IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 1, 2020, accepted February 24, 2020, date of publication March 6, 2020, date of current version March 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2978110

A New Convolutional Neural Network-Based Steganalysis Method for Content-Adaptive Image Steganography in the Spatial Domain

ZHILI XIANG, JUN SANG, QIAN ZHANG, BIN CAI, XIAOFENG XIA, AND WEIQUN WU
Key Laboratory of Dependable Service Computing in Cyber Physical Society of Ministry of Education, Chongqing University, Chongqing 400044, China
School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China

Corresponding author: Jun Sang (jsang@cqu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61971073.

ABSTRACT Convolutional neural network-based methods are attracting increasing attention in steganalysis. However, steganalysis for content-adaptive image steganography in the spatial domain is still a difficult problem. In this paper, a new convolutional neural network-based steganalysis approach was proposed with two contributions. 1) By adding more convolutional layers in the lower part of the model, we proposed a new arrangement of convolutional layers and pooling layers, which can process the local information better than the existing CNN models in steganalysis. 2) By adding the global average pooling layer before the softmax layer instead of using global average pooling before the fully connected layer, the global average pooling was placed in a better position for steganalysis. Two state-of-the-art steganographic algorithms in the spatial domain, namely, WOW and S-UNIWARD, were used to evaluate the effectiveness of our model. The experimental results on BOSSbase showed that the proposed CNN could obtain better steganalysis performance than YeNet across all tested algorithms when the payloads were 0.2, 0.3, and 0.4 bpp.

INDEX TERMS Content-adaptive image steganography, convolutional neural networks, local information, steganalysis, spatial domain.

I. INTRODUCTION

Image steganography is the technique of hiding secret data within an ordinary, non-secret image in order to avoid detection. At present, one of the most secure steganographic schemes for digital images is content-adaptive steganography (or adaptive steganography), in which distortion functions are heuristically defined to constrain the embedding changes in those parts of the image that are difficult to model, and then, syndrome-trellis code (STC) is used to minimize the embedding distortion [1]. Adaptive steganography can be applied to both the spatial domain and the JPEG domain. The current typical spatial adaptive steganographic algorithms include HUGO [2], WOW [3], S-UNIWARD [4] and HILL [5].

In steganalysis for spatial adaptive steganography, the spatial rich model (SRM) and its several variants are important methods [6]. The SRM is based on the concept of a rich model, which consists of a large number of diverse submodels formed by joint distributions of neighboring samples, and the samples are obtained by quantizing the image noise residuals, which are computed using linear and nonlinear high-pass filters. The projection spatial rich model (PSRM) is one of the variants of the SRM, and it projects the neighboring residual samples onto a set of random vectors and takes the first-order statistics of the projections as the feature [7]. By using projection, the features in the PSRM can potentially capture the dependencies among a large number of pixels.

Corresponding to the rapid developments in computer vision [8]–[12], convolutional neural networks (CNN) have attracted great interest in steganalysis [13]–[28]. However, steganalysis for content-adaptive image steganography in the spatial domain is still a difficult problem. In their pioneering works, Tan and Li [16] proposed a CNN initialized with unsupervised learning and obtained better results than the subtractive pixel adjacency matrix (SPAM) [29]. Qian *et al.* [17] proposed a CNN with a high-pass filter layer and Gaussian activation function to accommodate the differences between computer vision (CV) tasks and steganalysis.

The associate editor coordinating the review of this manuscript and approving it for publication was Shenghong Li.

Among these components, the high-pass filter layer came from the SRM, which enhanced the weak embedding noise signal and reduced the influence of the image content. In addition, the Gaussian activation function improved the fitting ability of the CNN. This model achieved a result comparable to that of the SRM. Inspired by [17], Xu *et al.* [18], [19] designed a CNN model (denoted as XuNet) for steganalysis in the spatial domain. The model possessed the following characteristics: 1) Batch normalization (BN) [30] layers were used; 2) The absolute value function and tanh activation function were used in the first and second layers; 3) The kernel size of the convolutional layers in the deep part of the networks was decreased, such that the kernel sizes of the first two convolutional layers were 5 × 5 and those of the other convolutional layers were 1 × 1; 4) The kernel sizes of most average pooling layers were set to 5 × 5; and 5) A high-pass filtering layer was used for preprocessing. The results showed that a well-designed CNN had the capacity to achieve better detection performance compared with feature-based steganalysis. Ye *et al.* [20] designed a CNN-based steganalyzer (denoted as YeNet) and obtained superior performance over other steganalysis methods, in which 1) A truncated linear unit (TLU) activation function was introduced. 2) To capture weak steganographic signals, the weights of the first layer were initialized with the 30 basic high-pass filters used in the SRM. 3) A selection-channel-aware structure was used. 4) Data augmentation was adopted to further improve the performance.

By adding statistical moments of the feature maps to the fully connected classifier part of YeNet, Tsang *et al.* proposed a model to better steganalyze images with arbitrary sizes [21]. Couchot *et al.* proposed a hybrid method combing the CNN and SRM + ensemble classifier (EC), which incorporates the advantages of the CNN model for high steganography payloads and the advantages of the SRM + EC for low steganography payloads [22]. Zeng *et al.* proposed a steganalysis method by using multiple CNNs to fit the substructure of the rich feature model [23]. Aiming at the problem of the severe fluctuation of the loss during training, Songtao *et al.* studied the setting of the BN parameters [24]. Yang *et al.* proposed a CNN combined with a channel selection mechanism [25]. Sedighi *et al.* designed a CNN with a Gaussian activation function to implement histogram features [26]. Yedroudj *et al.* proposed a model to combine some advantages of XuNet and YeNet. The model contains 30 high-pass filters, a TLU activation function, fewer convolutional layers, and a scale module [27]. Zeng *et al.* proposed a model for color images. In the structure, different color bands were first passed through independent channels and then merged in a deeper part of the model [28].

In this paper, a new CNN-based steganalysis method is proposed. On the one hand, the arrangement of the convolutional layer and pooling layer suitable for steganalysis is discussed, and in order to capture the weak steganographic signals of images, more convolutional layers are incorporated in the lower part of the network to enhance the ability to process image details. On the other hand, global average pooling (GAP) is introduced to reduce the number of parameters of the networks and enable the networks to simultaneously process different scale inputs. An appropriate position of the global average pooling layer for steganalysis is also discussed. Setting the global pooling layer just before the softmax layer, which was rare in other CNNs, is proposed. In the experiment on the BOSSbase dataset, different steganography methods, namely, WOW and S-UNIWARD, with different payloads are used to evaluate the effectiveness of the proposed model. The results show that the proposed model is mostly better than the current best deep learning-based steganalysis method.

II. THE PROPOSED NETWORKS

In this section, the proposed arrangement of the convolutional layers and pooling layers is introduced. Then, the global average pooling layer and its position are described. Finally, the details of the architecture are presented.

A. ARRANGEMENT OF CONVOLUTIONAL LAYERS AND POOLING LAYERS

In this section, some arrangements of CNN models in computer vision and steganalysis will be compared with the arrangement of the proposed CNN. In fact, there are some common structures for the CNN, such as convolutional layers and pooling layers. A convolutional layer always contains a convolution operation and an activation function to increase the nonlinearity of the network, and sometimes, a BN layer is included before the activation function. The convolution operation usually includes 4 dimensional parameters (kernel_length, kernel_width, input_channels, and output_channels). For the 4 dimensional parameters, kernel_length×kernel_width is the kernel size of a convolutional layer, which is usually 3 × 3, 5 × 5 or 7 × 7. It has been indicated that a convolutional layer with a kernel size of 5×5, 7 × 7 or larger can be replaced by stacking multiple convolutional layers with a kernel size of 3 × 3 [12]. Therefore, to facilitate the comparison between models with different convolutional kernel sizes, when a convolutional layer with a kernel size larger than 3 × 3 is used, we convert the number of convolutional layers into the corresponding number of 3×3 convolutional layers, in which one convolutional layer with a kernel size of 5 × 5 is replaced by 2 layers, and one 7 × 7 convolutional layer is converted into 3 layers. In general, the pooling layer will reduce the scale of the feature maps and result in the loss of detailed image features. Some models use the convolutional layer with a stride=2 to reduce the scale of the feature maps instead of a pooling layer. It can be seen that whether using pooling or a stride=2, the structure from the lower layers to the deeper layers can be divided into multiple parts according to the scale of the feature maps. In this paper, the arrangement of the convolutional layers and pooling layers of the model is denoted as ($n1$, $n2$, $n3$, $n4$, $n5$), where $n1$ represents the number of convolutional layers, in which the scale of the feature maps is equal to that of the

TABLE 1. Differences in the arrangement of VGG, XuNet, YeNet, and the proposed networks.

VGG19 [12]	ResNet34 [11]	XuNet [18]	YeNet [20]	The proposed networks
C(3)×2	C(7)	C(5)×1	C(3)×3	C(3)×4
Pooling	Stride 2	Pooling	Pooling	Pooling
C(3)×2		C(5)×1	C(5)×1	C(3)×4
Pooling	Pooling	Pooling	Pooling	Pooling
C(3)×4	C(3)×6	C(1)×1	C(5)×1	C(1)×1
Pooling	Stride 2	Pooling	Pooling	Pooling
C(3)×4	C(3)×8	C(1)×1	C(5)×1	C(1)×1
Pooling	Stride 2	Pooling	Pooling	Pooling
C(3)×4	C(3)×12	C(1)×1	C(3)×2	C(1)×1

original image. $n2$, $n3$, $n4$ and $n5$ respectively represent the number of convolutional layers, in which the scales of the feature maps are 1/4, 1/16, 1/64, or 1/256 of the original image. From Table 1, it can be found that in computer vision, more attention is paid to the deeper part of the networks. For example, the arrangement of VGG is (2, 2, 3, 3, 3), while the numbers of convolutional layers in the different parts of ResNet are (3, 0, 6, 8, 12). In steganalysis, there are more convolutional layers on the lower part of the networks, such as (2, 2, 1, 1, 1) in XuNet, and the numbers of convolutional layers in YeNet are (3, 2, 2, 2, 2, 2).

We think that the reason is because, for the steganalysis task, the embedded steganographic signals are at the pixel level, so the convolutional layers in the lower part of the networks are more effective than those in the deeper part of networks. Therefore, in our proposed model, before the first pooling layer and between the first and second pooling layers, there are both four convolutional layers with a kernel size of 3×3, and the kernel size of the subsequent convolutional layer is 1×1.

Table 1 shows the differences in the arrangement of layers in VGG, ResNet, XuNet, YeNet and the proposed networks, where $C(k) \times n$ denotes n convolutional layers with a kernel size of $k \times k$ stacked together, and stride 2 means that the size of the feature maps is reduced by that stride in the convolutional layer instead of the pooling layer.

B. GLOBAL AVERAGE POOLING LAYER

A global average pooling layer can be used to effectively reduce the number of networks parameters and prevent overfitting [10]. In the existing convolutional neural networks, there were two common positions for global average pooling layer. In the networks of [11], the position of the global average pooling layer was at the last layer just before the softmax layer. The last convolutional layer generated one feature map for each category, and the global average pooling layer calculated the average value of each feature map to obtain the final output. Another position of the global average pooling layer can be the second-to-last layer, and it always follows a fully connected layer. As in many computer vision networks, the output of the global average pooling was not equal to the number of categories. Then, a fully connected layer was used to obtain the final output. The performance of the two convolutional neural networks with different positions for the global average pooling layer are compared in Section III. C.

C. ARCHITECTURE

Our proposed networks (Fig. 1 and Table 2) were composed of a preprocessing layer, series of convolutional and pooling layers, a global average pooling layer and softmax layer.

The preprocessing layer of our proposed networks was a convolutional layer initialized with 30 high-pass filters used in the SRM [20]. Different from those in [20], the parameters of the preprocessing layer were untrainable in our networks. The effect of this change can be seen in Section III. E. Additionally, the BN layer was added to increase the adaptability of the preprocessing layer. Last, there was a rectified linear unit (ReLU) [31] layer following the BN layer.

After the preprocessing layer, the output data were passed through a series of convolutional layers and pooling layers. All the convolutional layers in the proposed networks were equipped with a BN and ReLU activation function. Additionally, zero padding was used to keep the scale of the feature maps unchanged after the convolutional layer. In addition, the average pooling layers were used just as in [17]–[20]. The kernel size of each average pooling layer was 2×2, and the stride was 2. In addition to the preprocessing layer, the other convolutional layers and pooling layers were constructed into five blocks. The first block contained four convolutional layers with a kernel size of 3×3 and one average pooling layer, and the number of channels for this block was 30. The second block also contained four convolutional layers with a kernel size of 3×3 and one average pooling layer. The first convolutional layer in this block changes the number of channels from 30 to 60 to increase the diversity of the features. In the following convolutional layers within this block, the number of channels was kept as 60. The third and the fourth blocks contained one convolutional layer with a kernel size of 1×1 and one average pooling layer, and the number of channels was still 60. The fifth block was composed of two convolutional layers with a kernel size of 1×1. In the second convolutional layer within the fifth block, the number of channels was changed to 2 to obtain the required number of channels for classification. A global average pooling layer was set after the fifth block, which reduced the size of the feature maps from 16×16 to 1×1. After the global average pooling layer, a softmax layer was

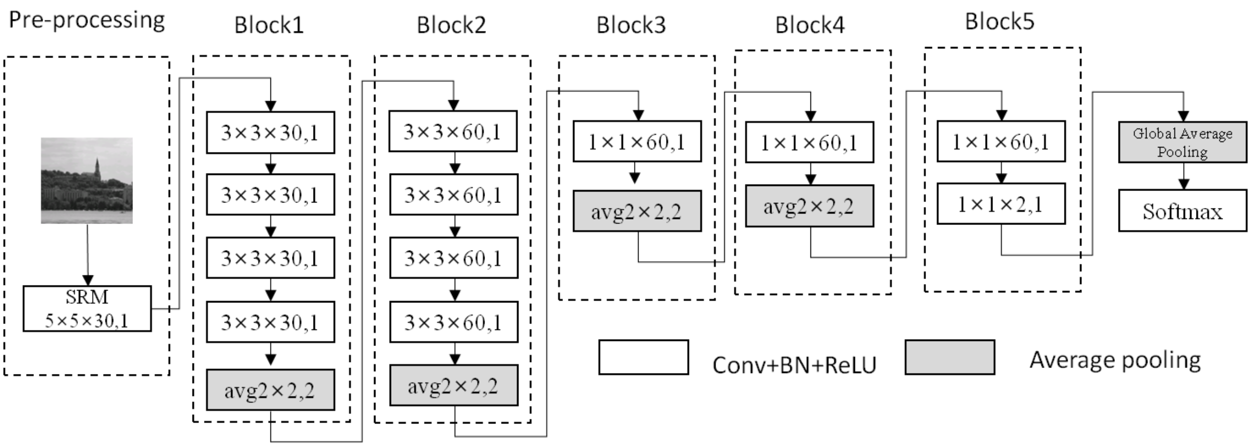


FIGURE 1. The architecture of the proposed convolutional neural networks.

TABLE 2. Architecture.

Layer	Input size	Output size	Process	Kernel size
1	256×256×1	256×256×30	Conv(SRM)-BN-ReLU	5×5(stride 1)
2	256×256×30	256×256×30	Conv-BN-ReLU	3×3(stride 1)
3	256×256×30	256×256×30	Conv-BN-ReLU	3×3(stride 1)
4	256×256×30	256×256×30	Conv-BN-ReLU	3×3(stride 1)
5	256×256×30	256×256×30	Conv-BN-ReLU	3×3(stride 1)
6	256×256×30	128×128×30	Average pooling	2×2(stride 2)
7	128×128×30	128×128×60	Conv-BN-ReLU	3×3(stride 1)
8	128×128×60	128×128×60	Conv-BN-ReLU	3×3(stride 1)
9	128×128×60	128×128×60	Conv-BN-ReLU	3×3(stride 1)
10	128×128×60	128×128×60	Conv-BN-ReLU	3×3(stride 1)
11	128×128×60	64×64×60	Average pooling	2×2(stride 2)
12	64×64×60	64×64×60	Conv-BN-ReLU	3×3(stride 1)
13	64×64×60	32×32×60	Average pooling	2×2(stride 2)
14	32×32×60	32×32×60	Conv-BN-ReLU	3×3(stride 1)
15	32×32×60	16×16×60	Average pooling	2×2(stride 2)
16	16×16×60	16×16×60	Conv-BN-ReLU	1×1(stride 1)
17	16×16×60	16×16×2	Conv-BN-ReLU	1×1(stride 1)
18	16×16×2	1×1×2	Global avg-pooling	16×16
	1×1×2	1×1×2	Softmax	--

applied to transform the feature vectors to output probabilities for each class.

III. EXPERIMENTS

A. DATASETS

The dataset used in our experiments was BOSSbase v1.01 [32], which contains 10000 8-bit gray images with a size of 512 × 512, and it has been widely used in steganalysis. Because 512 × 512 was a large size for the CNN method, some cropping methods were used to reduce the size of the images in the current research. For example, an image patch with a size of 256 × 256 in the middle of original image was used in YeNet (denoted as center-cropped256). Therefore, to facilitate the comparison with YeNet, the center-cropped256 images were used as the cover images, and the WOW and S-UNIWARD steganography algorithms were applied. In addition, when comparing our results with XuNet, the original 512 × 512 images were used as the cover images, and the S-UNIWARD steganography scheme was used.

B. IMPLEMENTATION DETAILS

Out of the 10000 pairs of images (cover image and its corresponding stego), 5000 pairs were set as training data, and 5000 pairs were set aside for testing to verify the performance. The minibatch gradient descent method was used for training. Each learning batch contained 20 training images, which were composed of 10 cover images and their corresponding steganographic images. The optimizer used was Adam [33], in which the learning rate was initialized to 5e-4, and the learning rate decayed 0.91 per 1000 batches. All parameters were initialized by Glorot and Bengio [34]. An L2 weight decay of 1e-5 was adopted.

C. PERFORMANCE COMPARISON OF MODELS WITH DIFFERENT GAP POSITIONS

In this experiment, the performance of two networks that set the global average pooling layers in different positions were compared. The lower parts of the two networks were the same as the networks described in Section II. The last four layers

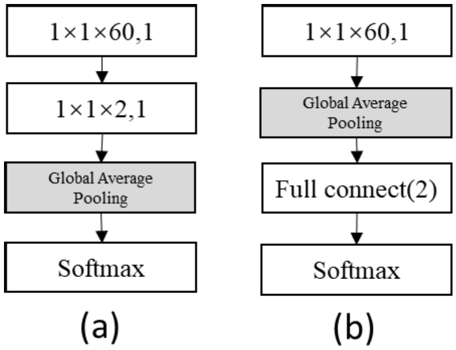


FIGURE 2. Different positions of the global average pooling layer. (a) shows the last four layers of the model when global average pooling is just before softmax, and (b) shows the last four layers when global average pooling is before the fully connected layer.

TABLE 3. Accuracy of CNN with the GAP in different positions.

	Model (a)	Model (b)
With_BN	86.17	--
Without_BN	85.60	84.45

of the two networks were different and are shown in Fig. 2. In the first model, the final four layers of the networks were $Conv(1, 60)$-$Conv(1, 2)$-GAP-$Soft$ max, and the last four layers of the second model were $Conv(1, 60)$-GAP-$FC(2)$-$Soft$ max. Here, $Conv(a, b)$ represents a convolutional layer with a kernel size of $a \times a$ and an output channel size of b, and $FC(c)$ represents a fully connected layer with an output channel size of c.

In TABLE 3, the accuracies of the two networks with the global average pooling layer at different positions are presented. The dataset used in this experiment were center-cropped256, and the steganography scheme was WOW with a payload of 0.4 bpp. The "With_BN" in TABLE 3 represents that the last convolutional layer or fully connected layer was equipped with a BN layer. The experimental results show that when BN is not used in model (a), there is still an improvement compared with that of model (b). By adding BN, the performance of model (a) can be further improved. Therefore, this experiment showed that setting the GAP before the softmax layer was helpful for steganalysis.

D. PERFORMANCE COMPARISON OF THE MODELS WITH DIFFERENT ARRANGEMENTS

In this experiment, we will verify our proposed arrangement of convolutional layers and pooling layers.

First, the number of arrangements of convolutional layers and pooling layers is very large. To make the comparison possible, the compared model has the following limitations. 1) Only several blocks (denote as N_{block}) in the lower part of the model have convolutional layers with a kernel size of 3×3. The kernel size of the other convolutional layers is 1×1. 2) In the first N_{block} blocks, the number of convolutional layers contained in each block is the same, and the number is denoted as N_{layer}. As shown in Fig. 3.

Table 4 shows the results of the models with different N_{block} and N_{layer} settings. The dataset used in this experiment was center-cropped256, and the steganography scheme used was WOW with a payload of 0.4 bpp. In Table 4, from the upper left corner to the lower right corner, as the depth increases, the performance continues to improve, and finally, the model cannot be trained because it is too deep. In Table 4, the maximum value was for $N_{block} = 2$, $N_{layer} = 4$ and the next was for $N_{block} = 3$, $N_{layer} = 3$. It can be seen that the number of convolutional layers with a kernel size of 3×3 that was most suitable for steganalysis was approximately $2 \times 4 = 8$ and $3 \times 3 = 9$. By comparing the models with depths of 8 or 9, including ($N_{block} = 2$, $N_{layer} = 4$), ($N_{block} = 3$, $N_{layer} = 3$), and ($N_{block} = 4$, $N_{layer} = 2$), we can find that the model with more convolutional layers in the lower layer ($N_{block} = 2$, $N_{layer} = 4$) was better than the model where convolutional layers were equally placed between the pooling layers ($N_{block} = 4$, $N_{layer} = 2$). This experiment shows that our proposed arrangement, in which four convolutional layers are set in each of the initial two blocks, is more adaptive to steganalysis than other arrangements when convolutional layers with a kernel size of 3×3 are only set in some lower layers of the CNN model and the number of convolutional layers with a kernel size of 3×3 in each block is the same.

E. PERFORMANCE COMPARISON WITH OTHER STEGANALYZERS

In this subsection, we compare the performance of the proposed CNN-based models with some other steganalyzers in the spatial domain, such as the SRM, YeNet and XuNet.

Table 5 shows the performances of our proposed networks, YeNet and SRM versus two other steganography schemes (WOW and S-UNIWARD) on the center-cropped256 images. From the experimental results in Table 5, it can be seen that the proposed model was better than YeNet when the steganalysis embedding rate was high (0.4, 0.3, and 0.2 bpp) but not as good as YeNet when the steganalysis embedding rate was 0.1 bpp.

In Table 6, the results of the SRM, XuNet and proposed networks trained on the 512 × 512 dataset versus S-UNIWARD are presented. From the experimental results in Table 6, it can be seen that the proposed CNN was better than XuNet versus S-UNIWARD when the embedding rate was 0.4 bpp and 0.1 bpp.

Because GAP was used in our proposed networks, the model can accept images with different scales as input. In Table 7, the accuracy of our CNN trained on center-cropped256 with a payload of 0.4 bpp using WOW and tested on the 512 × 512 dataset with a payload of 0.4 bpp using WOW are presented. It can be seen that the model trained on Cent-cropped256 obtained a better result of 89.84% when transferred to the 512 × 512 dataset. This result shows that the model trained on cent-cropped256 has good generalization ability on the 512 × 512 dataset.

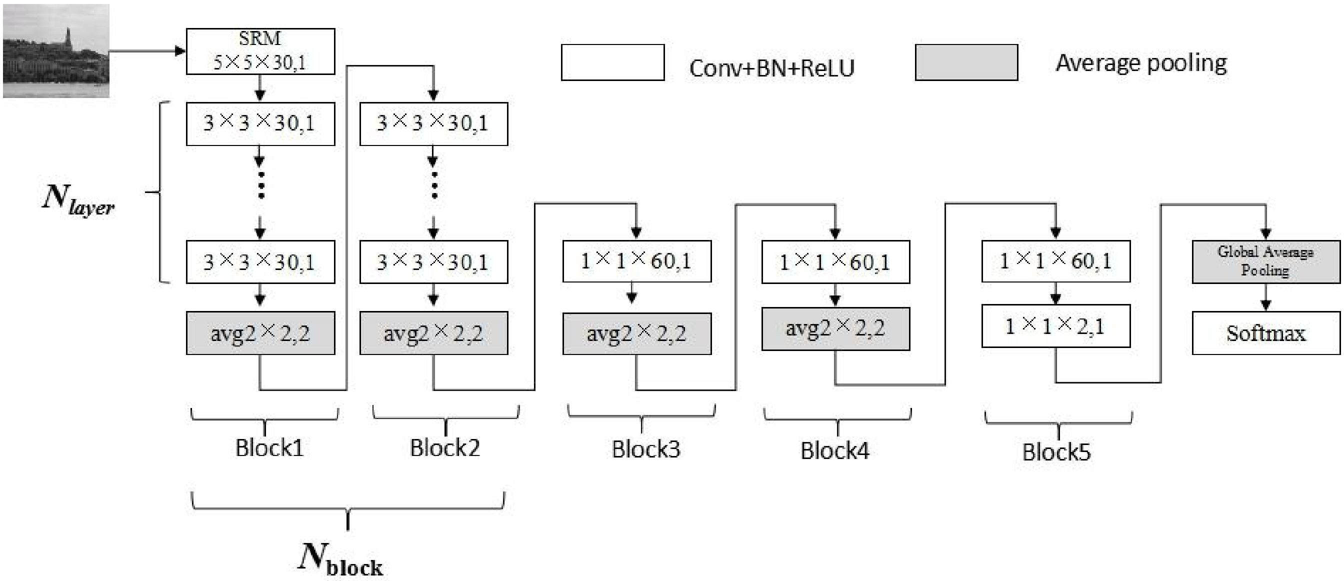


FIGURE 3. The architecture for verifying and the two parameters (N_{block} and N_{layer}) of the arrangements. N_{block} represents the number of blocks with 3 × 3 convolutional layers in the lower part of the networks and N_{layer} represents the number of 3 × 3 convolutional layers in the first N_{block} block.

TABLE 4. Accuracy of the proposed CNN with different $N_{block}\backslash N_{layer}$ on center-cropped256 versus WOW with a payload of 0.4 bpp.

$N_{block}\backslash N_{layer}$	2	3	4	5	6
1_block _3×3	81.36	82.44	84.21	84.25	84.77
2_block _3×3	84.01	85.34	86.17	86.03	85.67
3_block _3×3	85.37	86.14	85.78	-	-
4_block _3×3	85.1	85.6	-	-	-

TABLE 5. Accuracy of SRM\YeNet\proposed networks on center-cropped256 versus WOW and S-UNIWARD.

Algorithm	Payload (bpp)	SRM	YeNet [20]	Proposed networks
WOW	0.1	55.40	67.60	63.91
	0.2	61.47	75.65	77.10
	0.3	66.63	79.64	82.55
	0.4	71.13	82.93	86.17
S-UNIWARD	0.1	55.61	60.62	57.70
	0.2	61.76	67.82	70.29
	0.3	67.13	74.29	77.07
	0.4	72.95	80.45	81.79

TABLE 6. Accuracy of SRM\XuNet\ours on the 512 × 512 dataset versus S-UNIWARD.

Algorithm	Payload (bpp)	SRM	XuNet [18]	Ours
S-UNIWARD	0.1	59.25	57.33	64.6
	0.4	79.53	80.24	88.6

F. EXPERIMENTSFOR SOME STRUCTURAL DETAILS

The experiments in this section were conducted using the center-cropped256 dataset. The steganography scheme used was WOW, and the payload was 0.4 bpp.

We compared the model that uses average pooling in all pooling layers (denoted as "all avg" in Table 8) with the model that uses maximum pooling in all pooling layers (denoted as "all max" in Table 8), and the results are shown in Table 8. It can be seen that using maximum pooling may result in a slight performance reduction. According to [17], maximum pooling was more suitable for sparse features, but steganalysis features were not sparse.

In the proposed model, the number of channels increases to 60 in the second block (denoted as c-30-60). To verify

TABLE 7. Transfer accuracy of our CNN trained on center-cropped256 with a payload of 0.4 bpp using WOW and tested on the 512 × 512 dataset with a payload of 0.4 bpp using WOW.

train\test	Cent-cropped-256	Img-512
Cent-cropped-256	86.17	89.84

TABLE 8. Different pooling methods.

Pooling methods	Accuracy
All avg	86.17
All max	84.97

TABLE 9. Different channel settings.

Channel setting	Accuracy
c-30-60	86.17
c-30	85.69

TABLE 10. The parameters of the preprocessing layer were trainable or untrainable.

	Accuracy
Trainable	86.17
Untrainable	85.74

TABLE 11. Different activation functions in the preprocessing layer.

Activation function	Accuracy
ReLU	86.17
tanh	85.87

this structure, a comparison with the model whose number of channels for all convolutional layers were maintained at 30 (denote as c-30) was given. It is found that the performance can be weakly improved by increasing the number of channels in the second block.

In the proposed model, the parameters of the preprocessing layer were set as untrainable. Table 10 compares this model with the model in which the parameters of the preprocessing layer were set as trainable. The performance of the trainable model was slightly below that of the untrainable model.

In Table 11, the models with different activation functions (ReLU and tanh) in the preprocessing layer were analyzed. As seen from Table 11, the ReLU can slightly improve the network performance. We suspect that more convolutional layers with the ReLU in the proposed model have provided a sufficient nonlinear capacity, which may reduce the demands for the nonlinearity activation function such as tanh.

IV. CONCLUSION

In this paper, a new convolutional neural network-based steganalysis approach for adaptive spatial image steganography is proposed. The proposed CNN exhibits two characteristics: 1) There is a new arrangement of layers in the networks to enhance the local processing ability of the networks. Four convolutional layers with a kernel size of 3 × 3 were stacked before the first pooling layer and between first and second pooling layers respectively, and the other convolutional layers in the networks had a kernel size of 1 × 1. 2) Global average pooling is used to reduce the number of parameters in the networks and prevent overfitting, and by setting the global average pooling layer before the softmax layer, we obtain a better position for the global average pooling for steganalysis. Finally, the experimental results show that the proposed CNN obtains better performance than that of some of the latest CNN-based steganalysis methods on the BOSSbase dataset.

REFERENCES

[1] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011, doi: 10.1109/TIFS.2011.2134094.

[2] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. Int. Workshop Inf. Hiding*, in Lecture Notes in Computer Science, vol. 6387. Berlin, Germany: Springer, 2010, pp. 161–177.

[3] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2012, pp. 234–239, doi: 10.1109/WIFS.2012.6412655.

[4] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. Inf. Secur.*, vol. 2014, p. 1, Jan. 2014.

[5] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 4206–4210, doi: 10.1109/ICIP.2014.7025854.

[6] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012, doi: 10.1109/TIFS.2012.2190402.

[7] V. Holub and J. Fridrich, "Random projections of residuals for digital image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 1996–2006, Dec. 2013, doi: 10.1109/TIFS.2013.2286682.

[8] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444. 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[10] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: http://arxiv.org/abs/1312.4400

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[13] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale JPEG image steganalysis using hybrid deep-learning framework," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1200–1214, May 2018.

[14] G. Xu, "Deep convolutional neural network to detect J-UNIWARD," in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur. (IHMMSec)*, Philadelphia, PA, USA, 2017, pp. 67–73.

[15] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, "JPEG-Phase-Aware convolutional neural network for steganalysis of JPEG images," in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur. (IHMMSec)*, Philadelphia, PA, USA, 2017, pp. 75–84.

[16] S. Tan and B. Li, "Stacked convolutional auto-encoders for steganalysis of digital images," in *Proc. Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Dec. 2014, pp. 1–4, doi: 10.1109/APSIPA.2014.7041565.

[17] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," *Proc. SPIE*, vol. 9409, Mar. 2015, Art. no. 94090J.

[18] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, May 2016, doi: 10.1109/LSP.2016.2548421.

[19] G. Xu, H. Wu, and Y. Shi, "Ensemble of CNNs for Steganalysis: An empirical study," in *Proc. IH&MMSec*, Galicia, Spain, vol. 2016, pp. 103–107, 2016.

[20] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017, doi: 10.1109/TIFS.2017.2710946.

[21] C. F. Tsang and J. Fridrich, "Steganalyzing images of arbitrary size with CNNs," *Electron. Imag.*, vol. 2018, no. 7, pp. 121-1–121-8, Jan. 2018, doi: 10.2352/ISSN.2470-1173.2018.07.MWSF-121.

[22] J. Couchot, R. Couturier, and M. Salomon, "Improving blind steganalysis in spatial domain using a criterion to choose the appropriate steganalyzer between CNN and SRM+EC," in *Proc. IFIP Int. Conf. ICT Syst. Secur. Privacy Protection Inf. Secur.*, Italy, Rome, 2017, pp. 327–340.

[23] J. Zeng, S. Tan, B. Li, and J. Huang, "Pre-training via fitting deep neural network to rich-model features extraction procedure and its effect on deep learning for steganalysis," *Electron. Imag.*, vol. 2017, no. 7, pp. 44–49, Jan. 2017.

[24] S. Wu, S.-H. Zhong, and Y. Liu, "A novel convolutional neural network for image steganalysis with shared normalization," *IEEE Trans. Multimedia*, vol. 22, no. 1, pp. 256–270, Jan. 2020, doi: 10.1109/TMM.2019.2920605.

[25] J. Yang, K. Liu, X. Kang, E. Wong, and Y. Shi, "Steganalysis based on awareness of selection-channel and deep learning," in *Proc. Int. Workshop Digit. Watermarking*. Magdeburg, Germany: Springer, 2017, pp. 263–272,

[26] V. Sedighi and J. Fridrich, "Histogram layer, moving convolutional neural networks towards feature-based steganalysis," *Electron. Imag.*, vol. 2017, no. 7, pp. 50–55, Jan. 2017.

[27] M. Yedroudj, F. Comby, and M. Chaumont, "Yedroudj-net: An efficient CNN for spatial steganalysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2092–2096.

[28] J. Zeng, S. Tan, G. Liu, B. Li, and J. Huang, "WISERNet: Wider Separate-Then-Reunion network for steganalysis of color images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2735–2748, Oct. 2019.

[29] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010, doi: 10.1109/TIFS.2010.2045842.

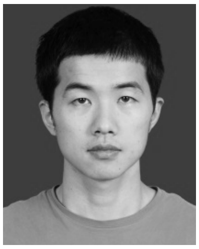
[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Lille, France, 2015, pp. 448–456.

[31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 807–814.

[32] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing boss," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2011, pp. 59–70.

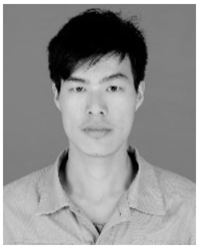
[33] P. Diederik Kingma and B. Jimmy, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLP)*, San Diego, CA, USA, 2015, pp. 1–41.

[34] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Sardinia, Italy, 2010, pp. 249–256.



ZHILI XIANG received the B.S. degree in engineering mechanics from the Inner Mongolia University of Science and Technology, China, in 2009. He is currently pursuing the Ph.D. degree in software engineering with Chongqing University, China. His research interests include information security, machine learning, and deep learning.

JUN SANG received the B.Sc. degree from Shanghai Jiaotong University, China, in 1990, and the M.E. and Ph.D. degrees in computer science from Chongqing University, China, in 1993 and 2005, respectively. He is currently a Professor with the School of Software Engineering, Chongqing University. His research interests include image processing, software engineering, digital image watermarking, and information security.



QIAN ZHANG received the B.S. degree in software engineering from Yangtze University, China, in 2016, and the master's degree in software engineering from Chongqing University, China, in 2019. His research interests include information security, machine learning, and deep learning.

BIN CAI received the B.Sc. degree from Southwest Normal University, China, in 2002, and the M.Sc. and Ph.D. degrees in mechanical engineering from Chongqing University, China, in 2005 and 2012, respectively. He is currently an Associate Professor with the School of Big Data and Software Engineering, Chongqing University. His research interests include software engineering, optimization method, and cryptography.

XIAOFENG XIA received the master's degree from Chongqing University, China. He is currently an Associate Professor with the School of Big Data and Software Engineering, Chongqing University. He is also a Research Associate with the Key Laboratory of Dependable Service Computing in Cyber Physical Society of MoE, China. His current research interests include cloud security, industrial control security, and cryptographic applications.

WEIQUN WU received the B.S. degree in software engineering from Chongqing University, China, in 2017, where he is currently pursuing the master's degree. His research interests include image processing, deep learning, software engineering, and information security.