Steganographer Detection via Multi-Scale Embedding Probability Estimation

SHENG-HUA ZHONG*, Shenzhen University, China
YUANTIAN WANG*, Nanjing University, China
TONGWEI REN†, Nanjing University, China
MINGJIE ZHENG, Shenzhen University, China
YAN LIU, Hong Kong Polytechnic University, China
GANGSHAN WU, Nanjing University, China

Steganographer detection aims to identify the guilty user, who utilizes steganographic methods to hide secret information in the spread multimedia data, especially image data, from a large amount of innocent users on the social networks. True embedding probability map illustrates the probability distribution of embedding secret information in the corresponding images by specific steganographic methods and settings, which has been successfully used as the guidance for content-adaptive steganographic and steganalytic methods. Unfortunately, in real-world situation, the detailed steganographic settings adopted by the guilty user cannot be known in advance. It thus becomes necessary to propose an automatic embedding probability estimation method. In this paper, we propose a novel content-adaptive steganographer detection method via embedding probability estimation. The embedding probability estimation is firstly formulated as a learning-based saliency detection problem and the multi-scale estimated map is then integrated into the CNN to extract steganalytic features. Finally, the guilty user is detected via an efficient Gaussian vote method with the extracted steganalytic features. The experimental results prove that the proposed method is superior to the state-of-the-art methods in both spatial and frequency domains.

CCS Concepts: • **Computing methodologies** → **Image representations**; • **Security and privacy**;

Additional Key Words and Phrases: Steganographer detection, embedding probability estimation, steganalytic feature extraction, Gaussian vote, multimedia security

ACM Reference Format:
Sheng-hua Zhong, Yuantian Wang, Tongwei Ren, Mingjie Zheng, Yan Liu, and Gangshan Wu. 2019. Steganographer Detection via Multi-Scale Embedding Probability Estimation. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1, Article 1 (January 2019), 26 pages. https://doi.org/10.1145/3352691

*Both authors contributed equally to this paper.
†T. Ren is the corresponding author of this paper.

Authors' addresses: Sheng-hua Zhong, csshzhong@szu.edu.cn, Shenzhen University, College of Computer Science and Software Engineering, Shenzhen, Guangdong, China; Yuantian Wang, wangyt@smail.nju.edu.cn, wangyt@szu.edu.cn, Nanjing University, State Key Laboratory for Novel Software Technology, Nanjing, Jiangsu, China; Tongwei Ren, Nanjing University, State Key Laboratory for Novel Software Technology, Nanjing, Jiangsu, China, rentw@nju.edu.cn; Mingjie Zheng, Shenzhen University, College of Computer Science and Software Engineering, Shenzhen, Guangdong, China, zhengmingjie@email.szu.edu.cn; Yan Liu, Hong Kong Polytechnic University, Department of Computing, Hong Kong, China, csyliu@comp.polyu.edu.hk; Gangshan Wu, Nanjing University, State Key Laboratory for Novel Software Technology, Nanjing, Jiangsu, China, gswu@nju.edu.cn.

1551-6857/2019/1-ART1 $15.00
https://doi.org/10.1145/3352691

1 INTRODUCTION

Large-scale multimedia data [36, 55, 56], especially image data, is generated every day from social networks, which brings great challenges in information security [39, 46, 58]. One challenge in multimedia security is to locate the guilty users [61], also named as steganographers [21] or culprit actors [30], who try to hide confidential messages with steganographic methods [34, 60, 63] in the spread images, among many innocent users. The solution to this problem is steganographer detection, which abstracts the user as the set of steganalytic features extracted from each image spread by the corresponding user, and identifies the guilty one according to the divergence from the other users.

Compared with steganalysis [13, 53], which is to classify images with the assumption of specific steganographic settings, steganographer detection is performed under the conditions of unknownness of the detailed steganographic methods and payloads, and large amount of involved users and images. From the view of the methodology, steganalysis and steganographer detection are also different. The dimension of the extracted features will not have a big impact on the performance of steganalysis, and ensemble methods have been widely used in steganalysis owing to their capability of working efficiently with high-dimensional feature spaces. But steganographer detection relies on the distribution of all extracted features from each user rather than the extracted feature from each image. Hence, in the task of steganographer detection, the dimension of features does matter.

The existing steganographer detection methods can be roughly divided into three steps, namely extracting steganalytic features from input images of all the users, calculating the distance between each pair of user according to the extracted steganalytic features, and identifying the guilty user from the comparison of user distance. According to the steganalytic feature extraction strategies, the existing methods can be classified into two categories, namely rich model based methods [24, 25, 29] and learning-based methods [61, 62]. According to the guilty user identification strategies, the existing methods can also be classified into two different categories, namely clustering based methods [24, 29, 61] and outlier detection based methods [25]. However, there are two main challenges in the current steganographer detection methods. First, steganalytic features are extracted equally from all regions in an image, despite the density discrepancy of embedding messages according to the image content. Second, owing to the user-to-user comparison, the time cost of detecting guilty user from the extracted steganalytic features, especially the step of user distance calculation, will increase violently along with the user expansion. As a result, the existing guilty user detection strategies are prohibitive in the real-world applications, which may contain thousands, even millions, of users.

A possible solution to the first challenge in the steganographer detection methods is to integrate embedding probability maps into feature extraction. Embedding probability maps illustrate the probability of embedding secret information in the corresponding position of images, which have been widely used in the content-adaptive steganographic [10, 14, 15] and steganalytic methods [5, 47, 54]. Unfortunately, most of these methods directly use the true embedding probability maps, which are generated by the specific steganographic strategies. However, due to the uncertainty of the specific steganographic strategies used by guilty users in the real-world situation, true embedding probability maps cannot be known in advance for steganographer detection task. To overcome this difficulty, it is necessary to estimate the embedding probability maps automatically. An example of the cover image and the corresponding true embedding probability maps generated by five current content-adaptive steganographic methods, namely HUGO-BD [10], WOW [14], S-UNIWARD [15], HILL [28] and MiPOD [45], is shown in Fig. 1. Owing to the different steganographic strategies, the true embedding probability maps have a degree of difference in appearance. However, they share some common properties, including owning high probability in contours and texture regions,

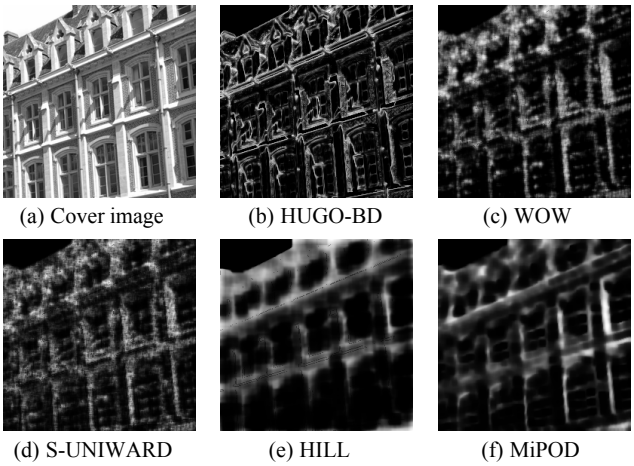


Fig. 1. An example of cover image and the corresponding normalized true probability maps of different content-adaptive steganographic methods with 0.4bpp payload. (a) Cover image. (b) HUGO-BD [10]. (c) WOW [14]. (d) S-UNIWARD [15]. (e) HILL [28]. (f) MiPOD [45].

and owning low probability in smooth and flat regions, which shows the preference of the current content-adaptive steganographic methods to embedding secret information. This indicates the predictability of embedding probability maps.

The second challenge in the steganographer detection methods, namely the efficiency of detecting guilty user from the extracted steganalytic features, relates to reducing the time complexity of the detection strategy. It can be solved by utilizing the user-to-mean comparison instead of the user-to-user comparison, which means to firstly represent the mean behaviors of all users, then select the user with the most deviation against the mean behaviors as the guilty one.

In this paper, we propose the first content-adaptive steganographer detection method, namely Multi-scale Embedding Probability Estimation based Steganographer Detection (MEPESD), which includes embedding probability estimation, steganalytic feature extraction and guilty user detection. Here, the "content-adaptive" means to adaptively adjust the steganalytic feature extraction according to the image content. The contributions of our work mainly include:

- We are the first to propose the content-adaptive steganographer detection method, which integrates estimated embedding probability maps into steganalytic feature extraction.
- We are the first to estimate embedding probability maps with a learning-based method, which is independent of the specific steganographic methods and settings. The estimated maps are proved to be generic to different steganographic strategies and payloads in our experiments.
- We propose a novel multi-scale integration method to enhance the content-adaptive steganalytic features in deep learning architecture.
- We use Gaussian vote to detect the guilty user from the extracted steganalytic features, which is effective and efficient.
- We validate the proposed method on the standard dataset BOSSbase ver 1.01 [2]. It shows that our method is superior to the state-of-the-art methods in both spatial and frequency domains.

The rest of the paper is organized as follows. In Section 2, we briefly review the framework of the prevailing steganographer detection method and the strategies of utilizing the knowledge of the embedding probability map in the existing steganalytic methods. The details of the proposed

MEPESD method is described in Section 3, and the experimental results and analysis are shown in Section 4. Finally, we conclude our proposed work and arrange the future work in Section 5.

2 RELATED WORK

In this section, we first review the state-of-the-art results in steganographer detection. Until now, steganographer detection that utilizes the knowledge of the embedding probability map has never been developed. Thus, we discuss the utilization of the embedding probability map in the existing steganalytic methods.

2.1 Steganographer detection methods

Ker [21] first defined the steganographer detection problem as the confrontation between the steganographer and the warden, in which the steganographer uses batch steganography [20] to allocate steganography payload between a large amount of covers, while the warden applies pooled steganalysis [20] to compare the steganalysis outputs of each image from the steganographer to determine whether the payload is transmitted or not. Ker *et al.* [24] further formulated the steganographer detection task as the clustering problem, which utilizes the traditional 274-dimensional steganalysis features PF-274 [40] to calculate the distance between users via Maximum Mean Discrepancy (MMD) [44], and applies hierarchical clustering to classify the guilty cluster and innocent cluster. In [22, 23, 25], Ker *et al.* formulated the steganographer detection task as the outlier detection problem, and used the Local Outlier Factor (LOF) [4] to rank the users by the deviation from the rest with the MMD results. Li *et al.* [30] extracted 250-D steganalytic features from the high-order joint matrices of images, and used clustering ensembles based on the majority voting strategy to select the suspicious steganographers. Li *et al.* [29] further improved their work by using DCT blocks with higher embedding probability to reconstruct the images, and extracting features from a reduced PEV feature set. However, these rich model based methods were evaluated on unpublished datasets, which leads to the lack of convincing comparisons with other methods.

Several works used learning-based methods to extract steganalytic features instead of the hand-crafted feature sets. Zheng *et al.* [61] were the first to extract 512-D features from deep residual networks. Zheng *et al.* [62] further improved their work by training the multi-class deep residual networks as the feature extractor. Both the above two methods use MMD to calculate user distances and detected guilty user with hierarchical clustering as mentioned in [24]. Different from the existing methods, our method is the first learning-based content-adaptive steganographer detection method, which automatically predicts the embedding probability maps and integrates the multi-scale estimated maps with the convolutional features to enhance the difference between cover features and stego features, and applies a more efficient Gaussian vote strategy to detect guilty user from the extracted features.

Recently, Li *et al.* [32] proposed a new definition of steganographer as the user who behaves differently from innocent users, and formulated the steganographer detection problem as a behavior analysis task. Here, the word "behavior" refers to the intrinsic connection of images transferred by users, including users' interests, habits or image contents.

2.2 Utilization of the embedding probability map

Carnein *et al.* [5] were the first to use weighted stego-image steganalysis to detect stego images generated with public-key steganographic methods, *e.g.*, Wet Paper Codes (WPC) [12], which calculates the embedding probability with the estimated embedding rate as 0.1 according to the experimental results, and proposes a Weighted Stego-Image (WS) steganalysis to detect the elements with high probability in the image, rather than the whole image. Similar adaptive steganalytic methods [6, 9] were proposed to attack the traditional non-adaptive Least Significant Bit (LSB) based

steganographic methods, but they are not effective against modern content-adaptive embedding schemes [14, 15].

An effective solution to steganalysis against content-adaptive steganographic methods is to incorporate embedding probability maps into rich model strategies [7, 8, 33, 47, 48]. Tang *et al.* [47] proposed the steganalytic method against WOW [14] by extracting the steganalytic features from the complex textural regions defined by WOW. Tang *et al.* [48] further improved their previous work by incorporating the steganalytic features with the corresponding weights, which were assigned according to the embedding probability estimated with optimal simulator [11] and re-embedding random experiments. Optimal simulator is designed under the framework of minimizing the distortion function used in the existing content-adaptive steganographic methods. Re-embedding random experiments utilize randomized embedding to simulate the steganographic methods. Denemark *et al.* [8] proposed a variant of the spatial rich model, namely maxSRM, by incorporating the maximum of the true embedding probability generated by the corresponding steganographic methods into four-dimensional neighboring noise residuals. They further extended the proposed method in JPEG images as proposed in [7]. Liao *et al.* [33] proposed a steganalytic method for color images by locating the suspected pixels in each color channel with the corresponding true embedding probabilities generated by HILL [28], and extracting spatial rich model features from the suspected pixels rather than the whole images.

Owing to the advantage in classification, Convolutional neural network (CNN) is another alternative approach to steganalytic methods [18, 54, 57]. And the embedding probability map is also integrated into the CNN model. Yang *et al.* [54] proposed the first CNN based steganalytic method using the knowledge of the embedding probability map, namely maxCNN, by assigning weights to features during the forward propagation step of Xu's CNN architecture [53], where the weight maps are generated from the max-pooling of the embedding probability maps estimated with optimal simulator [11]. Ye *et al.* [57] used an element-wise summation of the true embedding probability maps and the feature maps generated from the first convolutional layer in the proposed CNN framework. Hu *et al.* [18] selected the regions with the maximal sum of the embedding probabilities estimated with optimal simulator [11] as the input of the proposed CNN based steganalytic method.

Compared with the state-of-the-art utilization strategies of the knowledge of the embedding probability map in steganalytic methods, there are two main advantages in our proposed method. For one thing, the embedding probability maps used in the existing strategies are true probability maps or estimated with distortion function simulated from the specific steganographic methods, which limits the generalization of these methods to attack unknown steganographic methods. While our proposed embedding probability maps are estimated via the learning-based method, which are independent of the detailed steganographic methods. For another, the proposed integration strategy combines multi-scale embedding probability estimation with the convolutional features, which extracts more discriminative steganalytic features than the existing integration strategies.

3 PROPOSED METHOD

Figure 2 illustrates the framework of the proposed Multi-scale Embedding Probability Estimation based Steganographer Detection (MEPESD). Specifically, our method includes three steps. First, we propose a novel learning-based embedding probability estimation method via grid-like CNN network NLDF [37]. Then, the multi-scale estimated embedding probability map is integrated into the deep convolutional neural network to extract the steganalytic features of each image from users. Finally, a novel and efficient Gaussian vote method is utilized to identify the guilty user from innocent users with the extracted steganalytic features. The detailed discussion of these three steps is described as follows.

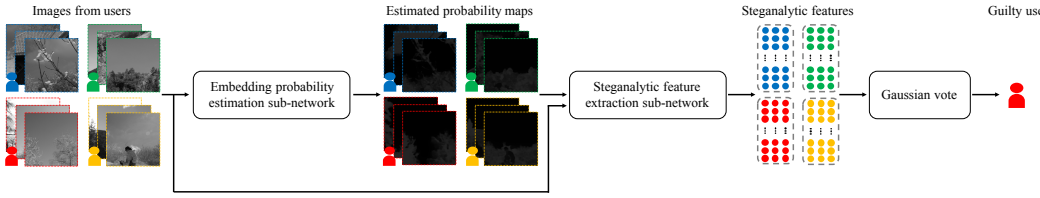
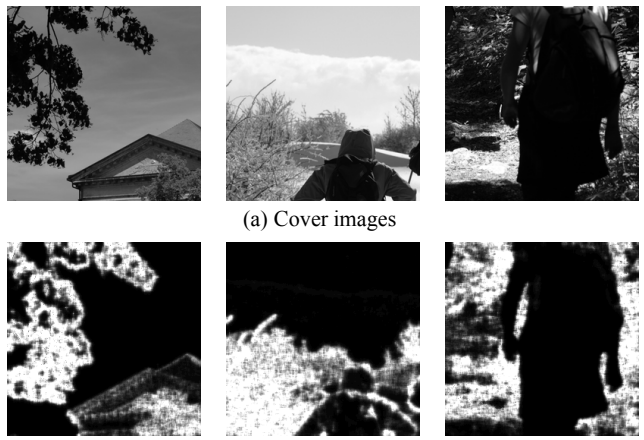


Fig. 2. The flowchart of the proposed steganographer detection method. First, we estimate the embedding probability maps of images from all the users with the embedding probability estimation sub-network. Then, we apply both images and the corresponding estimated probability maps as the input of the steganalytic feature extraction sub-network to extract steganalytic features of each user. Finally, we identify the guilty user via the Gaussian vote from the extracted steganalytic features.



(a) Cover images

(b) Embedding probability maps

Fig. 3. Examples of cover images and the corresponding normalized embedding probability maps. (a) Cover images. (b) Normalized embedding probability maps via S-UNIWARD [15] with 0.4bpp payload. Specifically, higher values in probability maps represent the higher probability to hide information.

3.1 Embedding Probability Estimation

Content-adaptive steganography executes the embedding process primarily in those regions where they are less detectable, such as texture areas, while keeping those smooth and flat regions as they are [14, 15]. Probability map illustrates the probability distributions of embedding messages in images, which provides guidance information for content-adaptive steganography. Therefore, higher values in embedding probability map mean the locations in image are inherently more vulnerable to hiding information. As is shown in Fig. 3, for different image contents, *i.e.*, building, plants and forest, regions with complex texture or obvious contours usually have higher embedding probability values. In human visual system, these kinds of regions are visually more salient and attractive [3]. Therefore, although the pixels with similar embedding probabilities may not belong to the same object or have an obvious semantic meaning, those pixels are similar with respect to some characteristics or computed properties, and we believe these properties can be captured.

From the above observation, we formulate the embedding probability estimation as a learning-based saliency detection task [26, 59]. Figure 4 shows an overview of the proposed probability

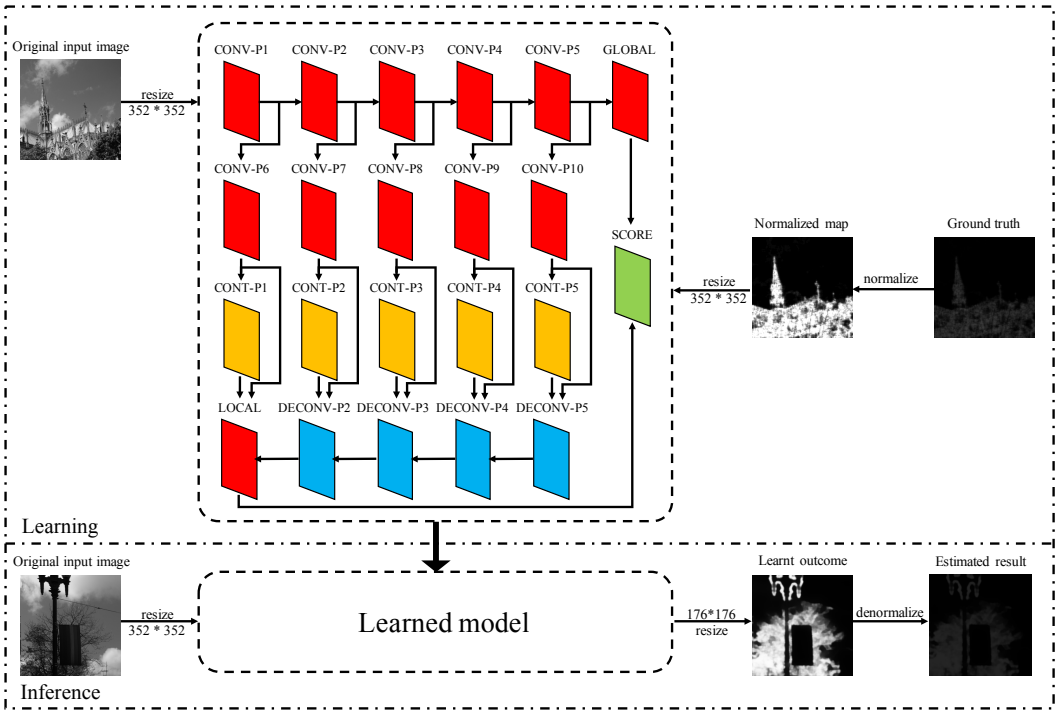


Fig. 4. An overview of the proposed embedding probability estimation method. The network architecture is inspired by the NLDF model [37]. Specifically, we first normalize the ground truth of real probability map in learning stage, and denormalize the learnt outcome into the estimated result in inference stage.

estimation method. The architecture of the proposed estimation network is inspired by the saliency detection model NLDF [37], which has never been used to estimate probability maps before. It is organized as a 4×5 grid-like architecture, including 10 convolution blocks, five contrast blocks, four deconvolution blocks, one local block, one global block and one score block. The local block constructs the local feature maps by combining multi-layer convolutional features and contrast features. The global block extracts global context of the cover image from block CONV-P1 to CONV-P5. The score block fuses both local and global features to compute the embedding probability. The details of each block are shown in the supplemental material. The loss L^{NLDF} consists of two terms, including a cross entropy term between the ground truth and estimated map and a boundary loss term:

$$L^{NLDF} = \sum_{j} (\lambda_j H(\Omega_j, \hat{\Omega}_j) + \gamma_j IoU(C_j, \hat{C}_j)), \tag{1}$$

where λ_j and γ_j are the positive weighting constants; $H(\Omega_j, \hat{\Omega}_j)$ denotes the cross entropy loss between the j^{th} segment in ground truth and estimated map; $IoU(C_j, \hat{C}_j)$ denotes the boundary loss between the boundary of Ω_j and $\hat{\Omega}_j$, which can be calculated as:

$$IoU(C_j, \hat{C}_j) = 1 - \frac{2|C_j \cap \hat{C}_j|}{|C_j| + |\hat{C}_j|}. \tag{2}$$

The saliency map generated from saliency detection task usually ranges from 0 to 1. However, different from saliency map, the embedding probability map usually has a narrower range of value.

As a result, directly applying the original NLDF model [37] in embedding probability estimation task may not reach the ideal performance. Thus, we make several modifications on NLDF model, which are discussed as follows.

Firstly, in order to expand the dynamic range of data to facilitate learning probability map, we normalize it as follows:

$$N_g(x,y) = \begin{cases} 1 & for \quad P_g(x,y) > \omega, \\ \frac{P_g(x,y)}{\omega} & for \quad P_g(x,y) \leq \omega, \end{cases} \tag{3}$$

where $P_g(x,y)$ denotes the probability value of each pixel in the probability map, which ranges from 0 to 1; to avoid the instability of the maximum probability value of real probability maps, a 99th percentile for probability value is applied instead of the maximum probability value, and ω denotes the mean of 99% probability value of each probability map in training set, which equals 0.15 in the experiments; and $N_g(x,y)$ denotes the probability value of each pixel in the normalized result, which ranges from 0 to 1.

Then, the learning outcome is denormalized into the estimated result of probability map as follows:

$$P_l(x,y) = \omega N_l(x,y), \tag{4}$$

where $N_l(x,y)$ denotes the probability value of the pixel (x,y) in the learning outcome, which ranges from 0 to 1; ω is the mean of 99% probability value mentioned in Eq. 3; and $P_l(x,y)$ denotes the probability value of the pixel (x,y) in the estimated result, which ranges from 0 to ω.

3.2 Steganalytic Feature Extraction

In this sub-section, we propose a deep convolutional neural network with multi-scale integration method to combine the estimated probability maps and the extracted deep feature maps together to extract steganalytic features. As illustrated in Fig. 5, given an image, we first estimate its probability map as mentioned in subsection 3.1. Then, we apply the image and the corresponding estimated probability map as the input of the steganalytic feature extraction sub-network. The main contribution of the proposed architecture is the multi-scale probability map integration method, which has never been applied in the existing steganographer detection methods.

This proposed network consists of three parts: the image pre-processing sub-network, the probability map pre-processing sub-network and the steganalytic feature learning sub-network. The details of these three parts are described as follows:

Image pre-processing sub-network. The image pre-processing sub-network contains one High-Passing-Filter layer, namely HPF-S1, which aims at enlarging the high frequency stego signal and suppressing the image content. In HPF-S1, the input image is convoluted as follows:

$$\varphi(I) = I * K, \tag{5}$$

where I denotes the input image; K denotes a 5×5 high-passing kernel as is employed in [42] and [41]; $*$ denotes the convolution operator; $\varphi(I)$ denotes the convolutional result.

Probability map pre-processing sub-network. The probability map pre-processing sub-network contains one High-Passing-Filter layer, namely HPF-S2, and four max-pooling layers, namely POOL-S4 to POOL-S7. Inspired by the combinatorial strategy of image and its corresponding real probability map in image steganalysis task [57], the estimated probability map is convoluted as follows:

$$\phi(P) = 2P * |K|, \tag{6}$$

where P denotes the estimated probability map; K denotes the same 5×5 high-passing kernel as is mentioned above; $|\cdot|$ denotes the absolute operator; $*$ denotes the convolution operator; $\phi(P)$ denotes the convolutional result.

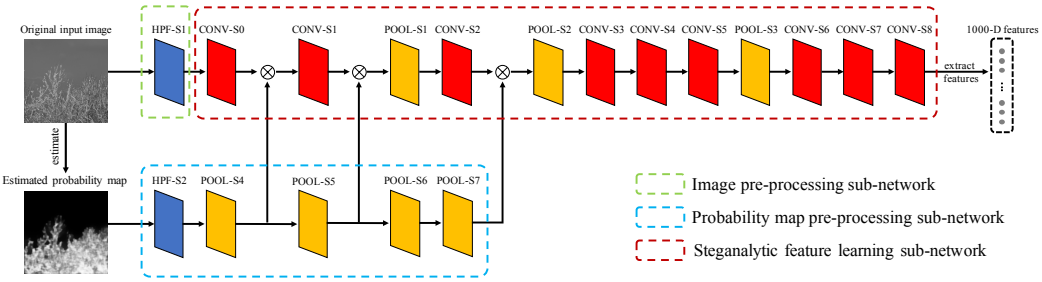


Fig. 5. An overview of the inference stage of the proposed steganalytic feature extraction network. It is composed of three parts: the image pre-processing sub-network, the probability map pre-processing sub-network and the steganalytic feature learning sub-network. The proposed network utilizes the original image as input, and extracts 1000-D steganalytic features as output. Specially, we use a pre-trained network to estimate probability map from the corresponding input image, which is mentioned in subsection 3.1.

In order to generate multi-scale probability maps and make them consistent with the size of the extracted deep feature maps, we downsample the convoluted probability map via max-pooling in POOL-S4 to POOL-S7.

Steganalytic feature learning sub-network.

The architecture of the proposed steganalytic feature learning sub-network is inspired by the pioneer work for feature extraction in natural images classification task [27], which contains six convolutional blocks, namely CONV-S0 to CONV- S5, three fully connected blocks, namely CONV-S6 to CONV-S8, and three max-pooling layers, namely POOL-S1 to POOL-S3. The similar structure has never been used in the existing steganographer detection methods.

Different from the original architecture [27], in order to transfer the estimated probability maps into the convolutional networks, we propose a multi-scale integration method to combine the estimated probability maps and the extracted deep feature maps. The integration includes three combinatorial layers as shown in Fig. 5. In each combinatorial layer, we apply an element-wise multiplication of the feature map and the corresponding probability map.

The details of the proposed steganalytic feature extraction networks are shown in the supplemental material. To train the proposed steganalytic feature extraction networks, the output of CONV-S8 is fed to a two-way softmax which produces a distribution over the two classes labeled as cover and stego. In the inference stage, the learnt model is applied as a feature extractor. A 1000-D feature vector is obtained for each image.

3.3 Guilty User Detection

In this subsection, we formulate the guilty user detection as a Gaussian vote sorting task. The Gaussian distribution value of each user is calculated via the extracted features of all the images from the corresponding user. Then, the Gaussian distribution value of each feature of the corresponding user is compared with the trained value to vote for the guilty user. Finally, the guilty user is detected with the most ballots.

Assume that N_U users spread and share images on social media platforms, one of which is the guilty user who hides secret information in the sharing images, and the rest of which are innocent users. Each user spreads N_I images. And each image can be represented as a N_D dimension feature set via the steganalytic feature extraction method proposed in subsection 3.2. As a conclusion, each user can be represented as a $N_I \times N_D$ dimension feature set.

To detect the guilty user, we first train the mean Gaussian distribution value of each feature in the feature set of training images, which is defined as follows:

$$\rho_{mean}(d) = \frac{1}{N_{TI}} \sum_{ti=1}^{N_{TI}} \frac{1}{\sigma(d)\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x(ti,d)-\mu(d)}{\sigma(d)})^2}, \tag{7}$$

where $\rho_{mean}(d)$ denotes the mean Gaussian distribution value of the d^{th} feature; N_{TI} is the number of images in the training set; $x(ti, d)$ denotes the value of the d^{th} feature in the feature set of ti^{th} image in the training set; $\mu(d)$ and $\sigma(d)$ denote the mean and standard deviation of the d^{th} feature; π and e are the constants.

Next, we calculate the Gaussian distribution value of each feature in the feature set of each user as follows:

$$\rho(u, i, d) = \frac{1}{\sigma(d)\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x(u,i,d)-\mu(d)}{\sigma(d)})^2}, \tag{8}$$

where $\rho(u, i, d)$ denotes the Gaussian distribution value of the d^{th} feature in the feature set of i^{th} image of u^{th} user; $x(u, i, d)$ denotes the value of the d^{th} feature in the feature set of i^{th} image of u^{th} user.

Then, we define the Gaussian vote of each user as the number of features in the feature set of each user larger than the corresponding feature in the mean Gaussian distribution value:

$$v(u) = \sum_{i=1}^{N_I} \sum_{d=1}^{N_D} ||\rho(u, i, d) - \rho_{mean}(d)||, \tag{9}$$

where $v(u)$ denotes the Gaussian vote of the u^{th} user; $||\cdot||$ denotes the operation that equals 1 when $\cdot > 0$, and equals 0 when $\cdot \leq 0$.

Finally, we sort the Gaussian vote of all users, and select the user with the maximum Gaussian vote as the guilty user.

4 EXPERIMENTS

4.1 Dataset and experiment settings

We validated the proposed method on dataset BOSSbase ver 1.01 [2]. BOSSbase ver 1.01 is a standard dataset for steganography and steganalysis tasks, which contains 10,000 grayscale natural images with the size of 512×512. Following the general settings in the existing works [42, 50, 61, 62], in order to increase the experimental data, each image in the original BOSSbase dataset is cropped into four non-overlapping sub-images with the size of 256×256. The training set consists of 20,000 sub-images which are selected randomly. The validation set consists of the rest of the 20,000 sub-images.

In the experiments, the default settings of the proposed method are listed as follows. In the training stage of the embedding probability estimation sub-network, the cover images in the training set, and the corresponding embedding probability maps generated by S-UNIWARD [15] with 0.4bpp payload, are utilized to train the network. The parameters of the network are applied as the default settings in the NLDF network [37].

In the training stage of the steganalytic feature extraction sub-network, the cover images in the training set, the corresponding probability maps estimated by the embedding probability estimation sub-network, and the corresponding stego images generated by S-UNIWARD with 0.4bpp payload, are utilized to train the network. The size of mini-batch is 64 (32 cover-stego pairs). The other parameters are applied as the default settings in AlexNet [27].

Here, we use 0.4bpp payload in the training stage of the above two networks for two reasons. First, 0.4bpp is a relatively high value in the existing works of steganography [15, 28], steganographic analysis [13, 53], and steganographer detection [61, 62]. In this case, the true embedding probability maps would be more differentiated from the background, which is beneficial to the feature learning in embedding probability estimation network. The similar situation happened in the case of steganalytic feature extraction network. It would be easier to learn a model to classify the cover images and stego images in the training stage when using a higher payload, such as 0.4bpp. Second, the models trained with 0.4bpp payload are universal for various test payloads, which ranges from 0.05bpp to 0.4bpp.

In the inference stage of the whole proposed networks, we define 100 users, including one guilty user and 99 innocent users. Each user randomly spread 200 images in the validation set. Images from the innocent users are all cover images without any secret information injection. While the spread images from the guilty user may consist of part or all of the stego images generated by different steganographic methods or settings. The specific settings of the guilty user will be described in each of the following experiments.

The performance of the proposed method and its comparison methods in steganographer detection task is evaluated with True Positive Rate (TPR). All the experiments were conducted on a Tesla K80 GPU, and repeated for 100 times.

4.2 Effectiveness evaluation of embedding probability estimation

In this sub-section, we evaluate the effectiveness of the proposed embedding probability estimation method.

Estimating performance of the proposed embedding probability estimation method. As is illustrated in sub-section 3.1, the embedding probability estimation is formulated as a learning-based saliency detection task. Hence, we evaluate the performance of the proposed embedding probability estimation method via two commonly-used evaluation metrics in saliency detection tasks [17, 19, 37], namely max F_β and Mean Absolute Error (MAE), and a novel metric designed for embedding probability estimation, namely Mean Relative Error (MRE).

We first calculate the Precision-Recall curve of each estimated embedding probability map by binarizing the probability map under probability values ranging from 0 to ω, which equals 0.15 in the experiments, and comparing against the corresponding ground truth. For each pair of precision and recall, the F_β is measured as follows:

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}, \tag{10}$$

where β^2 is a parameter to emphasize precision over recall, and equals 0.3 as suggested in [1]. The max F_β denotes the maximum of the F_β calculated in the Precision-Recall curve.

As is defined in [16], MAE is the average pixel-wise absolute difference between the estimated embedding probability map E and its corresponding ground truth T, which is as follows:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |E(x,y) - T(x,y)|, \tag{11}$$

where W and H denote the width and height of the embedding probability map, respectively.

The maximum of the estimated embedding probability maps is defined as 0.15 in the experiments, which is much smaller than the maximum of general saliency maps as 1. Hence the estimated deviation is more sensitive in embedding probability maps than in saliency maps. We utilize MRE to measure the average pixel-wise relative deviation between the estimated embedding probability

Table 1. Quantitative performance of embedding probability estimation based on saliency detection evaluation metrics.

Method	max F_β	MAE	MRE
EEP-Original	0.04	0.03	15.98%
EEP-ND	**0.99**	**0.01**	**4.19%**

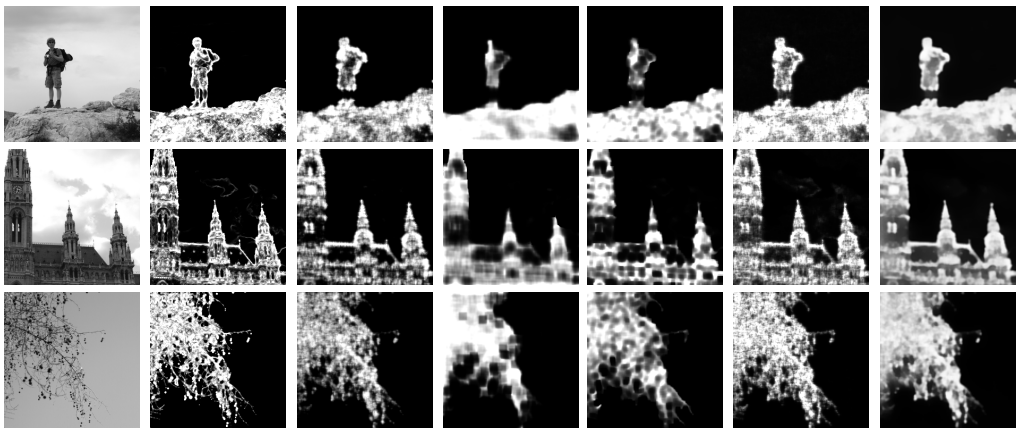


Fig. 6. Examples of cover images and the corresponding normalized embedding probability maps of the steganographic methods with 0.4bpp payload and the estimated results of the proposed EEP-ND. (a) Cover images. (b) HUGO-BD [10]. (c) WOW [14]. (d) HILL [28]. (e) MiPOD [45]. (f) S-UNIWARD [15]. (g) Estimated results of EEP-ND.

map E and its corresponding ground truth T, which is as follows:

$$MRE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \frac{|E(x,y) - T(x,y)|}{T(x,y)}. \tag{12}$$

We compare two strategies of embedding probability estimation, namely EEP-Original and EEP-ND. EEP-Original denotes the strategy of utilizing the original NLDF model [37] to estimate embedding probability maps. EEP-ND denotes the proposed strategy in subsection 3.1, which normalizes the ground truth embedding probability maps as input, and denormalizes the learnt results as output. As is illustrated in Table 1, the proposed EEP-ND model achieves competitive performance in max F_β and MAE, which are better than the reported results [37] on datasets for saliency detection, such as MSRA-B [35], HKU-IS [31] and SOD [38]. Meanwhile, as is shown in the MRE metric, the estimated results have only 4.19% pixel-wise deviation in probability values against ground truth. It means the proposed embedding probability estimation method provides fairly precise estimated results, which can be used as a substitute for true embedding probability maps. Compared with EEP-ND, EEP-Original obtains evidently worse estimation performance, which proves the effectiveness of the processes of normalization and denormalization in the embedding probability estimation method.

We further compare the estimated embedding probability maps with the ones generated by the current content-adaptive steganographic methods, including HUGO-BD [10], WOW [14], HILL [28], MiPOD [45] and S-UNIWARD [15], with 0.4bpp payload. As is shown in Fig. 6, the estimated results

of the proposed EEP-ND look similar to the corresponding true embedding maps from different content-adaptive steganographic methods, despite the different strategies adopted by them, in different image contents, including person, ground, building and plants. It illustrates the wide versatility of the proposed embedding probability estimation method.

Effectiveness evaluation of integrating embedding probability into steganographer detection method. We further compare the steganographer detection results of models integrated with no embedding probability maps (MEPESD-NEP), estimated embedding probability maps (MEPESD-EEP) and true embedding probability maps (MEPESD-TEP) when the guilty user spreads all the images as stego images generated by S-UNIWARD with series of payloads, including 0.05bpp, 0.1bpp, 0.2bpp, 0.3bpp and 0.4bpp. We apply two methods in the step of steganographer detection, namely MMD+AHC [24] and Gaussian vote. MMD+AHC is a commonly-used steganographer detection method in the current works [24, 30, 61, 62], which computes the distance of each pair of users by the Maximum Mean Discrepancy (MMD) based on the extracted steganalytic features, and utilizes the Agglomerative Hierarchical Clustering (AHC) to distinguish the guilty user from the innocent ones based on the user distance. Gaussian vote is the proposed steganographer detection method in this paper, which calculates the Gaussian distribution values of each steganalytic feature in the feature set of each user, and detects guilty user as the user with maximum vote from the Gaussian distribution values.

As is shown in Table 2, the performance of MEPESD-NEP does not exceed that of the other two models, namely MEPESD-EEP and MEPESD-TEP, in all the listed conditions. It demonstrates that the knowledge of embedding probability maps is beneficial to improve the performance of steganographer detection. Besides, the performance of MEPESD-EEP is similar with that of MEPESD-TEP in the same condition. Considering the real-world situation, we cannot obtain the truth probability maps due to the unknownness of the detailed steganographic settings adopted by the guilty user. The comparison results between MEPESD-EEP and MEPESD-TEP proves that the proposed embedding probability estimation method is effective for the steganographer detection task.

Comparing the performance of two steganographer detection methods, we can find that there is no evident difference between the results using MMD+AHC and Gaussian vote when adopting the steganalytic feature extraction models integrated with embedding probability maps, namely MEPESD-EEP and MEPESD-TEP. However, the performance of method using MEPESD-NEP and Gaussian vote is apparently worse than that using MEPESD-NEP and MMD+AHC. It is because the performance of Gaussian vote depends on the discriminability of the extracted steganalytic features. The detailed comparisons and explanations of the proposed steganographer detection method is described in sub-section 4.6.

4.3 Comparisons of different steganographer detection methods

In this sub-section, we compare the proposed Multi-scale Embedding Probability Estimation based Steganographer Detection (MEPESD) with one state-of-the-art steganographer detection method, namely MDNNSD [62], and two baseline methods, namely XuNet_SD and SRMQ1_SD. All the three comparison methods first extract steganalytic features, and then detect the guilty user from the extracted features via MMD+AHC [24]. MDNNSD extracts steganalytic features from the dilated residual networks trained with six classes of images in different payloads. XuNet_SD uses a classic deep learning based steganalytic method, namely XuNet [53], which has been widely applied and verified in the current steganographic and steganalytic methods [43, 49, 51, 57], to extract features. In the experiments, we modify the size of input images as 256×256 to fit the dataset, and set the size of mini-batch as 40 (20 cover-stego pairs). The stego images used in the training stage are generated by S-UNIWARD with 0.4bpp payload. SRMQ1_SD extracts steganalytic features from

Table 2. Effectiveness of embedding probability in the detection performance of the proposed method when the guilty user applies S-UNIWARD [15] with a single payload.

Payload (bpp)	Steganalytic feature extraction	Steganographer detection	TPR (%)
0.05	MEPESD-NEP	MMD+AHC [24]	5
	MEPESD-EEP		49
	MEPESD-TEP		**51**
	MEPESD-NEP	Gaussian vote	2
	MEPESD-EEP		50
	MEPESD-TEP		**51**
0.1	MEPESD-NEP	MMD+AHC [24]	73
	MEPESD-EEP		99
	MEPESD-TEP		**100**
	MEPESD-NEP	Gaussian vote	2
	MEPESD-EEP		**100**
	MEPESD-TEP		**100**
0.2	MEPESD-NEP	MMD+AHC [24]	**100**
	MEPESD-EEP		**100**
	MEPESD-TEP		**100**
	MEPESD-NEP	Gaussian vote	3
	MEPESD-EEP		**100**
	MEPESD-TEP		**100**
0.3	MEPESD-NEP	MMD+AHC [24]	**100**
	MEPESD-EEP		**100**
	MEPESD-TEP		**100**
	MEPESD-NEP	Gaussian vote	4
	MEPESD-EEP		**100**
	MEPESD-TEP		**100**
0.4	MEPESD-NEP	MMD+AHC [24]	**100**
	MEPESD-EEP		**100**
	MEPESD-TEP		**100**
	MEPESD-NEP	Gaussian vote	5
	MEPESD-EEP		**100**
	MEPESD-TEP		**100**

a handcrafted spatial rich model with a single quantization step, namely SRMQ1 [13], which has also been widely verified in many steganographic and steganalytic methods [8, 14, 15, 52]. As is discussed in the related work, we do not compare the proposed method with other rich model based methods, such as [21, 24, 25, 29, 30], because they were neither open source nor evaluated on public datasets.

Feature extraction comparisons for steganalysis. As a common part between steganalysis task and steganographer detection task, the extracted steganalytic features can also be used to classify cover images and stego images. Thus, we evaluate the classification performance of these four methods as steganalytic methods on S-UNIWARD [15] with 0.4bpp payload. Specially, DRNSD is a steganalysis model which is trained from MDNNSD [62] with a binary classifier. Table 3 shows the classification performance of different models. It is shown that SRMQ1 performs the best in these four methods, which is owing to the effectiveness of ensemble classifiers working with high-dimensional feature spaces. The ensemble classifier is able to aggregate predictions from different dimensional subspace of feature space to form a final and better prediction. From these results,

Table 3. Classification performance of different steganographer detection methods when performing the steganalysis task on S-UNIWARD [15] with 0.4bpp payload.

Method	Feature Dimension	TPR (%)
SRMQ1 [13]	12,753	**75.38**
XuNet [53]	128	71.85
DRNSD [62]	320	71.91
MEPESD	1,000	70.17

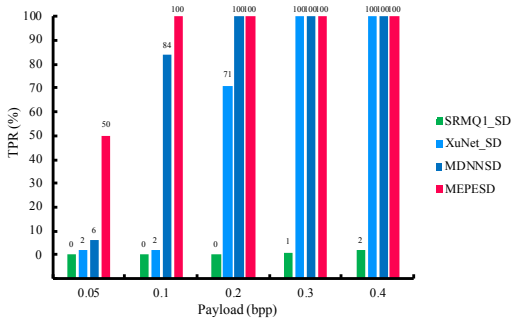


Fig. 7. The detection performance of different methods when the guilty user applies S-UNIWARD [15] with a single payload.

we can also find that our proposed method does not obtain the best performance for steganalysis. Although the performance is not too bad, it is still worse than that of SRMQ1 and others. These results may tell us that the proposed method MEPESD is not a best steganalytic method. From the next experiments, we begin to validate the effectiveness of it for steganographer detection task.

Model comparisons for steganographer detection. We further compare the performance of these four steganographer detection methods when the guilty user spreads all the images as stego images generated by S-UNIWARD with series of payloads, including 0.05bpp, 0.1bpp, 0.2bpp, 0.3bpp and 0.4bpp. Figure 7 shows the comparison results. We can find that the rich model based method SRMQ1_SD fails to detect the guilty user with all types of payloads. As an evident comparison, it obtains the best performance in steganalysis task as is mentioned above. We infer that the high dimension of steganalytic features extracted from the rich model based method will weaken the user distance representation from the steganalytic features, and make it difficult to distinguish the guilty user from the clustering results based on the user distance. Besides, the CNN based methods, namely MDNNSD, XuNet_SD and MEPESD, perform better than the traditional rich model based method SRMQ1_SD, whose dimensions of extracted features are all evidently lower than that of rich model based method. As the state-of-the-art CNN based method, MDNNSD obtains good performance in all the compared conditions. However, the proposed MEPESD obtains the most accurate performance with all types of payloads, and has evident advantages with low payloads, such as 0.05bpp and 0.1bpp. It proves the effectiveness of the proposed method in steganographer detection task, especially in more difficult conditions. These results indicate that there are essential differences in the feature extraction part between steganalysis and steganographer detection. The core problem of steganographer detection is to extract representative and discriminative features with low dimension.

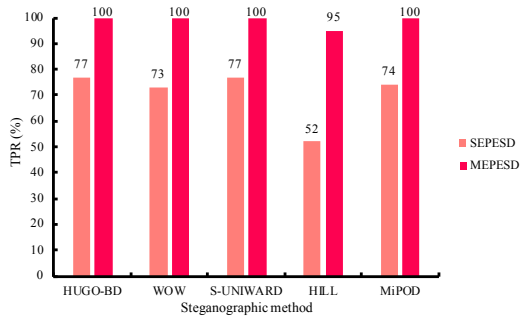


Fig. 8. The detection performance of SEPESD and MEPESD when the guilty user applies different steganographic methods with 0.1bpp payload.

4.4 Comparisons of different embedding probability combination strategies

In this sub-section, we discuss the different strategies to integrate the embedding probability map into the steganalytic feature extraction networks.

We compare two integration strategies, namely SEPESD and MEPESD. SEPESD denotes the integration strategy with a single combinatorial layer, which is after the first convolutional block CONV-S0 as is shown in Fig. 5. MEPESD is the proposed integration strategy with three combinatorial layers, which integrate the knowledge of multi-scale embedding probability maps into the convolutional feature maps of images. Both SEPESD and MEPESD apply the element-wise multiplication of the feature maps and the corresponding probability maps. These two strategies are compared when the guilty user spreads all the images as stego images generated by the state-of-the-art steganographic methods, including HUGO-BD [10], WOW [14], S-UNIWARD [15], HILL [28] and MiPOD [45], with a relatively challenging payload, namely 0.1bpp. As is shown in Fig. 8, MEPESD exceeds SEPESD in all comparable conditions. It is because the proposed MEPESD model integrates embedding probability maps into feature maps with different scales, which guides the steganalytic feature extraction in multiple levels and scales.

We also compare the integration strategies with single and triple combinatorial layers via element-wise summation of the feature maps and the corresponding probability maps. However, both strategies are failed in the experiments. It is because compared with multiplication strategies, the steganalytic features extracted via the summation strategies have a smaller range in extracted features, which are not distinguishable enough for the Gaussian vote model to detect the guilty user.

4.5 Comparisons of MDNNSD and MEPESD using multi-steganographies

With the proliferation of social media and multimedia data, especially image data, it is highly likely that the guilty user hides information into images by diverse steganographic methods with unknown embedding parameters. In this sub-section, we define a complex steganographic strategy, namely multi-steganographies, to compare the performance of the proposed MEPESD and the state-of-the-art CNN based steganographer detection method MDNNSD [62]. Multi-steganographies denotes the strategy to generate stego images using a fusion of multiple steganographic methods, namely HUGO-BD [10], WOW [14], S-UNIWARD [15], HILL [28] and MiPOD [45], with a single payload. Each steganographic method is adopted to generate 20% stego images in the experiments.

As is illustrated in Fig. 9, MDNNSD and MEPESD both obtain the best performance when the payloads are larger than 0.1bpp, which means the CNN based steganographer detection methods are

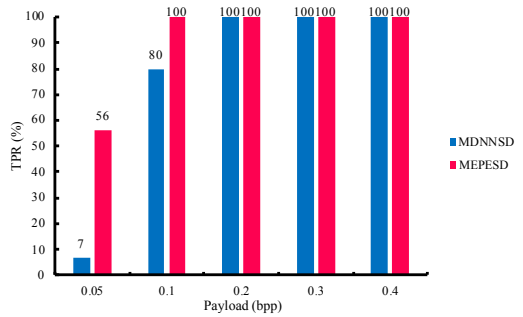


Fig. 9. The detection performance of MDNNSD [62] and MEPESD when the guilty user applies multiple steganographic methods, namely HUGO-BD [10], WOW [14], S-UNIWARD [15], HILL [28] and MiPOD [45], with a single payload.

Table 4. Efficiency comparison of MMD+AHC [24] and Gaussian vote using partial-embedding.

Steganographer detection strategy	Efficiency (s)
MMD+AHC	655.01
Gaussian vote	**1.22**

able to deal with the complex situation of multi-steganographies with these payloads. Meanwhile, MEPESD exceeds MDNNSD with the payload of 0.05bpp and 0.1bpp, which proves that the proposed method is more effective to detect guilty user who applies multiple steganographic methods to generate stego images, especially with a low payload.

We also compare the performance of MEPESD and MDNNSD when the guilty user uses the strategy to generate stego images with a single steganographic method and fused payloads, including 0.05bpp, 0.1bpp, 0.2bpp, 0.3bpp and 0.4bpp. Each payload is adopted to generate 20% stego images in the experiments. The true positive rate of MEPESD and MDNNSD are both 100% in this strategy, despite the steganographic method. It is because the steganalytic features extracted in the part of stego images with relatively large payloads, which are larger than 0.1bpp, are discriminative in steganographer detection.

4.6 Comparisons of MMD+AHC and Gaussian vote using partial-embedding

A common assumption of the previous experiments in steganographer detection is that the images spread by guilty users are all stego images. However, to hide the secret information from being detected, guilty users tend to spread images in a composition of both cover images and stego images. In this sub-section, we compare the performance of the proposed MEPESD method with two steganographer detection strategies, namely MMD+AHC [24] and Gaussian vote using partial-embedding, which denotes the strategy to insert α stego images with 0.1bpp payload into the spread images, where α equals 10%, 20%, 30%, 40% and 50% in the experiments.

Figure 10 shows the detection performance of MEPESD with MMD+AHC and Gaussian vote using the steganographic strategy of partial-embedding. It illustrated that the proposed MEPESD method can achieve more accurate performance using Gaussian vote in partial-embedding. What's more, as is shown in Table 4, the efficiency of Gaussian vote is more than 500 times faster than that of MMD+AHC using the steganographic strategy of partial-embedding. It is because the time complexity of Gaussian vote is lower than that of MMD+AHC. As is discussed in sub-section 3.3,

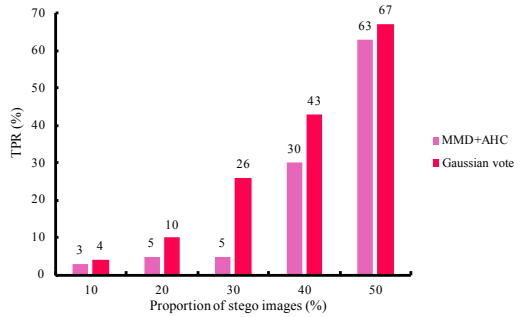


Fig. 10. The detection performance of MEPESD with MMD+AHC [24] and Gaussian vote when the guilty user applies partial-embedding to insert {10%, 20%, 30%, 40%, 50%} stego images with 0.1bpp payload into the spread images.

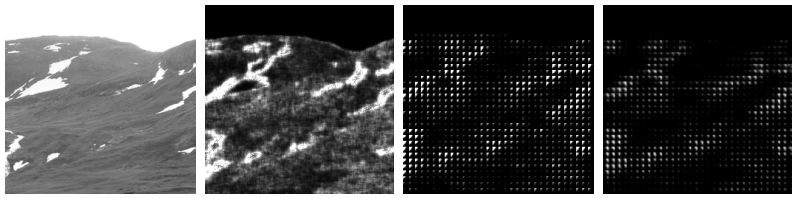
the Gaussian vote strategy compares the Gaussian distribution of each user with the mean Gaussian distribution in the training set, the time complexity of which is $O(N_U)$, where N_U denotes the number of users. As a comparison, the MMD+AHC strategy calculates the distance of each user against the rest of users, the time complexity of which is $O(N_U^2)$. In real-world applications, the efficiency difference between Gaussian vote and MMD+AHC will expand owing to the large amount of involved users, which makes the MMD+AHC strategy unacceptable in time cost.

However, in spite of the effectiveness and efficiency in the proposed MEPESD method, the Gaussian vote suffers robustness problem as is illustrated in sub-section 4.2 and sub-section 4.4. Because the result of Gaussian vote depends on the difference between the Gaussian distribution values of users in validation set and the mean Gaussian distribution values in training set, two situations may lead to the failure of Gaussian vote. For one thing, the data in validation set may share not much similarity with that in training set, so that the mean Gaussian distribution value in training set cannot represent the one in validation set. For another, the extracted steganalytic features are possibly not distinguishable enough, which will mislead the vote procedure. Further work remains to be done to improve the robustness of the proposed Gaussian vote method, including transferring trained data to validation set, and ensuring the discrimination of the extracted steganalytic features.

4.7 Extension experiment in frequency domain

The above experiments have evaluated the effectiveness of the proposed MEPESD method for the steganographer detection task in spatial domain. But as we know, the guilty user is also possible to hide information inside JPEG images. In order to verify the generalization ability of the proposed method, in this sub-section, we extend the task to frequency domain and try to apply the proposed method in this domain.

The JPEG version of BOSSbase ver 1.01 dataset [2] is utilized to evaluate the proposed method. Similar to the settings in the above experiments, the training set and validation set contain 20,000 images, respectively. Each image is compressed with JPEG quality factor 80 using Matlab's imwrite function. All images spread by the guilty user are generated by J-UNIWARD [15]. J-UNIWARD is the version of S-UNIWARD [15] in frequency domain, which is one of the most representative steganographic algorithms. It hides the secret message in the DCT coefficients of the transformed JPEG images, rather than directly hides message in the gray value of pixels in spatial domain. Two payloads are verified in this experiment, namely 0.1 and 0.4 bits per non-zeros Alternating Current DCT coefficient (bpnzAC), which represent the difficult condition and easy condition, respectively.



(a) Cover image (b) S-UNIWARD (c) J-UNIWARD (d) Estimated result

Fig. 11. An example of the comparison between the embedding probability map of S-UNIWARD in spatial domain and that of J-UNIWARD [15] in frequency domain. (a) Cover image. (b) S-UNIWARD [15] with 0.4bpp payload. (c) J-UNIWARD [15] with 0.4bpnzAC. (d) Estimated result of the proposed EEP.

Table 5. The detection performance of MDNNSD [62] and MEPESD when the guilty user applies J-UNIWARD [15] with different payloads, namely 0.1bpnzAC and 0.4bpnzAC.

Steganographer settings	Method	TPR (%)
J-UNIWARD with 0.1bpnzAC	MDNNSD	58
	MEPESD	**100**
J-UNIWARD with 0.4bpnzAC	MDNNSD	**100**
	MEPESD	**100**

In the training stage of the embedding probability estimation sub-network, the JPEG cover images in the training set, and the corresponding embedding probability maps generated by J-UNIWARD with 0.4bpnzAC payload, are utilized to train the network. As is shown in Fig. 11, owing to the blockwise DCT transform, the appearance of the embedding probability map of J-UNIWARD is quite different from that of S-UNIWARD. Although we can still use the same model to estimate the embedding probability map in frequency domain, some changes are made to adapt to the frequency domain in the following steps. First, in order to adapt to the probability distribution of the estimated results in frequency domain, we use the estimated embedding probability maps without denormalization, along with the JPEG cover images and the stego images generated by J-UNIWARD with 0.4bpnzAC payload, to train the steganalytic feature extraction sub-network. Second, we use MMD+AHC [24], which is introduced in section 4.2 and 4.6, instead of Gaussian vote, to detect the guilty user from the extracted features.

We compare the proposed method with the learning based method MDNNSD [62], by simply using the JPEG cover and stego images to train the networks. As is shown in Tabel 5, when the payload is 0.4bpnzAC, both MDNNSD and the proposed MEPESD get the best performance of steganographer detection. While when it comes to 0.1bpnzAC, MEPESD apparently exceeds MDNNSD. It illustrates that the proposed MEPESD is effective in both the easy and difficult conditions in frequency domain.

4.8 Extension experiment in uncertain number of guilty users

The above experiments follow the assumption that there is one and only guilty user in all the users. However, in real-world applications, the number of guilty users are unpredictable, which could be none or multiple. Thus, in this experiment, we extend the proposed MEPESD to the condition of uncertain number of guilty users, which is named as MEPESD_u. We make a new assumption of user composition, in which the summation of guilty users and innocent users is fixed to 100, and the number of guilty users ranges from 0 to 100. The images spread from the guilty users are all

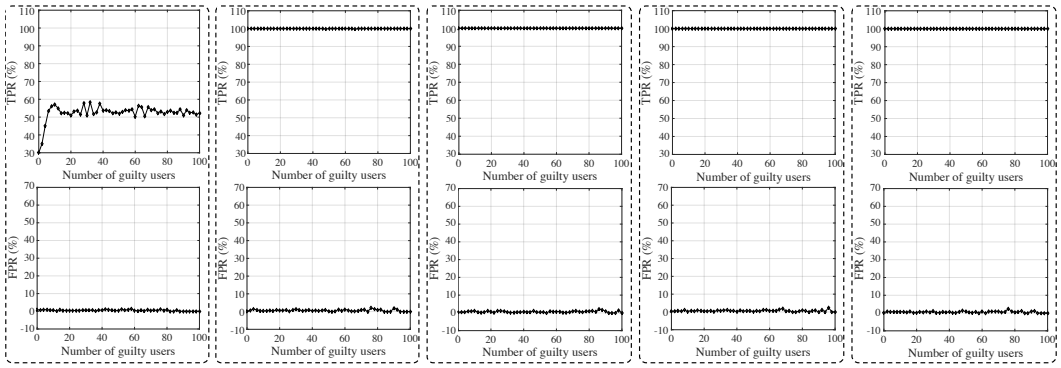


Fig. 12. The True Positive Rate (TPR) and False Positive Rate (FPR) of MEPESD_u when the number of guilty users ranges from 0 to 100, while the total number of users (including guilty ones and innocent ones) are fixed to 100. Images spread from the guilty users are all stego images using S-UNIWARD [15] with a specific payload, which is listed as follows. (a) 0.05bpp. (b) 0.1bpp. (c) 0.2bpp. (d) 0.3bpp. (e) 0.4bpp.

stego images using S-UNIWARD [15] with a specific payload. The other experimental settings are the same as the default settings described in section 4.1.

To extend the original MEPESD to the proposed MEPESD_u in this experiment, we modify the step of guilty user detection via Gaussian vote, by setting a threshold V to the vote number of each user, and selecting the users with the vote number over V as the guilty ones. The threshold V equals 169,000 in this experiment.

The performance of MEPESD_u is evaluated via two metrics, namely True Positive Rate (TPR) and False Positive Rate (FPR). As is shown in Fig. 12, the results of FPR are all close to 0%, regardless of the number guilty users and payloads. It proves that few innocent users are mis-detected via the proposed method. Besides, the results of TPR are close to 100% when the payload ranges from 0.1 to 0.4. It illustrates the effectiveness of the proposed method on the condition of uncertain number of guilty users. While MEPESD_u tends to miss about half of the guilty users in guilty user detection with 0.05bpp payload. It is because the difference between stego images and cover images are too small in this situation, which leads to the unideal performance of the proposed method.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose the first content-adaptive steganographer detection method, which is based on multi-scale estimated embedding probability map integration. The proposed method contains three steps, including estimating embedding probability maps via saliency detection networks, integrating multi-scale estimated embedding probability maps into a deep learning model to extract steganalytic features, and identifying the guilty user via the novel Gaussian vote strategy based on the extracted steganalytic features. To the best of our knowledge, we are the first to estimate embedding probability maps with a learning-based methods, which is independent of the steganographic methods. The experimental results show the effectiveness and rationality of each step in the proposed method, and the superiority of the proposed method against the state-of-the-art steganographer detection methods, especially in low payloads. Moreover, we validate the proposed method under more complex real-world circumstances, including multi-steganographies and partial-embedding. The proposed method achieves relatively good performance in both of

the listed situations. Finally, in the frequency domain, the proposed method also demonstrates its effectiveness.

In the future, we plan to carry our work forward in two ways. Our first future work is to extend the proposed method on the image data from large-scale social media networks. As we described in related work, many steganographic and steganalytic methods incorporate true embedding probability maps into their models to guide the embedding or attacking. Therefore, another meaningful future work is to integrate the proposed strategy of embedding probability estimation into the steganography and steganalysis to improve the performance of these methods.

6 ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of Guangdong Province (2016A030310053), the Shenzhen high-level overseas talents program, the National Science Foundation of China (61202320), the Science, Technology and Innovation Commission of Shenzhen Municipality (JCYJ20180307151516166), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

7 SUPPLEMENTAL

In this section, we provide the supplemental information to the proposed Multi-scale Embedding Probability Estimation based Steganographer Detection (MEPESD).

7.1 Details of the network architecture used in the proposed MEPESD

Table 6 shows the detailed network architecture of NLDF [37], which is organized as a 4×5 grid-like architecture, including 10 convolution blocks, five contrast blocks, four deconvolution blocks, one local block, one global block and one score block.

Table 7 shows the detailed architecture of the proposed steganalytic feature extraction networks, which includes two HPF blocks, nine convolution blocks and seven pooling blocks.

7.2 Extension experiment in integrating MEPESD into XuNet

In order to validate the effectiveness of the proposed probability map estimation method and the proposed multi-scale integration method in other steganalytic feature extraction networks, we try to integrate the proposed MEPESD into XuNet [53].

In this experiment, the extended MEPESD_Xu also includes three steps, namely embedding probability estimation, steganalytic feature extraction and guilty user detection. In the first step, we directly use the strategy in the proposed MEPESD to estimate probability maps. In the second step, we replace the original steganalytic feature learning sub-network with XuNet [53], and integrate the multi-scale estimated probability maps into the feature maps of the first three convolutional layers of XuNet. In the third step, we use MMD+AHC [24] to detect the guilty user from the extracted features.

To compare with the proposed MEPESD_Xu, we define the baseline method XuNet_SD, which extracts steganalytic features with XuNet [53], and detects the guilty user from the extracted features via MMD+AHC [24].

These two methods are compared when the guilty user spreads all the images as stego images generated by S-UNIWARD [15] with a single payload, which ranges from 0.05bpp to 0.4bpp. As is shown in Fig. 13, the proposed MEPESD_Xu exceeds the baseline method XuNet_SD, especially when the payload is 0.1bpp and 0.2bpp. It illustrates that the proposed probability map estimation method and the proposed multi-scale integration method will work on the other steganalytic feature extraction networks.

Table 6. Details of the NLDF model [37] utilized in the proposed embedding probability estimation method.

Block	Layer	Kernel	Stride	Zero padding
CONV-P1	2 conv	3×3	1	Yes
	max-pool	2×2	2	Yes
CONV-P2	2 conv	3×3	1	Yes
	max-pool	2×2	2	Yes
CONV-P3	3 conv	3×3	1	Yes
	max-pool	2×2	2	Yes
CONV-P4	3 conv	3×3	1	Yes
	max-pool	2×2	2	Yes
CONV-P5	3 conv	3×3	1	Yes
	max-pool	2×2	2	Yes
CONV-P6	conv	3×3	1	Yes
CONV-P7	conv	3×3	1	Yes
CONV-P8	conv	3×3	1	Yes
CONV-P9	conv	3×3	1	Yes
CONV-P10	conv	3×3	1	Yes
CONT-P1	avg-pool	3×3	1	No
CONT-P2	avg-pool	3×3	1	No
CONT-P3	avg-pool	3×3	1	No
CONT-P4	avg-pool	3×3	1	No
CONT-P5	avg-pool	3×3	1	No
DECONV-P2	deconv	5×5	2	Yes
DECONV-P3	deconv	5×5	2	Yes
DECONV-P4	deconv	5×5	2	Yes
DECONV-P5	deconv	5×5	2	Yes
LOCAL	conv	1×1	1	No
GLOBAL	conv-1	5×5	1	No
	conv-2	5×5	1	No
	conv-3	3×3	1	No
SCORE	conv-L	1×1	1	No
	conv-G	1×1	1	No

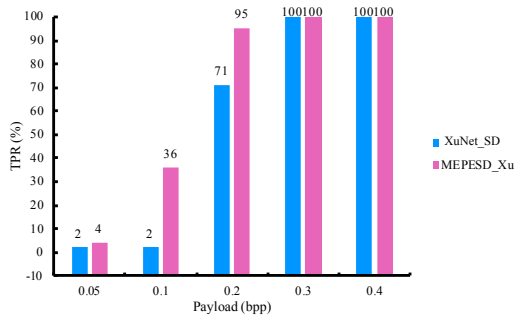


Fig. 13. The detection performance of XuNet_SD and MEPESD_Xu when the guilty user applies S-UNIWARD [15] with a single payload.

Table 7. Details of the network structure utilized in the proposed steganalytic feature extraction method.

Block	Layer	Kernel	Stride	Zero padding
HPF-S1	conv	5×5	1	Yes
CONV-S0	conv tanh	5×5	1	Yes
CONV-S1	conv bn relu	11×11	4	No
POOL-S1	max-pool	3×3	2	No
CONV-S2	conv bn relu	5×5	1	Yes
POOL-S2	max-pool	3×3	2	No
CONV-S3	conv bn relu	3×3	1	Yes
CONV-S4	conv bn relu	3×3	1	Yes
CONV-S5	conv bn relu	3×3	1	Yes
POOL-S3	max-pool	3×3	2	No
CONV-S6	conv bn relu	6×6	1	No
CONV-S7	conv bn relu	1×1	1	No
CONV-S8	conv	1×1	1	No
HPF-S2	conv	5×5	1	Yes
POOL-S4	max-pool	5×5	1	Yes
POOL-S5	max-pool	11×11	4	No
POOL-S6	max-pool	3×3	2	No
POOL-S7	max-pool	5×5	1	Yes

REFERENCES

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. 2009. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1597–1604.

[2] Patrick Bas, Tomáš Filler, and Tomáš Pevný. 2011. "Break Our Steganographic System": The Ins and Outs of Organizing BOSS. In *International Conference on Information Hiding*. 59–70.

[3] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. 2014. Salient Object Detection: A Survey. *Eprint Arxiv* 16, 7 (2014), 3118.

[4] Markus M. Breunig. 2000. LOF: identifying density-based local outliers. In *ACM SIGMOD International Conference on Management of Data*. 93–104.

[5] Matthias Carnein, Schöttle Pascal, and Böhme Rainer. 2014. Predictable rain?:steganalysis of public-key steganography using wet paper codes. In *2nd ACM IH & JMM Sec. Workshop*. 97–108.

[6] Rémi Cogranne and Florent Retraint. 2013. Application of hypothesis testing theory for optimal detection of LSB matching data hiding. *Signal Processing* 93, 7 (2013), 1724–1737.

[7] Tomas Denemark, Mehdi Boroumand, and Jessica Fridrich. 2017. Steganalysis Features for Content-Adaptive JPEG Steganography. *IEEE Transactions on Information Forensics and Security* 11, 8 (2017), 1736–1746.

[8] Tomas Denemark, Vahid Sedighi, Vojtech Holub, Remi Cogranne, and Jessica Fridrich. 2014. Selection-channel-aware rich model for Steganalysis of digital images. In *IEEE International Workshop on Information Forensics and Security*. 48–53.

[9] Lionel Fillatre. 2012. Adaptive Steganalysis of Least Significant Bit Replacement in Grayscale Natural Images. *IEEE Transactions on Signal Processing* 60, 2 (2012), 556–569.

[10] Tomáš Filler and Jessica Fridrich. 2010. Gibbs Construction in Steganography. *IEEE Transactions on Information Forensics and Security* 5, 4 (2010), 705–720.

[11] Jessica Fridrich and Tomas Filler. 2007. Practical methods for minimizing embedding impact in steganography. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, Vol. 6505. 650502.

[12] Jessica Fridrich, Miroslav Goljan, and David Soukal. 2005. Efficient wet paper codes. In *International Workshop on Information Hiding*. 204–218.

[13] Jessica Fridrich and Jan Kodovsky. 2012. Rich Models for Steganalysis of Digital Images. *IEEE Transactions on Information Forensics and Security* 7, 3 (2012), 868–882.

[14] Vojtech Holub and Jessica Fridrich. 2012. Designing steganographic distortion using directional filters. In *IEEE International Workshop on Information Forensics and Security*. 234–239.

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

[16] A. Hornung, Y. Pritch, P. Krahenbuhl, and F. Perazzi. 2012. Saliency filters: Contrast based filtering for salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 733–740.

[17] Qibin Hou, Ming Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. 2016. Deeply Supervised Salient Object Detection with Short Connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP, 99 (2016), 1–1.

[18] Donghui Hu, Qiang Shen, Shengnan Zhou, Xueliang Liu, Yuqi Fan, and Lina Wang. 2017. Adaptive Steganalysis Based on Selection Region and Combined Convolutional Neural Networks. *Security and Communication Networks* 2017, 4 (2017), 1–9.

[19] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang. 2017. Deep Level Sets for Salient Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 540–549.

[20] Andrew D. Ker. 2006. Batch steganography and pooled steganalysis. In *International Conference on Information Hiding*. 265–281.

[21] Andrew D. Ker. 2007. Batch steganography and the threshold game. In *Security, Steganography, and Watermarking of Multimedia Contents IX*. 401–413.

[22] Andrew David Ker and Tomáš Pevný. 2012. Batch steganography in the real world. In *Proc. 14th ACM Workshop Multimedia Security (MM&Sec)*. 1–10.

[23] Andrew D. Ker and Tomáš Pevný. 2012. Identifying a steganographer in realistic and heterogeneous data sets. In *Proc. SPIE, Media Watermark., Security, Forensics XIV*. 265–298.

[24] Andrew D. Ker and Tomáš Pevný. 2012. A new paradigm for steganalysis via clustering. *Proceedings of SPIE - The International Society for Optical Engineering* 7880, 2 (2012), 87–95.

[25] Andrew D. Ker and Tomáš Pevný. 2014. The Steganographer is the Outlier: Realistic Large-Scale Steganalysis. *IEEE Transactions on Information Forensics and Security* 9, 9 (2014), 1424–1435.

[26] Onkar Krishna and Kiyoharu Aizawa. 2018. Billboard Saliency Detection in Street Videos for Adults and Elderly. In *IEEE International Conference on Image Processing*. 2326–2330.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*. 1097–1105.

[28] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. 2015. A new cost function for spatial image steganography. In *IEEE International Conference on Image Processing*. 4206–4210.

[29] Fengyong Li, Mi Wen, Jingsheng Lei, and Yanli Ren. 2017. Efficient steganographer detection over social networks with sampling reconstruction. *Peer-to-Peer Networking and Applications* 7 (2017), 1–16.

[30] Fengyong Li, Kui Wu, Jingsheng Lei, Mi Wen, Zhongqin Bi, and Chunhua Gu. 2017. Steganalysis Over Large-Scale Social Networks With High-Order Joint Features and Clustering Ensembles. *IEEE Transactions on Information Forensics and Security* 11, 2 (2017), 344–357.

[31] Guanbin Li and Yizhou Yu. 2015. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5455–5463.

[32] Li Li, Weiming Zhang, Kejiang Chen, Hongyue Zha, and Nenghai Yu. 2018. Side Channel Steganalysis: When Behavior is Considered in Steganographer Detection. *Multimedia Tools and Applications* (2018), 1–15.

[33] Xin Liao, Guoyong Chen, and Jiaojiao Yin. 2016. Content-adaptive steganalysis for color images. *Security and Communication Networks* 9, 18 (2016), 5756–5763.

[34] Guo Shiang Lin, Yi Ting Chang, and Wen Nung Lie. 2010. A Framework of Enhancing Image Steganography With Picture Quality Optimization and Anti-Steganalysis Based on Simulated Annealing Algorithm. *IEEE Transactions on Multimedia* 12, 5 (2010), 345–357.
[35] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung Yeung Shum. 2011. Learning to Detect a Salient Object. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 2 (2011), 353–367.
[36] Weiqi Luo, Haodong Li, Qi Yan, Rui Yang, and Jiwu Huang. 2018. Improved Audio Steganalytic Feature and Its Applications in Audio Forensics. *ACM Transactions on Multimedia Computing, Communications, and Applications* 14, 2 (2018), 43:1–43:14.
[37] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre Marc Jodoin. 2017. Non-local Deep Features for Salient Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 6593–6601.
[38] David R. Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. 2001. A Database of Human Segmented Natural Images and its Application to. *IEEE International Conference on Computer Vision* 2, 11 (2001), 416–423.
[39] N. N. A. Molok, S. Chang, and A. Ahmad. 2013. Disclosure of organizational information on social media: Perspectives from security managers. *Nature Protocols* 4, 1 (2013), 102–106.
[40] Tomas Pevny and Jessica Fridrich. 2008. Multiclass Detector of Current Steganographic Methods for JPEG Format. *IEEE Transactions on Information Forensics and Security* 3, 4 (2008), 635–650.
[41] Lionel Pibre, Jérôme Pasquet, Dino Ienco, and Marc Chaumont. 2016. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source mismatch. *Electronic Imaging* 4, 8 (2016), 1–11.
[42] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. 2015. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, Vol. 9409. 94090J.
[43] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. 2017. Feature learning for steganalysis using convolutional neural networks. *Multimedia Tools and Applications* 2 (2017), 1–25.
[44] Bernhard Schölkopf, John Platt, and Thomas Hofmann. 2007. A Kernel Method for the Two-Sample-Problem. In *Conference on Advances in Neural Information Processing Systems*. 513–520.
[45] Vahid Sedighi, Rémi Cogranne, and Jessica Fridrich. 2015. Content-Adaptive Steganography by Minimizing Statistical Detectability. *IEEE Transactions on Information Forensics and Security* 11, 2 (2015), 221–234.
[46] Priyanka Singh, Balasubramanian Raman, Nishant Agarwal, and Pradeep K Atrey. 2017. Secure cloud-based image tampering detection and localization using POB number system. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 3 (2017), 23.
[47] Weixuan Tang, Haodong Li, Weiqi Luo, and Jiwu Huang. 2014. Adaptive steganalysis against WOW embedding algorithm. In *ACM Workshop on Information Hiding and Multimedia Security*. 91–96.
[48] Weixuan Tang, Haodong Li, Weiqi Luo, and Jiwu Huang. 2016. Adaptive Steganalysis Based on Embedding Probabilities of Pixels. *IEEE Transactions on Information Forensics and Security* 11, 4 (2016), 734–745.
[49] Weixuan Tang, Shunquan Tan, Bin Li, and Jiwu Huang. 2017. Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Processing Letters* PP, 99 (2017), 1–1.
[50] Songtao Wu, Shenghua Zhong, and Yan Liu. 2017. Deep residual learning for image steganalysis. *Multimedia Tools and Applications* (2017), 1–17. https://doi.org/10.1007/s11042-017-4440-4
[51] Songtao Wu, Shenghua Zhong, and Yan Liu. 2018. Deep residual learning for image steganalysis. *Multimedia Tools and Applications* 77, 9 (2018), 10437–10453.
[52] Zhihua Xia, Xinhui Wang, Xingming Sun, Quansheng Liu, and Naixue Xiong. 2016. Steganalysis of LSB matching using differences between nonadjacent pixels. *Multimedia Tools and Applications* 75, 4 (2016), 1947–1962.
[53] Guanshuo Xu, Han Zhou Wu, and Yun Qing Shi. 2016. Structural Design of Convolutional Neural Networks for Steganalysis. *IEEE Signal Processing Letters* 23, 5 (2016), 708–712.
[54] Jianhua Yang, Kai Liu, Xiangui Kang, Edward Wong, and Yunqing Shi. 2017. Steganalysis Based on Awareness of Selection-Channel and Deep Learning. In *International Workshop on Digital Watermarking*. 263–272.
[55] Xiaoshan Yang, Tianzhu Zhang, and Changsheng Xu. 2015. Cross-domain feature learning in multimedia. *IEEE Transactions on Multimedia* 17, 1 (2015), 64–78.
[56] Ying Yang and Ioannis Ivrissimtzis. 2014. Mesh Discriminative Features for 3D Steganalysis. *ACM Transactions on Multimedia Computing, Communications, and Applications* 10, 3 (2014), 27:1–27:13.
[57] Jian Ye, Jiangqun Ni, and Yang Yi. 2017. Deep Learning Hierarchical Representations for Image Steganalysis. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2545–2557.
[58] Hao Yin, Wen Hui, Hongzhi Li, Chuang Lin, and Wenwu Zhu. 2012. A Novel Large-Scale Digital Forensics Service Platform for Internet Videos. *IEEE Transactions on Multimedia* 14, 1 (2012), 178–186.
[59] Peng Zhang, Tao Zhuo, Wei Huang, Kangli Chen, and Mohan Kankanhalli. 2017. Online object tracking based on CNN with spatial-temporal saliency guided sampling. *Neurocomputing* 257 (2017), 115–127.

[60] Xiang Zhang, Fei Peng, and Min Long. 2018. Robust Coverless Image Steganography based on DCT and LDA Topic Classification. *IEEE Transactions on Multimedia* PP, 99 (2018), 1–1.

[61] Mingjie Zheng, Sheng-hua Zhong, Songtao Wu, and Jianmin Jiang. 2017. Steganographer detection via deep residual network. In *IEEE International Conference on Multimedia and Expo*. 235–240.

[62] Mingjie Zheng, Sheng-hua Zhong, Songtao Wu, and Jianmin Jiang. 2018. Steganographer Detection based on Multiclass Dilated Residual Networks. In *ACM International Conference on Multimedia Retrieval*. 300–308.

[63] Hang Zhou, Kejiang Chen, Weiming Zhang, and Nenghai Yu. 2017. Comments on "Steganography Using Reversible Texture Synthesis". *IEEE Transactions on Image Processing* 26, 4 (2017), 1623.