ABSTRACT

REVIEW OF STEGANALYSIS OF DIGITAL IMAGES

by
Xinlei Pan

Steganography is the science and art of embedding hidden messages into cover multimedia such as text, image, audio and video. Steganalysis is the counterpart of steganography, which wants to identify if there is data hidden inside a digital medium. In this study, some specific steganographic schemes such as HUGO and LSB are studied and the steganalytic schemes developed to steganalyze the hidden message are studied. Furthermore, some new approaches such as deep learning and game theory, which have seldom been utilized in steganalysis before, are studied. In the rest of thesis study some steganalytic schemes using textural features including the LDP and LTP have been implemented.

REVIEW OF STEGANALYSIS OF DIGITAL IMAGES

by
Xinlei Pan

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Helen and John C. Hartmann Department of
Electrical and Computer Engineering

May 2015

APPROVAL PAGE

REVIEW OF STEGANALYSIS OF DIGITAL IMAGES

Xinlei Pan

Dr. Yunqing Shi, Thesis advisor Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Edwin Hou, Committee Member Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Diqun Yan, Committee Member Date
Associate Professor of Electrical and Computer Engineering, NBU

BIOGRAPHICAL SKETCH

Author: Xinlei Pan

Degree: Master of Science

Major: Electrical Engineering

Date: May 2015

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2015

- Bachelor of Science in Electrical Engineering,
  Hangzhou Dianzi University, Zhejiang, P. R. China, 2013

To who helps me to overcome my own shortage.

ACKNOWLEDGMENT

Foremost, I would like to express my deepest gratitude to my advisor Professor Yun Q. Shi for the continuous support of my study. I have been amazingly fortunate to have an advisor who gave me the freedom to explore the newest and cutting edge area. Even though I had so many troubles in my experiment and writing. But his patience, immense knowledge and friendly communication had helped me overcome all these crisis.

I would like to thank my thesis committee: Dr. Edwin Hou and Dr. Diqun Yan, who have been always there to listen and give advice. I am deeply grateful to them for their constant patient and kindhearted help.

I have to thank to my mentor Guanshuo Xu, Jingyu Ye and Dr. Zhihua Xia for their insightful comments, hard questions and prompt response. When I felt stuck in a rut, they always help me to find another way to achieve my target and make it better. I could not finish my experiment without their support.

I am also thankful to Ms. Lilian Quiles, Ms. Clarisa Gonzalez and staff in Office of Graduate Studies and Writing Center, for their various forms of support during my thesis writing.

Finally, I would like to thank my beloved parents from my deep the heart. They have stood by me through the good times and bad.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Concept

The objective of this thesis is to study and classify methods in steganalysis and to try show some improvement in steganalysis.

Steganalysis is the study of detecting whether a suspected document has a payload encoded in it; in other words, steganalysis is the counter part of steganography. Steganography is the science and art of hiding secret messages into innocuous looking cover documents, such as speeches and images. Each steganographic communication system consists of data embedding part and the hidden date extraction part.

A lot of steganographic schemes are freely available today on the internet such as J-Stego [1], EzStego [2], MB [5], OutGuess [40], F5 [4] etc. Most of these steganographic methods modify the redundant bits in a carrier to hide secret messages. This doing however changes the statistical properties of the cover medium as creating a stego medium. There are two most popular methods used for steganography: spatial domain embedding and transform domain embedding.

Likely, steganalysis also can be classified in two types: specific and universal. The specific type focus on the particular steganographic algorithm and this type has a high success rate for detecting the presence of secret messages. The universal steganalysis algorithms are designed to be operatable all known and unknown steganography algorithms.

Universal steganalysis can be considered as a two-class pattern classification problem to classify the test images as a cover or a stego image. Generally, the classification consists of two procedures, the feature extraction and the pattern classification. Generally, a set of feature is a representation of an image with much lower dimensionality and is crucial for many pattern recognition problems, including steganalysis. The effective features for steganalysis should extract information about the changes incurred by data hiding rather and compress the content of the image.

It has been about 15 years since the research on steganalysis has been developed. As a specific steganalysis algorithm, the Raw Quick Pair (RQP) was proposed by Fridrich et al. [7] in 1999. This method is based on analyzing the close colors by LSB embedding [6]. Because the number of close colors in an image with embedded code is obviously larger than it with a normal image. Hence, this algorithm works very well as long as the number of close colors in the cover image is less than 30% of the number of pixels.

In 2001, Fridrich et al. [8] also claimed a new Specific Steganalysis called Regular and Singular group (RS Steganalysis) for detecting (least significant bit) LSB non-sequential embedding. The image is divided into disjointed groups of fixed shapes. Each group noise is measured by the mean absolute value of the differences between adjacent pixels. Those groups will be classified as regular or singular depending on whether the pixel noise is increased when using a ‘mask’. When data are embedded into an image using LSB method, one can analysis the regular and singular groups.

As the universal method, Li et al. [9] designed a feature extraction method

containing two parts. One is generated from the coefficient co-occurrence matrices which was proposed by Kodovsky et al. [18] in 2011, while another part is derived from the co-occurrence matrices of coefficient differences. They trained those features by sub-classifiers which are integrated by an ensemble classifier with a Bayesian mechanism [25]. In this way, the performance is improved by 2%.

Fridrich et al. [10] introduced a spatial-domain steganalytic method (Rich Model) for detecting common steganography. The rich model is assembled by submodel, which is based on its detection error: the out-of-bag error calculated from the training set. They estimate the detection accuracy by observing the difference between how different submodels engage in detection.

Shi et al. [3] had shown that textural features are a very helpful choice for steganalysis for Highly Undetectable Steganography (HUGO) in 2012 [19]. They learned and utilized the textural features from rich literature in the field of texture classification for further development of modern steganalysis. They use the local binary pattern as the textural feature framework with a group of textural feature masks, including Markov neighborhoods [30], cliques [3] and Laws' masks [30].

It is clearly that steganalysis is making progress. In recent years, game theory [11] and deep learning [17], which have never used before, are involved into steganalysis now. Tomáš et al. [11] introduced a powerful steganalytic method in the detection of content adaptive LSB Matching [26] with a gaming theory in 2014. They focused on the modern steganographic embedding paradigm based on minimizing an additive distortion function.

The strategies of both players are comprised of the probabilistic selection channel. In this paper, they demonstrate the example of a two-pixel cover that the Nash equilibrium that minimizes the KL divergence between cover and stego objects.

Qian et al. [17] demonstrated that deep learning can be used in steganalysis in 2015. They proposed a new paradigm to learn features automatically via customized Convolutional Neural Networks (CNN) [28], which has never been used in steganalysis before. What's more, the guidance of classification can be used during the feature extraction step. Even though the accuracy in HUGO cannot be higher than Spatial Rich Model (SRM), this technique still achieves comparable performances.

In this thesis, an attempt has been made to make a note of various approaches proposed for steganalysis of digital images and the classification of them. The rest of the paper is organized as follows. Chapter 2 shows six classic steganalytic techniques, while Chapter 3 presents the newest approaches in steganalysis; In Chapter 4, we propose an improved method for Shi et al. [3], by replacing the local binary pattern (LBP) with the Local Derivative Patter (LDP) [12] and LTP [13]. Summary and conclusions drawn from the study have been given in Chapter 5.

1.2 Overview of Steganography

1.2.1 General Concepts

The Internet offers great convenience in transmitting large amounts of data to different parts of the world. However, the safety and security of long distance communication remains an issue. This problem has led to the development of steganography schemes. Steganography is an ancient idea of hiding information. If it works well, the message does not attract attention from eavesdroppers and attackers.

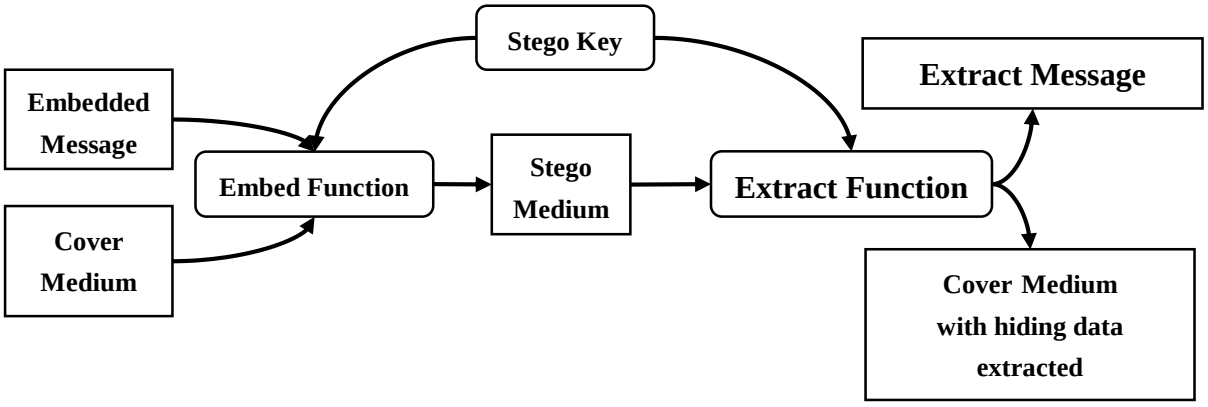


Figure 1.1 Framework of steganography.

Figure 1.1 shows how steganography operates over Cover Medium and the Embedded Message which may be text, or any other type of data, with a Stego Key which is a password to produce a Stego Medium.

1.3 Modern Techniques of Steganography

Steganography is the science and art of hiding information. Steganography techniques can be defined into four categories: Physical steganography, Digital steganography, Network steganography and Printed steganography. In this thesis, we mainly talk about Image steganography. Image files like Bit Map Picture (MBP), Portable Network Graphics (PNG), Joint Picture Expert Group (JPEG) and etc. are used to hide data.

1.3.1 LSB Embedding Algorithm

The most widely used technique to hide secret data at the early stage is the Least Significant Bit (LSB) [13], which is an Image Domain technique. This method uses the least significant bits of early pixels in a digital image. When using a 24 bit color image, bits of red, green and blue color component can be used; in this way more secret bits can be embedded.

Another category of image steganography techniques is transform domain techniques. Transform domain techniques encode secret messages in its transform areas of cover images which makes those messages more robust to attacks such as compression and cropping. The Discrete Cosine Transform (DCT) [29] domain is widely used in Transform Domain techniques, e.g. JPEG images used the DCT for compression. A JPEG encoder partitions an image into many 8×8 blocks. Each block is converted to frequency coefficients by using two-dimensional DCT. However many of the 8×8 coefficients are equal to zero, and it will have an effect on the compression rate if we change too many

zeros to non-zero values. It is the reason that the number of bits one could embed in to the DCT domain is less than the number by using LSB method.

1.3.2 F5 Algorithm

The F5 steganographic algorithm was introduced by Andreas Westfeld [4]. The goal of their research was to develop a practical embedding method for JPEG images that would provide high steganographic capacity without sacrificing security. Guided by their χ^2 attack, they challenged the paradigm of replacing bits of information in the cover-image with the secret message while proposed a different paradigm of incrementing image components to embed message bits.

Instead of replacing the LSBs of quantized DCT coefficients with the message bits, the absolute value of the coefficient is decreased by one. The authors argued that this type of embedding cannot be detected by using the χ^2 statistical attack.

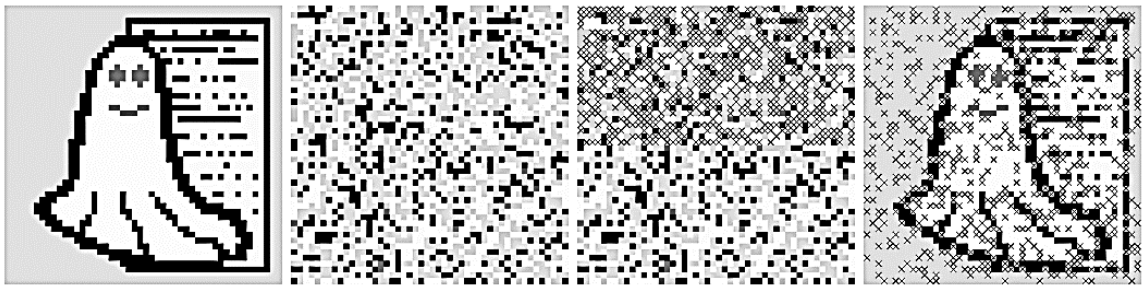


Figure 1.2 Permutative embedding scatters the changes (×).

Source: Andreas Westfeld, "F5—A Steganographic Algorithm High Capacity Despite Better Steganalysis," 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings, pp. 289-302.

The straddling mechanism used with F5 shuffles all coefficients using a permutation firstly. Then, F5 embeds data into the permuted sequence. The transformation does not change the number of coefficients. The permutation depends on a password. F5 delivers the steganographically changed coefficients in the original sequence to the Huffman coder. With the correct key, the receiver is able to repeat the permutation. The permutation has linear time complexity $O(n)$. Figure 1.2 shows the uniformly distributed changes over the whole image.

The embedding process starts with a seed for a PRNG from the user password and generating a random walk through the DCT coefficients of the cover image. The PRNG is also used to encrypt the value k using a stream cipher and embed it in a regular manner together with the message length in the beginning of the message stream. The body of the message is embedded using matrix embedding, inserting k message bits into one group of $2^k - 1$ coefficients by decrementing the absolute value of at most one coefficient from each group by one.

The embedding process consists of the following six steps:

1. Get the RGB representation of the input image.
2. Calculate the quantization table corresponding to quality factor Q and compress the image while storing the quantized DCT coefficients.
3. Compute the estimated capacity with no matrix embedding $C = h_{DCT} - \frac{h_{DCT}}{64} - h(0) - h(1) + 0.49h(1)$, where h_{DCT} is the number of all DCT coefficients, $h(0)$ is the number of AC DCT coefficients equal to zero.
4. The user-specified password is used to generate a seed for a PRNG that determines the random walk for embedding the message bits.

5. The message is divided into segments of k bits that are embedded into a group of $2^k - 1$ coefficients along the random walk.

The following example shows what happened with the Matrix Encoding. If we want to embed two bits x_1, x_2 in three modifiable bit places a_1, a_2, a_3 changing one place at most. Here are all of the four cases:

$$x_1 = a_1 \oplus a_3, x_1 = a_2 \oplus a_3 \Rightarrow change\ nothing$$

$$x_1 \neq a_1 \oplus a_3, x_1 = a_2 \oplus a_3 \Rightarrow change\ a_1$$

$$x_1 = a_1 \oplus a_3, x_1 \neq a_2 \oplus a_3 \Rightarrow change\ a_2$$

$$x_1 \neq a_1 \oplus a_3, x_1 \neq a_2 \oplus a_3 \Rightarrow change\ a_3$$

where $\oplus$ presents XOR operation. In general, we have a code word a with n modifiable bit places for k secret message bits x. Let f be a hash function that extracts k bits from a code word. Matrix encoding enables us to find a suitable modified code word a for every a and x with $x = f(a')$.

F5 implements matrix encoding only for $d_{max} = 1$. For $(1, n, k)$, the code words have the length $n = 2^k - 1$. The desity:

$$D(k) = \frac{1}{n+1} = \frac{1}{2^k} \tag{1.1}$$

And the embedding rate

$$R(k) = \frac{k}{n} = \frac{1}{n} \cdot ld(n+1) = \frac{k}{2^k - 1} \tag{1.2}$$

We can define the embedding efficiency $W(k)$:

$$W(k) = \frac{R(k)}{D(k)} = \frac{2^k}{2^k - 1} \cdot k \tag{1.3}$$

The embedding efficiency of the $(1, n, k)$ code is always larger than k.

1.3.3 HUGO Embedding Algorithm

Pevný et al. [19] presented the highly-undetectable steganography (HUGO) for digital media, which learned from the technique, known as steganalysis technique, known as SPAM [27]. Because SPAM uses further higher statistics, however, it leads to high dimensionality, which is hard to possess for steganalysis.

The main design process is to minimize a suitably-defined distortion by means of efficient coding algorithm. The distortion is defined as a weighted difference of extended state-of-the-art feature vectors already used in steganalysis. This allows them to 'preserve' the model used by steganalysis and thus be undetectable even for large payloads. What's more, the high dimensional model is necessary to avoid known security weaknesses which

is acceptable in this technique.

As same as SPAM, HUGO algorithm uses second order Markov Process.

$$\boldsymbol{D}_{\overrightarrow{i,j}} = \boldsymbol{I}_{i,j} + \boldsymbol{I}_{i,j+1} \tag{1.4}$$

where D is the difference array, which is for horizontal left to right, I is an image.

$$\boldsymbol{M}_{\overrightarrow{d_1,d_2}} = \Pr(\boldsymbol{D}_{\overrightarrow{i,j+1}} = d_1, \boldsymbol{D}_{\overrightarrow{i,j}} = d_2) \tag{1.5}$$

And the second-order Markov process is used,

$$\boldsymbol{M}_{\overrightarrow{d_1,d_2,d_3}} = \Pr(\boldsymbol{D}_{\overrightarrow{i,j+2}} = d_1 | \boldsymbol{D}_{\overrightarrow{i,j+1}} = d_2, \boldsymbol{D}_{\overrightarrow{i,j}} = d_3) \tag{1.6}$$

It is known that, the 2nd order Markov process is equivalent to the 3rd order co-occurrence, under certain condition, which can be satisfied.

$$\boldsymbol{C}_{\overrightarrow{d_1,d_2,d_3}} = \Pr(\boldsymbol{D}_{\overrightarrow{i,j+2}} = d_1, \boldsymbol{D}_{\overrightarrow{i,j+1}} = d_2, \boldsymbol{D}_{\overrightarrow{i,j}} = d_3) \tag{1.7}$$

Figure 1.3 shows the individual steps of the HUGO algorithm.

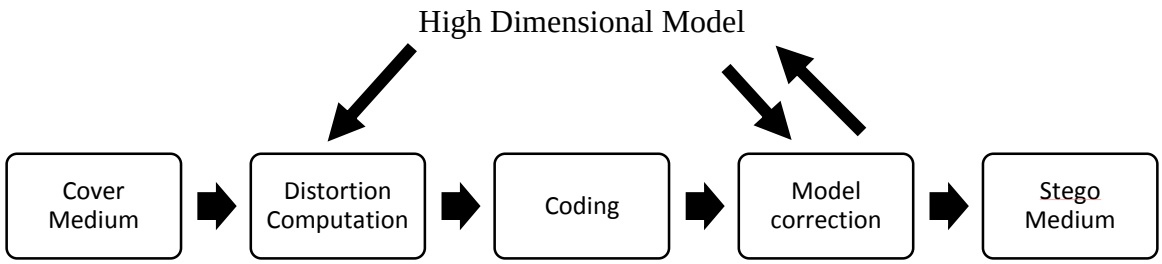


Figure 1.3 High-level diagram of HUGO.

Source: Tomáš Pevný, Tomáš Filler, Patrick Bas, "Using high-dimensional image models to perform highly undetectable steganography," Information Hiding. Springer, Heidelberg, 2010, pp. 161-177.

The accuracy is evaluated by examine the minimal average decision error equal of cover and stego images:

$$P_E = \min \frac{1}{2}\left(P_{F_P} + P_{F_n}\right) \tag{1.5}$$

where P_{F_P} and P_{F_n} stand for the probability of false alarm and probability of missed detection. The additive distortion measure:

$$D(X,Y) = \sum_{i=1}^{n} \rho_i |x_i - y_i| \tag{1.6}$$

where the constants $0 \leq \rho_i \leq \infty$ are fixed parameters expressing cost of pixel changes.

In order to stress those parts of the co-occurrence matrices that are more important for steganalysis, the Equation 1.7 is defined as a weighted sum of differences.

$$D(X,Y) = \sum_{d_1,d_2,d_3=-T}^{T} \left[w(d_1,d_2,d_3) \left| \sum_{k\in\{\rightarrow,\leftarrow,\uparrow,\downarrow\}} C_{d_1 d_2 d_3}^{X,k} - C_{d_1 d_2 d_3}^{Y,k} \right| + w(d_1,d_2,d_3) \left| \sum_{k\in\{\searrow,\nwarrow,\swarrow,\nearrow\}} C_{d_1,d_2,d_3}^{X,k} - C_{d_1,d_2,d_3}^{Y,k} \right| \right] \tag{1.7}$$

where $w(d_1,d_2,d_3)$ is a weight function:

$$w(d_1,d_2,d_3) = \frac{1}{\left[\sqrt{d_1^2+d_2^2+d_3^2}+\sigma\right]^{\gamma}} \tag{1.8}$$

where $\sigma,\gamma > 0$ are parameters that can be tuned in order to minimize the detectability.

Pevný et al. derived the algorithm to hide the message into the image parts so that it is difficult to be detected. The HUGO embedding algorithm is shown in Figure 1.4.

HUGO embedding algorithm

```
for (i,j) in PIXELS {            //function D is taken from (1.2)
   Yp = X; Yp(i,j)++; rho_p(i,j) = D(X,Yp);  //calculate emb. impact
   Ym = X; Ym(i,j)--; rho_m(i,j) = D(X,Ym);  //for each pixel
}
rho_min = min(rho_p, rho_m);    //elementwise; use minimum for embedding
PIXELS_TO_CHANGE = minimize_emb_impact(LSB(X), rho_min, message)
Y = X;                          //start making changes in cover image
for (i,j) in PIXELS_TO_CHANGE { //order given by the MC visit. strategy
   if ( model_correction_step_enabled ) {
      Yp = Y; Yp(i,j)++; dp = D(X,Yp); Ym = Y; Ym(i,j)--; dm = D(X,Ym);
      if ( dp<dm ) { Y(i,j)++; } else { Y(i,j)--; }
   } else {
      if ( rho_p(i,j)<rho_m(i,j) ) { Y(i,j)++; } else { Y(i,j)--; }
   }
}
```

Figure 1.4 Code of the HUGO embedding algorithm.

Source: Tomáš Pevný, Tomáš Filler, Patrick Bas, "Using high-dimensional image models to perform highly undetectable steganography," Information Hiding. Springer, Heidelberg, 2010, pp. 161-177.

The security of HUGO has been verified and compared to the prior art on a wide range of payloads in their experiments. When the fixed classification error P_E is 40% of SVM-based [32] in 2nd-order SPAM, the HUGO increases the secure payload from 0.25 bpp to 0.4 bpp. In contrast with the LSB matching, when $P_E = 40\%$ on BOWS dataset, HUGO allows the embedder to hide 7 times longer message with the same security.

CHAPTER 2

STEGANALYSIS METHODS

There are two categories in steganalysis: specific and universal. Specific steganalysis direct at particular image features, which are modified by the specific embedding algorithm. A steganalysis technique would perform well when tested only on that method and might fail on all others. Some specific steganalysis methods are even be able to estimate the length of hidden message. Unlike specific steganalysis concerting on a specific steganalysis techniques, the universal techniques try to identify the all of steganography algorithms. Hence, the general steganalysis methods are more flexible and practical. Universal techniques are, however, sometimes cannot detect the targeted embedding algorithms more effectively than the specific method designed for breaking the steganographic method. In this Chapter, six kinds of classic steganalysis schemes are presented according to their publication time. The effective features for steganalysis should be able to catch the changes incurred by data hiding.

2.1 Chi-Square Attack

2.1.1 Concept

This method is specific to LSB embedding based on powerful first order statistical analysis rather than visual inspection. LSB embedding overwrite least significant bits transforms values into each other. [1]

Figures 2.1 and 2.2 show that the total number of occurrence of two members of certain pairs of values (PoV) remains same after message embedding. This concept of pair wise dependencies is exploited to design a statistical Chi-square test to detect the hidden messages.

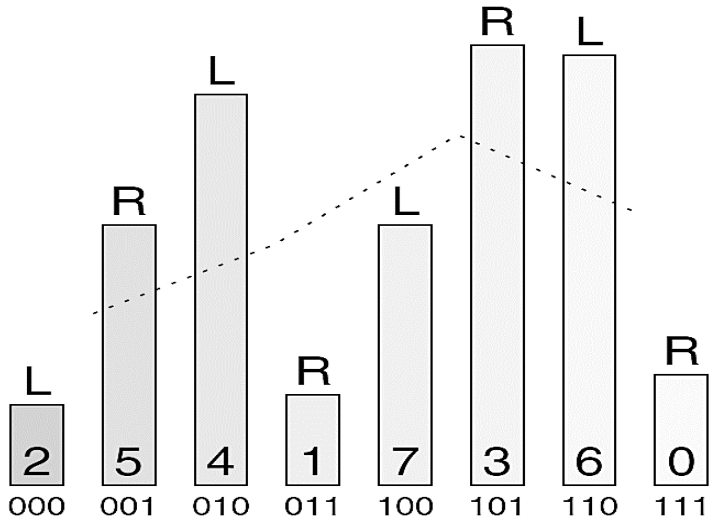


Figure 2.1 The color histogram before embedding.

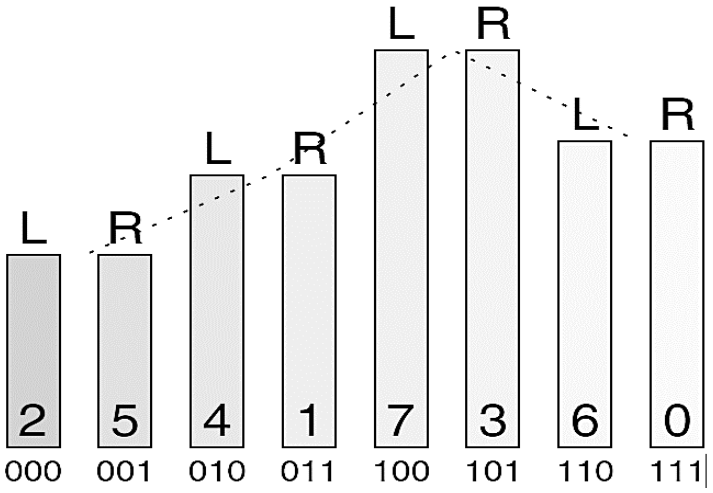


Figure 2.2 The color histogram after embedding.

Source: Andreas Westfeld, Andreas Pfitzmann, "Attacks on steganographic systems, in: Proc. of Information Hiding," 3rd Int. Workshop, Dresden, Germany, September 28–October 1, 1999, pp. 61–75.

A critical point is how to obtain the theoretically expected frequency distribution. In the original, the theoretically expected frequency is the arithmetic mean of the two frequencies in a PoV. Since swapping one value into another does not change the sum of occurrences of both values in each pair, the arithmetic mean of the two frequencies for each pair is the same in both cover and stego image.

2.1.2 Implement

This test performs the following steps:

A. Supposing there are k categories and we have a random sample of observations. Each observation must fall in only one category. Without restricting, we concentrate on the odd values of PoVs of the attacked carrier medium. For example, for a palette image with 256 color, which means most 128 PoVs and k=128.

B. The theoretically expected frequency in category i after embedding an equally distributed message is

$$n_i^* = \frac{|\{color|sorted\ Index\ of\ (color) \in \{2i, 2i+1\}\}|}{2} \tag{2.1}$$

C. The measured frequency of occurrence in random sample is

$$n_i = \{color|sorted\ Index\ of\ (color) \in \{2i, 2i+1\}\} \tag{2.2}$$

D. The χ^2 statistic is given as:

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - n'_i)^2}{n'_i} \tag{2.3}$$

with $k-1$ degrees of freedom.

E. p is the probability of the statistic with the distributions of n_i and n'_i which are equal. It is calculated by integration of the density function:

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi^2_{k-1}} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1} dx \tag{2.4}$$

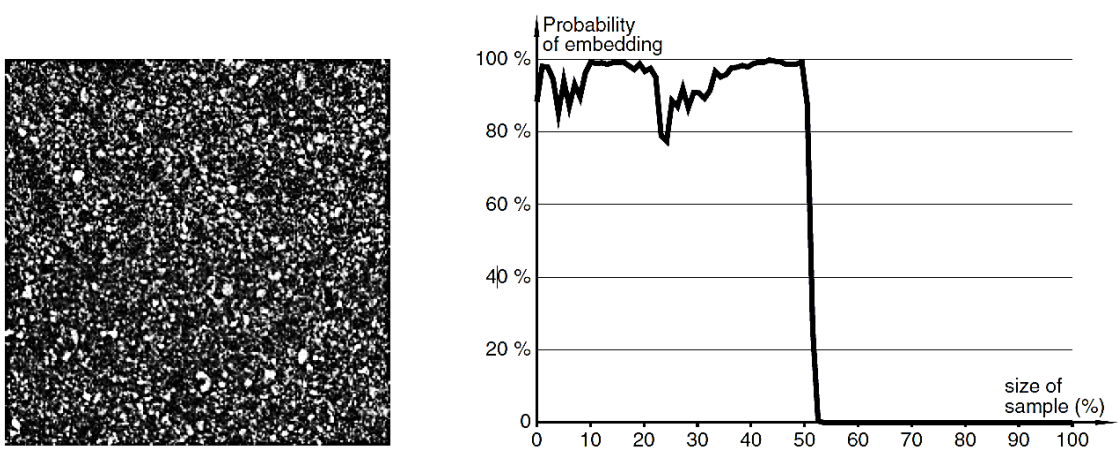


Figure 2.3 Probability of embedding with EzStego in the flooring tile image.

Source: Andreas Westfeld, Andreas Pfitzmann, "Attacks on steganographic systems, in: Proc. of Information Hiding," 3rd Third Int. Workshop, Dresden, Germany, September 28–October 1, 1999, pp. 61–75.

2.1.3 Accuracy

The diagram in Figure 2.3 presents the p-value of the Chi-square test as a function of an increasing sample. This p-value is roughly the probability of embedding. Initially, the sample comprises 1 % of the pixels, starting from the upper border. For this sample, $p = 0.8826$. The p-value increase to 0.9809 when the next sample comprises an additional 2 % of the pixels. As long as the sample comprises pixels of the upper half only, in which has been embedded, the p-value does not drop below 0.77. The pixels of the lower half of the picture are unchanged, because the message to be embedded was not such long.

2.2 Raw Quick Pair

2.2.1 Introduction

Fridrich et al. [7] introduced a powerful steganalytic technique that enables us to reliably detect the presence of a pseudo-random binary message randomly spread in a color image based on analyzing close pairs of colors created by LSB embedding. They estimated the probability of both false detections and missing a secret message.

Writers had observed that the number of unique colors for true-color images is significantly smaller than the number of pixels in an image. The ratio of the number of unique colors to the number of pixels from 1/2 for high quality scans to 1/6 or even lower for JPEG images.

This observation is very important because it means that many true-color images have a relatively small "palette". After LSB embedding, the new color palette will have a

very distinct feature.

2.2.2 Encoding

They proposed to test the presence of messages in true-color images using the following idea:

1. To find out whether or not an image has a secret message in it, calculate the ratio R between the number of all pairs of close colors P and the number of all color pairs (recall that U is the number of unique colors in the image):

$$\mathrm{R} = \frac{P}{\binom{U}{2}} \tag{2.5}$$

2. Using LSB embedding in randomly selected pixels (and channels for color $\mathrm{M} \times \mathrm{N}$ images), embed a test message of the size $\alpha 3MN$ bits. Smaller values of α will lead to faster techniques.

3. Denoting the corresponding quantities for the new image after embedding the test message as U' and P', and calculate the ratio R' for the new image with the test message.

$$\mathrm{R} = \frac{P'}{\binom{U'}{2}} \tag{2.6}$$

Obviously, if the secret message size is too small, the two ratios will be very close to each other and as a result we will not be able to distinguish images whether embedded

with messages or not.

Fridrich et al. ran the detection algorithm for both databases and tested the message presence by embedding a test message of size $\alpha \cong 1/30$. As a result, the values of R/R' were obtained for both databases. The results are shown in Figure 2.4. If the image has already had a large message hidden inside, those two ratios will be almost the same, if the image did not steganography it, we expect $R' > R$.

The dashed curve corresponds to the database of images with messages and the solid curve corresponds to the original database without messages, both after embedding the 1kB test message. To separate the two curves, we choose the threshold T_h as 1.1.

It is difference that enables to distinguish between cover images and stego images for the case of LSB steganography. The method works reliably well as long as the number of unique colors in the cover image is less than 30% of the number of pixels.

As reported, the method has higher detection rate than the method given by Westfeld [1], which is mentioned at Section 2.1. And it is possible to reliably detect the presence of secret message embedded in digital images using the LSB technique. The reliability of the detection method increases with decreasing number of unique colors in the original image. From the result, they had to notice that some high-quality scans stored losslessly may have a very high number of unique colors and the results of the detection technique may become unreliable.

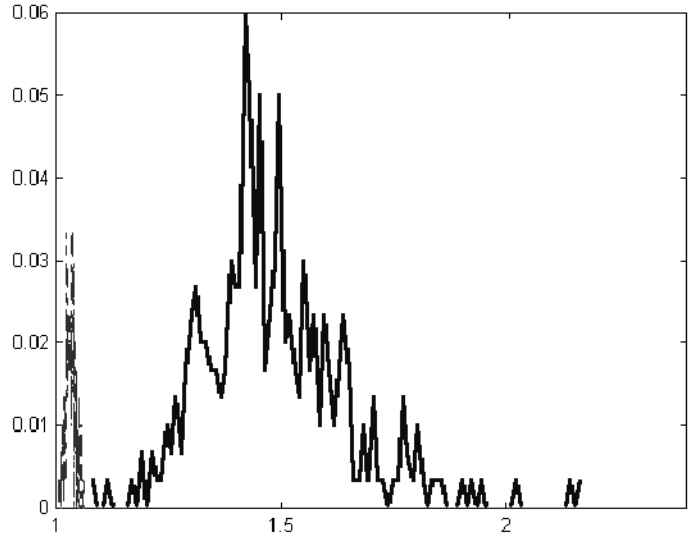


Figure 2.4 The ratio R'/R for 300 images. The thin dashed curve corresponds to images with an embedded message of length equal to 2/3 of the total available number of LSBs ($3MN$). The bold solid curve corresponds to images without any embedded messages.

Source: Jessica Fridrich, Rui Du and Long Meng, "Steganalysis of LSB Encoding in Color Images," ICME 2000, New York City, July 31-August 2, New York, USA.

2.3 RS Analysis

A more sophisticated technique Regular and Singular group (RS) steganalysis is presented by Fridrich for detection of LSB embedding in color and grayscale images in 2001 [8]. This method originated by analyzing the capacity for lossless data embedding in the LSBs.

2.3.1 Implement

Firstly, the image is divided into disjoint groups of fixed shape. Within each group noise is measured by the mean absolute value of the differences between adjacent pixels. Each group is classified as regular or singular depending on whether the pixel noise within

the group is increased or after flipping the LSBs using a mask.

Assuming the cover image with $M \times N$ pixels and its pixel values from the set P. For example, for an 8-bit grayscale image, $P = \{0, \dots, 255\}$, with n adjacent pixels $(x_1, \dots, x_n)$.

To capture the spatial correlation, a discrimination function f is defined as the mean absolute value of the differences between adjacent pixels.

$$f(x_1, \dots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \tag{2.7}$$

Then, writers defined an invertible operation F on P called flipping. Flipping will be a permutation of gray levels that entirely consists of 2-cycles. Thus, F will have the property that $F^2 = Identity$ or $F(F(x)) = x \; for \; all \; x \in P$.

$$F_1 : 0 \leftrightarrow 1, 2 \leftrightarrow 3, \dots, 254 \leftrightarrow 255 f(x_1, \dots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \tag{2.8}$$

$$F_{-1} : -1 \leftrightarrow 0, 1 \leftrightarrow 2, 3 \leftrightarrow 4, \dots, 253 \leftrightarrow 254, 255 \leftrightarrow 256 \tag{2.9}$$

$$F_0 : F_0(x) = x \tag{2.10}$$

the discrimination function f and the flipping operation F to define three types of pixel

groups: R, S and U

$$\text{Regular groups: } G \in R \Leftrightarrow f(F(G)) > f(G)$$

$$\text{Singular groups: } G \in S \Leftrightarrow f(F(G)) < f(G)$$

$$\text{Unusable groups: } G \in U \Leftrightarrow f(F(G)) = f(G).$$

From the expressions above, $F(G)$ means that the flipping function F were applied to the components of the vector $G = (x_1, \ldots, x_n)$

Fridrich defined the flipped group $F(G) = (F_{M(1)}(x_1), F_{M(2)}(x_2), \ldots, F_{M(n)}(x_n))$, where $M(i), i = 1,2, \ldots, n$ is the element of mask M which takes on the values -1, 0 and 1. The purpose of the flipping F is perturbing the pixel values in an invertible way by some small amount thus simulating the act of invertible noise adding.

Since the LSB flipping simulates the act of adding pixel noise, it more frequently results in an increase in the value of the discrimination function f rahter than a decrease. Thus the total number of the regular groups will be larger than that of singular groups. Let R_M and S_M be the realtive number of regular groups and singular groups. the expected value of R_M is equal to that of R_{-M}, and the same is true for S_M and S_{-M} :

$$R_M \cong R_{-M} \; and \; S_M \cong S_{-M} \tag{2.11}$$

The same for the relationship between S_{+M} and S_{-M}. Randomization of the LSB plane forces the difference between RM and SM to zero as the length m of the embedded message increases. After flipping the LSB of 50% of pixels, $R_M \cong S_M$ can be obtained.

Here is a simple explanation for the peculiar increase in the difference between R_{-M} and S_{-M} for the mask $M = [0\ 1\ 1\ 0]$. Writers defined sets $C_i = \{2i, 2i+1\}, i = 0, \dots, 127$, and cliques of groups $C_{rst} = \{G \mid G \in C_r \times C_s \times C_t\}$. There are 128^3 cliques, each clique consisting of 8 groups. Figure 2.5 demostrates R_M, S_M as functions of the number of pixels with flipped LSBs.

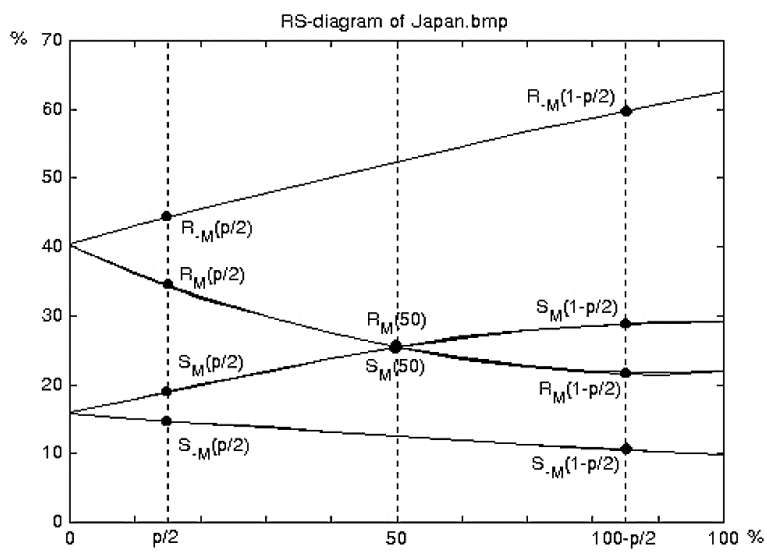


Figure 2.5 RS-diagram of an image taken by a digital camera. The x-axis is the percentage of pixels with flipped LSBs, the yaxis is the relative number of regular and singular groups with masks M and −M, M=[0 1 1 0].

Source: Jessica Fridrich, Miroslav Goljan, Rui Du, "Detecting LSB steganography in color and gray-scale images," IEEE Multimedia Magaz., Special Issue on Security 22–28, 2001.

The general shape of the four curves in the diagram varies with the cover-image

from almost perfectly linear to curved. They had collected experimental evidence that the R_{-M} and S_{-M} curves are well modeled with straight lines, while second-degree polynomialscan approximate the inner curves R_M and S_M reasonably well.

By flipping the LSBs of all pixels in the image and calculating the number of R and S groups, the four points $R_M(1-p/2), S_M(1-p/2), R_{-M}(1-p/2),$ and $S_{-M}(1-p/2)$can be calculated. The midldle points $R_M(1/2)$ and $S_M(1/2)$ will be obtained by randomizing the LSB plane of the stego image.

In this way, Fridrich fitted straight linesthrough the points $R_{-M}(p/2)$, $R_{-M}(1-p/2)$ and $S_{-M}(p/2)$, $S_{-M}(1-p/2)$. The points $R_M(p/2)$, $R_M(1/2)$, $R_M(1-p/2)$ and $S_M(p/2)$, $S_M(1/2)$ and $S_M(1-p/2)$ determin two parabolas. Each parabola and a corresponding line intersect to the left.

To estimation of the middle points by accepting two more assumptions:

1. The point of intersection of the curves R_M and R_{-M} has the same x coordinate as the point of intersection for the curves S_M and S_{-M}

2. The curves RM and SM intersect at $m = 50\%$, or $RM(1/2) = SM(1/2)$. This assumption is like saying that the lossless embedding capacity for a randomized LSB plane is zero.

Rescaling the x axis so that $p/2$ becomes 0 and $100-p/2$ becomes 1, the x-coordinate of the intersection point is a root of the following quadratic equation

$$2(d_1+d_0)\,x^2+(d_{-0}-d_{-1}-d_1-3d_0)\,x\;+d_0-d_{-0}=0 \qquad (2.12)$$

where $d_0 = R_M(p/2) - S_M(p/2)$, $d_1 = R_M(1-p/2) - S_M(1-p/2)$, $d_{-0} = R_{-M}(p/2) - S_{-M}(p/2)$, $d_{-1} = R_{-M}(1-p/2) - S_{-M}(1-p/2)$.

The message lenth p from the root x whose absolte value is smaller by

$$p = x/(x - 1/2) \tag{2.13}$$

2.3.2 Accuracy

Writers used equations above to estimate the size of the secret message which is embedded in the stego-image. Under certain assumptions the amount of embedded message could be accurately determined if the numbers of regular and singular groups are given.

The initial non-zero bias could be both positive and negative and it puts a limit on the theoretical accuracy of their steganalytic method. Smaller images tend to have higher variation in the initial bias because of the smaller number of R and S groups. Generally, color images exhibit larger variation in the initial bias than grayscales.

Writes used a small image with a short message. The test image was a scanned color photograph 422×296 and the message was a random bit sequence with 375 kb or 20% of the image full capacity (100% = 3bits per pixel). Since the initial bias is about 2.5% in each color channel the expected detected percentage of flipped pixels would be about 12.25%.

Table 2.1 Initial Bias and Estimated Number of Pixels with Flipped LSBs.

Imgae	Red(%)	Green(%)	Blue(%)
Cover Image	2.5(0.0)	2.4(0.0)	2.6(0.0)
Steganos	10.6(9.8)	13.3(9.9)	12.4(9.8)
S-Tools	13.4(10.2)	11.4(10.2)	10.3(10.2)
Hide4PGP	12.9(10.0)	13.8(10.1)	13.0(10.0)

Source: Jessica Fridrich, Miroslav Goljan, Rui Du, "Detecting LSB steganography in color and gray-scale images," IEEE Multimedia Magaz., Special Issue on Security 22–28, 2001.

The RS steganalysis is more accurate for messages that are randomly scattered in the stegoimage than for messages concentrated in a localized area of the image. To address this issue, we can apply the same algorithm to a sliding rectangular region of the image.

The experimental results obtained by RS steganalysis also provide a new estimate on safe size of secret messages embedded using LSB embedding. For high quality images from scanners and digital cameras, we estimate that messages requiring less than 0.005 bits per pixel are undetectable using RS Steganalysis. Higher bit rates are in the range of detectability using RS Steganalysis.

2.4 Histogram Attack

2.4.1 Attacking J-Steg

Yu et al. [14] propsosed a method of detecting secret message and estimating the secret message length of bitstreams embedded using J-Steg [1]. Firstly, the histogram of cover image is estimated from stego image, based on the model of statistical distribution of quantized DCT coefficients. Then the secret message is detected and the secret message length is estimated with the estimated cover histogram.

Let x denote an instance of a class of potential carrier media, such as pixel values or quantized DCT coefficients of an image. If x is treated as an instance of a random variable X, which can be discribed the probability distribution $P_X(x)$.

In order to detect whether there is a hidden message embedded or not. The detection is to perform a hypothesis test to find out whether the instance x obey the probability distribution $P_X(x)$. In order to get the length of unknown meassage, Yu and Wnag calculated the equation $S(m) = S_{stego}$ for m, where $S(m)$ is the macroscopic quantity, S_{stego} is the value of S for the stego image under investigation. In general, the function S has several undetermined parameters which can be determined by estimating some extreme values of S, such as $S(0)$. To get $S(0)$, the principle by four pixels were applied to estimate the length of hidden message.

In the JPEG compression standard, images are divided into 8×8 blocks. Each block is passed through a Discrete Cosine Transform (DCT) to produce 64 DCT coefficients and then the coefficients are quantized according to a quantization table and

encoded using an entropy encoder. These coefficients are called AC coefficients. The Laplacian distribution

$$p(x) = \frac{\lambda}{2} e^{-\lambda|x|} \tag{2.14}$$

A generalized Laplacian can still be used to fit the resulting histogram with integer width bins.

$$p(x) = \frac{p-1}{2s}\left(\left|\frac{x}{s}\right| - 1\right)^{-p} \tag{2.15}$$

There is a closed form solution for the cumulative density function which makes it easy to integrate the model density for individual histogram bins.When taking into account amore accurate estimation of the quantization effects, one would find this distribution appears to fit DCT coefficients better than the generalized Laplacian/Gaussian.

Let $H(d)$ be the histogram of cover image after embedding m pseudorandom bits in the LSBs, the histograms $H(d)$ and $h(d)$ will have relations as follow equations.

$$H(0) = h(0), H(1) = h(1) \tag{2.16}$$

$$H(2i) = h(2i) - \alpha\,[h(2i) - h(2i + 1)] \tag{2.17}$$

$$H(2i\ +1)\ =\ h(2i\ +1)\ +\alpha\ [h(2i)\ -\ h(2i\ +1)] \tag{2.18}$$

where $i = \pm 1, \pm 2, \ldots,$ $\alpha = \frac{m}{2\sum_{i\neq 0, i\neq 1} H(i)}$, Let $H_\alpha\ (i)\ =\ H(2i)\ +\ H(2i\ +1)$. $H_\alpha(i)\ =\ H(2i)\ +\ H(2i\ +1)$.

Once the model is fit to the histograms for a stego image, it is used to estimate the histogram of the cover image. Let $\hat{h}(d)$ be the estimated histogram of cover image.

$$\hat{h}(0) = H(0), \hat{h}(1) = H(1) \tag{2.19}$$

$$\hat{h}(0) = H(0), \hat{h}(1) = H(1) \tag{2.20}$$

$$\hat{h}(2i+1) = H_\alpha(i)\frac{P(2i+1)}{P_\alpha(i)} \tag{2.21}$$

where $P_\alpha(i) = \int_{2i-0.5}^{2i+1.5} p(x)dx$.

Figure 2.6 shows the original coefficient histogram of an image and the estimated histogram after message embedding. The coefficients are all AC coefficients. It can be seen from the figure that we can almost exactly estimate the histogram of cover image from a stego image.

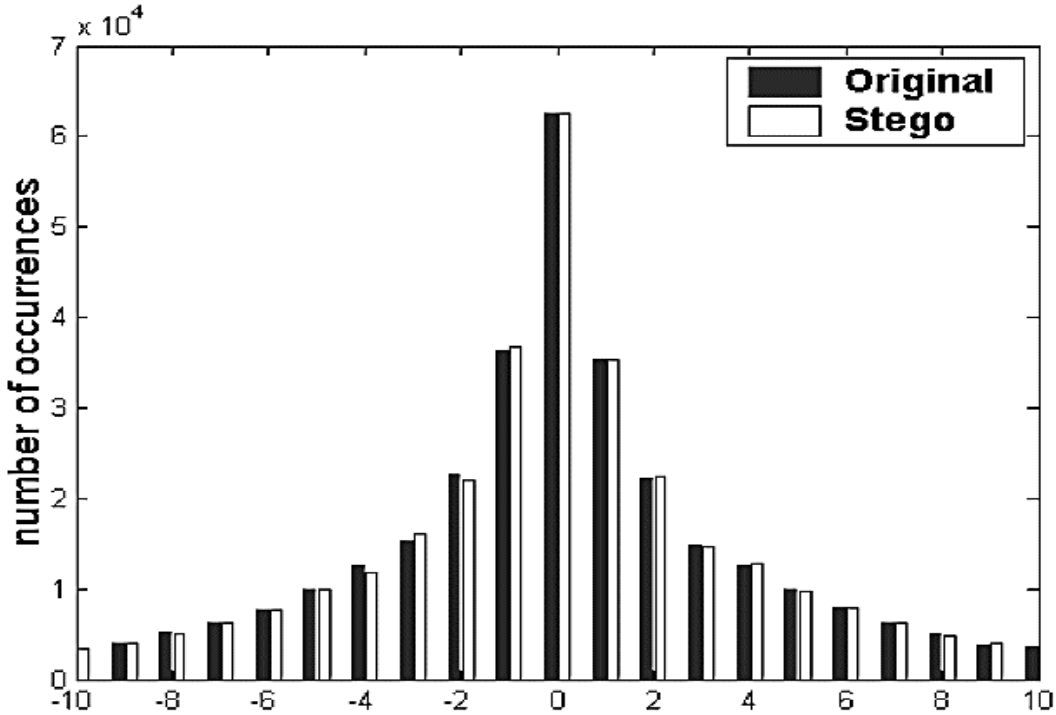


Figure 2.6 Comparsion of coefficient histograms between original and stego.

Source: Xiaoyi Yu, Yunhong Wang, Tieniu Tan, "On Estimation of Secret Message Length in JSteg-like Steganography," ICPR 2004. Proceedings of the 17th International Conference on Vol. 4 DOI: 10.1109/ICPR.2004.1333862

Since Yu, Wang and Tan estimated the histogram of the cover image, the detection of hidden message and estimation of hidden message length becomes easy. The detection is determined by using the Chi-square test.

The χ^2 statistic is given as $\chi^2 = \sum_{i=\pm 1}^{\pm k} \frac{\left(\hat{h}(i) - H(i)\right)^2}{\hat{h}(i)}$ with $2^k - 1$ degree of freedom. Writers can perform Chi-square test at significance level α and $2^k - 1$ degree of freedom to decide whether a suspect images contains secret message or not.

By calculating α the following equation, which is derived from Equtation 2.19-2.21, is used.

$$\alpha = \arg min \sum_{i=\pm 1}^{\pm k} \left(H(2i) - (1-\alpha)\hat{h}(2i) - \alpha\hat{h}(2i+1)\right)^2 \tag{2.22}$$

$$\alpha = \frac{\sum_{i=\pm 1}^{\pm k}\left(H(2i) - \hat{h}(2i))(\hat{h}(2i) - \hat{h}(2i+1)\right)}{\sum_{i=\pm 1}^{\pm k}\left(\hat{h}(2i) - \hat{h}(2i+1)\right)^2} \tag{2.23}$$

where k is the maximum quantized DCT AC coefficient. Thus the length of unknown message can be calculated as

$$M = 2\alpha \sum_{i\neq 0, i\neq 1} H(i) \tag{2.24}$$

from the experiment result this estimation is more accurate than Fridrichs cropping method.

2.4.2 Attacking F5

The F5 steganographic algorithm was introduced by Westfel in 2002 [20]. Fridrich and Goljan divided thier attack into two separate parts:

(1) Finding distinguishing statistical quantities T that correlate with the number of modified coefficients.

(2) Determining the baseline values of the statistics T.

When anaylzing the changes in the histogram by F5 Algorithm, Let $h(d), d =$

$0, 1, \ldots$ be the total number of AC coefficients in the cover-image with absolute value equal to d after the image has been compressed inside the F5 algorithm. Set $h_{kl}(d)$ as the total number of AC DCT coefficients corresponding the frequency (k, l),The corresponding histogram values for the stego-image will be denoted using the capital letters H and $H_{k,l}$.

If there are totally n non-zero AC coefficients to be modified during the embedding process, the number of relative modification od DCT coefficients would be $\beta = n/P$, where $P = h(1) + h(2) + \cdots$. The expected valuse of the histogram of stego-image are:

$$H_{kl}(d) = \begin{cases} (1-\beta)h_{kl}(d) + \beta h_{kl}(d+1) & for\ d > 0 \\ h_{kl}(0) + \beta h_{kl}(1) & for\ d = 0 \end{cases} \tag{2.25}$$

Because the first two values in the histogram ($d = 0$ and $d = 1$) experience the largest change during embeddin, β is the value that minimizes the square error between the stego-image histogram H_{kl}, and the expected values $H_{kl}(d)$ calculated from the estimated histogram $\hat{h}_{kl}$:

$$\begin{aligned} \beta_{kl} = \arg min \left[H_{kl}(0) - \hat{h}_{kl}(0) - \beta \hat{h}_{kl}(1)\right]^2 \\ + \left[H_{kl}(1) - (1-\beta)\hat{h}_{kl}(1) - \beta \hat{h}_{kl}(2)\right]^2 \end{aligned} \tag{2.26}$$

The least square approximation leads to the following formula:

$$\beta_{kl} = \frac{\hat{h}_{kl}(1)\left[H_{kl}(0) - \hat{h}_{kl}(0)\right] + \left[H_{kl}(1) - \hat{h}_{kl}(1)\right]\left[\hat{h}_{kl}(2) - \hat{h}_{kl}(1)\right]}{\hat{h}_{kl}^2(1) + \left[\hat{h}_{kl}(2) - \hat{h}_{kl}(1)\right]^2} \tag{2.27}$$

where the final value of the parameter β is calculated as an average over selected low frequency DCT coefficients $(k, l) \in \{(1,2), (2,1), (2,2)\}$.

According to their experiments, the estimated histogram is quite close to the histogram of the original image. We provide a simple heuristic explanation of why the method for obtaining the baseline histogram values is indeed plausible.

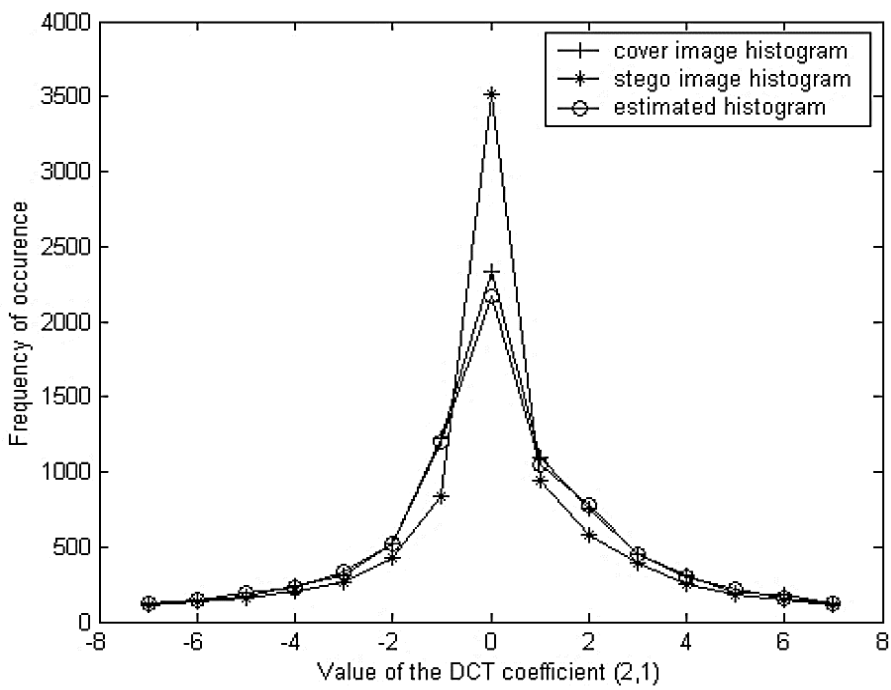


Figure 2.7 Effect of F5 on the histogram of DCT coefficient(2,1).

Source: Jessica Fridrich, Miroslav Goljan, Dorin Hogea, "Steganalysis of JPEG Images: Breaking the F5 Algorithm," 5th Information Hiding Workshop, Noordwijkerhout, The Netherlands, 7-9 October 2002, pp. 310-323.

In fact, unless the quality factor of the JPEG compression is too low (e.g., lower than 60), the stegoimage produced by F5 is still very close to the cover-image both visually and using measures, such as the PSNR. The spatial shift by 4 pixels effectively breaks the structure of quantized DCT coefficients and subsequent low-pass filtering helps to reduce any spurious frequencies due to discontinuities at block boundaries. Thus, it is not surprising that the statistical properties of DCT coefficients are similar to those of the cover-image.

Figure 2.7 shows a typical example of how good the histogram estimate is when compared to the histogram of the original image. The graph illustrates the original histogram values $h_{21}(d)$ (crosses), histogram values after applying the F5 algorithm with maximal possible message, or $\beta = 0.5$ (stars), and the estimate of the original histogram (circles). It is found that the largest change in histogram values occur in the first two values($d = 0$ and $d = 1$).

Once the raltive number of changes β has been determined, the stego image can be distinguished from the cover image.

2.5 Markov Model

2.5.1 1st Order Markov Features

In this thenique, steganalysis is considered as a task of two-class pattern recognition. [21] Firstly, Shi et al. choose to work on difference JPEG 2-D arrays formed from the magnitudes of JPEG quantized block DCT coefficients. Those four direction difference

JPEG 2-D arrays are used to enhance changes caused by JPEG steganography. Then markov process is applied to modeling these arrays. In addition to reduce the computation computational, a thresholding technique is developed.

Denote the JPEG 2-D array generated from a given test image by $F(u,v), u \in [1, S_u], v \in [1, S_v]$, where S_u is the size of the JPEG 2-D array in horizontal direction and S_v in vertical direction. Then, the difference arrays are generated by the following formulae:

$$F_h(u,v) = F(u,v) - F(u+1,v) \tag{2.28}$$

$$F_v(u,v) = F(u,v) - F(u,v+1) \tag{2.29}$$

$$F_d(u,v) = F(u,v) - F(u+1,v+1) \tag{2.30}$$

$$F_{md}(u,v) = F(u+1,v) - F(u,v+1) \tag{2.31}$$

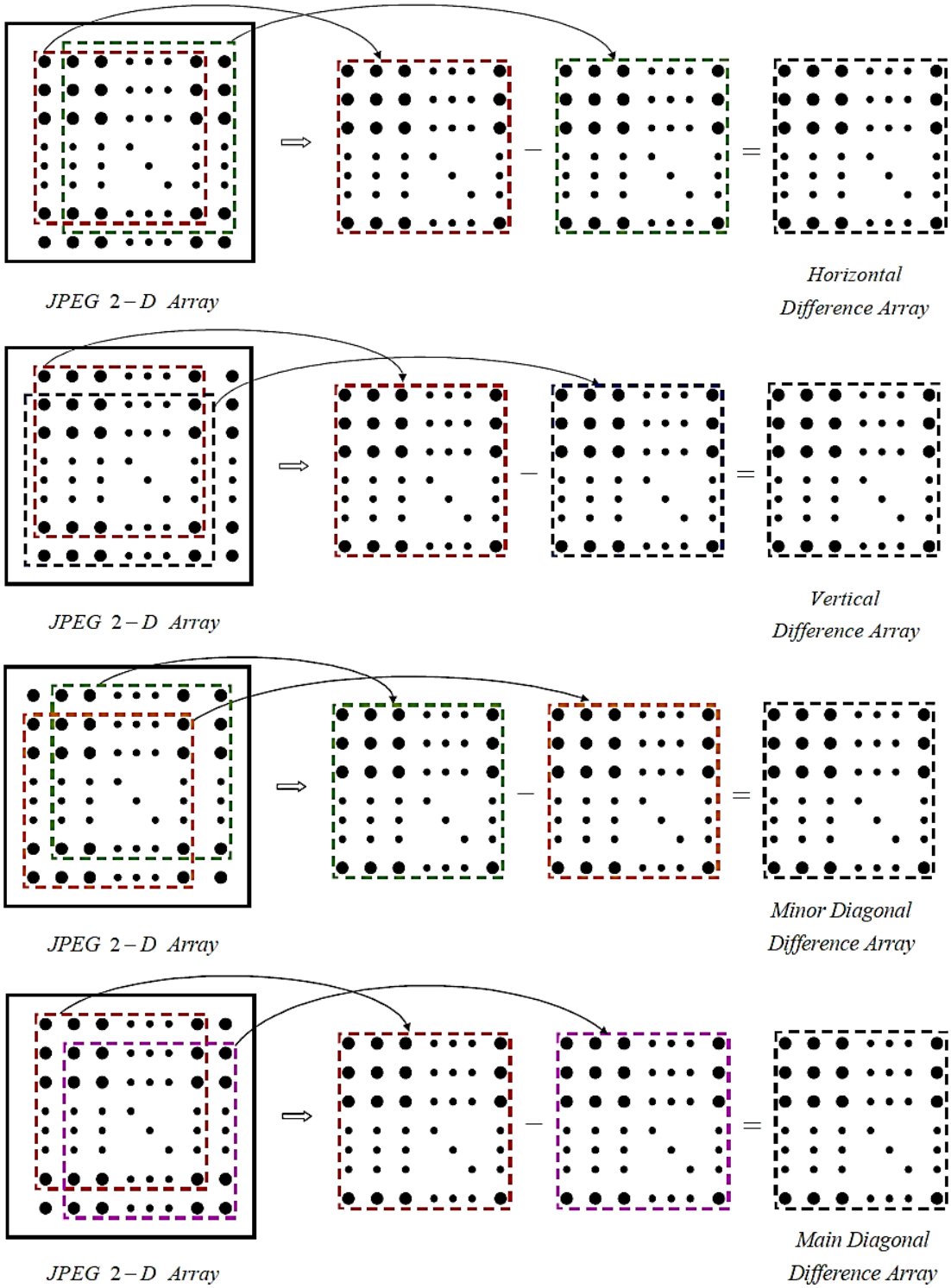


Figure 2.8 The generation of four difference JPEG 2-D arrays.

Source: Yun Q. Shi, Chunhua Chen, Wen Chen, "A Markov Process Based Approach to Effective Attacking JPEG Steganography", 8th International Workshop, IH 2006, Alexandria, VA, USA, July 10-12, 2006, pp 249-264.

Most of the difference values are close to zero. As their experimental works reported, an image set consisting of 7560 JPEG images with quality factors ranging from 70 to 90 is used. The arithmetic average of the histograms of the horizontal difference JPEG 2-D arrays generated from this JPEG image set and the histogram of the horizontal difference JPEG 2-D array generated from a randomly selected image from the set.

It is observed that most elements in the horizontal difference JPEG 2-D arrays fall into the interval [-T, T] as long as T is large enough. The values of mean and standard deviation of percentage number of elements of horizontal difference JPEG 2-D arrays for the image set falling into [-T, T] when T = {1, 2, 3, 4, 5, 6, 7} are shown in Table 2.2.

Table 2.2 Mean of Percentage Numbers of Elements of Horizontal Difference JPEG 2-D Arrays Falling with [-T,T]

	[-1,1]	[-2,2]	[-3,3]	[-4,4]*	[-5,5]	[-6,6]	[-7,7]
Mean	84.72	88.58	90.66	91.99	92.92	93.60	94.12

Source: Yun Q. Shi, Chunhua Chen, Wen Chen, "A Markov Process Based Approach to Effective Attacking JPEG Steganography," 8th International Workshop, IH 2006, Alexandria, VA, USA, July 10-12, 2006, pp. 249-264.

Threshold value T means that only those elements in the difference JPEG 2-D arrays whose value falls into $\{-T, -T+1, \ldots, -1, 0, 1, \ldots, T-1, T\}$ will be considered. If an element whose value is either larger than T or smaller than $-T$, it will be represented by T or $-T$ correspondingly. This procedure results in a transition probability matrix of

dimensionality $(2T+1)\times(2T+1)$.

In total, since they set T as 4, depending on their experimental works, $4\times(2T+1)\times(2T+1)=324$ elements will be calculated.

The feature construction procedure is summarized in Figure 2.9.

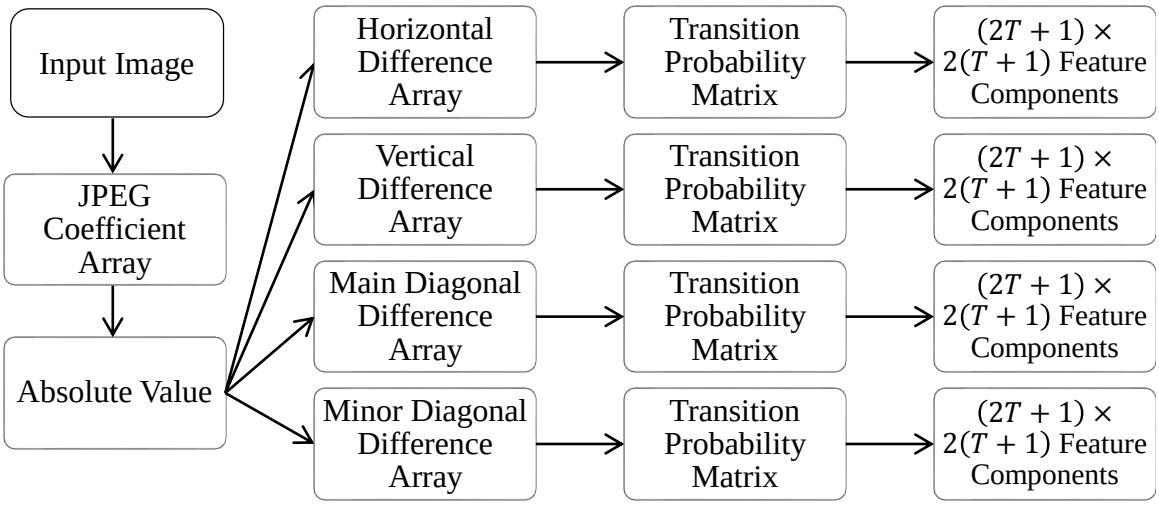


Figure 2.9 The block diagram of the feature formation procedure.

Source: Yun Q. Shi, Chunhua Chen, Wen Chen, "A Markov Process Based Approach to Effective Attacking JPEG Steganography," 8th International Workshop, IH 2006, Alexandria, VA, USA, July 10-12, 2006, pp. 249-264.

Table 2.3 Performance Comparison

	bpc	Farid's [37]	Shi et al.'s [38]	Fridrich's [39]	324D's
F5	0.4	63.9	74.3	92.8	96.8
MB1	0.5	59.4	77.1	84.8	99.1

Source: Yun Q. Shi, Chunhua Chen, Wen Chen, "A Markov Process Based Approach to Effective Attacking JPEG Steganography," 8th International Workshop, IH 2006, Alexandria, VA, USA, July 10-12, 2006, pp. 249-264.

As a result, this steganalyzer beated out other competitors by more than 20% at bpc 0.4 by F5 algrithm and got 99.1% in bpc 0.4 by MB1, all along nobody can achieve so high successful rate.

2.5.2 Prediciton-Error in Non-JPEG Images

A steganalysis method based on 2-D Markov chain of thresholded prediction-error image is proposed by Zou et al. in 2006 [16] – the same year 324D method was presented. The prediction errors are extracted from empirical transition matrices by a threshold technique: pixels are predicted by their neighboring pixels and the prediction-error image is generated by subtracting the prediction value from the pixel value and then through a predefined threshold.

These features are evaluated with Support Vector Machines (SVM) [32]. SVM with both linear and non-linear kernels are used as classifier. The non-linear SVM performs much better than linear SVM for proposed higher-dimensional features. It has been reported by the author that the results are more effective than Fridrich‘s.

Zou et al. [15] used neighboring pixels to predict the current pixel. The predictions are made in three directions: horizontal, vertical and diagonal since a digital image is actually a 2-D array. For each prediction the error can be obtained by subtracting the predicted pixel value from the original pixel value as following,

$$e_h(i,j) = x(i+1,j) - x(i,j) \tag{2.32}$$

$$e_v(i,j) = x(i,j+1) - x(i,j) \tag{2.33}$$

$$e_d(i,j) = x(i+1,j+1) - x(i,j) \tag{2.34}$$

where $e_h(i,j)$ indicates the prediction error for pixel (i,j) along horizontal direction while $e_v(i,j)$ and $e_d(i,j)$ is the prediction error for pixel (i,j) on vertical and diagonal directions.

A Markov chain is a random process that undergoes transitions from one state to another state space. It is required to possess a property called memoryless: the probability distribution of the next state only depends on the current state and not on the sequence of events that preceded it.

A Markov chain is a sequence of random variables $X_1, X_2, X_3, \ldots$ with the Markov proper, given the present state, the future and past states are independent. Formally,

$$\begin{aligned} P(X_{n+1} = x | X_1 = x_1, X_1 = x_2, , \ldots, X_n = x_n,) \\ = P(X_{n+1} = x | X_n = x_n,) \end{aligned} \tag{2.35}$$

A stochastic matrix describes a Markov chain X_t over a finite state space S.

If the probability of moving from i to j in one time step is $P(j|i) = P_{i,j}$, the stochastic matrix P is given by using $P_{i,j}$ as the ith row and j^{th} column element, like:

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,j} & \cdots \\ p_{2,1} & p_{2,2} & \cdots & p_{2,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ p_{i,1} & p_{i,1} & \cdots & p_{i,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \tag{2.36}$$

This matrix is a right stochastic matrix, so that $\sum_j P_{i,j} = 1$.

The probability of transitioning from i to j in n steps is given by the $(i,j)^{\text{th}}$ element of the square of P described as:

$$P = (P^n)_{i,j} \tag{2.37}$$

A 2-D Markov chain model is applied to the thresholded prediction error images. Figure 2.10-2.12 display the transition model for horizontal, vertical and diagonal prediction error an image of size 8 by 8. The arrows represent the changing of state in Markov chain. And the elements of the transition matrices are served as features for steganalysis.

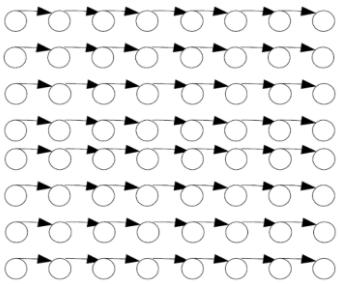


Figure 2.10 Transition model for prediction-error image E_h.

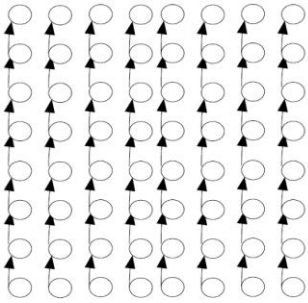


Figure 2.11 Transition model for prediction-error image E_v.

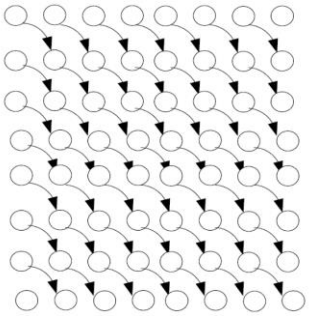


Figure 2.12 Transition model for prediction-error image E_d.

Source: Dekun Zou, Yun Q. Shi, Wei Su, Guorong Xuan, "Steganalysis Based on Markov Model of Thresholded Prediction-error Image," Multimedia and Expo, 2006 IEEE International Conference on DOI: 10.1109/ICME.2006.262792

Compared with Sullivan et al.'s [33] scheme, the detection rate of Markov Model of prediction-error is absolutely higher. When detecting LSB in 0.3 bpp with linear SVM, this technique obtains 92.92% accuracy. However, Sullivan et al.'s method gain 65.68%. The result displayed in Table 2.4 and Table 2.5

Table 2.4 Results of Two Steganalysis Method with Linear SVM

Embedding Method	Zou et al.'s	Sullivan et al.'s
Cox's SS	80.58%	75.81%
Piva's SS	88.57%	76.34%
LSB(0.1 bpp)	77.27%	53.73%
LSB(0.2 bpp)	88.27%	60.15%
LSB(0.3 bpp)	92.91%	65.68%

Table 2.5 Results of Two Steganalysis Method with Non-Linear SVM

Embedding Method	Zou et al.'s	Sullivan et al.'s
Cox's SS	89.15%	77.60%
Piva's SS	94.10%	77.58%
LSB(0.1 bpp)	86.30%	49.82%
LSB(0.2 bpp)	94.45%	61.13%
LSB(0.3 bpp)	97.75%	68.98%

Source: Dekun Zou, Yun Q. Shi, Wei Su, Guorong Xuan, "Steganalysis Based on Markov Model of Thresholded Prediction-error Image," Multimedia and Expo, 2006 IEEE International Conference on DOI: 10.1109/ICME.2006.262792

2.5.3 Multi-Directional JPEG Attack

Xuan et al. [31] also presented a scheme based on Markov process in 2007. They

modeled the 2-D JPEG coefficient array by using Markov model. There is three different scanning orders – zigzag, horizontal and vertical. Unlike single direction scanning, multi-direction scanning can more effectively catch the change and thus provides better performance.

$$\begin{bmatrix} 0 & 1 & 5 & 6 & 14 & 15 \\ 2 & 4 & 7 & 13 & 16 & \\ 3 & 8 & 12 & 17 & & \\ 9 & 11 & 18 & & & \\ 10 & 19 & & & & \\ 20 & & & & & \end{bmatrix}$$

Figure 2.13 Zigzag scanning order.

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 10 & 9 & 8 & 7 & 6 & \\ 11 & 12 & 13 & 14 & & \\ 17 & 16 & 15 & & & \\ 18 & 19 & & & & \\ 20 & & & & & \end{bmatrix}$$

Figure 2.14 Horizontal scanning order.

$$\begin{bmatrix} 0 & 10 & 11 & 17 & 18 & 20 \\ 1 & 9 & 12 & 16 & 19 & \\ 2 & 8 & 13 & 15 & & \\ 3 & 7 & 15 & & & \\ 4 & 6 & & & & \\ 5 & & & & & \end{bmatrix}$$

Figure 2.15 Vertical scanning order.

Source: Guorong Xuan, Xia Cui ; Shi, Y.Q. ; Wen Chen ; Xuefeng Tong ; Cong Huang, "JPEG Steganalysis Based on Classwise Non-principal Components Analysis and Multi-directional Markov Model," Multimedia and Expo, 2007 IEEE International Conference on, 2-5 July 2007, pp. 903-906, DOI:10.1109/ ICME.2007.4284797

where the numbers 0, 1, …, 20 represent the sequence of the low-frequency coefficients.

In the JPEG 8×8 DCT blocks, most of high frequency coefficients after quantization are zero, whereas low frequency AC coefficients are often non-zero and utilized by JPEG steganography.

Therefore, low frequency coefficients are scanned to generate three coefficient sequences, each consisting of the DC coefficient and the first twenty low-frequency AC coefficients.

In 2-D Markov chain of thresholded prediction-error image, the distortions introduced by data hiding are usually small comparing to the presence of different objects. Otherwise, the distortion will raise alarm when inspected by human eyes. Therefore, a predefined threshold T is adopted and the prediction errors are adjusted according to the following rule.

$$e(i,j) = \begin{cases} e(i,j) & |e(i,j)| < T \\ 0 & |e(i,j)| > T \end{cases} \tag{2.38}$$

Large prediction errors are regarded as 0. At this point, the range of the prediction-error image are limited to $[-T, T]$, which means only $2 \times T + 1$ values left.

Since the dynamic range of JPEG coefficient is large, the dimension of transition matrix is non-trivial. In order to reduce complexity, they also proposed that make a threshold to elements in the coefficient sequence with selecting value T.

Table 2.6 Percentage of AC Coefficients in $[-T, T]$

$[-T, T]$	[-5,5]	[-6,6]	[-7,7]	[-8,8]	[-9,9]
Percentage	97.3	98.1	98.6	98.9	99.2

Source: Guorong Xuan, Xia Cui ; Shi, Y.Q. ; Wen Chen ; Xuefeng Tong ; Cong Huang, "JPEG Steganalysis Based on Classwise Non-principal Components Analysis and Multi-directional Markov Model," Multimedia and Expo, 2007 IEEE International Conference on, 2-5 July 2007, pp. 903-906, DOI:10.1109/ ICME.2007.4284797

Table 2.6 shows that most JPEG coefficients are falling into the selected threshold arrange, indicating that the information loss is negligible for these threshold values.

With the same quality factor, the length of embedded message is 0.04bpp. To evaluate the performance of the proposed approach, the 1096 BMP images with size of 768x512 embedded with F5, MB1 and MB2, Xuan et al.

Detection accuracy of Xuan et al.'s method is displayed in Table 2.7.

Table 2.7 Detection Accuracy

Payload	Fridrich[39]	Xuan et al.
F5	87%	92%
OG	97%	100%
MB1	86%	97%
MB2	84%	99%

Source: Guorong Xuan, Xia Cui ; Shi, Y.Q. ; Wen Chen ; Xuefeng Tong ; Cong Huang, "JPEG Steganalysis Based on Classwise Non-principal Components Analysis and Multi-directional Markov Model," Multimedia and Expo, 2007 IEEE International Conference on, 2-5 July 2007, pp. 903-906, DOI:10.1109/ ICME.2007.4284797

2.6 Co-occurrence Matrix

2.6.1 Coefficient Difference Features

In 2013, Li et al. [9] proposed scheme employs 15700 dimensional features calculated from the co-occurrence matrices of DCT. This algorithm is comprised of two parts: feature extraction and Bayesian ensemble classifier. In the first part, they calculated a high-dimensional feature vector generated from each JPEG image in a training set which contains original and stego samples. In the second part, a group of sub-classifiers trained on those feature vectors is integrated to make optimized decisions for suspicious images by Bayesian mechanism [25].

The features extraction also include two parts: one part is generated from the coefficient co-occurrence matrices, which are proposed by Kodovsky et al. [18]. While another part is derived from the co-occurrence matrices of coefficient differences, both the coefficient features and the difference features will contribute to the steganalysis.

Kodovsky et al. [18] designed 7850-dimensional features which extracted from the co-occurrence matrices of DCT coefficient pairs. Since both the intra-block and inter-block dependencies are represented by the features, the steganalysis method can effectively detect the hidden data in JPEG images.

Li et al. defined the differences of adjacent coefficients along the horizontal, vertical, diagonal and minor diagonal directions:

$$d_{m,n}^{(h)}(u,v) = c_{m,n}(u,v) - c_{m,n}(u+1,v) \tag{2.39}$$

$$d_{m,n}^{(v)}(u,v) = c_{m,n}(u,v) - c_{m,n}(u,v+1) \tag{2.40}$$

$$d_{m,n}^{(d)}(u,v) = c_{m,n}(u,v) - c_{m,n}(u+1,v+1) \tag{2.41}$$

$$d_{m,n}^{(m)}(u,v) = c_{m,n}(u+1,v) - c_{m,n}(u,v+1) \tag{2.42}$$

In order to lower the complexity, they change values of DCT coefficient and values of DCT difference into $[-T,T]$.

$$\bar{c}_{m,n}(u,v) = \begin{cases} T, & if\ c_{m,n}(u,v) \geq T \\ c_{m,n}(u,v), & if -T < c_{m,n}(u,v) < T \\ -T, & if\ c_{m,n}(u,v) \leq T \end{cases} \tag{2.43}$$

$$\bar{d}_{m,n}^{(s)}(u,v) = \begin{cases} T, & if\ d_{m,n}^{(s)}(u,v) \geq T \\ d_{m,n}^{(s)}(u,v), & if -T < d_{m,n}^{(s)}(u,v) < T \\ -T, & if\, d_{m,n}^{(s)}(u,v) \leq T \end{cases} \tag{2.44}$$

$[\bar{c}_{m,n}(u,v), \bar{c}_{m+\Delta m,,n+\Delta n}(u+\Delta u, v+\Delta v)]$ and $[\bar{d}_{m,n}^{(s)}(u,v), \bar{d}_{m+\Delta m,n+\Delta n}^{(s)}(u+\Delta u, v+\Delta v)]$ are the coefficient pair and difference pair for the index block (m,n), coefficient position (u,v) and offset $(\Delta u, \Delta v, \Delta m, \Delta n)$.

Li et al. calculated the co-occurrence matrices of both coefficient pairs and difference

pairs according to the same 157 patterns. Which implies each pattern corresponds to a co-occurrence matrix,

$$\boldsymbol{C}_{u,v,\Delta u,\Delta v,\Delta m,\Delta n}(x,y) = \frac{1}{M \cdot N}\sum_{m=1}^{M}\sum_{n=1}^{N}\delta\left[\bar{c}_{m,n}(u,v) = x, \bar{c}_{m+\Delta m,,n+\Delta n}(u+\Delta\, u, v+\Delta\, v) = y\right] \quad (2.45)$$

$$\boldsymbol{D}^{(s)}{}_{u,v,\Delta u,\Delta v,\Delta m,\Delta n}(x,y) = \frac{1}{M \cdot N}\sum_{m=1}^{M}\sum_{n=1}^{N}\delta\left[\bar{d}^{(s)}_{m,n}(u,v) = x, \bar{d}^{(s)}_{m+\Delta m,n+\Delta n}(u+\Delta\, u, v+\Delta\, v) = y\right] \quad (2.46)$$

Since all DCT coefficients and coefficient difference are truncated to [-3, 3], there are 49 elements in $\mathbf{C}$ and $\mathbf{D}$. Then fold the two kinds of matrices:

$$\bar{\mathbf{M}}(x,y) = \mathbf{M}(x,y) + \mathbf{M}(y,x) \quad (2.47)$$

where $\mathbf{M} \in \{\mathbf{C}, \mathbf{D}\}$.

What's more, they used Cartesian calibration method [41] to produce other 7850 features. Hence a total of 15700 high dimensional features firstly are used to train for steganalysis.

Since there are a total of 15700 high dimensional feature set used for steganalysis. The extracted features firstly are used to train a number of sub-classifiers, which are integrated as an ensemble classifier with a Bayesian mechanism. In construction of each sub classifier d features from 15700 are used to train Fisher linear discriminate (FLD)

classifier [42] by using the *N* sub-vectors.

Threshold in the FLD classifier is determined by the minimal sum of probabilities of false alarm and miss detection. By using the searching algorithm, they found 600 and 201 sub classifiers obtained with different subset of features is the optimal value.

In the experiment, Li et al. used the proposed steganalytic scheme to detect the secret data embedded by two steganographic methods nsF5 and model-based steganography (MBS) [42].

As a result, the error of merging the two sets of features is only 3.3%, in contrast, the error of CF* is 6.5% at payload 0.10 bpac. On the other hand, by replacing the majority-voting mechanism with the Bayesian mechanism, can be lowered by 0.4%–1% (from 96.7% to 95.9%). That means the proposed scheme benefits from both the merged features and the Bayesian mechanism.

2.6.2 Rich Models

Fridrich et al. [10] proposed a general method for steganalysis of digital images in 2012, which based on the concept of a rich model consisting of a large number of diverse submodels. The submodels consider various types of relationships among neighboring samples of noise residuals obtained by linear and non-linear filters with compact supports. They made the model assembly by a part of the training process driven by samples drawn from the corresponding cover and stego sources. In order to increase the detection accuracy, they also apply a submodel-selection to adopt different steganographic techniques: HUGO, edge adaptive algorithm by Luo et al. [43], and optimally coded ternary ±1 embedding.

Ensemble classifiers are used to assemble the model as the steganalyzer because of their low computational complexity and ability to efficiently work with high-dimensional feature spaces and lager training sets.

Rich model focuses on the spatial domain because the best detection is usually achieved by building the model directly in the domain where the embedding changes are localized. The rich model steps shows as following.

A. Residual Images

Fridrich et al. formed the model by merging *many smaller submodels* instead of a single model because the single model will not produce a good results as the enlarged model will have too many underpopulated bins.

1) **Computing Residuals**: The submodel are formed from noise residuals, $\mathrm{R} = (\mathrm{R}_{ij}) \in \mathbb{R}^{n_1 \times n_2}$, computed using high-pass filters of the following form:

$$R_{ij} = \hat{X}_{ij}(\mathcal{N}_{ij}) - cX_{ij} \tag{2.48}$$

where $c \in \mathbb{N}$ is the residual order, $\mathcal{N}_{ij}$ is a local neighborhood of pixel X_{ij}, $X_{ij} \notin \mathcal{N}_{ij}$, and $\hat{X}_{ij}(\cdot)$ is a predictor of cX_{ij} defined on $\mathcal{N}_{ij}$. The set $\{X_{ij}, \mathcal{N}_{ij}\}$ is called the support of the residual.

All residuals used in the rich model are shown in Figure 2.16. They are built as locally supported linear filters whose outputs are possibly combined with minimum and maximum operators to increase their diversity.

If there are two or more different symbols other than the black dot, we can call it 'minmax'. While in type 'spam', the residual is computed as a linear high-pass filter of neighboring pixels with the corresponding coefficients. For example: 2a stands for the second-order $R_{ij} = X_{i,j-1} + X_{i,j+1} - 2X_{ij}$ and 1a for the first-order $R_{ij} = X_{i,j+1} - X_{ij}$. 2b is obtained as $R_{ij} = min\{X_{i,j-1} + X_{i,j+1} - 2X_{ij}, X_{i-1,j} + X_{i+1,j} - 2X_{ij}\}$.

The 'min' and 'max' operators introduce non-linearity into the residuals and increase the model diversity. All operations make the distribution of the residual samples non-symmetrical, thickening one tail of the distribution of R_{ij}.

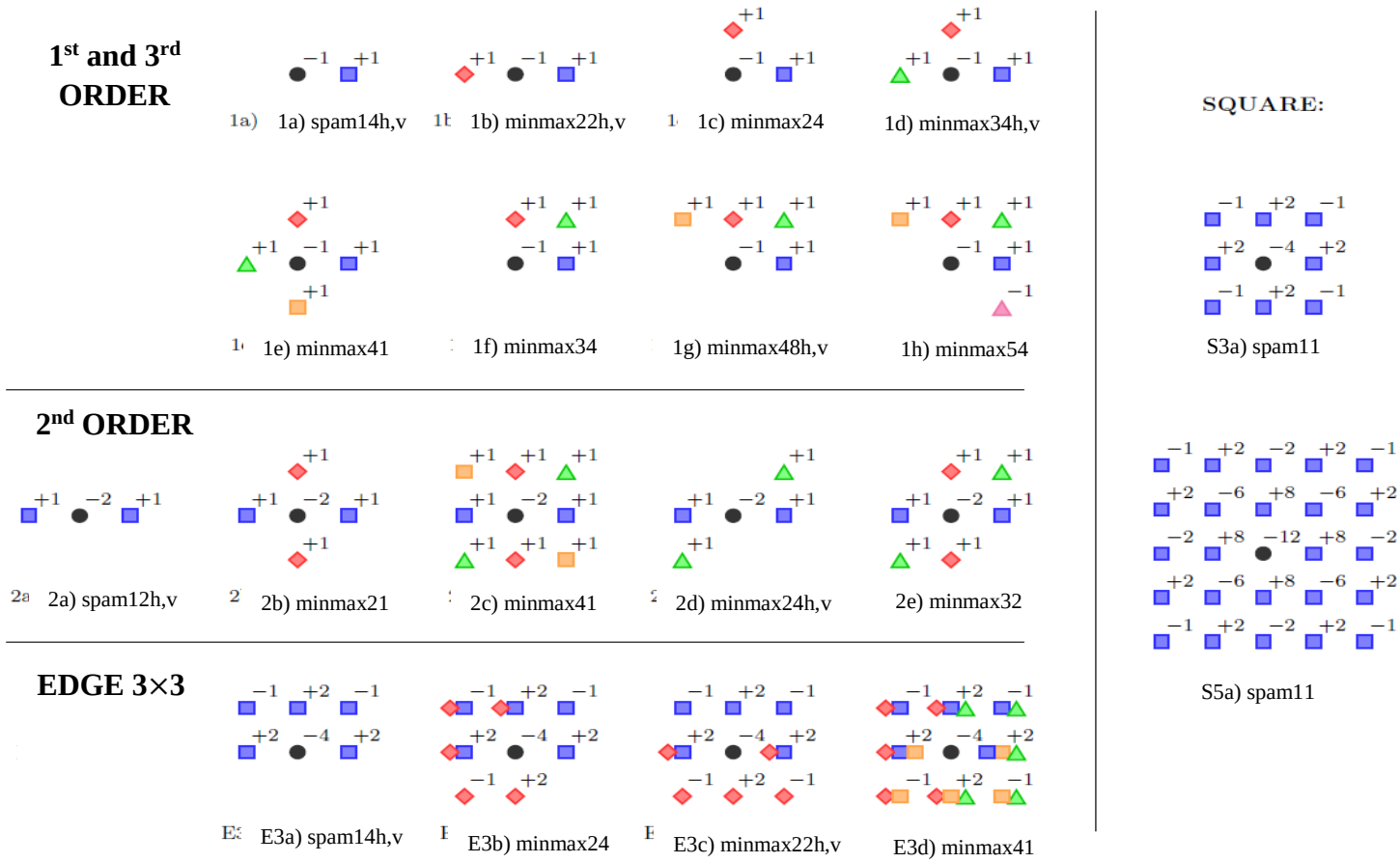


Figure 2.16 Definitions of all residuals. The residuals 3a – 3h are defined similar to the first-order residuals, while E5a – E5d are similar to E3a – E3d defined using the corresponding part of the 5 × 5 kernel displayed in S5a.

Source: Jessica Fridrich and Jan Kodovský, "Rich Models for Steganalysis of Digital Images", IEEE Trans. on Info. Forensics and Security, vol. 7(3),2012, pp. 868-882.

2) Truncation and Quantization: Each submodel is formed from a quantized and truncated version of the residual:

$$R_{ij} \leftarrow \mathrm{trunc}_T\left(\mathrm{round}\left(\frac{R_{ij}}{q}\right)\right) \tag{2.49}$$

where q > 0 is a quantization step.

The authors acknowledge that the individual performance of each submodel can likely be improved by replacing the simple scalar quantizer with an optimized design.

To select the quantization step q.

$$q \in \begin{cases} \{c, 1.5c, 2c\} & \text{for } c > 1 \\ \{1,2\} & \text{for } c = 1 \end{cases} \tag{2.50}$$

3) Co-Occurrences: Those submodels will be constructed from horizontal and vertical co-occurrences of four consecutive residual samples. Formally, each co-occurrence matrix **C** is a four-dimensional array indexed with $\boldsymbol{d} = (d_1, d_2, d_3, d_4,) \in \mathcal{T}_4 \triangleq \{-T, \ldots, T\}^4$, which gives the array $(2T + 1)4 = 625$ elements.

The $\boldsymbol{d}$th element of the horizontal co-occurrence for residual $\boldsymbol{R} = (R_{ij})$ means the number of groups of four neighboring residual samples with values equal to d_1, d_2, d_3, d_4:

$$\boldsymbol{C}_{\boldsymbol{d}}^{(h)} = \frac{1}{Z}\left|\left(R_{ij}, R_{i,j+1}, R_{i,j+2}, R_{i,j+3}\right)\middle| R_{i,j+k-1} = d_k, k = 1, \dots, 4\right| \tag{2.51}$$

where Z is the normalization factor ensuring that $\sum_{d \in \mathcal{T}_4} C_d^{(h)} = 1$.

Individual submodels of the rich image model obtained by 78 co-occurrence matrices shown in Figure 2.16 by leveraging symmetries of natural images. Fridrich used the sign-symmetry as well as the directional symmetry of images, making those models more compact and improving the performance to dimensionality ratio. Different types of residuals were applied with different symmetrized methods, like 'spam' $\overline{C_d} \leftarrow C_d + C_{-d}$, $\overline{\overline{C_d}} \leftarrow \bar{C}_d + \bar{C}_{\tilde{d}}$, while the 'minmax' residuals will be possessed like $\overline{C_d} \leftarrow C_d^{(min)} + C_{-d}^{(max)}$, $\overline{\overline{C_d}} \leftarrow \bar{C}_d + \bar{C}_{\tilde{d}}$. When all submodels are put together, their combined dimensionality is only 12,753.

The framework is demonstrated on three stego algorithms operating in the spatial domain: ±1 embedding and two content-adaptive methods: HUGO and an edge-adaptive method by Luo et al. They used the best rich model when were assembled at dimensions approximately 3300 and trained by ensemble classifier [32] and Gaussian SVM [44]. The following table shows the result for the 0.4 bpp payload as carrying out these types of experiments.

Table 2.8 Detection Error for Three Algorithms for Payload 0.4 bpp when Ensemble is Used with the Rich 12753-Dimensional Model

Algorithm	Ensemble		G-SVM	
	MED	MAD	MED	MAD
±1 Embedding	0.0785	0.0035	0.00683	0.0042
HUGO	0.1355	0.0035	0.1310	0.0065
EA	0.0695	0.0020	0.0643	0.0030

Table 2.9 The Average Running Time of The Experiments in Table 2.8

Algorithm	Ensemble	G-SVM
±1 Embedding	1 hr 20 min	4 days 22 hr 37 min
HUGO	4 hr 35 min	8 days 15 hr 31 min
EA	3 hr 09 min	3 days 23 hr 50 min

Source: Jessica Fridrich and Jan Kodovský," Rich Models for Steganalysis of Digital Images," IEEE Trans. on Info. Forensics and Security, vol. 7(3), pp. 868-882, 2012

In Table 2.8, writers compared results with the detection error of classifiers implemented as ensembles using the 12,753-dimensional rich model in 0.4 bpp payload dataset. Interestingly, the smaller model with a G-SVM provided better detection results. The improvement is roughly by 0.5–1% over all three steganographic methods. While the running time of a G-SVM classifier was 30–90 times higher than the running time of the ensemble classifier, as reported in Table 2.9.

CHAPTER 3

NEW APPROACHES

3.1 Introduction

With the development of steganalysis, more and more techniques are involved in this area. The diversity of steganalysis can provide theoretical foundation and promote the development of detecting method. In this chapter, game theory and deep learning, which have never been used in steganalysis before, will be presented. Game theory is a theory to analyze the interaction between Alice and Bob in steganography and steganalysis. The value of this work lies primarily in shedding more light on the problem of optimal steganography. And the convolution layers in deep learning may be of great assistance to feature extraction in researches for steganalysis.

3.2 Game Theory Approach

3.2.1 Concept

Game theory is the study of strategic decision making. Specifically, it is "the study of mathematical models of conflict and cooperation between intelligent rational decision-makers."[22]

"*The prisoner's problem"* can illustrate game theory clearly. Alice and Bob were caught transferring state secrets. Now, sadly, they separated into two rooms, homeland security tries to get them to confess. They are each told independently that if they both confess, they will be put in prison for two years. If one confesses and the other does not, the confessor will be let free in exchange for testifying against the other, who will receive

four years in prison. If they both keep quiet they will be let off with a slap on wrist: one year each. The outcomes are represented in the following table:

Table 3.1 All Results of Alice and Bob's Decision

	Confess	Decline
Confess	2 , 2	0 , 4
Decline	4 , 0	1 , 1

When the equilibrium is stable: both players confess. If Alice knows that Bob will confess, then she can do nothing but confess: we can see from the equilibrium it result in a four year for her, if she keeps silent. Of course it is possible that both players will choose not to confess, but this is an unstable equilibrium. If Alice gets wind of the fact that Bob plans to stay quiet, she'll turn him in! And if bob in turn realizes this, he will choose to confess as well. Hence, while this situation can occur, in some instances it is an "unstable" outcome.

This steganalysis is attempting to analyze the interaction between Alice and Bob. Writers focus on the modern steganographic embedding paradigm based on minimizing an additive distortion function.

Content-adaptive steganography constrains its embedding changes to those parts of image where one expects their detection to be harder. In steganography, that minimizes an additive embedding distortion, each pixel is changed with probability

$$\beta_i = \frac{\exp(-\lambda\rho_i)}{1 + \exp(-\lambda\rho_i)} \tag{3.1}$$

where $\rho_i > 0$ is the costs of changing pixel i. The cost ρ_i are usually obtained by using some deterministic rule, which is applied to the cover image.

Since the stego image is a slightly modified version of the cover image, Bob could estimate the set of change rate. Since the introduction of content-adaptive stego schemes, it has been hypothesized that any information about the selection channel given to Bob could be used to improve her detection.

Considering half of a cover image is composed of random noise while the other half is a completely flat content, it is far better for Alice to embed in the random part even though Bob knows it. In fact, the sender could even hide data with perfect security using naive embedding if she knew the cover model. Obviously, the information about the selection channel available to Bob may be a weakness only to a degree depending on how detectable the changes are at each pixel. So Tomáš considered two options for Alice: (1) Assuming an omnipotent Bob, she also knows Alice's actions. (2) Assuming that Bob has no idea about the probabilities with which Alice changes each pixel.

3.2.2 Cover Model and Embedding Method

A. Cover Model

When choosing a cover image, we should ignore the noise components by staying consistently the same, like taking the same picture multiple times with the same camera settings, the remaining noise components are random in nature and well modeled as an independent Gaussian noise. The cover model using is a simplified model for one channel of a raw imaging sensor output:

$$\mathbf{X} = (X_1, \dots, X_n), X_i \sim N(0, \sigma_i^2), \mathrm{i} = 1, \dots, n \tag{3.2}$$

B. Embedding Method

In Tomáš's paper [11], LSB matching is chosen for simplicity. Denoting the stego image $\mathbf{Y} = (Y_1, \dots, Y_n)$, Alice changes pixel x_i by ±1 with probability $\rho_i^{(A)}$ and leaves it unchanged with probability $1 - \rho_i^{(A)}$.

$$\Pr(Y_1 = x_i + s_i) = \begin{cases} \rho_i^{(A)}/2 & \text{for } s_i = -1 \\ 1 - \rho_i^{(A)} & \text{for } s_i = 0 \\ \rho_i^{(A)}/2 & \text{for } s_i = 1 \end{cases} \tag{3.3}$$

Therefore, each stego pixel follows a Gaussian mixture:

$$Y_i \sim f_{\rho_i^{(A)}}(x, \sigma_i^2) \tag{3.4}$$

When Alice embeds α bpp, the embedding probabilities must satisfy constraint

$$\sum_{i=1}^{n} h\left(\rho_i^{(A)}\right) = \alpha n \tag{3.5}$$

Thus, Alice's action is captured with $n - 1$ parameters: $\rho_i^{(A)}, i = 1, \dots, n - 1$ as $\rho_n^{(A)}$ is determined from the payload constraint.

3.2.3 Detector

A. Detector Concept

Bob will run a simple binary hypothesis test. The null hypothesis corresponds to observing a cover image, while the alternative hypothesis corresponds to a stego object.

Given an image $x = (x_1, \ldots, x_n)$, Bob uses the Likelihood Ratio Test (LRT) as her detector:

$$\mathrm{T}\left(x; \rho_i^{(W)}, \sigma^2\right) = \prod_{i=1}^{n} \frac{f_{\rho_i^{(w)}}(x_i, \sigma_i^2)}{f(x_i, 0, \sigma_i^2)} \tag{3.6}$$

B. Payoff Function

Some scalar characteristic is needed in the detector as a payoff function. While they adopt the customary method, which is minimal total error probability for Equation 3.6, as the payoff function.

$$P_E = \min \frac{1}{2}(P_{FA} + P_{MD}(P_{FA})) \tag{3.7}$$

P_{FA} and P_{MD} are the probabilities of false detection. To evaluate the payoff function, Writers compute the distribution of Bob's statistic under both hypotheses. C.d.f. obtained after a series of transform,

$$F(y) = \int_{x-}^{x+} f_{\beta(A)}(x, \sigma_i^2) dx \tag{3.8}$$

The p.d.f. is obtained by differentiating w.r.t.y:

$$h(y) = f_{\beta(A)}(x, \sigma_i^2) x'_+(y) - f_{\beta(A)}(x, \sigma_i^2) x'_-(y) \tag{3.9}$$

C. Strategies

Alice's and Bob's strategies are the following sets of $n-1$ real values,

$$S_A = \left\{\beta_1^{(A)}, \dots, \beta_{n-1}^{(A)}\right\} \tag{3.10}$$

$$S_W = \left\{\beta_1^{(W)}, \dots, \beta_{n-1}^{(W)}\right\} \tag{3.11}$$

3.2.4 Solution of Game Theory

A game with continuous strategies and a smooth payoff function admits solution in pure strategies, which coincides with the saddle point, the Nash equilibrium. The solution can be determined numerically using a gradient search in which the payoff function is minimized over Bob's strategies and maximized over Alice's strategies [11].

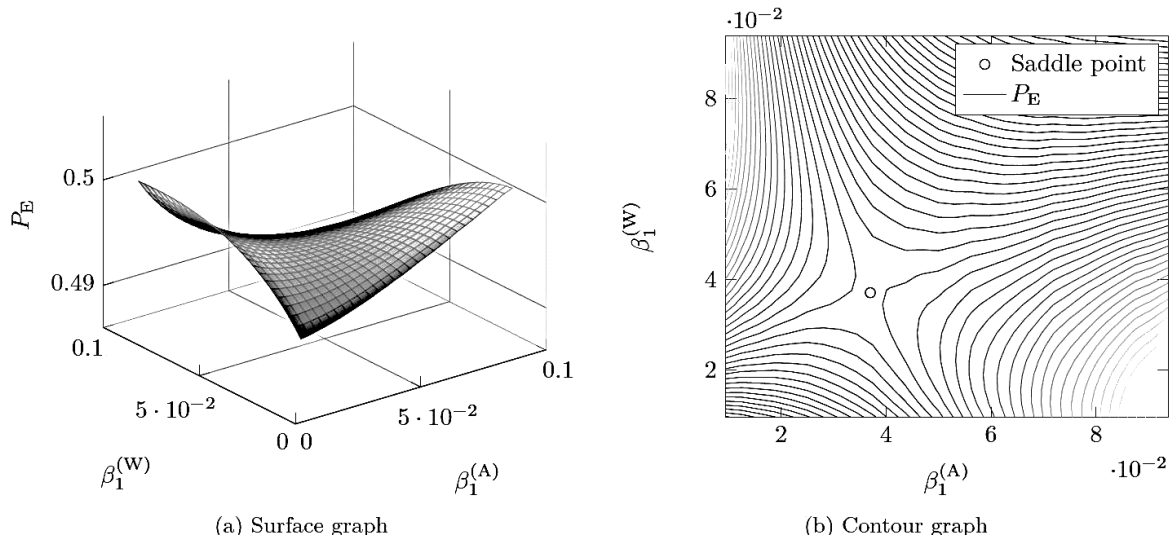


Figure 3.1 Payoff function $P_E(\rho_1^{(A)}, \rho_1^{(W)})$ for $\alpha = 0.2, \sigma_1{}^2 = 1, \sigma_2{}^2 = 1.2$.

Source: Tomáš Denemark and Jessica Fridrich," Detection of Content Adaptive LSB Matching (a Game Theory Approach)," Media Watermarking, Security, and Forensics 2014, vol. 9028, San Francisco, CA, February 26, 2014.

For a two-pixel cover, the one-dimensional strategies must lie in a range determined by the payload， $\rho_1^{(A)}, \rho_1^{(W)} \in [\beta_{min}, \beta_{max}]$, where

$$\beta_{max} = \min(0.5, h^{-1}(2\alpha)) \tag{3.12}$$

$$\beta_{min} = h^{-1}(2\alpha - h(\beta_{max})) \tag{3.13}$$

where $h^{-1}(x)$ is the inverse binary entropy on [0,1/2].

The value of this work lies primarily in shedding more light on the problem of optimal steganography under an ignorant Bob.

3.3 Deep Learning

Qian et al. [17] proposed a customized Convolutional Neural Network (CNN) called Gaussian-Neuron CNN (GNCNN) for steganalysis. This model can automatically learn to extract feature by several convolutional layers.

3.3.1 CNN

In deep learning, a Convolutional Neural Network [23] is comprised of one or more convolutional layers and then followed by one or more fully connected layers. The architecture of a CNN is designed to take advantage of the 2D structure of an input image. Additionally, CNNs is easy to train with fewer parameters compared with other fully connected networks with the same number of hidden units.

Either before or after the subsampling layer an additive bias and sigmoidal nonlinearity is applied to each feature map. The figure below illustrates a full layer in a CNN consisting of convolutional and subsampling sublayers.

Here are the reasons of choosing CNN as the basic framework:

1. *CNNs can take raw data as inputs* without the need for a feature extraction step, which means that only learning feature representations from images instead of treating a CNN.
2. *The data processed by CNN is easy to use* in steganalysis, regarding covers and stegos as positive and negative samples.

3. *CNN can be trained the models on large scale*.

Qian customized the CNN as a GNCNN shows in Figure 3.2.

3.3.2 Image Processing

Generally, the high frequency stego noise added to the cover is a kind of very weak signal, which is greatly impacted by image content. Hence, reduce the impact of image content and strengthen the weak stego signal, Qian apply a high pass filter.

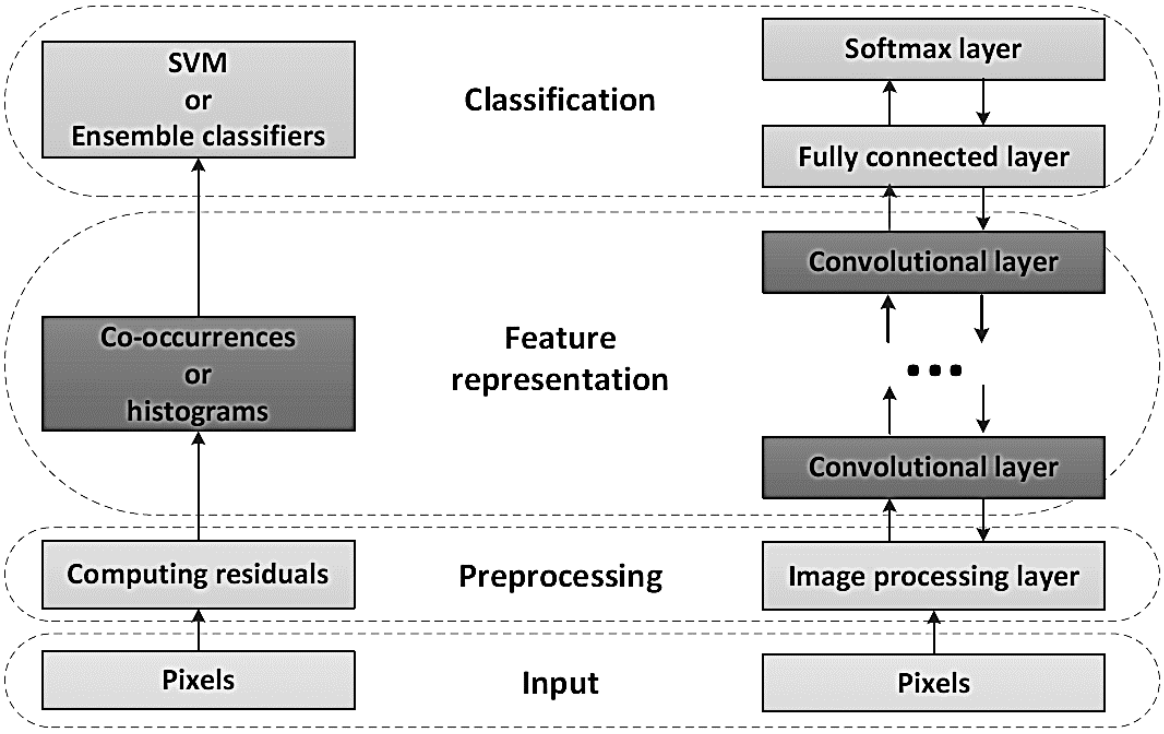


Figure 3.2 The GNCNN model (right) and traditional steganalysis architecture based on hand-crafted features (left). The up and down arrows in the right flowchart show forward and back propagation directions.

Source: Yinlong Qian, Jing Dong, Wei Wang, Tieniu Tan, "Deep learning for steganalysis via convolutional neural networks," SPIE 9409, Media Watermarking, Security, and Forensics 2015, 94090J (4 March 2015); DOI: 10.1117/12.2083479

R is the image after high-pass filtering, I denotes an image and K is a shift-invariant finite-impulse response linear filter to compute the residual.

$$R = K * I \tag{3.14}$$

where $*$ denots convolution. Equation 2.30 shows the k filter (kernel).

$$K_{kv} = \frac{1}{12}\begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix} \tag{3.15}$$

3.3.3 Convolutional Layer

The input and output of each convolutional layer are sets of arrays called feature maps. At the output, each feature map is a particular feature representation extracted at all locations on the input. At a convolutional layer, three kinds of operations, which are convolution, non-linearity, and pooling, are usually applied sequentially as expressed below.

$$X_j^l = \text{pool}\left(f\left(\sum_i X_i^{l-1} * K_{ij}^l + b_j^l\right)\right) \tag{3.16}$$

where $f(\cdot)$ as the non-linearity operation, $\text{pool}(\cdot)$ denotes pooling, X_i^{l-1} is the j-th feature map in layer l, K_{ij} is the trainable convolution kernel connecting the j-th output

map and the i-th input map, b_j^l is an trainable bias parameter for the j-th output map.

For natural images, it is common that, subsequent in-camera processing during image acquisition, such as color interpolation, color correction and filtering, introduces complex dependencies into noise component of neighboring pixels. Most steganalysis methods try to utilize these dependencies to detect stego noise. Because of the complex dependencies, a good estimate of the central pixel can be obtained from the neighboring pixels, excluding the pixel being estimated. Then by subtracting the true value of the central pixel from the estimated one, the prediction error value can be obtained, which directly reflects whether the pixel is changed or not.

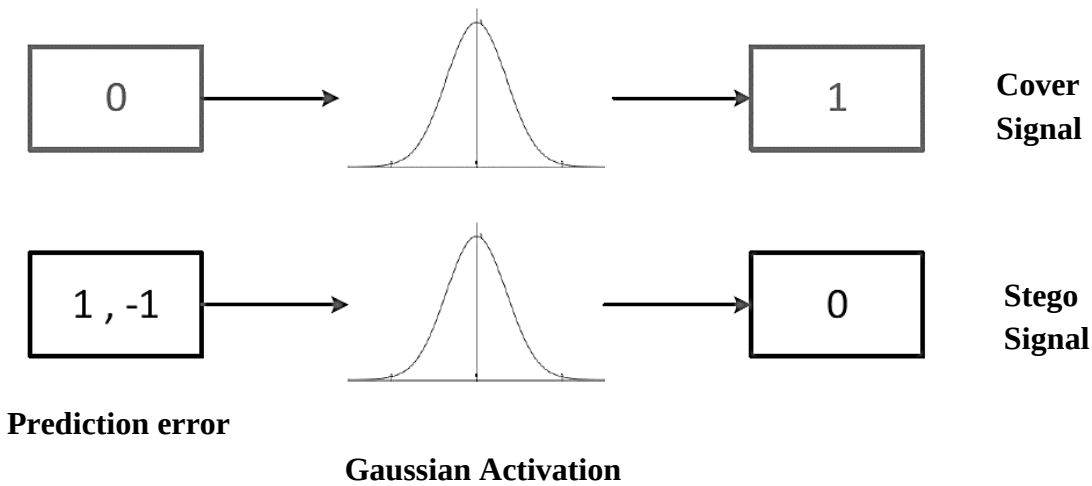


Figure 3.3 A Simple example illustrating why Gaussian activation works for steganalysis in the proposed model. Gaussian activation can distinguish between stego signals and cover signal from the prediction error values.

Source: Yinlong Qian, Jing Dong, Wei Wang, Tieniu Tan, "Deep learning for steganalysis via convolutional neural networks," SPIE 9409, Media Watermarking, Security, and Forensics 2015, 94090J (4 March 2015); doi: 10.1117/12.2083479

As mentioned, the aim of convolution operation is to compute prediction error values by exploiting the dependencies among neighboring elements. The motivation behind Gaussian non-linearity is to transform prediction error values to distinguish between stego signals and cover signals.

Figure 3.3 gives a simple example of why Gaussian activation works. Ideally, it is supposed that there should be three kinds of prediction error values: 1, -1 and 0. The values 1 and -1 means that the pixel is modified by embedding operation, a stego signal. The value 0 means that the pixel is unchanged, which means this is a cover signal.

The resulting activations are then passed to the pooling part of the layer. Pooling operation aims to transform the low level feature representation into a more usable one which preserves important information and discards irrelevant details. Pooling has the effect of merging the information within a set of small local regions while reducing computation time. In a pooling operation, the outputs of neighboring groups of neurons in the same feature map are summarized.

Generally, there are two conventional choices for pooling: average pooling and max pooling. The former takes the average value within the pooling region:

$$\mathrm{pool}(R_j) = \frac{1}{|R_j|} \sum_{i \in R_j} a_i \tag{3.17}$$

while the max pooling operation selects the maximum value:

$$\text{pool}(R_j) = \max_{i \in R_j} a_i \tag{3.18}$$

where R_j is pooling region j in a feature map, a_i is the i-th element within it.

In their proposed model, rather than max pooling, they use average pooling. Because in average pooling, all the activations in the pooling region are taken into account, which is supposed to discard the disturbances caused by individual elements. By merging all the signal within the pooling region, the stego signal on the whole region is strengthen.

3.3.4 Classification Layer

The classification module consists of several fully connected layers. The learned features are passed to these layers. On the top layer, an activation function is used to produce a distribution over all the class labels.

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{2} e^{x_j}} \tag{3.19}$$

for $i = 1, 2$, where x_i is the total input to the neuron i in the top layer, and y_i is the output.

3.3.5 Accuracy

Qian et al. carried out on the standardized database called BOSSbase 1.01.22. This database contains 10,000 images acquired by seven digital cameras.

Table 3.2 Detection Error of GNCNN Model VS. The SRM Set Implemented with Ensemble Classifiers and the SPAM Set Implemented with a Gaussian SVM for Three State-of-the-Art Spatial Domain Steganographic Algorithms

bpp	HUGO			WOW			S-UNIWARD		
	GNCNN (256D)	SRM (34,671D)	SPAM (686D)	GNCNN (256D)	SRM (34,671D)	SPAM (686D)	GNCNN (256D)	SRM (34,671D)	SPAM (686D)
0.3	33.8%	29.6%	42.9%	34.3%	31.2%	42.2%	35.9%	31.5%	40.0%
0.4	28.9%	25.2%	39.1%	29.3%	25.7%	38.2%	30.9%	26.3%	35.1%
0.5	25.7%	21.4%	35.7%	24.8%	22.1%	34.9%	26.3%	21.4%	30.6%

Source: Yinlong Qian, Jing Dong, Wei Wang, Tieniu Tan, "Deep learning for steganalysis via convolutional neural networks," SPIE 9409, Media Watermarking, Security, and Forensics 2015, 94090J (4 March 2015); DOI: 10.1117/12.2083479.

For BOSSbase, across all three embedding algorithms and payloads, the method with deep learning achieves a much lower detection error than the SPAM set implemented with a Gaussian SVM. When compared to the SRM set implemented with ensemble classifiers, the detection error is just about $2\% - 5\%$ higher depending on the payload. Experiments on ImageNet show that this method achieves a close detection error to the SRM set.

CHAPTER 4

TEXTUREAL FEATURE

4.1 Textural Features

4.1.1 Introduction

Shi et al. [3] learned and utilized the textural features from the rich literature in the field of texture classification for further development of the modern steganalysis. They have applied textural features to steganalyzing the HUGO stego dataset designed for the BOSS contest. In this scheme, they construct a steganalyzer with 22,153 features derived from the textural features.

They applied LBP in 59 dimensional features and used these for some filtered 2-D array. Meanwhile, 256 dimension and variance features derived from the multi-resolution way were used for others.

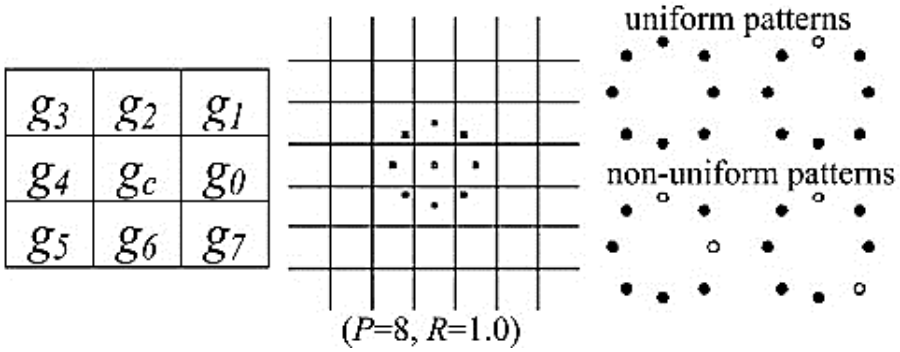
In addition, they used Laws mask and the mask and cliques associated with Markov Random fields. And the classifier utilized is the FLD-based ensemble classifier.

4.1.2 Framework

HUGO tended to embed data into cover image locally into some regions so as to make the image statistical modeling difficult, especially into highly texture regions. While the LBP operators have been popularly used in texture classification arena.

A. Image Statistical Measures

Ojala et al. [34] proposed LBP to model the statistics of a texture unit defined within a neighborhood of 3×3 pixels. Each of eight neighboring pixels of a 3×3 neighborhood is thresholded by the gray value of its central pixel to form an 8-bit binary pattern shown in Figure 4.1(a). We can see that the Figure 4.1(b) is a version of LBPs. The LBP is circular with different radius, the pixel values of the neighbors falling outside the center of the pixel grids are estimated by interpolation. While Figure 4.1(c) is a derivation of Figure 4.1(b). There are two kinds of LBP: *uniform* and *non-uniform* patterns. The uniform patterns have the number of binary transitions over the whole neighborhood circle less than two, while the number of transitions that are greater than two are considered as non-uniform. Uniform patterns often occupy the majority of the histogram which makes merging non-uniform patterns into the same bin. In this situation, we can reduce the number of bins in a histogram from 256 to 59.



(a) (b) (c)

Figure 4.1 (a) 3×3 neighborhood. (b) Example of circular symmetric neighborhood. (c) Examples of "uniform" and "non-uniform" local binary patterns.

Generalized to different P values and correspondingly defined neighborhoods, Equation 4.1 expresses the formulation of LBP shown in Figure 4.1(a) mathematically.

$$LBP = \sum_{P=0}^{P-1} s(g_p - g_c)2^p \tag{4.1}$$

where *s(x)* equals one if the *x* is less than or equal to zero, or zero otherwise.

From experiments, they found that features generated from LBP8 are much more powerful than those from co-occurrence matrix but with a higher dimensionality. Features generated from LBP perform slightly better than those from co-occurrence matrix although they are of lower dimensionality.

4.1.3 Content-Adaptive Prediction Error Image

Small perturbation to cover image caused by steganographic schemes may be considered as a high frequency additive noise; as a result, eliminating low-frequency representation of images before feature extraction process would make the resulting image features better represent the underlying statistical artifacts associated with steganography.

With the modern steganographic schemes such as HUGO, it is intuitive that the prediction error images, like residual images, generated in a content adaptive manner would effectively reveal such artifacts caused by data embedding. Denoting I as image, R as residual image, and $Pred(I)$ as corresponding predicted image:

$$R = I - Pred(I) \tag{4.2}$$

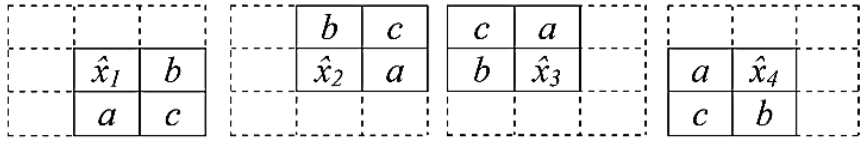
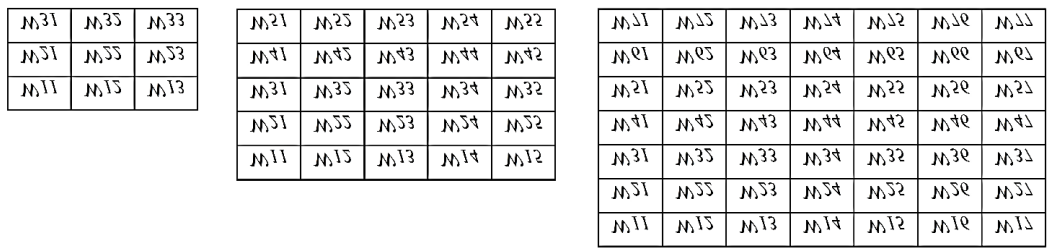


Figure 4.2 2×2 Neighboorhood used to predict the center pixel of a 3×3 neighborhood.



(a) (b) (c)

Figure 4.3 Symbolic representations of pixel locations used in the creation of median-filter-based prediction error images. (a) 3×3, (b) 5×5, and (c) 7×7 neighborhood.

Table 4.1 Configuration of Median Filters Employed in Generating Median-Filter-Based Prediction Error Images

Mask size	Filter number	Pixel locations used in computing median image
3×3	1	w_{11}, w_{13}, w_{22}, w_{31}, w_{33}
	2	w_{12}, w_{21}, w_{22}, w_{23}, w_{32}
5×5	1	w_{12}, w_{14}, w_{21}, w_{22}, w_{24}, w_{25}, w_{33}, w_{41}, w_{42}, w_{44}, w_{45}, w_{52}, w_{54}
	2	w_{11}, w_{13}, w_{15}, w_{31}, w_{33}, w_{35}, w_{51}, w_{53}, w_{55}
	3	w_{13}, w_{22}, w_{23}, w_{24}, w_{31}, w_{32}, w_{33}, w_{34}, w_{35}, w_{42}, w_{43}, w_{44}, w_{53}
7×7	1	w_{12}, w_{13}, w_{15}, w_{16}, w_{21}, w_{22}, w_{23}, w_{25}, w_{26}, w_{27}, w_{31}, w_{32}, w_{33}, w_{35} w_{36}, w_{37}, w_{44}, w_{51}, w_{52}, w_{53}, w_{55}, w_{56}, w_{57}, w_{61}, w_{62}, w_{63}, w_{65}, w_{66} w_{67}, w_{72}, w_{73}, w_{75}, w_{76}
	2	w_{14}, w_{22}, w_{24}, w_{26}, w_{34}, w_{41}, w_{42}, w_{43}, w_{44}, w_{45}, w_{46}, w_{47}, w_{54}, w_{62} w_{64}, w_{66}, w_{74}
	3	w_{11}, w_{13}, w_{15}, w_{16}, w_{31}, w_{33}, w_{35}, w_{37}, w_{44}, w_{51}, w_{53}, w_{55}, w_{57}, w_{71} w_{73}, w_{75}, w_{77}
	4	w_{14}, w_{23}, w_{24}, w_{25}, w_{32}, w_{33}, w_{34}, w_{35}, w_{36}, w_{41}, w_{42}, w_{43}, w_{44}, w_{45} w_{46}, w_{47}, w_{52}, w_{53}, w_{54}, w_{55}, w_{56}, w_{63}, w_{64}, w_{65}, w_{74}

Source: Yun Q. Shi, Patchara Sutthiwan, Licong Chen,Shi, "Textural Features for Steganalysis," IH'12 Proceedings of the 14th international conference on Information Hiding, 2012, pp 63-77

Table 4.2 High-Pass Filters Employed in the Creation of Residual Images

Category	Number of Taps	Name	Filter
Laws' Mask	3	Edge 3 (E3)	[-1 0 1]
		Spot 3 (S3)	[-1 2 -1]
	5	Edge 5 (E5)	[-1 -2 0 2 1]
		Spot 5 (S5)	[-1 0 2 0 -1]
		Wave 5 (W5)	[-1 2 0 -2 1]
		Ripple 5 (R5)	[1 -4 6 -4 -1]
	7	Edge 7 (E7)	[-1 -4 -5 0 5 4 1]
		Spot 7 (S7)	[-1 -2 1 4 1 -2 -1]
		Wave 7 (W7)	[-1 0 3 0 -3 0 1]
		Ripple 7 (R7)	[1 -2 -1 4 -1 -2 1]
		Oscillation 7 (O7)	[-1 6 -15 20 -15 6 -1]
Even Taps	2	Filter 2 (F2)	[-1 1]
	4	Filter 4 (F4)	[1 -3 3 -1]
	6	Filter 6 (F6)	[1 -3 2 2 -3 1]

Source: Yun Q. Shi, Patchara Sutthiwan, Licong Chen,Shi, "Textural Features for Steganalysis," IH'12 Proceedings of the 14th international conference on Information Hiding, 2012, pp 63-77

Table 4.1 shows some spatial filters which have been widely used as low-pass filters. It can generate residual images by using median filters to compute predicted images.

Writers also generate some residual images in this part in a content adaptive manner by incorporating two non-linear operators, minimum and maximum in order to catch the desired artifacts.

Image statistical features is formulated by two major set of 1-D spatial high-pass filters. The first set of high-pass filters is Laws' mask which are odd sizes, while the other set which contains even-tap high-pass filters. Those filters are shown in Figure4.4 and 4.5.

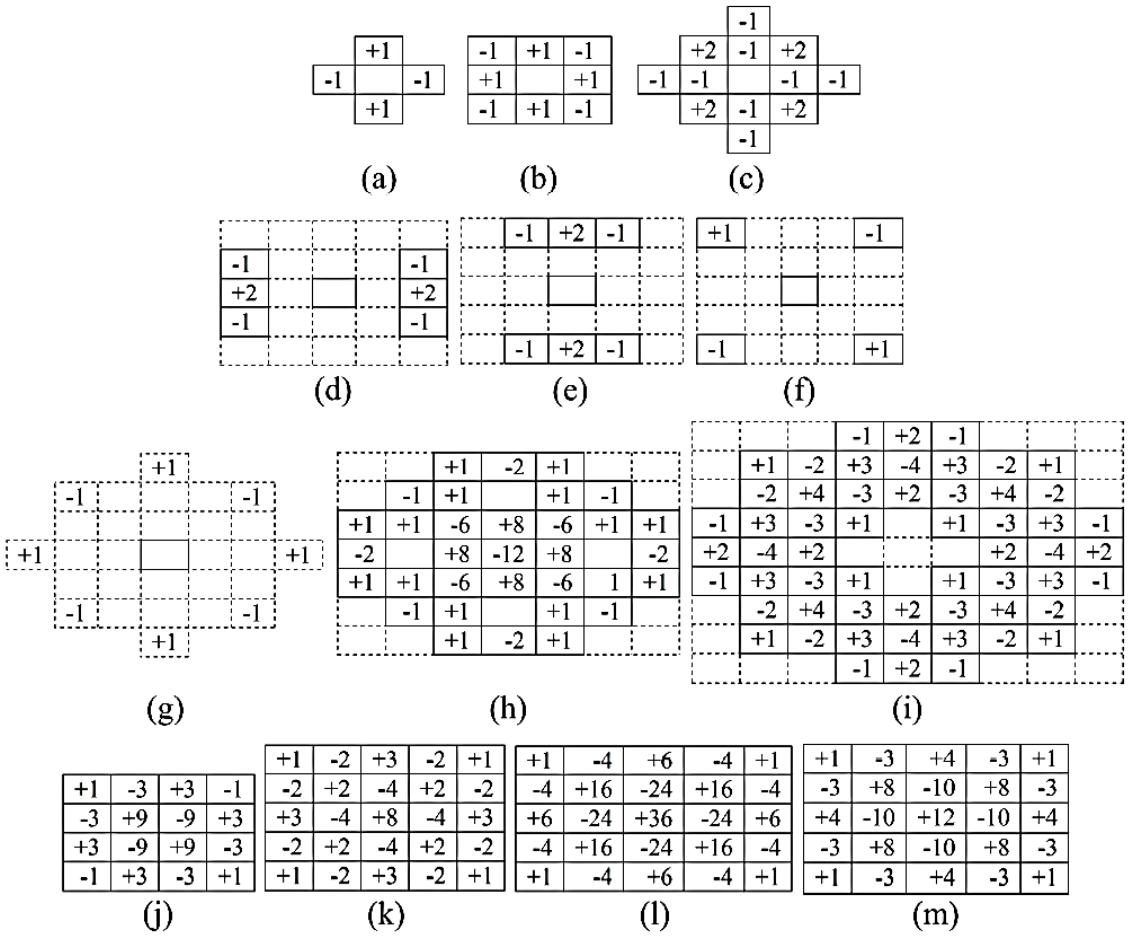


Figure 4.4 High-pass filters based on Markov neighborhoods.

Source: Yun Q. Shi, Patchara Sutthiwan, Licong Chen, "Textural Features for Steganalysis," IH'12 Proceedings of the 14th international conference on Information Hiding, 2012, pp. 63-77

The Figure 4.4 and Figure 4.5 show the mask based on Markov. Markov Random Field (MRF) has been widely used in texture classification, segmentation and texture defect detection.

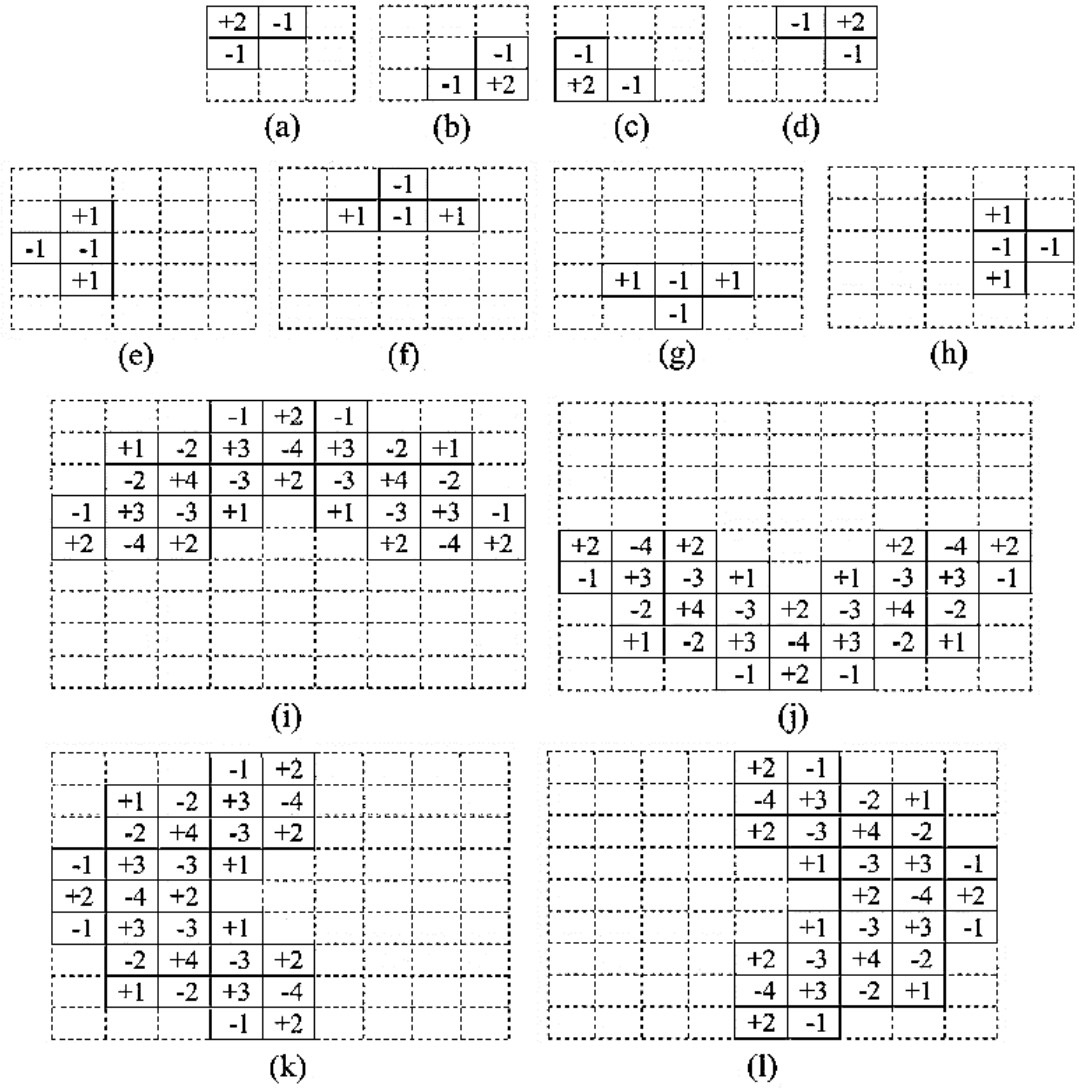


Figure 4.5 High-pass filters based on Cliques.

Source: Yun Q. Shi, Patchara Sutthiwan, Licong Chen, "Textural Features for Steganalysis," IH'12 Proceedings of the 14th international conference on Information Hiding, 2012, pp. 63-77

In MRF, a neighborhood can be constructed, which the Markov parameters can be assigned as weights. These neighborhoods are characterized by a group of pixels with a variety of orientations often symmetrically inscribed within a square window of odd size. For steganalysis, Markov neighborhood should be for high-pass filtering instead of texture classification. Figure 4.5 represents the masks they generated.

4.1.4 Accuracy

With a variety of features made in the above section, there are multiple ways to construct a feature set for steganalysis. An effective combination of features with a dimensionality of 22,153 is constructed based on the HUGO at 0.4 bpp on BOSSbase 0.92 which contains of 10,000 images. All the LBP operators used to construct features are based on uniformity mapping with $P = 8$ and different combination of R's. Whole feature sets with each individual type of features dropped out are evaluated and shown in Table 4.3

Table 4.3 Ensemble Performance on Feature Elimination at d = 2,600

Feature Set	D	AC	L
Whole	22,593	83.92%	50
Whole-Pes	21,268	83.57%	46
Whole-VARpe	21,268	83.57%	57
Whole-MEDpe	20,560	83.67%	63
Whole-LMased	9,763	82.72%	65
Whole-MN13	18,825	83.52%	45
Whole-CL12	19,081	83.67%	52

Source: Yun Q. Shi, Patchara Sutthiwan, Licong Chen, "Textural Features for Steganalysis," IH'12 Proceedings of the 14th international conference on Information Hiding, 2012, pp. 63-77

The statistics in Table 4.3 reveals that each type of the proposed features is essential to the final accuracy, that is, the final accuracy decreases upon the absence of each type of features.

4.2 Ensemble Classifier

4.3.1 Introduction

To reconstruct the method, we will use the ensemble classifier as described in Shi's paper. The ensemble classifier [18] is essentially a random forest consisting of L binary classifiers called base learners, B_l, $l = 1, \ldots, L$, each trained on a different dsub-dimensional subspace of the feature space selected uniformly at random. Each random subspace will be described using an index set $\{1, \ldots, d\}$, $|D_l| = d_{sub}$. The ensemble reaches its decision by fusing all L decisions of individual base learners using majority voting.

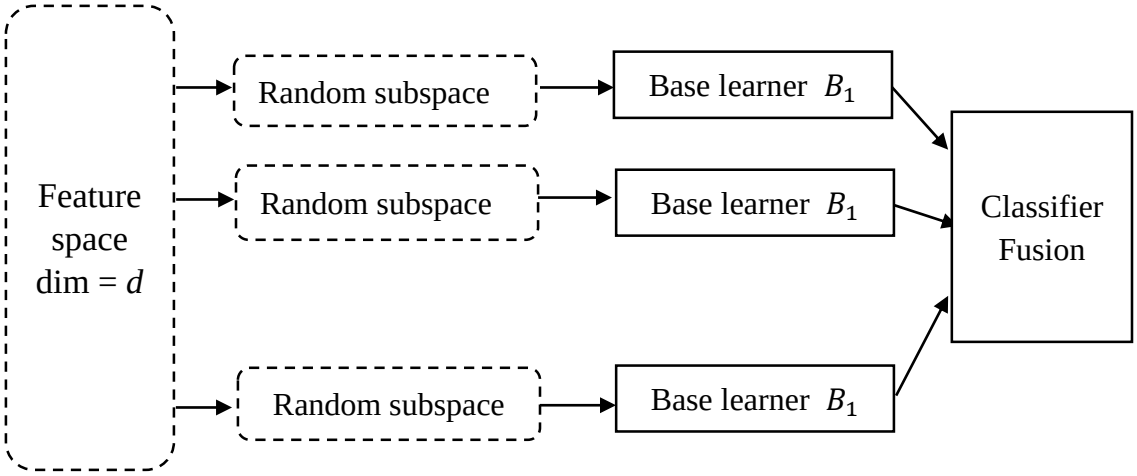


Figure 4.6 Diagram illustrating the ensemble classifier.

4.2.2 Algorithm

To formally describe the ensemble classifier, we introduce the following notation. The symbol d stands for the feature space dimensionality, d_{sub} for the dimensionality of the

feature subspace on which each base learner operates, N^{trn} and N^{tst} are the number of training and testing examples from each class, and L is the number of base learners.

Steps of algorithm are as follows.

1. For $l = 1$ to L form a random subspace

$$\mathcal{D}_l \subset \{1, \dots, d\}, |\mathcal{D}_l| = d_{sub} < d \tag{4.3}$$

2. Forming a bootstrap sample $\mathcal{N}_l^b, |\mathcal{N}_l^b| = N^{trn}$ by uniform sampling with replacement form the set $\{1, \dots, N^{trn}\}$.

3. Training a base learner B_l on features

$$\mathcal{X}_l = \left\{x_m^{(\mathcal{D}_l)}, \bar{x}_m^{(\mathcal{D}_l)}\right\}_{m \in \mathcal{N}_l^b} \tag{4.4}$$

to obtain eigenvector v_l and threshold T_l.

4. For all test examples $y \in \mathcal{Y}^{tst}$ make lth decisions:

$$B_l\left(u^{(\mathcal{D}_l)}\right) \triangleq \begin{cases} 1 & when\ \mathbf{v}_l^T \boldsymbol{y}^{(\mathcal{D}_l)} > T_l \\ 0 & otherwise \end{cases} \tag{4.5}$$

5. If $l = L$, end the loop, otherwise, return to step 1.

6. Forming the final decision $B(\boldsymbol{y})$ by majority voting

$$B(\boldsymbol{y}) = \begin{cases} 1 & when \sum_{l=1}^{L} B_l\left(y^{(\mathcal{D}_l)}\right) > L/2 \\ 0 & when \sum_{l=1}^{L} B_l\left(y^{(\mathcal{D}_l)}\right) < L/2 \\ random & otherwise \end{cases} \tag{4.6}$$

The individual base learners $B_l, l = 1, \ldots, L$, are mappings $\mathbb{R}^d \rightarrow \{0, 1\}$, where 0 stands for cover and 1 for stego. Even though the performance of individual base learners can be weak, the accuracy quickly improves after fusion and eventually levels out for a sufficiently large L.

From the experiments in Kodovsky's paper, they show the ensemble is especially useful for fast feature development when attacking a new scheme. Ensemble classifiers offer accuracy comparable and often even better than the much more complex SVMs at a fraction of the computational cost.

4.3 High-order Local Pattern

A good object representation or object descriptor is one of the key issues for a well. The writers propose a novel object descriptor, the high order Local Derivative Pattern (LDP), for robust face recognition. The **n**th LDP is proposed to encode the (n-1)th-order local derivative direction variations.

4.3.1 Local Binary Pattern

Derived from a general definition of texture in a local neighborhood, LBP is defined as a grayscale invariant texture measure and is a useful tool to model texture images.

The thresholding function for the basic LBP can be formally represented as

$$f\left(I(Z_0), I(Z_i)\right) = \begin{cases} 0, & if\ I(Z_i) - I(Z_0) \leq \text{threshold} \\ 1, & if\ I(Z_i) - I(Z_0) > \text{threshold} \end{cases},\ i = 1, 2, \cdots, 8 \tag{4.7}$$

Figure. 4.7 shows an example of obtaining an LBP micro-pattern when the threshold is set to zero.

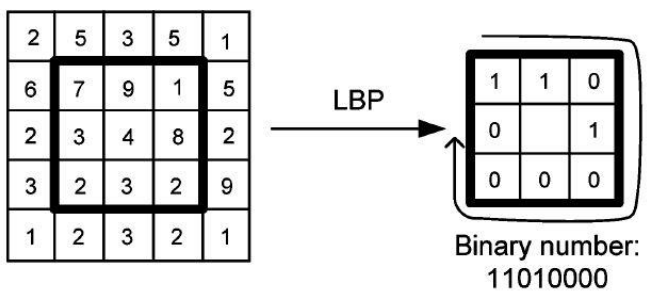


Figure 4.7 Example of obtaining the LBP micro-pattern for the region.

4.3.2 Local Ternary Patterns

Local binary pattern is a 2-valued (binary) code that is successfully used in many applications .The LBP operator idea is based on just two bit values either 1 or 0. This basis does not allow the LBP operator to discriminate between multiple patterns. The LBP operator has two main points of weakness [13]:

1) The LBP operator cannot distinguish between two pixel values if the first one is near the central pixel but a little bit below that pixel and the second undistinguishable one is far below the center pixel value.

2) In flat image areas, such as in face images, where all pixels nearly have the same gray value, if a slight amount of noise were added to these areas the LBP operator will give some bits the value 0 and others the value 1. So the LBP feature will be unstable and thus the LBP operator will not be suitable for analyzing these areas.

To solve these problems a new 3-valued texture operator, Local Ternary Patterns (LTP), that can be considered as an extension to LBP, was introduced recently.

Instead of a thresholding that is based only on the central pixel value of the neighborhood, the user will define a threshold say t and any pixel value within the interval of $-t$ and $+t$, thus assigns the value 0 to that pixel, while the user assigns the value 1 to that pixel if it is above this threshold and a value -1 if it is below it when compared to the central pixel value.

$$LTP(i) = \begin{cases} 1 & if\ p_i - p_c \geq t \\ 0 & if\ |p_i - p_c| < t \\ -1 & if\ \ p_i - p_c \leq t \end{cases} \tag{4.8}$$

where t is a user specified threshold, p_i is a pixel value in the neighborhood and p_c is the central pixel value.

Figure 4.8 shows an example of how the LTP operator works by using a threshold value $t = 5$:

38	**44**	**60**
42	**46**	**63**
32	**56**	**40**

→

-1	**0**	**1**
0		**1**
-1	**1**	**-1**

Figure 4.8 LTP Computation.

To get rid of the negative values in Figure 4.8, the LTP values are divided into two LBP channels, the upper LTP (LTPU) and the lower LTP (LTPL) as in Figure 4.9. The LTPU is obtained by replacing the negative values in the original LTP by zeros. The LTPL is obtained in two steps: first, we replaced all the value of 1's in the original LTP to be zeros then we changed the negative values to be 1's.

-1	0	1
0		1
-1	1	-1

LTPU = $(10000101)_2 = (133)_{10}$

0	0	1
0		1
0	1	0

LTPL = $(00101010)_2 = (42)_{10}$

1	0	0
0		0
1	0	1

Figure 4.9 Splitting LTP into two LBP channels.

4.3.3 Local Derivative Pattern

In LDP the $(\mathrm{n}-1)^{th}$-order derivative direction variations based on a binary coding function.

$$LDP_{\alpha}^{2}(Z_0) = \left\{f\left(I'_{\alpha}(Z_0), I'_{\alpha}(Z_1)\right), f\left(I'_{\alpha}(Z_0), I'_{\alpha}(Z_2)\right), \dots, f\left(I'_{\alpha}(Z_0), I'_{\alpha}(Z_8)\right)\right\} \quad (4.9)$$

It encodes the co-occurrence of two derivative directions at different neighboring pixels as

$$f\left(I'_{\alpha}(Z_0), I'_{\alpha}(Z_i)\right) = \begin{cases} 0 & if\ I'_{\alpha}(Z_i) \cdot I'_{\alpha}(Z_0) > 0 \\ 1 & if\ I'_{\alpha}(Z_i) \cdot I'_{\alpha}(Z_0) \leq 0 \end{cases} \quad (4.10)$$

$i = 1,2, \dots, 8$. The second-order Local Derivative Pattern is defined as the concatenation of the four 8-bit directional LDPs.

$$LDP^{2}(Z) = \{LDP_{\alpha}^{2}(Z) | \alpha = 0°, 45°, 90°, 135°\} \quad (4.11)$$

The derivative direction comparisons defined in Figure 4.10 are performed on 16 templates reflecting various distinctive spatial relationships in a local region. An example of the second-order LDP computation is illustrated in Figure 4.11.

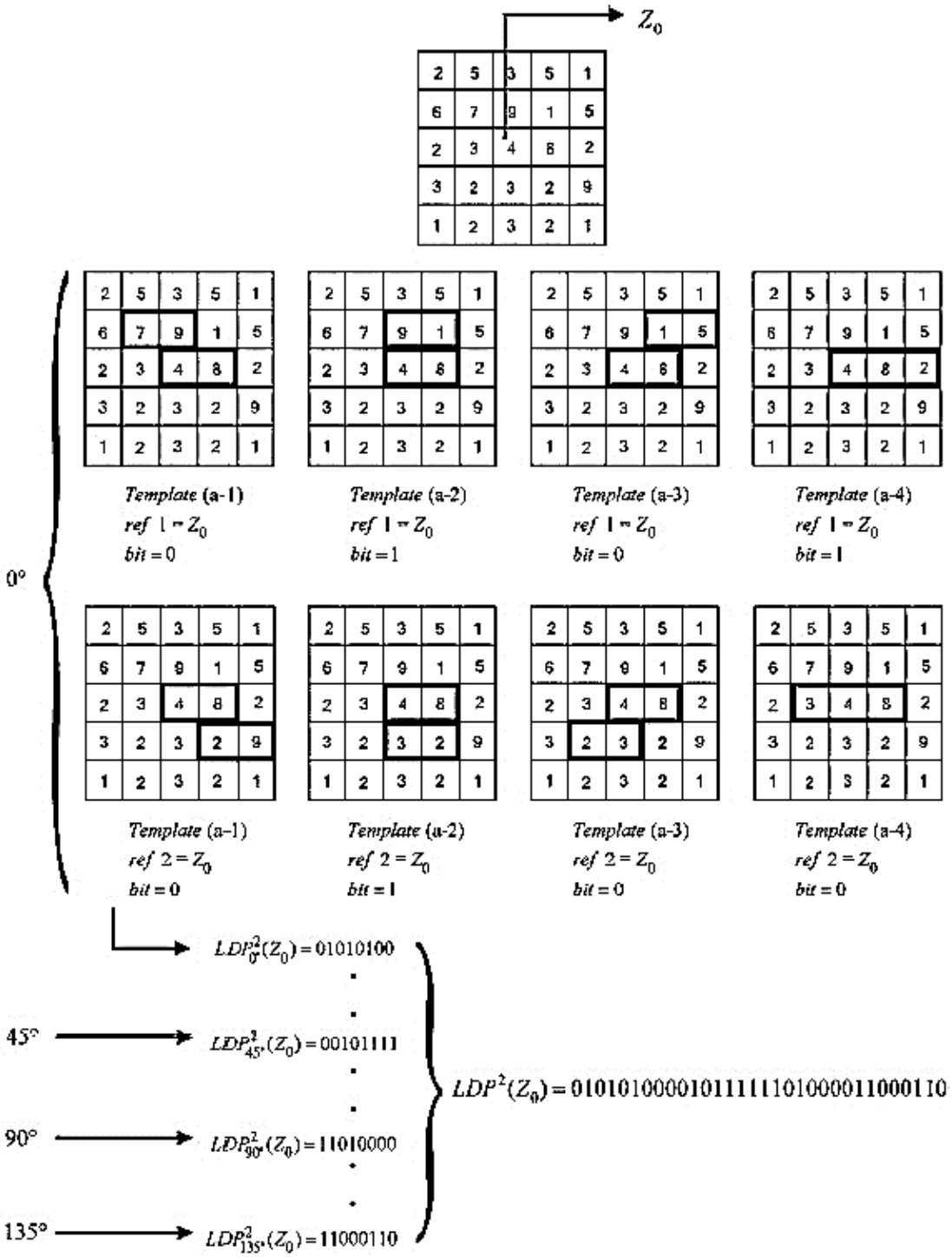


Figure 4.10 Example to obtain the second-order LDP micro-patterns.

Source: Baochang Zhang, Yongsheng Gao, Sanqiang Zhao, Jianzhuang Liu, "Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order," IEEE Transactions on Image Processing, VOL. 19, NO. 2, Feburary 2010.

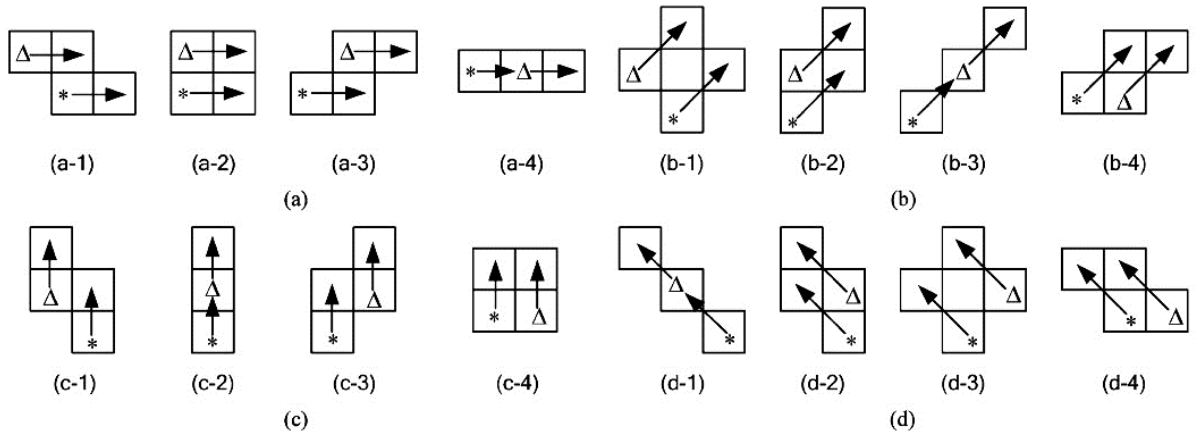


Figure 4.11 Illustration of LDP templates.

Source: Baochang Zhang, Yongsheng Gao, Sanqiang Zhao, Jianzhuang Liu, "Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order," IEEE Transactions on Image Processing, VOL. 19, NO. 2, February 2010.

4.3.4 Experiments

A thorough system performance investigation, which covers various conditions of face recognition including lighting, accessory, pose, expression and aging variations, has been conducted. The comparative experiments between LDP and LBP were first conducted on the FERET face database, which is widely used to evaluate face recognition algorithms.

Experimental results in Figure 4.12 demonstrate that the recognition accuracy in average is significantly improved when the order of local pattern is increased from the first-order LBP to the second-order and the third-order LDPs. While illustrating this the third-order LDP performs much better than LBP.

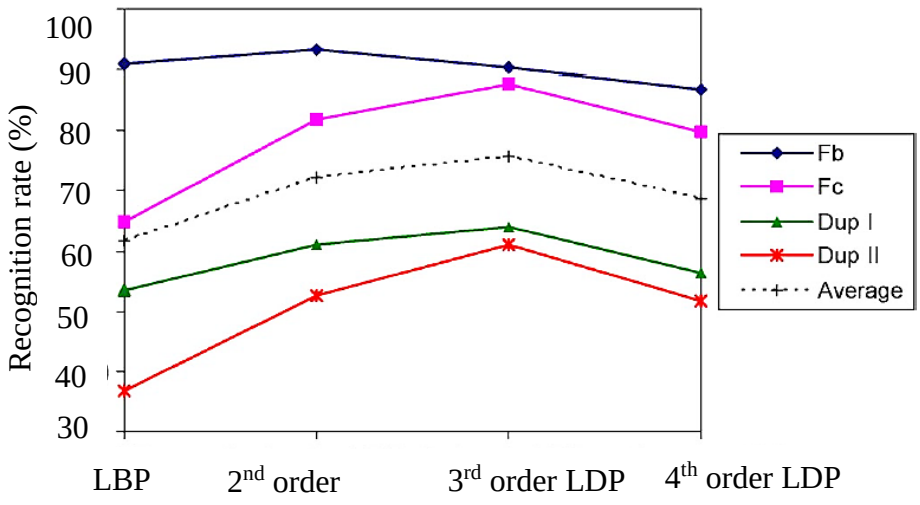


Figure 4.12 Results on the gray-level images

The average recognition rates on the four probe sets against different Gaussian noise are illustrated in Figure 4.12, showing that LDP maintains a 13.7% to 15.0% higher accuracy over LBP

This work investigates the feasibility and effectiveness of using high-order local pattern for face description and recognition. A Local Derivative Pattern (LDP) is proposed to capture the high-order local derivative variations. To model the distribution of LDP micro-patterns, an ensemble of spatial histograms is extracted as the representation of the input face image. LDP can be performed by using histogram intersection as the similarity measurement.

4.4 Experiments

We construct Shi et al.'s experiment [3] in MATLAB to steganalyze HUGO at 0.4 bpp on BOSSbase 0.92 which is different from the database in Shi's paper. Hence the result may be different from the accuracy in Shi's paper. Table 4.4 shows accuracies of each feature type by ensemble classifier [18].

Table 4.4 The result by reconstructed LBP, LTP and LDP

Feature Set	LBP	LTP	LDP
Whole	83.00%	83.21%	82.63%
Pes	74.41%	75.32%	73.63%
VARpe	68.61%	69.33%	68.56%
MEDpe	67.49%	66.45%	67.91%
LMbased	81.61%	82.31%	82.00%
MN13	82.01%	83.68%	80.19%
CL12	78.58%	80.00%	80.43%

where Feature Sets represent residual images. Pes means successive prediction error images and VARpe corresponds variance of Pes; MEDpe is median-filter-based prediction error; LMbased corresponds residual images based on Law's Masks; MN13 is high-pass filters based on Markov neighborhoods and CL12 is cliques filters.

Because of the updated database and different parameters, we can see the result in LBP is a little different from the Shi et al.'s paper. Even though the error is acceptable (lower than 1%).

LDP, which is very useful in texture detection and face detection, is imported to steganalysis in this thesis. Ironically, LDP does not work better than LBP as we thought but the result is very similar to LBP's. Meanwhile, feature sets point out that the LDP and LTP in complex filters bring higher accuracy compared with some simple mask like variance filter and medium filter. This is we didn't modify those filters for residuals images, which are suitable for LBP. Therefore, all results displayed above are very similar to LBP's. We cannot create some new useful filters for LTP and LDP because of lack of time. Theoretically, LDP is a directional pattern, and we choose only two direction in this thesis. In that case, some images will perform better, while some perform terribly. The output is not always stable, which makes a great influence to the results.

Therefore we deem the accuracy of LDP in steganalysis will increase in the future.

CHATPER 5

CONCLUSION

In this thesis, we have studied and given an overview of quite some number of steganalytic techniques for digital images. The techniques are broadly classified as specific and universal steganalysis. Universal statistical steganalysis are more robust as they are designed to detect messages embedded using any steganographic technique and without the knowledge of embedding technique. Specific detection method may be able to detect some specific steganographic scheme with a high detection accuracy, but with the development of Steganalysis, we can see that the accuracy of universal steganalysis is increasing and sometimes even higher than specific algorithm.

Furthermore, we reconstruct the texture feature for steganalysis and replace the LBP by using LDP and LTP. Even though the improvement in steganalytic capability is not as large as expected, but the result is still comparable, which means texture feature works in steganalysis. In 2015, it is reported that some researchers used deep learning in steganalysis which has never been used in this area before. Although the performance in steganalysis achieved by the initial trial in using deep learning technology has not met the expectation, it is expected it may be novel approach for steganalysis. It will be an active future research for steganalysis. Unfortunately, the performance has not met our expectation compared with rich models. The accuracy is about 4% lower than SRM's.

It is clear that the development for steganalysis and steganography will continue to move ahead. Our knowledge in this area will never end.

REFERENCES

[1] A. Westfeld, A. Pfitzmann, “Attacks on Steganographic Systems,” in 3rd Int. Workshop, LNCS1768, Springer-Verlag, Berlin, Heidelberg, 1999, pp. 61–76.

[2] N. Provos, “Defending Against Statistical Steganalysis,” Proc. 10th USENIX Security Symposium. Washington, DC, 2001.

[3] Y. Q. Shi, Patchara Sutthiwan, Licong Chen (2012). “Textural Features for Steganalys,” 14th Int. Conf., IH 2012, Berkeley, CA, USA, May 15-18, 2012.

[4] A. Westfeld, “High Capacity Despite Better Steganalysis (F5–A Steganographic Algorithm),” Information Hiding 4th Int. Workshop, Lecture Notes in Computer Science, Vol.2137. Springer-Verlag, Berlin Heidelberg, New York, 2001, pp. 289–302.

[5] P. Sallee, “Model-based methods for steganography and steganalysis,” in Int. Journal of Image and Graphics, 2005, pp. 5(1): 167-190.

[6] A. Miller. (May 2012). *Least Significant Bit Embedding: Implementation and Detection* [Online]. Available: http://www.aaronmiller.in/thesis/

[7] J. Fridrich, R. Du, L. Meng,“ Steganalysis of LSB Encoding in Color Images,” ICME 2000, New York City, July 31-August 2, New York, USA, 2000.

[8] J. Fridrich, M. Goljan, R. Du,“ Detecting LSB steganography in color and gray-scale images“, IEEE Multimedia Magaz., Special Issue on Security 22–28, 2001.

[9] F. Li, X. Zhang, B. Chen, G. Feng, “JPEG Steganalysis With High-Dimensional Features and Bayesian Ensemble Classifier”, Signal Processing Letters, IEEE, Vol.20(3), 2013, pp. 233-236

[10] J. Fridrich, J. Kodovský, “Rich Models for Steganalysis of Digital Images,” IEEE Trans. on Info. Forensics and Security, Vol. 7(3), 2012, pp. 868-882

[11] T. Denemark, J. Fridrich “Detection of Content-Adaptive LSB Matching (a Game Theory Approach),” Electronic Imaging, Media Watermarking, Security, and Forensics 2014, Vol. 9028, San Francisco, CA, 2015, pp. 125-141.

[12] B. Zhang, Y. Gao, S. Zhao, J. Liu, “Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor,” IEEE Transactions on Image Processing, Vol. 19, 2010, pp. 533-544.

[13] X. Tan, B. Triggs, “Enhanced Local Texture Feature Sets for Face Recognition Under

Difficult Lighting Conditions," IEEE Transactions on Image Processing, Vol.19(6), 2010, pp. 1635-1650.

[14] X. Yu, Y. Wang, T. Tan, "On Estimation of Secret Message Length in JSteg-like Steganography", ICPR 2004, Proc. of the 17th Int. Conf. Vol. 4, 2004, pp. 673-676.

[15] D. Zou, Y. Q. Shi, W. Su, G. Xuan, "Steganalysis Based on Markov Model of Thresholded Prediction-error Image", Multimedia and Expo, 2006 IEEE Int. Conf., 2006, pp.1365-1368.

[16] G. Xuan, X. Cui, Y. Q. Shi, W. Chen, X. Tong, C. Huang, "JPEG Steganalysis based on Classwise Non-Principal Components Analysis and Multi-Directional Markov Model," Multimedia and Expo, 2007 IEEE Int. Conf., 2006, pp. 903-906.

[17] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," SPIE 9409, Media Watermarking, Security, and Forensics, 2015, .

[18] J. Kodovský, J. Fridrich, Ensemble Classifiers for Steganalysis of Digital Media, Information Forensics and Security, IEEE Transactions,Vol.7(2), 2011, pp. 432-444.

[19] Tomáš Pevný, Tomáš Filler, Patrick Bas (2010), Using High-Dimensional Image Models to Perform Highly Undetectable Steganography, 12th Int. Conf., IH 2010, Calgary, AB, Canada, June 28-30, 2010, pp. 161-177.

[20] J. Fridrich, M. Goljan, D. Hogea,"Steganalysis of JPEG Images: Breaking the F5 Algorithm", 5th Information Hiding Workshop, Noordwijkerhout, The Netherlands, 2002, pp. 310-323.

[21] Y. Q. Shi, C. Chen, W. Chen, "A Markov Process Based Approach to Effective Attacking JPEG Steganography", 8th Int. Workshop, IH 2006, Alexandria, VA, USA, July 10-12, 2006, pp. 249-264.

[22] M. Shor. (2001) *Game Theory .net*, [Online]. Available: http://www.gametheory.net/

[23] Theano Development Team. (2008). *Convolutional Neural Networks*, [Online]. Available: http://deeplearning.net/tutorial/lenet.html

[24] V. Holub, J. Fridrich, "Random Projections of Residuals for Digital Image Steganalysis", IEEE Transactions on Information Forensic and Security, Vol.8(12), 2013, pp. 1996-2006.

[25] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," IEEE Trans. Patt. Anal. Mach. Intell., Vol. 20(3), Mar. 1998, pp. 226–239.

[26] T. Filler, J. Judas, and J. Fridrich. “Minimizing additive distortion in steganography using syndrome-trellis codes”, IEEE Transactions on Information Forensics and Security, Vol. 6(3), September 2011, pp. 920–935,

[27] T. Pevný, P. Bas, and J. Fridrich. “Steganalysis by subtractive pixel adjacency matrix”, IEEE Transactions on Information Forensics and Security, Vol. 5(2), June 2010, pp. 215–224.

[28] Y. LeCun, L, Bottou, Y, Bengio, P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE Vol. 86(11), 1998, pp. 2278–2324.

[29] Dave Marshall. (2001, Oct. 4). *The Discrete Cosine Transform*, [Online]. Available: http://www.cs.cf.ac.uk/Dave/Multimedia/node231.html

[30] M. Petrou, P. G. Sevilla, “Image Processing Dealing with Texture”, John Wiley & Sons Inc., 2006.

[31] G. Xuan, X. Cui, Y. Q. Shi, W. Chen, X. Tong, C Huang, “JPEG Steganalysis Based on Classwise Non-principal Components Analysis and Multi-directional Markov Model”, Multimedia and Expo, 2007 IEEE Int. Conf. on, 2-5 July 2007, pp. 903-906.

[32] O. Ivanciuc. (2005) *Support Vector Machines*, [Online]. Available: http://www.support-vector-machines.org/

[33] K. Sullivan, U. Madhow, S. Chandrasekaran, B.S. Manjunath, “Steganalysis of Spread Spectrum Data Hiding Exploiting Cover Memory”, SPIE2005, vol. 5681, pp. 38-46.

[34] T. Ojala, M. Pietikainen, D. Harwood, “A Comparative Study of Texture Measures with Classification Based on Feature Distributions”, Pattern Recognition, Vol. 29, 1996, pp. 51–59.

[35] J. Fridrich, J. Kodovský, V. Holub, M. Goljan, “Steganalysis of Content-Adaptive Steganography in Spatial Domain,” Information Hiding, 13th Int. Workshop, Lecture Notes in Computer Science, Prague, Czech Republic, May 18–20, 2011, pp. 102-117.

[36] J. Fridrich, “Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes,” 6th Information Hiding Workshop, Toronto, ON, Canada, 2004.

[37] H. Farid, “Detecting hidden messages using higher-order statistical models,” Int. Conf. on Image Processing, Rochester, NY, USA, 2002.

[38] Y. Q. Shi, G. Xuan, D. Zou, J. Gao, C. Yang, Z. Zhang, P. Chai, W. Chen, and C. Chen, "Steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network," Int. Conference on Multimedia and Expo, Amsterdam, Netherlands, 2005.

[39] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," 6th Information Hiding Workshop, Toronto, ON, Canada, 2004

[40] Niels Provos. (Oct 2 2001) *OutGuess*, [Online]. Available: http://www/outguess.org

[41] J. Kodovsky and J. Fridrich, "Calibration revisited," Proc. 11th ACM Multimedia &Security Workshop, 2009, pp. 7–8.

[42] P. Sallee, "Model-based methods for steganography and steganalysis," J. Image Graph., vol. 5(1) , 2005, pp. 167–190.

[43] W. Luo, F. Huang, and J. Huang. Edge adaptive image steganography based on LSB matching revisited. IEEE Transactions on Information Forensics and Security, 5(2), June 2010, pp. 201–214.

[44] J. Kodovský, T. Pevný, and J. Fridrich. Modern steganalysis can detect YASS. In .D. Memon, E.J. Delp, P.W. Wong, and J. Dittmann, editors, Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII, volume 7541, San Jose, CA, January 17–21, 2010, pp. 02–01–02–11.