mathematics

MDPI

Article

Preprocessing Enhancement Method for Spatial Domain Steganalysis

Xueming Duan [1], Chunying Zhang [1,2,*], Yingshuo Ma [1] and Shouyue Liu [1]

[1] College of Science, North China University of Science and Technology, Tangshan 063210, China
[2] Key Laboratory of Data Science and Application of Hebei Province, Tangshan 063210, China
* Correspondence: hblg_zcy@126.com

Abstract: In the field of steganalysis, in recent years, the research focus has mostly been on optimizing the structures of neural networks, while the application of high-pass filters is still limited to the simple selection of filters and simple adjustment of the number of filters. In this paper, we propose a method to enhance the assistance and contribution of high-pass filters to the detection capability of a spatial domain steganalysis model, which mainly contains the preprocessing enhancement of high-pass filters and cross-layer enhancement of high-pass filters, and we construct a preprocessing enhancement model, the HPF-Enhanced Model, for spatial domain steganalysis, based on Yedroudj-Net. In the experimental part, we find the best preprocessing enhancement method through various validations, and we compare the HPF-Enhanced Model with the classical models. The results show that the proposed enhancement method can bring a significant improvement, and they also show that the preprocessing enhancement method can help to reduce the model size, and it thus can be used to construct a lightweight spatial domain steganalysis model with strong performance.

Keywords: high-pass filters; spatial domain steganalysis; preprocessing enhancement; cross-layer enhancement

MSC: 68T07

check for updates

Citation: Duan, X.; Zhang, C.; Ma, Y.; Liu, S. Preprocessing Enhancement Method for Spatial Domain Steganalysis. *Mathematics* **2022**, *10*, 3936. https://doi.org/10.3390/math10213936

Academic Editor: Christopher Goodrich

Received: 28 September 2022
Accepted: 21 October 2022
Published: 23 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As two important components of network security, steganography and steganalysis are the main research objects in the current network security field. In traditional steganography methods such as LSB [1], steganography algorithms generally pay more attention to the embedding method and ignore the consideration of the embedding location; too much embedding in the flat area of the image makes the image's steganographic features more sensitive and easily detected by steganalysis methods, and the steganographic security is not high. Due to the gradually increasing attention to steganographic security, adaptive steganographic algorithms have been developed in the field of steganography. They usually choose a complex region of the image texture with higher security for embedding, and achieve the purpose of ensuring steganographic security by minimizing the sum of the distortion cost of embedding. Typical spatial domain adaptive steganographic algorithms are HUGO [2], WOW [3], S-UNIWARD [4], and HILL [5], etc. Correspondingly, steganalysis has gradually led to the development of adaptive steganalysis methods and become a hot topic in current research.

Based on the convolutional neural network (CNN), the adaptive steganalysis methods can enhance the feature extraction ability of the embedding location of the adaptive steganography algorithm, so as to better cope with the adaptive embedding method of the steganography algorithm. One method that has greatly improved the detection performance of steganalysis in the spatial domain is the use of high-pass filters (HPF). The image is usually first processed with high-pass filters before it enters the convolutional network. After filtering, the image loses a large amount of irrelevant content information to

become image residuals, allowing the subsequent convolutional layers to extract adaptive steganographic features more easily. The initial weights of the filters are mostly taken from the Spatial Rich Model (SRM) [6] and its derivatives. Qian et al. [7] used a 5×5 sized filter for preprocessing and normalized the weights in the proposed steganalysis model, GNCNN. Using a similar preprocessing approach to GNCNN, Xu et al. [8] designed the Xu-Net model by adding absolute value (ABS) and batch normalization (BN) [9] layers and using the Tanh activation function [10] in the first two convolutional layers to retain more information about the filtered residuals. Ye et al. [11] proposed a model called Ye-Net, which used 30 high-pass filters taken from SRM to extract diverse steganographic noise, and they introduced the Truncation Linear Unit (TLU) to be used before the convolutional layers, after the filtering layer, achieving excellent detection results for steganographic analysis. Following this, Yedroudj et al. [12] proposed the Yedroudj-Net model, also using a filter bank in the preprocessing layer, and with BN layers and TLU activation functions.

The high-pass filter essentially removes the influence of the image content itself on the steganalysis network, because steganalysis is fundamentally different from traditional convolutional classification in that steganalysis does not focus on what the image content represents, while the filtering operation removes most of the useless image content and retains the residual information in complex regions of the texture, preventing the steganalysis network from extracting too many unnecessary features and thus becoming more sensitive to steganographic noise. This also illustrates the importance of high-pass filters for spatial domain steganalysis. However, the application of high-pass filters is still limited to the simple selection of filters and simple adjustment of the number of filters, and the research is often focused on the optimization of convolutional networks [13–15]. We believe that the improvement of high-pass filters regarding the capability of spatial domain steganalysis is crucial, and this improvement would exceed the gain from the change in network structure. As the current use of high-pass filters does not maximize its effect, research on the improvement of the means of using high-pass filters has great significance.

The main research objective of this paper is to enhance the assistance and contribution of high-pass filters to the detection capability of spatial domain steganalysis models, and the main work includes the following:

(1) We study the enhancement method of high-pass filters to extract residual information in the preprocessing layer, so that the subsequent convolution layers can extract more steganographic features;

(2) We investigate methods to more fully utilize the enhanced residual information extracted by high-pass filters for feature reuse and overfitting mitigation;

(3) We apply the first two research elements to the improved Yedroudj-Net model and compare it with the advanced classical steganalysis model to test the effectiveness of the preprocessing enhancement method for spatial domain steganalysis.

2. Enhancement of Filter Extraction

The scope of this section is restricted to the preprocessing layer, and the residual information extracted by the enhanced high-pass filters in the preprocessing layer is studied. The specific method is as follows: first, group the filters according to the characteristics of their weights, select the most suitable bank of filters as the experimental object according to the contribution of each bank to the detection ability of the steganalysis model, continue the experiments with different enhancement methods, and finally select the filter enhancement method with the best enhancement effect and apply it to each bank of filters.

(1) Grouping the filters

The high-pass filters used in the preprocessing layer of spatial steganalysis are basically taken from the SRM, and we consider that the extraction of residual information using these filters is sufficient and comprehensive, so the enhancement method involves only SRM-related filters. There are 30 general-purpose high-pass filters, many of which are obtained by rotating the weights by 45° or 90°. If we disregard the new filters obtained by

rotation and observe only the basis weights, we can divide these filters into seven banks, as in Table 1.

Table 1. Details of filter banks.

Filter Bank	Base Weights	Rotation Angle	Number of Filters
Bank 1	$\begin{bmatrix} -1 & 1 \end{bmatrix}$	45°	8
Bank 2	$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$	45°	4
Bank 3	$\begin{bmatrix} 1 & -3 & 3 & -1 \end{bmatrix}$	45°	8
Bank 4	$\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$	-	1
Bank 5	$\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \end{bmatrix}$	90°	4
Bank 6	$\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}$	-	1
Bank 7	$\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \end{bmatrix}$	90°	4

The basis weight of the first bank in Table 1 is a first-order $[-1, 1]$, which generates 8 filters after rotating one turn at 45°, while the basis weight of the fourth bank is a symmetric matrix, which does not need to be rotated and contains only one filter. The meanings of the basis weights, rotation angles, and the number of filters obtained for the other banks in the table are the same as above.

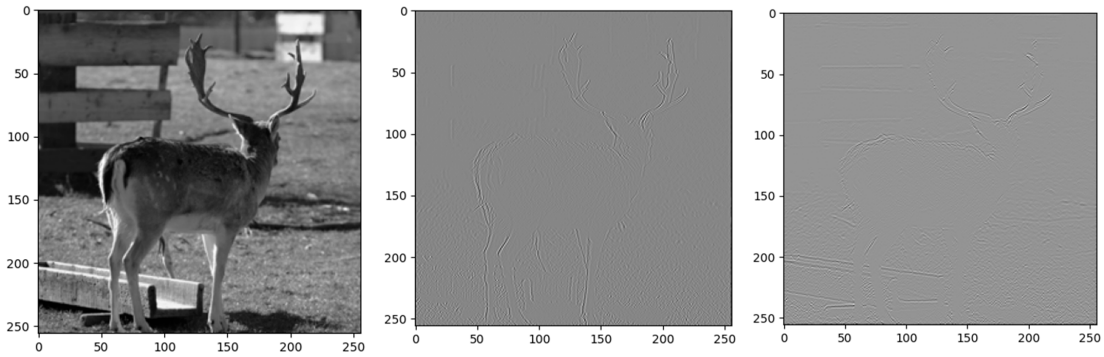
To place filters in the convolutional neural network as a preprocessing layer, it is common practice to form square filter kernels of 5×5 size, with the weights as the center and zeroes around them. For filters with a very short length of the basis weights (e.g., the first bank), the number of zeros is destined to be much larger than the actual number of weights after the zeroes are made up into a 5×5 filter kernel, which is detrimental to the feature extraction task. Once the high-pass filters in the preprocessing layer are set to participate in network learning, the learning effect is greatly reduced in the presence of a large number of zeros. In order to eliminate these disadvantages, different complementary zero operations are performed according to the size of the basis weight length. The basis weights with lengths less than or equal to 3 are zeroed to the filter of size 3×3, and the basis weights with lengths greater than 3 are zeroed to the filter of size 5×5. Then, we set the padding so that the output sizes of the 3×3 size and 5×5 size filters are the same.

The filters rotated out of the same bank of weights can extract residual information from multiple directions. The steganographic image of the S-UNIWARD steganographic algorithm at a 0.4 embedding rate (bpp) is used as an example to demonstrate the extraction effect of several filters in the seventh bank, as shown in Figure 1.

We use Yedroudj-Net as the benchmark model and investigate the contributions of different banks of filters to steganalysis separately. The filters in the preprocessing layer of Yedroudj-Net, i.e., the high-pass filter layer, are replaced with filters of each bank, and the filters with smaller weight sizes are supplemented with 3×3 size, and the parameters are set according to the original settings. Section 5.4 shows the accuracy results of each bank of filters on the validation set.

It can be said that the first three banks of filters have basically no detection capability when used as preprocessing layers alone. This is due to the fact that their underlying

weights are relatively thin compared to the other filters, and the residual information contains less feature information, but this does not mean that such weights are useless, and they work together with other filters to make the residual information more comprehensively expressed.



(**a**) (**b**) (**c**)

Figure 1. Filtering effect of Bank 7's filters. (**a**) Grayscale image of 256 × 256 size. (**b**) Filtering effect of Bank 7's base weight on the grayscale image after complementing zero. (**c**) Filtering effect of Bank 7's base weight on the grayscale image after complementing zero and rotating 90° clockwise.

(2) Enhanced representation of filters

Due to the large number of banks of filters, we would like to isolate one group for further exploration. Firstly, the bank with an accuracy close to 50% need not be considered, because it is likely that, even if enhanced, the performance is not significant; secondly, the bank with only one filter is not considered, as there is almost no room for enhancement; finally, to make the difference in comparison between different enhanced representations more obvious, the bank with the highest known accuracy is not considered, so the final benchmark filter bank used in the enhancement experiment is Bank 7.

The question of how to perform the augmented representation is a problem. The residuals extracted by filters in the same bank are of the same class and differ only in direction, while the features of the residuals in different banks differ greatly, so the probe direction is framed within the bank rather than enhanced across banks. The preprocessing layers consisting of high-pass filters in SRM all have the characteristic wherein there is a large number of multiple filters generated when rotating the same weights. For each of the multiple filters, it is only responsible for the extraction of residuals in one direction under this filtering method, and the residual information extracted from these filters in different directions is only input as the feature maps of different channels when they are input to the first convolution layer. Moreover, no actual connection is established between the filters in different directions, leading to the fact that although the extracted residuals have multiple directions and seem to be comprehensive, the residual representations in each direction are actually very scattered.

To strengthen the association in each direction and make the residual features within a filter bank more expressive, the feature outputs of the filters in the bank are fused. For example, the first bank of filters produces eight residual feature maps, and the fusion of features on them produces an enhanced residual feature map. We consider that the methods of fusion are summing the feature map weights, taking the mean value, taking the absolute maximum value, and taking the extreme value. Taking the first bank of 8 filters as an example, the feature map weights are summed, i.e., an image passing through the first bank of filters will produce 8 weight matrices, and the 8 weights at the same position of the matrix are summed to form a weight matrix as the output by fusing the residual information in each direction. The elements in the matrix generated by each filter have their own size, positive and negative. When added by bit, weights with the same positive number or the same negative number will gain, making the features more obvious; for

the case of one positive and one negative, the weights will cancel each other out, masking irrelevant or unclear features to prevent disruption to the network's extraction of obvious steganographic features. The fusion method of taking the mean value and the absolute maximum value is the same, which is to form a weight matrix by taking the mean value or the maximum value of the absolute value of the weight matrix in each direction. The mean value represents a common and relatively average feature expression in each direction, without causing too much weight gain and leading to weight size differentiation. The maximum value of the absolute value represents the most significant features of each position in the matrix after filtering in different directions, aiming to retain the obvious steganographic features in the residual information. The fusion method of taking the extreme value means to take the one with the greater absolute value of the minimal or extreme value of the weights at the same position in multiple weight matrices—for example, choosing -2 between -2 and 1. Compared with the fusion method of taking the absolute maximum value, this method is more appropriate, as it preserves the most prominent weights at each position in the matrix while retaining the positive and negative information of the weights, avoiding the loss of features. However, because the operation of comparing absolute values and then selecting extreme values is very time-consuming in the network learning process, we do not consider it a very cost-effective and implementable fusion scheme, so we did not set up experiments using this method.

Figure 2 shows the effect of these fusion methods applied to the output feature map of the seventh bank of filters. It can be seen that, compared with the effect of processing the seventh bank of partial filters alone, as shown in Figure 1, the fusion enhancement method produces a feature map containing richer and more comprehensive information about the residuals, and the image texture part is more prominent and obvious. Moreover, it can be seen that the fusion enhancement method has a higher probability of covering the embedded region than the embedding changes of the image by the steganography algorithm in Figure 2.

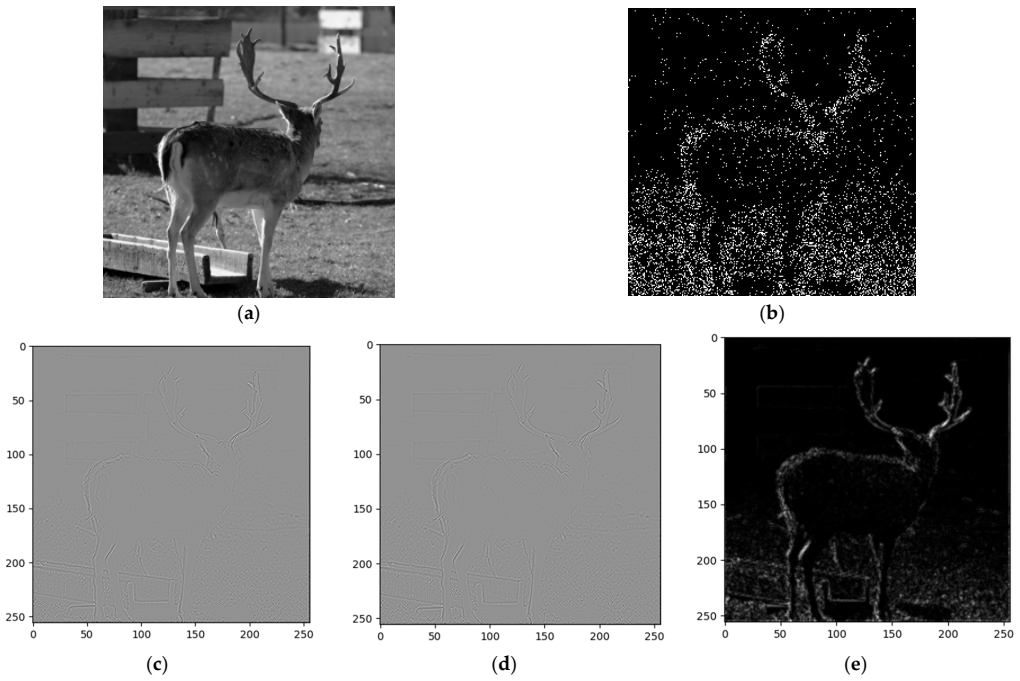


Figure 2. Output feature maps applying different fusion methods. (**a**) Grayscale image of 256 × 256 size. (**b**) Altered positions (white dots) of the grayscale image for the S-UNIWARD steganography algorithm with an embedding rate of 0.4 bpp. (**c**) Output after using the feature fusion method of weight summation for grayscale image. (**d**) Output after using the feature fusion method of taking the weight mean for grayscale image. (**e**) Output after using the feature fusion method of taking the absolute maximum value of weights for grayscale image.

The experimental part of Section 5 concludes that the enhancement of the steganographic feature extraction capability using the fusion method of summing and averaging the feature map weights is more obvious. Then, the operation of summing and averaging the weights can be performed after all filter banks are processed, and the output feature maps of filter banks containing multiple filters are fused from multiple to two, i.e., a feature map generated by summing the weights and a feature map generated by averaging the weights, and the fusion enhancement method is not applied to filter banks containing only a single filter.

3. Cross-Layer Enhancement of Filters

The enhanced filters' extraction capability alone is not sufficient for the steganalysis improvement of the model. We hope that the fused and enhanced residual information can be transferred to multiple convolutional layers behind the network, as in DenseNet [16], for the purpose of feature reuse and overfitting mitigation. The residual information is transmitted across layers to the later convolutional layer and is destined to intersect with the output feature maps of the previous layer of this layer. Two issues should be considered in terms of of how to intersect: one is that the size of the input feature maps of the later convolutional layer may not be equal to the size of the original residual feature maps, and it is necessary to consider how to reduce the residual feature maps; and the other is how to combine the residual feature maps with the original input feature maps of the later convolutional layer.

First, we consider the first problem. The residual feature maps are the result of preprocessing before the image enters the convolutional layer, containing a large amount of image texture information, where the steganographic features are hidden, and the cross-layer connection between them and the later convolutional layer must first preserve the information of the original residual feature maps as much as possible. Reducing the size of the residual feature maps using convolutional layers with large step sizes will change the original feature representations, while the use of average pooling layers will achieve the goal without the network learning process. To preserve the original feature representation as much as possible, the pooling window of the average pooling layer performing the reduction process is a small size of 3×3. For multiple reductions in size, the average pooling layers of 3×3 window size are superimposed and used.

Addressing the second problem, referring to the residual network [17] and Inception structure [18], the combination of the residual feature maps and the original input feature maps of the later convolutional layers involves two aspects, splicing and weight summing, as shown in Figure 3. The combination of element-level summation, shown on the left side of Figure 3, does not change the number of feature maps between the upper output and the current input, meaning that the number of channels remains the same, but it changes the feature representation of the original output feature maps. The right side of Figure 3 shows the concatenation with the original output feature maps in the channel dimension; the number of channels is increased, and the next layer can receive feature maps from different layers to achieve feature reuse. Another advantage of increasing the number of channels is that we can replace part of the number of convolutional kernels in the convolutional layer, reducing the number of model parameters and increasing the training and detection speed. For example, if the number of input channels of a convolutional layer is 30, which means that the number of convolutional kernels of the previous layer is 30, and the number of feature maps concatenated together after cross-layer enhancement of the filters is 10, then the number of convolutional kernels of the previous layer can be reduced to 20, and with the 10 channels concatenated together, the number of input channels of the layer can still reach 30. Therefore, channel concatenation is chosen as the feature map combination method in filter cross-layer enhancement.

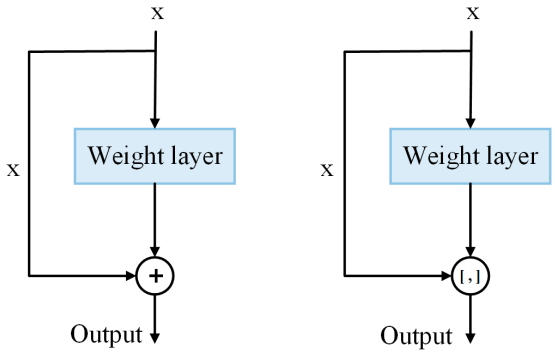


Figure 3. Two methods of combining feature maps.

4. Preprocessing Enhanced Model

This section refers to Yedroudj-Net and makes some changes to apply the filter extraction enhancement method and the filter cross-layer enhancement method to construct a preprocessing enhanced model for spatial domain steganalysis, named the HPF-Enhanced Model. Its structure is shown in Figure 4.

$$TLU(x) = \begin{cases} T, x \geq T \\ x, -T < x < T \\ -T, x \leq -T \end{cases} \tag{1}$$

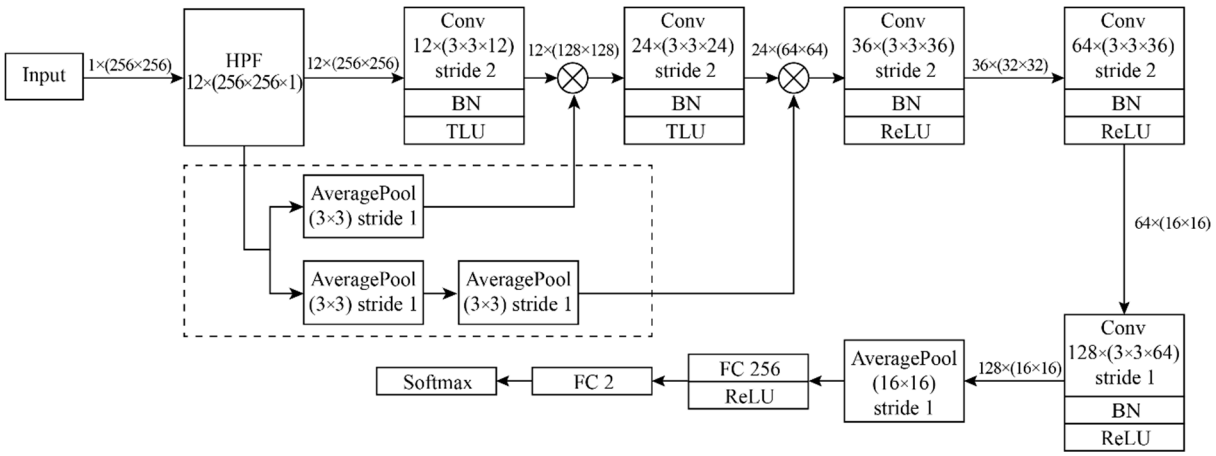


Figure 4. Structure of HPF-Enhanced Model.

The preprocessing layer of the HPF-Enhanced Model contains 30 high-pass filters of 3×3 or 5×5 size, taken from SRM. After filtering, the filter extraction enhancement method selected in Section 2 is applied, and the output channel is 12. There are five convolutional layers. The first two layers use the TLU activation function, whose expression is shown in Equation (1), to be able to retain the feature information of negative values. The last three layers use the ReLU [19] activation function to accelerate the model fitting speed. The pooling layer is not used in the convolution process; the convolution layer with the step size of 2 is used instead to avoid the loss of steganographic features. The convolutional layers are followed by a global average pooling layer, and then followed by two fully connected layers with 256 and 2 neurons, respectively. In addition, the filter cross-layer enhancement method in Section 3 is used in the second and third convolutional layers, with one average pooling and two average pooling steps performed before channel combination, respectively.

5. Experiment

5.1. Dataset and Software Platform

We use the well-known grayscale image dataset BOSSBase v.1.01 [20] for our experiments and produce steganographic datasets using the content-adaptive steganographic algorithm S-UNIWARD and its Matlab implementation, where the Matlab code can avoid the incorrect use of fixed and unique embedding keys in C++ code. In the model comparison, in addition to our newly constructed HPF-Enhanced Model, advanced and classical models Xu-Net and Yedroudj-Net are involved, and all models are trained and tested on the same subsampled images of the same dataset. All experiments are conducted using the Pytorch deep learning framework on a Linux system environment and are run on NVIDIA GeForce RTX 2080 SUPER GPU cards.

5.2. Training, Validation, and Testing

Due to GPU memory limitations, we use the Matlab function "imresize()" with default parameters to resample the 512×512 pixel image of BOSSBase v.1.01 into a 256×256 pixel image. Then, the 10,000 cover/stego pairs are divided into a training set, validation set, and test set according to the ratio of 4:1:5, and the division is randomly assigned. During the training of the HPF-Enhanced Model, we set a maximum training period of 900 epochs, and the actual training is manually stopped when the network shows signs of overfitting using the early stop method. Finally, the model is saved for subsequent validation when the detection accuracy reaches the maximum.

5.3. Hyper-Parameters

The batch size for training is set to 16, i.e., 8 cover/stego pairs. The training strategy uses Stochastic Gradient Descent (SGD), with momentum set to 0.95 and weight decay to 0.0001. The first two convolutional layers and the fully connected layer are initialized using the Xavier method, while the last three convolutional layers are initialized using the Kaiming method, and the weights follow a Gaussian distribution. The BN layer does not participate in weight decay and bias learning, the fully connected layer does not participate in bias learning, and the weights in the preprocessing layer are frozen from learning. During training, we use Pytorch's dynamic adjustment strategy to reduce the learning rate, with the initial learning rate set to 0.01 and halving the learning rate when the training loss is no longer decreasing beyond 20 epochs. The truncation threshold T in the TLU activation function is set to 3, and all high-pass filters in the preprocessing layer are not normalized.

5.4. Comparison Results of Filter Banks

The 30 high-pass filters are divided into 7 banks, as described in Section 2. Using Yedroudj-Net as the benchmark model, the preprocessing layer is replaced with each filter bank separately, and the contributions of different filter banks to the steganalysis are analyzed in terms of the detection accuracy on the validation set. The parameters and initialization follow the original Yedroudj-Net settings, except for the preprocessing layer change, and the S-UNIWARD steganography algorithm with an embedding rate of 0.4 bpp is used. The accuracy of each bank on the validation set is shown in Table 2.

Table 2. Accuracy of filter banks on validation set.

Filter Bank	Filter Kernel Size	Number of Filters	Accuracy (%)
Bank 1	3×3	8	50.00
Bank 2	3×3	4	50.35
Bank 3	5×5	8	57.60
Bank 4	3×3	1	78.15
Bank 5	3×3	4	79.60
Bank 6	5×5	1	76.20
Bank 7	5×5	4	77.45

The accuracy of the first two banks can be regarded as 50%, and the detection results are no different from random guesses. This is because the two filter kernels themselves are not strong in extracting image edges, which makes it difficult for the later convolutional layers to extract steganographic features from the residual information. Banks 4–6's filters achieve similar accuracies and are more helpful for spatial domain steganalysis, with Bank 5 achieving the best results, containing four filters of size 3×3. From another perspective, it can be seen that filters with larger weight sizes do not perform better, and the 3×3 size is also able to perform well for extraction.

5.5. Comparison Results of Feature Map Fusion Methods

The feature fusion methods of summing the feature map weights, taking the mean value, and taking the absolute value maximum were proposed in Section 2. We still use Yedroudj-Net as the benchmark model, apply the fusion methods to the output of the seventh bank of filters selected in the previous section, and analyze the performance of different fusion methods on the validation set using the S-UNIWARD steganography algorithm with an embedding rate of 0.4 bpp. The results are shown in Table 3. The application models of each feature graph fusion method are called "Group7-Add-Model", "Group7-Mean-Model", and "Group7-AbsMax-Model" in order.

Table 3. Accuracy of feature map fusion methods on validation set.

Model for Applying Feature Map Fusion Method	**Accuracy (%)**
Group7-Add-Model	76.65
Group7-Mean-Model	76.70
Group7-AbsMax-Model	51.75

It can be seen that the fusion method of summing the feature map weights and taking the mean value is more effective, while the feature fusion method of taking the absolute value maximum is not effective. This is because the operation of taking the absolute value erases the negative values in the weights, which causes feature loss, which is also contrary to the original intention of using the TLU activation function in the first two convolutional layers of the HPF-Enhanced Model. We continue the experiments with Yedroudj-Net as the benchmark model to explore how best to use these fusion methods together, and to determine whether it is better to use the fusion method of summing and averaging the feature map weights or to use all three fusion methods. In addition to the feature maps after fusion, do the 30 feature maps generated by the original 30 SRM filters need to be retained? The results are shown in Table 4; we applied the fusion method to all applicable filter banks, i.e., all filter banks containing multiple filters. In the names of the models in the table, the notations "Add", "Mean", and "AbsMax" represent the models using the fusion methods of adding the weights of the feature maps, taking the mean value, and taking the absolute maximum value, respectively, while the "*" symbol represents the models retaining the feature maps generated by the original 30 filters, which coexist with the feature maps generated by the fusion methods, and the models without the "*" symbol discard the output of the 30 original feature maps.

From the results, we find that using an additional fusion method of taking the absolute maximum value does not bring a considerable improvement to the model, but it increases the computational effort by adding five channels in the preprocessing layer, so the method of taking the absolute maximum value can be discarded. Moreover, it can be seen that discarding the original 30 feature maps does not significantly reduce the detection accuracy, but it can greatly reduce the number of output channels in the preprocessing layer to improve the performance—even the model with the highest accuracy also discarded the original 30 feature maps. The Yedroudj-Add-Mean-Model performs well on both the validation and test sets and shows minimal accuracy degradation on the test set, so the HPF-Enhanced Model uses only the feature maps produced by the two

fusion methods of summing feature map weights and taking the mean as the output of the preprocessing layer.

Table 4. Accuracy of different fusion methods.

Model	Accuracy on Validation Set (%)	Accuracy on Test Set (%)	Number of Output Channels of Preprocessing Layer
Yedroudj-Add-Mean-Model *	77.20	75.92	40
Yedroudj-Add-Mean-Model	78.60	78.03	12
Yedroudj-Add-Mean-AbsMax-Model *	77.85	76.84	45
Yedroudj-Add-Mean-AbsMax-Model	77.65	76.64	17

* represents the models retaining the feature maps generated by the original 30 filters.

5.6. Comparison Results with Other Models

In Table 5, we compare the HPF-Enhanced Model with the advanced classical steganalysis model on the test set. Each model is trained on the S-UNIWARD steganography algorithm with embedding rates of 0.2 bpp and 0.4 bpp, respectively, using the training and validation sets to complete the training, save the structure and parameters in the optimal state, and execute the test code on the test set. The test set contains 10,000 images. One of the advantages of the HPF-Enhanced Model, which uses cross-layer enhancement, is that the number of model parameters is reduced, making the model smaller. To check the effect of this parameter's reduction on model performance, we designed the HPF-IncompletelyEnhanced Model, which uses increased neurons instead of the HPF-Enhanced Model's preprocessing layer to pass backward the number of channels across layers, to ensure that both models have the same number of channels for each layer's input and output. In addition, we have labeled the size of the file storing each model in the table.

Table 5. Comparison results of models.

Model	Accuracy (%)		Model Size
	0.2 bpp	0.4 bpp	
Xu-Net	60.94	74.71	134 KB
Yedroudj-Net	56.01	77.06	3.41 MB
HPF-IncompletelyEnhanced Model	67.37	81.94	1.15 MB
HPF-Enhanced Model	66.35	81.69	1.10 MB

The preprocessing-enhanced HPF-Enhanced Model using filter extraction enhancement and filter cross-layer enhancement achieves a steganalysis capability that exceeds that of the classical model and maintains a smaller model size, with a 4.63% accuracy improvement compared to Yedroudj-Net, where the HPF-Enhanced Model is only one third of its size. In comparison with the HPF-IncompletelyEnhanced Model, the cross-layer enhancement reduces the size of the HPF-Enhanced Model by 0.05 MB with little loss of accuracy; moreover, this is a small network, and if this strategy was applied to a large network, the effect of cross-layer enhancement on model streamlining would be more obvious. Undoubtedly, the preprocessing enhancement method described in this paper is also well suited for constructing a lightweight spatial domain steganalysis model with strong performance.

6. Conclusions

In this paper, we verified that for spatial domain steganalysis, the improvement obtained by improving the feature extraction capability of the preprocessing layer is not smaller than the improvement obtained by optimizing the network structures, and the

extraction capability of the preprocessing layer is equally important for spatial domain steganalysis. The HPF-Enhanced Model, which applies the preprocessing enhancement method proposed in this paper, shows good results and brings another benefit: it can significantly reduce the size of the model, which is suitable for building lightweight networks. In future work, we will further explore the feasibility of applying the preprocessing enhancement method to large networks and investigate its impact on the model mismatch capability.

Author Contributions: Conceptualization, X.D. and C.Z.; Data curation, S.L.; Formal analysis, X.D. and S.L.; Funding acquisition, C.Z.; Investigation, X.D. and S.L.; Methodology, X.D.; Project administration, X.D. and C.Z.; Resources, S.L.; Software, X.D.; Supervision, X.D. and C.Z.; Validation, X.D., Y.M. and S.L.; Visualization, X.D.; Writing—original draft, X.D. and S.L.; Writing—review and editing, X.D. and Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Hebei Province Professional Degree Teaching Case Establishment and Construction Project (No. KCJSZ2022073) and Hebei Postgraduate Course Civic Politics Model Course and Teaching Master Project (No. YKCSZ2021091).

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous reviewers and associate editor for their comments, which greatly improved this paper.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Kurak, C.; McHugh, J. A cautionary note on image downgrading. In Proceedings of the Proceedings Eighth Annual Computer Security Application Conference, San Antonio, TX, USA, 30 November–4 December 1992; pp. 153–159.
2. Pevný, T.; Filler, T.; Bas, P. Using high-dimensional image models to perform highly undetectable steganography. In *International Workshop on Information Hiding*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 161–177.
3. Holub, V.; Fridrich, J. Designing Steganographic Distortion Using Directional Filters. In Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS'2012, Tenerife, Spain, 2–5 December 2012; pp. 234–239.
4. Holub, V.; Fridrich, J.; Denemark, T. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Inf. Secur.* **2014**, *2014*, 1. [CrossRef]
5. Li, B.; Wang, M.; Huang, J.; Li, X. A New Cost Function for Spatial Image Steganography. In Proceedings of the 2014 IEEE International Conference on Image Processing, Paris, France, 27–30 October 2014; pp. 4206–4210.
6. Fridrich, J.; Kodovsky, J. Rich Models for Steganalysis of Digital Images. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 868–882. [CrossRef]
7. Qian, Y.; Dong, J.; Wang, W.; Tan, T. Deep learning for steganalysis via convolutional neural networks. In *Proceedings Volume 9409, Media Watermarking, Security, and Forensics*; SPIE: Bellingham, WA USA, 2015.
8. Xu, G.; Wu, H.; Shi, Y. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Process. Lett.* **2016**, *23*, 708–712. [CrossRef]
9. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *J. Mach. Learn. Res.* **2015**, *37*, 448–456.
10. Fan, E. Extended tanh-function method and its applications to nonlinear equations. *Phys. Lett. A* **2000**, *277*, 212–218. [CrossRef]
11. Ye, J.; Ni, J.; Yang, Y. Deep Learning Hierarchical Representations for Image Steganalysis. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2545–2557. [CrossRef]
12. Yedroudj, M.; Comby, F.; Chaumont, M. Yedroudj-Net: An efficient CNN for spatial steganalysis. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, AB, Canada, 15–20 April 2018; pp. 2092–2096.
13. Li, B.; Wei, W.; Ferreira, A.; Tan, S. ReST-Net: Diverse Activation Modules and Parallel Subnets-Based CNN for Spatial Image Steganalysis. *IEEE Signal Process. Lett.* **2018**, *25*, 650–654. [CrossRef]
14. Boroumand, M.; Chen, M.; Fridrich, J. Deep residual network for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1181–1193. [CrossRef]
15. Shen, J.; Liao, X.; Qin, Z.; Liu, X. Spatial Steganalysis of Low Embedding Rate Based on Convolutional Neural Network. *J. Softw.* **2021**, *32*, 2901–2915.
16. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
17. He, K.; Zhang, X.; Ren, S.; Jian, S. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

18. Szegedy, C.; Wei, L.; Jia, Y.; Pierre, S.; Scoot, R.; Dragomir, A.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA,, 7–12 June 2015; pp. 1–9.
19. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv* **2015**, arXiv:1505.00853.
20. Bas, P.; Filler, T.; Tomás, P. "Break Our Steganographic System": The Ins and Outs of Organizing BOSS. In *Information Hiding*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 59–70.