

Novel Steganalysis Method for Unknown Embedding Rates using Transfer and Multi-Task Learning

Lan Wu^{*} and Xiaolei Han

College of Electrical Engineering, Henan University of Technology, Zhengzhou, 451200, China

Abstract

Existing image steganalysis methods based on deep learning assume that the embedding rates are known, whereas for most practical applications, these rates are unknown, leading to a sharp drop in model detection performance. This study combined transfer learning (TL) and multi-task learning (MTL) and proposed an image steganalysis method for a specific steganographic algorithm and unknown embedding rates. The proposed method used stego images with high embedding rates to pre-train the steganalysis model, constructed a steganalysis model based on MTL, and then transferred the parameter values of the pre-trained model as the initial values. The parameters were further fine-tuned on the training set, which consists of cover images and stego images with various embedding rates. A new objective function was designed by applying the weighting losses to the uncertainty method, dynamically adjusting the weight of each sub-task during the training process. The proposed method extracted the common features of images with various embedding rates more effectively, achieved better detection accuracy on images with unknown embedding rates, and demonstrated improved generalization ability.

Keywords: steganalysis; transfer learning; multi-task learning; weighting losses with uncertainty

(Submitted on xx xx, 2019; Revised on xx xx, 2019; Accepted on xx xx, 2019)

© 2019 Totem Publisher, Inc. All rights reserved.

1. Introduction

Image steganography utilizes the spatial redundancy of an image to hide meaningful secret information in the cover image, thereby obtaining a stego image. Image steganalysis is used to detect whether an image contains secret information by distinguishing the cover image or stego image. It is usually regarded as a binary classification problem [1]. Image steganalysis methods can be divided into specific detection and general blind detection. Specific detection occurs when the steganographic algorithm and embedding rates of the image are known, while general blind detection occurs when the steganographic algorithm or the embedding rates are unknown. The detection accuracy of the former is usually higher than that of the latter. However, for most practical applications, steganographic algorithms and embedding rates are unknown, leading to a sharp drop in model detection performance. Therefore, it is very important to explore general blind detection technology in the field of steganalysis.

Steganalysis methods are mainly based on artificial features or based on deep learning. The steganalysis method based on artificial features can be divided into two steps: artificial feature extraction and classifier training. Features refer to the statistics that can distinguish the cover images and the stego images, and classifiers refer to a kind of classifier that can be trained and optimized, such as support vector machines or ensemble classifiers. Among these two steps, artificial feature extraction is a key problem in research and plays a decisive role in detection performance. Its basic idea is to find statistics with obvious differences between images before and after steganography. The latest artificial feature extraction methods, such as the spatial rich model (SRM) [2] and local binary pattern (LBP) [3], all use filters to pre-process the input images to obtain residual images and then combine features extracted from residual images to obtain more rich high-dimensional features, so as to improve the detection performance of the model. However, with the development of steganography, this method faces many difficulties and challenges. Artificial design features are highly dependent on expert experience and extremely time-consuming, and the feature completeness is inadequate. In addition, steganography and steganalysis are

^{*} Corresponding author.

E-mail address: wulan@haut.edu.cn

progressing in mutual confrontation. The development of steganography puts forward more and more requirements for steganalysis, and the difficulty of designing artificial features for steganalysis is also increasing.

By contrast, steganalysis based on deep learning can integrate feature learning into the modelling process. It can effectively alleviate the shortcomings of artificial design features and has attracted the attention of many researchers [4-6]. Yu et al. [7] proposed a steganalysis method based on MTL [8] according to the steganalysis method based on convolutional neural network (CNN). The method adds an auxiliary task to estimate whether each pixel in an image has been modified. By sharing features between related tasks, the obtained model can achieve better detection accuracy. Aimed at the difficulty of the detection of stego images with low embedding rates, Qian et al. [9] proposed a steganalysis method based on TL [10]. The embedding region of stego images with high embedding rates contains those with low embedding rates, and the features of stego images with high embedding rates are relatively easy to extract. The method uses cover images and stego images with high embedding rates to pre-train a steganalysis model and then further fine-tunes the model parameters of the training set that consists of cover images and stego images with low embedding rates. Thus, the steganalysis performance for low embedding rate stego images was improved. However, when the embedding rates are unknown, it is difficult to utilize the methods mentioned above. Cancelli et al. [11] studied the problem associated with detection performance degradation when the embedding rates used in the training and testing models were inconsistent. They found that the steganalysis model trained under certain embedding rates can easily be mismatched, leading to a drastic reduction in the detection performance.

To the best of our knowledge, existing steganalysis methods based on deep learning assume that the image embedding rates are known, while the steganalysis methods for unknown embedding rates are seldom studied. Therefore, in this paper, we proposed a novel steganalysis method for a specific steganographic algorithm and unknown embedding rates. The proposed method uses stego images with high embedding rates to pre-train the steganalysis model and then constructs a steganalysis model based on MTL. The parameter values of the pre-trained model are transferred as the initial values and are further fine-tuned on the training set that consists of cover images and stego images with various embedding rates. In addition, we designed an objective function for our proposed steganalysis method by applying the weighting losses to the uncertainty method [12-13] to allocate the weight of each sub-task dynamically.

In summary, the key contributions of the present study are the following:

- A steganalysis framework based on MTL and TL was developed to realize image steganalysis blind detection for a specific steganographic algorithm and unknown embedding rates.
- A new objective function was designed for our proposed steganalysis method to allocate the weight assignment of each task dynamically. This reduced the consumption of time and computing resources from the artificial adjustment of weights and improved the training efficiency and the detection performance of the model.

The paper is organized as follows. Section 2 provides a brief description of the steganalysis method based on deep learning, transfer learning, and multi-task learning. Section 3 details the implementation steps of the proposed steganalysis method and the design of its objective function. Section 4 implements four sets of progressive experiments and demonstrates the effectiveness of the proposed method through the experimental results. Section 5 summarizes the results of the study and discusses future research.

2. Related Works

2.1. Steganalysis based on Deep Learning

Deep learning has attracted the attention of researchers in various fields due to its powerful non-linear fitting ability. The steganalysis framework based on deep learning was first introduced in [4]. The framework consists of a pre-processing layer, several convolutional layers, a fully connected layer, and a softmax classifier (Figure 1). The pre-processing layer uses a high-pass filter to convolute the input image and obtain the residual image, which can suppress image content, expose stego noise, and improve signal-to-noise ratio (SNR), thereby improving the performance of steganalysis. The convolutional layers are used to abstract features. The fully connected layer and softmax classifier squash the abstracted features to the probability prediction value corresponding to the category to which the image belongs. This has become the most commonly used steganalysis framework based on deep learning. Although many researchers have improved the framework [5-6], the basic form has not changed much.

2.2. Steganalysis based on Transfer Learning

Transfer learning provides an effective solution to the problems of over-fitting, class imbalance, and cross-domain feature

distribution differences during the modeling of small-scale datasets. Pan et al. [10] defined transfer learning as follows:

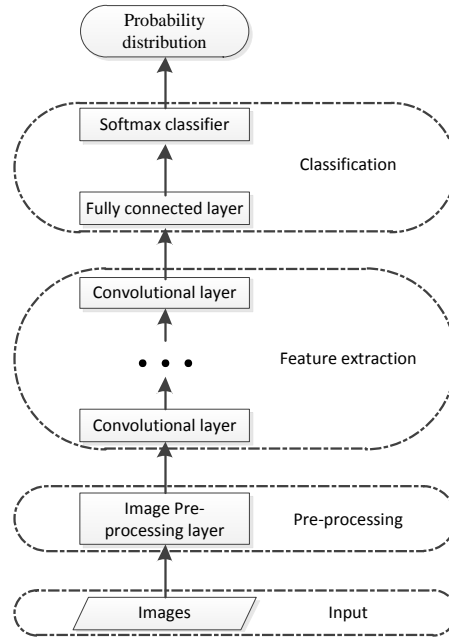


Figure 1. Steganalysis framework based on deep learning

Definition 1 (Domain): The domain can be expressed as $D = \{\mathcal{X}, p(X)\}$ where \mathcal{X} denotes the feature space and $p(X)$ denotes the marginal distribution of $X = \{x_1, x_2, \dots, x_n\} \in \mathcal{X}$, $x_i \in \mathbb{R}^d$, $i = 1, 2, \dots, n$.

Definition 2 (Task): When given a domain $D = \{\mathcal{X}, p(X)\}$, the task can be expressed as $\Gamma = \{Y, f(\cdot)\}$, where $Y = \{y_1, y_2, \dots, y_n\}$ denotes the class label space and $f(\cdot)$ is the target predicting function obtained from the training data $\{x_i, y_i\}$, i.e., the trained model. For a sample x , from the perspective of probability distribution, $f(x)$ can be written as $p(y|x)$.

Definition 3 (Transfer learning): Given a source domain D_s and a source task Γ_s , as well as a target domain D_t and a target task Γ_t , transfer learning uses the existing knowledge of D_s and Γ_s to improve the learning of the target prediction function on the target domain D_t , where $D_s \neq D_t$ and $\Gamma_s \neq \Gamma_t$. The process of transfer learning is essential for transferring the knowledge learned in the source domain D_s to solve the task Γ_s to the target domain D_t , and it constructs a model for the target task Γ_t .

Pan et al. [10] believed that mining the commonality between the source and target domains via TL is beneficial for improving the performance of the target task. If the source and target domain tasks are similar, the model parameters in the source task contribute to the learning of the parameters in the target task. Yosinski et al. [14] studied the transferability of deep neural networks and found that the effectiveness of the feature transfer is expected to grow as the dataset in the source and target domains became more similar. When training a model, it is better to use TL to initialize the parameters and fine-tune them than it is to use randomly initialized parameters.

To solve the problem that stego image features with low embedding rates are difficult to extract and the model does not converge, Qian et al. [9] used stego images with high embedding rates and their steganalysis as the source domain and source task, and they used stego images with low embedding rates and their steganalysis as the target domain and target task. Then, the parameters learned from the source task were transferred to the target task and further fine-tuned by using TL, thus effectively improving the steganalysis performance of the model for low embedding rates stego images. This method trains the steganalysis model for specific embedding rates based on the embedding rates of the stego images from high to low. As shown in Figure 2, the embedding rates of stego images are $A > B > C > D$.

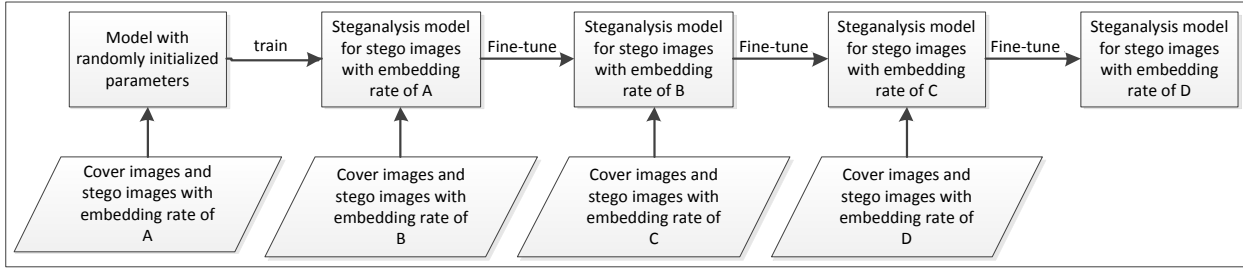


Figure 2. Steganalysis framework based on TL, the embedding rates of stego images are $A > B > C > D$

2.3. Steganalysis based on MTL

MTL allows a single model to perform multiple tasks simultaneously. Because it involves the sharing and transfer of knowledge among multiple tasks, it is regarded as a category of TL [10]. The purpose of MTL is to learn common feature expressions from multiple goals to improve learning efficiency and task performance. It has been widely used in the fields of computer vision [15], natural language processing [16], and speech recognition [17]. It is considered a knowledge induction method, which improves the generalization ability of the model by sharing domain information between complementary tasks.

MTL involves the joint optimization problem of multiple targets. Usually, the loss function of each task is weighted and summed to obtain the objective function to be optimized, as shown in Equation (1) (n is the number of tasks).

$$L_{\text{total}} = \sum_{i=1}^n w_i L_i \quad (1)$$

Yu et al. [7] proposed a steganalysis method based on MTL. As shown in Figure 3, the main task is a binary classification task to distinguish the cover and stego image, and the auxiliary task is a pixel binary classification task to estimate whether each pixel in an image has been modified. By employing the method introduced in Equation (1), weighted summation of these two tasks was used as the objective function to be optimized, as shown in Equation (2).

$$L_{\text{total}} = L_m + \alpha L_a \quad (2)$$

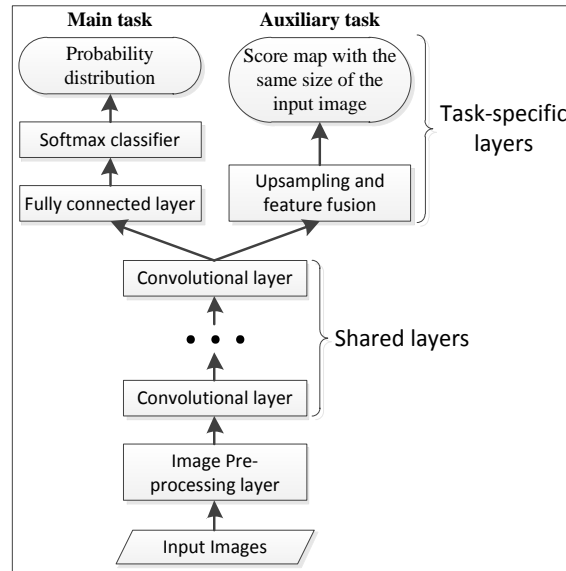


Figure 3. Steganalysis framework based on MTL

Where L_m and L_a are correspond to the loss of the main and auxiliary tasks, respectively, and α represents the weight of loss of the auxiliary task and needs to be adjusted manually. The model needs to be retrained every time α is adjusted.

3. Steganalysis Method based on TL and MTL

Existing image steganalysis methods based on deep learning usually assume that the embedding rates are known. However, in practical applications, it is easy to decrease the detection performance of a steganalysis model that has been trained under a certain embedding rates due to model mismatch. Therefore, in this paper, we studied the image steganalysis blind detection method for a specific steganographic algorithm and unknown embedding rates. The proposed method combines TL and MTL to conduct steganalysis on the same model for images with unknown embedding rates. Aimed at the problem of weight assignment in MTL, the present study designed a new objective function for steganalysis based on task uncertainty [13] and maximum likelihood estimation to dynamically adjust the weight allocation among each sub-task.

3.1. Design of the Proposed Steganalysis Method

First, a classification model for steganalysis is built. As shown on the left of Figure 4, the model consists of an image pre-processing layer, feature extraction component, and softmax classifier. The image pre-processing layer is used to suppress the image content, enhance the steganographic noise, and improve the signal-to-noise ratio. The feature extraction component is formed by a convolutional layer, a batch normalization layer, a pooling layer, and a fully connected layer series. The softmax classifier discretizes the predicted output and used cross-entropy as the loss function. The classification model used randomly initializes parameters and is pre-trained on the training set that consists of the cover images and stego images with high embedding rates.

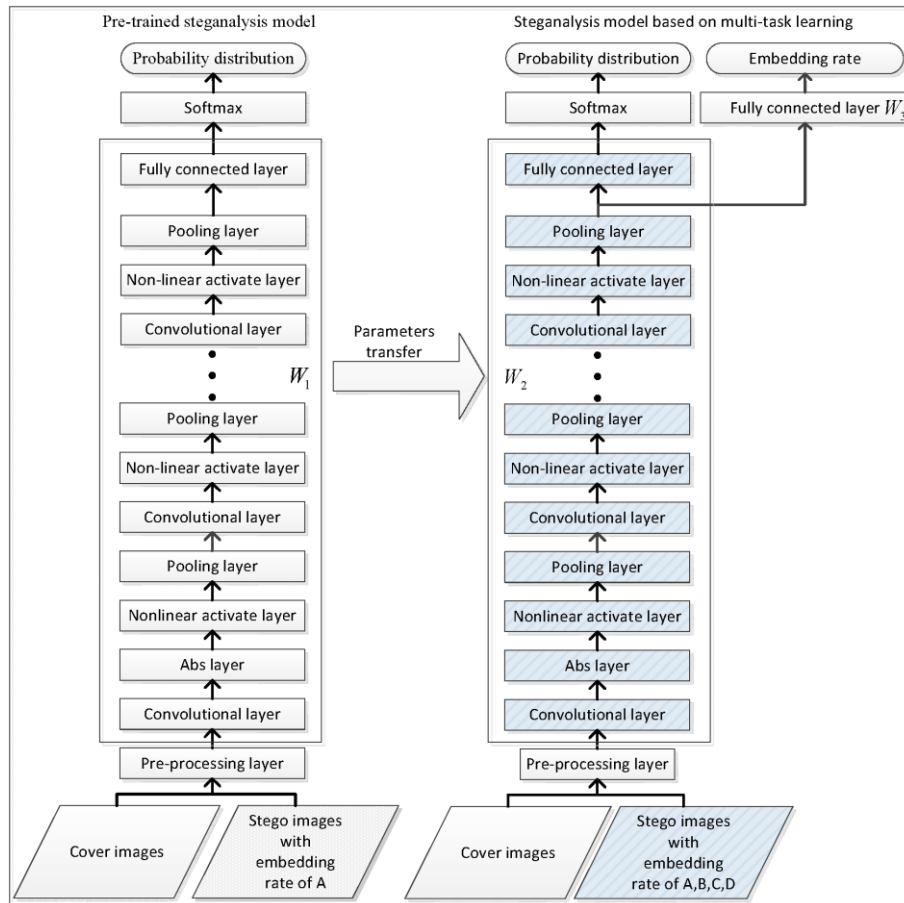


Figure 4. The proposed steganalysis framework based on MTL and TL

The steganalysis model based on MTL that was designed in the present study is shown on the right in Figure 4 and is based on the structure of the pre-trained classification model. The output from the last pooling layer adds a fully connected layer branch to perform the regression task. The classification task is set as the main task to distinguish the cover/stego images, and the regression task is set as the auxiliary task to fit the degree of embedding of secret information in the images.

As shown in Figure 4, the parameters of the pre-trained model are denoted as W_1 , the parameters between the pre-

processing layer and the softmax classifier in the MTL model are denoted as W_2 , and the parameters from the output of the last pooling layer to the auxiliary task are denoted as W_3 . When training the steganalysis model based on MTL, W_1 is assigned to W_2 as the initial value, and W_3 is randomly generated. Then, these parameters are fine-tuned on the training set that is composed of cover images and stego images with various embedding rates. The input image labeling methods for the main and auxiliary tasks are Onehot encoding and serial number coding, respectively, as shown in Table 1.

Table 1. Examples of the input image labeling methods for the main and auxiliary tasks

	Label of main task	Label of auxiliary task
Cover image	[1, 0]	0
0.1 bpp stego image	[0, 1]	1
0.2 bpp stego image	[0, 1]	2
0.3 bpp stego image	[0, 1]	3
0.4 bpp stego image	[0, 1]	4

3.2. Design of the Objective Function

As shown in Equations (1) and (2), the objective functions commonly used in MTL are obtained by weighted summation of each sub-task. However, there are two problems with this function. First, the performance of the model is very sensitive to the values of the weights, and the assignment of inappropriate values might result in poor performance. Second, manually adjusting the weights will consume a large amount of time and computing resources, and it is often difficult to determine the optimal weight assignment. Therefore, in this paper, weighting losses with the uncertainty method [12-13] were applied to design a new objective function for our proposed steganalysis method, and the new objective function can allocate the weight of each sub-task dynamically during the training process.

Suppose the input image of the steganalysis model based on MTL is x , the weight is $W=W_2 \cup W_3$, the expected output of auxiliary task is Y_a , the expected output of the main task is $Y_m=[y_1, y_2]$, s.t. $y_1, y_2 \in \{0,1\}$, $y_1 + y_2 = 1$, and σ_a^2 , σ_m^2 is the error variance of prediction output of the model.

The auxiliary task is a regression task, with its prediction denoted as $f^w(x)$, and the Gaussian likelihood estimate is defined as

$$\begin{aligned}
 L_1(W, \sigma_a) &= p(Y_a | f^w(x), \sigma_a) \\
 &= N(f^w(x), \sigma_a^2) \\
 &= \frac{1}{\sqrt{2\pi}\sigma_a} \exp\left(-\frac{(Y_a - f^w(x))^2}{2\sigma_a^2}\right)
 \end{aligned} \tag{3}$$

The log likelihood function of the auxiliary task can be obtained from Equation (3),

$$\begin{aligned}
 \log L_1(W, \sigma_a) &= \log p(Y_a | f^w(x), \sigma_a) \\
 &\propto -\frac{1}{2\sigma_a^2} \|Y_a - f^w(x)\|^2 - \log \sigma_a \\
 &= -\frac{1}{2\sigma_a^2} L_a(W) - \log \sigma_a
 \end{aligned} \tag{4}$$

The main task is a binary classification task, which also denotes the output of its fully connected layer as $f^w(x)$, and then the scaled $f^w(x)$ by σ_m is squashed to a vector \hat{Y}_m that represents the probability distribution via a softmax function. As shown in Equation (5), $y_c, \hat{y}_c, f_c^w(x)$ represents the c^{th} element of $Y_m, \hat{Y}_m, f^w(x)$, $c \in \{1, 2\}$, $\hat{Y}_m = [\hat{y}_1, \hat{y}_2]$, s.t. $0 \leq \hat{y}_1, \hat{y}_2 \leq 1, \hat{y}_1 + \hat{y}_2 = 1$.

$$\hat{y}_c = p(y_c | f^W(x), \sigma_m) = \text{softmax} \left(\frac{1}{\sigma_m^2} f^W(x) \right) = \frac{\exp \left(\frac{f_c^W(x)}{\sigma_m^2} \right)}{\sum_{k=1}^2 \exp \left(\frac{f_k^W(x)}{\sigma_m^2} \right)} \quad (5)$$

The cross-entropy of the probability distribution of the softmax function from $f^W(x)$ is not scaled by σ_m .

$$\begin{aligned} L_m(W) &= -\frac{1}{2} \left(y_1 \log \frac{\exp(f_1^W(x))}{\sum_{k=1}^2 \exp(f_k^W(x))} + y_2 \log \frac{\exp(f_2^W(x))}{\sum_{k=1}^2 \exp(f_k^W(x))} \right) \\ &= -\frac{1}{2} (f_1^W(x) + f_2^W(x)) + \log \sum_{k=1}^2 \exp(f_k^W(x)) \end{aligned} \quad (6)$$

The log likelihood function of the main task is obtained from Equations (5)-(6).

$$\begin{aligned} \log L_2(W, \sigma_m) &= \log p(y_1, y_2 | f^W(x), \sigma_m) \\ &= \log p(y_1 | f^W(x), \sigma_m) + \log p(y_2 | f^W(x), \sigma_m) \\ &= \frac{1}{\sigma_m^2} f_1^W(x) + \frac{1}{\sigma_m^2} f_2^W(x) - 2 \log \sum_{k=1}^n \exp \left(\frac{1}{\sigma_m^2} f_k^W(x) \right) \\ &= -\frac{2}{\sigma_m^2} L_m(W) + \frac{2}{\sigma_m^2} \log \sum_{k=1}^n \exp(f_k^W(x)) - 2 \log \sum_{k=1}^n \exp \left(\frac{1}{\sigma_m^2} f_k^W(x) \right) \\ &= -\frac{2}{\sigma_m^2} L_m(W) - \log \frac{\sum_{k=1}^n \exp \left(\frac{1}{\sigma_m^2} f_k^W(x) \right)}{\left(\sum_{k=1}^n \exp(f_k^W(x)) \right)^{\frac{1}{\sigma_m^2}}} \end{aligned} \quad (7)$$

When $\sigma_m \rightarrow 1$, introduce $\frac{1}{\sigma_m^2} \sum_{k=1}^n \exp \left(\frac{1}{\sigma_m^2} f_k^W(x) \right) \approx \left(\sum_{k=1}^n \exp(f_k^W(x)) \right)^{\frac{1}{\sigma_m^2}}$ and simplify the second item of Equation (7) to obtain

$$\begin{aligned} \log L_2(W, \sigma_m) &= -\frac{2}{\sigma_m^2} L_m(W) - 2 \log \sigma_m \\ &\propto -\frac{1}{\sigma_m^2} L_m(W) - \log \sigma_m \end{aligned} \quad (8)$$

When these two tasks are jointly optimized, the log likelihood function is

$$\begin{aligned} \log L(W, \sigma_a, \sigma_m) &= \log p(Y_a, Y_m | f^W(x), \sigma_a, \sigma_m) \\ &= \log L_1(W, \sigma_a) + \log L_2(W, \sigma_m) \\ &= \log p(Y_a | f^W(x), \sigma_a) + \log p(y_1, y_2 | f^W(x), \sigma_m) \\ &= -\frac{1}{2\sigma_a^2} L_a(W) - \log \sigma_a - \frac{1}{\sigma_m^2} L_m(W) - \log \sigma_m \end{aligned} \quad (9)$$

As shown in Equation (10), the gradient descent method can be applied to solve the minimum point $W^*, \sigma_a^*, \sigma_m^*$. It is equivalent to determining the parameters that make Equation (9) reach the maximum value.

$$\begin{aligned} W^*, \sigma_a^*, \sigma_m^* &= \underset{W^*, \sigma_a^*, \sigma_m^*}{\operatorname{argmin}} \left(-\log L(W, \sigma_a, \sigma_m) \right) \\ &= \underset{W^*, \sigma_a^*, \sigma_m^*}{\operatorname{argmin}} \left(\frac{1}{2\sigma_a^2} L_a(W) + \frac{1}{\sigma_m^2} L_m(W) + \log \sigma_a + \log \sigma_m \right) \end{aligned} \quad (10)$$

Where $L_a(W) = \frac{1}{M} \sum_{i=1}^M (y_i - \hat{y}_i)^2$ and $L_m(W) = -\frac{1}{M} \sum_{i=1}^M \sum_{j=1}^2 y_j^{(i)} \log \hat{y}_j^{(i)}$, M is the number of samples involved in each batch of the training process, and y, \hat{y} are the expected output and predicted output of the model, respectively.

As can be seen from Equation (10), the reciprocal of the error variance of the corresponding task prediction output is used as the task weight. Hence, a larger weight is assigned to the item with a smaller error, and vice versa. The logarithm of the error standard deviation is used as a regularization term of the loss function to prevent $\sigma \rightarrow +\infty$ during the training process.

4. Experiments and Analysis

To verify the effectiveness of the proposed method, the present study undertakes four experiments as follows: training a steganalysis model for stego images with specific embedding rates based on TL, training a steganalysis model on CNN by mixing the cover images and stego images with various embedding rates, training a steganalysis model using the MTL with random initialization parameters, and training a steganalysis model based on the TL and MTL. Finally, the detection performances of these four experiments on the test set are compared and analysed.

4.1. Dataset

The experiments are performed on the standard Bossbase v1.01 dataset. Using 10,000 original images as the cover images, the adaptive image steganographic algorithm wavelet obtained weights [18] and universal wavelet relative distortion (S-UNIWARD) [19] are applied to generate stego image datasets with embedding rates of 0.4 bpp, 0.3 bpp, 0.2 bpp, and 0.1 bpp. Each dataset is divided into a training set, validation set, and test set according to 60%, 20%, and 20%, and fixed division is adopted for all experiments.

4.2. Evaluation Indicators

The stego images are positive class, and the cover images are negative class. Suppose the numbers of cover images and stego images are C and S , and the numbers of correctly detected cover images and stego images are TN and TP , respectively. The present study uses the following evaluation indicators to assess the performance of the steganalysis model:

- True positive rate (TPR) is the proportion of stego images that were correctly identified, and $TPR = TP / S$.
- True negative rate (TNR) is the proportion of cover images that were correctly identified, and $TNR = TN / C$.
- Accuracy is the proportion of correctly classified samples in all samples, and $Accuracy = (TP + TN) / (C + S)$.

4.3. Experiment 1: Training a Steganalysis Model for Stego Images with Specific Embedding Rates based on TL

This experiment is conducted by using the steganalysis method based on CNN and TL which was proposed by Qian et al. [9], and the results were used as the benchmark for the evaluation of effect. The CNN model was proposed by Xu et al. [5], and it consists of a pre-processing layer, five convolutional modules, a fully connected layer, and a softmax classifier. The pre-processing layer uses a high-pass filter (KV kernel) to convolute the input image to obtain a residual image. The residual image goes through the five convolutional modules (referred to as M1-M5) and a fully connected layer and finally outputs the predicted probabilities from the softmax classifier. M1 is composed of a convolutional layer (the kernel size is 5×5 , kernel number is 8, and stride is 1), absolute layer, batch normalization (BN) layer, non-linear activate layer (TanH), and average pooling layer (the size is 5×5 and stride is 2). M2 is composed of a convolutional layer (the kernel size is 1×1 , kernel number is 16, and stride is 1), BN layer, non-linear activate layer (TanH), and average pooling layer (the size is 5×5 and stride is 2). M3 and M4 are respectively composed of a convolutional layer (the kernel size is 5×5 and stride is 2),

BN layer, non-linear activate layer (Relu), average pooling layer (the size is 5×5 and stride is 2), and the kernel numbers of M3 and M4 are 32 and 64, respectively. M5 is composed of a convolutional layer (the kernel size is 1×1 , kernel number is 128, and stride is 1), BN layer, non-linear activate layer (Relu), and a global average pooling layer. Each input image is abstracted into a 128-d feature vector. Then, the fully connected layer and softmax classifier are examined, and the probability prediction value corresponding to the category to which the image belonged is obtained.

The steganalysis model is trained by using the cover images and stego images with embedding rates of 0.4 bpp, and all parameters are initialized based on a mean of 0.0 and a variance of 0.01. The initial learning rate is 0.001, with a decay of 0.95 per 50 epochs. The moving average decay is set to 0.95. Each batch contains 100 images (randomly selected from the dataset, not in pairs). Then, the steganalysis model for the stego images with embedding rates of 0.3 bpp, 0.2 bpp, and 0.1 bpp sequentially use TL.

4.4. Experiment 2: Training a Steganalysis Model on CNN by Mixing the Stego Images with Various Embedding Rates

The structure and hyper-parameter settings used in this experiment are the same as those used in Experiment 1. The difference is that the training set is a mixture of stego images with various embedding rates and cover images.

Based on the division of the dataset in Section 4.1, there are 6,000 cover images and $6,000 \times 4 = 24,000$ stego images in the training set. According to the experiment, TPR is very high (close to 1), and TNR is very low (approximately 20%), because the numbers of cover images and stego images were not balanced. Thus, the present study selects 1,500 stego images from each stego image dataset with an embedding rate between 0.1 and 0.4 bpp, so that the numbers of cover images and stego images for training are the same.

4.5. Experiment 3: Training a Steganalysis Model using MTL with Random Initialization Parameters

The structure, hyper-parameter settings, and training set used in this experiment are the same as those used in Experiment 2. The only difference is that the output of M5 adds a fully connected layer branch to detect the embedding degree of secret information in an image. The initial values of σ_a and σ_m are set to 1, and the remaining parameters are randomly initialized. Equation (10) is used as the optimization goal, and the back-propagation algorithm is applied to train the model. On the WOW dataset and S-UNIWARD dataset, the curves of σ_a , σ_m and the scaled loss term during the training process are shown in Figures 5 and 6. $loss_aux$ and $loss_main$ represent the first and second term of Equation (10).

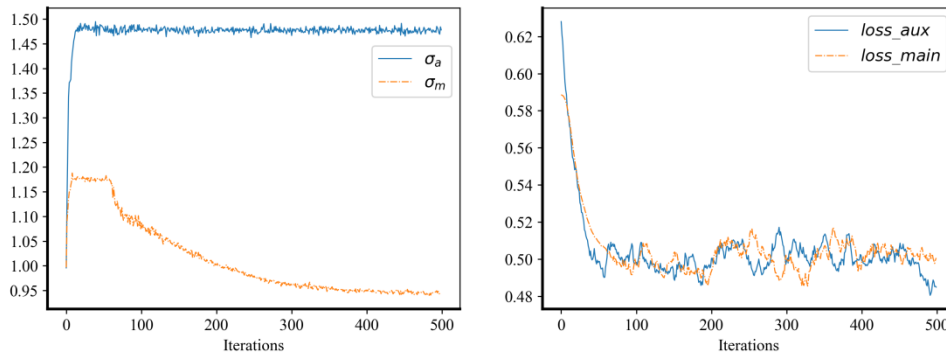


Figure 5. Standard deviation of error and scaled loss term of Experiment 3 on WOW dataset

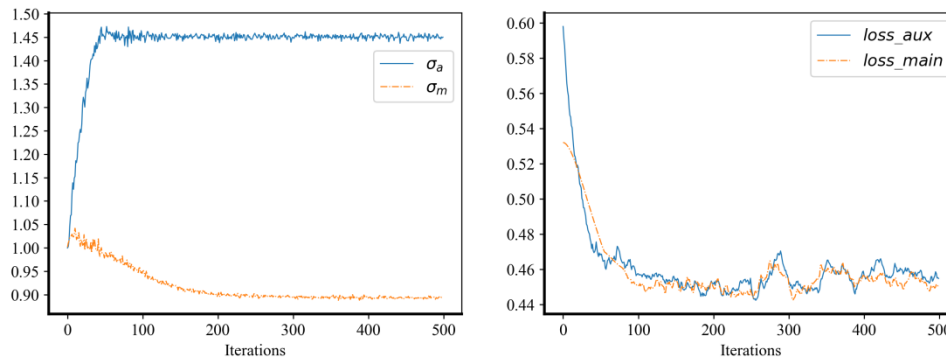


Figure 6. Standard deviation of error and scaled loss term of Experiment 3 on S-UNIWARD dataset

4.6. Experiment 4: Training a Steganalysis Model based on TL and MTL

The structure, hyper-parameter setting, and training set used in this experiment were the same as those used in Experiment 3. As described in Section 3.1, the parameter values of the model trained with 0.4 bpp stego images in Experiment 1 were transferred to the model based on MTL. Its initial values and the rest of the parameters were randomly initialized, with Equation (10) used as the optimization goal, and the back-propagation algorithm was applied to train the model. On the WOW dataset and S-UNIWARD dataset, the curves of σ_a , σ_m and scaled loss term during the training process are shown in Figures 7 and 8. $loss_aux$ and $loss_main$ represent the first and second terms of Equation (10), respectively.

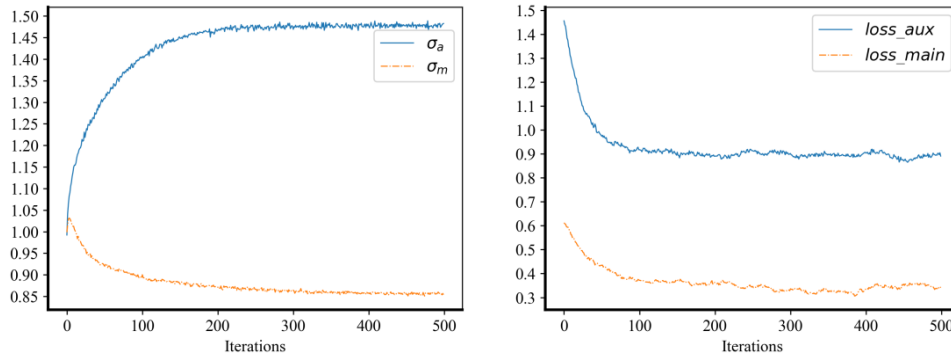


Figure 7. Standard deviation of error and scaled loss term of Experiment 4 on WOW dataset

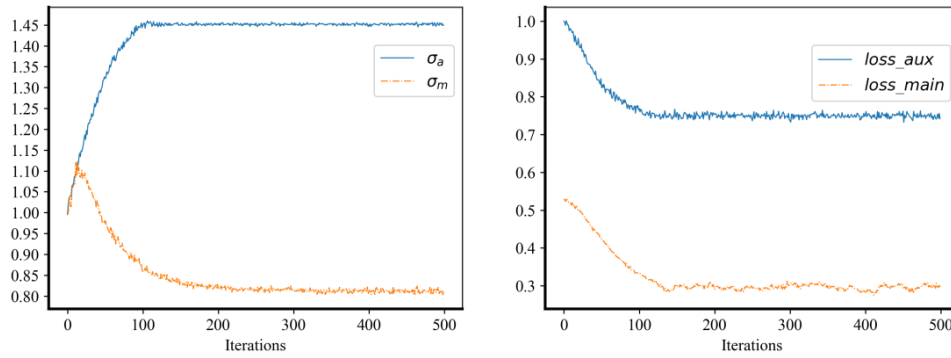


Figure 8. Standard deviation of error and scaled loss term of Experiment 4 on S-UNIWARD dataset

4.7. Comparative Analysis of Experimental Results

As shown in Figures 9 and 10, the proposed method achieved better results from the steganalysis of the stego images with multi embedding rates on the dataset generated by different steganographic algorithms.

In Experiment 2, the cover images and stego images with various embedding rates were mixed to train the steganalysis model. The down-sampling of the stego images was performed to eliminate the negative influence of the imbalance of the number of cover/stego images on the detection performance of the model. Compared with Experiment 1, the detection accuracy was improved.

Experiment 3 introduced the error variance to adjust the weight of each task, and it effectively eliminated the dimensional differences between tasks, which made the scaled loss term almost identical. The additional auxiliary task used the embedding degree of the secret information to help the model capture the effective and distinguishable feature expression in the cover images and stego images with various embedding rates. The method effectively improved the deficiency of artificial adjustment weights and improved the training efficiency and detection accuracy of the model.

In contrast with Experiment 3, the error variance of the main task of Experiment 4 was reduced, the error variance of the auxiliary task remained unchanged, and the scaled loss term of the main task decreased, and yet the scaled loss term of the auxiliary task increased. This is because the priori information obtained from the main task in the pre-trained model helped improve the detection performance of the main task, which proved the effectiveness of the proposed method.

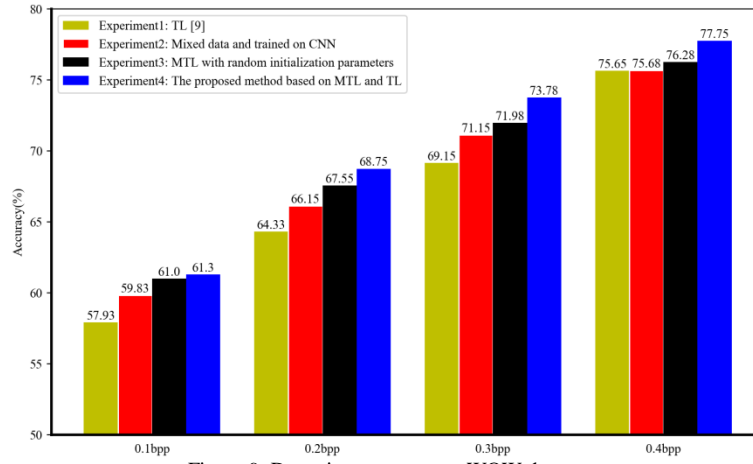


Figure 9. Detection accuracy on WOW dataset

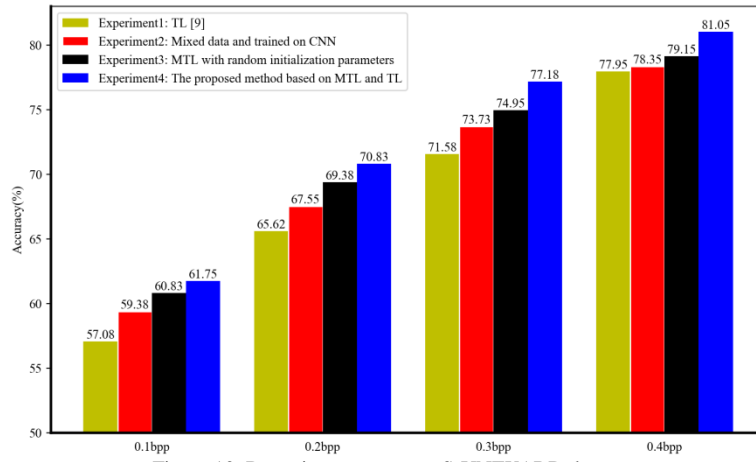


Figure 10. Detection accuracy on S-UNIWARD dataset

5. Conclusions and Outlook

The present study combined TL and MTL and proposed an image steganalysis method for a specific steganographic algorithm and unknown embedding rates. The proposed method used stego images with high embedding rates to pre-train a steganalysis model, constructed a steganalysis model based on MTL, and then transferred the parameter values of the pre-trained model as its initial values. The parameters were further fine-tuned on a training set consisting of cover images and stego images with various embedding rates. The objective function was designed based on task uncertainty and maximum likelihood estimation, which dynamically adjusted the weight of each sub-task during the training process. This effectively reduced the consumption of artificial adjustment weights on time and computing resources and improved the training efficiency of the model. The experimental results showed that the proposed method could better extract the common features of images with various embedding rates. It performed steganalysis well on images with unknown embedding rates and demonstrated improved generalization ability. Based on the present study, the blind detection method of image steganalysis for multi steganographic algorithms and unknown embedding rates will be studied further in the future.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61973103, 61751304, and 61603366) and the Henan Province Central Plains Thousand Talents Plan: Top Young Talents.

References

1. T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by Subtractive Pixel Adjacency Matrix," *IEEE Transactions on Information Forensics and Security*, Vol. 5, No. 2, pp. 215-224, June 2010
2. J. Fridrich and J. Kodovsky, "Rich Models for Steganalysis of Digital Images," *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 3, pp. 868-882, 2012

3. B. Li, Z. Li, and S. Zhou, "New Steganalytic Features for Spatial Image Steganography based on Derivative Filters and Threshold LBP Operator," *IEEE Transactions on Information Forensics and Security*, Vol. 13, No. 5, pp. 1242-1257, May 2018
4. Y. Qian, J. Dong, and W. Wang, "Deep Learning for Steganalysis via Convolutional Neural Networks," in *Proceedings of Media Watermarking, Security, and Forensics 2015*, San Francisco, USA, February 2015
5. G. Xu, H. Wu, and Y. Shi, "Structural Design of Convolutional Neural Networks for Steganalysis," *IEEE Signal Processing Letters*, Vol. 23, No. 5, pp. 708-712, May 2016
6. Y. Yuan, L. Wei, and B. Feng, "Steganalysis with CNN using Multi-Channels Filtered Residuals," in *Proceedings of 3rd International Conference on Cloud Computing and Security (ICCCS 2017)*, pp. 110-120, Nanjing, China, June 2017
7. X. Yu, H. Tan, and H. Liang, "A Multi-Task Learning CNN for Image Steganalysis," in *Proceedings of 10th IEEE International Workshop on Information Forensics and Security (WIFS 2018)*, Hong Kong, China, December 2018
8. Y. Zhang and Q. Yang, "An Overview of Multi-Task Learning," *National Science Review*, Vol. 5, No. 1, pp. 34-47, 2018
9. Y. Qian, D. Jing, and W. Wei, "Learning and Transferring Representations for Image Steganalysis using Convolutional Neural Network," in *Proceedings of 23rd IEEE International Conference on Image Processing (ICIP 2016)*, pp. 2752-2756, Phoenix, USA, September 2016
10. S. J. Pan and Y. Qiang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 10, pp. 1345-1359, 2010
11. G. Cancelli, G. Doerr, and M. Barni, "A Comparative Study of ± 1 Steganalyzers," in *Proceedings of 2008 IEEE 10th Workshop on Multimedia Signal Processing (MMSP 2008)*, pp. 791-796, Cairns, Australia, October 2008
12. A. Kendall, Y. Gal, and R. Cipolla, "Multi-Task Learning using Uncertainty to Weigh Losses for Scene Geometry and Semantics," in *Proceedings of 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, pp. 7482-7491, June 2018
13. A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision," in *Proceedings of 31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 5575-5585, Long Beach, USA, December 2017
14. J. Yosinski, J. Clune, and Y. Bengio, "How Transferable are Features in Deep Neural Networks," in *Proceedings of 28th Annual Conference on Neural Information Processing Systems 2014 (NIPS 2014)*, pp. 3320-3328, Montreal, Canada, December 2014
15. S. Ren, K. He, and R. Girshick, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, pp. 1137-1149, 2017
16. R. Collobert and J. Weston, "A Unified Architecture for Natural Language Processing," in *Proceedings of 25th International Conference on Machine Learning (ICML 2008)*, pp. 160-167, Helsinki, Finland, July 2008
17. J. Huang, J. Li, and Y. Dong, "Cross-Language Knowledge Transfer using Multilingual Deep Neural Network with Shared Hidden Layers," in *Proceedings of 2013 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, pp. 7304-7308, Vancouver, Canada, May 2013
18. V. Holub and J. Fridrich, "Designing Steganographic Distortion using Directional Filters," in *Proceedings of 2012 IEEE International Workshop on Information Forensics and Security (WIFS 2012)*, pp. 234-239, Tenerife, Spain, December 2012
19. V. Holub and J. Fridrich, "Digital Image Steganography using Universal Distortion," in *Proceedings of 1st ACM Workshop on Information Hiding and Multimedia Security (IH and MMSec 2013)*, pp. 59-68, Montpellier, France, June 2013