symmetry

MDPI

Article

DIGDH: A Novel Framework of Difference Image Grafting Deep Hiding for Image Data Hiding

Xintao Duan [1,2,*], Lei Li [1], Yao Su [1], Wenxin Wang [1], En Zhang [1,2] and Xianfang Wang [3]

[1] College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China; LL1437715969@163.com (L.L.); 18623740132@163.com (Y.S.); wangwenxin3023@126.com (W.W.); zhangen@htu.edu.cn (E.Z.)
[2] Key Laboratory of Artificial Intelligence and Personalized Learning in Education of Henan Province, Xinxiang 453007, China
[3] College of Computer Science and Technology, Henan Institute of Technology, Xinxiang 453003, China; wxf@hait.edu.cn
* Correspondence: duanxintao@htu.edu.cn

Abstract: Data hiding is the technique of embedding data into video or audio media. With the development of deep neural networks (DNN), the quality of images generated by novel data hiding methods based on DNN is getting better. However, there is still room for the similarity between the original images and the images generated by the DNN models which were trained based on the existing hiding frameworks to improve, and it is hard for the receiver to distinguish whether the container image is from the real sender. We propose a framework by introducing a key_img for using the over-fitting characteristic of DNN and combined with difference image grafting symmetrically, named difference image grafting deep hiding (DIGDH). The key_img can be used to identify whether the container image is from the real sender easily. The experimental results show that without changing the structures of networks, the models trained based on the proposed framework can generate images with higher similarity to original cover and secret images. According to the analysis results of the steganalysis tool named StegExpose, the container images generated by the hiding model trained based on the proposed framework is closer to the random distribution.

Keywords: DIGDH; image hiding; framework of image hiding; high capacity data hiding

Citation: Duan, X.; Li, L.; Su,Y.; Wang, W.; Zhang, E.; Wang, X. DIGDH: A Novel Framework of Difference Image Grafting Deep Hiding for Image Data Hiding. *Symmetry* **2022**, *14*, 151. https://doi.org/10.3390/sym14010151

Academic Editors: Fengyong Li and Chuan Qin

Received: 8 December 2021
Accepted: 5 January 2022
Published: 13 January 2022

1. Introduction

With the progress of machine learning and the rapid development of DNN, great achievements have been made in many fields, while it has also shown good performance in the field of data hiding [1–4]. Image hiding is one category of data hiding [5]. Data hiding based on deep convolutional neural networks (DCNN) has developed rapidly in the past few years. The hiding capacity of image hiding based on DCNN—such as hiding data with deep networks (HiDDeN) [6] and end-to-end image steganography [7]—can be several times greater or more than traditional image hiding algorithms such as highly undetectable stego (Hugo) [8] and wavelet obtained weights (WOW) [9]. The three main characteristics of data hiding are the capacity of hiding, security and robustness, but the most basic requirement of data hiding is to try not to modify the cover. According to the different forms of secret information, image hiding can be divided into hiding an image message into another image and hiding a binary message into an image [10]. According to the different emphasis of image hiding, image hiding can be divided into high capacity data hiding, secure steganography and robust image watermarking [4]. High capacity image hiding is characterized by a large amount of secret information embedded in the cover image which means the sender can hide one or more secret images into one cover image to generate the container image and transmit it to the receiver through the public channel, and the receiver can obtain the revealed secret image from the container image. Secure

steganography is more inclined to anti steganalysis than high capacity image hiding. Robust image watermarking requires the ability of anti modification [4,11]. For the convenience of description, we call the original cover images and the original secret images as the original images, the container images and the revealed secret images which generated by the network models as the generated images in this article.

Until the universal deep hiding (UDH) framework has been proposed by Zhang et al. in [10], researchers improve the effects of image hiding and revelation mainly by introducing new neural network models or improving the existing neural network models on the basis of cover-dependent deep hiding (DDH). The cover image and the secret image concatenated by a separable convolution with residual block (SCR) was proposed by Wu et al. in [12]. A new cost function to reduce the influence of noise in the generated container image called the variance loss is proposed by Wu et al. in [13]. A pyramid pooling module [14] based encoder–decoder architecture is used for hiding and the structure of the reveal network that is similar to its hiding network named StegoPNet was proposed by Duan et al. in [15]. Three networks, namely pre-processing (P) network, hiding network and reveal network were proposed by Baluja et al. in [16].

Figure 1 shows the DDH consisting of DDH without the pre-processing (P) network [12] and DDH with the pre-processing (P) network [17].

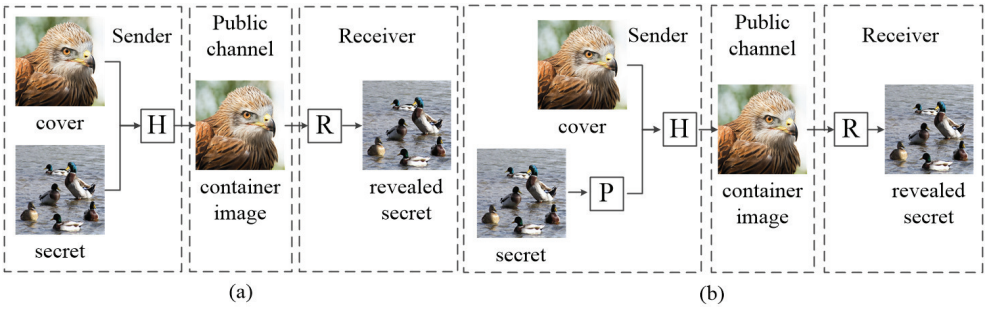

Figure 1. Existing framework without (**a**) or with (**b**) P network of DDH.

In Figure 2, the sender could input one secret image into the hiding network H and add the output of H to any one of the cover images to obtain the corresponding container image of each cover image. As described in [10], the dependence on the cover images of models trained based on UDH is not high. The container images are transmitted from the sender to the receiver through the public channel, and the receiver can obtain the revealed secret image by using the reveal network R.

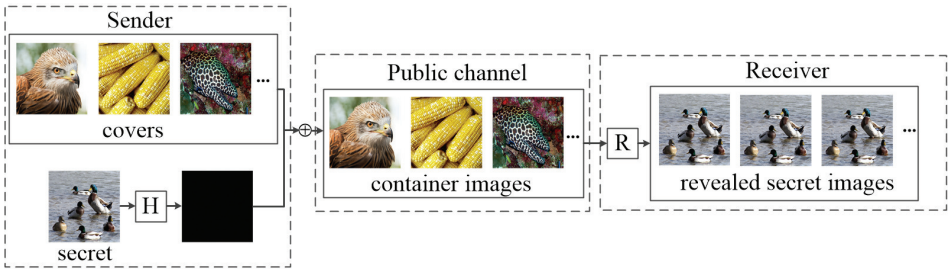

Figure 2. Universal deep hiding (UDH) framework proposed in [10].

What has been verified by us in the experiments is that the over-fitting feature of DCNN can improve the effects of image hiding, but the selection of cover cannot be very flexible, while difference image grafting symmetrically can solve the problem of excessive cover dependence exactly. We propose a framework with one key_img which can generate images that are more similar to the original images in this article, this framework is named difference image grafting deep hiding.

The main contributions of our works in this article are summarized as follows:

1. We propose a novel image hiding framework with the characteristic of symmetry in which the receiver can use the right key_img and R to determine whether the container image is generated by the real sender.
2. We combine the over-fitting characteristic of DNN and the method of difference image grafting symmetrically to generate quality images that with higher similarity to the original images.
3. The ROC curves drawn according to the analysis results of one kind of batch lossless images steganalysis tool which named StegExpose [18] show that the images generated by our proposed framework is closer to the random distribution.

The rest of this article is organized as follows: Section 2 is about the related work; Section 3 is the description of the proposed framework; Section 4 is the experimental results and analysis; Section 5 is about the conclusion of this article.

2. Related Work

According to the different forms of secret information, image hiding can be divided into hiding a binary message into an image and hiding an image message into another image. No matter what information needs to be hidden, it needs corresponding theories to support.

2.1. Hiding a Binary Message into an Image

As early as 2011, DNN based steganography and watermarking were proposed by ISAC et al. in [19]. There is a method of hiding binary messages into images in an end-to-end manner against DNN is proposed by Hayes et al. [20]. However, Zhu et al. proposed a method to hide the binary message as a watermarking in the image and use adversarial training to minimize the artificial effect on the cover in HiDDeN [6]. A kind of the hyperlink is encoded into binary bits and hidden in the cover image and DNN models can be trained to perform robust encoding and decoding of physical photos was proposed by Tancik et al. [21]. While the focus of these methods is on the trade-off among capacity, security and robustness.

2.2. Hiding an Image into Another Image

A kind of separable convolution with residual block named SCR which can be used to concatenate the cover image and the secret image, the concatenated image is given as the input to the encoder for generating the container image which is fed to the decoder to output the decoded secret image was proposed by Wu et al. [12]. Later, Wu et al. proposed a new cost function in order to reduce the effect of noise in the generated container image which named the variance loss [13]. The framework with three networks of hiding an image into another is proposed by Baluja [16,17] which is named DDH with P as we can see as Figure 1b.

2.3. Model Formulation

The theoretical basis of two mainstream frameworks will be described in detail in this section, because of DDH with P is very similar to that without P, here is only the description of DDH without P. In order to facilitate the description, the following simplified symbols are used in the following content if without a special instruction.

c represents cover image
s represents secret image
c' represents container image
s' represents revealed secret image
H represents Hiding network model
R represents Reveal network model
θ_H represents the parameters of Hiding network model
θ_R represents the parameters of Reveal network model

Image hiding based on DDH is combined with the cover image and secret image as the input of H and output the c', the formula of the process of hiding is:

$$c' = H(c, s; \theta_H) \tag{1}$$

The process of secret image revelation based on DDH is to input the c' into R and output the s', the formula of the process of secret revelation is:

$$\begin{aligned} s' &= R(c'; \theta_R) \\ &= R(H(c, s; \theta_H); \theta_R) \end{aligned} \tag{2}$$

The purpose of training H is to enhance the similarity between c and c'. In other words, the purpose of training H is to get better θ_H which can reduce the difference between c and c' as much as possible [4], the formula of process of training H is as follows:

$$\begin{aligned} \theta_H^* &= \arg\min_{\theta_H} \mathrm{dist}_c(c, c') \\ &= \arg\min_{\theta_H} \mathrm{dist}_c(c, H(c, s; \theta_H)) \end{aligned} \tag{3}$$

Similar to the purpose of training H, the purpose of training R is to obtain better θ_R which can reduce the difference between s and s' as much as possible [4].The formula of process of training R is as follows:

$$\begin{aligned} \theta_R^* &= \arg\min_{\theta_R} \mathrm{dist}_s(s, s') \\ &= \arg\min_{\theta_R} \mathrm{dist}_s(s, R(c'; \theta_R)) \\ &= \arg\min_{\theta_R} \mathrm{dist}_s(s, R(H(c, s; \theta_H); \theta_R)) \end{aligned} \tag{4}$$

The hiding process based on UDH is that the sender could take the secret image as the input of H directly, and add the output of H to a cover to get the container image, the formula of hiding is as follows:

$$c' = H(s; \theta_H) + c \tag{5}$$

According to the formula (5), it can be concluded that the purpose of the image hiding process based on UDH is to obtain better θ_H which can make the difference between c and c' smaller and smaller, the formula of training H is as follows:

$$\begin{aligned} \theta_H^* &= \arg\min_{\theta_H} \mathrm{dist}_c(c, c') \\ &= \arg\min_{\theta_H} \mathrm{dist}_c(c, H(s; \theta_H) + c) \end{aligned} \tag{6}$$

The process of secret revelation based on UDH is that the receiver could take c' as the input of R to obtain s', the formula of secret revelation is as follows:

$$\begin{aligned} s' &= R(c'; \theta_R) \\ &= R(H(s; \theta_H) + c; \theta_R) \end{aligned} \tag{7}$$

and the formula of training R based on UDH is as follows:

$$\begin{aligned} \theta_R^* &= \arg\min_{\theta_R} \mathrm{dist}_s(s, s') \\ &= \arg\min_{\theta_R} \mathrm{dist}_s(s, R(c'; \theta_R)) \\ &= \arg\min_{\theta_R} \mathrm{dist}_s(s, R(H(s; \theta_H) + c; \theta_R)) \end{aligned} \tag{8}$$

In [10], there is another method to reveal the secret image with better effect, that is, take the difference image between the container image and the cover image as the input of R, the formula of this process is as follows:

$$\begin{aligned} s' &= R((c' - c); \theta_R) \\ &= R(((H(s; \theta_H) + c) - c); \theta_R) \end{aligned} \tag{9}$$

the better results of revelation based on formula (9) are used to compare with the proposed.

The mean squared error (MSE) loss function is used to constraint the networks' training process and the backpropagation algorithm [22] is used to drive the networks to contin-

uously adjust the weights to complete the networks' training, image hiding consists of hiding and revelation. Specifically, the global loss function is expressed as follows:

$$\begin{aligned} l_{Sum} &= l_H + \alpha \cdot l_R \\ &= \frac{1}{n}\sum_{i=1}^{n}\|c' - c\|^2 + \alpha \cdot \frac{1}{n}\sum_{i=1}^{n}\|s' - s\|^2 \end{aligned} \tag{10}$$

where α is used to adjust the heft between H and R.

3. Proposed Framework

Among the existing methods of image hiding, whether it is based on traditional algorithms or DCNN, the purpose of hiding the secret is achieved by trying not to modify the cover image in order not to reduce the similarity between the original images and generated images, DIGDH is also committed to this. DIGDH with the characteristics of symmetry, which requires the sender and receiver to hold the same right key_img can obtain the normal container image and revealed secret image.

As is shown in Figure 3, a three-step framework of image hiding is proposed by us, and the specific steps are as follows:

1. The secret image and key_img are combined as the input of the hiding network and output a key_img with secret.
2. The key_img with secret obtained in the first step is used to minus the key_img to obtain a difference image and add it to the cover image to obtain the container image, the sender could transmit the container image to the receiver through the public channel.
3. The receiver can use the container image to minus the cover image to obtain the difference image between them and add it to the key_img as the input of R to obtain the revealed secret image.

In the process hiding and revelation, the acquisition process of difference and difference' with the characteristic of symmetry, and the grafting process of them also with the characteristic of symmetry.

The formula of the hiding process based on DIGDH is as follows:

$$c' = H(\text{ key }\_img, s; \theta_H) - key\_img + c \tag{11}$$

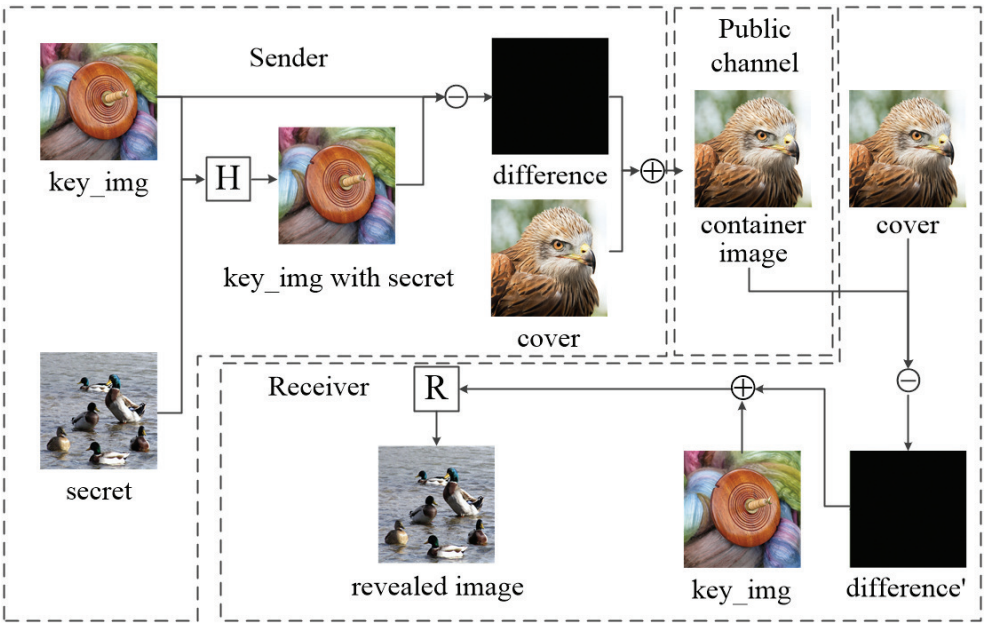

Figure 3. The proposed difference image grafting based deep hiding framework.

In order to endow the trained models with the ability to generate high quality images, key_img is used to train DNN models with over-fitting characteristic, and also as the background image to obtain difference images. At the same time, in order to reduce the difference between c and c' as much as possible, we need to get better θ_H based on the Formula (12) which is applied to train the hiding network model:

$$\begin{aligned} \theta_H^* &= \arg\min_{\theta_H} \text{dist}_c(c, c') \\ &= \arg\min_{\theta_H} \text{dist}_c(c, H(\text{ key }\_img, s; \theta_H) - key\_img + c) \end{aligned} \tag{12}$$

The formula of the process of revealing secret based on DIGDH is as follows:

$$\begin{aligned} s' &= R(c' - c + key\_img; \theta_R) \\ &= R((H(key\_img, s; \theta_H) - key\_img + c) - c + key\_img; \theta_R) \end{aligned} \tag{13}$$

correspondingly, we use the Formula (14) to train R based on DIGDH:

$$\begin{aligned} \theta_R^* &= \arg\min_{\theta_R} \text{dist}_s(s, s') \\ &= \arg\min_{\theta_R} \text{dist}_s(s, R(c' - c + key\_img; \theta_R)) \\ &= \arg\min_{\theta_R} \text{dist}_s(s, R((H(key\_img, s; \theta_H) - key\_img + c) - c + key\_img; \theta_R)) \end{aligned} \tag{14}$$

4. Experimental Results and Analysis

The dataset used in the experiments in this article is from the ImageNet [23], which contains 48,000 images are randomly selected as the training set, 6000 images are used as the verification set, and 6000 images are used as the test set, the images in the experiments are RGB images with the size of 256 × 256. The experiments are completed in the Ubuntu 18.4, the hiding and reveal network models are built based on pytorch 1.2.0 and the version of Python is 3.6.2. The Adm optimizer is used to optimize the models, at the same time, the initial learning rate of each network model is 0.001, and the batch_size of training is 40. The value of α appeared in the Formula (10) is set to 0.75, the constructed models have been deployed on two GeForce GTX 1080 8G Graphics Cards for training. We take the optimal check point as the final hiding model and reveal model. The hiding network model and reveal network model have been proposed in the StegoPNet [15] are used in the experiment as the H and R of DIGDH, the models based on DIGDH have been trained for 200 epochs, and the optimal check point is obtained at the 188th epoch.

4.1. Subjective Analysis

As mentioned in this article, only using the right key_img can the sender get the normal container image and the receiver get the normal revealed image, it can be found from the analysis and observation of Figure 4 according to Table 1.

What has to be mentioned here is that the right key is the key_img which has been used in the process of training the network models, on the contrary, the wrong key is an image is unrelated to the process of training the models but used as the key_img. Let us get started with Figure 4 which is made up of eight columns and nine rows, and the eight columns of Figure 4 corresponding to eight rows in Table 1. In Table 1, row A means to hide with the right key_img and reveal with the right key_img, row B means to hide with the right key_img but reveal with the wrong key_img, the remaining rows have the same rules with row A and B.

Table 1. The key_imgs reference table of Figure 4.

Group	Key of Hiding	Key of Revelation
A	right key	right key
B	right key	wrong key
C	wrong key	right key
D	wrong key	wrong key
E	wrong key	right key
F	wrong key	wrong key
G	white (wrong) key	right key
H	black (wrong) key	right key

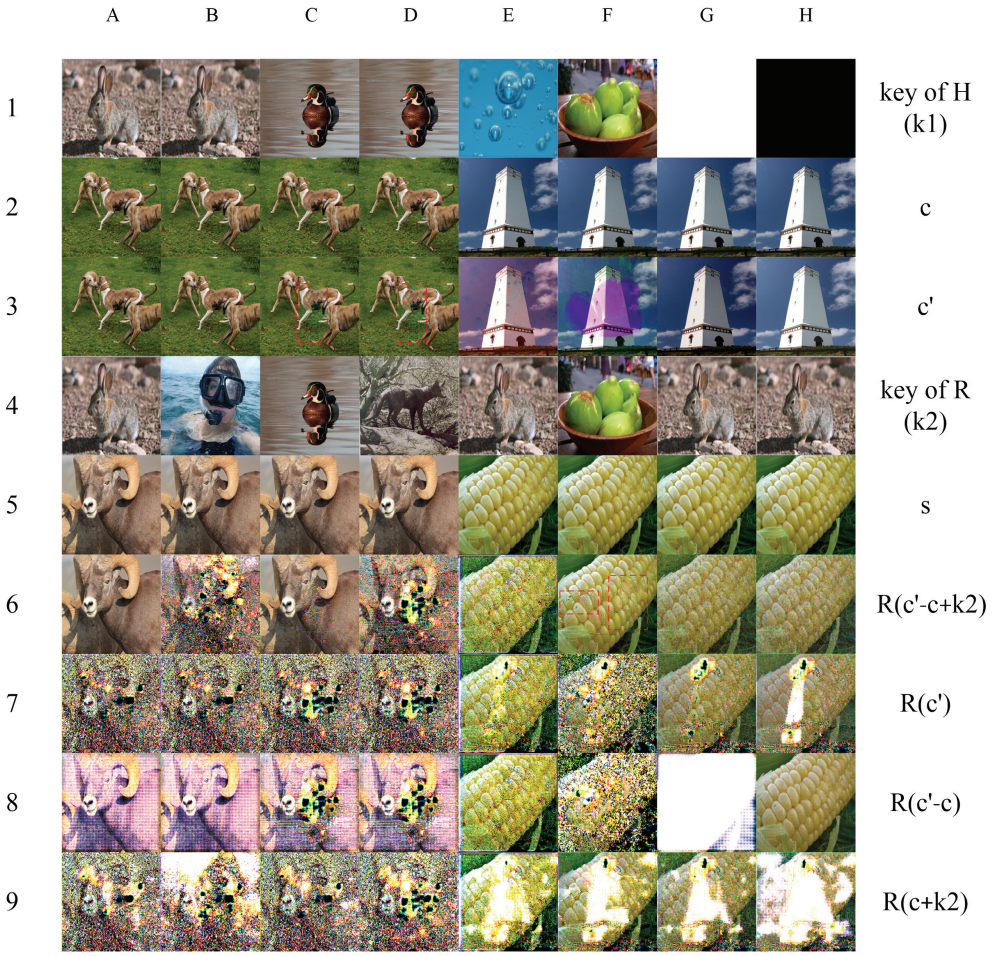

Figure 4. The comparison of hiding and revealing effects when using different key_imgs, columns A-H correspond to rows with the same name in Table 1. Lines 1–9 are the images in experiment of DIGDH, which are numbered for convenience of description. The content represented by each line refers to the simplified symbol on the rightmost of current line.

In Figure 4, among A1 to H9, A1 to A6 are the images in the normal process of DIGDH, others are when the sender or receiver does not have the right key_img or original cover image. From A6 to H9, except that A6 looks like a normal image, the rest of them with distortion more or less. The seventh row is the revelation results when the receiver only holds the container images, the revelation effects of row 9 perform not so better than that in the row 7, and the receiver just holds the key_img of R extraly. In row 8 compared to row 7, the receiver only holds one more cover image, and the eye can see a significant improvement in the revelation effects of R.

As can be seen from columns C and F, even if the sender and receiver use the same key_img but not the right key_img, container image and revealed secret image will be with

noise, to say nothing of using different key_imgs such as columns B, D and E. As can be seen from columns G and H, when the pixel values of key_img are 0 or 255, the container image has no obvious noise than the cover image, but the receiver only needs to use the right key_img and R to distinguish whether the container image is from the real sender. In other words, the receiver can find out whether the right key_img is used in generating the container image by observing the revealed secret image.

In order to observe the hiding effects intuitively, there shows the difference images between original images and generated images of StegoPNet [15] based on different frameworks in Figure 5.

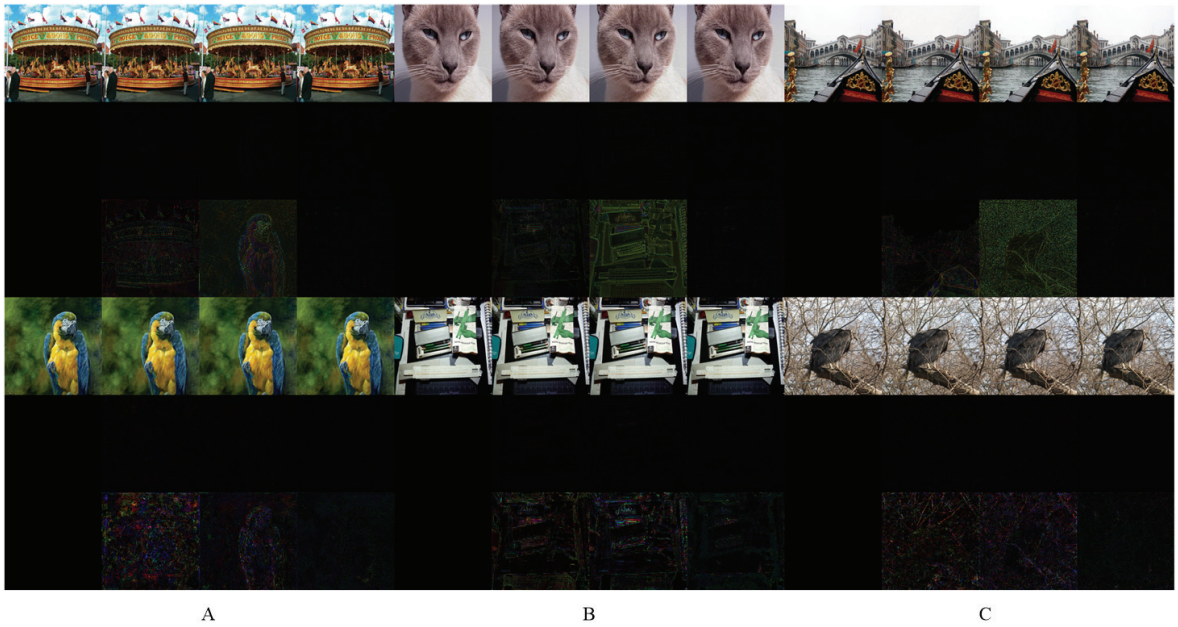

Figure 5. The comparison of difference images between the generated images and the original images based on DDH, UDH and DIGDH, A, B, and C are three groups of randomly selected test results.

In Figure 5, A, B, and C are three groups of randomly selected test results, and each of them as one comparison diagram. Each comparison diagram is divided into four columns and six rows. The first row is the original cover image c, c′ of DDH, c′ of UDH and c′ of DIGDH from left to right, and the second row is the difference images between the image directly above the column and c. Because the pixel values of difference images are very small, in order to get clearer observation, we enlarge the values of the second row by 10 times to get the third row. The fourth row is the original secret images, s′ of DDH, s′ of UDH and s′ of DIGDH from left to right. The fifth and sixth rows are the images of difference images between s′ and s, and the corresponding rules can ben consulted from the second and third rows.

Figure 5 displays that the trained hiding model can generate container images with high similarity to the original cover images, and the trained reveal model can obtain revealed secret images with high similarity to the original secret images. Therefore, we selected two groups from the test data randomly to observe the influence of the trained hiding model and the reveal model on the histogram of each image.

In Figure 6, we selected two groups of test images randomly to observe the images' changes and histograms' changes before and after processing of models. We found that the histogram change of each image is not so obvious.

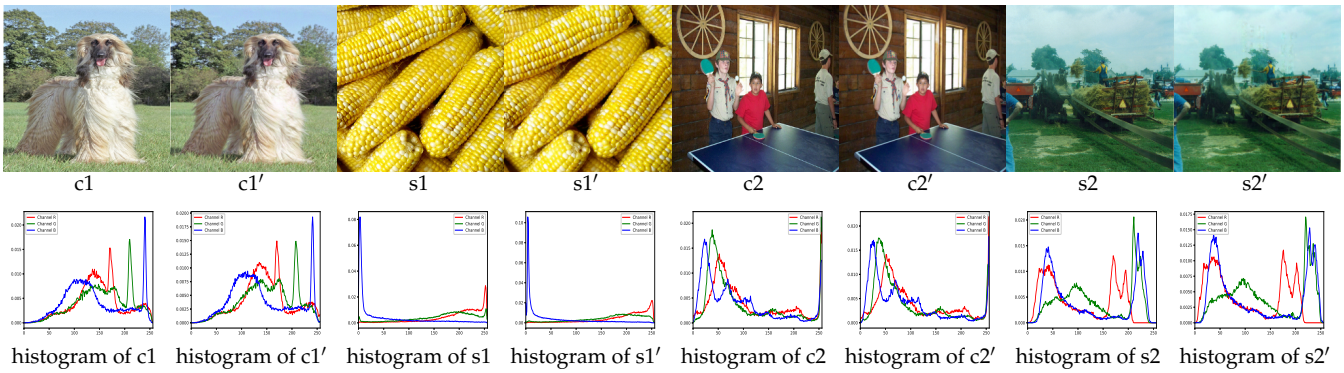

Figure 6. The comparison of the original images and generated images and the change of their RGB-channel histograms, $c1$ and $c2$ are the original cover images, $s1$ and $s2$ are the original secret images, $c1'$ and $c2'$ are the container images generated by the hiding network model, and $s1'$ and $s2'$ are the revealed secret images generated by the reveal network model, and below each image is the corresponding histogram of its.

4.2. Objective Analysis

The peak signal to noise ratio (PSNR) [24] and structural similarity (SSIM) [25] are used to measure the experimental results in this article. In the field of image hiding, the PSNR is usually used to measure the quality of the images generated by trained models, the higher value of PSNR between generated image and original image, the higher the similarity of them. The formula of PSNR is as follows:

$$PSNR = 10 \cdot \lg\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \lg\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{15}$$

where MAX_I represents the maximum possible value of image point pixel, here $MAX_I = 2^n - 1$. In this article, for the reason that the images in the experiments which are RGB color images with three channels, and the pixels of each channel are represented by 8 bits, consequently, the value of MAX_I should be 255 here.

There is a well-known quality metric used to measure the similarity between two images named SSIM, and is regarded to be correlated with the quality perception of the human visual system (HVS). In contrast to using error summation methods of traditional, the SSIM is designed by modeling the image distortion as a combination of three different factors that are luminance distortion, loss of correlation and contrast distortion. Given two images named X and Y, respectively, the value of SSIM of the two images can be calculated as follows:

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{16}$$

where μ_x and μ_y represent the average values of X and Y, respectively, σ_x^2 and σ_y^2 represent the variance of X and Y, respectively, σ_{xy} represents the covariance between X and Y, c_1 and c_2 are use to maintain a stable constant in order to avoid the denominator being 0.

Researchers use relative capacity to measure the load capacity of the scheme in the field of image hiding based DCNN as usual, and the calculation method of relative capacity is as follows:

$$Relative\ capacity = \frac{Absolute\ capacity}{size\ of\ image} \tag{17}$$

We apply H and R with the same structures as the models in [15] to DIGDH, and apply H and R with the same structures as the models in [10] to DIGDH. The experimental results are shown in Table 2.

Table 2. The table of comparative analysis of average PSNR and SSIM of hiding and revelation.

Framework Based Method	PSNR(c, c′), SSIM(c, c′)	PSNR(s, s′), SSIM(s, s′)	RC
[10] based [26]	39.13, 0.985	39.18, **0.992**	1
DIGDH based [26]	39.64, 0.983	40.29, 0.991	1
DDH based [15]	40.48, 0.986	38.97, 0.985	1
DIGDH based [15]	**42.35**, **0.988**	**41.67**, 0.991	1

In Table 2, we use RC to represent the relative capacity, what we can see is that when the relative capacity is 1, the PSNR and SSIM between the container images obtained based on DIGDH and the original cover images are higher, and the PSNR and SSIM between the revealed secret images and the original secret images are still outstanding.

In order to illustrate the effect of DIGDH on reducing models' loss, the change of loss during the models' training is shown as Figure 7. The loss is calculated by the Formula (10).

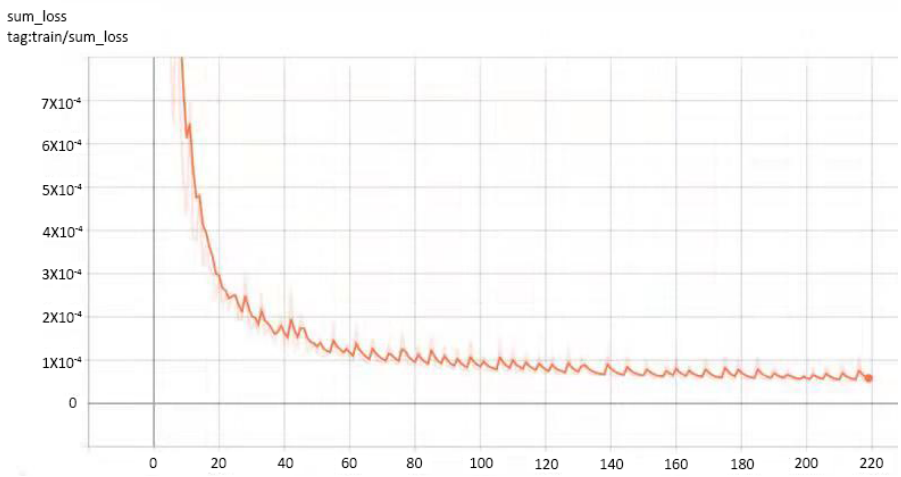

Figure 7. The change of loss during models' training based on DIGDH.

We compared the losses of training StegoPNet based on DDH and DIGDH, and the results are shown in Table 3. Obviously, the loss function value corresponding to the images generated by the models based on DIGDH training is smaller.

Table 3. The loss function value of StegoPNet based on the DDH and DIGDH.

Epoch	Loss Based DIGDH	Loss Based DDH
20	2.8897×10^{-4}	7.8497×10^{-4}
40	1.3723×10^{-4}	4.7537×10^{-4}
60	1.0605×10^{-4}	4.3778×10^{-4}
80	8.8538×10^{-5}	4.3683×10^{-4}

We also use the steganalysis detection tool which named StegExpose [18] to analyze our experimental results which consist of StegoPNet based on DDH and StegoPNet based on DIGDH, and the corresponding receiver operating characteristic (ROC) curves are drown according to the analysis results.

As we can see from Figure 8, the ROC curve of DIGDH is closer to random distribution. Only an image hiding framework has been proposed by us rather than a secure steganography.

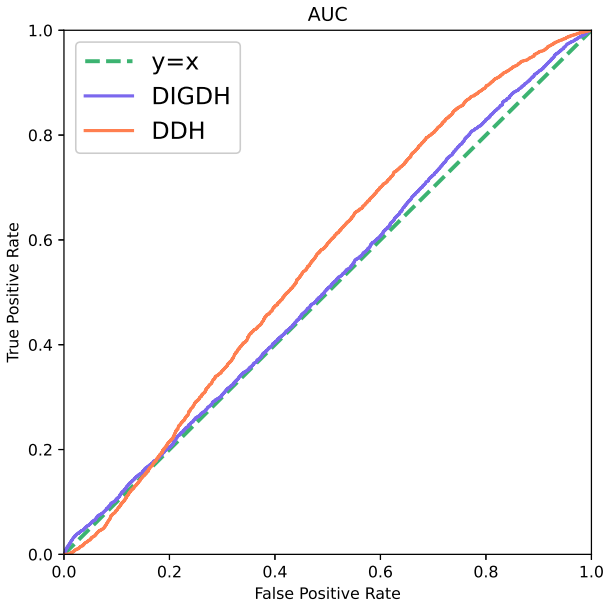

Figure 8. The ROC curves of StegoPNet based on DIGDH and DDH.

From Figure 8 we can observe whether the DIGDH brings negative impacts to the original network models.

5. Conclusions

The existing mainstream deep hiding frameworks mainly consist of DDH and UDH. However, based on the training models of these two frameworks, there is still room for improvement in the similarity between the generated images and the original images, and it is hard for the receiver to distinguish whether the encrypted image comes from the real sender.

We propose a novel hiding framework named DIGDH, which combines the over-fitting characteristic of DNN and the grafting method of difference image. When the relative capacity is 1, the models trained based on DIGDH can generate images with higher similarity to the original images, and the receiver can identify whether the container image comes from the real sender easily. DIGDH can better explore the potential of the network models.

However, DIGDH still with some shortcomings. Similar to UDH, the receiver needs to have the original cover image to obtain the revealed secret images but DDH does not need. UDH can be used for robust watermarking, but DIGDH does not have this ability or has yet to be developed.

For the DNN, the results of over-fitting in most cases are contrary to the researchers' expectations. Similarly, we also encounter this problem in the research process, but we can avoid the disadvantages of over-fitting through the method of difference image grafting, and we can also make use of its advantages. We have learned the idea of turning waste into treasure from this study and will apply it to future scientific research.

Author Contributions: Conceptualization, X.D., W.W., L.L., Y.S., X.W. and E.Z.; methodology, L.L.; software, L.L. and W.W.; validation, L.L. and Y.S.; formal analysis, all authors; writing—original draft preparation, L.L.; writing—review and editing, L.L.; visualization, L.L.; project administration, X.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant (U1904123, U20B2051, 62172280 and 62072157).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Ghadami, N.; Gheibi, M.; Kian, Z.; Faramarz, M.G.; Tian, G. Implementation of solar energy in smart cities using an integration of artificial neural network, photovoltaic system and classical Delphi methods. *Sustain. Cities Soc.* **2021**, *74*, 103149. [CrossRef]
2. Moosavi, J.; Naeni, L.M.; Fathollahi-Fard, A.M.; Fiore, U. Blockchain in supply chain management: A review, bibliometric, and network analysis. *Environ. Sci. Pollut. Res.* **2021**. [CrossRef] [PubMed]
3. Qin, C.; Liu, E.; Feng, G.; Zhang, X. Perceptual Image Hashing for Content Authentication Based on Convolutional Neural Network With Multiple Constraints. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4523–4537. [CrossRef]
4. Zhang, C.; Lin, C.; Benz, P.; Chen, K.; Zhang, W.; Kweon, I.S. A brief survey on deep learning based data hiding, steganography and watermarking. *arXiv* **2021**, arXiv:2103.01607.
5. Wang, R.Z.; Lin, C.F.; Lin, J.C. Image hiding by optimal LSB substitution and genetic algorithm. *Pattern Recognit.* **2001**, *34*, 671–683. [CrossRef]
6. Zhu, J.; Kaplan, R.; Johnson, J.; Li, F.-F. Hidden: Hiding data with deep networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 657–672.
7. Subramanian, N.; Cheheb, I.; Elharrouss, O.; Al-Maadeed, S.; Bouridane, A. End-to-End Image Steganography Using Deep Convolutional Autoencoders. *IEEE Access* **2021**, *9*, 135585–135593. [CrossRef]
8. Pevnỳ, 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.
9. Holub, V.; Fridrich, J. Designing steganographic distortion using directional filters. In Proceedings of the 2012 IEEE International Workshop on Information Forensics and Security (WIFS), Costa Adeje, Spain, 2–5 December 2012; pp. 234–239.
10. Zhang, C.; Benz, P.; Karjauv, A.; Sun, G.; Kweon, I.S. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 10223–10234.
11. Qin, C.; Ji, P.; Chang, C.C.; Dong, J.; Sun, X. Non-uniform Watermark Sharing Based on Optimal Iterative BTC for Image Tampering Recovery. *IEEE MultiMedia* **2018**, *25*, 36–48. [CrossRef]
12. Wu, P.; Yang, Y.; Li, X. Image-into-image steganography using deep convolutional network. In *Pacific Rim Conference on Multimedia*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 792–802.
13. Wu, P.; Yang, Y.; Li, X. Stegnet: Mega image steganography capacity with deep convolutional network. *Future Internet* **2018**, *10*, 54. [CrossRef]
14. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
15. Duan, X.; Wang, W.; Liu, N.; Yue, D.; Xie, Z.; Qin, C. StegoPNet: Image Steganography with Generalization Ability Based on Pyramid Pooling Module. *IEEE Access* **2020**, *8*, 195253–195262. [CrossRef]
16. Baluja, S. Hiding images in plain sight: Deep steganography. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 2069–2079.
17. Baluja, S. Hiding images within images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 1685–1697. [CrossRef] [PubMed]
18. Boehm, B. Stegexpose—A tool for detecting LSB steganography. *arXiv* **2014**, arXiv:1410.6656.
19. Isac, B.; Santhi, V. A study on digital image and video watermarking schemes using neural networks. *Int. J. Comput. Appl.* **2011**, *12*, 1–6. [CrossRef]
20. Hayes, J.; Danezis, G. Generating steganographic images via adversarial training. In Proceedings of the Advances in Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
21. Tancik, M.; Mildenhall, B.; Ng, R. Stegastamp: Invisible hyperlinks in physical photographs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2117–2126.
22. Wu, K.C.; Wang, C.M. Steganography using reversible texture synthesis. *IEEE Trans. Image Process.* **2014**, *24*, 130–139. [PubMed]
23. Li, F.-F.; Deng, J.; Li, K. ImageNet: Constructing a large-scale image database. *J. Vis.* **2009**, *9*, 1037.
24. Horé, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In Proceedings of the International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
25. Ye, Y.; Shan, J.; Bruzzone, L.; Shen, L. Robust registration of multimodal remote sensing images based on structural similarity. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2941–2958. [CrossRef]
26. Duan, X.; Jia, K.; Li, B.; Guo, D.; Zhang, E.; Qin, C. Reversible image steganography scheme based on a U-Net structure. *IEEE Access* **2019**, *7*, 9314–9323. [CrossRef]