symmetry

MDPI

Article

Deep-Learning Steganalysis for Removing Document Images on the Basis of Geometric Median Pruning

Shangping Zhong [1,*], Wude Weng [1], Kaizhi Chen [1] and Jianhua Lai [2]

[1] Department of Mathematics and Computer Science, Fuzhou University, Fuzhou 350100, China; n180327068@fzu.edu.cn (W.W.); ckz@fzu.edu.cn (K.C.)

[2] Fujian Institute of Scientific and Technological Information, Fuzhou 350003, China; laijh@heidun.net

* Correspondence: spzhong@fzu.edu.cn

Received: 31 July 2020; Accepted: 25 August 2020; Published: 28 August 2020

Abstract: The deep-learning steganography of current hotspots can conceal an image secret message in a cover image of the same size. While the steganography secret message is primarily removed via active steganalysis. The document image as the secret message in deep-learning steganography can deliver a considerable amount of effective information in a secret communication process. This study builds and implements deep-learning steganography removal models of document image secret messages based on the idea of adversarial perturbation removal: feed-forward denoising convolutional neural networks (DnCNN) and high-level representation guided denoiser (HGD). Further—considering the large computation cost and storage overheads of the above model—we use the document image-quality assessment (DIQA) as threshold, calculate the importance of filters using geometric median and prune redundant filters as extensively as possible through the overall iterative pruning and artificial bee colony (ABC) automatic pruning algorithms to reduce the size of the network structure of the existing vast and over-parameterized deep-learning steganography removal model, while maintaining the good removal effects of the model in the pruning process. Experiment results showed that the model generated by this method has better adaptability and scalability. Compared with the original deep-learning steganography removal model without pruning in this paper, the classic indicators params and flops are reduced by more than 75%.

Keywords: active steganalysis; deep-learning steganography removal; document image secret message; geometric median pruning; ABC automatic pruning; DIQA

1. Introduction

Image steganography is a technique for concealing secret messages in cover images and transmitting stego images to complete transmission of secret messages in a common channel. The receiving end of the transmission can leak the secret message. In recent years, deep-learning frameworks, especially convolutional neural networks (CNNs) have recently achieved remarkable superiority over conventional approaches in many fields. In the meanwhile, from early AlexNet [1] and VGGNet [2], to later more advanced Inception models [3] and ResNet [4]. Deep learning has also been introduced to the field of information hiding. Deep-learning steganography has progressed considerably with larger payloads in secret messages than the traditional steganography and successfully distributes secret messages to available bits of the cover image. However, lossy deep steganography limits secret messages to images. Baluja [5] proposed a CNN based on the encoder–decoder structure, the encoder network can successfully conceal a secret image into a same-size cover image and the decoder network can reveal the secret image completely. Wu et al. [6] put forward a deep-learning steganography based on CNN architecture to provide better payload and performance compared with the traditional steganography method. Dong et al. [7] offered a deep-learning steganography called ISGAN by

introducing generative adversarial networks (GANs) into CNN networks while enhancing security and increasing invisibility of the secret message by minimizing the difference between empirical probability distributions of stego and natural images. Existing studies on steganography fail to use document images as secret steganography message. Unlike general images, document images can transmit a large amount of text messages, its privacy and credibility need to be effectively protected in social networks [8]. The document image, which has certain practical and research significance, is used in this study as the secret message of deep-learning steganography. General image-quality assessment methods, such as peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [9], are full-reference objective evaluation algorithms for examining images, that fail to measure the removal quality of document images, Hence, this study uses document image-quality assessment (DIQA) [10] to measure the removal quality of the document image's secret message.

Steganalysis is an attack on steganography algorithms that intercepts images through listeners in a common channel and analyses the confidentiality of information contained in images. Since the introduction of steganalysis, steganography and steganalysis have been mutually reinforcing. Steganalysis [11] is a technique for detecting or deleting steganography secret messages. However, detecting the existence of hidden information and extracting such hidden information without knowing the steganography method are difficult; hence, the removal of steganography secret messages has become an important research area. Limited studies are available on the removal of secret messages of deep-learning steganography. Jung et al. [12] proposed a framework called PixelSteganalysis, which is inspired by PixelCNN [13]. The proposed deep-learning steganography removal method in this study is inspired by adversarial examples proposed by Szegedy et al. [14] that treat the hidden secret message as tiny adversarial perturbation and adopt the idea of adversarial perturbation removal as deep-learning steganography removal. For example, Jia et al. [15] put forward an end-to-end image model called ComDefend, which consists of a compression convolutional neural network (ComCNN) and a reconstruction convolutional neural network (ResCNN), to protect against adversarial sample attacks. ComCNN is used to maintain the structure information of the original image, and ResCNN is utilized to construct the original image with high quality. This study proposes a deep-learning steganography removal model with better adaptability and scalability that combines deep-learning steganography and adversarial perturbation removal methods.

Although deep neural networks progressed considerably in the field of steganography, a large number of data sets are required to obtain good performance and deployment is difficult because a substantial amount of parameters and storage overheads [16,17] are involved. proposed framework of deep-learning steganography removal based on the deep neural network in this study also has a large number of params and flops. Deploying steganography firewalls in network nodes is difficult. Model optimization has recently become an important research topic. Network pruning [18,19] is a technology that can effectively compress and accelerate CNNs and allows users to deploy efficient networks on hardware devices with limited storage and computing resources. For example, Han et al. [20] initially pruned a deep learning model by removing weights below a certain threshold. Han et al. [21] further combined weight pruning with Huffman coding; however, this method may lead to unstructured sparseness of filters that may be inefficient in saving memory usage and computing costs. Structured sparseness and effective memory usage can be achieved in the model via the filter pruning method. Therefore, filter pruning is effective in accelerating the development of deep neural networks. Accordingly, this study uses filter pruning to compress the model, reduce params and flops of the deep-learning steganography removal model and ensure the model performance.

The main contributions of this article include the following:

The idea of adversarial perturbation removal is adopted as deep-learning steganography removal, and the two types of deep-learning steganography removal models, namely, DnCNN and HGD, are implemented.

The document image is used as the secret message of the deep-learning steganography method called ISGAN [7], and the document image-quality assessment called DIQA [10] is utilized to evaluate the secret message of deep-learning steganography.

Geometric median pruning is used to analyze filters and the performance of each convolutional layer of the deep-learning steganography removal model.

DIQA and image-quality assessment [9] are used as thresholds, and iterative pruning is applied to the two proposed deep-learning steganography removal models to achieve satisfactory results.

DIQA, image-quality assessment, params and flops are used as nectar source fitness, and artificial bee colony (ABC) automatic pruning is applied to the two proposed deep-learning steganography removal models to achieve satisfactory results.

The remainder of this paper is organized as follows: Technology related to this study is discussed in Section 2. Two types of deep-learning steganography removal frameworks are proposed in Section 3. Deep-learning steganalysis for removing document images based on geometric median pruning is put forward in Section 4. The experimental results are presented in Section 5. Finally, the conclusions of this study are summarized, and future research directions are discussed in Section 6.

2. Related Technologies

2.1. Active Steganalysis

The steganography secret message is primarily removed via active steganalysis, which is divided into two categories, namely, conventional active and deep-learning steganalyses.

Fridrich et al. [22] proposed a method of conventional active steganalysis to overwrite random bits suggested by the gaussian noise or others where messages may reside. Guo et al. [23] proposed a median filter for removing noise in an image, especially sporadic noise of large variance. Amritha et al. [24] removed the hidden secret message by using a denoising filter to reduce the cover image quality and then restoring the cover image to some extent through deconvolution operations.

Jung et al. [12] proposed a framework of deep-learning steganalysis based on PixelCNN [13] called PixelSteganalysis, which exploits sophisticated pixel distributions and edge areas of images using a deep neural network. Accordingly, we adaptively remove secret information at the pixel level. Corley et al. [25] offered a deep digital steganography purifier based on GAN that destroys steganography content without compromising the perceived quality of the original image.

This study proposes the use of adversarial perturbation removal as deep-learning steganography removal. Zhang et al. [26] put forward an end-to-end trainable Gaussian denoising architecture called the denoising convolutional neural networks (DnCNN); batch normalization and residual learning are integrated to accelerate the training process and boost the denoising performance. Song et al. [27] offered a defense method using pixel-level CNN called PixelDefend and determined that adversarial examples primarily lie in low-probability regions of the training distribution; PixelDefend can reconstruct the low-probability regions of adversarial examples into a clean image that meets the requirements of high-probability regions. Liao et al. [28] proposed several high-level representation guided denoiser (HGD) methods by treating adversarial examples as noise, to achieve defensive adversarial examples.

2.2. Pruning Methods

Network pruning methods can be divided into two categories for CNNs, namely, unstructured weight pruning [20] and channel-based structured pruning. Unstructured pruning typically results in irregular network structures, which require dedicated hardware and software to support the actual acceleration. The network complexity cannot be reduced without the dedicated hardware support. Therefore, using the structured pruning method is practical.

The channel-pruning method compresses the model structure by deleting the entire filters and is appropriately supported by general hardware. Several works have evaluated the importance of filter weights. Li et al. [29] assessed the importance of filter using L_p norm; unimportant filters in the

convolution layer are deleted after artificially setting the pruning ratio. Yang et al. [30] proposed a soft filter pruning method that allows updating the pruned filters when pruning the training model; the network with good model learning ability can be trained from scratch and pruned at the same time without fine tuning to achieve a good effect. He et al. [31] offered a geometric median pruning method that prunes redundant filters by regarding the convolution kernel near the geometric center as similar and unimportant. Chen et al. [32] proposed a self-adaptive network pruning method (SANP) to reduce cost for CNNs, that introduces a general Saliency-and-Pruning Module (SPM) for each convolutional layer, which learns to predict saliency scores and applies pruning for each channel.

Several works have focused on pruning based on reconstruction errors by using channel pruning as an optimization problem and selecting representative filters. Madaan et al. [33] proposed a new loss for adversarial learning to minimize the feature-level vulnerability during training and proposed a Bayesian framework to prune features with high vulnerability in order to reduce both vulnerability and loss on adversarial samples. Luo et al. [34] proposed the ThiNet architecture that uses greedy method to delete channels with minimal effect on the activation value of the next layer. He et al. [35] used lasso regression to select channels for pruning. Yu et al. [36] determined pruning filters by minimizing reconstruction errors of the penultimate layer of the network and considering cumulative back propagation errors.

Some studies have focused on pruning based on regularization. Liu et al. [37] used scaling factor γ in the normalization layer to impose sparseness constraints, measure the importance of channels during the training process, filter out channels with low scores and prune layer-by-layer. Huang et al. [38] and Lin et al. [39] proposed a sparse regularization mask method based on channel pruning; the mask is optimized via data-driven selection or generative adversarial learning. Zhao et al. [40] further developed the norm-based importance estimation by taking the dependency between the adjacent layers into consideration and propose a novel mechanism to dynamically control the sparsity-inducing regularization so as to achieve the desired sparsity.

Other studies have focused on searchable automatic pruning methods. Guerra et al. [41] proposed an effective pruning strategy for selecting redundant low-precision filters by combining neural network quantification and pruning to realize automation process. Dong et al. [42] proposed a search architecture called transformable architecture that combines knowledge distillation and searchability to find a good network structure. Liu et al. [43] proposed a heuristic search algorithm that trains the remaining weights while pruning to obtain a structurally sparse model of weight distribution and further searches and deletes a small part of redundant weights through network structure purification. Lin et al. [44] proposed a channel-pruning method based on artificial bee colony (ABC) algorithm; searching for the optimal pruning structure is regarded as an optimization problem and the ABC algorithm is integrated to solve the problem of selecting the optimal pruning structure with the best fitness automatically.

3. Deep-Learning Steganography Removal Model

This study adopts the idea of adversarial perturbation removal as deep-learning steganography removal and implements two types of deep-learning steganography removal models, namely DnCNN [26] and HGD [28]. Approximately 50,000 ISGAN-generated [7] stego images with a size of 256 × 256 resolution and their corresponding cover images are used as data sets to train the model.

SSIM and PSNR [9] are used to measure the image quality. SSIM measures the image similarity in terms of brightness, contrast and structure, and its value is within the range $[0, 1]$. PSNR is a widely used objective evaluation assessment for images based on errors between corresponding pixels.

Self-made black and white document images and Google's tesseract document image recognizer are used to score document images. We classify approximately 40,000 document images with a size of 256 × 256 resolution and their label data sets in each rating segment for the training of the documentation evaluation model [10]. The trained model is used as DIQA [10] to measure the degree of destruction of document images as secret messages.

3.1. DnCNN Model

The DnCNN model is modified on the basis of VGG network [2] to make it suitable for image denoising. We use the DnCNN denoising model as deep-learning steganography removal (see Figure 1).

Figure 1 illustrates an end-to-end deep convolutional steganography removal network. Given a depth of D network, the different types of layer composition are presented as follows:

Convolutional layer (Conv) + rectified linear unit (ReLU): The first layer uses 64 filters with a size of $3 \times 3 \times 3$ to generate 64 feature maps. Using ReLu enables the output of some neurons to tend toward zero and lead to a sparse network with improved mining-related features.

Conv + batch normalization (BN) + ReLU: The 2 ~(D-1) layers use 64 filters with a size of $3 \times 3 \times 64$. The BN layer is added between Conv and ReLU to accelerate the training and improve the performance of steganography removal. The document image regarded as a secret image in the stego image is gradually removed as noise in the layer-by-layer iteration.

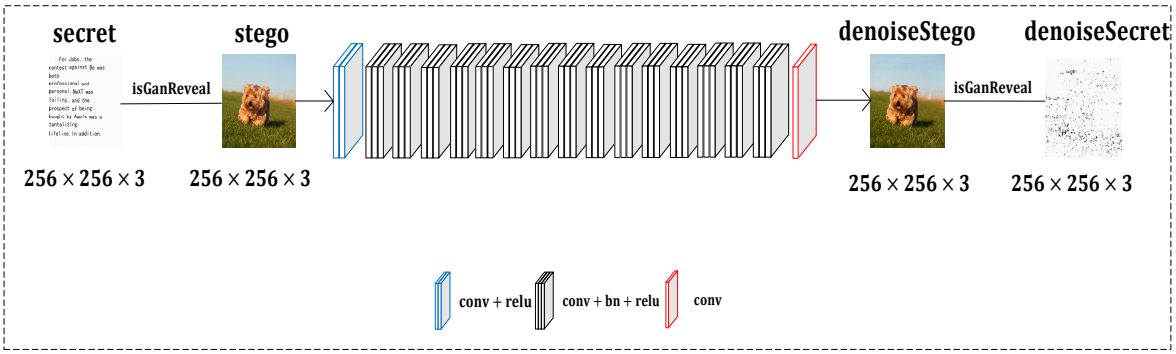


Figure 1. Deep-learning steganography removal model called the feed-forward denoising convolutional neural network (DnCNN).

Conv: The last layer uses three filters with a size of $3 \times 3 \times 64$ to reconstruct the image as the output of the deep-learning steganography removal model.

$$loss = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} |C_{i,j} - R_{i,j}|}{I \odot J} \quad (1)$$

The loss function of DnCNN is expressed as Formula (1), where C represents the original image; R represents the image after removing the steganography secret message; and I and J represent the length and width of the image, respectively. The difference between the original and purified images is used as the loss function to ensure that the image is as close to the original image as possible after removing the secret message as well as improves the SSIM and PSNR values of original and purified images as well as stego and purified images.

3.2. HGD Model

Denoising autoencoder (DAE) [45] is a popular denoising model, DAE has a bottleneck structure between the encoder and decoder. This bottleneck may constrain the transmission of fine-scale information necessary for reconstructing high resolution images. Hence, the HGD model was proposed by modifying DAE with *U*-net [46] structure to overcome the bottleneck. We use the HGD denoising model as deep-learning steganography removal (see Figure 2).

The entire model consists of a feedforward path and a feedback path and is divided into upper and lower layers (see Figure 2). HGD is primarily composed of 3×3 convolutional layers, BN and ReLU (Conv + BN + ReLU) to form a block and the last layer of 1×1 convolutional layer as the output layer. The upper layer network structure that uses 256×256 image X^* as the input and generates a set of feature maps with increasingly low resolution is called the Com layer, which is composed of 14 blocks.

The lower layer network that primarily enlarges the size of feature maps through upsampling and then merges with the output of a certain level of the Com layer as the input of the feedback path to connect the upper and lower layers is called the Res layer, which is composed of 11 blocks and a convolutional layer. The resolution of feature maps continues to increase and restores to the original image size through the feedback path.

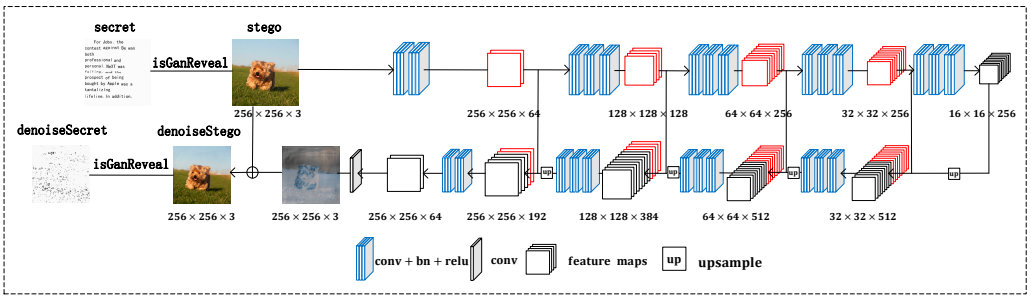


Figure 2. Deep-learning steganography removal model called the high-level representation guided denoiser (HGD).

The last output layer of HGD is a residual image $-Y^*$, the final $\hat{X} = X^* - Y^*$ as a purified image after steganography removal. The secret message of the hidden document image in the stego image is removed as small perturbations through compression and reconstruction. HGD uses a loss function consistent with DnCNN to ensure the quality of the purified image.

The DnCNN and HGD models are trained for about three days under a single GeForce GTX 1080 Ti graphics card. The performance of the final DnCNN and HGD is presented in Table 1.

Table 1. Performance of the DnCNN and HGD models.

Model	Orig_Remove_ SSIM	Orig_Remove_ PSNR	Stego_Remove_ SSIM	Stego_Remove_ PSNR	DIQA	Params (M)	Flops (G)
DnCNN	0.961	29,323	0.966	33,696	0.068	0.558	36,591
HGD	0.978	31,618	0.963	30,868	0.069	11,034	50,937

According to the results in Table 1, the trained DnCNN and HGD deep-learning steganography removal models have excessively large amount of params and flops, which are unfavorable for deployment in network nodes. Therefore, pruning is performed on the trained deep-learning steganography removal model. Redundant filters are pruned as much as possible to reduce the size of the network structure of the existing vast and over-parameterized deep-learning steganography removal model while ensuring that the secret message of the hidden document image is invisible to a certain extent. The specific method is discussed in Section 4.

4. Methods

4.1. Pruning Strategy for the Deep-Learning Steganography Removal Model

The filter pruning method allows the model to have structured sparseness and effective memory usage. Therefore, filter pruning is preferred in the process of accelerating the development of deep neural networks. Therefore, this study applies the filter pruning method of geometric median to compress the deep-learning steganography removal model.

DnCNN and HGD models generally utilize two pruning methods. As shown in Figures 3 and 4, let x_i and y_i represent the width and height of the feature map, respectively; P_i represents the feature map, where $P_i \in R^{n_i \times x_i \times y_i}$; R represents the set; $R^{n_i \times x_i \times y_i}$ represents a set of three dimensions for P_i; n_i represents the number of input channels of the i_{th} convolution layer; n_{i+1} represents the number of output channels of the i_{th} convolution layer; K represents the convolution kernel of the convolutional

layer, where $K \epsilon R^{k \times k}$($K$ is generally 1 or 3); $R^{k \times k}$ represents a set of two dimensions for K; $F_{i,j}$ represents a filter of the convolutional layer, and all filters of the layer constitute a kernel matrix $Q_i \in R^{n_i \times n_{i+1} \times k \times k}$; $R^{n_i \times n_{i+1} \times k \times k}$ represents a set of four dimensions for Q_i.

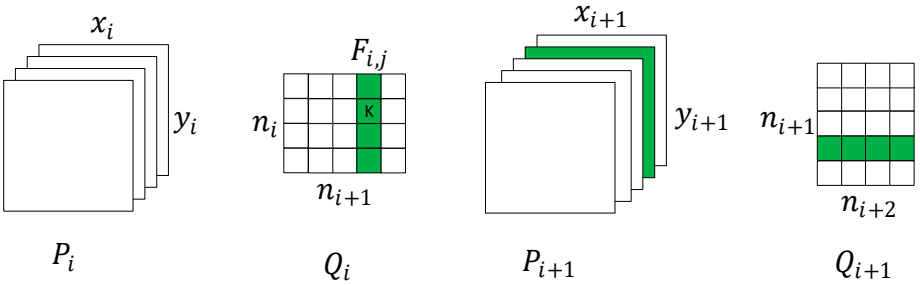


Figure 3. General pruning form in the base layer of DnCNN and HGD.

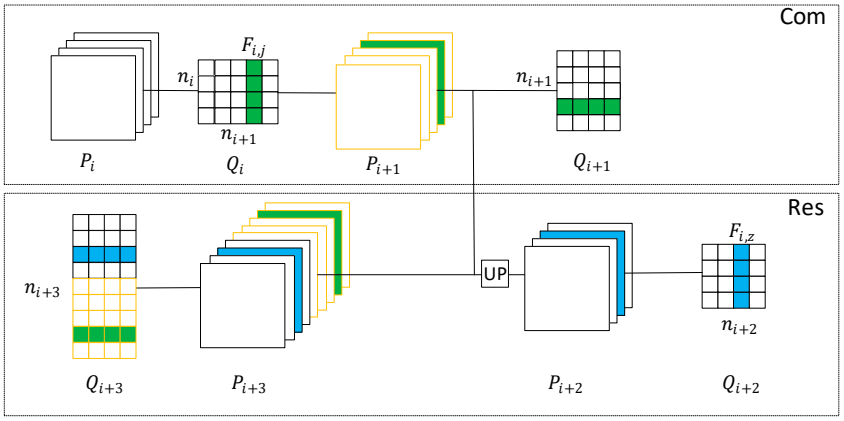


Figure 4. Upper and lower layers of HGD with connected pruning form.

The network structure shown in Figure 3 prunes a filter $F_{i,j}$ in the convolution layer kernel matrix Q_i, affects the number of corresponding channels of the output feature maps P_{i+1} and uses pruned feature maps as the input of the next convolution layer. Hence, the corresponding kernel matrix Q_{i+1} of the next convolutional layer should also be removed accordingly. Hence, pruning a filter $F_{i,j}$ in the model will reduce $n_i \times k^2 \times x_{i+1} \times y_{i+1}$ flops.

The HGD network structure is illustrated in Figure 4. Pruning a filter $F_{i,j}$ in the convolutional layer kernel matrix Q_i in the Com layer will affect corresponding channels of output feature maps P_{i+1} in the Com layer and the corresponding kernel matrix Q_{i+1} of the next convolutional layer, and then the corresponding weights in n_{i+2}–n_{i+3} of the kernel matrix Q_{i+3} in the Res layer must be removed.

4.2. Geometric Median Pruning

We analyze all filters of each convolutional layer of the proposed DnCNN and HGD via geometric median. The geometric median minimizes the sum of Euclidean distances to all filters $F_{i,j}$ in i_{th} convolutional layers. The information center of all filters in the convolutional layer of this layer is expressed as Formula (2).

$$f = \underset{t \epsilon R^{n_{i+1} * k * k}}{argmin} \sum_{j \epsilon [1, n_{i+1}]} \| t - F_{i,j} \|_2 \quad (2)$$

Filters at or near the geometric median containing redundant information can be replaced with the remaining filters, as shown in Formula (3).

$$F_{i,j'} = \underset{F_{i,j}}{argmin} \| F_{i,j} - f \|_2, \text{s.t. } j \epsilon [1, n_{i+1}] \quad (3)$$

4.3. Overall Iterative Pruning of Deep-Learning Steganography Removal Model Based on Geometric Median

Geometric median is used in this study to prune the pre-trained deep-learning steganography removal model, as shown in Algorithm 1.

The document image can contain a large amount of information when used as a secret message in deep-learning steganography. Hence, the deep-learning steganography removal model is used as the steganography firewall to remove the hidden secret message in stego images. The removal model proposed in this study uses DIQA as the threshold to evaluate the degree of secret message removal while ensuring that the steganography secret message is invisible to a certain extent. In addition, the quality of the images must be considered, that is, the PSNR and SSIM indicators of the original and purified images, stego images and purified images. Algorithm 2 sets the DIQA threshold, SSIM > 0.9 and PSNR > 26 and iteratively prunes every layer in each convolution layer of the proposed DnCNN and HGD.

Algorithm 1 Pruning the deep-learning steganography removal model via geometric median

1: Prepare the pre-trained DnCNN and HGD models;
2: Calculate geometric median on all filters $F_{i,j}$ of a convolutional layer in the deep-learning steganography removal model, as in Formula (2), find the data center point f of filters in the convolutional layer;
3: Filters $F_{i,j'}$ that have redundant information are closed to the geometric median according to Formula (3). We prune these redundant filters to achieve the purpose of pruning, while maintaining the performance of the models;
4: Prune a filter in a convolution layer, which will affect the kernel matrix and corresponding feature map channels in the next convolutional layer, As analyzed in 4.1. Therefore, it is necessary to remove the number of feature maps channels and corresponding weights involved in pruning. and to match the number of input and output channels of the relevant convolution layer;
5: Retain the remaining kernel matrix after removing filters of one convolution layer of the deep-learning steganography removal model, complete a filter pruning operation based on geometric median.

Algorithm 2 Iterative pruning process of the deep-learning steganography removal model

1: Input: Prepare the pre-trained DnCNN and HGD models;
2: Initialization: Set m pruning layers; set the maximum channels pruning rate r; set the channels iteration pruning rate p of the deep-learning steganography removal model. According to the maximum channels pruning rate and channels iterative pruning rate, set the number of iterative pruning channels of each convolutional layer, C $= (c_1, c_2, \cdots c_n), n = \frac{1}{p}, 0 \leq c \leq$ channel $\times$ r;
3: Conditions: Set the DIQA threshold while ensuring the image quality, that is, SSIM > 0.9 for the original image and purified image, SSIM > 0.9 for the stego image and purified image, PSNR > 26 for the original image and purified image, PSNR > 26 for the stego image and purified image;
4: **for** i = 1:m **do**
    for j in C **do**
        Call Algorithm 1 to perform a geometric median pruning;
        Verify the network after each pruning. If the network after pruning meets the DIQA threshold and image-quality assessment, we should prune more channels for this convolutional layer. If the network after pruning does not meet the DIQA threshold and image-quality assessment, jump out of this layer loop, determine the final pruning result of the convolutional layer, save the pruning model and start pruning of the next convolutional layer;
    end
end
5: Output: The pruned models.

4.4. ABC Automatic Pruning of Deep-Learning Steganography Removal Model Based on Geometric Median

Section 4.3 does violent iterative pruning in 90% channel pruning space of each convolutional layer of the deep-learning steganography removal model. The process considers the convolutional

layer of the deep-learning steganography removal model separately and does not realize the automatic process. Thus, this section shrinks the combinations where the preserved channels of the deep-learning steganography removal model are limited to a specific space based on geometric median. Moreover, then, we formulate the search of optimal pruned structure as an optimization problem and integrate the ABC algorithm to solve it in an automatic manner.

The ABC algorithm contains three basic elements: nectar sources, employed bees and unemployed bees and three basic behavior models: search for nectar sources, recruit bees for nectar sources and give up a nectar source.

- **Nectar sources**: Its value is composed of many factors, such as the amount of nectar, the distance from the hive and the difficulty of obtaining nectar. The fitness of nectar source is used to express the above factors;
- **Employed bees**: The number of employed bees and nectar sources is usually equal. Employed bees have memory function to store relevant information of a certain nectar source, including the distance, direction and abundance of nectar source and share this information with a certain probability to other bees;
- **Unemployed bees**: The responsibility of unemployed bees including onlooker bees and scout bees is to find the nectar source to be mined. Onlooker bees observe the swinging dance of the employed bees to obtain important nectar source information and choose the bees that they are satisfied with to follow. The number of onlooker bees and employed bees is equal. Scout bees that account for 5–20% of total bee colonies do not follow any other bees and randomly search for nectar sources around the hive.

The corresponding relationship between the foraging behavior of ABC and the channel pruning problem of the deep-learning steganography removal model is shown in Table 2. Thus, the optimization problem of structure search of the deep-learning steganography removal model is abstracted into the foraging behavior of bees.

Table 2. The corresponding relationship between the foraging behavior of ABC and the channel pruning problem of the deep-learning steganography removal model.

Foraging Behavior of ABC	Channel Pruning Problem of the Deep-Learning Steganography Removal Model
Nectar sources	The combinations of each convolutional layer channels of model.
Quality of nectar sources	The quality of combinations is achieved by calculating the combination fitness value, that is, sets the DIQA threshold, SSIM > 0.9, PSNR > 26 and params and flops of the model as small as possible.
Optimal quality of nectar sources	The params and flops of the model are the smallest and the image quality and document image quality are guaranteed.
Pick nectar	Search the pruning structure of the model.

4.4.1. Initialization of Nectar Sources

The combinations of each convolutional layer channels in deep-learning steganography removal are regarded as nectar sources, and the quality of the nectar sources corresponds to the fitness value f_i of combinations. Let D represents the number of convolutional layers participating in combinations. The position of nectar sources $i = \{1, 2, \cdots n\}$ is expressed as $X_i^t = \left[x_{i1}^t, x_{i2}^t, \cdots, x_{iD}^t\right]$. $x_{id} \in (L_d, U_d)$ represents a convolutional layer of the deep-learning steganography removal model. Let $d\epsilon\{1, 2, \cdots, D\}$, L_d and U_d represent the lower and upper limits of the search pace, respectively. The lower limit means that a certain convolutional layer of the model is not pruned, and the upper limit means that a certain convolutional layer of the model is pruned 90% channels. The initial position of nectar i is randomly generated in the search space according to Formula (4).

$$x_{id} = L_d + rand(0, 1)(U_d - L_d) \tag{4}$$

4.4.2. Search Process of Employed Bees

At the beginning of the search, the employed bee searches for a new nectar source near the old nectar source i according to Formula (5). Among them, $j \neq i$, $j \epsilon \{i = 1, 2, \ldots n\}$ means to randomly select a nectar source not equal to i from n nectar sources. $\varnothing_{id}$ is a uniformly distributed random number $[-1, 1]$, which determines the degree of pruning of each convolutional layer channels. x'_{id} is a new nectar source, calculated on the basis of comparing the previous nectar source x_{id} with the neighbor nectar source. If the quality of the new nectar source f_{X_i} is better than the quality of the previous nectar source $f_{X'_i}$, the employed bees memorize the new nectar source, otherwise the old nectar source is memorized.

$$x'_{id} = x_{id} + \varnothing_{id}\left(x_{id} - x_{jd}\right) \tag{5}$$

4.4.3. Search Process of Onlooker Bees

Onlooker bees select nectar sources through roulette from the nectar sources searched by employed bees, and the probability of a nectar source is selected according to Formula (6). f_{X_i} is the quality of the nectar source i. After the onlooker bee chooses the nectar source, it also searches for a new nectar source according to Formula (5):

$$P_{X_i} = 0.9\frac{min\left(f_{X_i}\right)}{f_{X_i}} + 0.1 \tag{6}$$

4.4.4. Search Process of Scout Bees

There is an important parameter *limit* in the search process of the scout bees in order to prevent the algorithm from falling into a local optimum, which is responsible for controlling the number of iterations that the quality of the nectar source has not improved. If the nectar source X_i reaches the *limit* threshold after $trial_i$ iterations and no better nectar source is found, the nectar source X_i will be abandoned. The role of the corresponding employed bee will become a scout bee, and the scout bee will randomly generate a new nectar source in the search space to replace X_i. If the *limit* threshold is not reached, the search process of the employed bee will continue, as in Formula (7):

$$X_i^t = \begin{cases} L_d + rand(0,1)(U_d - L_d), trial_i \geq limit \\ X_i^{t-1}, trial_i < limit \end{cases} \tag{7}$$

Using the ABC automatic pruning algorithm based on geometric median is not easy to fall into local extreme points because the role of scout bee searchers for new nectar sources around the hive and can converge to an optimal nectar source with maximum probability. An optimal deep-learning steganography removal model is obtained. The deep-learning steganography removal algorithm based on ABC pruning is as Algorithm 3.

Algorithm 3 ABC automatic pruning of deep-learning steganography removal model based on geometric median

1: Input: Prepare the pre-trained DnCNN and HGD models.
2: Initialization: Set t pruning rounds; initialization of nectar sources according to Formula (4); set n pruning layers; maximum channel pruning φ of each convolutional layer of the model; maximum of poor quality of nectar source is *limit*; the number of iteration search for poor quality of nectar source is *trail*.
3: Conditions: Set the DIQA threshold while ensuring the image quality, that is, SSIM > 0.9 for the original image and purified image, SSIM > 0.9 for the stego image and purified image, PSNR > 26 for the original image and purified image, PSNR > 26 for the stego image and purified image, params and flops as small as possible. Use the above conditions as the nectar source fitness value f_{X_i}.
4: **for** i = 1:t **do**
for j = 1:D **do**
The employed bee searches for a new nectar source X_i' around the nectar source X_i through Formula (5), and calls algorithm 1 to obtain the combinations of each convolutional layer channels of the model, and calculates the fitness value $f_{X_i'}$;
if $f_{X_i} < f_{X_i'}$ **then**
$X_i = X_i'$;
$f_{X_i} = f_{X_i'}$;
$trail_i = 0$;
else
$trail_i = trail_i + 1$;
end
for j = 1:D **do**
Calculate the probability of nectar being selected through Formula (6);
Generate a random number $\theta_i \in [0, 1]$;
if $\theta_i <= P_{X_i}$ **then**
The employed bee searches for a new nectar source X_i' around the nectar source X_i through Formula (5), and calls algorithm 1 to obtain the combinations of each convolutional layer channels of the model, and calculates the fitness value $f_{X_i'}$;
if $f_{X_i} < f_{X_i'}$ **then**
$X_i = X_i'$;
$f_{X_i} = f_{X_i'}$;
$trail_i = 0$;
else
$trail_i = trail_i + 1$;
end
for j = 1:n **do**
if $trail_i > limit$
thenperform Formula (7);
end
end
5: endend Output: The pruned models.

4.5. Analysis of Algorithm

Convolutional layers, except the last output layer, are pruned in each layer to satisfy the following conditions: DIQA as the threshold and image-quality assessment of SSIM > 0.9, PSNR > 26. Moreover, iterative search for the number of channels in each convolutional layer. It can ensure that each convolutional layer of the deep-learning steganography removal model can prune redundant filters to the maximum extent and the amount of params and flops of the models can be remarkably reduced while ensuring a certain degree of invisible secret message and the quality of the images. Given that iterative pruning is performed using a double-layered cyclic structure, the time complexity of the algorithm is $O(m \times n)$. The specific experimental results are presented in Section 5.

The ABC automatic pruning algorithm, using the combinations of pruning channels of each convolutional layer of the deep-learning steganography removal model as nectar sources and performing random target search by combining probability rules according Formula (6) without prior knowledge, is robust and adaptability. In addition, the employed bee and the onlooker bee form a positive feedback mechanism when looking for the optimal deep-learning steganography removal structure, which speeds up the convergence of the algorithm.

5. Experiments

5.1. Experimental Preparation and Environment

This experiment prepares the trained deep-learning steganography called ISGAN. The cover image data set uses the ILSVRC2012 data set, which is scaled and randomly cut into 256 × 256 resolution, to select 50,000 pieces. Document images are regarded as secret images using 50,000 self-made black-and-white document images.

Prepare the trained document image evaluation model as DIQA. Details are presented in Section 3.

Prepare the pre-trained deep-learning steganography removal models called DnCNN and HGD. Train both deep-learning steganography removal models using the same data set for comparability. Use the corresponding 50,000 stego images generated by ISGAN and the corresponding original images as the data set. The data set image size is 256 × 256 resolution. In addition, the same loss function and optimizer SGD are used for training. Details are presented in Section 3.

All experiments are completed using the PyTorch platform with development language of Python v.3.6, which is accelerated with 1 NVIDIA GTX 1080Ti graphics card (NVIDIA, Santa Clara, CA, USA).

5.2. Results of Pruning Experiments

DnCNN has 17 convolutional layers (convolutional layers are counted from left to right in Figure 1), and the last layer is the output layer without pruning. HGD has 26 convolutional layers (convolutional layers are counted from left to right in the Com layer and then from right to left in the Res layer, see Figure 2), and the last layer is not pruned.

Pruning experiments in this study are primarily conducted using the DnCNN and HGD models. We analyze the models via individual pruning, overall iterative pruning and ABC automatic pruning. Individual pruning gradually prunes 90% of each convolutional layer and analyses the sensitivity of each convolutional layer of the deep-learning steganography removal model. Overall, iterative pruning is the maximum pruning of each layer under the conditions of SSIM, PSNR and DIQA threshold. Ensuring the image quality and invisibility of the secret message to a certain extent is necessary after pruning. ABC automatic pruning uses DIQA threshold, SSIM, PSNR, params and flops as the fitness of the nectar source and uses the combinations of pruning channels of each convolutional layer of the deep-learning steganography removal model as the nectar source to search the structure of the DnCNN and HGD models automatically.

5.2.1. Sensitivity Analysis of the Individual Pruning

The sensitivity of pruning of each convolutional layer is different in the process of geometric median pruning of the 16 convolutional layers of the DnCNN model. The performance of PSNR, SSIM and DIQA begin to decrease to a certain extent when pruning the 50% filters. A few convolutional layers, such as conv16 shown in Figure 5. The similar performance of PSNR, SSIM and DIQA compared with the proposed and original models indicate that the conv16 layer filter has significant redundancy when pruning the 90% filters.

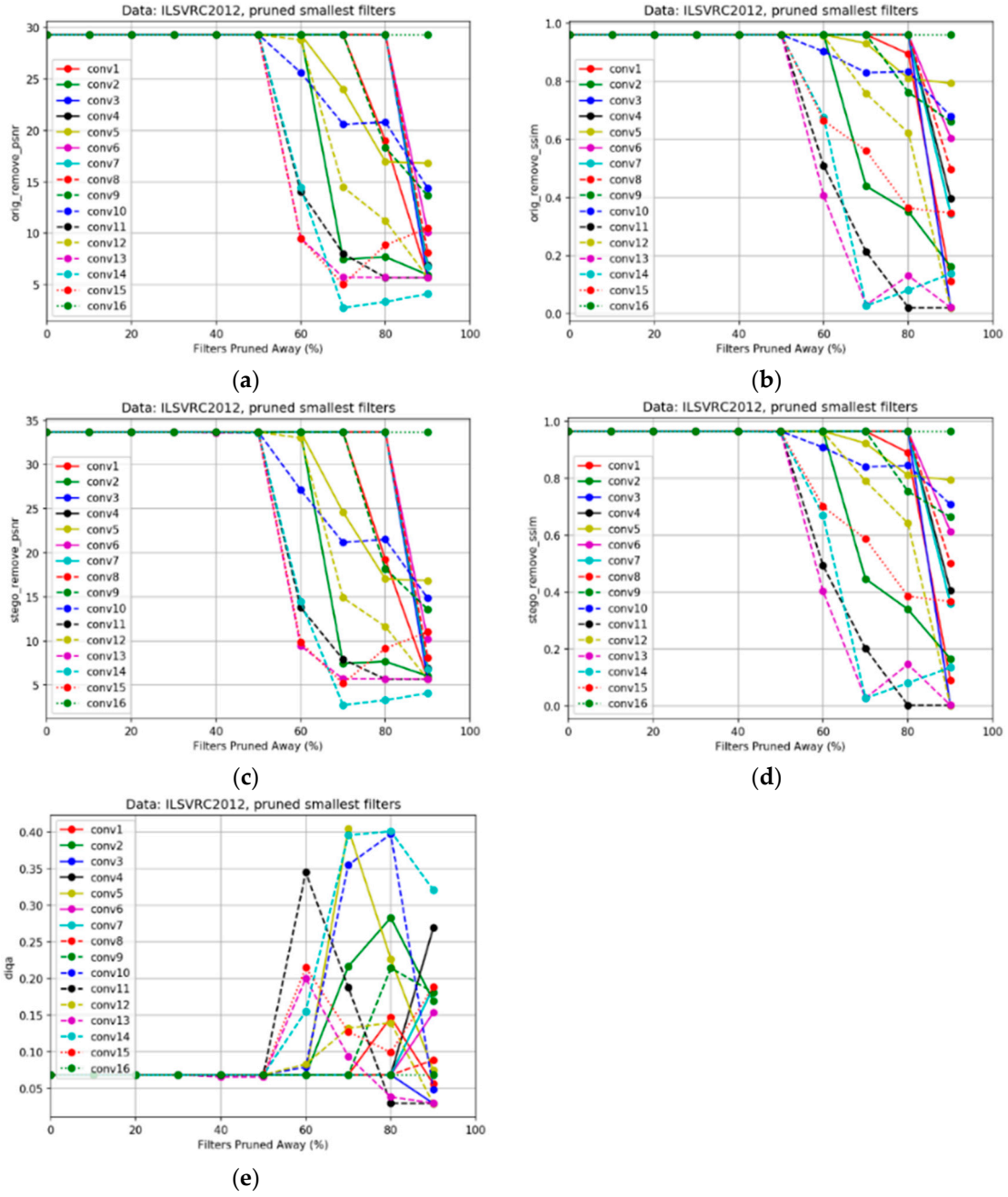
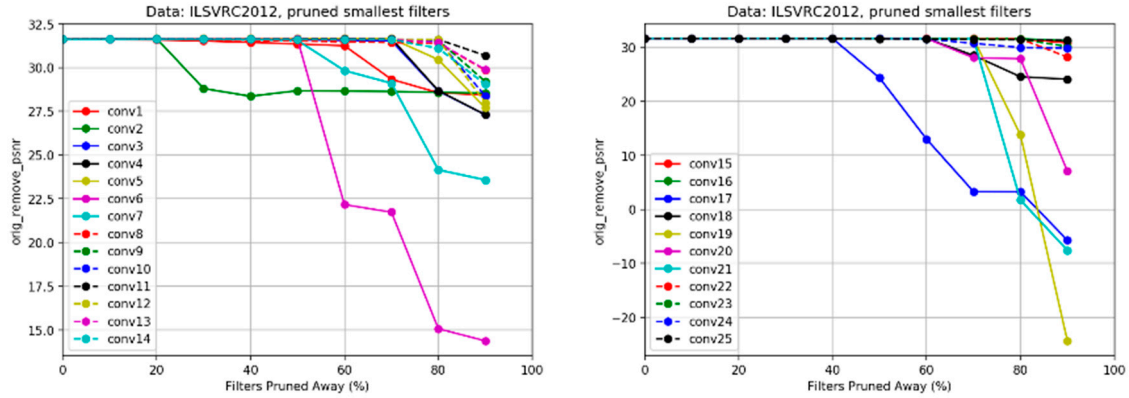
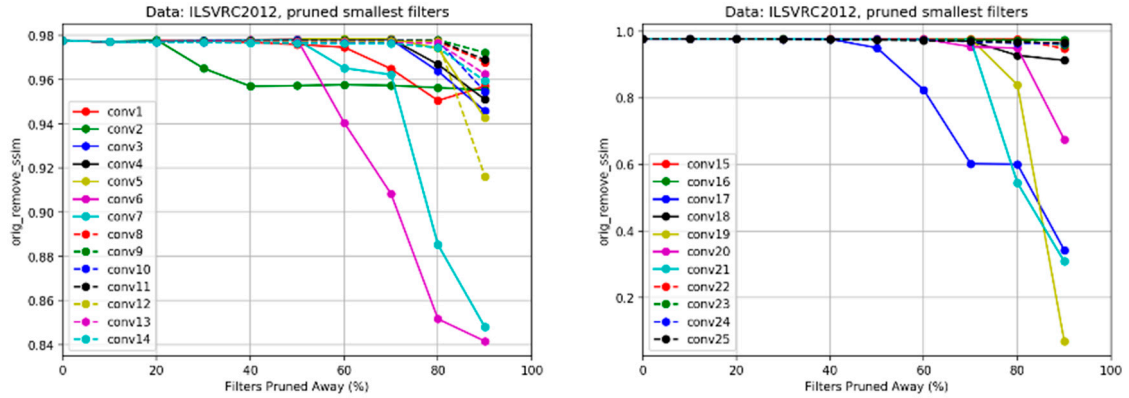


Figure 5. Sensitivity analysis of each convolutional layer of the DnCNN model based on geometric median pruning. (**a**) peak signal-to-noise ratio (PSNR) value between the original image and the purified image of each convolutional layer pruning 0–90%, (**b**) structural similarity index measure (SSIM) value between the original image and the purified image of each convolutional layer pruning 0–90%, (**c**) PSNR value between the stego image and the purified image of each convolutional layer pruning 0–90%, (**d**) SSIM value between the stego image and the purified image of each convolutional layer pruning 0–90%, (**e**) document image-quality assessment (DIQA).

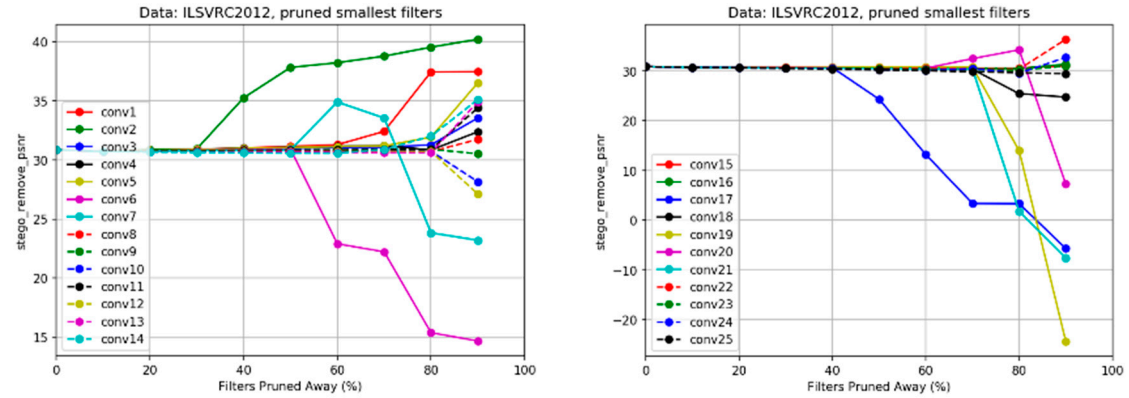
Several convolutional layers are very sensitive in the process of geometric median pruning the 25 convolutional layers of the HGD model, as shown in the conv2 layer of Figure 6. The performance of PSNR, SSIM, and DIQA in the model begins to have a certain impact when pruning the 30% filters.



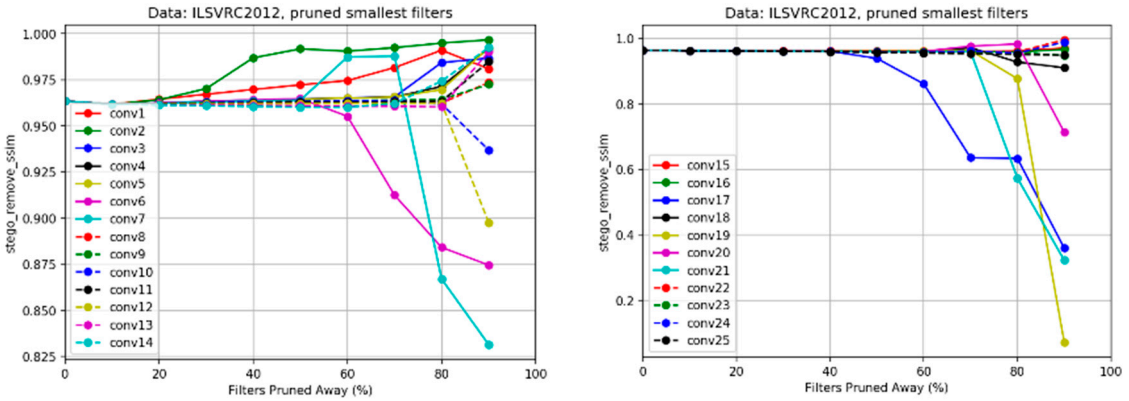
(**a**)



(**b**)

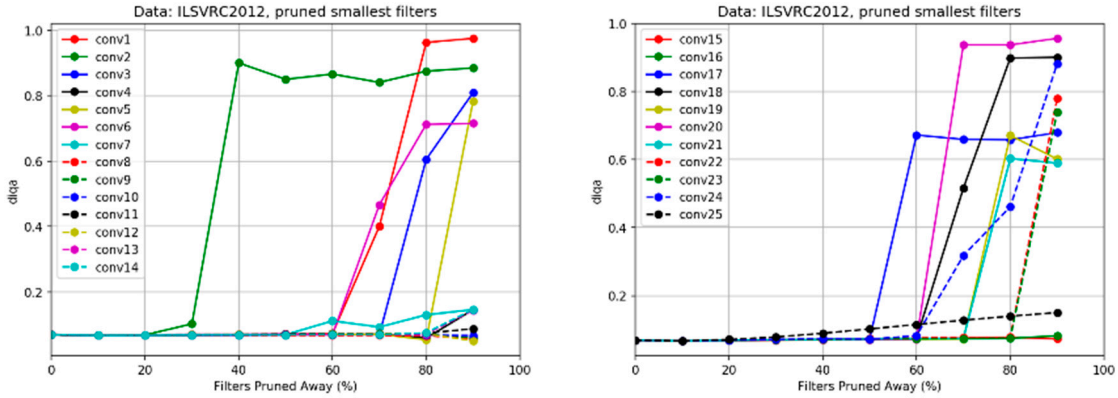


(**c**)



(**d**)

Figure 6. *Cont.*



(**e**)

Figure 6. Sensitivity analysis of each convolutional layer of the HGD model based on geometric median pruning. (**a**) PSNR value between the original image and the purified image of each convolutional layer pruning 0–90%, (**b**) SSIM value between the original image and the purified image of each convolutional layer pruning 0–90%, (**c**) PSNR value between the stego image and the purified image of each convolutional layer pruning 0–90%, (**d**) SSIM value between the stego image and the purified image of each convolutional layer pruning 0–90%, (**e**) DIQA.

5.2.2. Analysis of the Overall, Iterative Pruning

Geometric median pruning is performed on the DnCNN and HGD models under the condition of $DIQA < 0.2$. Each convolutional layer of the model prunes redundant filters to the maximum extent under limited conditions and achieves significant results.

Geometric median pruning of the DnCNN model demonstrates that params of the model reduce from the initial 0.558 M to 0.073 M, flops reduce from 36,591 G to 4775 G, DIQA changes from 0.068 to 0.092, and the image quality is nearly the same as the result of the original model to ensure the performance of the DnCNN deep-learning steganography removal model and compress the model (Table 3).

Table 3. Geometric median pruning of each convolutional layer in the DnCNN model when the document image-quality assessment (DIQA) valve is equal to 0.2.

Prune	Orig_Remove_ SSIM	Orig_Remove_ PSNR	Stego_Remove_ SSIM	Stego_Remove_ PSNR	DIQA	Params (M)	Flops (G)
oralModel	0.961	29,323	0.966	33,696	0.068	0.558	36,591
conv1_rm_44channels (70%)	0.961	29,323	0.966	33,695	0.068	0.532	34,852
conv2_rm_38channels (60%)	0.961	29,323	0.966	33,695	0.068	0.503	32,965
conv3_rm_50channels (80%)	0.961	29,323	0.966	33,695	0.068	0.462	30,304
conv4_rm_50channels (80%)	0.961	29,323	0.966	33,695	0.068	0.427	27,997
conv5_rm_38channels (60%)	0.961	29,323	0.966	33,695	0.068	0.400	26,244
conv6_rm_50channels (80%)	0.961	29,323	0.966	33,695	0.068	0.360	23,583
conv7_rm_50channels (80%)	0.961	29,323	0.966	33,695	0.068	0.325	21,276
conv8_rm_50channels (80%)	0.961	29,323	0.966	33,695	0.068	0.289	18,969
conv9_rm_44channels (70%)	0.961	29,323	0.966	33,695	0.068	0.258	16,939
conv10_rm_31channels (50%)	0.961	29,323	0.966	33,695	0.068	0.235	15,399
conv11_rm_31channels (50%)	0.961	29,323	0.966	33,695	0.068	0.208	13,622
conv12_rm_38channels (60%)	0.956	28,811	0.962	33,018	0.083	0.175	11,443
conv13_rm_38channels (60%)	0.954	28,560	0.959	32,554	0.091	0.144	9420
conv14_rm_31channels (50%)	0.954	28,548	0.957	32,497	0.092	0.119	7771
conv15_rm_31channels (50%)	0.954	28,548	0.957	32,497	0.092	0.091	5993
conv16_rm_57channels (90%)	0.954	28,548	0.957	32,497	0.092	0.073	4775

Geometric median pruning of the HGD model, shows that params of the model reduce from the initial 11,034 M to 0.721 M, flops reduce from 50,937 G to 5731 G, DIQA changes from 0.069 to 0.160 and SSIM and PSNR of stego and purified images are better than the original model. Hence, geometric median pruning has a significant effect on the HGD model of deep-learning steganography removal (Table 4).

Table 4. Geometric median pruning of each convolutional layer in the HGD model when the DIQA valve is equal to 0.2.

Prune	Orig_Remove_ SSIM	Orig_Remove_ PSNR	Stego_Remove_ SSIM	Stego_Remove_ PSNR	DIQA	Params (M)	Flops (G)
oralModel	0.978	31,618	0.963	30,868	0.069	11,034	50,937
conv1_rm_38channels (60%)	0.978	31,618	0.963	30,868	0.069	11,011	49,430
conv2_rm_19channels (30%)	0.963	29,368	0.966	31,257	0.090	10,974	48,060
conv3_rm_89channels (70%)	0.963	29,368	0.966	31,257	0.090	10,835	45,787
conv4_rm_89channels (70%)	0.963	29,368	0.966	31,257	0.090	10,701	43,593
conv5_rm_102channels (80%)	0.958	28,823	0.978	35,270	0.159	10,312	40,115
conv6_rm_127channels (50%)	0.958	28,823	0.978	35,270	0.159	9990	38,794
conv7_rm_178channels (70%)	0.952	28,457	0.988	34,200	0.145	9373	36,266
conv8_rm_204channels (80%)	0.952	28,457	0.988	34,200	0.145	8289	33,271
conv9_rm_230channels (90%)	0.952	28,654	0.987	34,693	0.175	7651	32,618
conv10_rm_230channels (90%)	0.952	28,495	0.988	35,965	0.132	7067	32,020
conv11_rm_230channels (90%)	0.949	27,910	0.988	35,346	0.172	5953	31,286
conv12_rm_230channels (90%)	0.943	26,070	0.988	30,255	0.128	5369	31,136
conv13_rm_230channels (90%)	0.950	28,292	0.991	37,972	0.144	4784	30,987
conv14_rm_230channels (90%)	0.950	28,076	0.987	36,073	0.184	4200	30,427
conv15_rm_230channels (90%)	0.951	28,334	0.987	36,249	0.137	3562	29,774
conv16_rm_230channels (90%)	0.949	28,031	0.990	36,867	0.155	2978	29,176
conv17_rm_102channels (40%)	0.949	28,041	0.990	36,803	0.154	2719	28,184
conv18_rm_153channels (60%)	0.949	28,041	0.990	36,803	0.154	2082	25,577
conv19_rm_178channels (70%)	0.949	28,041	0.990	36,803	0.154	1507	23,220
conv20_rm_153channels (60%)	0.949	28,041	0.990	36,803	0.154	1223	19,863
conv21_rm_89channels (70%)	0.949	28,041	0.990	36,803	0.154	1017	16,488
conv22_rm_102channels (80%)	0.949	28,041	0.990	36,803	0.154	0.863	13,973
conv23_rm_102channels (80%)	0.949	28,041	0.990	36,803	0.154	0.781	9654
conv24_rm_38channels (60%)	0.948	27,994	0.990	36,519	0.160	0.734	6623
conv25_rm_57channels (90%)	0.948	27,994	0.990	36,519	0.160	0.721	5731

DnCNN and HGD deep-learning steganography removal models have achieved significant results using geometric median pruning. Figures 7 and 8 illustrate that most of the purified images, cover images and stego images are basically indistinguishable to the naked eye after pruning. The color of a few purified images may change during the pruning process. However, in the process of secret message communication, the receiver may not have seen the stego image. Hence, the degree of image color change is still acceptable after the steganography removal. The removal effect of the document image's secret information after pruning based on the DIQA threshold is still within the range acceptable to the naked eye. Therefore, pruning of the deep-learning steganography removal model is reliable and effective to a certain extent.

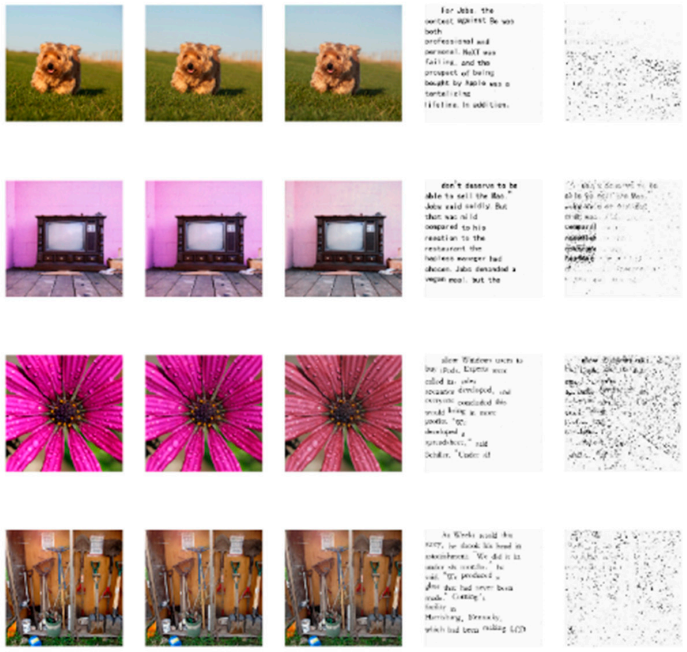


Figure 7. Renderings after the geometric median pruning in the DnCNN model when DIQA is equal to 0.2. First column of figure represents the cover image, the second column represents the stego image, the third column represents the purified image, the fourth column represents the document image secret message, and the fifth column represents the document image after removing.

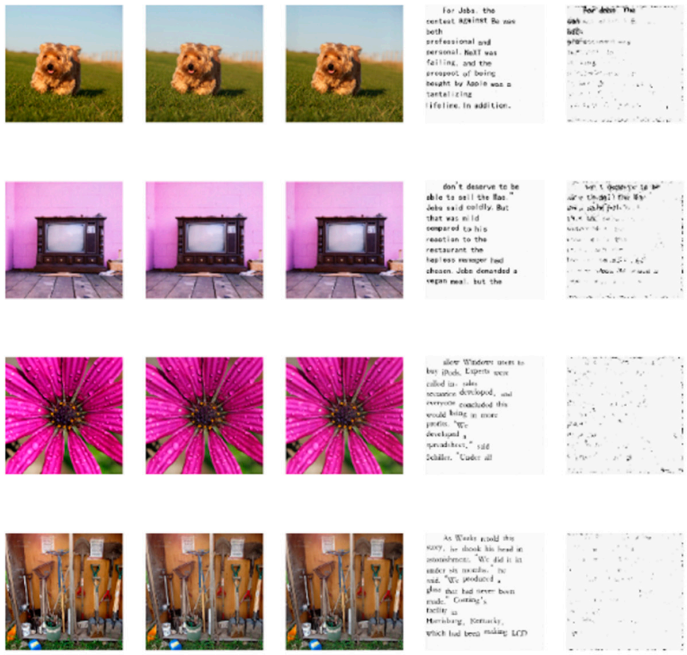
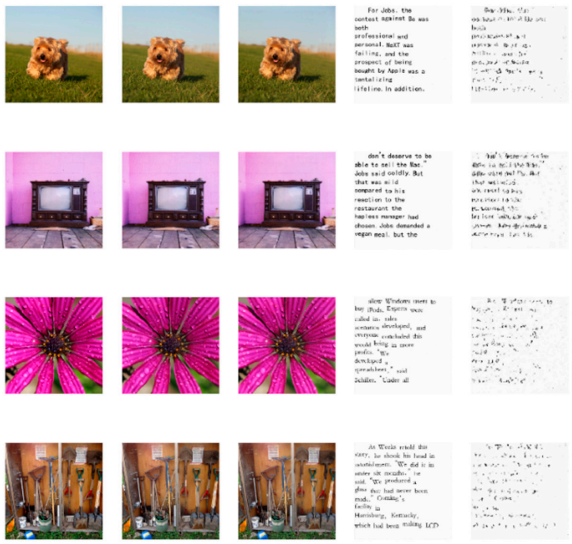


Figure 8. Renderings after the geometric median pruning in the HGD model when DIQA is equal to 0.2. First column of figure represents the cover image, the second column represents the stego image, the third column represents the purified image, the fourth column represents the document image secret message, and the fifth column represents the document image after removing.

5.2.3. Analysis of the Overall Iterative Pruning Threshold

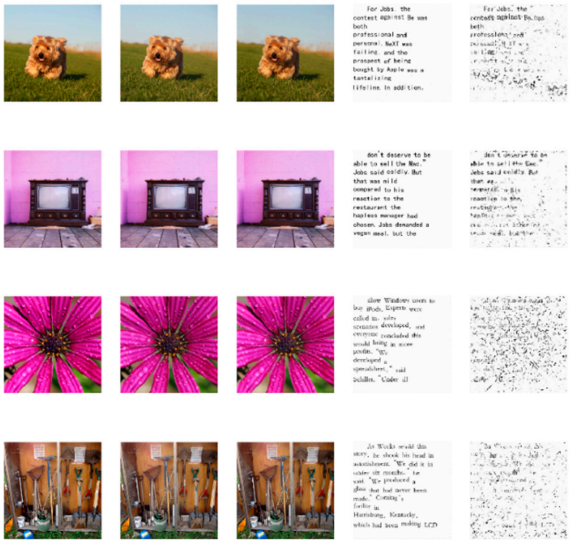
Deep steganography removal in the DnCNN model, when DIQA < 0.2, DIQA < 0.4 and DIQA < 0.6, the filter pruning by each convolutional layer are the same, as shown in Table 5.

Deep steganography removal in the HGD model demonstrates significant differences in pruning at each DIQA threshold. As shown in Table 6, under different DIQA value conditions, the model compression rate, and the performance of the model are different. The problem of compression degradation will occur when the DIQA threshold is set excessively large. Therefore, HGD has the optimal compression effect when the DIQA threshold is set to 0.4. The renderings are illustrated in Figures 8 and 9.



Geometric Median Pruning of HGD Model with DIQA<0.4

(**a**)



Geometric Median Pruning of HGD Model with DIQA<0.6

(**b**)

Figure 9. Renderings after the geometric median pruning in the HGD model under each DIQA threshold. (**a**) Model renderings when DIQA < 0.4; (**b**) model renderings when DIQA < 0.6.

Table 5. Pruning situation analysis in the DnCNN model under each DIQA threshold.

Model	Orig_Remove_ SSIM	Orig_Remove_ PSNR	Stego_Remove_ SSIM	Stego_Remove_ PSNR	DIQA	Params (M)	Flops (G)
oralModel	0.961	29,323	0.966	33,696	0.068	0.558	36,591
DIQA < 0.2	0.954	28,548	0.957	32,497	0.092	0.073	4775
DIQA < 0.4	0.954	28,548	0.957	32,497	0.092	0.073	4775
DIQA < 0.6	0.954	28,548	0.957	32,497	0.092	0.073	4775

Table 6. Pruning situation analysis in the HGD model under each DIQA threshold.

Model	Orig_Remove_ SSIM	Orig_Remove_ PSNR	Stego_Remove_ SSIM	Stego_Remove_ PSNR	DIQA	Params (M)	Flops (G)
oralModel	0.978	31,618	0.963	30,868	0.069	11,034	50,937
DIQA < 0.2	0.948	27,994	0.990	36,519	0.160	0.721	5731
DIQA < 0.4	0.953	27,925	0.981	33,609	0.321	0.603	4349
DIQA < 0.6	0.946	27,296	0.990	33,878	0.275	0.758	5642

5.2.4. Analysis of the ABC Automatic Pruning

On the basis of the geometric median, the ABC automatic pruning algorithm combines the convolutional layers that the last layer of the deep-learning steganography removal model does not participate in the arrangement under the condition of DIQA and the maximum channel pruning φ of each convolutional layer of the model. $\varphi = 9$ represents the maximum channel pruning of each convolutional layer of the model is 90% and $\varphi = 6$ represents the maximum channel pruning of each convolutional layer of the model is 60%. Each arrangement is a compressed structure of the model. According to the process of nectar source collection and role transition of three types of bees, the optimal pruning structure that meets the conditions is finally found under t = 150 rounds, as shown in Tables 7 and 8. Compared with the overall iterative pruning, the ABC pruning has realized an automated process.

Table 7. Optimal nectar source of ABC automatic pruning under the condition of DIQA threshold and maximum channel pruning φ of the convolutional layer of DnCNN model.

Threshold	φ	Nectar Source
DIQA < 0.2	9	[7, 6, 7, 8, 5, 8, 7, 8, 7, 5, 4, 4, 4, 4, 3]
	6	[4, 6, 6, 5, 5, 5, 5, 4, 6, 5, 5, 5, 5, 4, 5, 4]
DIQA < 0.4	9	[7, 6, 8, 8, 6, 8, 7, 7, 7, 3, 5, 6, 3, 5, 5, 2]
	6	[6, 6, 6, 6, 6, 4, 6, 5, 4, 5, 5, 6, 4, 4, 5, 2]
DIQA < 0.6	9	[7, 6, 7, 7, 5, 6, 3, 8, 5, 4, 3, 6, 6, 4, 4, 3]
	6	[3, 5, 5, 6, 6, 6, 6, 6, 6, 4, 4, 6, 6, 4, 5, 4]

Table 8. Optimal nectar source of ABC automatic pruning under the condition of DIQA threshold and maximum channel pruning φ of the convolutional layer of HGD model.

Threshold	φ	Nectar Source
DIQA < 0.2	9	[5, 2, 3, 4, 7, 4, 9, 5, 9, 8, 7, 2, 5, 4, 4, 5, 3, 5, 7, 5, 7, 8, 6, 6, 1]
	6	[2, 2, 6, 5, 6, 6, 6, 5, 6, 6, 6, 6, 6, 3, 3, 6, 3, 2, 6, 3, 6, 6, 3, 5, 2]
DIQA < 0.4	9	[6, 2, 9, 3, 9, 4, 7, 4, 9, 5, 9, 7, 2, 2, 9, 7, 3, 8, 3, 3, 5, 8, 4, 5, 4]
	6	[4, 2, 6, 6, 5, 3, 5, 6, 6, 6, 6, 6, 5, 6, 6, 6, 3, 6, 6, 6, 6, 1, 5, 4, 4]
DIQA < 0.6	9	[3, 3, 7, 7, 6, 5, 9, 9, 9, 4, 3, 9, 4, 7, 9, 8, 3, 4, 4, 6, 3, 7, 5, 7, 2]
	6	[3, 4, 6, 4, 6, 6, 6, 4, 2, 4, 5, 4, 6, 2, 4, 6, 3, 6, 4, 4, 6, 4, 3, 6, 2]

Tables 9 and 10 are the conditions of the best searched nectar source quality under t = 150 rounds. From table, it can be seen that the ABC algorithm on the basis of geometric median pruning can

efficiently search for nectar source under the limited nectar source fitness, and finally find a lightweight deep-learning steganography removal model with robustness and adaptability. The amount of model params and flops can be greatly compressed compared with the original model under the DIQA threshold and the maximum channel pruning φ of the convolutional layer of the model, and the image quality and document image removal are guaranteed to be within the acceptable range.

Table 9. Analysis of ABC pruning under the conditions of DIQA threshold and maximum channel pruning φ of the convolutional layer of the DnCNN model.

Threshold	φ	Orig_Remove_ SSIM	Orig_Remove_ PSNR	Stego_Remove_ SSIM	Stego_Remove_ PSNR	DIQA	Params (M)	Flops (G)
	oralModel φ = 9	0.961	29,323	0.966	33,696	0.068	0.558	36,591
DIQA < 0.2	9	0.944	27,499	0.956	31,491	0.061	0.116	7626
	6	0.960	29,291	0.965	33,586	0.066	0.136	8900
DIQA < 0.4	9	0.955	28,797	0.960	32,940	0.084	0.104	6807
	6	0.956	28,788	0.962	32,930	0.077	0.141	9264
DIQA < 0.6	9	0.938	26,901	0.950	30,541	0.131	0.131	8565
	6	0.954	28,560	0.959	32,556	0.091	0.133	8709

Table 10. Analysis of ABC pruning under the conditions of DIQA threshold and maximum channel pruning φ of the convolutional layer of the HGD model.

Threshold	φ	Orig_Remove_ SSIM	Orig_Remove_ PSNR	Stego_Remove_ SSIM	Stego_Remove_ PSNR	DIQA	Params (M)	Flops (G)
	oralModel φ = 9	0.978	31,618	0.963	30,868	0.069	11,034	50,937
DIQA < 0.2	9	0.944	28,458	0.942	30,541	0.061	2261	10,723
	6	0.957	28,420	0.995	35,815	0.105	2993	15,843
DIQA < 0.4	9	0.955	27,726	0.985	34,092	0.385	1832	10,529
	6	0.978	31,617	0.964	30,886	0.071	2312	13,967
DIQA < 0.6	9	0.949	28,044	0.985	34,651	0.398	1556	10,451
	6	0.953	28,291	0.993	38,681	0.582	3329	14,303

5.2.5. Analysis of Pruning Rate Based on the Geometric Median Pruning

The original DnCNN model has 1030 channels, 0.558 M params and 36,591 G flops. The overall iterative pruning based on geometric median is a violent pruning process, which can prune the channels to the maximum extent. Under the conditions of DIQA threshold and maximum channel pruning φ of the convolutional layer, about 65% of the channels are pruned, the classic indicators params and flops are reduced by more than 85%. ABC pruning based on geometric median is an automatic pruning process, which can automatically search for the required channels. Under the conditions of DIQA threshold and maximum channel pruning φ of the convolutional layer, about 50% of the channels are pruned, the params and flops are reduced about 75% (Table 11).

The original HGD model has 4870 channels, 11,034 M params and 50,937 G flops. The overall iterative pruning based on geometric median results in about 75% channel pruning, about 93% params reducing and 88% flops reducing under the conditions of DIQA threshold and maximum channel pruning φ of the convolutional layer. ABC pruning based on geometric median results in more than 44% channel pruning, more than 69% params reducing and more than 68% flops reducing under the conditions of DIQA threshold and maximum channel pruning φ of the convolutional layer (Table 12).

Table 11. Analysis of pruning rate under the conditions of DIQA threshold and maximum channel pruning φ of the convolutional layer of the DnCNN model based on geometric median.

Model	Threshold	φ	Channels	Pruned	Params (M)	Pruned	Flops (G)	Pruned
oralModel		9	1030	0%	0.558	0%	36,591	0%
overall iterative pruning	DIQA < 0.2	9	359	65.14%	0.073	86.92%	4775	86.95%
	DIQA < 0.4	9	359	65.14%	0.073	86.92%	4775	86.95%
	DIQA < 0.6	9	359	65.14%	0.073	86.92%	4775	86.95%
ABC pruning	DIQA < 0.2	9	455	55.83%	0.116	79.21%	7626	79.16%
		6	515	50.00%	0.136	75.63%	8900	75.68%
	DIQA < 0.4	9	441	57.18%	0.104	81.36%	6807	81.40%
		6	524	49.13%	0.141	74.73%	9264	74.68%
	DIQA < 0.6	9	499	51.55%	0.131	76.52%	8565	76.59%
		6	511	50.39%	0.133	76.16%	8709	76.20%

Table 12. Analysis of pruning rate under the conditions of DIQA threshold and maximum channel pruning φ of the convolutional layer of the HGD model based on geometric median.

Model	Threshold	φ	Channels	Pruned	Params (M)	Pruned	Flops (G)	Pruned
oralModel		9	4870	0%	11,034	0%	50,937	0%
overall iterative pruning	DIQA < 0.2	9	1210	75.15%	0.721	93.47%	5731	88.75%
	DIQA < 0.4	9	1134	76.71%	0.603	94.54%	4349	91.46%
	DIQA < 0.6	9	1263	74.07%	0.758	93.13%	5642	88.92%
ABC pruning	DIQA < 0.2	9	2242	53.96%	2261	79.51%	10,723	78.95%
		6	2536	47.93%	2993	72.87%	15,843	68.90%
	DIQA < 0.4	9	2185	55.13%	1832	83.40%	10,529	79.33%
		6	2324	52.28%	2312	79.05%	13,967	72.58%
	DIQA < 0.6	9	1906	60.86%	1556	85.90%	10,451	79.48%
		6	2725	44.05%	3329	69.83%	14,303	71.92%

6. Conclusions

A considerable amount of resources are consumed when pre-trained DnCNN and HGD deep-learning steganography removal models are used to deploy network steganography firewalls. A geometric median pruning method that can be used for multiparameter, large-scale and pre-trained deep-learning steganography removal models is proposed in this study to search for effective network structures, prune redundant filters, and thus compress the deep-learning steganography removal model.

The pruning method significantly reduces params and flops of the model while ensuring the robustness of the deep-learning steganography removal model. The quality of purified image is still within an acceptable range, and the document image as secret message achieves a certain removal effect. However, we only explore the lightweight deep-learning steganography removal model from the perspective of pruning, so our future work will focus on the following aspects: (1) Explore a fast adaptive structure adjustment algorithm (2) Explore the feasibility of deep-learning steganography removal based on knowledge distillation [47] and quantification (3) We plan to use dimensionality reduction technology [48] to document images to improve model performance.

Author Contributions: Conceptualization, S.Z., W.W. and K.C.; methodology, S.Z. and W.W.; validation, W.W.; formal analysis, W.W.; investigation, S.Z., K.C. and J.L.; data curation, W.W.; writing—original draft preparation, W.W.; writing—review and editing, S.Z., W.W., K.C. and J.L.; visualization, W.W.; supervision, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC), Grant Number 61972187; the Scientific Research Project of Science and Education Park Development Center of Fuzhou University, Jinjiang, Grant Number 2019-JJFDKY-53 and the Tianjin University-Fuzhou University Joint Fund, Grant Number TF2020-6.

Acknowledgments: Thanks to the editors and reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Neural Information Processing Systems 25 (NIPS)*; Curran Associates Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
2. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
3. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; 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; IEEE: New York, NY, USA, 2015; pp. 1–9. [CrossRef]
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26–30 June 2016; IEEE: New York, NY, USA, 2016; pp. 770–778. [CrossRef]
5. Baluja, S. Hiding Images in Plain Sight: Deep Steganography. In *Neural Information Processing Systems (NIPS)*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 2069–2079.
6. Wu, P.; Yang, Y.; Li, X. StegNet: Mega Image Steganography Capacity with Deep Convolutional Network. *Future Internet* **2018**, *10*, 54. [CrossRef]
7. Zhang, R.; Dong, S.; Liu, J. Invisible Steganography via Generative Adversarial Networks. *Multimed. Tools Appl.* **2018**. [CrossRef]
8. Xu, G.; Liu, B.; Jiao, L.; Li, X.; Feng, M.; Liang, K.; Ma, L.; Zheng, X. Trust2Privacy: A Novel Fuzzy Trust-to-Privacy Mechanism for Mobile Social Networks. *IEEE Wirel. Commun.* **2020**, *27*, 72–78. [CrossRef]
9. Horé, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In Proceedings of the 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 23–26 August 2010; IEEE: New York, NY, USA, 2010; pp. 2366–2369. [CrossRef]
10. Kang, L.; Ye, P.; Li, Y.; Doermann, D. A deep learning approach to document image quality assessment. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; IEEE: New York, NY, USA; pp. 2570–2574. [CrossRef]
11. Johnson, N.F.; Jajodia, S. Steganalysis of Images Created Using Current Steganography Software. In *International Workshop on Information Hiding*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 273–289. [CrossRef]
12. Jung, D.; Bae, H.; Choi, H.S.; Yoon, S. PixelSteganalysis: Pixel-wise Hidden Information Removal with Low Visual Degradation. *arXiv* **2019**, arXiv:1902.10905.
13. Oord, A.V.D.; Kalchbrenner, N.; Vinyals, O.; Espeholt, L.; Graves, A.; Kavukcuoglu, K. Conditional Image Generation with PixelCNN Decoders. In Proceedings of the 30th International Conference on Neural Information Processing, Barcelona, Spain, 5–10 December 2016.
14. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. ntriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
15. Jia, X.; Wei, X.; Cao, X.; Foroosh, H. ComDefend: An Efficient Image Compression Model to Defend Adversarial Examples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; IEEE: New York, NY, USA, 2019; pp. 6084–6092.
16. Lei, B.J.; Caurana, R. Do Deep Nets Really Need to be Deep? In Proceedings of the Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–11 December 2014; Curran Associates Inc.: Red Hook, NY, USA, 2014; pp. 2654–2662.
17. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
18. Cun, Y.L.; Denker, J.S.; Solla, S.A. Optimal brain damage. In *Neural Information Processing Systems (NIPS)*; Curran Associates Inc.: Red Hook, NY, USA, 1990; pp. 598–605.

19. Liu, Z.; Sun, M.; Zhou, T.; Huang, G.; Darrell, T. Rethinking the Value of Network Pruning. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
20. Han, S.; Pool, J.; Tran, J.; Dally, W.J. Learning both Weights and Connections for Efficient Neural Networks. In *Neural Information Processing Systems (NIPS)*; Curran Associates Inc.: Red Hook, NY, USA, 2015; pp. 1135–1143.
21. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
22. Fridrich, J.; Goljan, M.; Hogea, D. Steganalysis of JPEG Images: Breaking the F5 Algorithm. In *5th International Workshop on Information Hiding*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 310–323. [CrossRef]
23. Gou, H.; Swaminathan, A.; Wu, M. Noise Features for Image Tampering Detection and Steganalysis. *IEEE Int. Conf. Image Process.* **2007**, *6*, VI-97–VI-100. [CrossRef]
24. Amritha, P.P.; Sethumadhavan, M.; Krishnan, R. On the Removal of Steganographic Content from Images. *Def. Ence J.* **2016**, *66*, 574–581. [CrossRef]
25. Corley, I.; Lwowski, J.; Hoffman, J. Destruction of Image Steganography using Generative Adversarial Networks. *arXiv* **2019**, arXiv:1912.10070.
26. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef] [PubMed]
27. Song, Y.; Kim, T.; Nowozin, S.; Ermon, S.; Kushman, N. PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. *arXiv* **2017**, arXiv:1710.10766.
28. Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Hu, X.; Zhu, J. Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser. *arXiv* **2017**, arXiv:1712.02976.
29. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning Filters for Efficient ConvNets. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
30. He, Y.; Kang, G.; Dong, X.; Fu, Y.; Yang, Y. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018. [CrossRef]
31. He, Y.; Liu, P.; Wang, Z.; Hu, Z.; Yang, Y. Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019. [CrossRef]
32. Chen, J.; Zhu, Z.; Li, C. Self-Adaptive Network Pruning. *arXiv* **2019**, arXiv:1910.08906v1.
33. Madaan, D.; Shin, J.; Hwang, S.J. Adversarial Neural Pruning with Latent Vulnerability Suppression. *arXiv* **2020**, arXiv:1908.04355.
34. Luo, J.-H.; Wu, J.; Lin, W. ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
35. He, Y.; Zhang, X.; Sun, J. Channel Pruning for Accelerating Very Deep Neural Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
36. Yu, R.; Li, A.; Chen, C.-F.; Lai, J.-H.; Morariu, V.I.; Han, X.; Gao, M.; Lin, C.-Y.; Davis, L.S. NISP: Pruning Networks using Neuron Importance Score Propagation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 9194–9203.
37. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning Efficient Convolutional Networks through Network Slimming. In *IEEE International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2017; pp. 2755–2763. [CrossRef]
38. Huang, Z.; Wang, N. Data-Driven Sparse Structure Selection for Deep Neural Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
39. Lin, S.; Ji, R.; Yan, C.; Zhang, B.; Cao, L.; Ye, Q.; Huang, F.; Doermann, D. Towards Optimal Structured CNN Pruning via Generative Adversarial Learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019. [CrossRef]
40. Zhao, K.; Zhang, X.-Y.; Han, Q.; Cheng, M.-M. Dependency Aware Filter Pruning. *arXiv* **2020**, arXiv:2005.02634.
41. Guerra, L.; Zhuang, B.; Reid, I.; Drummond, T. Automatic Pruning for Quantized Neural Network. *arXiv* **2020**, arXiv:2002.00523.

42. Dong, X.; Yang, Y. Network Pruning via Transformable Architecture Search. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2019; pp. 760–771.
43. Liu, N.; Ma, X.; Xu, Z.; Wang, Y.; Ye, J. AutoCompress: An Automatic DNN Structured Pruning Framework for Ultra-High Compression Rates. *AAAI* **2020**. [CrossRef]
44. Lin, M.; Ji, R.; Zhang, Y.; Zhang, B.; Wu, Y.; Tian, Y. Channel Pruning via Automatic Structure Search. *IJCAI* **2020**. [CrossRef]
45. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the International Conference on Machine Learning, Helsinki, Finland, 5–9 June 2008. [CrossRef]
46. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015. [CrossRef]
47. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge Distillation: A Survey. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, DC, USA, 16–18 June 2020.
48. Zhou, G.; Xu, G.; Hao, J.; Chen, S.; Xu, J.; Zheng, X. Generalized Centered 2-D Principal Component Analysis. *IEEE Trans. Cybern.* **2019**, 1–12. [CrossRef] [PubMed]