A Multi-Task Learning CNN for Image Steganalysis

Xiangyu Yu, Huabin Tan, Hui Liang
School of Electronic and Information Engineering, South China University of Technology
Guangzhou, Guangdong, P. R. China
yuxy@scut.edu.cn, eethb, l.h04@mail.scut.edu.cn

Chang-Tsun Li
School of Computing and Mathematics,Charles Sturt University
Australia
chli@csu.edu.au

Guangjun Liao
Faculty of Forensic Science and Technology, Guangdong Police College
Guangzhou, Guangdong, P. R. China
56114827@qq.com

Abstract

Convolutional neural network (CNN) based image steganalysis are increasingly popular because of their superiority in accuracy. The most straightforward way to employ CNN for image steganalysis is to learn a CNN-based classifier to distinguish whether secret messages have been embedded into an image. However, it is difficult to learn such a classifier because of the weak stego signals and the limited useful information. To address this issue, in this paper, a multi-task learning CNN is proposed. In addition to the typical use of CNN, learning a CNN-based classifier for the whole image, our multi-task CNN is learned with an auxiliary task of the pixel binary classification, estimating whether each pixel in an image has been modified due to steganography. To the best of our knowledge, we are the first to employ CNN to perform the pixel-level classification of such type. Experimental results have justified the effectiveness and efficiency of the proposed multi-task learning CNN.

1. Introduction

Image steganography, a data hiding technique frequently used in multimedia communications, aims to embed secret messages into an image while making the embedding traces as undetectable as possible. Nowadays, the most secure steganographic algorithms are content-adaptive [21], [10], [11], [17], [33], and typical algorithms of such category in spatial domain include WOW [10], S-UNIWARD [11], and HILL [17]. Correspondingly, image steganalysis is the art of detecting the existence of secret messages embedded into an image. The conventional pipeline of steganalysis consists of two steps, feature exaction and classification, and the latter step is usually implemented using ensemble classifier (EC) [15]. One typical algorithm for feature extraction in spatial domain is Spatial Rich Models (SRM) [6], which consists of multiple co-occurrence matrices formed by four neighboring quantized noise residual samples. And one of its variant, maxSRM [5], which incorporate the probability of each pixel being modified when executing embedding (the so-called selection channel) into the features of SRM, is the state-of-the-art handcrafted feature for image steganalysis. However, it should be noted that, in the conventional framework of steganalysis, feature extraction and classification are two separate steps, which means that it is hard to optimize them simultaneously.

In the past few years, the great superiority of convolutional neural networks (CNN) has been exhibited experimentally in a variety of computer vision problems [16], [8], [20]. By using CNN, feature extraction and classification can be easily unified in a single architecture and optimized jointly, which is expected to attain better performance. Impressed by the extraordinary advantage of CNN, researchers also seek to design proper CNN structures for image steganalysis [27], [23], [22], [29], [28], [19], [30], [31], [32], [18]. The first effective attempt is the work of Tan and Li [27] in 2014. They proposed a method using the mechanism of auto-encoder to pre-train a CNN for im-

age steganalysis, which achieved much better result than CNN without this pre-train step, but was still quite inferior to SRM. In [23], Qian *et al*. proposed a CNN inheriting the traditional steganalysis schemes to initialize the first layer of the network with a high-pass filter used in SRM. Experiments showed that their scheme could achieve comparable performance with SRM, and the use of the high-pass filter at the beginning of a CNN becomes a standard configuration [22], [29], [28], [19], [30], [31], [32], [18]. However, it is still difficult to train CNNs with stego images of low payload [22]. To circumvent this obstacle, Qian *et al*. [22] proposed an approach based on transfer learning. In 2016, Xu *et al*. [29] designed a CNN with an absolute activation function, hyperbolic tangent activation function and batch normalization [12]. Using an architecture similar to [29] as base learner, Xu *et al*. [28] estimated the performance of three different ensemble strategies. Another ensemble method, which combines a CNN with SRM-EC, is proposed in [19]. In 2017, Ye *et al*. [31] proposed to initialize the first layer of their CNN with 30 high-pass filters used in SRM and introduced a novel activation function called truncated linear unit (TLU). To further improve the performance, they incorporate the knowledge of the selection channel into CNN and this architecture, named SCA-TLU-CNN, which is the state-of-the-art selection-channel-aware CNN-based steganalyzer, outperforms the maxSRM by a significant margin. Similarly, Yang *et al*. [30] proposed a CNN structure considering the selection channel and also observed performance gain.

Inspired by the fact that properly incorporating the pixel-level information [30], [31] can further boost the performanceis, pixel binary classification is considered in CNN-based steganalyzer in this paper. Specifically, a multi-task learning architecture, which optimize the main task of image steganalysis and the auxiliary task of pixel binary classification estimating whether each pixel has been modified due to secret message embedding, is employed. Extensive experiments show that the proposed multi-task learning CNN can further improve the performance of its corresponding single-task learning version for image steganalysis. Moreover, the proposed CNN is comparable with the SCA-TLU-CNN [31] in terms of detection error rate, while ours might be more efficient in the testing phase as our approach does not need to explicitly calculate the pixel-level information when inference.

2. The propesed method

Multi-task Learning(MTL), a commonly used approach to train at least two related tasks simultaneously, can improve the generalization power of each task or the main task by leveraging the domain-specific information contained in the training signals of related tasks [3]. In recent years, reenforcing CNNs with MTL has been proved effective in various computer vision problems [34], [25], [7], [4]. A common approach of applying MTL to CNN is to share some hidden layers of the network between all tasks, while keeping several task-specific output layers [24]. In this paper, we attempt to learn a CNN for image steganalysis through the MTL mechanism.

The proposed architecture consists of three parts, the output layers for image steganalysis, specific layers for pixel binary classification and a simple backbone shared by the two tasks, which is indicated in green, blue and yellow in Figure 1 respectively.

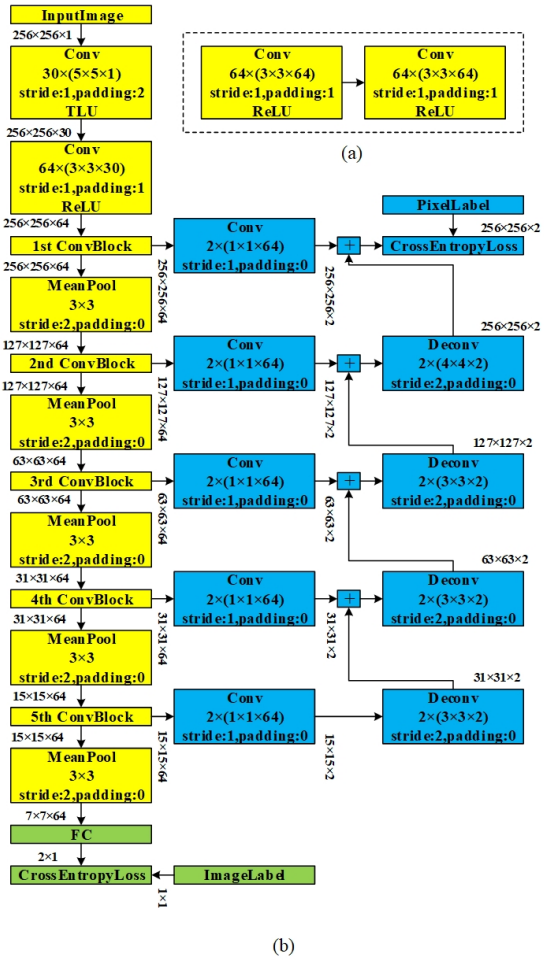


Figure 1. The architecture of proposed multi-task learning CNN. (a)The repeating convolutional block. (b) The overall training structure of the proposed architecture. Best viewed in color.

The shared backbone of the proposed architecture is a simple VGG-like [26] network, which is consists of a stack of 3×3 convolutional layers and pooling layers. More specifically, the first layer of the backbone is a 5×5 convolutional layer, whose weights is initialized with the 30 high-pass filters used in SRM to compute residuals as suggested in [31]. The second convolutional layer contains 64 filters

of size 3×3. Then a simple convolutional block (illustrated in Figure 1(a)) and a mean pooling layer are stacked alternatively for 5 times, where the convolutional block composes of 2 convolutional layers with the kernel size of 3×3. The specific layer for image steganalysis is a fully connected layer with 2 neurons appended after the last pooling layer, outputting the result of image steganalysis.

As for the specific branches for pixel binary classification, it is similar to the structure of Fully Convolutional Network (FCN) [20], a simple but effective architecture originally proposed for semantic segmentation. By upsampling and fusing the feature maps from deep to shallow stages, FCN finally can output a score map with the same size of the input image. Each element in the score map can be viewed as the confidence of the corresponding pixel belonging to a certain category. In our implement, feature from all five scales are extracted and fused so as to capture more accuracy information about location. To be specific, a convolutional layer with 2 kernels of size 1×1 is appended to the output of each of the 5 simple convolutional blocks mentioned above to compute score maps for different scales. In order to fuse these score maps from different scales, those from small scales are up-sampled firstly. Specifically, a deconvolutional layer is applied to up-sample the small-scale score map, making it have the same scale as its adjacent bigger one, and then those 2 score maps are summed. The summed score map is again up-sampled and added to its adjacent bigger score map. The above process repeats until the final score map, whose scale is the same as that of the input image, is constructed. The value of each element in the final score map indicates whether the corresponding pixel in the original image is modified.

Moreover, it should be noted that ReLU is applied to all the convolutional layers except the first one in the backbone. For the first convolutional layer, a TLU is used, which has been experimentally proved to be better than ReLU for the early layers of CNN [31]. TLU can be formulated as follows:

$$f(x) = \begin{cases} -T, & x < -T \\ x, & -T \leq x \leq T \\ T, & x > T. \end{cases} \tag{1}$$

where T is the truncated threshold.

In order to train a CNN classifier to estimate whether a pixel has been modified due to steganographic operations, ground truth labels need to be available firstly. There should be a pixel label map for an image. To construct this map, each stego image will be compared with its corresponding cover image pixel by pixel to generate its label map. Therefore, the obtained pixel label map has the same size as its stego image. And each element in the map indicates the ground truth of whether a pixel in the same position of the corresponding stego image is changed.

During training, the two tasks, image steganalysis and pixel binary classification, are optimized simultaneously. To this end, one cross-entropy loss layer is added after the fully connected layer to compute the loss L_{image} for image steganalysis, and another cross-entropy loss layer is added to the final score map mentioned above to calculate the loss L_{pixel} for pixel binary classification. The image steganalysis loss L_{image} is:

$$L_{image} = -\frac{1}{N}\sum_{i=0}^{N-1} log\frac{e^{(f^{image}(x_i))_{y_i^{image}}}}{\sum_{c=0}^{1} e^{(f^{image}(x_i))_c}} \tag{2}$$

where x_i is the ith input image, $y_i^{image} \in \{0, 1\}$ is the image steganalysis label of x_i, N is the amount of input images, $f^{image}(\cdot)$ is the transform function of the network for image steganalysis, and $(f^{image}(\cdot))_c$ is the output of the function for the cth class ($c = \{0, 1\}$). And these two classes correspond to the two neurons after the fully connected layer in Figure 1. For pixel classification, cross-entropy loss is computed for each pixel independently, and then the losses for all pixels of an image are averaged. Mathematically, the pixel classification loss L_{pixel} is:

$$L_{pixel} = -\frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{mn}\sum_{j=0}^{m-1}\sum_{k=0}^{n-1} log\frac{e^{(f^{pixel}(x_{i,j,k}))_{y_{i,j,k}^{pixel}}}}{\sum_{c=0}^{1} e^{(f^{pixel}(x_{i,j,k}))_c}} \tag{3}$$

where $x_{i,j,k}$ is the pixel at the jth row and kth column of the ith input image, m and n are the height and width of input images respectively, $y_{i,j,k}^{pixel} \in \{0, 1\}$ is the pixel classification label of $x_{i,j,k}$, $f^{pixel}(\cdot)$ is the transform function of the network for pixel classification, and $(f^{pixel}(\cdot))_c$ is the output of the function for the cth class ($c = \{0, 1\}$).

Since cover images are always unchanged, only stego images are concerned in our implementation when optimizing the CNN for pixel binary classification. Generally, the amount of unchanged pixels is much larger than that of changed ones (for example, the average change rate at payload of 0.2 bpp for HILL in BOSSbase is 3.56%), so proper strategy should be taken to deal with this extremely unbalanced situation to optimize the task of pixel classification better. A very simple but effective method is adopted here. Specifically, in each stego image, only certain unchanged pixels of equal quantity as changed ones are randomly sampled for training, while the remaining unchanged pixels are ignored during optimization. Suppose the image labels for cover images and stego images are 0 and 1, and the pixel labels for changed pixels, sampled unchanged pixels and ignored unchanged pixels are 0, 1, and 2, respectively. Then

Eq. (3) can be rewritten as:

$$L_{pixel} = -\frac{1}{\sum_{i=0}^{N-1} I_i} \sum_{i=0}^{N-1} \left(\frac{I_i}{\sum_{j=0}^{m-1}\sum_{k=0}^{n-1} P_{i,j,k}} \sum_{j=0}^{m-1}\sum_{k=0}^{n-1} (P_{i,j,k} \log(L_{i,j,k})) \right) \tag{4}$$

where

$$I_i = [y_i^{image} = 1] \tag{5}$$

$$P_{i,j,k} = [y_{i,j,k}^{pixel} \neq 2] \tag{6}$$

$$L_{i,j,k} = \frac{e^{(f^{pixel}(x_{i,j,k}))_{y_{i,j,k}}}}{\sum_{c=0}^{1} e^{(f^{pixel}(x_{i,j,k}))_c}} \tag{7}$$

where $[\cdot]$ outputs 1 if the condition to be judged is satisfied, and 0 if not.

In our MTL framework, the above two losses are summed as follows:

$$L_{total} = L_{image} + \lambda_{pixel} L_{pixel} \tag{8}$$

where λ_{pixel} is the weight for the loss of the pixel classification, which will be determined through experiments. The summed loss L_{total} is minimized during training, and in this way, the two tasks can be simultaneously optimized. It should be noted that the optimization will be reduced to common single-task learning when $\lambda_{pixel} = 0$, and the performance of this case in company with that of MTL will be reported in the experiments.

3. Experiments

3.1. Datasets

Following [31], our experiments are carried out on the dataset of BOSSbase 1.01 [1] and BOWS2 [2], both of which contain 10,000 grayscale images of size 512×512. Constrained by our available computing resource, the central part of all images is cropped firstly to obtain images of size 256×256, just as what Ye *et al.* [31] did. Then three different data sets are generated as follows. Training set contains 4,000 images randomly selected from BOSSbase and all the 10,000 images from BOWS2. As for the 6,000 remaining images in BOSSbase, validation set contains 1000 and testing set contains 5,000. The three datasets are not intersecting.

For each image in all datasets, three state-of-the-art steganographic methods, WOW [10], S-UNIWARD [11], and HILL [17], are employed to embed secret messages to it at certain payload to obtain its corresponding stego image. Then each dataset contains cover images and their corresponding stego images.

Each CNN is independently trained on the three different training sets, and for each training, the parameter model with lowest error rate on the corresponding validation set will be tested on the corresponding testing set. The three obtained testing results are averaged as the final performance of that CNN.

3.2. Implementation details

1) Preprocessing: before training, the gray level of all images are firstly divided by 255 to be scaled to the range of 0 to 1, then further normalized by subtracting the mean value and dividing the standard deviation computed from the training set. During training, augmentations of randomly flipping and rotation are conducted.

2) Network initialization: as mentioned in Section 2, the weights of the first convolutional layer is initialized with 30 high-pass filters used in SRM, which is proposed in [31]. The weights of all the remaining convolutional and deconvolutional layers are initialized via "Xavier" method [9], while the weights of the fully connected layer are initialized with values randomly generated from a Gaussian source with zero mean and standard deviation of 0.01. And all the biases in the model are initialized to 0. The parameter T for the aforementioned TLU is set to 0.3.

3) Optimizer setting: In our experiments, Adam [14] is used to optimize the proposal CNN. The two main parameters of Adam, betas, which are used for computing running averages of gradient and its square, are set to 0.9 and 0.999 respectively. Other parameters are set as follows: the batch size is 32, with 16 cover images and their corresponding stego images in each batch; the weight decay is 1×10^{-4}. For models at payloads from 0.3 bpp (bit per pixel) to 0.5 bpp, the networks are trained from scratch. As for the lower payloads from 0.05 bpp to 0.2 bpp, transfer learning [22] is applied.

Based on the above settings, the proposed multi-task learning CNN is trained to minimize the cross-entropy losses of image steganalysis and that of pixel classification simultaneously.

3.3. Results and discussions

To quantitatively analysis the effectiveness of estimating whether each pixel in an image has been modified using CNN, the comparison of the Recall/Precision/F1 Score for the modified pixels between randomly guessing and our proposed architecture for WOW, S-UNIWARD, and HILL are performed. And the results for HILL algorithm are summarized in Table 1. From Table 1 it can be found that our approach can recall most of the modified pixels with acceptable precision.

Then we verify the effectiveness of the proposed multi-task learning CNN and determine the proper loss weight λ_{pixel} in Eq. (8) through experiments. To this end, the proposed architecture is trained for the three steganographic algorithms at payload of 0.4 bpp with six different λ_{pixel}

Algorithm	Payload (bpp)	Recall		Precision		F1 Score	
		Random	MTL-CNN	Random	MTL-CNN	Random	MTL-CNN
HILL	0.05	0.5000	0.8774	0.0071	0.0316	0.0140	**0.0611**
	0.1	0.5000	0.8519	0.0158	0.0593	0.0306	**0.1108**
	0.2	0.5000	0.8345	0.0357	0.1040	0.0666	**0.1849**
	0.3	0.5000	0.8182	0.0579	0.1470	0.1038	**0.2492**
	0.4	0.5000	0.8145	0.0822	0.1846	0.1412	**0.3010**
	0.5	0.5000	0.8068	0.1083	0.2212	0.1780	**0.3472**

Table 1. The comparations of the recall/precision/F1 score for the modified pixels between randomly guessing and our method ($\lambda_{pixel} = 1$).

Algorithm	$\lambda_{pixel} = 0$	$\lambda_{pixel} = 0.5$	$\lambda_{pixel} = 0.75$	$\lambda_{pixel} = 1$	$\lambda_{pixel} = 1.25$	$\lambda_{pixel} = 1.5$
WOW	0.1955	0.1776	0.1761	**0.1731**	0.1737	0.1734
S-UNIWARD	0.2295	0.2198	0.2137	**0.2065**	0.2114	0.2097
HILL	0.2515	0.2304	0.2312	0.2278	**0.2262**	0.2305

Table 2. The performance in terms of detection error (P_E) with different λ_{pixel} settings. The embedding payload is 0.4 bpp.

(0/0.5/0.75/1/1.25/1.5), and the experimental results are summarized in Table 2. It is observed that, for all the five involved non-zero weights, the multi-task learning CNN achieves consistently better performance than the baseline single-task learning CNN with $\lambda_{pixel} = 0$. Thus, it is advantageous to train the two tasks jointly, and superiors result can be attained with suitable λ_{pixel}. It seems that setting $\lambda_{pixel} = 1$ is a good choice according to the results of the experiments. Therefore, we fix $\lambda_{pixel} = 1$ and refer the model trained with this weight as MTL-CNN, while referring the one trained with $\lambda_{pixel} = 0$ as STL(single task learning)-CNN in the following experiments.

After the weight is determined, extensive experiments are conducted to compare the performance of the proposed MTL-CNN with its corresponding STL-CNN, maxSRM as well as the SCA-TLU-CNN [31]. The comparisons are summarized in Table 3 and illustrated in Figure 2. Firstly, it is observed that the STL-CNN outperforms the maxSRM for all embedding methods and payloads. As we expect, the MTL-CNN further decreases the detection error rate of STL-CNN by a margin from 0.66% to 3.96%. We owe this performance gain to the use of pixel binary classification. Specifically, we speculate that when optimized simultaneously with the pixel binary classification, the image steganalysis can be aware of the the suspicious regions embedded secret messages implicitly. It is also observed that the performance gains for very low payload (0.05bpp) and very high payload (0.5bpp) are not evident. This is because, for very low payload, only few pixels are modified, so there may be not sufficient training samples to train a discriminating model for pixel classification and thus less useful information can be shared by the main task of image steganalysis. On the other hands, for very high payload, we consider that the STL-CNN can also be able to be aware of the suspicious regions embedded secret messages implicitly by itself, and therefore, the advantage of pixel binary classification as an auxiliary task is not so prominent.

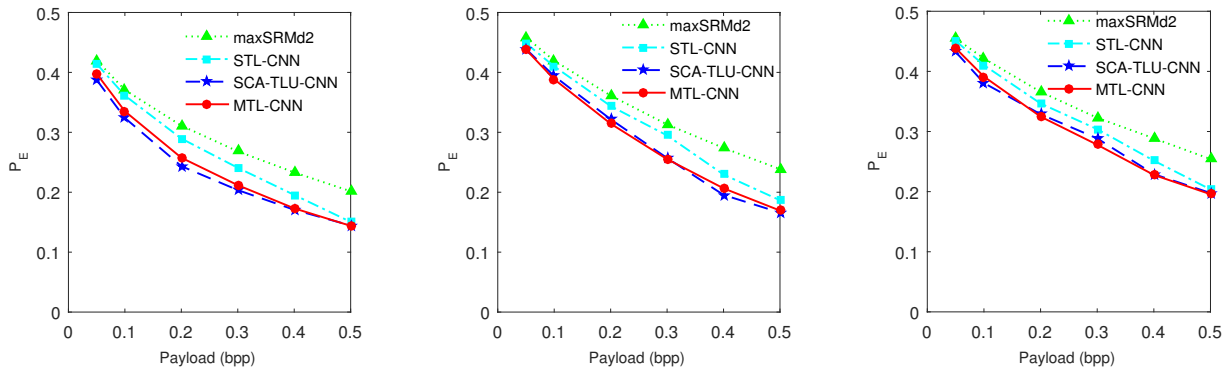
When compared with the SCA-TLU-CNN [31], it is observed that the MTL-CNN obtains comparable performance for S-UNIWARD and HILL, while is slightly inferior for WOW. However, SCA-TLU-CNN need to calculate the selection channel for each image during both training and testing phases, while the branches for pixel classification of our MTL-CNN can be removed and we just need to keep the backbone during inference. That is, our approach amounts to eliminating the step of evaluating pixel-level information while still maintaining its performance with this extra step. This makes our approach is more efficient, since it is time-consuming to compute the selection channel explicitly using the open source codes running on CPU [13]. Specifically, in our experimental platform (Intel Xeon(R) CPU E-5-2609 v3, GeForce GTX 1080 Ti), it takes over than 100ms to explicitly to calculate the selection channel in SCA-TLU-CNN, while it just takes less than 10ms for our proposed architecture to output the result using GPU after removing the branches for pixel classification.

4. Conclusion

In this paper, we explored the possibility of improving the performance of CNN-based image steganalysis via multi-task learning. Specifically, we propose to train a CNN for image steganalysis and classfying whether a pixel has been modified or not due to steganography simultaneously. Extensive experiments show that the proposed multi-task learning CNN can further boost the performance of its corresponding single-task learning CNN. And it is comparable with the state-of-the-art selection-channel-aware CNN-based steganalyzer in terms of detections error rate, but more efficient during inference.

Algorithm	Payload (bpp)	maxSRMd2 [5] (P_E)	STL-CNN (P_E)	SCA-TLU-CNN [31] (P_E)	MTL-CNN (P_E)
WOW	0.05	0.4202	0.4143	**0.3874**	0.3978
	0.1	0.3707	0.3613	**0.3240**	0.3351
	0.2	0.3112	0.2896	**0.2435**	0.2576
	0.3	0.2682	0.2408	**0.2036**	0.2118
	0.4	0.2331	0.1955	**0.1707**	0.1731
	0.5	0.2016	0.1505	0.1445	**0.1439**
S-UNIWARD	0.05	0.4587	0.4486	0.4390	**0.4381**
	0.1	0.4195	0.4100	0.3938	**0.3884**
	0.2	0.3613	0.3438	0.3218	**0.3144**
	0.3	0.3131	0.2948	0.2571	**0.2552**
	0.4	0.2739	0.2295	**0.1955**	0.2065
	0.5	0.2386	0.1866	**0.1660**	0.1694
HILL	0.05	0.4548	0.4508	**0.4325**	0.4386
	0.1	0.4215	0.4104	**0.3806**	0.3913
	0.2	0.3660	0.3466	0.3288	**0.3248**
	0.3	0.3228	0.3039	0.2885	**0.2780**
	0.4	0.2887	0.2515	0.2291	**0.2278**
	0.5	0.2541	0.2037	0.1977	**0.1957**

Table 3. Performance comparison of the involved steganalyzers in terms of detection error (P_E) for 3 state-of-the-art steganographic schemes at different payloads.



(a) (b) (c)

Figure 2. Detection error (P_E) of 3 steganographic schemes as a function of payload for the involved steganalysis methods. (a) WOW. (b) S-UNIWARD. (c) HILL.

Acknowledgment

This work is jointly sponsored by the China Scholarship Council and the Marie Skodowska-Curie actions of the EU Horizon 2020 programme through the project entitled Computer Vision Enabled Multimedia Forensics and People Identification (Project No. 690907, Acronym: IDENTITY), along with the Fundamental Research Funds for the Central Universities(2017MS046), Science and Technology Foundation of Guangdong Province(2017A050501002), Sino-Singapore Joint Research Institute, Research on Key Technologies of Biometric Identity Authentication for Financial Services(206-A017023), Natural Science Foundation of Guangdong Province(2017A030310320) and Educational Commission of Guangdong Province of China(2017KTSCX132).

References

[1] P. Bas, T. Filler, and T. Pevn. *Break Our Steganographic System: The Ins and Outs of Organizing BOSS*. Springer Berlin Heidelberg, 2011. 4

[2] P. Bas and T. Furon. Bows-2, 2007. 4

[3] R. Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998. 2

[4] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Computer Vision and Pattern Recognition*, pages 3150–3158, 2016. 2

[5] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *IEEE International Workshop on Information Forensics and Security*, pages 48–53, 2014. 1, 6

[6] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics & Security*, 7(3):868–882, 2012. 1

[7] R. Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 2

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research*, 9:249–256, 2010. 4

[10] V. Holub and J. Fridrich. Designing steganographic distortion using directional filters. In *IEEE International Workshop on Information Forensics and Security*, pages 234–239, 2012. 1, 4

[11] V. Holub, J. Fridrich, and T. Denemark. Universal distortion function for steganography in an arbitrary domain. *Eurasip Journal on Information Security*, 2014(1):1, 2014. 1, 4

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. pages 448–456, 2015. 2

[13] T. D. Jessica Fridrich, Vojtech Holub. Feature extractors for steganalysis. http://dde.binghamton.edu/download/feature_extractors/, 2016. 5

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014. 4

[15] J. Kodovsky, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics & Security*, 7(2):432–444, 2012. 1

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012. 1

[17] B. Li, M. Wang, J. Huang, and X. Li. A new cost function for spatial image steganography. In *IEEE International Conference on Image Processing*, pages 4206–4210, 2015. 1, 4

[18] B. Li, W. Wei, A. Ferreira, and S. Tan. Rest-net: Diverse activation modules and parallel subnets-based cnn for spatial image steganalysis. *IEEE Signal Processing Letters*, 25(5):650–654, 2018. 1, 2

[19] K. Liu, J. Yang, and X. Kang. Ensemble of cnn and rich model for steganalysis. In *International Conference on Systems, Signals and Image Processing*, pages 1–5, 2017. 1, 2

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1, 3

[21] T. Pevn, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. *Lecture Notes in Computer Science*, 6387:161–177, 2010. 1

[22] Y. Qian, J. Dong, W. Wang, and T. . Tan. Learning and transferring representations for image steganalysis using convolutional neural network. In *IEEE International Conference on Image Processing*, pages 2752–2756, 2016. 1, 2, 4

[23] Y. Qian, J. Dong, W. Wang, and T. Tan. Deep learning for steganalysis via convolutional neural networks. *Proceedings of SPIE - The International Society for Optical Engineering*, 9409:94090J–94090J–10, 2015. 1, 2

[24] S. Ruder. An overview of multi-task learning in deep neural networks. 2017. 2

[25] L. I. Sijin, Z. Q. Liu, and A. B. Chan. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 488–495, 2014. 2

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014. 2

[27] S. Tan and B. Li. Stacked convolutional auto-encoders for steganalysis of digital images. In *Signal and Information Processing Association Summit and Conference*, pages 1–4, 2014. 1

[28] G. Xu, H. Z. Wu, and Y. Q. Shi. Ensemble of cnns for steganalysis: An empirical study. In *ACM Workshop on Information Hiding and Multimedia Security*, pages 103–107, 2016. 1, 2

[29] G. Xu, H. Z. Wu, and Y. Q. Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, 2016. 1, 2

[30] J. Yang, K. Liu, X. Kang, E. Wong, and Y. Shi. Steganalysis based on awareness of selection-channel and deep learning. pages 263–272, 2017. 1, 2

[31] J. Ye, J. Ni, and Y. Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics & Security*, 12(11):2545–2557, 2017. 1, 2, 3, 4, 5, 6

[32] M. Yedroudj, F. Comby, and M. Chaumont. Yedrouj-net: An efficient cnn for spatial steganalysis. *arXiv preprint arXiv:1803.00407*, 2018. 1, 2

[33] X. Yu, H. Liang, M. Li, and C. T. Li. An adaptive tri-pixel unit steganographic algorithm using the least two significant bits. In *International Workshop on Biometrics and Forensics*, pages 1–6, 2017. 1

[34] Z. Zhang, P. Luo, C. L. Chen, and X. Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108, 2014. 2