Non-structured Pruning for Deep-learning based Steganalytic Frameworks

Qiushi Li*, Zilong Shao*, Shunquan Tan*, Jishen Zeng†, and Bin Li†
* College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
† College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China
*† Shenzhen Key Laboratory of Media Security,
Guangdong Key Laboratory of Intelligent Information Processing,
National Engineering Laboratory for Big Data System Computing Technology,
Shenzhen Institute of Artificial Intelligence and Robotics for Society,
Shenzhen 518060, China.

***Abstract*—Image steganalysis aims to discriminate innocent cover images and those suspected stego images embedded with secret message. Recently, increasing advanced deep neural networks have been proposed and used in image steganalysis. Though those deep learning models can gain superior performance, they also result in redundancy of computational resource and memory storage. In this paper, we apply a non-structured pruning method to prune XuNet2 and SRNet — the two state-of-the-art deep-learning framework in the field of JPEG image steganalysis. We obtain the priorities of the connections among neurons according to a certain criterion, then keep those significant weights and prune those nonsignificant ones in the meantime. We have conducted extensive experiments on BOSSBase and BOWS image dataset. The experimental results demonstrate that our proposed non-structured pruning method can significantly reduce the cost of computation and storage required by the original deep-learning frameworks without affecting their detection accuracy.**

I. Introduction

Image steganography is a popular convert communication technology. The sender tries to transmit secret message to the receiver by embedding it into some innocent cover images. Due to the insensitivity of human eyes to noise in texture regions, it is hard to discover embedded message from those stego images. The most widely used steganographic algorithms today are Hill [1], S-UNIWARD [2] in spatial domain and UERD [3], J-UNIWARD [2] in JPEG domain.

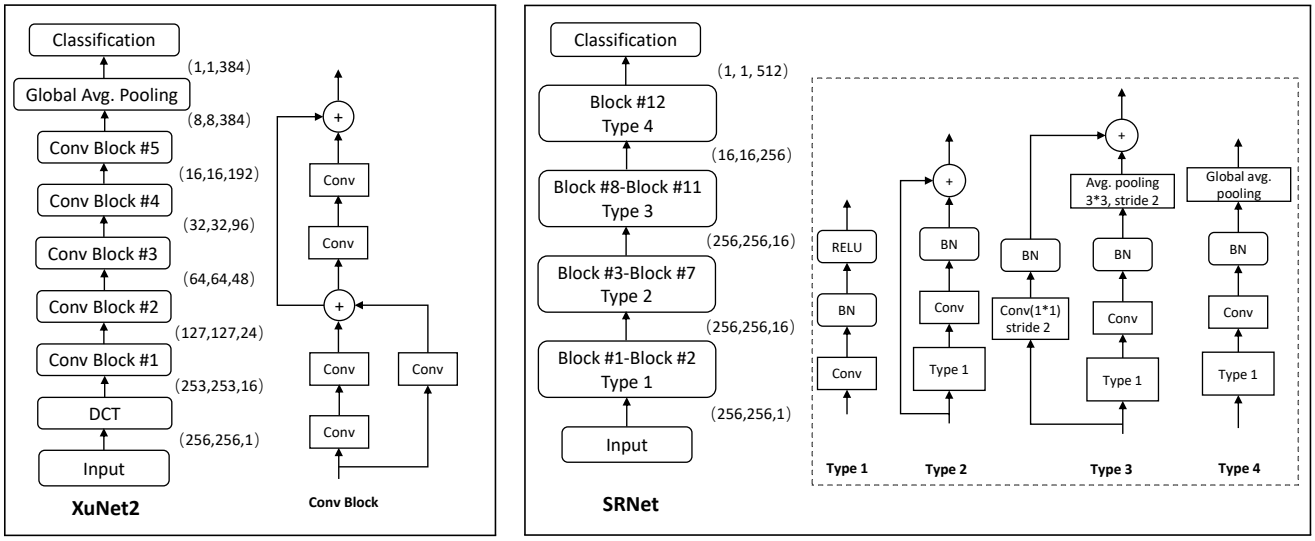
As the rival to steganography, steganalysis has become popular with the development of steganographic algorithms. Steganalysis determines whether an image hides secret information by analyzing the statistical characteristics of a suspicious image, and even extracts the secret information if possible. Steganalysis is generally regarded as a two-class classification task. Existing steganalytic schemes firstly suppress image contents and therefore improve the signal-to-noise ratio between the stego signal and the corresponding host image, which is useful for extracting the characteristic of stego modifications. The most famed traditional steganalytic scheme is the so-called "rich model" [4], [5]. With the rapid development of deep learning in the field of pattern recognition, quite a few researchers have proposed to apply this technology to steganalysis. In 2014, Tan and Li [6] proposed the first deep-learning steganalytic framework. Then in [7], [8], [9], deep-learning based, especially CNN based steganalytic models gradually surpassed spatial-domain rich models. In JPEG domain, XuNet2 [10] is a deep and complex steganalytic framework with very competitive performance. The SRNet [11] proposed by Boroumand et al. is the first completely end-to-end deep-learning steganalytic framework. As far as we know, it is the best performing JPEG steganalyzer at present.

However, since there is no universal guideline for the design of the structure of deep-learning frameworks, it is highly probable that a mass of parameters/weights in existing deep-learning frameworks are redundant. They achieve superior detection performance at the cost of heavy computational resources and enormous memory storage. Therefore, the research on compression algorithms of deep-learning models attracts notable attention of researchers. Deep-learning steganalytic frameworks are no exception. The non-structured weight pruning method is a major branch of deep-learning model compression. In a very early stage of the evolution of deep-learning frameworks, Lecun et al. [12] have proposed the method of pruning weights inspired by brain science. In 2015, Han et al. [13] proposed a distinguished non-structured pruning scheme which removes individual weights below a given threshold. Han's scheme can result in compact and sparse weight matrices without affecting the performance of the target deep-learning model.

In this paper, we apply Han's non-structured pruning scheme to XuNet2 and SRNet—the two most advanced deep-learning models in the field of JPEG steganalysis, and use the pruned models to detect JPEG stego images generated by UERD and J-UNIWARD, the two popular JPEG steganographic algorithms. Experimental results demonstrate that our approach can effectively sparse weight matrices of the target model as well as maintain their detection performance.

The remainder of this paper is organized as follows. In

This work was supported in part by the NSFC (61772349, U1636202, 61872244), Shenzhen R&D Program (JCYJ20160328144421330). *(Corresponding author: Shunquan Tan)*



(a) (b)

Fig. 1. The architecture of the two target deep-learning steganalytic models. The sizes of the output features maps (height, width, number of channels) are displayed besides the network diagram. (a) The architecture of XuNet2. The detailed structure of the "Conv Block" is shown on the right. (b) The architecture of SRNet. The detailed structures of four types of blocks are shown on the right, surrounded by dotted-line box.

Sect. II, we firstly give a brief introduction to Han's non-structured pruning method, and the architecture of the two target models: XuNet2 and SRNet. Then we present our approach of using Han's scheme to prune XuNet2 and SRNet. The experimental results are detailed in Sect. III. Finally, the concluding remarks are summarized in Sect. IV.

II. Non-structured pruning method for XuNet2 and SRNet

A. Preliminaries

1) Han's non-structured weight pruning method [13]: The object of non-structured pruning method is individual weights as well as corresponding connections between neurons. Han et al. claimed that in a given layer, the smaller the absolute value of the weight is, the less significant the weight becomes. It is owing to the fact that the weights with small absolute values have less impact on the input.

When Han's method is used to prune a given convolutional layer L_i, a pre-defined threshold T_i is set up for it. Then every filter $F_{i,j}$ in L_i is assigned a corresponding mask $M_{i,j}$, where i is the layer index while j is the filter index in L_i. Given a pruning rate γ_i, the process of pruning L_i is as follows:

1) Sort the absolute values of the weights of all the filters $\{F_{i,j}, 1 \leq j \leq m\}$ in L_i from small to large to get the ordered sequence $\{S_k\}$, where $k \in \{1, 2, \cdots, n\}$, m is the number of the filters in L_i, and n is the length of the sequence;
2) Compute the threshold $T_i = S_{\lceil n \times \gamma_i \rceil}$ of the i^{th} layer L_i;
3) Set the corresponding values of $M_{i,j}$ to 0, where the absolute values of weights are less than T_i. Otherwise set the corresponding values of $M_{i,j}$ to 1.

2) The architectures of XuNet2 and SRNet: The architecture of XuNet2 [10] is shown in Fig. 1(a). It is a 20-layer deep steganalytic network with five "Conv Blocks" which in turns equipped with residual shortcuts [14]. The residual shortcuts can accelerate the process of convergence and prevent gradient vanishing. Each "Conv Block" is accordingly contains five convolutional layers with shortcut connections. There is a pre-processing layer at the bottom of the architecture, which contains sixteen 4×4 discrete cosine transformation basis kernels to improve the signal-to-noise ratio before the inputs are fed to the deep-learning model itself. The sizes of all the convolutional kernels in XuNet2 are set to 3×3.

The architecture of SRNet [11] is shown in Fig. 1(b). The whole network consists of twelve blocks with four different block types. Among them, block type 2 and type 3 have residual shortcuts [14]. Functionally, the network can be divided into three parts. The first part (from Block 1 to Block 7) is used for the extraction of noise residuals, which prevents the suppression of stego signals since there is no pooling/down-sampling operations. The second part (from Block 8 to Block 12) is devoted to features extraction and dimensionality reduction. The third part is the top single-layer linear classifier.

B. Our proposed pruning approach

In our proposed pruning approach, we apply Han's non-structured pruning method to prune XuNet2 and SRNet. We follow the well-established three-step deep-learning model pruning pipeline:

1) obtain a converged, over-parameterized steganalytic model after the whole train-validation procedure;
2) prune the over-parameterized steganalytic model with Han's pruning method;

3) fine-tune the pruned steganalytic model with further training epochs.

Majority of the parameters of XuNet2 and SRNet are contained in the convolutional layers. Therefore we only prune the weights in those convolutional filters. For XuNet2, we decide to neglect the bottom pre-processing layer since it only contains fixed hand-crafted filters. For SRNet, we decide to neglect the bottom two blocks (namely Block #1 and Block #2), since our experimental result shows that the weights in them are ultra sensitive to pruning, and non-structured pruning results in severe decline in detection performance of the corresponding steganalytic framework.

For both XuNet2 and SRNet, we regard the convolutional layers belonging to an identical block as a whole. We adaptively set a united pruning rate for all the convolutional layers come from the same block. With a converged steganalytic model, we adopt a trial-and-error strategy. In XuNet2, all of the "Conv Blocks" are involved, while in SRNet, the blocks from #3 to #12 are involved. Every block of the target model is individually pruned with the pruning rate from 0% to 100% with a step of 10%. After every pruning procedure, the resulting model is tested with standalone testing set, and the corresponding detection performance is recorded. The most appropriate pruning rate for every block is determined according to the criterion of "sensitivity" defined as follows:

1) If the detection accuracy steadily decreases along with increasing pruning rate, then we adopt a conservative pruning tactic. The largest pruning rate with the decline of detection accuracy on standalone testing set lower than 5% is selected as the target pruning rate.
2) With increasing pruning rate, if we witness a bluff-type slump ($\geq 5\%$) of the detection accuracy, then the last pruning rate before the slump is selected as the target pruning rate.

After the pruning rates of all of the blocks for a given steganalytic model are determined, we apply the pruning procedure to the original converged model. From bottom to top, we prune the blocks one by one. Then the pruned model undergoes a fine-tune procedure. It is retrained for several epochs to regain the once lost detection performance.

During the pruning procedure, we reject the filter weights whose absolute value is lower than T_i from training and inference, by setting $F_{i,j} = F_{i,j} \cdot M_{i,j}$ and $\nabla F_{i,j} = \nabla F_{i,j} \cdot M_{i,j}$ for every filter in L_i, where $\cdot$ denotes dot product and $\nabla F_{i,j}$ denotes the corresponding gradient of $F_{i,j}$. One demonstration is shown in Fig. 2.

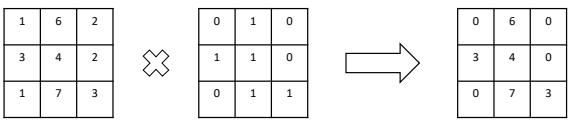


Fig. 2. One example of the filter weights rejection procedure. The cross signal denotes dot product operation.

TABLE I
PRUNING RATE FOR EVERY BLOCK IN XUNET2.

Steganography	*J-UNIWARD*		*UERD*	
bpnzAC	0.4	0.2	0.4	0.2
Block #1	0.6	0.3	0.5	0.5
Block #2	0.7	0.8	0.7	0.6
Block #3	0.6	0.5	0.7	0.6
Block #4	0.9	0.7	0.8	0.8
Block #5	0.9	0.9	0.9	0.9

TABLE II
PRUNING RATE FOR EVERY BLOCK IN SRNET.

Steganography	*J-UNIWARD*		*UERD*	
bpnzAC	0.4	0.2	0.4	0.2
Block #1	0	0	0	0
Block #2	0	0	0	0
Block #3	0.4	0.4	0.4	0.5
Block #4	0.6	0.5	0.6	0.6
Block #5	0.4	0.7	0.6	0.8
Block #6	0.5	0.5	0.5	0.8
Block #7	0.5	0.4	0.6	0.6
Block #8	0.3	0.4	0.5	0.4
Block #9	0.8	0.7	0.5	0.7
Block #10	0.9	0.9	0.9	0.9
Block #11	0.9	0.9	0.9	0.9
Block #12	0.9	0.9	0.9	0.9

III. EXPERIMENTS

All experiments in this paper were conducted on the union of BOSSBase 1.01 [15] and BOWS2 [16]. Each dataset contains 10,000 grayscale images with the size 512×512. We resized all of the images in the dataset from the original size to 256×256 by using MATLAB function "*imresize*", and then compressed them to JPEG format with quality factor 75. The images are divided into three subsets: the training set contains 4,000 images randomly selected from BOSSBase and the entire BOWS2 dataset, a total of 14,000 images; the validation set contains another 1,000 randomly selected images from BOSSBase; the remaining 5,000 images from BOSSBase constitute the testing set. We have four steganographic data embedding configurations used in the experiments: UERD at 0.2 bpnzAC of embedding rate, UERD at 0.4 bpnzAC, J-UNIWARD at 0.2 bpnzAC, and J-UNIWARD at 0.4 bpnzAC.

The XuNet2 was trained with the "Momentum" optimizer with batch size of 100 images (50 cover/stego pairs), and no data augmentation. The filter weights in all convolutional layers were initialized with a zero mean Guassian with standard deviation 0.01 and no bias. The weights and biases in fully connected classifier layer were initialized with zero mean Guassian with standard deviation 0.01. The learning rate started from 0.001 and was decreased by 10% after every 5,000 training iterations. The SRNet was trained with the "Adamax" optimizer with batch size of 16 images (8 cover/stego pairs). The training set was randomly shuffled before every training epoch, and we use data augmentation during the training procedure with two kinds of input processing methods: random mirror and random rotation by 90° degrees. For the initialization of other training parameters please refers to [11].

The implementation of our proposed non-structured network

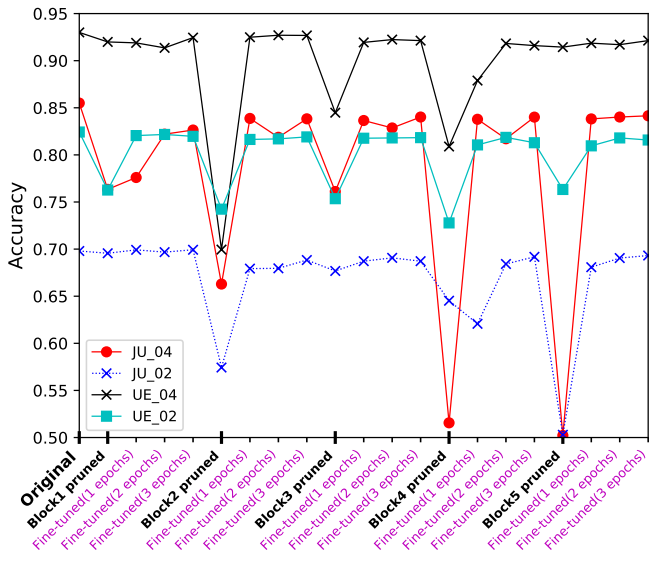


Fig. 3. Accuracy changes along with the pruning and the following three-epoch fine-tuning procedure of every block of XuNet2. "JU-04", "JU-02", "UE-04", and "UE-02" refers to J-UNIWARD of 0.4 bpnzAC embedding rate, J-UNIWARD of 0.2 bpnzAC, UERD of 0.4 bpnzAC, UERD of 0.2 bpnzAC, respectively.

pruning method was based on Tensorflow™ [17]. We applied Han's method to XuNet2 and SRNet as shown in Sect. II-A1. The pruning rates for all the blocks in the two models were adaptively determined with the criterion of "sensitivity" proposed in Sect. II-B. In Tab. I and Tab. II, we show the finally determined pruning rate for every block of XuNet2 and SRNet in the experiments, respectively. It can be seen that for both XuNet2 and SRNet, the original models have quite a few unnecessary filter weights, and the redundancy becomes higher as the blocks become deeper. For instance, in the last two blocks of XuNet2 and the last three blocks of SRNet, the pruning rates reach 0.9, which means that most filter weights of those top blocks are unnecessary and can be rejected from training and inference.

For XuNet2, every time after a block was pruned, we fine-tuned the intermediate model for three epochs. Fig. 3 shows accuracy changes along with the pruning and fine-tuning procedure of every block of XuNet2. We can see in Fig. 3 that the non-structured pruning procedure has huge impact on the detection performance of the resulting model. The detection performance usually decreases dramatically. However, after one or two epochs of retraining/fine-tuning procedure, the detection performance can be regained, and even is approaching to the original model. Tab. III gives a comparison of the detection accuracy of the pruned and retrained model with the original XuNet2 model. We can see that after pruning and retraining, the detection accuracy of the resulting model is almost identical to the original one.

Please note that for SRNet, we only fine-tuned/retrained the model after all of the involved blocks in it were pruned. Tab. IV gives a comparison of the detection accuracy of the

TABLE III
ACCURACY COMPARISON OF XUNET2 BEFORE AND AFTER PRUNING. "*Orig*" MEANS THE ACCURACY OF ORIGINAL MODEL. "*Pruned/Retrained*" MEANS THE ACCURACY AFTER RETRAINING THE PRUNED MODEL, AND "*Decline*" MEANS DESCENT IN DETECTION ACCURACY FROM THE ORIGINAL MODEL TO THE CORRESPONDING PRUNED AND RETRAINED MODEL.

Steganography	*bpnzAC*	*Orig*	*Pruned/ Retrained*	*Decline*
J-UNIWARD	0.4	0.855	0.845	-0.0117
	0.2	0.698	0.691	-0.0100
UERD	0.4	0.930	0.924	-0.0064
	0.2	0.824	0.817	-0.0085

TABLE IV
ACCURACY COMPARISON OF SRNET BEFORE AND AFTER PRUNING. "*Orig*" MEANS THE ACCURACY OF ORIGINAL MODEL. "*Pruned*" MEANS THE ACCURACY OF THE MODEL JUST AFTER PRUNING. "*Retrained*" MEANS THE ACCURACY AFTER RETRAINING THE PRUNED MODEL, AND "*Decline*" MEANS DESCENT IN DETECTION ACCURACY FROM THE ORIGINAL MODEL TO THE CORRESPONDING PRUNED AND RETRAINED MODEL.

Steganography	*bpnzAC*	*Orig*	*Pruned*	*Retrained*	*Decline*
J-UNIWARD	0.4	0.9224	0.5979	0.9215	-0.00097
	0.2	0.7769	0.6084	0.7777	0.00102
UERD	0.4	0.9685	0.6609	0.9676	-0.00092
	0.2	0.8808	0.5040	0.8802	-0.00068

TABLE V
MODEL COMPRESSION RESULT. "*Params*" MEANS THE NUMBER OF THE NON-ZERO WEIGHTS IN THE CONVOLUTIONAL KERNELS OF THE MODEL. THE COMPRESSION RATE IS EQUAL TO THE AMOUNT OF PARAMETERS IN THE ORIGINAL MODEL DIVIDED BY THE AMOUNT OF PARAMETERS IN THE PRUNED MODEL.

Model	Steganography	*bpnzAC*	*Params*	*Compression Rate*
SRNet	*J-UNIWARD*	0.4	73022	65.305
		0.2	74949	63.625
	UERD	0.4	82149	58.049
		0.2	71493	66.701
XuNet2	*J-UNIWARD*	0.4	1447384	3.970
		0.2	1649920	3.483
	UERD	0.4	1156072	4.970
		0.2	1546600	3.715

pruned and retrained model with the original SRNet model. From Tab. IV we can see that without retraining, the detection performance of the pruned model declines severely compared with that of the original one. However, after retraining, the detection accuracy of the resulting model recovers and remains competitive even though a majority of its weights have become zero. Please compare the *Decline* column of Tab. IV with that of Tab. III, the readers can notice that the gap between the performance of the pruned-and-then-retrained SRNet model and the original one is even narrower, which may indicate that our proposed non-structured pruning approach is more efficient for those more advanced and complicated models, like SRNet.

This pruning method realizes the model compression by sparsing weight matrix of the model. The number of weights in convolutional layers of original SRNet and XuNet2 are 4,768,704 and 5,746,720, respectively. As is shown in Tab. V, for Xunet2, pruning reduces the number of weights in convolutional layers by 3×, and for SRNet, pruning reduces the

number of weights in convolutional layers by 60×. Theoretically, the pruned model can decrease the computation cost, but to achieve such an effect the support of specialized hardware or algorithms library is required. With the support of potential specialized hardware/ algorithms library, the compression scheme proposed in this paper can greatly reduce the number of floating-point calculations of the target model and improve the calculation speed.

IV. Conclusion

In this paper, we apply a non-structured pruning method to prune XuNet2 and SRNet — the two state-of-the-art deep-learning framework in the field of JPEG image steganalysis. We combine Han's non-structured weight pruning method with our proposed block based trial-and-error pruning rate decision strategy. Our experimental results show that the proposed pruning method can efficiently reject the nonsignificant weights and hence reduce those redundant connections. After the pruning and retraining procedure, the detection performance of the pruned version of both XuNet2 and SRNet can still be regained. Therefore, our work indicate that the parameters of both XuNet2 and SRNet are with great redundancy. Our future work will focus on exploring more effective specific pruning methods for deep-learning steganalytic frameworks.

References

[1] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proc. IEEE 2014 International Conference on Image Processing, (ICIP'2014)*, 2014, pp. 4206–4210.

[2] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security, Spacial Issue on Revised Selected Papers of the 1st ACM 1H and MMS Workshop*, vol. 2014, no. 1, pp. 1–13, 2014.

[3] L. Guo, J. Ni, W. Su, C. Tang and Y. Shi, "Using statistical image model for JPEG steganography: Uniform embedding revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, 2015.

[4] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[5] T. Denemark, V. Sedighi, V. Holub, R. Cogranne and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images", in *Proc. IEEE International Workshop Information Forensics Security, (WIFS'2014)*, 2014, pp. 48-53.

[6] S. Tan and B. Li, "Stacked convolutional auto-encoders for steganalysis of digital images," in emphProc. AsiaPacific Signal Information Processing Association Annual Summit Conference 2014, (APSIPA'2014), 2014, pp. 1-4.

[7] Y. Qian, J. Dong, W. Wang and T. Tan "Deep learning for steganalysis via convolutional neural networks," *Proc. SPIE*, vol. 9409 pp. 94090J, 2015.

[8] G. Xu, H. Z. Wu, and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[9] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, 2017.

[10] G. Xu, M. Stamm and M. Kirchner, "Deep convolutional neural network to detect J-UNIWARD," *Proc. 5th ACM Workshop Information Hiding Multimedia Security, (IH&MMSec'2017)*, 2017, pp. 67-73.

[11] M. Boroumand, M. Chen and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2019.

[12] Y. LeCun, J. Denker, and S. Solla. "Optimal brain damage," in *Conference and Workshop on Neural Information Processing Systems (NIPS)*, 1990.

[13] S. Han, J. Pool, J, Tran and W. Dally, "Learning both weights and connections for efficient neural network," in *Conference and Workshop on Neural Information Processing Systems (NIPS)*, 2015.

[14] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference Computer Vision and Pattern Recognition, (CVPR'2016)*, 2016, pp. 770-778.

[15] P. Bas, T. Filler, T. Pevny, A. Ker and S. Craver, "Break our steganographic system: The ins and outs of organizing BOSS," in *Proc. 13th International Conference Information Hiding, (IHIP'2011)*, vol. 6958 pp. 59–70 2011.

[16] P. Bas and T. Furon, "BOWS-2," Jul. 2007. Available: http://bows2.ec-lille.fr, [Online].

[17] "TensorFlow™," https://www.tensorflow.org/, [Online].