition (August)*: 4376–4379. doi:10.1109/ICPR.2010.1064.
- Bishop, Christopher. 2006. *Pattern Recognition and Machine Learning*. Springer.
- Böhme, Rainer. 2010. *Advanced Statistical Steganalysis*. Edited by Ueli Maurer and David Basin. Berlin: Springer Berlin / Heidelberg.
- Breiman, L. 2001. "Random Forests." *Machine Learning*: 5–32.
- Cessie, S Le, and JC Van Houwelingen. 1992. "Ridge Estimators in Logistic Regression." *Applied Statistics* 41 (1): 191–201.
- Chanu, Yambem Jina, Themrichon Tuithung, and Kh. Manglem Singh. 2012. "A Short Survey on Image Steganography and Steganalysis Techniques." *2012 3rd National Conference on Emerging Trends and Applications in*

Computer Science (March): 52–55.
doi:10.1109/NCETACS.2012.6203297.

- Chen, Chunhua, and Yun Q. Shi. 2008. “JPEG Image Steganalysis Utilizing Both Intrablock and Interblock Correlations.” *2008 IEEE International Symposium on Circuits and Systems* (May): 3029–3032. doi:10.1109/ISCAS.2008.4542096.
- Cox, Ingemar, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. 2008. *Digital Watermarking and Steganography*. Second. Burlington: Elsevier Inc.
- De Leeuw, Karl, and Jan Bergstra. 2007. *The History of Information Security: A Comprehensive Handbook*. Amsterdam: Elsevier Inc.
- Fridrich, Jessica, and Miroslav Goljan. 2003. “Digital Image Steganography Using Stochastic Modulation.” Edited by Edward J. Delp III and Ping W. Wong. *Proceedings of SPIE* (June 13): 191–202. doi:10.1117/12.479739.
- Fridrich, Jessica, and Jan Kodovsky. 2012. “Rich Models for Steganalysis of Digital Images.” *IEEE Transactions on Information Forensics and Security* 7 (3): 868–882.
- Gaffga, Christoph. “F5-Steganography.” <https://code.google.com/p/f5-steganography/>.
- Guan, Qingxiao, Jing Dong, and Tieniu Tan. 2011a. “An Effective Image Steganalysis Method Based on Neighborhood Information of Pixels.” *IEEE International Conference on Image Processing: 2721–2724*.
- . 2011b. “Blind Quantitative Steganalysis Based on Feature Fusion and Gradient Boosting.” *Digital Watermarking*.
- Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. “The WEKA Data Mining Software: An Update.” *SIGKDD Explorations* 11 (1).
- Harwood, David, Timo Ojala, Matti Pietikäinen, Shalom Kelman, and Larry Davis. 1995. “Texture Classification by Center-Symmetric Auto-

- Correlation, Using Kullback Discrimination of Distributions.” *Pattern Recognition Letters* 16 (1): 1–10.
- Hetzl, Stefan. 2002. “Steghide.” <http://steghide.sourceforge.net/>.
- Hetzl, Stefan, and Petra Mutzel. 2005. “A Graph–theoretic Approach to Steganography.” *Communications and Multimedia Security*: 119–128.
- Hui, L I, S U N Ziwen, and Zhou Zhiping. 2011. “An Image Steganalysis Method Based on Characteristic Function Moments and PCA”: 3005–3008.
- Johnson, Neil F., Zoran Duric, and Jajodia Sushil. 2001. *Information Hiding: Steganography and Watermarking: Attacks and Countermeasures*. Kluwer Academic Publishers.
- Kawaguchi, Eiji, and RO Eason. 1998. “Principles and Applications of BPCS Steganography.” *Multimedia Systems and Applications*.
- Kharrazi, Mehdi, Husrev T Sencar, and Nasir Memon. 2004. “Image Steganography: Concepts and Practice.”
- Kirmizi, Hakki Caner. 2010. “Spread Spectrum.” <https://github.com/israkir/cox-watermarking>.
- Kodovsky, Jan, Jessica Fridrich, and Vojtech Holub. 2012. “Ensemble Classifiers for Steganalysis of Digital Media.” *Information Forensics and Security* 7 (2): 432–444.
- Lafferty, Patricia, and Farid Ahmed. 2004. “Texture-Based Steganalysis: Results for Color Images.” *Proceedings of SPIE* 5561: 145–151. doi:10.1117/12.559896.
- Latham, Allan. 1999. “JPHide and JPSeek.” <http://linux01.gwdg.de/~alatham/stego.html>.
- Li, Bin. 2011. “A Survey on Image Steganography and Steganalysis.” *Information Hiding and Multimedia Signal Processing* 2 (2): 142–172.

- Luo, Weiqi, Fangjun Huang, and Jiwu Huang. 2010. "Edge Adaptive Image Steganography Based on LSB Matching Revisited." *IEEE Transactions on Information Forensics and Security* 5 (2) (June): 201–214. doi:10.1109/TIFS.2010.2041812.
- Lyu, Siwei, and Hany Farid. 2003. "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines." *Information Hiding*.
- Maity, SP, Seba Maity, Jaya Sil, and Claude Delpha. 2012. "Spread Spectrum Watermarking: Principles and Applications in Fading Channel." *Watermarking-Volume 1*.
- Maney, Kevin. 2001. "Bin Laden's Messages Could Be Hiding in Plain Sight." *USA Today*, December 19.
- Mielikainen, J. 2006. "LSB Matching Revisited." *IEEE Signal Processing Letters* 13 (5) (May): 285–287. doi:10.1109/LSP.2006.870357.
- Niimi, Michiharu, and Hideki Noda. 2011. "An Application of Sparse Code Shrinkage to Image Steganalysis Based on Supervised Learning." *18th IEEE International Conference on Image Processing: 1941–1944*.
- Nissar, Arooj, and a.H. Mir. 2010. "Classification of Steganalysis Techniques: A Study." *Digital Signal Processing* 20 (6) (December): 1758–1770. doi:10.1016/j.dsp.2010.02.003.
- Ojala, Timo, Matti Pietikäinen, and David Harwood. 1996. "A Comparative Study of Texture Measures with Classification Based on Featured Distributions." *Pattern Recognition* 29 (1): 51–59.
- Pevný, Tomáš, Patrick Bas, and Jessica Fridrich. 2010. "Steganalysis by Subtractive Pixel Adjacency Matrix." *IEEE Transactions on Information Forensics and Security* 2 (5): 215–224.
- Pevný, Tomáš, Tomáš Filler, and Patrick Bas. 2010. "Break Our Steganographic System." <http://www.agents.cz/boss>.
- Provos, N. 2001. "Defending Against Statistical Steganalysis." *Usenix Security Symposium*: 323–335.

- Robertson, Nic, Paul Cruickshank, and Tim Lister. 2012. "Documents Reveal Al Qaeda's Plans for Seizing Cruise Ships, Carnage in Europe." *CNN*, May 1.
- Rocha, Anderson, and Siome Goldenstein. 2008. "Steganography and Steganalysis in Digital Multimedia: Hype or Hallelujah?" *Journal of Theoretical and Applied Computing*.
- Rodriguez, Benjamin, Gilbert Peterson, and Kenneth Bauer. 2008. "Fusion of Steganalysis Systems Using Bayesian Model Averaging." In *IFIP International Federation for Information Processing, Advances in Digital Forensics IV*, 285:345–355. Boston: Springer.
- Sallee, Phil. 2004. "Model-Based Steganography." *Digital Watermarking*: 154–167.
- Shi, Y.Q., Guorong Xuan, Dekun Zou, Jianjiong Gao, Chengyun Yang, Zhenping Zhang, Peiqi Chai, Wen Chen, and Chunhua Chen. 2005. "Image Steganalysis Based on Moments of Characteristic Functions Using Wavelet Decomposition, Prediction-Error Image, and Neural Network." *2005 IEEE International Conference on Multimedia and Expo*: 269–272. doi:10.1109/ICME.2005.1521412.
- Simmons, GJ. 1983. "The Prisoners' Problem and the Subliminal Channel." *Advances in Cryptology: Proceedings of CRYPTO '83*: 51–67.
- Solanki, Kaushal, Anindya Sarkar, and BS Manjunath. 2007. "YASS: Yet Another Steganographic Scheme That Resists Blind Steganalysis." *Information Hiding*: 1–15.
- Sun, Ziwen, Jiajie Liu, and Zhicheng Ji. 2011. "Distributed Fusion Steganalysis Based on Combination System Likelihood Function." *2011 10th International Symposium on Distributed Computing and Applications to Business, Engineering and Science* (October): 221–225. doi:10.1109/DCABES.2011.38.
- Westfeld, Andreas. 2001. "F5 — A Steganographic Algorithm High Capacity Despite Better Steganalysis": 289–302.

Xin, Xu. "Multinomial Logistic Regression Model with a Ridge Estimator." *Machine Learning Group at the University of Waikato*. <http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/Logistic.html>.

Zhang, Xinpeng, and Shuozhong Wang. 2005. "Steganography Using Multiple-Base Notational System and Human Vision Sensitivity." *IEEE Signal Processing Letters* 12 (1) (January): 67–70. doi:10.1109/LSP.2004.838214.