JPEG Steganalysis Based on DenseNet

Jianhua Yang, *Student Member, IEEE,* Yun-Qing Shi, *Life Fellow, IEEE,*
Edward K.Wong, *Senior Member, IEEE,* Xiangui Kang*, *Senior Member, IEEE*

Abstract—Different from the conventional deep learning work based on an images content in computer vision, deep steganalysis is an art to detect the secret information embedded in an image via deep learning, pose challenge of detection weak information invisible hidden in a host image thus learning in a very low signal-to-noise (SNR) case. In this paper, we propose a 32-layer convolutional neural Networks (CNNs) in to improve the efficiency of preprocess and reuse the features by concatenating all features from the previous layers with the same feature-map size, thus improve the flow of information and gradient. The shared features and bottleneck layers further improve the feature propagation and reduce the CNN model parameters dramatically. Experimental results on the BOSSbase, BOWS2 and ImageNet datasets have showed that the proposed CNN architecture can improve the performance and enhance the robustness. To further boost the detection accuracy, an ensemble architecture called as CNN-SCA-GFR is proposed, CNN-SCA-GFR is also the first work to combine the CNN architecture and conventional method in the JPEG domain. Experiments show that it can further lower detection errors. Compared with the state-of-the-art method XuNet [1] on BOSSbase, the proposed CNN-SCA-GFR architecture can reduce detection error rate by 5.67% for 0.1 bpnzAC and by 4.41% for 0.4 bpnzAC while the number of training parameters in CNN is only 17% of what used by XuNet. It also decreases the detection errors from the conventional method SCA-GFR by 7.89% for 0.1 bpnzAC and 8.06% for 0.4 bpnzAC, respectively.

Index Terms—Adaptive steganography, JPEG steganalysis, convolutional neural Networks (CNNs), ensemble.

I. INTRODUCTION

Image steganography is a technology that can hide secret messages in images for covert communication. It includes two research categories: spatial domain and JPEG domain. Current steganography methods for both domains have become more and more sophisticated by embedding messages into complex texture regions. In the spatial domain, there are some methods, such as spatial version of UNIWARD (S-UNIWARD) [2], HILL [3], and MiPOD [4]. Whereas in the JPEG domain, there are some examples, such as UED [5], UERD [6] and JPEG version of UNIWARD (J-UNIWARD) [2].

With the development of steganography, steganalysis has also made substantial progress to detect hidden messages in a suspicious image. In the spatial domain, prevailing algorithms

J. Yang, X. Kang are with Guangdong Key Lab of Information Security, School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China 510006, (e-mail: isskxg@mail.sysu.edu.cn).

Y. Shi is with the department of ECE, New Jersey Institute of Technology, Newark, NJ, USA 07102, (e-mail:shi@njit.edu).

E. Wong is with the department of Computer Science and Engineering, New York University, Tandon School of Engineering, Brooklyn, NY 11201, (e-mail:ewong@nyu.edu).

calculate residuals from 30 high pass filters, then quantized and truncated the residuals to [-T, T] for co-occurrence operator [7–9]. In the JPEG domain, popular algorithms, such as DCTR [10], GFR [11], PHARM [12] and the relevant versions based on selection-channel-aware [13] focus on extracting features from the decompressed image. Both in spatial domain and JPEG domain, the ensemble classifiers [14] is applied for binary classification.

Recently, image steganalysis has made progress by using convolutional neural networks (CNNs) in the spatial domain [15–24]. In work [20], Xu *et al.* find that use Tanh activation to replace ReLU can ensure the features from high pass filters located on the quasi-linear region. In [24], Ye *et al.* used Truncation Linear Layer (TLU) to accelerate the convergence in spatial steganalysis. However, there are very few works in the JPEG domain [1, 25, 26].

In [25], Zeng *et al.* proposed a hybrid deep-learning structure based on the large-scale ImageNet dataset. In the preprocessing phase, they used hand-crafted convolutional layers with twenty-five 5×5 DCT patterns from DCTR [10] for decompressed image, then performed quantization and truncation as in conventional steganalysis methods.

In [26], Chen *et al.* proposed a phase-split module to consider the influence of JPEG-phase. To suppresses the image content and increase the high frequency stego signal, four 5×5 high pass filters, which include a "KV filter", a "point filter", and 2 Gabor filters were used. Experimental results showed that different kinds of high pass filters can complement each other and the detection performance will be improved by using JPEG phase awareness.

In [1], Xu proposed a 20-layer CNN structure based on residual net (ResNet) [27], he indicated that both the pooling method and network depth are critical for steganalysis. All of the pooling layers were performed by 3×3 convolutional with a stride of 2. To make the CNN architecture deeper, the shortcut connection was incorporated. Experimental results have showed that this method can obtain better results than CNN proposed in [25] and conventional method selection-channel aware Gabor filter residuals (SCA-GFR) [28]. Thus, we refer to it as XuNet in the rest of this paper.

In this paper, we study the preprocess and feature reuse for JPEG deep steganalysis. The features can be reused by concatenating all previous layers with matching feature size similiar to the method of Dense Convolutional Network (DenseNet) [29]. To further boost the detection accuracy, we propose an ensemble method called CNN-SCA-GFR by combining the CNN architecture and conventional method SCA-GFR.

The rest of this paper is organized as follows. Details of

the proposed CNN architecture are described in Section 2. Experiments and analysis are given in Section 3. Section 4 introduces the proposed ensemble method CNN-SCA-GFR. The conclusion and future works are described in Section 5.

II. THE PROPOSED JPEG STEGANALYSIS ARCHITECTURE

The whole architecture are shown on Table I. It contain the preprocess layers, features reuse layers and classifier layers. The preprocess layers includes high pass filtering and truncate process. Features reuse layers were combined by the convolution-batch normalization-ReLU layers. The classifier layers contains a fully-connected layer and a Softmax layer to obtain class probabilities of cover/stego.

TABLE I: The proposed CNN architecture. Note that each "conv" layer shown in the table corresponds to the sequence Conv-BN-ReLU. The size of the convolutional kernels follows the form: (number of kernels $\times$ height $\times$ width).

Group	Output size	Process	Times
Group 1	256×256	High pass filtering	$\times 1$
Group 2	256×256	Truncation	$\times 1$
Group 3	128×128	$32 \times (3 \times 3)$ conv (stride 1) $64 \times (3 \times 3)$ conv (stride 2)	$\times 1$
Group 4	64×64	$96 \times (1 \times 1)$ conv (stride 1) $32 \times (3 \times 3)$ conv (stride 1)	$\times 2$
		$128 \times (1 \times 1)$ conv (stride 1) $96 \times (3 \times 3)$ conv (stride 2)	$\times 1$
Group 5	32×32	$96 \times (1 \times 1)$ conv (stride 1) $32 \times (3 \times 3)$ conv (stride 1)	$\times 2$
		$128 \times (1 \times 1)$ conv (stride 1) $96 \times (3 \times 3)$ conv (stride 2)	$\times 1$
Group 6	16×16	$96 \times (1 \times 1)$ conv (stride 1) $32 \times (3 \times 3)$ conv (stride 1)	$\times 2$
		$128 \times (1 \times 1)$ conv (stride 1) $96 \times (3 \times 3)$ conv (stride 2)	$\times 1$
Group 7	8×8	$96 \times (1 \times 1)$ conv (stride 1) $32 \times (3 \times 3)$ conv (stride 1)	$\times 2$
		$128 \times (1 \times 1)$ conv (stride 1) $96 \times (3 \times 3)$ conv (stride 2)	$\times 1$
Group 8	1×1	$96 \times (1 \times 1)$ conv (stride 1) $32 \times (3 \times 3)$ conv (stride 1)	$\times 2$
		8×8 global average pooling	$\times 1$
Group 9	1×1	2D fully-connected Softmax	$\times 1$

A. High pass filtering

In steganalysis, the embedded signal, i.e., hidden information, is very weak compared with the image content, which is regarded as the embedding noise. That is, it is learning in a very low signal-to-noise (SNR) case. In order to make the CNN architecture concentrate on the weak embedding signals rather than the strong image contents, the JPEG image was decompressed into spatial domain without rounding off the pixel values to integers, then the decompressed image is high-pass filtered via convolution with sixteen high-pass filters which are initialized to DCT basis patterns. The DCT basis patterns are initialized as follows:

$$B_{mn}^{(k,l)} = \frac{w_k w_l}{4} \cos(\frac{k\pi(2m+1)}{8}) \cos(\frac{l\pi(2n+1)}{8}) \quad (1)$$

$w_0 = 1$, $w_x = \frac{1}{\sqrt{2}}$ for $x > 0$; $1 \leq k, l \leq 4, 1 \leq m, n \leq 4$.

After initialization, here in this work, these high-pass filters are optimized through training with other filter parameters rather than fixed [24].

B. Truncation Layer

In this work, the value of the feature maps (x) generated by high pass filters are first truncated with a threshold value of T = 8. TLU [24] activation function is applied as follows:

$$f(x) = \begin{cases} -T & x < -T \\ x & -T \leq x \leqslant T \\ T & x > T. \end{cases} \quad (2)$$

It is observed that applying TLU function can obtain best results in JPEG steganalysis, it is different from the conventional deep learning in computer vision which popularity applies ReLU or Tanh activation function.

C. Feature reuse

Motivated by the deep learning work of DenseNet [29] in conventional computer vision, in this low SNR deep learning work, the features learned from previous layers are also reused by concatenation layers. The feature-maps from all of the preceding layers can be used by all subsequent layers, thus the flow of information and gradients throughout the network are improved.

Eq. (3) illustrates the dense connection manner, where the ℓth layer receives the feature-maps of all preceding layers $x_0, ..., x_{\ell-1}$ as input:

$$x_\ell = H_\ell([x_0, x_1, ..., x_{\ell-1}]) \quad (3)$$

where $[x_0, ..., x_{\ell-1}]$ refers to the concatenation of the feature-maps produced in layers $0, ..., \ell-1$. H_ℓ represents the serial process of layer ℓ with Convolution, Batch Normalization[30], ReLU (Conv-BN-ReLU).

Figure. 1 show the Group 4 of the proposed CNN architecture. Please note that we doesn't apply the DenseNet directly, the architecture is varied according to the target of low SNR steganalysis task. All of the average pooling layers were also replaced by convolution layers with stride 2 to propagate the weakly signal. Because of the GPU memory, each convolution layer is followed by a Batch Normalization (BN) and ReLU function (Conv-BN-ReLU), we also doesn't conduct the concatenate process on Group 3.

Comparing with current best steganalysis method XuNet, the main difference are as follows.

1) During the preprocess phase, the parameters of sixteen DCT kernels are updated in our method while the parameters are fixed in XuNet.

2) We ignore the absolution layer after the truncation process.

3) The 1×1 bottleneck layer and transition layer have been employed to improve computation efficiency and decrease the number of parameters.
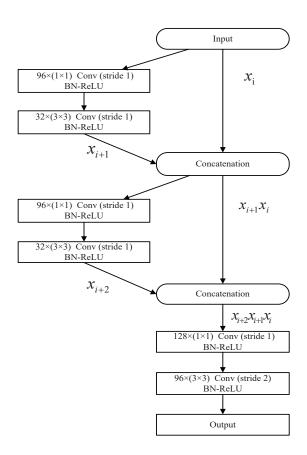
Fig. 1: The concatenating process of Group 4.

4) The proposed CNN architecture reuses the features by concatenating all previous layers with the same feature-map size. In XuNet, the identity features and the output of non-linear transformation are combined by performing element-wise addition, so that the propagation of information will be impeded and lead to optimization problems [29].

III. EXPERIMENTS AND ANALYSIS

A. Platform and Hyperparameters Settings

Caffe toolbox [31] was selected to implement the proposed CNN architecture. Parameters were updated by stochastic gradient descent (SGD). A mini-batch of 32 images with 16 cover-stego pairs were used as the input for each training iterations. The momentum was set to 0.9 and the learning rate was initialized with a value 0.001, then divided by 5 every 30,000 iterations. The convolutional kernels were initialized using a zero-mean Gaussian distribution with a standard deviation 0.01, except that the fully connected layers were initialized using "Xavier" initialization. The biases are initialized to 0.2 in the 3×3 convolutions and disabled in the 1×1 convolutions.

B. Datasets

BOSSbase. We use the standard dataset Bossbase v1.01 [32] which contains 10,000 cover images. All of the images have been resampled to a size of 256×256 by using "imresize()"

function in Matlab and then JPEG compressed by using "imwrite()" function with a quality factor 75. We selected 6,000 images for training and the remaining 4,000 images were used for validation. All images in the training set were randomly horizontally mirrored and rotated by a multiple of 90 degrees for data augmentation.

BOWS2. The 10,000 images from dataset BOWS2 [33] were resampled to size of 256×256 and then JPEG compressed with a quality factor 75 as processed on BOSSbase. All 10,000 images were used for testing the performance of models which trained by dataset BOSSbase.

ImageNet. In order to verify the performance on large scale dataset, 500,000 images were randomly selected from ImageNet ILSVRC 2013 classification dataset [34]. We use 80% images for training, 10% for validation and 10% for testing. All the images were cropped from the left-top region of the original image to size 256×256, then convert to grayscale and JPEG recompressed with a quality factor 75.

C. Classification Results on BOSSbase

Due to computational constraints in our facilities, we only compare our results with XuNet and the conventional steganalysis methods SCA-GFR for J-UNIWARD with payloads ranging from 0.1 bpnzAC to 0.4 bpnzAC. We trained the CNN architecture 3 times separately with 120,000 iterations, saved the trained models every 5,000 iterations. We select the last 3 saved models from each experiment, and report the ensemble results from 9 trained models to reduce the impact of random variations. The validation error rates from the ensemble results are shown in Table II. It can be seen that the proposed architecture can reduce the detection errors dramatically than XuNet and SCA-GFR.

TABLE II: Detection error rates (%) for J-UNIWARD on BOSSbase.

Payload	Proposed	XuNet [1]	SCA-GFR [28]
0.4 bpnzAC	**10.84**	14.16	17.81
0.3 bpnzAC	**16.61**	20.30	25.22
0.2 bpnzAC	**25.72**	29.27	33.91
0.1 bpnzAC	**37.30**	41.63	43.85

In order to investigate the influence caused by different parts of the proposed CNN architecture, we vary the CNN architecture as follows:
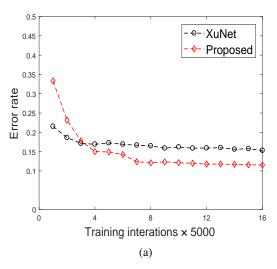
Variant #1: Replace all of the concatenation layer with added layer;

Variant #2: Delete all of the $96 \times (1 \times 1)$ bottleneck layer and $128 \times (1 \times 1)$ transition layers;

Variant #3: Replace all of the $96 \times (1 \times 1)$ bottleneck layer and $128 \times (1 \times 1)$ transition layers with $96 \times (3 \times 3)$ and $128 \times (3 \times 3)$ convolution layers;

Variant #4: Fixed the parameters of the sixteen DCT kernels during the training stage.

Variant #5: Add an absolute layer after the high pass filters.

The validation errors after 80,000 iterations on BOSSbase for J-UNIWARD at 0.4 bpnzAC are shown in Figure 2. The
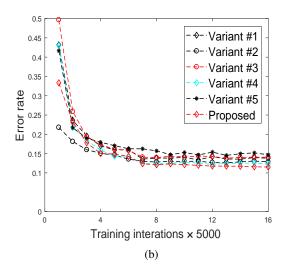
Fig. 2: Detection errors of different methods on Bossbase for J-UNIWARD at 0.4 bpnzAC. (a) the comparison of the proposed method and XuNet, (b) different variants of the proposed method.

lowest validation errors for comparison have been shown in Table III. It can be seen that the proposed method can obtain better performance than XuNet and different variants.

TABLE III: Detection error (%) for different variants on the BOSSbase for J-UNIWARD at 0.4 bpnzAC.

Proposed	XuNet	# 1	# 2	# 3	# 4	# 5
11.51	15.30	13.39	12.60	13.83	12.21	14.54

D. Analysis

From variant #1, replace all of the concatenation layer with added layer will decreased the detection accuracy. This process will make the propagation of information be impeded and lead to optimization problems [29].

From variant #2 and #3, delete all of the 1×1 bottleneck layer and transition layer or replaced by 3×3 convolutions will also decreased the detection accuracy, this result may caused by the dramatically increment of training parameters and more harder to train.

From variant #4, update the DCT kernels during the training stage will improve the detection performance. This process may make the high pass filters more adaptively.

From variant #5, it's interesting that add an absolute layer will increase the detection accuracy for spatial steganalysis [20], but the performance will decreased for JPEG steganalysis.

Because the feature reuse and the utilization of bottleneck layer and transition layer, the architecture become more compactly. Thus the parameter is only 17% of what used by XuNet (0.88 M versus 5.48 M).

E. Classification Results on BOWS2

In order to investigate the robustness of the proposed CNN architecture, we test detection errors on the BOWS2 dataset for

models trained on the BOSSbase dataset. Experimental results are presented in Table IV, from which it can be observed that the proposed method is robust for different datasets and can obtain better performance than XuNet and SCA-GFR.

TABLE IV: Detection error rates (%) for J-UNIWARD on BOWS2.

Payload	Proposed	XuNet [1]	SCA-GFR [28]
0.4 bpnzAC	**15.05**	18.41	21.05
0.3 bpnzAC	**21.70**	25.02	28.07
0.2 bpnzAC	**29.91**	33.92	35.55
0.1 bpnzAC	**40.26**	44.19	43.99

F. Classification Results on ImageNet

We trained models 280,000 iterations for 0.4 bpnzAC, results are shown in Figure 3. The models for 0.2 bpnzAC were fine-tuned from the models optimized from 0.4 bpnzAC [19].

Test results based on models with best validation accuracy are shown in Table V. It is noted that the performance in terms of detection accuracy will be enhanced on the large scale dataset ImageNet if the batch size can be increased. To be fair and considering the constraint of GPU memory, here in the experiments, we only input 20 cover-stego pairs for both the XuNet and the proposed CNN architecture. It has been observed from Table V that the proposed method can also achieve lower error rates on large scale dataset.

TABLE V: Detection error (%) compared on ImageNet for J-UNIWARD.

Algorithm	0.2 bpnzAC	0.4 bpnzAC
Proposed	**33.04**	**15.34**
XuNet[1]	38.68	22.28

TABLE VI: Detection error rates (%) of CNN-SCA-GFR on BOSSbase for J-UNIWARD.

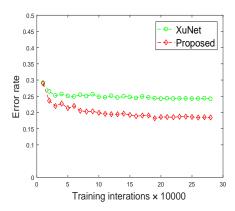| Payload	CNN-SCA-GFR	XuNet [1]	SCA-GFR [28]	Proposed CNN
0.4 bpnzAC	**9.75**	14.16	17.81	10.84
0.3 bpnzAC	**15.7**	20.30	25.22	16.61
0.2 bpnzAC	**24.25**	29.27	33.91	25.72
0.1 bpnzAC	**35.96**	41.63	43.85	37.30

Fig. 3: Comparison of validation error rate vs training iterations between the proposed CNN and XuNet [1] on ImageNet for J-UNIWARD at 0.4 bpnzAC.

IV. ENSEMBLE CNN WITH CONVENTIONAL METHOD

Here, we investigate the ensemble approach to combine the proposed CNN architecture with the conventional JPEG steganalysis method SCA-GFR on BOSSbase dataset. The proposed ensemble architecture is shown in Figure 4, we set $i = 9$ and $k = 6$ by considering the diversity and complexity. For each trained CNN model, 160 dimensional intermediate features are extracted from average pooling layer [21]. Then, 1,440 dimensional features from 9 CNNs are concatenated to generate final dimensional features. From SCA-GFR, 17,000 dimensional features are calculated from selection-channel aware variant of GFR feature extractors. Calculate the probability P_0 to P_6 by using the ensemble classifier. The final class probabilities can be merged by taking the average of 7 probabilities, thus determined the image as cover or stego.

Different from previous ensemble method in the spatial domain which obtain the probability from Softmax layer [22] in CNN architecture, here in this work, intermediate features in CNN architecture are used as inputs to the ensemble classifier to obtain probabilities for further process. We call this method CNN-SCA-GFR.

Experimental results of CNN-SCA-GFR by analysing J-UNIWARD are shown in Table VI. It can be seen that the proposed method can decrease the detection errors from XuNet by 5.67% for 0.1 bpnzAC and by 4.41% for 0.4 bpnzAC, respectively. It also decreases the detection errors from the conventional method SCA-GFR by 7.89% for 0.1 bpnzAC and 8.06% for 0.4 bpnzAC, respectively.

V. CONCLUSION

In this paper, we have proposed an architecture to detect the JPEG steganalysis signal on low signal-to-noise (SNR) situation. The proposed 32-layer CNN architecture can improve the efficiency of preprocess and reuse the features by concatenating all of the previous layers with the same feature-map size, thus improve the flow of information and decrease the training parameters dramatically. We have also proposed the CNN-SCA-GFR method by combining the proposed CNN architecture and the conventional methods SCA-GFR. Our method is the first time to combine the CNN architecture and conventional method in the JPEG domain. Experiments show that CNN-SCA-GFR can further lower detection errors.

In the future, we will incorporate the selection channel aware (SCA) into the CNN architecture as in conventional method for JPEG steganalysis. We will also improve the memory efficient of the proposed CNN architecture. It's also an interesting work to improve the feature propagation for steganalysis in the spatial domain.

REFERENCES

[1] Guanshuo Xu. Deep convolutional neural network to detect j-uniward. In *5th ACM Workshop Inf. Hiding Multimedia Secur. (IH&MMSec)*, 2017.

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

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

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

[5] Linjie Guo, Jiangqun Ni, and Yun Qing Shi. An efficient jpeg steganographic scheme using uniform embedding. In *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*, pages 169–174. IEEE, 2012.

[6] Linjie Guo, Jiangqun Ni, and Yun Qing Shi. Uniform embedding for efficient jpeg steganography. *IEEE transactions on Information Forensics and Security*, 9(5):814–825, 2014.

[7] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.
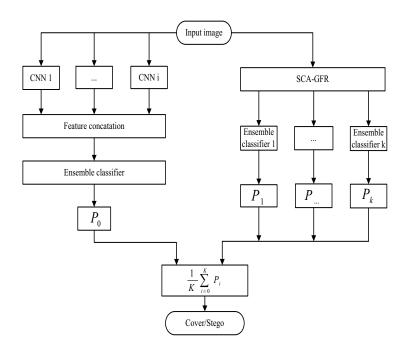
Fig. 4: Block diagram of the proposed CNN-SCA-GFR architecture.

[8] Weixuan Tang, Haodong Li, Weiqi Luo, and Jiwu Huang. Adaptive steganalysis against wow embedding algorithm. In *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, pages 91–96. ACM, 2014.

[9] Tomas Denemark, Vahid Sedighi, Vojtech Holub, Rémi Cogranne, and Jessica Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*, pages 48–53. IEEE, 2014.

[10] V. Holub and J. Fridrich. Low-complexity features for jpeg steganalysis using undecimated dct. *IEEE Transactions on Information Forensics and Security*, 10(2):219–228, Feb 2015.

[11] Xiaofeng Song, Fenlin Liu, Chunfang Yang, Xiangyang Luo, and Yi Zhang. Steganalysis of adaptive jpeg steganography using 2d gabor filters. In *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pages 15–23. ACM, 2015.

[12] Vojtech Holub and Jessica J Fridrich. Phase-aware projection model for steganalysis of jpeg images. In *Media Watermarking, Security, and Forensics*, page 94090T, 2015.

[13] Tomáš Denemark Denemark, Mehdi Boroumand, and Jessica Fridrich. Steganalysis features for content-adaptive jpeg steganography. *IEEE Transactions on Information Forensics and Security*, 11(8):1736–1746, 2016.

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

[15] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. *Media Watermarking, Security, and Forensics*, 9409:94090J–94090J, 2015.

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

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

[18] Vahid Sedighi and Jessica Fridrich. Histogram layer, moving convolutional neural networks towards feature-based steganalysis. *Electronic Imaging*, 2017(7):50–55, 2017.

[19] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Learning and transferring representations for image steganalysis using convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2752–2756. IEEE, 2016.

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

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

[22] Kai Liu, Jianhua Yang, and Xiangui Kang. Ensemble of cnn and rich model for steganalysis. In *Systems, Signals and Image Processing (IWSSIP), 2017 International*

Conference on, pages 1–5. IEEE, 2017.

[23] Jianhua Yang, Kai Liu, Xiangui Kang, Edward Wong, and Yunqing Shi. Steganalysis based on awareness of selection-channel and deep learning. In *International Workshop on Digital Watermarking*, pages 263–272. Springer, 2017.

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

[25] Jishen Zeng, Shunquan Tan, Bin Li, and Jiwu Huang. Large-scale jpeg steganalysis using hybrid deep-learning framework. *arXiv preprint arXiv:1611.03233*, 2016.

[26] Mo Chen, Vahid Sedighi, Mehdi Boroumand, and Jessica Fridrich. Jpeg-phase-aware convolutional neural network for steganalysis of jpeg images. In *5th ACM Workshop Inf. Hiding Multimedia Secur. (IH&MMSec)*, 2017.

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

[28] T. D. Denemark, M. Boroumand, and J. Fridrich. Steganalysis features for content-adaptive jpeg steganography. *IEEE Transactions on Information Forensics and Security*, 11(8):1736–1746, Aug 2016.

[29] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[30] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[31] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[32] Patrick Bas, Tomáš Filler, and Tomáš Pevný. break our steganographic system: The ins and outs of organizing boss. In *Information Hiding*, pages 59–70. Springer, 2011.

[33] P. Bas and T. Furon. Bows-2. In *http://bows2.ec-lille.fr.*, 2007.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[35] Geoff Pleiss, Danlu Chen, Gao Huang, Tongcheng Li, Laurens van der Maaten, and Kilian Q Weinberger. Memory-efficient implementation of densenets. *arXiv preprint arXiv:1707.06990*, 2017.