STATISTICAL STEGANALYSIS OF IMAGES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Min Huang

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Prof. Vernon Rego, Chair

Department of Computer Science

Prof. Samuel Wagstaff

Department of Computer Science

Prof. Bharat Bhargava

Department of Computer Science

Prof. Wojciech Szpankowski

Department of Computer Science

Approved by:

Prof. Voicu Popescu

Head of the department Graduate Program

To my family.

ACKNOWLEDGMENTS

I would like to express my gratitude and appreciation to my advisor, Professor Vernon Rego, for his support during my studies at Purdue University. Also, I would like to give a special thank you to Professor Wojciech Szpankowski for providing me with the support from the Center for the Science of Information, a Science and Technology Center at Purdue University. Besides, I would like to sincerely thank the rest of my committee, Professors Samuel Wagstaff and Bharat Bhargava for their guidance in this work.

I would like to extend special thanks to Dr. William J. Gorman and Monica Shively in the Department of Computer Science. With their help, I was provided with teaching assistantships and had experience working as a Graduate Teaching Assistant.

Finally, I would like to thank my family for their continuous support. I could not have completed my PhD study at Purdue University without them.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Huang, Min Ph.D., Purdue University, August 2019. Statistical Steganalysis of Images. Major Professor: Vernon J. Rego.

Steganalysis is the study of detecting secret information hidden in objects such as images, videos, texts, time series and games via steganography. Among those objects, the image is the most widely used object to hide secret messages. Detection of possible secret information hidden in images has attracted a lot of attention over the past ten years. People may conduct covert communications by exchanging images in which secret messages may be embedded in bits. One of main advantages of steganography over cryptography is that the former makes this communication insensible for human beings. So statistical methods or tools are needed to help distinguish cover images from stego images.

In this thesis, we start with a discussion of image steganography. Different kinds of embedding schemes for hiding secret information in images are investigated. We also propose a hiding scheme using a reference matrix to lower the distortion caused by embedding. As a result, we obtain Peak Signal-to-Noise Ratios (PSNRs) of stego images that are higher than those given by a Sudoku-based embedding scheme. Next, we consider statistical steganalysis of images in two different frameworks. We first study staganalysis in the framework of statistical hypothesis testing. That is, we cast a cover/stego image detection problem as a hypothesis testing problem. For this purpose, we employ different statistical models for cover images and simulate the effects caused by secret information embedding operations on cover images. Then the staganalysis can be characterized by a hypothesis testing problem in terms of the embedding rate. Rao's score statistic is used to help make a decision. The main advantage of using Rao's score test for this problem is that it eliminates an

assumption used in the previous work where approximated log likelihood ratio (LR) statistics were commonly employed for the hypothesis testing problems.

We also investigate steganalysis using the deep learning framework. Motivated by neural network architectures applied in computer vision and other tasks, we propose a carefully designed a deep convolutional neural network architecture to classify the cover and stego images. We empirically show the proposed neural network outperforms the state-of-the-art ensemble classifier using a rich model, and is also comparable to other convolutional neural network architectures used for steganalysis.

1 INTRODUCTION

Steganography is the art and science of covert communication, whose goal is to hide secret messages within innocuous-looking cover objects. Currently, most of steganographic schemes are developed on the image domain since they are easy to be implemented and able to hide secret information in plain sight, and most importantly have a great amount of embedding capacity on the image domain. Steganalysis, on the other hand, is the art and science of detecting whether a given object is hiding secret information. The image steganalysis is based on the assumption that reasonably small changes in images may result in detectable changes in some image statistics. Finding those changes is the core of the image staganalysis.

1.1 Information Hiding

We begin with a brief discussion of the difference between steganography and cryptography. Generally speaking, both steganography and cryptography are information hiding in the sense of covert communication. The critical difference between them is that the former hides the *existence* of secret information whereas the latter hides the *content* of secret information, which leads to quite different information hiding schemes as well as detection schemes. We will consider steganography and focus on its detection in this thesis.

Steganography has a long history and a nice introduction can be found in [Fri09]. Modern steganography is based on the model from the famous prisoners' problem [Sim84]. That is, Alice and Bob are prisoners and allowed to conduct communication which is observed by Warden. If Warden finds any steganographic schemes are used to pass secret information, then Alice and Bob may be punished. In this case, Warden is considered successfully detecting steganography. One assumption in this model that

Warden has complete knowledge of information hiding schemes that Alice and Bob might apply. This is the well-known Kerckhoffs' principle. It is worth noting that this principle is still commonly used in the study of staganalysis even though it is less practical. The blind steganalysis is also widely investigated which weakens the principle at the expense of detection performance.

In image steganography, least significant bits (LSBs) of image pixel values or Discrete Cosine Transform (DCT) coefficients (of JEPG images) are slightly modified by steganographic methods with the hope of undetectability of the modifications. Essentially, those modifications are very small and values of pixels or DCT coefficients are changed at most by one. Two things need to be addressed here. The first thing is how the value of a given pixel or a DCT coefficient is modified, if we know, for instance, that the value (which is an integer) is changed by one. The approach using the LSB replacement is to simply flip the LSB of the value, and the approach using the LSB matching iss to randomly modify the value by one. The former results in asymmetry which can be exploited to construct effective detectors. We will discuss this in the next chapter in more details. The second thing is how to determine a set of pixels or coefficients that are used for information embedding. The most popular approach was to randomly choose those pixels or coefficients using random seeds before 2010. Such a choice is simple and the scheme is easy to be implemented, see [Mie06]. But the embedding scheme is independent of the image content and is thereby less secure [PBF10]. The content adaptive embedding scheme became popular after a challenge was proposed [BFP11]. Such a content adaptive embedding scheme is to minimize distortion functions such that smooth regions in images are less likely to be chosen for embedding. In other words, noise areas such as edges have high probability to be used for embedding. Many steganographic methods based on the contend adaptive embedding scheme have been proposed, such as HUGO, WOW, S-UNIWARD and MiPOD, see, e.g., [PFB10,HF12,LWHL14,SCF16] and the references therein.

1.2 Information Detection

Statistical steganalysis is the study of information detection using statistical tools. Distortion in images caused by carefully designed steganographic methods is hardly detected by human eyes. Statistical methods need to be employed to conduct the detection task. As seen above, this task is a binary classification task. Namely, we have to decide if given images are cover (clean) or stego (dirty) images. Two frameworks are considered in image staganalysis: statistical hypothesis testing and machine learning/deep learning. The former is based on cover image models and the latter relies on feature extraction.

In the hypothesis testing framework, the values of pixels or DCT coefficients are assumed to have distributions such as generalize Gaussian distributions. The embedding conducted by steganographic methods may be modelled as changes of model/distribution parameters, see, e.g., [CZF$^+$11,ZCR$^+$11,Fil12,CZR$^+$12]. It should be pointed out that the i.i.d. assumption or the independent assumption is used in the distribution model. This is impractical but the model seems to work well for certain steganographic methods, as demonstrated by experiments in the references above as as well as in Chapter 3. However, from my perspective this framework has difficulty dealing with content adaptive steganographic methods. It is more common to use machine leaning/deep learning methods for complex steganographic methods as we will show in Chapter 4.

Machine learning is naturally applied in steganalysis since the detection of secret information is regarded as a classification task. Lyu *et. al.* [LF04] first used wavelet statistics (as features) and the surport vector machine (SVM) for image steganalysis. Following this pioneering work, steganalysis based on machine learning attracted much attention. A great deal of work for steganalysis focused on feature extraction has been done since then, see, e.g., [Ker05b, GFH06, PF06, LF06, PF07, PF08, RG10] and the references therein. For steganalysis using the machine learning framework, feature extraction is the main focus. Even though classification methods are very important

in steganalysis, we usually employ well-developed classification methods (e.g., SVM, ensemble methods) for the task. Also, dimension reduction techniques dealing with high-dimensional feature space are extensively used for steganalysis, such as PCA, random projection, see, e.g., [FQY07, QSXN09, KF11, HF13, HF15]. Before 2015, image steganalysis was greatly involved in seeking handcrafted feature which heavily relies on domain expertise. Much effort has been made by many researchers through years and eventually a collection of more than 30,000 features has completed. The model with these features is called the *rich model* [KFH12]. Note that the complete model has a high-dimensional feature space which may require a lot of memory and a great deal of computations during training.

Deep learning is a subfield of machine learning and it uses deep neural networks to extract hierarchical features automatically through the training process. With the success of deep neural networks in many tasks such as computer vision [KSH12], deep learning was introduced in image steganalysis in 2015 by Qian *et. al.* [QDWT15]. The results obtained by them in the paper were promising though the method using deep neural works fell short of catching the ensemble method with the rich model in performance. The focus of image steganalysis has been shifted to the design of deep neural work architectures since then, see, e.g., [PPIC16, CCS17, XWS16b, YNY17, YCC18b, BCF19] and the references therein.

1.3 Contribution

We first consider a information hiding problem. We develop an embedding scheme using a reference matrix to improve the quality of stego images by enhancing PSNRs. We compare the proposed embedding scheme with previously used schemes based on Sudoku and show the former achieves higher PSNRs than the latter by experiments. Next, we investigate image steganalysis using the hypothesis testing framework. We consider an inhomogeneous cover image model and the least significant bit (LSB) matching scheme for embedding. The resulting stego images can be represented by

an inhomogeneous Gaussian mixture model which contains the parameter of interest: the embedding rate. The cover/stego image detection problem is cast by a hypothesis testing problem in terms of the embedding rate. We theoretically analyze the Rao's score test for this problem and show the score test is not only asymptotically most powerful (AMP) but locally asymptotically uniformly most powerful (LAUMP). Compared with the widely used likelihood ratio (LR) test, the score test does not rely on an assumption on the image model, i.e., the local variances are greater than one, which is applied in the LR test and considered unrealistic. Also, experiments are carried out on two image datasets for the comparison of the two statistical tests. The results demonstrate that the score test performs reasonably well and outperforms the LR test when the embedding rate is small. We also use deep learning for image steganalysis. Observing that high-pass filters play an import role in the design of convolutional neural networks (CNNs) for steganalysis, we propose a CNN architecture whose first layer is a convolutional layer and consists of kernels of high-pass filters. These kernels are trainable and we add some constraints (e.g., symmetry and zero-sum of kernel elements) on them so that the first layer can perform high pass filtering during the neural network training process. It is worth pointing out that many CNN architectures for image steganalysis employ fixed kernels to conduct high-pass filtering for input images. The fixed kernels are not trainable thereby cannot learn from the training data. Some CNN architectures for image steganalysis use trainable kernels and initialize them with the kernels of high-pass filters. These trainable kernels are unlikely to continue performing high-pass filtering since the kernel parameters/weights are changing during the training process and some properties (e.g., zero-sum) are no longer guaranteed. The design of the first layer in our proposed CNN architecture combines the two cases. Also, we introduce the residual modules in our architecture which allows us to build a deep CNN architecture. For comparison, we test the proposed architecture , the rich mode with ensemble learning (which is the state-of-the-art using handcrafted features) and other popular CNN architectures for steganalysis on a large image dataset. Three complex embedding schemes are consid-

ered and the experimental results show the proposed CNN architecture gains better performance in most cases.

2 IMAGE STEGANOGRAPHY

Steganography is the art and science of hiding the existence of secret information which is embedded into cover objects in a way that the resulting stego objects do not raise suspicion. Images are widely employed as cover objects by steganographic methods, although other types of objects which contain redundancy are applicable as well. For images, commonly used steganographic methods are based on changes of LSBs. More specifically, steganographic methods change the LSB of the pixel values for spatial domain images, and change the LSB of transformation coefficients (e.g., DCT coefficients) for frequency domain images (e.g., JPEG images). That is, secret information represented by a string of binary digits is hidden in an image by modifying LSBs of pixels or coefficients of the image.

2.1 LSB Replacement vs Matching

We start with a simple example which describes the LSB replacement and matching. Six values of either pixels or coefficients are chosen to embed a secret message coded as a binary string 011010. The six values are assume to be 7, 7, 8, 6, 5, 5. Alice (the sender) would possibly modify the values such that the LSBs of the modified six values are consistent with the secret binary string 011010. The LSBs of the original six values are 110011. It is easily seen that the first value 7, the third value 8 and the last value 5 need to be changed. For the LSB replacement, we simply flip the LSB of the three values and so the three values become 6, 9 and 4. The six values that Bob (the receiver) obtain would be 6, 7, 9, 6, 5, 4 and he can easily extract the secret binary string 011010 from the LSBs. It should be pointed out that Bob would know where he can get the six values by the Kerckhoffs' principle. Notice that there is a pattern for the LSB replacement. That is, the odd value always decreases and

the even value always increases when the LSB replacement applies. A lot of detectors were proposed by exploiting this pattern, such as sample pairs analysis, structure steganalysis and the weighted-stego method, see, e.g., [DWW02, Ker05a, KB08].

The LSB matching aims at eliminating the asymmetry by randomly decreasing or increasing the values by one. Again, consider the example above. The first value 7 could be randomly changed to either 8 or 6. The same approach applied for the other two values. Note that the value is modified by one using the LSB matching but more LSBs may get involved in modifications. A description of the LSB matching using mathematical notations will be introduced in the next chapter when the hypothesis testing is discussed.

2.2 Random vs Content Adaptive Embedding Schemes

As mentioned before, the choice of a set of pixels or coefficients used for information embedding (e.g., the LSB replacement or matching) is an important topic in image steganography. For the random embedding scheme, Alice and Bob may share random seeds (by the Kerckhoffs' principle) so that both of them know the set of pixels or coefficients. It is easy for Bob to extract secret information coded by a binary string. As shown in the example above, Bob simply looks up to the LSBs of the pixels from the set in a particular order by using, for instance, random seeds. A disadvantage of the random embedding is that it is independent of the image content. This may result in a lot of distortion in the smooth regions of the cover image. To address this issue, the content adaptive embedding scheme was introduced by minimize a distortion cost function.

Consider a grayscale cover image denoted by $\mathbf{X} = \{x_1, x_2, ...x_N\}$. Here we ignore the correlation among pixels for simplicity and use a sequence for the two-dimensional image. We denote its stego image by $\mathbf{Y} = \{y_1, y_2, ...y_N\}$, where $x_i, y_i \in$

$\{0, 1, ..., 255\},\ i = 1, 2, ..., N$. To evaluate the embedding distortion, a distortion cost function $\rho_i(\mathbf{X}, y_i)$ is introduced. We specify $\mathbf{Y}$ by minimizing the following distortion

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

where ρ_i are the costs of the change of x_i to y_i. Different options of the cost function ρ lead to different content adaptive embedding schemes such as HUGO, WOW, S-UNIWARD and MiPOD, see, e.g, [HFD14, LWHL14, SCF16] and the references therein.

Figure 2.1 shows a cover image, its stego image and the difference between them. The LSB matching with the random embedding scheme is applied here. For comparison, a content adaptive embedding scheme, namely S-UNIWARD, is used for the same cover image, see Figure 2.3. It can be seen from the comparison that Figure 2.1 exhibits the randomness and Figure 2.3 shows the adaptivity in terms of embedding location. Figure 2.2 demonstrates that the embedding scheme WOW is also content adaptive.

2.3 Sudoku-based Steganography

In this section, a concrete example using steganogrphy is provided. I would like to point out that this example is not content adaptive. It shows image steganography could be conducted by other approaches.

We start with the descrpition of Sudoku-based steganographic methods for hiding secret information in images. Sudoku solutions play an important role in constructing a reference matrix based on which pixel values of a cover image are modified to hide secret information. More specifically, suppose we have a Sudoku solution which can written as a 9×9 matrix S. Let $B(i, j)$ denote the entry in the i-th row and the j-th column of matrix B. and A reference matrix M associated with a Sudoku solution S is constructed by setting

$$M(i, j) = S(i', j') - 1, \qquad i' = i \bmod 9, \quad j' = j \bmod 9. \tag{2.1}$$

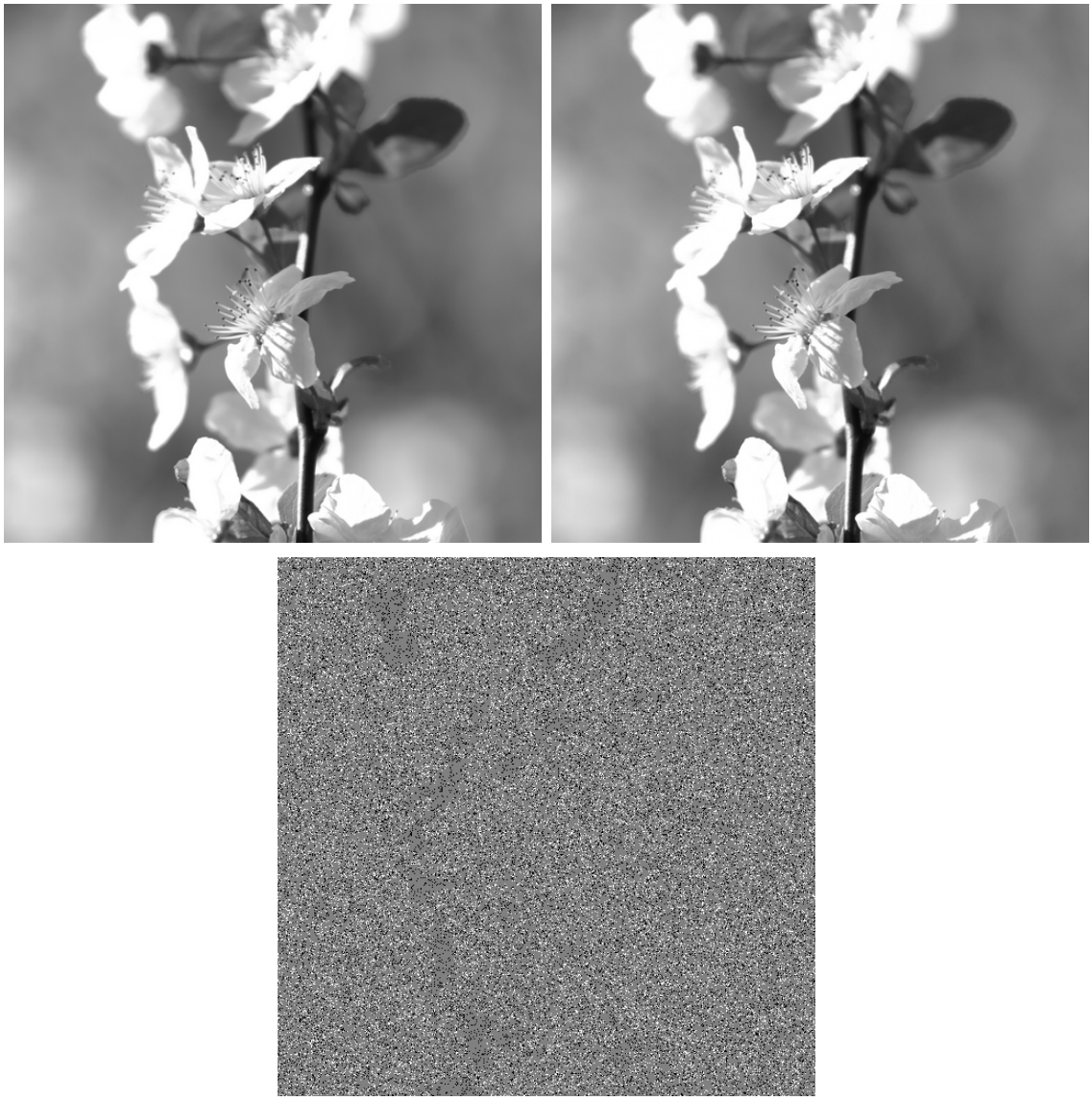


Figure 2.1. Embedding: LSB Matching 0.4bpp; Top Left: cover image; Top Right: stego image; Bottom: changed pixels: white $\rightarrow$ 1, black $\rightarrow$ -1

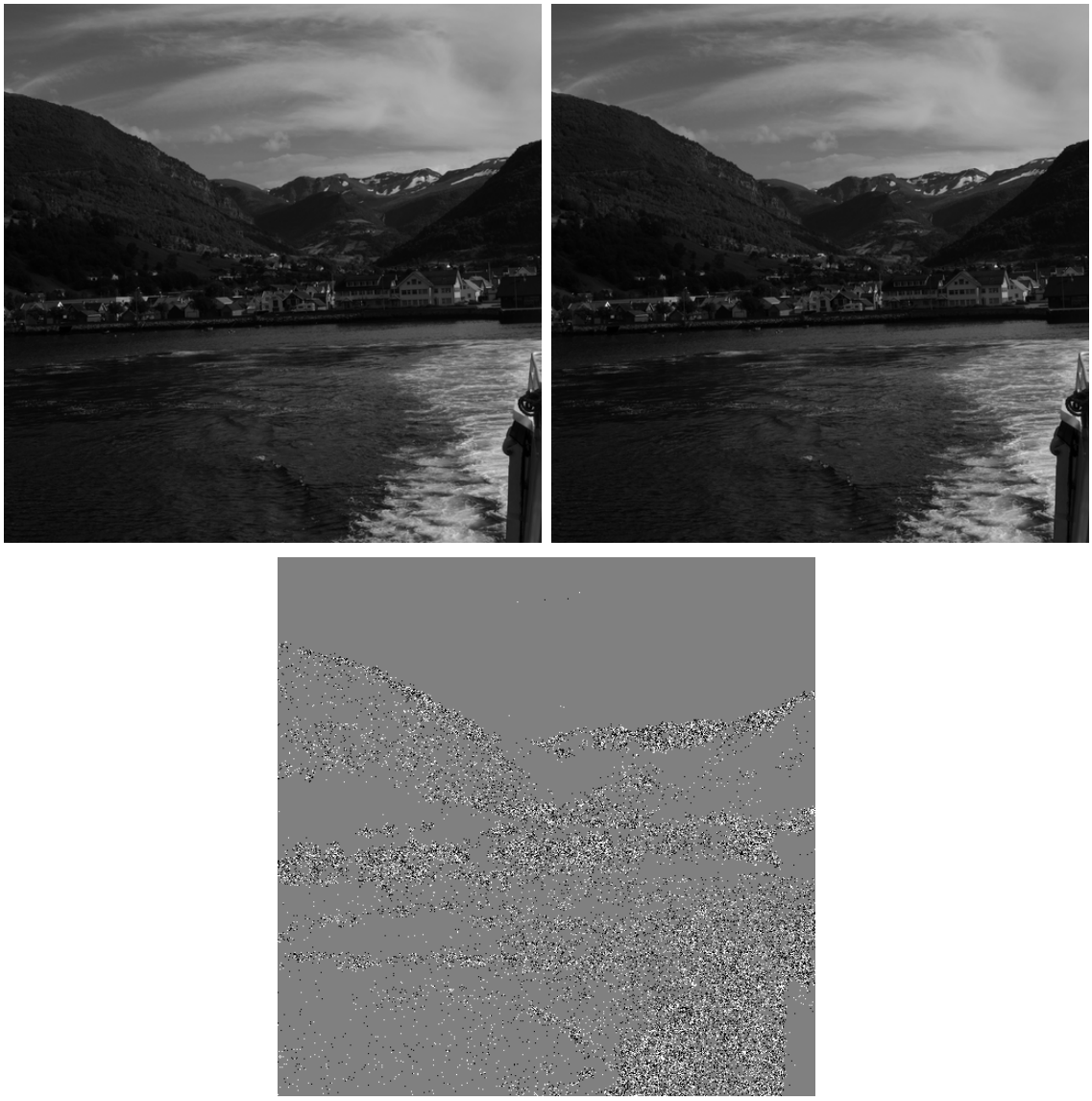


Figure 2.2. Embedding: WOW 0.4bpp; Top Left: cover image; Top Right: stego image; Bottom: changed pixels: white $\rightarrow$ 1, black $\rightarrow$ -1

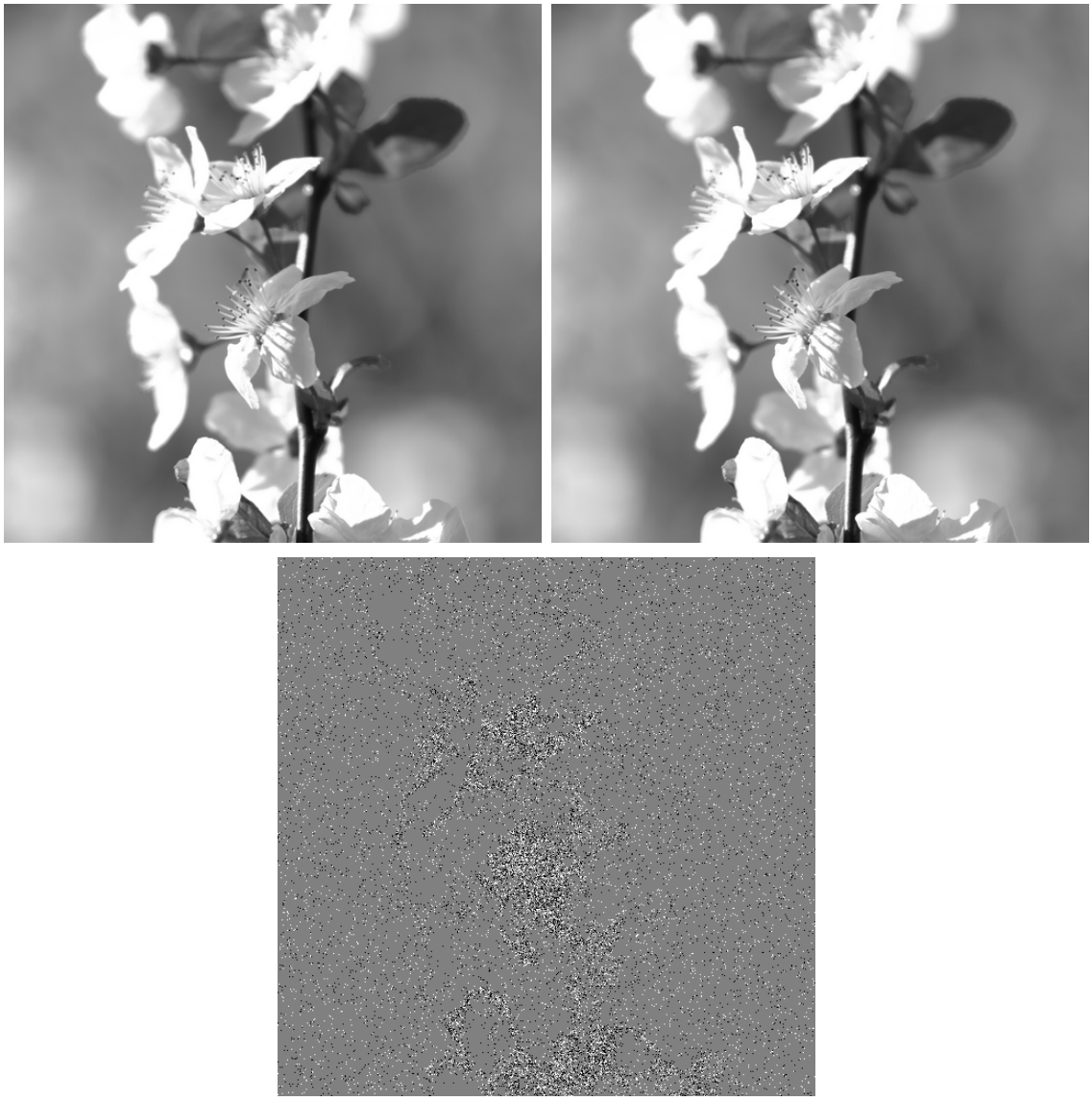


Figure 2.3. Embedding: S-UNIWARD; 0.4bpp; Top Left: cover image; Top Right: stego image; Bottom: changed pixels: white $\rightarrow$ 1, black $\rightarrow$ -1

Here and throughout the rest of this section, row and column indices of a matrix start from 0. In order to embed secret information into a given grayscale cover image, we first generate a sequence of pixels by randomly permuting all of the pixels of the cover image via a seed and then create a list L of non-overlapping pixel pairs by simply pairing up adjacent pixels in the sequence. Since pixel values of a grayscale image are integers between 0 and 255, the size of the reference matrix M is fixed to be 256×256. The following is a Sudoku solution used in [HCS08].

$$\mathrm{S} = \left(\begin{array}{ccc|ccc|ccc} 7 & 8 & 2 & 4 & 9 & 1 & 5 & 3 & 6 \\ 1 & 4 & 6 & 5 & 7 & 3 & 9 & 2 & 8 \\ 5 & 3 & 9 & 6 & 2 & 8 & 7 & 4 & 1 \\ \hline 3 & 5 & 8 & 1 & 6 & 4 & 2 & 9 & 7 \\ 4 & 9 & 1 & 7 & 5 & 2 & 8 & 6 & 3 \\ 6 & 2 & 7 & 3 & 8 & 9 & 4 & 1 & 5 \\ \hline 2 & 7 & 5 & 9 & 3 & 6 & 1 & 8 & 4 \\ 8 & 1 & 3 & 2 & 4 & 5 & 6 & 7 & 9 \\ 9 & 6 & 4 & 8 & 1 & 7 & 3 & 5 & 2 \end{array}\right). \tag{2.2}$$

Notice that each row or column in any Sudoku solution contains exactly nine distinct digits from 1 to 9. Also, there are nine non-overlapping 3×3 blocks with each of them having distinct digits from 1 to 9. This property of Sudoku solutions are employed in [CCK08, HCS08] for designing search algorithms. The secret key shared by both the sender and the receiver consists of a seed, a reference matrix and a possibly used symbol indicating the end of secret information.

2.3.1 Embedding Procedure

Assume that the secret information including an ending symbol can be represented by a sequence R of secret digits in the 9-ary notational system. Let r_i and (x_i, y_i) denote the i-th secret digit in the sequence R and the i-th pair in the list L,

respectively. Note that each pixel pair in the list L corresponds to a location in the reference matrix M. The embedding procedure is performed as follows.

(i) Initially, $i \leftarrow 1$;

(ii) Find location (u, v) in M such that $M(u, v) = r_i$ and (u, v) is closest to (x_i, y_i) in L_1 (or L_2) norm;

(iii) Modify the pixel pair (x_i, y_i) by $(x_i, y_i) \leftarrow (u, v)$;

(iv) $i \leftarrow i + 1$;

(v) Repeat (i)-(iv) until $i >$ length(R) or $i >$ length(L).

Taking advantage of the property of a Sudoku solution, a search in [CCK08] for the closest location (u, v) to (x_i, y_i) in L_1 norm (i.e., Manhattan distance) in the embedding procedure above is carried out over three blocks containing (x_i, y_i): a horizontal 1×9 block, a vertical 9×1 block and a 3×3 block. Specifically, in the case of $3 < x_i < 252$ and $3 < y_i < 252$, the three blocks above are

$$\begin{pmatrix} \mathrm{M}(x_i, y_i - 4) & \cdots & \mathrm{M}(x_i, y_i + 4) \end{pmatrix}, \quad \begin{pmatrix} \mathrm{M}(x_i - 4, y_i) \\ \vdots \\ \mathrm{M}(x_i + 4, y_i) \end{pmatrix},$$

$$\begin{pmatrix} \mathrm{M}(s, t) & \mathrm{M}(s, t+1) & \mathrm{M}(s, t+2) \\ \mathrm{M}(s+1, t) & \mathrm{M}(s+1, t+1) & \mathrm{M}(s+1, t+2) \\ \mathrm{M}(s+2, t) & \mathrm{M}(s+2, t+1) & \mathrm{M}(s+2, t+2) \end{pmatrix},$$

where s and t satisfy $x_i = 3 \cdot s + z,\ 0 \le z < 3$, and $y_i = 3 \cdot t + w,\ 0 \le w < 3$, respectively. In other cases, we may need to shift the horizontal block or the vertical block. A detailed description of the search algorithm on these three blocks can be found in [CCK08].

It was observed in [HCS08] that the closest location obtained [CCK08] from the search over the three blocks above may not be globally optimal over the whole reference matrix M. That is, for some given pixel pair (x_i, y_i) in a cover image and some secret digit r_i, there may exist a location in M, say (x_i', y_i'), which satisfies

$M(x_i', y_i') = r_i$ and is closest to (x_i, y_i) in L_1 norm but is not contained in any of the three blocks associated with (x_i, y_i) above. Based on this observation, a search over a larger range for the closest location was proposed in [HCS08] to achieve the globally optimal solution, which results in obtaining PSNRs for stego images higher than those in [CCK08].

It should be noted that the only L_1 norm is employed in [CCK08, HCS08] for finding the closest location in the embedding procedure. For a pixel pair (x, y), there may be two qualified closest location, say $(x, y+2)$ and $(x+1, y+1)$, in L_1 norm, and which one is chosen depends on the search algorithm. It could be either of the two location. However, only the location $(x+1, y+1)$ is chosen if L_2 norm is applied. It should be more suitable to use L_2 norm because the mean-squared error between a pixel pair and its closest location is used in the computation of PSNRs. The lower mean-squared error, the higher PSNR. As we can see, the mean-squared error is associated with L_2 norm instead of L_1 norm.

2.3.2 Extracting Procedure

Using the secret key, the receiver can generate a list L of pixel pairs from the received stego image. Then, the extracting procedure is performed as follows.

(i) Initially, $i \leftarrow 1$;

(ii) Extract the i-th pixel pair (x_i, y_i) from L and obtain the i-th secret digit $M(x_i, y_i)$ from matrix M;

(iii) $i \leftarrow i + 1$;

(iv) Repeat (i)-(iv) until $i >$ length(L) or an ending symbol is obtained.

2.3.3 Improvements by New Reference Matrices

we next show that distortion caused by the embedding procedure may be reduced by replacing a reference matrix constructed from a Sudoku solution with a new one

which will be described below. Consider a pixel pair (u, v) in a grayscale cover image with both u and v being unsaturated, i.e., $0 < u, v < 255$. It is observed that u and v may be increased or decreased by up to two by the embedding algorithm in the section 2.1, if a reference matrix associated with a Sudoku solution is used. For instance, suppose the pixel pair and secret digit are $(3, 4)$ and 2, respectively. Using an improved search algorithm employed in [HCS08] and the reference matrix M constructed from (2.1) and (2.2), the closest qualified locations to $(3, 4)$ are $(1, 5)$ and $(5, 3)$ in L_2 norm. So either v or u needs to be decreased or increased by two in order to hide the secret digit. To further lower the distortion, we now present a new construction of the reference matrix M used in the embedding procedure. The steganographic method based on this newly constructed reference matrix has the same embedding capacity as that in [HCS08] but gains higher PSNRs, which will be demonstrated by experimental results in the next section. The new construction of the reference matrix is described as follows. First, we create a 3×3 matrix T containing exactly nine integers from 0 to 8. The reference matrix M associated with T is constructed by setting

$$M(i, j) = T(i', j'), \quad i' = i \bmod 3, \quad j' = j \bmod 3, \quad i,\ j = 0, 1, ..., 255. \tag{2.3}$$

It can be easily seen that there are totally $9! = 362880$ different choices for matrix T. We can arbitrarily choose one of them. As an example, we let

$$\mathrm{T} = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix}. \tag{2.4}$$

Thus, the reference matrix M constructed from (2.3) and (2.4) looks like

$$\mathrm{M} = \begin{pmatrix} 0 & 1 & 2 & 0 & 1 & 2 & 0 & \cdots & 0 \\ 3 & 4 & 5 & 3 & 4 & 5 & 3 & \cdots & 3 \\ 6 & 7 & 8 & 6 & 7 & 8 & 6 & \cdots & 6 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & \cdots & 0 \\ 3 & 4 & 5 & 3 & 4 & 5 & 3 & \cdots & 3 \\ 6 & 7 & 8 & 6 & 7 & 8 & 6 & \cdots & 6 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & \cdots & 0 \end{pmatrix}_{256\times 256} \tag{2.5}$$

Note that any 3×3 block in matrix M constructed from (2.3) contains exactly nine integers from 0 to 8. This can be proved as follows. Assume that there exists a 3×3 block in matrix M having the same integer at two different locations. We denote by (p_1, q_1) and (p_2, q_2) the two different locations, respectively, in matrix M. So $M(p_1, q_1) = M(p_2, q_2)$. It follows from the construction (3) that $M(p_1, q_1) = M(p_2, q_2)$ if and only if

$$(p_1 - p_2) = 0 \bmod 3 \quad \text{and} \quad (q_1 - q_2) = 0 \bmod 3.$$

This contradicts that both of (p_1, q_1) and (p_2, q_2) are in a 3×3 block. Therefore, all 3×3 block in matrix M constructed from (2.3) have distinct integers from 0 to 8.

For an unsaturated pixel pair (u, v) in a grayscale cover image, the above property of matrix M guarantees that either u or v is increased or decreased by at most one. To see this, we consider a 3×3 block in matrix M in the following

$$\begin{pmatrix} \mathrm{M}(u-1, v-1) & \mathrm{M}(u-1, v) & \mathrm{M}(u-1, v+1) \\ \mathrm{M}(u, v-1) & \mathrm{M}(u, v) & \mathrm{M}(u, v+1) \\ \mathrm{M}(u+1, v-1) & \mathrm{M}(u+1, v) & \mathrm{M}(u+1, v+1) \end{pmatrix}. \tag{2.6}$$

Since the block above contains all of the nine digits from 0 to 8, the closest qualified location to (u, v) in L_2 norm must be in this block. Thus, either u or v only needs to

be changed by at most one for hiding any of the nine digits. As mentioned earlier in an example, u or v in an unsaturated pixel pair may be increased or decreased by two if the reference matrix is constructed from a Sudoku solution by using (2.1). So the new construction (2.3) of the reference matrix M provides possibilities for unsaturated pixel pairs to reduce distortion caused by the embedding procedure. Pixel values in a saturated pixel pair may be changed by up to two in our proposed steganographic method. But note that the number of saturated pixels is a small portion of the total pixel number in most of nature images. Improvements on the quality of stego images can be expected by using the reference matrix M constructed from (2.3), as will be shown next.

The embedding and extracting procedure in our proposed steganographic method is the same as that appearing in the previous subsection, except that the reference matrix M constructed from (2.1) is replaced with the one constructed from (2.3).

2.3.4 Experimental results

In this section, we report performance of the Sudoku-based steganographic method used in [CCK08, HCS08] and our proposed steganographic method. To evaluate the quality of stego images, the PSNR (Peak Signal-to-Noise Ratio) is widely used, see, e.g., [CCK08, HCS08, JY09, LCW08], which is defined, for a grayscale stego image N' with size $r \times c$, as

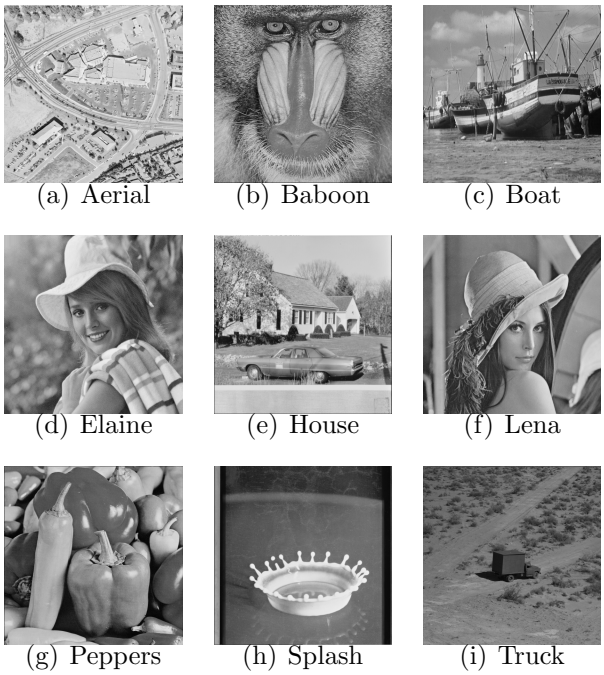
$$\text{PSNR} \;=\; 10 \cdot \log_{10} \frac{255^2}{\text{MSE}},$$

where the MSE (mean-squared error) between the stego image N' and the corresponding cover image N is defined as

$$\text{MSE} \;=\; \frac{1}{rc} \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} \big[N(i,j) - N'(i,j)\big]^2.$$

Nine 512×512 images are chosen from USC-SIPI image database [Web18], among which Baboon, House, Lena, Peppers and Splash were converted into grayscale images using Matlab function *rgb2gray*, see Figure 2.4. The nine images are employed as

cover images and tested by the Sudoku-based steganographic method [CCK08,HCS08] and our proposed method using new reference matrices. All of the three steganographic methods have the same embedding capacity, namely $(\log_2 9)/2$ bits per pixel. At the maximum embedding rate, a sequence of secret digits produced by using a pseudo-random number generator is embedded into the nine cover images by employing the three methods, respectively. The reference matrix M used in this experiment is constructed from (2.1) and (2.2) for the Sudoku-based steganographic method in [CCK08, HCS08], and is for the proposed method. A comparison of performance of the methods is given in Table 2.1, in terms of the PSNR. As shown, the Sudoku-based steganographic method in [HCS08] results in an average PSNR 0.6333 db higher than that obtained by the method in [CCK08]. In the meantime, the PSNR caused by the proposed method is 1.5824 higher than that on average achieved by the method in [HCS08]. It is demonstrated by the experimental results that the proposed method reduces the distortion causing by embedding, compared with the Sudoku-based steganographic method in [CCK08, HCS08].



(a) Aerial (b) Baboon (c) Boat

(d) Elaine (e) House (f) Lena

(g) Peppers (h) Splash (i) Truck

Figure 2.4. Nine 512×512 grayscale images

Table 2.1.
A comparison of performance of three steganographic methods

	PSNRs		
Images	The method in [CCK08]	The method in [HCS08]	The proposed method
Aerial	47.6651	48.3037	49.8787
Baboon	47.6603	48.2918	49.8931
Boat	47.6721	48.3108	49.8886
Elaine	47.6614	48.2954	49.8998
House	47.6391	48.2858	49.8969
Lena	47.6650	48.2972	49.8855
Peppers	47.6658	48.3031	49.8934
Splash	47.6553	48.2966	49.8863
Truck	47.7878	48.3869	49.8971
Average	47.6746	48.3079	49.8903

3 STATISTICAL HYPOTHESIS TESTING FOR STEGANALYSIS

3.1 Introduction

We investigate in this chapter image steganalysis in the framework of statistical hypothesis testing. The main advantage of this approach is that the theoretical analysis of the detector can be established once the problem is well modeled. But the performance of detectors or statistical tests based on hypothesis testing would be compromised if cover images or embedding operations are not well modeled.

Currently, the most commonly used test statistic for steganalysis in the framework of hypothesis testing is an approximated log LR statistic, and the cover image model considered is the inhomogeneous Gaussian image model, see, e.g., [CR13, CZF$^+$11, CZR$^+$12, Fil12]. In the case that the embedding rate is unknown, the approximated log LR statistic is obtained by employing Taylor expansions carried out on the log LR statistic in term of reciprocals of the local variances in the inhomogeneous Gaussian image model. This approximated log LR statistic is independent of the embedding rate which is the parameter of interest. For more details, the interested reader is referred to [CR13]. This approximation is, however, based on an assumption that local variances in the inhomogeneous Gaussian image model are greater than one, an assumption which might be considered unrealistic. For instance, local variances in smooth regions of a nature image are more likely to be smaller than one. This is also supported by our experiments on large image databases when local variances are obtained by commonly used estimators, see [CR13, CZR$^+$12, JW88]. To drop this assumption without sacrificing the performance of the detector, we will instead use the Rao's score statistic in this paper as the detector for the LSB matching. The score statistic used in the hypothesis problem stated in the next section is essentially the linear approximation of the log LR statistic in terms of the embedding rate around

the point where the embedding rate is zero. Also, as will be shown in Section 3, the score test is not only an AMP test for any given embedding rate but also a LAUMP test.

3.2 Inhomogeneous Image Models

In this section, we introduce a cover image model, the LSB matching embedding operation, and then derive an inhomogeneous Gaussian mixture image model which can represent either a cover model or a stego model, depending on the value of the embedding rate. This inhomogenous Gaussian mixture image model helps us state the hypothesis testing problem in terms of the embedding rate in the next section.

The cover image model we consider here for steganalysis is the inhomogeneous Gaussian model which captures the nonstationarity of the first-order and second-order statistics of the image. The nonstationary mean basically exhibits the gross structure of the image and the nonstationary variance describes edge and elementary texture information of the image [KSSC85]. Most importantly, this model allows extraction of local feature information of the image. The model has been widely used in image estimation and restoration [JW88, KSSC85], and steganalysis [CR13, CZF$^+$11, CZR$^+$12, Fil12, ZCR$^+$11] . We also assume in this chapter that pixels in the image are independent in order to not complicate the model and to simplify the computation. Based on the assumptions above, a cover image can be represented by a realization of a collection of N independent but not necessarily identical Gaussian random variables $\boldsymbol{X} = (X_1, X_2, ..., X_N)$ with $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\mu} = (\mu_1, \mu_2, ..., \mu_N), \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_N^2 \end{pmatrix}.$$

To derive the inhomogenous Gaussian mixture image model, we start with a description of the LSB matching which is one of most commonly used embedding

schemes in the spatial domain. Suppose that the secret information is encrypted and can be represented by a sequence of binary bits with 0 and 1 being randomly distributed. Let $\theta \in [0,1]$ be the embedding rate and assume that the embedding operation is carried out uniformly throughout the cover image. This implies that the probability of the event that one bit is hidden in any given pixel is θ. Given a pixel with the integer value w and one bit b being inserted at this pixel, the LSB matching embedding operation can be described as follows. If the LSB of w is the same as b, then w does not change. Otherwise, w is either increased or decreased by one, each with equal probability 1/2. Note that special care should be taken when w is at boundary values. But, we may reasonably assume that the effect is negligible and simply do not take it into consideration. In this view, the stego image can be regarded as a realization of a collection of N independent but not necessarily identical random variables $\boldsymbol{Y} = (Y_1, Y_2, ..., Y_N)$, satisfying

$$P\{Y_i = x_i+1|X_i = x_i\} = P\{Y_i = x_i-1|X_i = x_i\} = \frac{\theta}{4}, \quad P\{Y_i = x_i|X_i = x_i\} = 1-\frac{\theta}{2}. \tag{3.1}$$

Each random variable $Y_i, i = 1, 2, ..., N$, has a Gaussian-Mixture distribution. More specifically, let

$$f_{\mu_i,\sigma_i}(x) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{\frac{(x-\mu_i)^2}{2\sigma_i^2}\right\} \tag{3.2}$$

be probability density function of the i-th Gaussian random variable X_i, for $i = 1, 2, ..., N$. It is easily seen from (1) that the corresponding random variable Y_i has the probability density function

$$g_{\mu_i,\sigma_i}(x;\theta) = \frac{\theta}{4}\big[f_{\mu_i,\sigma_i}(x+1) + f_{\mu_i,\sigma_i}(x-1)\big] + \big(1-\frac{\theta}{2}\big) f_{\mu_i,\sigma_i}(x), \quad i = 1, 2, ..., N. \tag{3.3}$$

Note that $g_{\mu_i,\sigma_i}(x;0)$ is consistent with $f_{\mu_i,\sigma_i}(x)$. So an image, either a cover or a stego image, can be modeled by a collection of N independent random variables $\boldsymbol{Y} = (Y_1, Y_2, ..., Y_N)$ with the density function for each Y_i being given by (3.3). For notational convenience we simply denote by $g(\boldsymbol{x};\theta)$ the joint density function of $\boldsymbol{Y}$ with $\boldsymbol{x} = (x_1, x_2, ..., x_N)$ being a N-dimensional variable.

3.3 Score Tests

Having obtained the inhomogenous Gaussian mixture model, the steganalysis of LSB matching can be characterized by a hypothesis testing problem in terms of the embedding rate θ:

$$\mathcal{H}_0: \quad \theta = 0 \quad \text{against} \quad \mathcal{H}_1: \quad 0 < \theta \leq 1, \tag{3.4}$$

where $\mathcal{H}_0$ is called the null hypothesis and $\mathcal{H}_1$ is called the alternative hypothesis. Here, it is implicitly assumed that the embedding rate θ is unknown, which is a more realistic scenario for steganalysis.

We assume in the present section that the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are known. The only unknown is the embedding rate θ, the parameter of interest. Note that, for testing a simple null hypothesis against a simple alternative, one can always use the LR test which is the most powerful (MP) test by the Neyman-Pearson Lemma. But this is not the case for the hypothesis testing problem (3.4) which is the focus of this paper, because the alternative $\mathcal{H}_1$ is a composite hypothesis. The most powerful test is in general hard to be obtained for finite sample sizes, and the uniform most powerful (UMP) test may not even exist [LR06].

For this reason, we now turn our attention to seeking asymptotically optimal tests. We shall show that under mild assumptions, the score test exhibits asymptotic optimality properties. To describe the score test, we first define the log-likelihood function

$$L_N(\theta) = \sum_{i=1}^{N} \log g_{\mu_i,\sigma_i}(Y_i;\theta). \tag{3.5}$$

Let P_θ be the probability with the density function $g(\mathbf{x};\theta)$ and $E_\theta(\cdot)$ be the expectation with respect to $g(\mathbf{x};\theta)$. We make the assumptions used in [RM97]. In particular, it is assumed that the second moment of

$$Z_N(\theta) = \frac{1}{\sqrt{N}} \frac{\partial}{\partial\theta} L_N(\theta) \tag{3.6}$$

possesses an asymptotic expansion of the form

$$E_\theta\left[Z_N(\theta)^2\right] = I(\theta) + O_{P_\theta}\left(\frac{1}{N}\right), \tag{3.7}$$

where $I(\theta)$ is independent of N. For the hypothesis testing problem (3.4), the power against any fixed $\theta > 0$ tends to 1 as $N \to \infty$,as mentioned in [LR06]. Therefore, we shall consider sequences of alternatives of the form

$$\theta_{N,\epsilon} = \frac{\epsilon}{\sqrt{N}}, \quad \epsilon > 0, \tag{3.8}$$

for which the limiting power is nondegenerate, i.e., strictly between the level of significance α and 1. The asymptotic optimality of tests is most naturally investigated in terms of these alternatives.

Let $\Phi(\cdot)$ be the cumulative distribution function of the standard normal distribution. Note that a random test can be completely characterized by a critical function ϕ, with $0 \leq \phi(\cdot) \leq 1$. The Rao's score test $\hat{\phi}_N$ for testing $\theta = 0$ against $\theta_{N,\epsilon}$ at the level of significance α is therefore given by

$$\hat{\phi}_N = \begin{cases} 1, & \text{if } S_N \geq \tau_{1-\alpha} \\ 0, & \text{otherwise,} \end{cases} \tag{3.9}$$

where S_N is the studentized score statistic defined by

$$S_N = \frac{Z_N(0)}{\sqrt{I(0)}} \tag{3.10}$$

and $\tau_{1-\alpha} = \Phi^{-1}(1-\alpha)$ is the $1-\alpha$ quantile of the standard normal distribution $\mathcal{N}(0,1)$. It follows from (3.3), (3.5), (3.6) and (3.10) that

$$S_N = \frac{1}{\sqrt{NI(0)}} \sum_{i=1}^{N} \left\{ \frac{f_{\mu_i,\sigma_i}(Y_i+1) + f_{\mu_i,\sigma_i}(Y_i-1)}{4 f_{\mu_i,\sigma_i}(Y_i)} - \frac{1}{2} \right\} \tag{3.11}$$

Next, we derive the asymptotic properties of the score test $\hat{\phi}_N$. A direct calculation yields

$$E_0 \left[\frac{\partial}{\partial \theta} g(\mathbf{x};\theta)\big|_{\theta=0} \right] = 0. \tag{3.12}$$

It follows from (3.7) and (3.12) that, under P_0,

$$S_N \xrightarrow{d} \mathcal{N}(0,1), \tag{3.13}$$

where $\xrightarrow{d}$ denotes *convergence in distribution*. We define the log-likelihood ratio

$$LR_{N,\epsilon} = L_N(\theta_{N,\epsilon}) - L_N(0). \tag{3.14}$$

Recalling the Taylor's expansion of $LR_{N,\epsilon}$, we obtain

$$LR_{N,\epsilon} = \epsilon\sqrt{I(0)}S_N - \frac{1}{2}\epsilon^2 I(0) + o_{P_0}(1). \tag{3.15}$$

Thus, under P_0,

$$LR_{N,\epsilon} \xrightarrow{d} \mathcal{N}\left(-\frac{1}{2}\epsilon^2 I(0), \epsilon^2 I(0)\right), \tag{3.16}$$

which means $P_{\theta_{N,\epsilon}}$ is contiguous to P_0 in the sense that if $P_0(F_N) \to 0$ then $P_{\theta_{N,\epsilon}}(F_N) \to 0$ for every measurable set F_N, see Corollary 12.3.1 in [LR06]. We now derive the distributions of S_N and $LR_{N,\epsilon}$ under $P_{\theta_{N,\epsilon}}$ by applying Le Cam's Third Lemma, see Corollary 12.3.2 in [LR06]. Note that the joint behavior of S_N with the log-likelihood ratio $LR_{N,\epsilon}$ satisfies

$$(S_N, LR_{N,\epsilon}) = (S_N, \epsilon\sqrt{I(0)}S_N) + \left(0, -\frac{1}{2}\epsilon^2 I(0)\right) + o_{P_0}(1).$$

It then follows from the bivariate Central Limit Theorem that the joint expression above converges under P_0 to a bivariate normal distribution denoted by (S, LR_ϵ). That is, under P_0,

$$(S_N, LR_{N,\epsilon}) \xrightarrow{d} (S, LR_\epsilon)$$

with covariance

$$Cov_0(S, LR_\epsilon) = Cov_0(S, \epsilon\sqrt{I(0)}S) = \epsilon\sqrt{I(0)}E_0[S^2] = \epsilon\sqrt{I(0)}.$$

Hence, under $P_{\theta_{N,\epsilon}}$,

$$S_N \xrightarrow{d} \mathcal{N}(\epsilon\sqrt{I(0)}, 1). \tag{3.17}$$

It is easily seen from (3.16) that, under P_0,

$$Cov_0(LR_\epsilon, LR_\epsilon) = \epsilon^2 I(0).$$

By a similar argument, we have, under $P_{\theta_{N,\epsilon}}$,

$$LR_{N,\epsilon} \xrightarrow{d} \mathcal{N}\left(\frac{1}{2}\epsilon^2 I(0), \epsilon^2 I(0)\right). \tag{3.18}$$

There is another way to verify (3.18). Since $P_{\theta_{N,\epsilon}}$ is contiguous to P_0, we obtain from (3.15) that

$$LR_{N,\epsilon} = \epsilon\sqrt{I(0)}S_N - \frac{1}{2}\epsilon^2 I(0) + o_{P_{\theta_{N,\epsilon}}}(1),$$

which combines (3.17) to confirm (3.18) by employing the Slutsky's Theorem.

We now show the optimality property for the score test. For this purpose, we first recall the definition of the AMP test in [LR06].

Definition 3.3.1 *For testing $\theta = 0$ against $\theta = \theta_N$, $\{\phi_N\}$ is AMP at (asymptotic) level α if $\limsup_N E_0[\phi_N] \leq \alpha$ and if for any other sequence of test $\{\psi_N\}$ satisfying $\limsup_N E_0[\psi_N] \leq \alpha$,*

$$\limsup_N \{E_{\theta_N}[\psi_N] - E_{\theta_N}[\phi_N]\} \leq 0. \tag{3.19}$$

It can be easily verified that the score test $\hat{\phi}_N$ defined by (3.9) is AMP for testing $\theta = 0$ against $\theta_{N,\epsilon}$ at level α. To see this, we define the log-likelihood test by

$$\hat{\psi}_{N,\epsilon} = \begin{cases} 1, & \text{if } LR_{N,\epsilon} \geq -\epsilon^2 I(0) + \epsilon\sqrt{I(0)}\tau_{1-\alpha} \\ 0, & \text{otherwise.} \end{cases} \tag{3.20}$$

It follows from (3.13) and (3.17) that the power of the score test $\hat{\phi}_N$ satisfies

$$E_{\theta_{N,\epsilon}}\left[\hat{\phi}_N\right] \to 1 - \Phi\left(\tau_{1-\alpha} - \epsilon\sqrt{I(0)}\right). \tag{3.21}$$

Similarly, we obtain from (3.16) and (3.18) that the power of the log-likelihood test $\hat{\psi}_{N,\epsilon}$ satisfies

$$E_{\theta_{N,\epsilon}}\left[\hat{\psi}_{N,\epsilon}\right] \to 1 - \Phi\left(\tau_{1-\alpha} - \epsilon\sqrt{I(0)}\right). \tag{3.22}$$

By the Neyman-Pearson Lemma, the test $\hat{\psi}_{N,\epsilon}$ is MP for testing $\theta = 0$ against $\theta = \theta_{N,\epsilon}$ at level α. We conclude from (3.21) and (3.22) that the score test $\hat{\phi}_N$ is AMP. To investigate the (locally) uniform optimality of the score test, we recall the definition of LAUMP provided in [LR06].

Definition 3.3.2 *For testing $\theta = 0$ against $\theta > 0$, a sequence of tests $\{\phi_N\}$ is called LAUMP at (asymptotic) level α if $\limsup_N E_0[\phi_N] \leq \alpha$ and if for any other sequence of test $\{\psi_N\}$ satisfying $\limsup_N E_0[\psi_N] \leq \alpha$,*

$$\limsup_N \sup\left\{E_\theta[\psi_N] - E_\theta[\phi_N], ; 0 < \sqrt{N}\theta \leq c\right\} \leq 0 \tag{3.23}$$

for any $c > 0$.

To show the score test $\hat{\phi}_N$ is LAUMP, we only need to prove, for any $c > 0$,

$$\sup_{0\le\epsilon\le c}\left|E_{\theta_{N,\epsilon}}\left[\hat{\phi}_N\right] - \left[1-\Phi\left(\tau_{1-\alpha}-\epsilon\sqrt{I(0)}\right)\right]\right| \to 0, \tag{3.24}$$

which can be done by following a similar argument in [LR06]. That is, if (3.24) does not hold, there exists a sequence $\epsilon_i \to \hat{\epsilon} \in [0, c]$ such that

$$E_{\theta_{N,\epsilon_j}}\left[\hat{\phi}_N\right] - \left[1-\Phi\left(\tau_{1-\alpha}-\hat{\epsilon}\sqrt{I(0)}\right)\right] \to \gamma \neq 0,$$

which contradicts (3.21). We summarize the results as follows.

Theorem 3.3.1 *Consider testing $\theta = 0$ against $\theta > 0$ at level α. Let $\hat{\phi}_N$ be the test defined by* (3.9)*. The power of $\hat{\phi}_N$ satisfies*

$$E_\theta\left[\hat{\phi}_N\right] \to 1-\Phi\left(\tau_{1-\alpha}-\theta\sqrt{N\cdot I(0)}\right). \tag{3.25}$$

Also, $\hat{\phi}_N$ is not only AMP, but LAUMP.

It is easily seen from (3.25) that the condition for which the optimal limiting power against the alternative is nondegenerate is $\theta\sqrt{N} = O(1)$, as $N \to \infty$.

3.4 Practical Considerations

In the previous section, the local means μ_i and variances σ_i in the inhomogeneous image model are assumed to be known. In practice, we have to estimate them from images in order to design statistical tests. Also, the quantity $I(0)$ used in the score statistic defined by (3.10) might be hard to obtain analytically. And once it is, its estimation has to be addressed.

One way to estimate both local means and local variances is by using a weighted sample average over a rectangular window of size $(2U+1)\times(2V+1)$. To better describe the estimation, we rewrite the local means and variances in the two-dimensional form, namely, $\mu_{i,j}$ and $\sigma^2_{i,j}$, where (i, j) corresponds to the pixel position. Similarly, we use $Y_{i,j}$ for the random variable corresponding to the position (i, j) in the inhomogeneous

image model described in the previous section. A formula for estimating the local means is

$$\hat{\mu}_{i,j} = \sum_{k=-U}^{U} \sum_{l=-V}^{V} w_{k,l} x_{i+k,j+l}, \tag{3.26}$$

where $x_{i,j}$ is the pixel value at the position (i, j) and the sum of all weight coefficients $w_{k,l}$ is one. Similarly, a formula for estimating the local variance is given by

$$\hat{\sigma}^2_{i,j} = \sum_{k=-U}^{U} \sum_{l=-V}^{V} s_{k,l} (x_{i+k,j+l} - \hat{\mu}_{i,j})^2. \tag{3.27}$$

To avoid the numerical instability, we place a lower bound σ^2_{thres} on the estimates of the local variances. Thus, we use in practice

$$\tilde{\sigma}^2_{i,j} := \max\{\sigma^2_{thres}, \hat{\sigma}^2_{i,j}\}.$$

The estimation schemes (4.1) and (4.2) were employed in image estimation and restoration, see [JW88,KSSC85], where uniform weights were considered. The estimation schemes were also commonly used in steganalysis, see, e.g., [CR13,CZF$^+$11,KB08] and references cited therein. Note that the implicit assumption on the schemes for estimating local means and variances is that an image is locally ergodic.

As pointed out earlier, the quantity $I(0)$ in the score statistic needs to be estimated if it cannot be analytically calculated. In this case, we replace $I(0)$ with I_N in practice which is defined by

$$I_N = \frac{1}{N} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \gamma^2_{\hat{\mu}_{i,j},\tilde{\sigma}_{i,j}}, \tag{3.28}$$

where $N_1 \times N_2 = N$ and

$$\gamma_{\hat{\mu}_{i,j},\tilde{\sigma}_{i,j}} = \frac{\partial}{\partial\theta} \log g_{\hat{\mu}_{i,j},\tilde{\sigma}_{i,j}}(Y_{i,j};\theta)\big|_{\theta=0} = \frac{f_{\hat{\mu}_{i,j},\tilde{\sigma}_{i,j}}(Y_{i,j}+1) + f_{\hat{\mu}_{i,j},\tilde{\sigma}_{i,j}}(Y_{i,j}-1)}{4 f_{\hat{\mu}_{i,j},\tilde{\sigma}_{i,j}}(Y_{i,j})} - \frac{1}{2}.$$

Inserting I_N, the estimates of local means and variances, into (3.11), we obtain the studentized score statistic used in practice

$$\hat{S}_N = \frac{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \gamma_{\hat{\mu}_{i,j},\tilde{\sigma}_{i,j}}}{\sqrt{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \gamma^2_{\hat{\mu}_{i,j},\tilde{\sigma}_{i,j}}}} \tag{3.29}$$

The corresponding score test is the same as (3.10) except that S_N is replaced with $\hat{S}_N$.

3.5 Experiments

In this section, we report performance of the Rao's score test and the LR test proposed in [CR13]. The estimation of local means and variances described in the previous section is applied in the experiments. All experiments in this section are carried out on the image databases UCID [SS03] and the BOSSbase image database with the version 0.92 [BFP11]. The original UCID database contains $1,338$ uncompressed color images with the size of 384×512 or 512×384 pixels. These uncompressed color images are converted to grayscale images in our experiments by using Matlab function *rgb2gray*. The BOSSbase database contains $9,074$ processed grayscale images with the size of 512×512 pixels. All stego images in the experiments are generated from the two image databases by using the LSB matching embedding scheme. Three different embedding rates are considered in the experiments which are $0.25,\ 0.5$ and 1.

We estimate the local means and variances over a window of size 3×3, i.e., $U = V = 1$ in (4.1) and (4.2). The estimation of the local means is obtained by applying a low-pass filter

$$\frac{1}{4}\begin{pmatrix} -1 & 2 & -1 \\ 2 & 0 & 2 \\ -1 & 2 & -1 \end{pmatrix},$$

which corresponds to using the estimation scheme (4.1) with weights

$$w_{0,0} = 0, \quad w_{-1,1} = w_{-1,-1} = w_{1,-1} = w_{1,1} = -\frac{1}{4}, \quad w_{-1,0} = w_{0,-1} = w_{0,1} = w_{1,0} = \frac{1}{2}.$$

This formula has been widely used in local mean estimation (see, e.g., [CR13,CZR$^+$12, KB08]). It was also observed in our experiments that both the score test and the LR test using this local mean estimation basically provide better performance on both the UCID and the BOSSbase image databases than those using other low-pass filters such as box filters, Gaussian filters, etc., when the same local variance estimation and

the same variance threshold σ^2_{thres} are applied. So we only report the experimental results based on this local mean estimation.

The estimation of the local variances in all of our experiments is based on the formula (4.2) with a lower bound threshold $\sigma^2_{thres} = 0.3$. For the proposed score test and the LR test, we use uniform weights for the local variance estimation, namely,

$$s_{i,j} = \frac{1}{9}, \qquad i, j = -1, 0, 1. \tag{3.30}$$

In addition, we report the performance of the LR test using the local variance estimation with weights (see, [CR13])

$$s_{-1,0} = s_{0,-1} = s_{0,1} = s_{1,0} = \frac{1}{3}. \tag{3.31}$$

The performance of the proposed score test and the LR test described in [CR13] is measured by receiver operating characteristic (ROC) curves. In the following, the statistical tests using local variance estimation with uniform weights (3.32) are referred to as *9-point* tests, and the tests using local variance estimation with weights (3.33) are referred to as *4-point* tests.

Figure 3.1 exhibits a comparison of performance of the score test and the LR test on both the UCID and the BOSSbase image databases with the embedding rate is 0.25. It is clear that the score test outperforms the LR test which uses either (3.32) or (3.33) for the local variance estimation.

It is presented in Figure 3.2 that the score test performs better than the score test on the UCID image database, especially when the false alarm rate is less than 0.5. The LR test using the local variance estimation (3.32) performs equally well as the score test when the false alarm rate is greater than 0.5. The case is different when they perform on the BOSSbass image database. As shown in Figure 5.2, the score test clearly outperforms the LR test which employs the local variance estimation (3.32). It performs much better than the LR test using the local variance estimation (3.33) when the false alarm rate is less than 0.3, but the latter performs slightly better than the former when the false alarm rate is greater than 0.3.

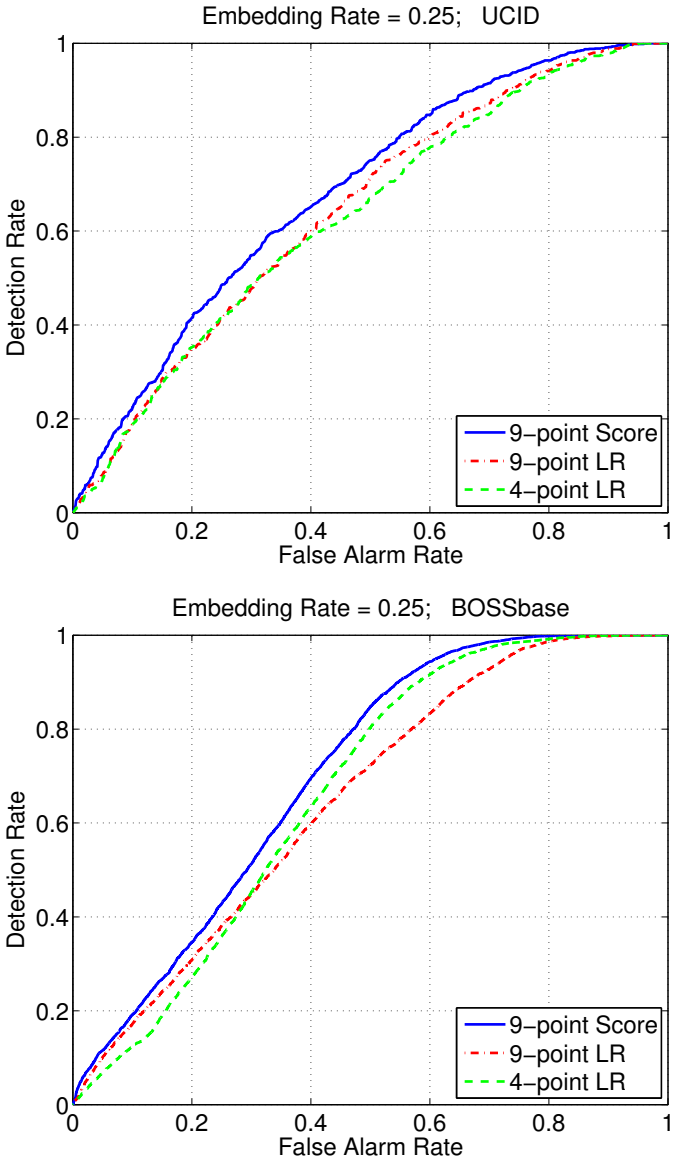


Figure 3.1. Comparison of ROCs with embedding rate $\theta = 0.25$; Top: performance on the UCID database; Bottom: performance on the BOSSbase database

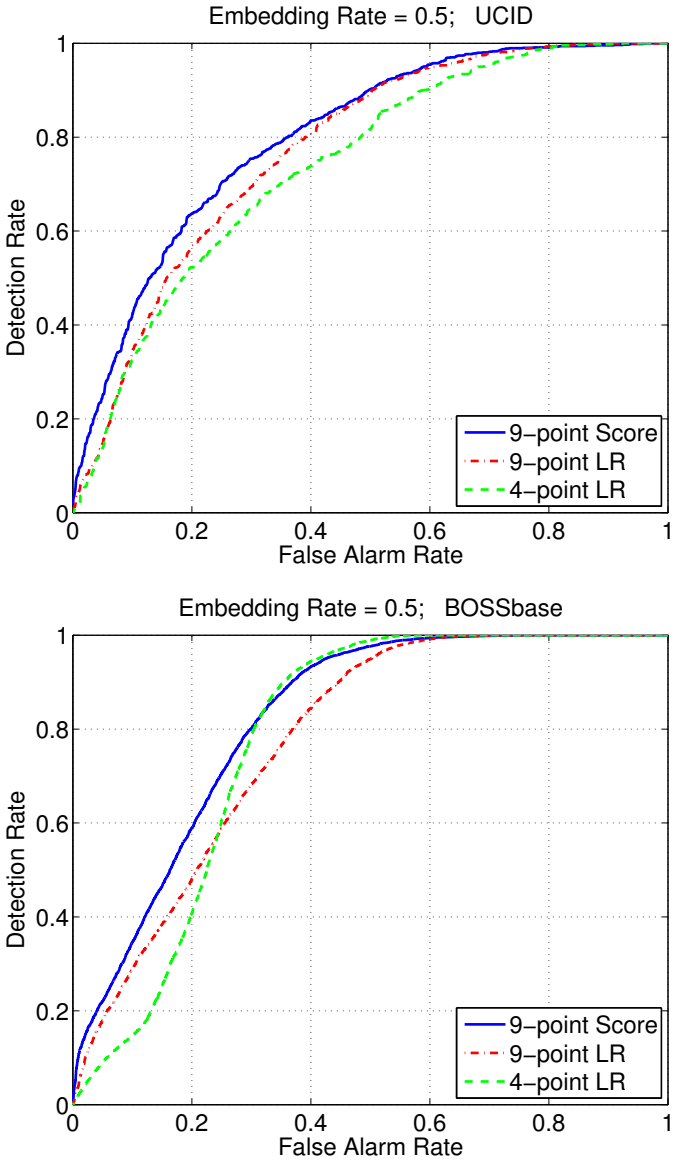


Figure 3.2. Comparison of ROCs with embedding rate $\theta = 0.5$; Top: performance on the UCID database; Bottom: performance on the BOSSbase database

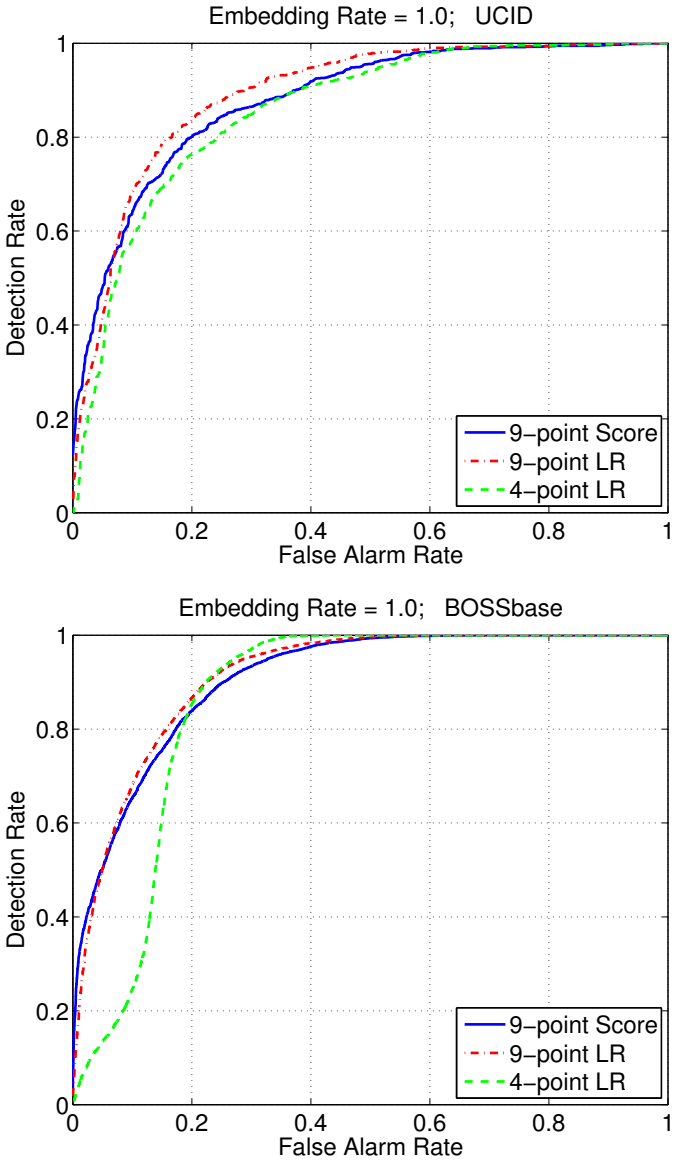


Figure 3.3. Comparison of ROCs with embedding rate $\theta = 1.0$; Top: performance on the UCID database; Bottom: performance on the BOSSbase database

Finally, the score test and LR test perform quite different on both the UCID and the BOSS image databases when the embedding rate is one, as shown in Figure 3.3. Basically, the LR test with the local variance estimation (3.32) slightly outperforms the score test when the false alarm rate is greater than 0.1 on both of the image databases, but the latter outperforms the former when the false alarm rate is less than 0.1. The score test performs better than the LR test with the local variance estimation (3.33) on the UCID image database. The score test is outperformed by the the LR test on the BOSS image database when the false alarm rate is greater than 0.2, but the former performs much better than the latter when the false alarm rate is less than 0.2.

3.6 JEPG DOMAIN

In the previous sections, we described hypothesis testing for image steganalysis on spatial domain. The values involved in statistical models and tests are pixel values of spatial images. For completeness, I will briefly discuss steganalysis of JEPG images using hypothesis testing in the final section of this chapter. JEPG images are widely employed for information hiding since they are everywhere on Internet. Note that quantized DCT coefficients are used to embed secret binary string and therefore statistical models for quantize DCT coefficients are needed if the hypothesis testing framework is applied. The derivation of the models involve a few steps and it is not as straightforward as those used for pixel values of spatial images from my perspective.

3.6.1 Statistical Models

The JEPG compression involves the DCT followed by the quantization. Let us start with a description of the DCT used in JEPG which is perform within each of 8×8 blocks [PM92]

$$I_{m,n} = \frac{1}{4} T_m T_n \sum_{i=0}^{7} \sum_{j=0}^{7} x_{i,j} \cos\left(\frac{(2i+1)m\pi}{16}\right) \cos\left(\frac{(2j+1)n\pi}{16}\right), \tag{3.32}$$

where $x_{i,j},\ i,j = 0,1,...,7$, denote pixels within an 8×8 block for $i,j = 0,1,...,7$, and $I_{m,n},\ m,n = 0,1,...,7$, denote the corresponding DCT coefficients and T_m is defined by

$$T_m = \begin{cases} 1/\sqrt{2}, & m = 0 \\ 1, & m > 0. \end{cases} \tag{3.33}$$

T_n has the same expression as T_m by replacing n with m. The coefficient $I_{0,0}$ is called the Direct Current (DC) coefficient and represents the mean value of pixels in the 8×8 block. The other 63 coefficients are called the Alternating Current (AC) coefficients. We assume that $x_{i,j},\ i,j = 0,1,...,7$, are identically distributed but not necessarily Gaussian Applying the central limit theorem (CLT), we have that the weighted summation $I_{m,n}$ is approximately distributed as a (zero-mean) Gaussian for given m and n. Note that the CLT is valid even when $x_{i,j}$ are spatially correlated as long as the correlation is not strong. Typically, the variance of the blocks varies, meaning $\sigma^2_{m,n}$ itself can be regarded as a random variable which is simply denoted by σ^2. We can thereby use the doubly stochastic model to compute the probability density function (pdf) of AC coefficients [LG00, TCR14]

$$f_I(x) = \int_0^\infty f_{I|\sigma^2}(x|s) g_{\sigma^2}(s) ds. \tag{3.34}$$

As mentioned above, $f_{I|\sigma^2}(x|s)$ is approximated by a zero-mean Gaussian for given σ^2 and can be written as

$$f_{I|\sigma^2}(x|s) = \frac{1}{\sqrt{2\pi s}} \exp\left(-\frac{x^2}{2s}\right), \tag{3.35}$$

and $g_{\sigma^2}(s)$ is the distribution of the random variable σ^2. There are many options for $g_{\sigma^2}(s)$, see, e.g., [LG00, TCR14], resulting in different pdfs of AC coefficients. One of the options is the Gamma distribution, i.e.,

$$g_{\sigma^2}(s) = \frac{s^{\eta-1}}{\nu^\eta \Gamma(\eta)} \exp\left(-\frac{s}{\nu}\right), \tag{3.36}$$

where $\eta > 0$ is a shape parameter, $\nu > 0$ is a scale parameter, and $\Gamma(\cdot)$ is the gamma function. By this choice, it turns out that the pdf $f_I(x)$ in (3.34) can be written as the following expression [GS01, SLG02, TCR14]

$$f_I(x) = \sqrt{\frac{2}{\pi}} \frac{\left(|x|\sqrt{\frac{\nu}{2}}\right)^{\eta-\frac{1}{2}}}{\nu^{\eta}\Gamma(\eta)} K_{\eta-\frac{1}{2}}\left(|x|\sqrt{\frac{2}{\nu}}\right), \tag{3.37}$$

where $K_\eta(\cdot)$ is the modified Bessel. Another common choice of $g_{\sigma^2}(s)$ is the generalize Gaussian distribution

$$g_{\sigma^2}(s) = \frac{\eta}{2\nu\Gamma(1/\eta)} \exp\left(-\left(\frac{|s|}{\nu}\right)^{\eta}\right), \tag{3.38}$$

where $\nu > 0$ and $\eta > 0$ are the scale and shape parameters respectively.

Next, we introduce the quantization on the AC coefficients and then obtain the quantized AC coefficients. Let $P_V(k)$ be the probability mass function (pmf) of the quantized AC coefficient V using the quantization step Δ. Note that V is a discrete random variable. The pmf $P_V(k)$ is defined by

$$P_V(k) = \int_{\Delta(k-1/2)}^{\Delta(k+1/2)} f_I(x)dx, \tag{3.39}$$

where $f_I(x)$ is calculated from (3.34). Therefore, a statistical model for quantized AC coefficients is obtained once we choose $g_{\sigma^2}(\cdot)$ and the quantization step Δ.

3.6.2 Steganalysis for Jsteg Algorithm

The Jsteg algorithm is the LSB replacement embedding scheme on the DCT domain, see, e.g., [ZP03, YWT04, KF10, TCR14]. Steganalysis using hypothesis testing for the Jsteg algorithm is similar to that for the LSB matching described in previous sections except that statistical models are different.

A cover JPEG image can be represented by 64 vectors of quantized DCT coefficients $C_i, i = 1, 2, ..., 64$. We denote by P_{λ_i,Δ_i} the pmf with the parameter vector λ_i and the quantization step Δ_i. Notice that $\lambda_i = (\nu_i, \eta_i)$ for the previous models mentioned above. Note that the Jsteg algorithm does not choose the DC coefficient

for embedding for the security reason. Also, it does not choose the quantized AC coefficients with the value 0 or 1. Those are constraints on the Jsteg algorithm. Let θ be the embedding rate and $Q_{\theta,\lambda_i,\Delta_i}$ be the pmf of the resulting stego JPEG image which can be written as [ZCR+11, TCR14]

$$Q_{\theta,\lambda_i,\Delta_i}(k) = \left(1 - \frac{\theta}{2}\right) P_{\lambda_i,\Delta_i}(k) + \frac{\theta}{2} P_{\lambda_i,\Delta_i}(\bar{k}), \quad k \neq 0, 1, \tag{3.40}$$

where $\bar{k}$ denotes the integer k with the LSB flipping operation $\bar{k} = k + (-1)^k$ [DSM+04]. In addition,

$$Q_{\theta,\lambda_i,\Delta_i}(0) = P_{\lambda_i,\Delta_i}(0), \tag{3.41}$$

and

$$Q_{\theta,\lambda_i,\Delta_i}(1) = P_{\lambda_i,\Delta_i}(1) \tag{3.42}$$

Note that $Q_{0,\lambda_i,\Delta_i}(\cdot) = P_{\lambda_i,\Delta_i}$. The hypothesis testing for steganalysis of the Jsteg algorithm is established in the same way as (3.4). The Rao's score test is obtained by following the steps similar to those from (3.5) to (3.10). In addition, nuisance parameters λ_i's are unknown and need to be estimated in practice. The most commonly used methods include the method of moments (MM) estimates and maximum likelihood (ML) estimates, see, e.g., [Mül93, CSKM05, TCR14]. It is worth noting that there are up to 63 different distributions (instead of a single distribution) for quantized AC coefficients and so the log-likelihood function in (3.5) should be similar to that in [BG62] where observations did not come from a single population but from distinct but related populations.

Figure 3.4 exhibits performance of the Rao's score test on the BOSSbase JPEG75 which is generated from the original BOSSbase dataset via JPEG compression with a factor of 75. Here we used the Gammar distribution and the generalized Gaussian distribution, respectively, for g_{σ^2}. It can be observed that the Rao's score test with both of the distributions work well for the small embedding rate. As mentioned above, the Jsteg embedding scheme is the LSB replacement scheme on DCT domain which shows some patterns that can be exploited by steganalysis. We also extend the score test to the symmetric Jsteg [KF10] embedding scheme and the result is shown in 3.5.

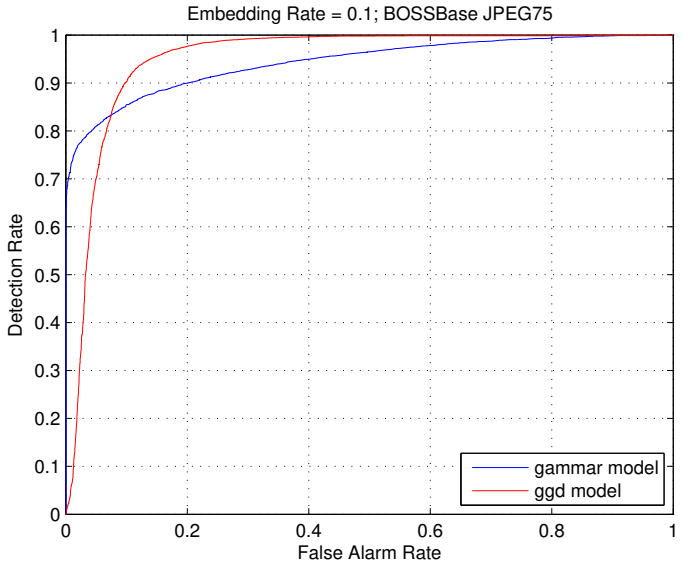


Figure 3.4. Comparison of ROCs for two models with embedding rate $\theta = 0.1$ on BOSSbase JPEG75; Embedding method: JSteg

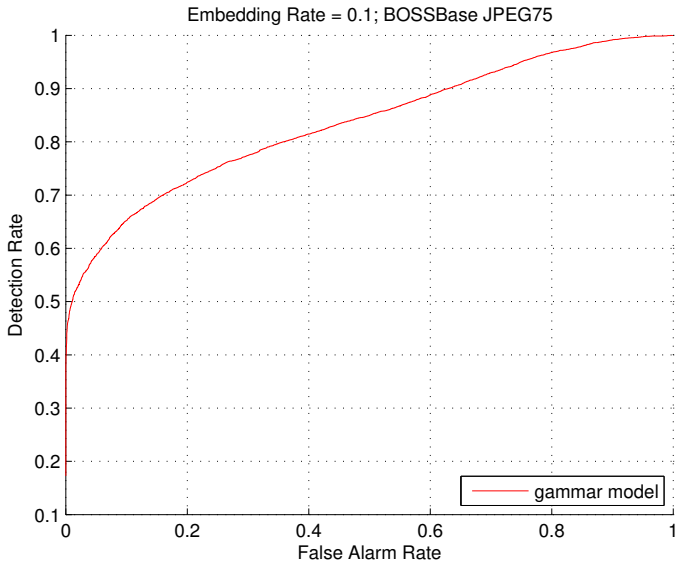


Figure 3.5. ROC for the gammar model with embedding rate $\theta = 0.1$ on BOSSbase JPEG75; Embedding method: Symmetric JSteg

4 NEURAL NETWORKS FOR STEGANALYSIS

4.1 Introduction

The image steganalysis using the hypothesis testing framework has been discussed in the previous chapter. This hypothesis testing framework for steganalysis heavily relies on cover image models (e.g., inhomogeneous Gaussian distribution) and the embedding scheme (e.g., LSB matching with the random embedding). The pixel correlation or dependence is basically ignored in order to simplify the models. Also, the content adaptive embedding scheme is much harder to be integrated with the cover models than the random embedding scheme. To develop effective detectors for complex steganographic methods, such as HUGO, WOW and S-UIWARD, people resort to machine learning techniques. The machine learning methods for steganalysis have shown very good performance, see, e.g., [SCC06, PF07, CS08, PBF10, PFB10, KFH12] and the references therein. Since the rich model, which includes more than 30,000 features, was proposed [FK12], the approach which combines the rich model with the ensemble learning has became the state-of-the-art detector for steganalysis. Still, performance of the machine learning based detector is greatly impacted by the quality of handcrafted features. Finding the *useful* handcrafted features is a challenge task and needs domain expertise and a great deal of experiments using all kinds of machine learning methods. After completeness of the rich model, it seems new approaches are needed which go beyond the extraction of handcrafted features.

With the successful applications of deep neural networks in compute vision, natural language processing and other tasks, people paid much attention and considered applying neural networks to their own tasks. One of main advantages of using deep neural networks is to self-learn/extract features from data during the training process. This is a totally different approach for featrue extraction. Also, deep neural

networks have very large capacity because they can easily have of thousands of millions) of parameters. Even a shallow neural network may have hundreds of thousands of parameters. Such a huge capacity allows neural networks to self-learn complex structures/features from data and achieve outstanding performance on many tasks. Typically, neural networks combine feature extraction and classification and provide a end-to-end solution for steganalysis. Most of layers in neural network architectures are typically used for feature extraction and the last layer is employed for classification. Challenges of using deep neural networks include data collection, design of neural network architectures, computations, etc. Progresses have been greatly made by researchers across different disciplines to address those challenges.

For image steganalysis, the deep learning framework provides a new approach going beyond handcrafted features. More importantly, we may be motivated by the progress of deep learning for other tasks, especially for computer vision. In 2014 Tan *et al.* [TL14] made an attempt to apply stacked auto-encoders for steganalysis though the results did not look good. In 2015, Qian *et al.* [QDWT15] proposed a convolutional neural network (CNN) architecture which showed promising results. This is regarded as an early influential effort using deep learning for image staganalysis. Their design of the CNN architecture was inspired by those applied in computer vision. The only exception is that a fixed high-pass filter was employed by Qian *et al.* [QDWT15] in the beginning of the CNN architecture. The 5×5 high-pass filter is defined as follows.

$$F^{(0)} = \frac{1}{12} \begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix}. \tag{4.1}$$

This is a high-pass KV filter used in [KFH11] and is also used in [FK12] called the square S5a filter. Basically, this high-pass filter is regarded as simulating the feature extraction process. It was reported in [QDWT15] that this high-pass filter help improve the performance of the neural network. It is worth noting that

this filter is symmetric and the sum of all elements is zero. This module is used to extract stego information hidden in images. Notice that the secret information is embedded in LSBs of values of pixels or DCT coefficients to avoiding much distortion. Also, as mentioned before, content adaptive embedding schemes are very likely to hide secret information in regions of images that are not smooth, such as lines. For these reasons, the stego signals are widely believed to hide in noise level of images. Therefore, applying high-pass filters is likely to enhance the stego information, which is very helpful for image stanalysis. This high-pass filtering design has a great influence. Xu *et al.* [XWS16b] employed this high-pass filter, along with absolute value layer, batch normalization, average pooling and 1×1 convolutions to improve performance. Ye *et al.* [YNY17] used 30 high-pass filters in the first convolutional layer as initialization which was followed by an new activation function called the Thresholded Linear Unit (TLU). Yedroudj *et al.* [YCC18b, YCC18a] presented another architecture which takes advantage of the novel design in Xu-Net and Ye-Net, such as high-pass filtering, absolution value layer, TLU, etc. Recently, a CNN architecture without high-pass filtering design was proposed [BCF19]. It applied residual module [HZRS16a, HZRS16b], along with average pooling and 1 convolutions, in the architecture to boost performance. Other CNN architectures for image steganalysis can be found in [CCGS16, PPIC16, YSWK17, ZTLH18] and the references therein.

In order to better understand the CNN architecture, we need to introduce some terminologies. This introduction of concepts is brief and selective. A comprehensive review of deep neural networks can be found in a recent book [GBC16].

4.1.1 Layer

Figure 4.1 show a basic neural network architecture which consists of an input layer, two hidden layers and an output layer. Each layer has a bunch of neurons except the output layer (in this architecture). The neurons in adjacent layers are fully connected shown in 4.1. They are actually fully connected layers. In our case,

the input which is a 16-dimensional vector moves forward to pass the first hidden layer. A product of a matrix and a vector is computed and the output is a 12-dimensional vector. When the second hidden layer is passed, the computational process repeats and the output is a 10-dimensional vector. The concept can be generalized to the CNN architecture. In terms of CNN, each neuron in hidden layers is considered a filter with certain sizes (e.g., 3×3, 5×5). The hidden layers are actually called convolutional layers in this case. Each input unit is regarded as a channel of an image. For instance, a RGB image has three channels and a grayscale image has only one channel. When an input image pass through the first convolutional layer, the convolution is involved and the output is called a *feature map* which is a collection of 12 images in our case. This process repeats when the input moves forward layer by layer.

4.1.2 Convolution

We next use mathematical notations to describe the convolution. The notations are adopted from [QDWT15]. Let $I^{(0)}$ be the filtered image which is the output of passing through the fixed high pass filter (4.1). Let $F_k^{(l)}$ denote the k^{th} filter from layer $l = \{1, ..., L\}$, with L being the number of convolutional layers, and $k \in \{1, ..., K^{(l)}\}$, with $K^{(l)}$ being the number of filters of the l^{th} layer. A convolution from the first layer with the k^{th} filter results in a filtered image denoted by $\bar{I}_k^{(1)}$, such that

$$\bar{I}_k^{(1)} = I^{(0)} * F_k^{(l)} \tag{4.2}$$

From the second layer to the last convolutional layer, the convolution is less conventional since there are $K^{(l-1)}$ feature maps (i.e., $K^{(l-1)}$ images) as input, denoted by $I_k^{(l-1)}$ with $k = 1, ..., K^{(l-1)}$. Note that the convolution which will lead to the k^{th} filter images $\bar{I}_k^{(l)}$ resulting from the convolutional layer l, is actually the sum of $K^{(l-1)}$ convolution operations. That is,

$$\bar{I}_k^{(l)} = \sum_{i=1}^{K^{(l-1)}} I_i^{(l-1)} * F_{k,i}^{(l)}, \tag{4.3}$$

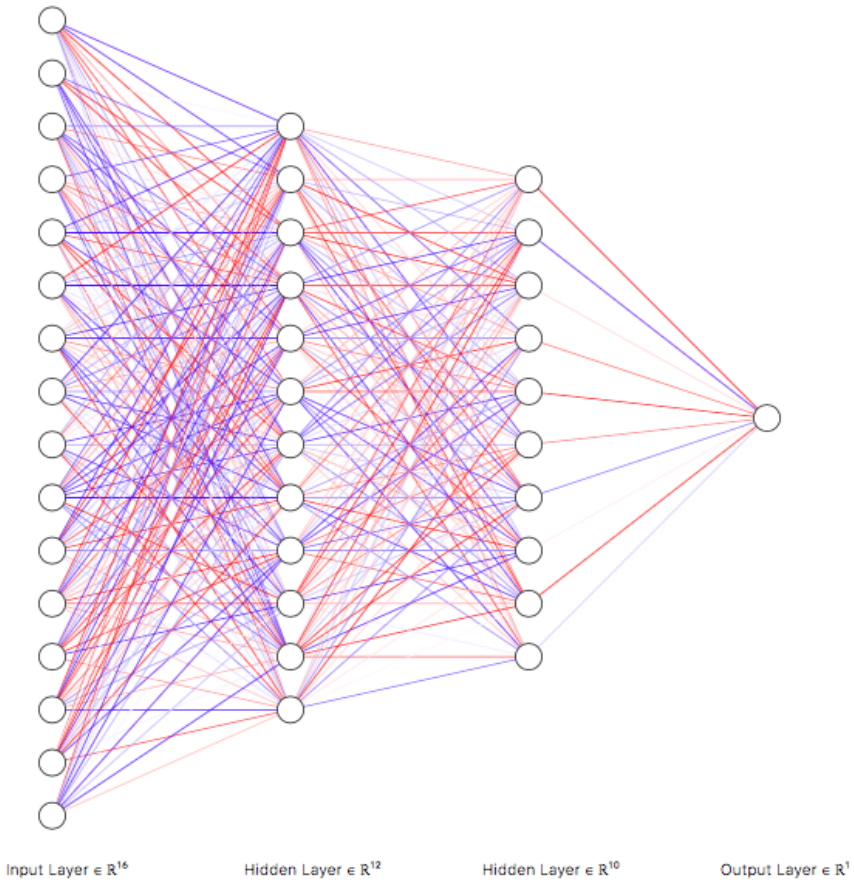


Figure 4.1. A simple neural network

where $F_{k,i}^{(l)}$ are $K^{(l-1)}$ filters for given k.

4.1.3 Activation

An activation function is a nonlinear function which introduce nonlinearity to networks. The sigmoid function was a popular activation function. In terms of CNN, the most commonly used activation function after the convolution operations is a rectified linear unit (ReLU), which is define as

$$ReLU(x) = max(0, x).$$

Another activation function TLU proposed by Ye *et al.* [YNY17] is defined by

$$TLU(x) = \begin{cases} -T, & x < -T \\ x, & -T \leq x \leq T \\ T, & x > T, \end{cases}$$

where T is a parameter. In addition, an absolute activation function used in [XWS16b] will is simply defined as

$$f(x) = |x|$$

Note that this absolute activation function has been used in Xu-Net and Yedroudj-Net, see [XWS16b, YCC18b].

4.1.4 Batch Normalization (BN)

The batch normalization [IS15] is a type of layer which aims at normalizing data adaptively. It normalizes the distribution of each feature to zero-mean and unit-variance and then scales and translates the distribution during training. The main advantage of using the batch normalization is to help maintain gradient propagation. It could speed up the learning process by using a larger learning rate, see [IS15] for more details. Note that two parameters: the shift parameter and the scale parameter are learned from training data.

4.1.5 Pooling

The pooling operation falls into two categories: the average pooling and the maximum pooling. For image recognition, the maximum pooling is preferable and has a local invariance in translation when the features are recalculated. However, the average pooling is most commonly used in steganalysis because the embedding information is considered hiding in image noise. The use of maximum pooling may result in the useful information which help classify cover and stego images. Often, pooling is used to reduce the size of the output feature maps by choosing an appropriate stride. For instance, a 2×2 average or maximum pooling stride 2 on an image would down sample the image by half.

Again, a more detailed description of convolutional networks can be found in [GBC16], Chapter 9.

4.2 Proposed Neural Networks

4.2.1 Related Work

The initial efforts using deep learning framework for image steganalysis were made by Tan *et al.* [TL14] and Qian *et al.* [QDWT15]. An important contribution in [QDWT15] is to introduce a fixed high pass-filter as a preprocessing module. Other researchers have used this preprocessing module when other CNN architectures were developed, see, e.g., [PPIC16, XWS16b, XWS16a, PPIC16, YCC18b]. Those results obtained from CNN architectures were encouraging. Ye-Net was proposed by using 30 high pass filters as initialization of 30 trainable filters in the first convolutional layer [YNY17]. Yedroudj *et al.* [YCC18b] also used these 30 high-pass filters in the first layer of their proposed CNN architecture but they fixed these high pass filters as the preprocessing module during the training. Note that Yedroudj *et al.* employed 30 fixed high pass filters whereas Qian *et al.* only used one. Recently, a deep residual network, called SRNet, was proposed [BCF19] which did not use any

high pass filters as a preprocessing module in the beginning of the CNN architecture. Instead, they use multiple convolutional layers to extract stego signals during the training process. It was reported that SRNet has achieved good performance the BOSSbase and BOWS2 datasets [BFP11, BF]. There are many other CNN architectures used for image steganalysis that are inspired by well-known architectures or modules used in computer vision, such as VGG [SZ14], ResNet [HZRS16a,HZRS16b], DensNet [HLVDMW17], etc. Among those CNN architectures for image steganalysis, Xu-Net [XWS16b, XWS16a] and Ye-Net [YNY17] as well as recently proposed Yedroudj-Net [YCC18b] are widely used for comparison.

4.2.2 Proposed CNN Architectures

Motivated by previous works mentioned above, I propose a CNN architecture which uses 30 trainable high-pass filters in the first convolutional layer in the beginning of the architecture. The idea behind this high-filtering design is that we want to place a *self-learning* high-filtering convolutinal layer in the beginning. The term self-learning means the kernels of this CNN layer are trainable, namely, kernel weights are adjusting during the training process. As mentioned before, Qian *et al.* [QDWT15] used one fixed high-pass filter (4.1) which is not trainable. Ye-Net [YNY17] employed a trainable CNN layer with the kernels of 30 high-pass filters as initial assignments. But the kernel weights are changing during training and unlikely to continue doing high-pass filtering.

In order to make the 30 filters function as high pass filters during the training process, we add some constraints on these filters. We make them have the symmetry and keep the sum of the kernel weights (for each kernel) to be zero which is a property of high pass filters. We hope the trainable kernels can self-learn from data during the training process. Again, the bottom line is to put some constraints on trainable kernels. Yedroudj-Net [YCC18b] fixed 30 high-pass filters in the beginning to conduct high-pass filtering.

It is worth pointing out that there are successful modules/layers in Xu-Net, Ye-Net, etc, such as the truncation layer, average pooling layer. Those layers I believe being helpful are introduced in our proposed CNN architectures. In addition, residual connections/modules [HZRS16a, HZRS16b] have been commonly employed in deep neural networks to help model training. They are also considered in the proposed architecture.

Figures 4.3 and 4.4 present the proposed CNN architecture which consists of two blocks shown in Figure 4.2. The architecture diagram of Xu-Net [YNY17] is presented in Figure 4.5, the architecture diagram of Ye-Net [YNY17] is exhibited in Figure 4.6 and the architecture diagram of Yedroudj-Net [YCC18b] is shown in Figure 4.7.

4.3 Experiments

4.3.1 Setup

We consider three content adaptive embedding schemes: WOW, MiPOD and S-UIWARD. We compare the performance of the rich model with the ensemble learning, Xu-Net, Ye-Net, Yedroudj-Net and our proposed neural network on an image dataset consisting 40,000 images with size 256×256. This image dataset [PPIC19] was actually created by Pibre *et al.* from the popular BOSSbase dataset [BFP11]. The BOSSbase dataset contains 10,000 grayscale images with size 512×512. Each image in the BOSS dataset is divided into four images with size 256. Therefore, they obtained 40,000 images from the BOSSbass dataset and put them on their website.

For each embedding scheme, we produce 40,000 stego images and therefore we have 40,000 cover/stego image pairs (i.e., 40,000 cover images + 40,000 stego images). We generate the training, validation and test datasets as follows. We randomly choose 4,000 cover/stego image pairs to form the test dataset, and 8,000 cover/stego image pairs to form the validation dataset. The training dataset consists of the remainder 28,000 cover/stego image pairs. All of the CNN architectures are run on a single GPU (TITAN XP or GeForce GTX) with 12 GB of memory. During the

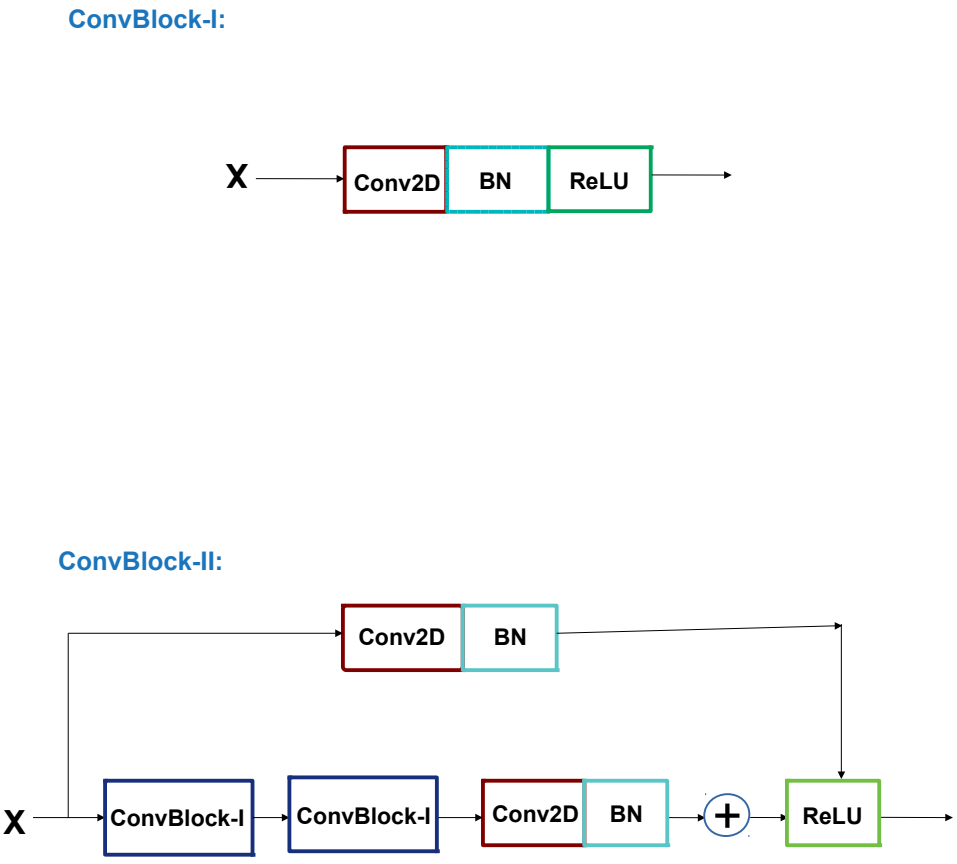


Figure 4.2. Two types of blocks: Convolutional Block-I and Convolutional Block-II

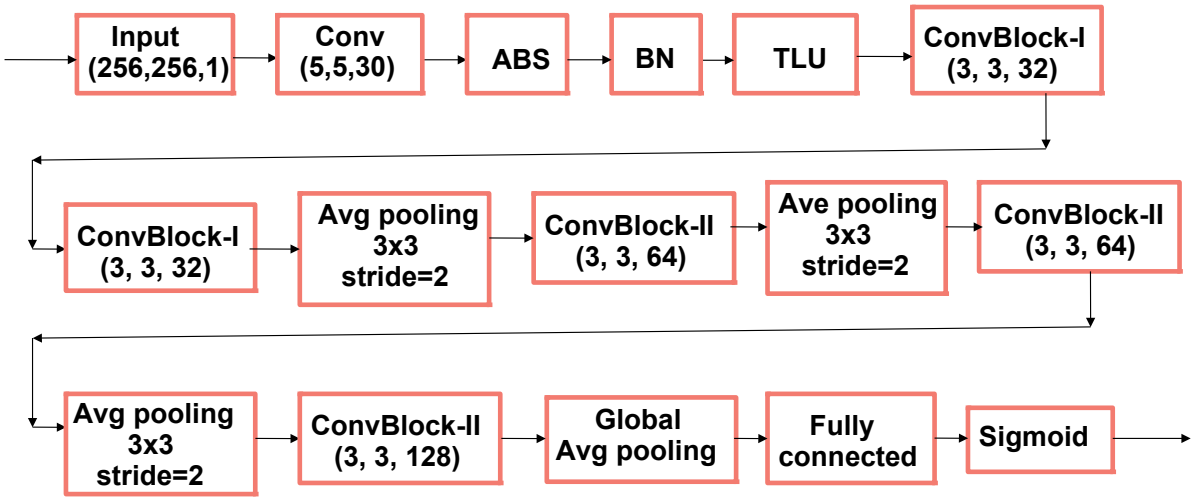


Figure 4.3. Proposed CNN architecture diagram-I

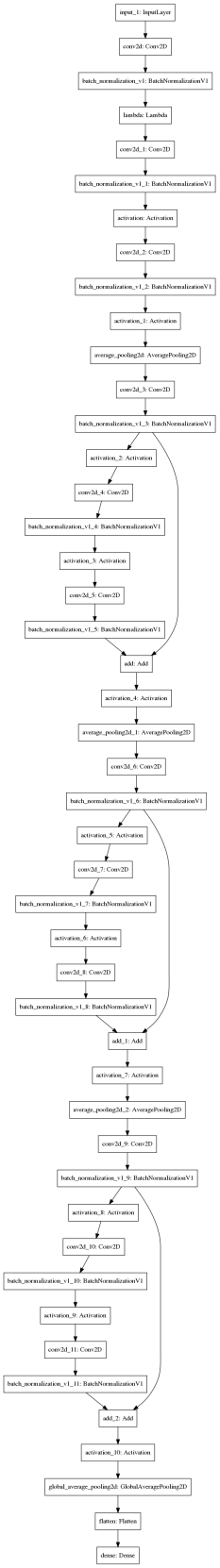


Figure 4.4. Proposed CNN architecture diagram-II

CNN training, we fix the maximum of 450 epochs. Also, I use the popular deep learning APIs *Keras* [C+15] with *Tensorflow* being backend to implement all three CNN architectures used in the experiment.

4.3.2 Experimental Results

In the following, we will show some training logs for Ye-Net, Yedroudj-Net and the proposed neural network. Figure 4.8 shows the training and validation accuracy as well as loss for Ye-Net. The embedding scheme is WOW with the payload 0.4. Figures 4.9 and 4.10 exhibit the training and validation accuracy as well as loss for Yedroudj-Net. The embedding scheme is S-UNIWARD with the payloads being 0.4 and 0.2 respectively. Figure 4.11 presents the training and validation accuracy as well as loss for the proposed neural network. The embedding scheme is WOW with the payload being 0.4.

Table 4.1 shows the detection error rates for three different embedding schemes using SRM (the rich model with the ensemble learning), Xu-Net, Ye-Net, Yedroudj-Net and the proposed CNN architecture. As can be seen, the proposed neural network outperforms the others for the embedding schemes WOW and S-UNIWARD with the payloads 0.2, 0.3 and 0.4, and for the embedding scheme MiPOD with the payload 0.2. Yedroudj-Net performs best for the embedding scheme MiPOD with the payloads 0.3 and 0.4.

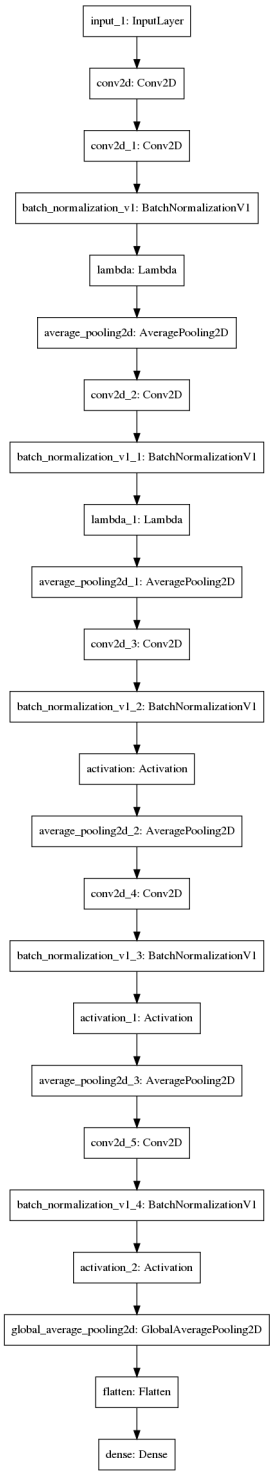


Figure 4.5. Xu-Net architecture diagram

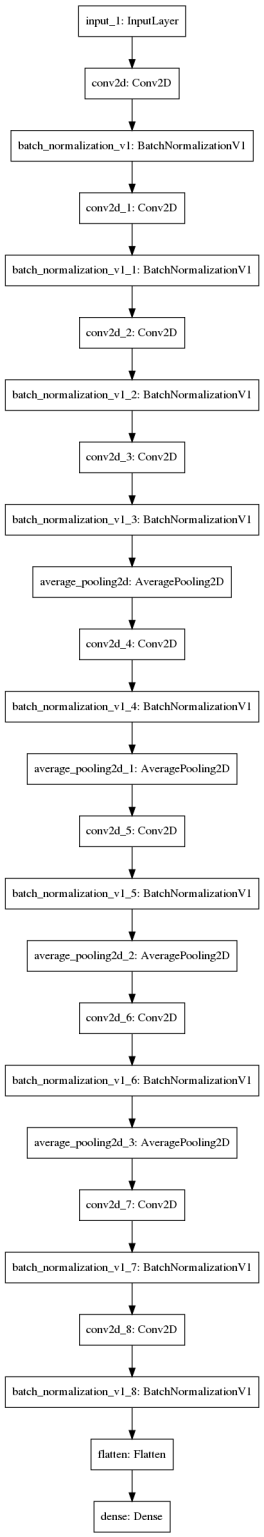


Figure 4.6. Ye-Net architecture diagram

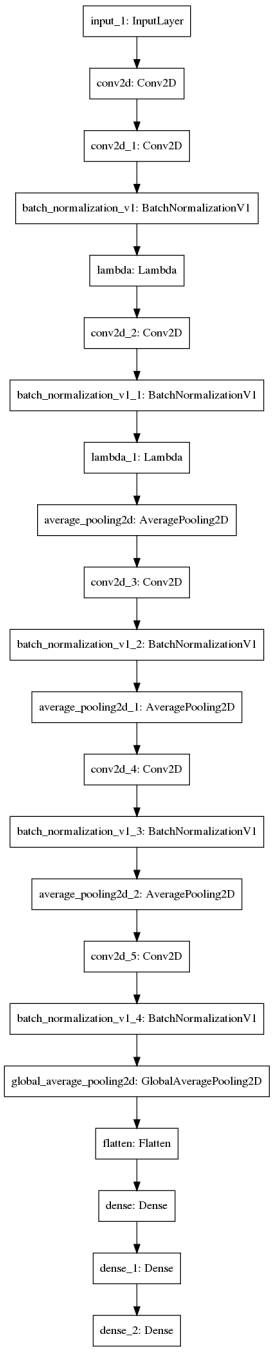


Figure 4.7. Yedroudj-Net architecture diagram

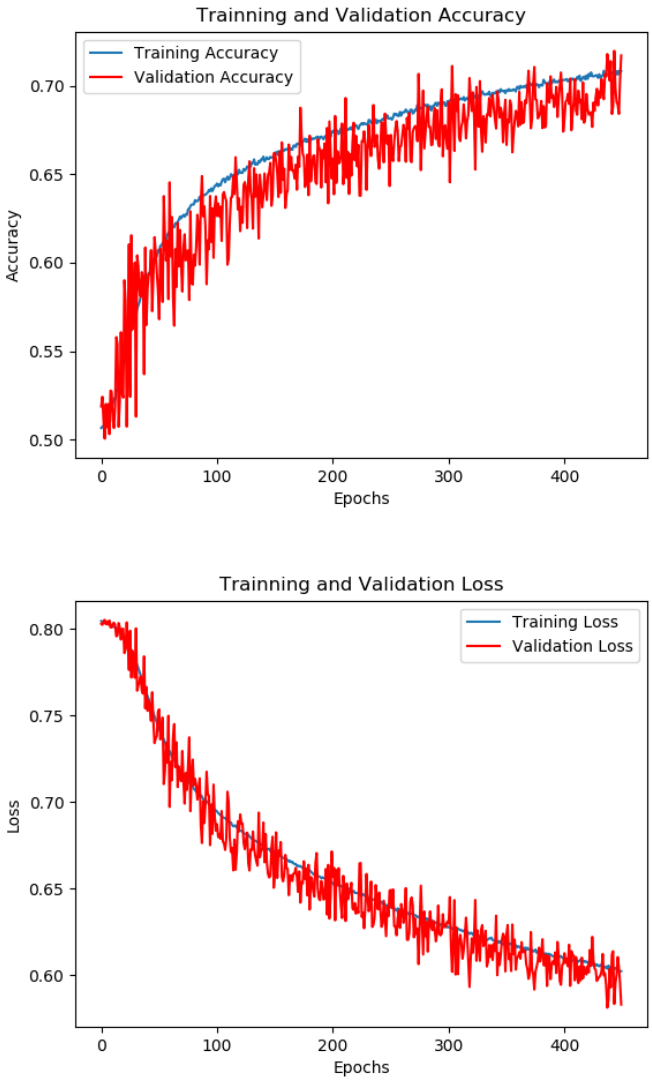


Figure 4.8. Ye-Net for WOW with payload 0.4pp

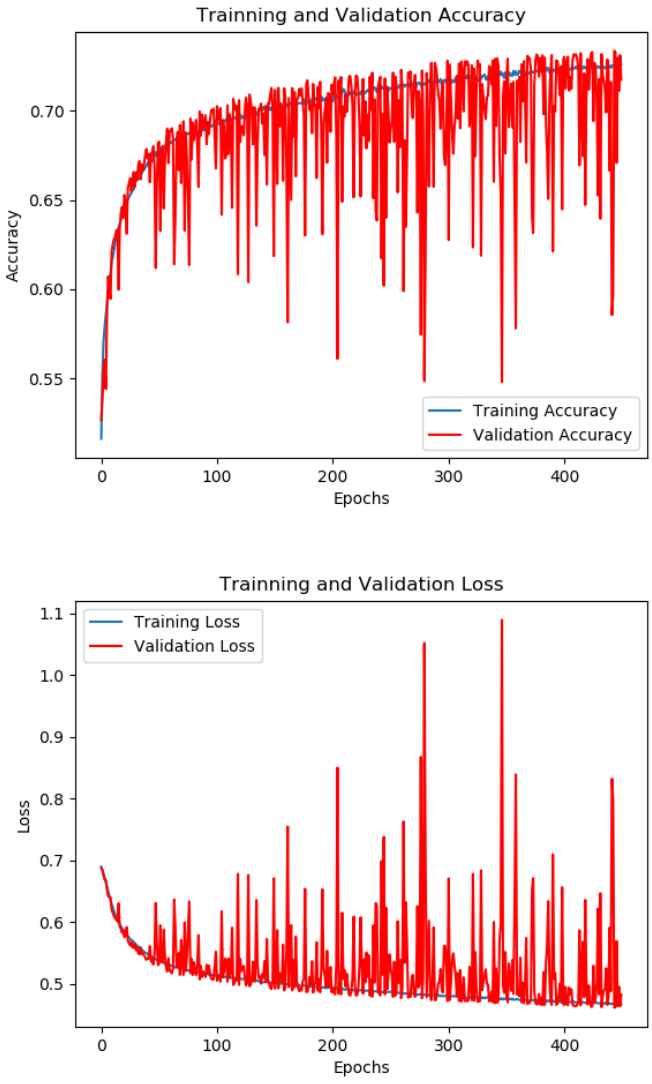


Figure 4.9. Yedroudj-Net for S-UIWARD with payload 0.4pp

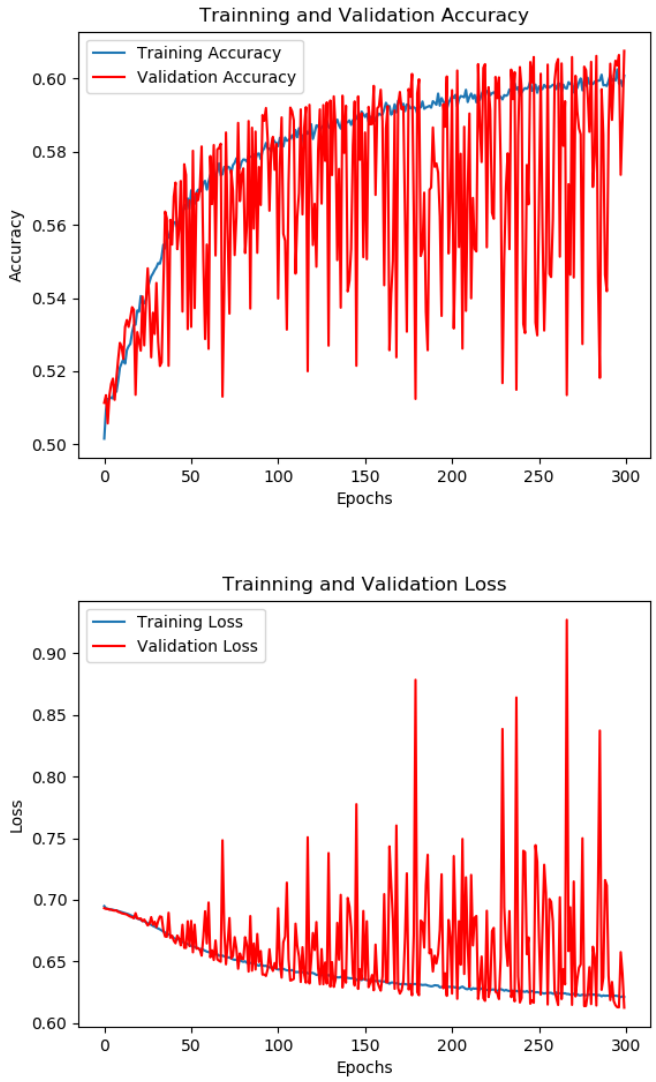


Figure 4.10. Yedroudj-Net for S-UIWARD with payload 0.2pp

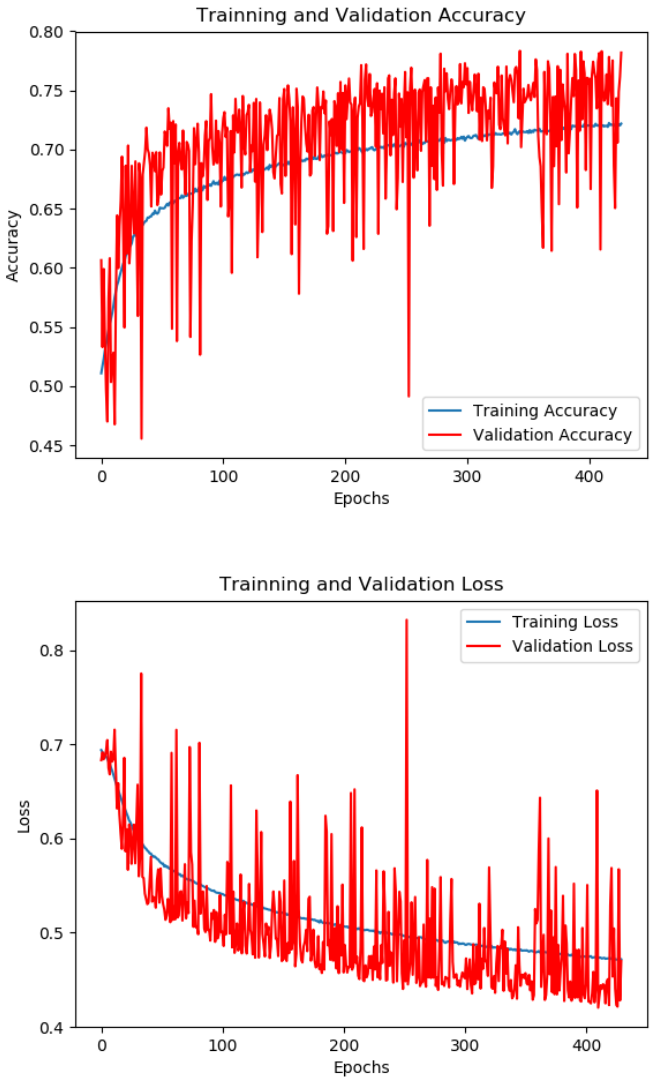


Figure 4.11. The proposed Net for WOW with payload 0.4pp

Table 4.1.
Detection error rates for WOW, MiPOD and S-UNIWARD.

Steganography	Payload (bpp)	SRM	Xu-Net	Ye-Net	Yedroudj-Net	Proposed Net
WOW	0.2	0.391	0.409	0.397	0.392	**0.388**
	0.3	0.345	0.348	0.343	0.340	**0.337**
	0.4	0.286	0.302	0.288	0.281	**0.274**
MiPOD	0.2	0.396	0.386	0.380	0.379	**0.377**
	0.3	0.322	0.318	0.317	**0.306**	0.311
	0.4	0.284	0.262	0.255	**0.246**	0.249
S-UNIWARD	0.2	0.398	0.391	0.387	0.385	**0.379**
	0.3	0.353	0.349	0.341	0.315	**0.302**
	0.4	0.299	0.287	0.276	0.271	**0.267**

5 SUMMARY AND FUTURE WORK

In this thesis we investigate statistical steganalysis of images. We start with a description of image steganography. We introduce different kinds of embedding schemes and then study an information hiding problem using Sudoku. We propose an hiding scheme using a reference matrix to improve the quality of resulting stego images in terms of PSNRs. Experiments demonstrate the effectiveness of the proposed scheme, see Chapter 2.

Next, we consider image steganalysis using the hypothesis testing framework. This framework allows us to conduct a theoretical analysis of performance of steganalysis. We show the score test for the resulting hypothesis problem is not only AMP but also LAUMP. In comparison to the commonly used LR test, the proposed score test drop an unrealistic assumption used by the LR test which requires the local variances in an inhomogenous image mode are greater than one. Experiments on the BOSSbase image dataset show the score test is comparable to the LR test, and outperforms it when the embedding rate is small, see Chapter 3. The challenge for the hypothesis testing approach is that it relies on cover image models and certain embedding schemes. It is challenging to establish a hypothesis testing problem when complex embedding schemes are used, such as content adaptive embedding schemes. Those challenges lead us to adopting the machine learning/deep learning approach for image steganalysis.

Deep neural networks can self-learn complex structures or features from data through the training process in contrast to the machine learning approach using handcrafted features. Inspired by the wide use of high-pass filtering module, we propose a CNN architecture which includes a trainable CNN layer in the beginning with some constraints on its kernels. Such constraints can force this CNN layer to conduct high-pass filtering and at the same time update their weights from the

training data. This design combines two ideas used in previous work: (1) the use of fixed high-pass filters with non-trainable kernels; (2) the use of the high-pass filters only for initialization of a convolutional layer. In addition, the residual module is employed so that the architecture may go deeper (e.g., more convolutional layers may be used in the architecture). Experiments on a large image dataset show that the proposed CNN architecture outperforms commonly used the state-of-the-art detector using handcrafted features, Xu-Net, Ye-Net and Yedroudj-Net in most cases.

In the future, we plan to use the proposed CNN for steganalysis on JPEG domain with possible adjustment. Note that training a deep neural network is expensive and so the use of transfer learning [SRASC14] would be a start of addressing this issue and be worth making efforts. Also, developing lightweight depthwise convolutions, see [SHZ$^+$18], for image steganalysis. This may save a lot of computations without compromising the performance.

REFERENCES

[BCF19] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2019.

[BF] P Bas and T Furon. Bows-2 (july 2007), http://bows2.ec-lille.fr/.

[BFP11] Patrick Bas, Tomáš Filler, and Tomáš Pevnỳ. break our steganographic system: the ins and outs of organizing boss. In *International workshop on information hiding*, pages 59–70. Springer, 2011.

[BG62] Ralph A Bradley and John J Gart. The asymptotic properties of ml estimators when sampling from associated populations. *Biometrika*, 49(1/2):205–214, 1962.

[C+15] François Chollet et al. Keras, 2015.

[CCGS16] Jean-François Couchot, Raphaël Couturier, Christophe Guyeux, and Michel Salomon. Steganalysis via a convolutional neural network using large convolution filters for embedding process with same stego key. *arXiv preprint arXiv:1605.07946*, 2016.

[CCK08] Chin-Chen Chang, Yung-Chen Chou, and The Duc Kieu. An information hiding scheme using sudoku. In *2008 3rd international conference on innovative computing information and control*, pages 17–17. IEEE, 2008.

[CCS17] Jean-Francois Couchot, Raphaël Couturier, and Michel Salomon. Improving blind steganalysis in spatial domain using a criterion to choose the appropriate steganalyzer between cnn and srm+ ec. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 327–340. Springer, 2017.

[CR13] Remi Cogranne and Florent Retraint. An asymptotically uniformly most powerful test for lsb matching detection. *IEEE transactions on information forensics and security*, 8(3):464–476, 2013.

[CS08] Chunhua Chen and Yun Q Shi. Jpeg image steganalysis utilizing both intrablock and interblock correlations. In *2008 IEEE International Symposium on Circuits and Systems*, pages 3029–3032. IEEE, 2008.

[CSKM05] Joon-Hyuk Chang, Jong Won Shin, Nam Soo Kim, and Sanjit K Mitra. Image probability distribution based on generalized gamma function. *IEEE Signal Processing Letters*, 12(4):325–328, 2005.

[CZF+11] Rémi Cogranne, Cathel Zitzmann, Lionel Fillatre, Florent Retraint, Igor Nikiforov, and Philippe Cornu. A cover image model for reliable steganalysis. In *International Workshop on Information Hiding*, pages 178–192. Springer, 2011.

[CZR+12] Rémi Cogranne, Cathel Zitzmann, Florent Retraint, Igor Nikiforov, Lionel Fillatre, and Philippe Cornu. Statistical detection of lsb matching using hypothesis testing theory. In *International Workshop on Information Hiding*, pages 46–62. Springer, 2012.

[DSM+04] Onkar Dabeer, Kenneth Sullivan, Upamanyu Madhow, Shivkumar Chandrasekaran, and BS Manjunath. Detection of hiding in the least significant bit. *IEEE Transactions on Signal Processing*, 52(10):3046–3058, 2004.

[DWW02] Sorina Dumitrescu, Xiaolin Wu, and Zhe Wang. Detection of lsb steganography via sample pair analysis. In *International Workshop on Information Hiding*, pages 355–372. Springer, 2002.

[Fil12] Lionel Fillatre. Adaptive steganalysis of least significant bit replacement in grayscale natural images. *IEEE Transactions on Signal Processing*, 60(2):556–569, 2012.

[FK12] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.

[FQY07] Jian-Wen Fu, Yin-Cheng Qi, and Jin-Sha Yuan. Wavelet domain audio steganalysis based on statistical moments and pca. In *2007 International Conference on Wavelet Analysis and Pattern Recognition*, volume 4, pages 1619–1623. IEEE, 2007.

[Fri09] Jessica Fridrich. *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GFH06] Miroslav Goljan, Jessica Fridrich, and Taras Holotyak. New blind steganalysis and its implications. In *Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, page 607201. International Society for Optics and Photonics, 2006.

[GS01] Ulf Grenander and Anuj Srivastava. Probability models for clutter in natural images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):424–429, 2001.

[HCS08] Wien Hong, Tung-Shou Chen, and Chih-Wei Shiu. Steganography using sudoku revisited. In *2008 Second International Symposium on Intelligent Information Technology Application*, volume 2, pages 935–939. IEEE, 2008.

[HF12] Vojtěch Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*, pages 234–239. IEEE, 2012.

[HF13] Vojtech Holub and Jessica Fridrich. Random projections of residuals for digital image steganalysis. *IEEE Transactions on Information Forensics and Security*, 8(12):1996–2006, 2013.

[HF15] Vojtěch Holub and Jessica Fridrich. Phase-aware projection model for steganalysis of jpeg images. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090T. International Society for Optics and Photonics, 2015.

[HFD14] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1):1, 2014.

[HLVDMW17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[JW88] F-C Jeng and John William Woods. Inhomogeneous gaussian image models for estimation and restoration. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(8):1305–1312, 1988.

[JY09] Ki-Hyun Jung and Kee-Young Yoo. Improved exploiting modification direction method by modulus operation. *International Journal of Signal processing, Image processing and pattern*, 2(1):79–87, 2009.

[KB08] Andrew D Ker and Rainer Böhme. Revisiting weighted stego-image steganalysis. In *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, page 681905. International Society for Optics and Photonics, 2008.

[Ker05a] Andrew D Ker. A general framework for structural steganalysis of lsb replacement. In *International Workshop on Information Hiding*, pages 296–311. Springer, 2005.

[Ker05b] Andrew D Ker. Steganalysis of lsb matching in grayscale images. *IEEE signal processing letters*, 12(6):441–444, 2005.

[KF10] Jan Kodovsky and Jessica Fridrich. Quantitative structural steganalysis of jsteg. *IEEE Transactions on Information Forensics and Security*, 5(4):681–693, 2010.

[KF11] Jan Kodovskỳ and Jessica Fridrich. Steganalysis in high dimensions: Fusing classifiers built on random subspaces. In *Media Watermarking, Security, and Forensics III*, volume 7880, page 78800L. International Society for Optics and Photonics, 2011.

[KFH11] Jan Kodovsky, Jessica Fridrich, and Vojtech Holub. On dangers of overtraining steganography to incomplete cover model. In *Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security*, pages 69–76. ACM, 2011.

[KFH12] Jan Kodovsky, Jessica Fridrich, and Vojtěch Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, 2012.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KSSC85] Darwin T Kuan, Alexander A Sawchuk, Timothy C Strand, and Pierre Chavel. Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2):165–177, 1985.

[LCW08] Chin-Feng Lee, Chin-Chen Chang, and Kuo-Hua Wang. An improvement of emd embedding method for large payloads by pixel segmentation strategy. *Image and Vision Computing*, 26(12):1670–1676, 2008.

[LF04] Siwei Lyu and Hany Farid. Steganalysis using color wavelet statistics and one-class support vector machines. In *Security, steganography, and watermarking of multimedia contents VI*, volume 5306, pages 35–45. International Society for Optics and Photonics, 2004.

[LF06] Siwei Lyu and Hany Farid. Steganalysis using higher-order image statistics. *IEEE transactions on Information Forensics and Security*, 1(1):111–119, 2006.

[LG00] Edmund Y Lam and Joseph W Goodman. A mathematical analysis of the dct coefficient distributions for images. *IEEE transactions on image processing*, 9(10):1661–1666, 2000.

[LR06] Erich L Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

[LWHL14] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. A new cost function for spatial image steganography. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4206–4210. IEEE, 2014.

[Mie06] Jarno Mielikainen. Lsb matching revisited. *IEEE signal processing letters*, 13(5):285–287, 2006.

[Mül93] F Müller. Distribution shape of two-dimensional dct coefficients of natural images. *Electronics Letters*, 29(22):1935–1936, 1993.

[PBF10] Tomáš Pevny, Patrick Bas, and Jessica Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security*, 5(2):215–224, 2010.

[PF06] Tomáš Pevnỳ and Jessica Fridrich. Multi-class blind steganalysis for jpeg images. In *Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, page 60720O. International Society for Optics and Photonics, 2006.

[PF07] Tomas Pevny and Jessica Fridrich. Merging markov and dct features for multi-class jpeg steganalysis. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, page 650503. International Society for Optics and Photonics, 2007.

[PF08] Tomas Pevny and Jessica Fridrich. Multiclass detector of current steganographic methods for jpeg format. *IEEE Transactions on Information Forensics and Security*, 3(4):635–650, 2008.

[PFB10] Tomáš Pevnỳ, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In *International Workshop on Information Hiding*, pages 161–177. Springer, 2010.

[PM92] William B Pennebaker and Joan L Mitchell. *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[PPIC16] Lionel Pibre, Jérôme Pasquet, Dino Ienco, and Marc Chaumont. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch. *Electronic Imaging*, 2016(8):1–11, 2016.

[PPIC19] Lionel Pibre, Jérôme Pasquet, Dino Ienco, and Marc Chaumont. *CroppedBOSSBase, http://www.lirmm.fr/ chaumont/SteganalysisWithDeepLearning.html*, 2016 (accessed March 13, 2019).

[QDWT15] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090J. International Society for Optics and Photonics, 2015.

[QSXN09] Jiaohua Qin, Xingming Sun, Xuyu Xiang, and Changming Niu. Principal feature selection and fusion method for image steganalysis. *Journal of Electronic Imaging*, 18(3):033009, 2009.

[RG10] Mahdi Ramezani and Shahrokh Ghaemmaghami. Towards genetic feature selection in image steganalysis. In *2010 7th IEEE Consumer Communications and Networking Conference*, pages 1–4. IEEE, 2010.

[RM97] C Radhakrishna Rao and Rahul Mukerjee. Comparison of lr, score, and wald tests in a non-iid setting. *Journal of Multivariate Analysis*, 60(1):99–110, 1997.

[SCC06] Yun Q Shi, Chunhua Chen, and Wen Chen. A markov process based approach to effective attacking jpeg steganography. In *International Workshop on Information Hiding*, pages 249–264. Springer, 2006.

[SCF16] Vahid Sedighi, Rémi Cogranne, and Jessica Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2):221–234, 2016.

[SHZ+18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[Sim84] Gustavus J Simmons. The prisoners problem and the subliminal channel. In *Advances in Cryptology*, pages 51–67. Springer, 1984.

[SLG02] Anuj Srivastava, Xiuwen Liu, and Ulf Grenander. Universal analytical forms for modeling image probabilities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1200–1214, 2002.

[SRASC14] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[SS03] Gerald Schaefer and Michal Stich. Ucid: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 472–481. International Society for Optics and Photonics, 2003.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TCR14] Thanh Hai Thai, Remi Cogranne, and Florent Retraint. Statistical model of quantized dct coefficients: Application in the steganalysis of jsteg algorithm. *IEEE Transactions on Image Processing*, 23(5):1980–1993, 2014.

[TL14] Shunquan Tan and Bin Li. Stacked convolutional auto-encoders for steganalysis of digital images. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–4. IEEE, 2014.

[Web18] Allan G Weber. *The USC-SIPI Image Database (1997), http://sipi.usc.edu/database/*, 1997 (accessed January 26, 2018).

[XWS16a] Guanshuo Xu, Han-Zhou Wu, and Yun Q Shi. Ensemble of cnns for steganalysis: An empirical study. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 103–107. ACM, 2016.

[XWS16b] Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, 2016.

[YCC18a] Mehdi Yedroudj, Marc Chaumont, and Frédéric Comby. How to augment a small learning set for improving the performances of a cnn-based steganalyzer? *Electronic Imaging*, 2018(7):1–7, 2018.

[YCC18b] Mehdi Yedroudj, Frédéric Comby, and Marc Chaumont. Yedroudj-net: An efficient cnn for spatial steganalysis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2092–2096. IEEE, 2018.

[YNY17] Jian Ye, Jiangqun Ni, and Yang Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, 2017.

[YSWK17] Jianhua Yang, Yun-Qing Shi, Edward K Wong, and Xiangui Kang. Jpeg steganalysis based on densenet. *arXiv preprint arXiv:1711.09335*, 2017.

[YWT04] Xiaoyi Yu, Yunhong Wang, and Tieniu Tan. On estimation of secret message length in jsteg-like steganography. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 673–676. IEEE, 2004.

[ZCR+11] Cathel Zitzmann, Rémi Cogranne, Florent Retraint, Igor Nikiforov, Lionel Fillatre, and Philippe Cornu. Statistical decision methods in hidden information detection. In *International Workshop on Information Hiding*, pages 163–177. Springer, 2011.

[ZP03] Tao Zhang and Xijian Ping. A fast and effective steganalytic technique against jsteg-like algorithms. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 307–311. ACM, 2003.

[ZTLH18] Jishen Zeng, Shunquan Tan, Bin Li, and Jiwu Huang. Large-scale jpeg image steganalysis using hybrid deep-learning framework. *IEEE Transactions on Information Forensics and Security*, 13(5):1200–1214, 2018.

VITA

Min Huang received the B.S. degree and the M.S. degree in mathematics from Shandong University in 1994 and 1997, respectively. After working in industry, he went to the Chinese Academy of Sciences and received the Ph.D. degree in mathematics in 2003 under the advisement of Professor Yuesheng Xu. He received the M.S. degree in computer science from Purdue University in 2010. His research interests include developing efficient methods for PDEs, statistical image steganalysis and applications to machine learning and deep neural networks.