electronics

*Article*

# Layerwise Adversarial Learning for Image Steganography

**Bin Chen [1], Lei Shi [2,3,*], Zhiyi Cao [4,*] and Shaozhang Niu [5]**

[1] Department of Computer Application, Shijiazhuang Information Engineering Vocational College, Shijiazhuang 050025, China
[2] State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China
[3] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China
[4] College of Computer and Cyberspace Security, Hebei Normal University, Shijiazhuang 050025, China
[5] Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China
* Correspondence: leiky_shi@cuc.edu.cn (L.S.); zhiyicao@hebtu.edu.cn (Z.C.)

**Abstract:** Image steganography is a subfield of pattern recognition. It involves hiding secret data in a cover image and extracting the secret data from the stego image (described as a container image) when needed. Existing image steganography methods based on Deep Neural Networks (DNN) usually have a strong embedding capacity, but the appearance of container images is easily altered by visual watermarks of secret data. One of the reasons for this is that, during the end-to-end training process of their Hiding Network, the location information of the visual watermarks has changed. In this paper, we proposed a layerwise adversarial training method to solve the constraint. Specifically, unlike other methods, we added a single-layer subnetwork and a discriminator behind each layer to capture their representational power. The representational power serves two purposes: first, it can update the weights of each layer which alleviates memory requirements; second, it can update the weights of the same discriminator which guarantees that the location information of the visual watermarks remains unchanged. Experiments on two datasets show that the proposed method significantly outperforms the most advanced methods.

**Keywords:** image generation; image steganography; layerwise adversarial learning; generative adversarial networks

**Citation:** Bin, C.; Shi, L.; Cao, Z.; Niu, S. Layerwise Adversarial Learning for Image Steganography. *Electronics* **2023**, *12*, 2080. https://doi.org/10.3390/electronics12092080

Academic Editor: Catalin Stoean

Received: 3 April 2023
Revised: 26 April 2023
Accepted: 29 April 2023
Published: 1 May 2023

## 1. Introduction

Image steganography belongs to a branch of the pattern recognition field. It is widely used for covert communication and copyright protection. Image steganography usually requires both the accurate extraction of hidden information and the perfect restoration of the container image. DNN-based image steganography usually needs a Hiding Network to hide secret data in a cover image to generate a container image that has a similar appearance to the cover image, and it also requires a Reveal Network to extract the secret data from a container image when needed. Due to the development of DNN in recent years, the embedding capacity of secret data has made significant progress. However, the challenge of image steganography arises mainly because some secret data with visual watermarks may change the appearance and underlying statistics of the container image [1]. For example, when a container image and a cover image display different colors, it can be considered a failure of information hiding.

Recently, DNN-based image steganography has attracted prominent attention from researchers. Of these methods, some practical methods borrow heavily from auto-encoding networks [1] to directly hide full-size secret data in the cover images. Although the embedding capacity is improved, its training process is more complicated. These methods train a U-Net architecture generator [2] to further simplify the training process. However, the image quality of container images synthesized by most models is poor. To achieve more visually realistic container images, most algorithms utilize Generative Adversarial Networks (GANs) [3] to hide information [4–6]. Although existing DNN-based methods have a strong capacity for container images when the secret data contain visual watermarks, the appearance of the container images generated by these methods will change significantly; this can be considered a failure of information hiding. The Position Encoding Network (PEN) [7] reveals that position information is implicitly learned from the commonly-used padding operation (zero-padding). Based on the PEN, we speculated that one of the reasons for this phenomenon is that, during the end-to-end training process of their Hiding Network, the location information of the visual watermarks has changed.

Inspired by efficient layerwise adversarial training [8], we noticed that existing DNN-based methods have not fully utilized the expressive power of all layers. As a result, we proposed a layerwise adversarial training method based on GANs to improve this situation. Different from the existing auto-encoding networks, the U-Net architecture generator, and GANs models, the proposed model is a new layerwise GAN. Besides the output layer of the Hiding Network, we added a single-layer sub-network and a discriminator behind each layer to capture the representational power of these layers.

The main contributions of the proposed method for image steganography are summarized as follows: Firstly, we proposed a layerwise adversarial training method to address the question of the location information changed in the end-to-end training process. In addition, the proposed discriminator of the layerwise adversarial training method can serve as an improved steganalyzer. Finally, the combination of cosine similarity and GAN loss effectively captures location information through the layerwise adversarial training approach.

## 2. Related Works

This section deliberates the literature review of image steganography. Firstly, we reviewed the traditional image steganography methods and DNN-based image steganography methods. Then, the layerwise learning methods were reviewed.

### *2.1. Traditional Image Steganography Methods*

The secret data hiding capacity of the traditional image steganography methods is usually lower than 0.4 bpp (bits per pixel). They mainly involve image steganography of the spatial domain and the transform domain. The spatial domain hiding methods include the Least Significant Bit (LSB)-related hiding method [9]. To improve the robustness of LSB-related methods, some methods, such as in [10,11], were proposed to embed secret data in the transform domain. These LSB methods embedded the same payload as the LSB matching. Then, an improved LSB method [12] was introduced to embed more data in the noisy or complex texture regions of the cover image. Later, refs. [13,14] successfully embedded more secret data directly into the image pixel values with more complex rules. Subsequently, the Wavelet Obtained Weights (WOW) method [15] can embed the payload into the cover image. Although these spatial domain methods are excellent, they have less robustness. To improve the robustness of these methods, some methods such as a Discrete Fourier Transform (DFT)-based [16] method, a Discrete Cosine Transform (DCT)-based [10] method, and a Discrete Wavelet Transform (DWT)-based [11] method were proposed to embed secret data in the transform domain. Liu et al. [17] proposed a large feature mining-based approach to address the highly challenging detection problems. However, these methods have poor quality and limited embedding capacity for the container images, which aim to embed secret data by modifying the pixel values of the cover image or the coefficients of the transformed image.

### *2.2. DNN-Based Image Steganography Methods*

Recently, DNN-based image steganography methods have attracted great attention from researchers because they can generate a container image rather than modify the cover image. In the previous pioneer works, Ma et al. [18] proposed a general steganalysis feature selection method based on a decision rough set of positive region reduction. In [19], Ye et al. presented an alternative approach to the steganalysis of digital images based on a Convolutional Neural Network. Wang et al. [20] presented a novel performance evaluation method of steganalysis based on posterior accuracy. In [21], Qian et al. proposed a new paradigm for steganalysis to learn features automatically via deep learning models. Xu et al. [22] reported a Convolutional Neural Network architecture that takes into account knowledge of steganalysis. In [23], Husien et al. concluded that the TICSS is very rapid in performing the extraction process, and the size of the embedded text does not affect the speed of the system very much. Brandao et al. [24] presented a technique for transmitting information efficiently and securely. In [25], Boroumand et al. describe a deep residual architecture for both spatial-domain and JPEG steganography. Zeng et al. [26] proposed a generic hybrid deep-learning framework for JPEG steganalysis incorporating the domain knowledge behind rich steganalytic models. In [27], Zhang et al. proposed a new strategy that constructs enhanced covers against neural networks with the technique of adversarial examples. These models successfully utilized DNN for image steganography or incorporated DNN into the hidden process. Meanwhile, Hayes et al. [4] and Tang et al. [5] directly hid full-size images in the cover images by exploiting the GANs model [3]. Other excellent methods [1] (Encode1) built an auto-encoding type Hiding Network and [2] (U-Net1) designed a U-Net-type [28] Hiding Network followed by a discriminator. Hu et al. [29] presented a new cover-lossless robust image watermarking method by efficiently embedding a watermark into low-order Zernike moments and reversibly hiding the distortion due to the robust watermark as the compensation information for restoration of the cover image. Tancik et al. [30] presented an architecture, algorithms, and a prototype implementation addressing this vision. Sua et al. [31] aimed to improve the performance results by using a novel combination with Convolutional Neural Networks and sequence graph transform. Pugliese et al. [32] presented a comprehensive view of geo-worldwide trends of ML-based approaches, highlighting the rapid growth in the last 5 years attributable to the introduction of related national policies. Kha et al. [33] proposed a novel model constructed on the multi-scan Convolutional Neural Network and position-specific scoring matrix profiles to address these limitations. Lu et al. [34] proposed a large-capacity Invertible Steganography Network for image steganography. Mahdy et al. [35] presented the Elzaki transform homotopy perturbation technique to address the nonlinear Emden–Fowler systems. Ray et al. [36] used a Convolutional Neural Network with a Deep Supervision-based edge detector, which can retain more edge pixels over conventional edge detection algorithms. Mahdy et al. [37] applied fractional order to the glioblastoma multiforme (GBM) and IS interaction models. Liu et al. [6] proposed the Image DisEntanglement Autoencoder for Steganography as novel steganography without an embedding technique. Xu et al. [38] presented a novel flow-based framework for robust invertible image steganography. Although existing methods have better quality and strong embedding capacity of the container images when the secret data contain visual watermarks, the appearance of the container images generated by these methods will change significantly.

### *2.3. Layerwise Learning Methods*

The layerwise learning methods usually take advantage of each layer for local backpropagation. The layerwise learning methods update the weights before the forward and backward pass has completed [39–45]. More similar to our method, Efficient Layerwise Adversarial Training (ELAT) [8] practices adversarial perturbations of intermediate layer activations to provide a stronger regularization and improves adversarial robustness comparable to traditional adversarial training approaches. Another similar method is Greedy Layerwise Learning (GLL) [46], which employs 1-hidden layer learning problems to sequentially build deep networks layer by layer, and is able to inherit properties from shallow networks. Though layerwise learning methods achieve the better-reported result, they focus on image classification tasks rather than image generation tasks. In essence, image steganography belongs to image generation tasks.

## 3. Proposed Method

In this section, we introduce the proposed image steganography method based on the layerwise adversarial training. As shown in Figure 1, the proposed method constructs a Hiding Network to hide secret data in the cover images and a Reveal Network to extract the secret data from the container images. We first introduce the Network Architecture of the proposed method. Then, two objective functions will be presented separately for the layerwise adversarial training Hiding Network and the Reveal Network.

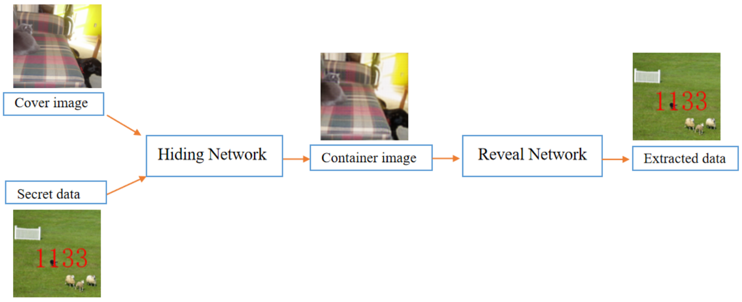

**Figure 1.** An overview of the network architecture. The proposed method includes a Hiding Network and a Reveal Network.

### 3.1. The Proposed Network Architecture

In this section, we introduce the network architecture of the proposed method shown in Figure 2. To build the Hiding Network, the single-layer sub-network and the discriminator network should be introduced first. Then, we present the network architecture of the Reveal Network.

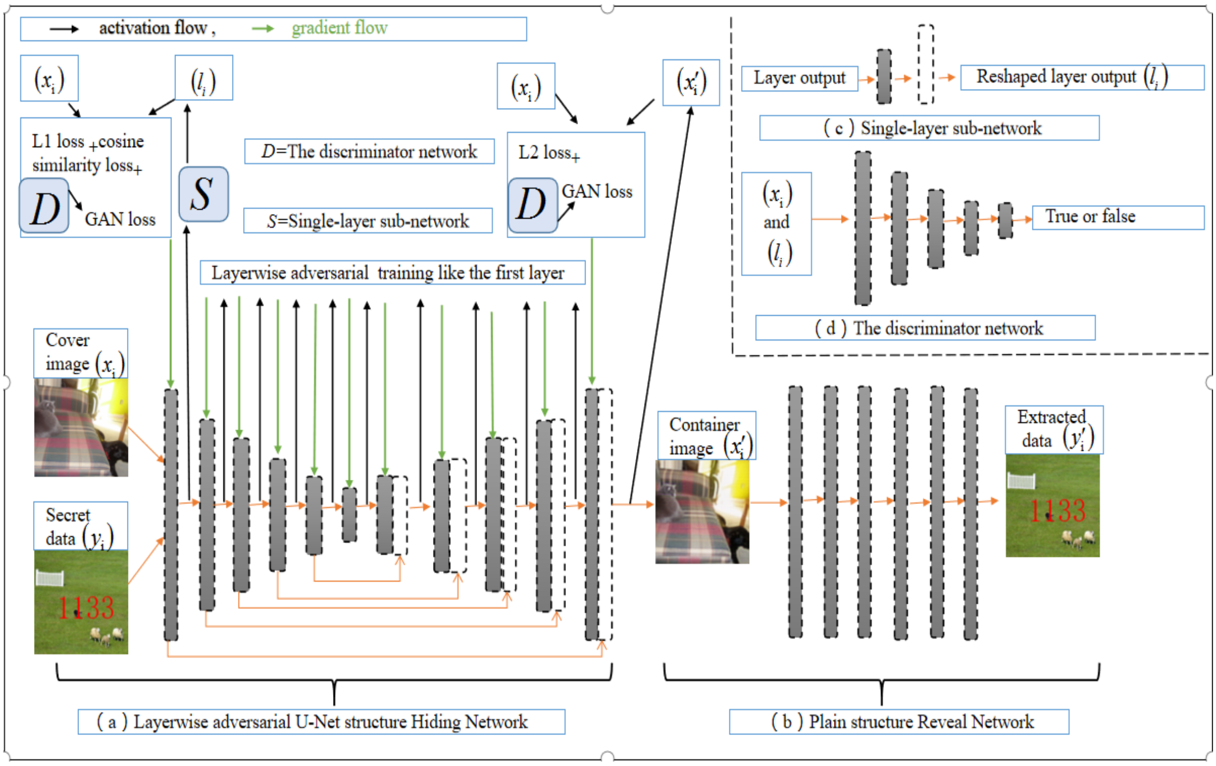
activation flow ,
gradient flow
$(x_i)$
$(l_i)$
L1 loss +cosine similarity loss+
D
GAN loss
S
D=The discriminator network
S=Single-layer sub-network
$(x_i)$
$(x'_i)$
L2 loss+
D
GAN loss
Layerwise adversarial training like the first layer
Cover image $(x_i)$
Secret data $(y_i)$
1133
Container image $(x'_i)$
Extracted data $(y'_i)$
1133
Layer output
Reshaped layer output $(l_i)$
( c ) Single-layer sub-network
$(x_i)$ and $(l_i)$
True or false
( d ) The discriminator network
( a ) Layerwise adversarial U-Net structure Hiding Network
( b ) Plain structure Reveal Network

**Figure 2.** The proposed layerwise adversarial training network architecture. (**a**) The Hiding Network is essentially a layerwise adversarial U-Net type "fountain" architecture. (**b**) The Reveal Network is essentially a plain DNN structure. (**c**) The single-layer sub-network *S* consists of a deconvolution layer followed by a ReLU activation layer. (**d**) The discriminator network *D*.

Different from the U-Net type Hiding Network of U-Net1[2], as shown in Figure 2a, our Hiding Network is essentially a layerwise adversarial U-Net type "fountain" architecture. Obviously, the "fountain" outputs the activation flow but receives the gradient flow. Compared with ELAT [8], besides the output layer of the Hiding Network, this paper also added a single-layer sub-network and a discriminator behind each layer to capture the representational power of these layers. Therefore, compared to ELAT, the similarity is that the representational power of both is used to update the weights of each layer, which alleviates memory requirements, while the difference is that the representational power of our method is used to update the weights of the same discriminator. It should be noted that the layerwise adversarial training method only exists in the Hiding Network. More importantly, all the layers are used to update the weights of the same discriminator. The discriminator network guarantees that the location information of the visual watermarks remains unchanged. In the layerwise adversarial training process, the discriminator is continuously learned to discriminate against the location difference. A better discriminator enhances the generation ability of the proposed method and it will generate a better-quality container image. Note, when the proposed method was tested, all the sub-networks and discriminators were dropped.

#### 3.1.1. The Single-Layer Sub-Network

The single-layer sub-network keeps the output shape of a convolutional unit that matches the shape of the cover image. As shown in Figure 2c, the single-layer sub-network consists of a deconvolution layer followed by a ReLU activation layer. The single-layer sub-network was placed to follow each layer of the Hiding Network. It should be noted that, since the shape of each layer of the Hiding Network is different, the single-layer sub-network also contains different parameters to output the same shape.

#### 3.1.2. The Discriminator Network

The discriminator network makes a distinction between the output of the single-layer sub-network and the cover image. As shown in Figure 2d, $D$ consists of five convolution layers followed by the IN layer and a ReLU activation layer to downsample the input features.

#### 3.1.3. The Network Architecture of Hiding Network

The Hiding Network hides secret data in the cover images and generates container images that have a similar appearance to the cover images. As shown in Figure 2a, the proposed U-Net architecture consists of six convolution layers followed by a Batch Normalization (BN) layer and a ReLU activation layer [47] to downsample the input features and five deconvolution layers, followed by a BN layer and a ReLU activation layer to upsample the features. For the output layer, the ReLU activation function was replaced by the sigmoid activation function. For each convolution layer and deconvolution layer, $4 \times 4$ spatial filters were utilized with stride 2. For the Hiding Network, this paper took each convolutional or deconvolutional layer followed by the BN layer and the ReLU activation layer as a convolutional unit. Except for the output layer of the Hiding Network, each convolutional unit was followed by a single-layer sub-network $S$ and a discriminator $D$. This paper employed the sigmoid activation function instead of ReLU at the final layer to output probabilities from the logits.

#### 3.1.4. The Network Architecture of Reveal Network

Similar to the Encode1 method [1], the Reveal Network of this paper was a plain DNN structure to extract the secret data from the container images. As shown in Figure 2b, this paper used six convolution layers followed by the BN layer and the ReLU activation layer to accurately recover information from the container images. For the output layer, this paper employed the sigmoid activation layer instead of ReLU. For each convolution layer, $3 \times 3$ spatial filters were employed.

### 3.2. The Objective Function

We started by defining some notations. Let $\{x_i\}_{i=1}^{n}$ denote the set of cover images, $\{y_i\}_{i=1}^{n}$ denote the set of secret data, and $\{x_i'\}_{i=1}^{n}$ denote the set of container images. For the Hiding Network, this paper took a cover image $x_i$ and the secret data $y_i$ as inputs and generated a container image $x_i'$.

#### 3.2.1. The Objective Function of the Layerwise Adversarial Learning Hiding Network

To prevent the location information of the visual watermarks being changed during the end-to-end training process, we proposed a Hiding Network with layerwise adversarial learning. In the end-to-end deep training process, the loss function produces a gradient from the output layer, and this gradient is backpropagated to hidden layers to dictate the location information of the visual watermarks. Since the loss function does not directly see the location information of the visual watermarks in each hidden layer, layerwise learning provides a solution for this. In the layerwise learning process, the loss function produces a gradient from each hidden layer, and this gradient is backpropagated to itself to dictate the location information of the visual watermarks.

Besides the output layer, this paper took each convolutional or deconvolutional layer followed by the BN layer and the ReLU activation layer [47] as a convolutional unit. We assumed that the Hiding Network with loss function $J$ and parameters $\theta$ contains a convolutional unit $C$ . Here, a convolutional unit followed by a single-layer sub-network can be seen as a small generator network $G$. $\{G_c\}_{c=1}^{C}$ store the gradient signal with the location information $P = \{p_c\}_{c=1}^{C}$, initialized with zero. Backward pass is performed using loss function $J$. The gradient signal is backpropagated to its layer by:

$$p_c = \text{sign}(\nabla_c \mathcal{J}(\theta, x + P, y)), \forall c = [1, C],$$

where $\text{sign}(\cdot)$ denotes the signum function and $\mathcal{J}(\cdot)$ denotes the loss function. Forward pass is performed, keeping the location information: $G_c$ storing the gradient signal acts as follows:

$$G_c(x_c) = x_c + \cdot p_c.$$

Through each single-layer sub-network, this paper obtained the reshaped outputs of each convolutional unit $\{l_i\}_{i=1}^{n}$ that should match the shape of the cover images. As shown in Figure 2a, the L1 loss and the cosine similarity loss between $x_i$ and $l_i$ is calculated first. Then, $D$ is exploited to discriminate which of them is true. Furthermore, the GAN loss [3] between $x_i$ and $l_i$ is calculated. Finally, the L1 loss, the cosine similarity loss, and the GAN loss are utilized to handle local backpropagation and update the weights of each convolutional unit and the weights of the same discriminator. For the adversarial learning [3] between $G$ and $D$ of these layers, we train the proposed layer-wise adversarial process by solving a minimax optimization problem given by:

$$J = \min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda_1 \mathcal{L}_1(G) + \lambda_2 \mathcal{L}_2(G) \tag{1}$$

where $\mathcal{L}_{GAN}$, $\mathcal{L}_1$, $\mathcal{L}_2$ are the GAN loss, the L1 loss and the cosine similarity loss. The L1 loss can be expressed as:

$$\mathcal{L}_1(G) = \mathbb{E}_{x_i \sim p_{data}(x_i), l_i \sim p_{data}(l_i)}[\|l_i - x_i\|_1]. \tag{2}$$

The L1 loss ensures that the reshaped layer output $l_i$ of the generator $G$ is close to the cover image $x_i$. The GAN loss can be expressed as:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{l_i \sim p_{data}(l_i)}\left[\log D_{L_i}(l_i)\right] + \mathbb{E}_{x_i \sim p_{data}(x_i)}\left[\log(1 - D_{X_i}(G)\right]. \tag{3}$$

The GAN loss guarantees that $D$ is not fooled by the reshaped layer output $l_i$ of $G$. Here, we denote $CS(\cdot,\cdot)$ as the cosine similarity loss. Then, the cosine similarity loss can be expressed as:

$$\mathcal{L}_2(G) = \mathbb{E}_{x_i \sim p_{data}(x_i), l_i \sim p_{data}(l_i)}[CS(l_i, x_i)]. \tag{4}$$

The combination of cosine similarity loss and the GAN loss can capture the location information of the layerwise adversarial training method. For the output layer, this paper takes the deconvolutional layer followed by the BN layer and the sigmoid activation layer as a convolutional unit. Since the container image $x_i'$ from the output layer has the same shape as the cover image $x_i$, the last convolutional unit is only followed by a discriminator. Here, a convolutional unit can be seen as a generator network $G$, and the L2 loss between $x_i'$ and $x_i$ is employed to create local backpropagation and update the weights of this convolutional unit and the weights of the same discriminator. For the output layer, we trained the proposed layerwise adversarial process by solving a minimax optimization problem given by:

$$\min_G \max_D \mathcal{L}_{GAN}(G,D) + \lambda_3 \mathcal{L}_2(G), \tag{5}$$

where $\mathcal{L}_{GAN}$, $\mathcal{L}_2$ are the GAN loss and the L2 loss, respectively. The L2 loss can be expressed as:

$$\mathcal{L}_3(G) = \mathbb{E}_{x_i, x_i'}\left[\frac{1}{n}\sum_{i=1}^{n} \left\|x_i' - x_i\right\|^2\right]. \tag{6}$$

The L2 loss ensures that the container image $x_i'$ of the generator $G$ has a similar appearance to the cover image $x_i$. In this way, the layerwise adversarial training method for the Hiding Network has been completed. Finally, a container image $x_i'$ is generated by hiding the secret data $y_i$ in a cover image $x_i$.

### 3.2.2. The Objective Function of the Reveal Network

The Reveal Network is used to extract the secret data $\{y_i\}_{i=1}^{n}$ from the container images $\{x_i'\}_{i=1}^{n}$. Let $\{y_i'\}_{i=1}^{n}$ denote the set of extracted secret data; we can obtain the l2 loss for the Reveal Network:

$$\mathcal{L}_2 = \mathbb{E}_{y_i, y_i'}\left[\frac{1}{n}\sum_{i=1}^{n} \left\|y_i' - y_i\right\|^2\right]. \tag{7}$$

The L2 loss ensures that the extracted secret data $y_i'$ have a similar appearance to the secret data $y_i$. In this way, the secret data extracting process of the Reveal Network was completed.

## 4. Experiments

To explore the ability of the proposed image steganography method, we trained and tested the method on two datasets and compared it with the advanced methods. In the training process, this paper employed Adam with $\beta_1$ = 0.5 and $\beta_2$ = 0.999. The batch size was set to 16, the learning rate was 0.0001, and the employed parameter values were $\lambda_1 = 10$, $\lambda_2 = 20$, $\lambda_3 = 10$. The entire training procedure took about 12 h on a single GTX1080Ti GPU for 80 epochs.

### *4.1. Datasets and Preprocessing*

The two datasets of this paper came from COCO [48], which is a large image dataset designed for object detection, segmentation, and person keypoints detection. Based on the category id, the dogs and cats images were selected as two datasets for training and testing. For each dataset, 4000 images were employed to train the model, and 200 images were employed for model testing. Both the secret data and the cover images were from the datasets mentioned above, and the selected order of them was randomized. Then, all the images were further resized to 128 × 128. To compare with existing DNN-based image steganography methods, the visual watermarks were added to the center of the secret data.

### *4.2. Baselines*

To compare the performance of different methods, we chose Encode1 [1], U-Net1 [2], GANs1 [4], and GANs2 [6] as our baselines. GANs1 introduces a game between three parties—Alice, Bob, and Eve—and simultaneously train a steganographic method and a steganalyzer. GANs2 hides the secret message in a cover image by transforming it into a synthesized image with a generator and three discriminators. Encode1 borrows heavily from auto-encoding networks [49] and encodes two images such that the intermediate representation (the container image) appears as similar as possible to the cover image. U-Net1 presents a generator with a U-Net architecture to translate a cover image into a container image, and an enhanced steganalyzer based on a Convolutional Neural Network together with multiple high pass filters as the discriminator.

### *4.3. Evaluation Metrics*

We compared the proposed method qualitatively and quantitatively with several baselines using four evaluation metrics. Here, four evaluation indicators (e.g., PSNR, SSIM, ATS, DLAL) were chosen to compare different methods. The first evaluation indicator was Peak Signal to Noise Ratio (PSNR) [50]; PSNR evaluates the peak difference of different images. The unit of PSNR is dB and the larger the value, the smaller the image distortion. It can be expressed as:

$$PSNR = 10\log_{10}\left(\frac{(2^n-1)^2}{MSE}\right), \tag{8}$$

where MSE is the mean square error of the original image and the test image, $(2^n-1)^2$ is the square of the maximum value of the signal, and $n$ is the number of bits of each sample value.

The second evaluation indicator was Structural Similarity (SSIM) [51], and SSIM is used to evaluate the structural difference between different images. It measures image similarity in three ways: brightness, contrast, and structure. The range of SSIM value is $[0,1]$. The closer the SSIM value is to 1, the smaller the distortion effect is. It can be expressed as:

$$\begin{aligned}
\mathrm{SSIM}(X,Y) &= l(X,Y)\cdot c(X,Y)\cdot s(X,Y)\\
l(X,Y) &= \frac{2\mu_X\mu_Y + C_1}{\mu_X^2+\mu_Y^2+C_1}\\
c(X,Y) &= \frac{2\sigma_X\sigma_Y + C_2}{\sigma_X^2+\sigma_Y^2+C_2}\\
s(X,Y) &= \frac{2\sigma_{XY} + C_3}{\sigma_X\sigma_Y+C_3};
\end{aligned}$$

among them, in this paper, $X$ is represented as a cover image and an extracted secret image, respectively; $Y$ is a container image, respectively; $\mu_X$ and $\mu_Y$ represent the mean of the images $X$ and $Y$, respectively; $\sigma_X$ and $\sigma_Y$ represent the standard deviation of the cover image and the container image; $\sigma_X\sigma_Y$ denotes the covariance of the cover image and the container image. The third evaluation indicator is ATS [52]; ATS is used to evaluate the steganography capacity per cover image. The fourth evaluation indicator was the proposed discriminator network (short for DLAL); DLAL can be seen as a steganalyzer to make a distinction between the container image and the cover image.

### *4.4. Experimental Results*

The experimental results of the proposed method on the dogs dataset and the cats dataset are shown in Figure 3. For the dogs dataset, the image steganography results are shown in the first two rows. For the cats dataset, the image steganography results are shown in the last two rows. It is shown that the proposed method is able to generate visually realistic container images that have the same appearance as the cover images. Here, cov-histogram denotes the histogram of cover images and con-histogram denotes the histogram of container images. It has been observed that our container images have a similar histogram to that of the cover images. This means that the proposed Hiding Network successfully hides secret data in the cover images. Additionally, Figure 3 shows that the Reveal Network of this paper is able to extract the secret data from the container images, and the extracted secret data are close to the input secret data.

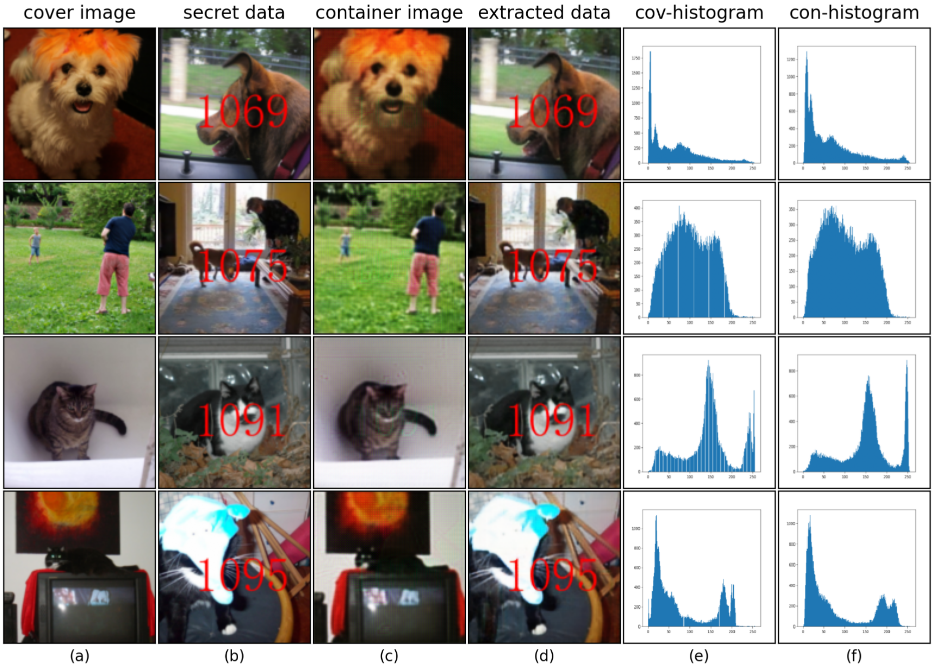

**Figure 3.** The image steganography results from two datasets. (**a**) cover images; (**b**) secret data; (**c**) container images; (**d**) extracted secret data; (**e**) the histogram of cover images; (**f**) the histogram of container images.

Although the proposed method was trained on two small-scale datasets, the trained model can also be applied to other datasets and generate reasonable results. As displayed in Figure 4, the reversible image steganography results for the horses dataset are listed in the first two rows and the reversible image steganography results for the giraffes dataset are listed in the next two rows. Even if we exchange secret data for cover images, listed in the last two rows of Figure 4, the model testing results on the dogs dataset are still reasonable.

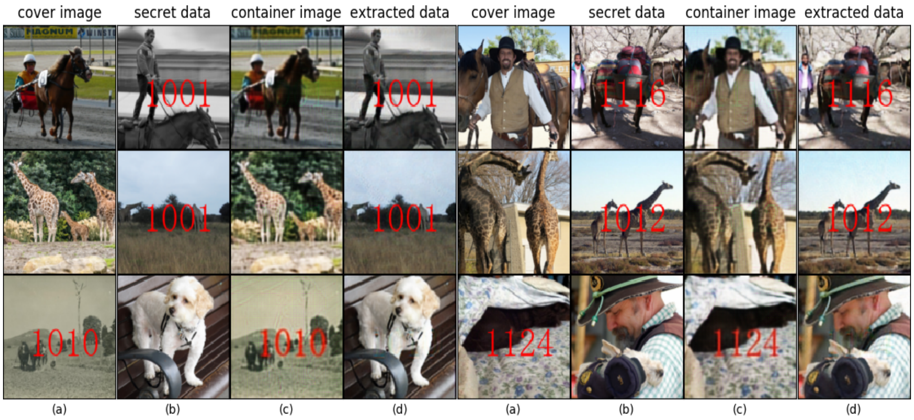

**Figure 4.** More image steganography results from horse, giraffe, and dog datasets. (**a**) cover images; (**b**) secret data; (**c**) container images; (**d**) extracted secret data.

### *4.5. Comparison with Other Methods*

In this section, we compared the proposed image steganography method with four advanced methods.

**Qualitative results.** The randomly selected image steganography experimental comparison results on two datasets are shown in Figure 5. Here, we immediately observe that the container images of the proposed method (ours $_C$) perform better on the two datasets than the GANs1 method ( GANs1$_C$), the GANs2 method (GANs2$_C$), the Encode1 method (Encode1$_C$), and the U-Net1 method (U-Net1$_C$). The obvious limitation is that the appearance difference between their container images and cover images is relatively large. In general, the appearance difference affected by the visual watermarks of secret data leads to the unsafe factors of these methods. This is due to the fact that these methods focus on the representational power of the output layer but ignore the representational power of other layers. Surprisingly, Figure 5 shows that the extracted secret data from all the methods are close to the secret data.

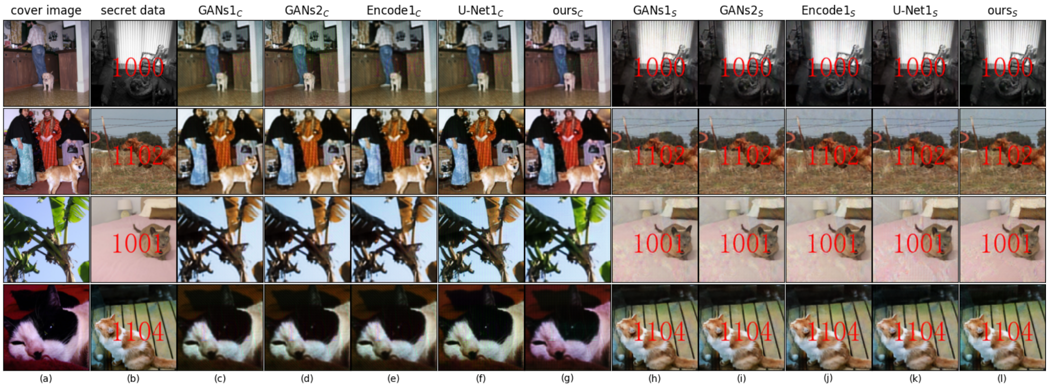

**Figure 5.** The image steganography experiment comparison results. (**a**) cover images; (**b**) secret data; (**c**) GANs1$_C$ denotes the container images of GANs1 method; (**d**) GANs2$_C$ denotes the container images of GANs2 method; (**e**) Encode1$_C$ denotes the container images of Encode1 method; (**f**) U-Net1$_C$ denotes the container images of U-Net1 method; (**g**) our container images; (**h**) GANs1$_S$ denotes the extracted secret data of GANs1 method; (**i**) GANs2$_S$ denotes the extracted secret data of GANs2 method; (**j**) Encode1$_S$ denotes the extracted secret data of Encode1 method; (**k**) U-Net1$_S$ denotes the extracted secret data of U-Net1 method; (**l**) our extracted secret data.

**Quantitative evaluations.** In addition to the visual results, this paper also performed a quantitative evaluation using four metrics on two datasets. As listed in Table 1, PSNR$_{C/S}$ and SSIM$_{C/S}$ denote the average PSNR, SSIM value between 200 cover images and container images and the average PSNR, SSIM value between 200 secret data and the extracted secret data. Since the average PSNR, SSIM value is close on two datasets, only one table was employed to list them.

According to Table 1, the proposed method achieves a leading average PSNR, SSIM value compared to other methods. For two datasets, the average PSNR and SSIM value between 200 cover images and container images reached (31.27/0.8002), the average PSNR and SSIM value between 200 secret data and the extracted secret data reached (41.09/0.9713). For each steganographic algorithm, we trained both ATS on 800 cover images and 800 container images and then reported the accuracy of the steganalyzer on 400 cover images and 400 container images. For DLAL, we used the trained model to test the accuracy on the same test set. From Table 1, DLAL performs competitively against the steganalyzer, ATS, and the proposed steganographic algorithm also performs well against other steganographic methods.

**Table 1.** Quantitative evaluations in terms of two datasets.

| Method | PSNR$_{C/S}$ | SSIM$_{C/S}$ | ATS | DLAL |
|---|---|---|---|---|
| GANs1 | 25.13/39.80 | 0.7612/0.9635 | 0.58 | 0.67 |
| GANs2 | 25.52/39.86 | 0.7622/0.9645 | 0.59 | 0.68 |
| Encode1 | 26.32/40.34 | 0.7637/0.9667 | 0.61 | 0.70 |
| U-Net1 | 26.35/40.56 | 0.7652/0.9705 | 0.63 | 0.72 |
| ours | **31.27/41.09** | **0.8002/0.9713** | **0.70** | **0.86** |

### *4.6. Ablation Study*

In this section, this paper will introduce which part has a greater effect on the hiding secret data with a visual watermark. We defined the layerwise adversarial learning as LAL, the L1 loss as L1, the cosine similarity loss as CS, and the GAN loss as GAN. For the first step, we removed LAL and used only L1 + CS + GAN to train the model. The second step was to remove the CS and use LAL + L1 + GAN to train the model. The third step was to remove GAN and use L1 + LAL + CS to train the model. Some test results are shown in Figure 6:

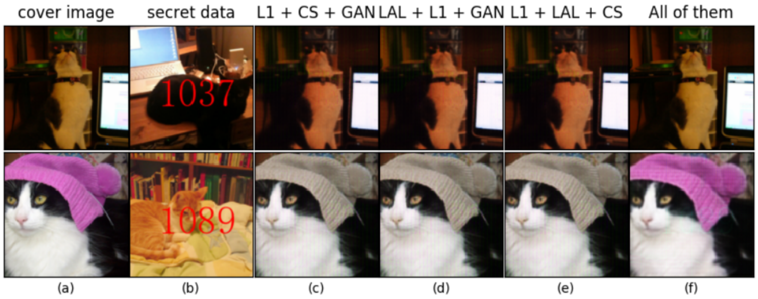

**Figure 6.** Ablation study. (**a**) input cover images; (**b**) input secret data; (**c**) the container images of L1 + CS + GAN; (**d**) the container images of LAL + L1 + GAN; (**e**) the container images of LAL + L1 + GAN; (**f**) the container images by using all of them.

Figure 6 shows that, after removing LAL or removing CS or removing GAN, the watermark information in the secret data cannot be hidden. Therefore, it can be concluded that only the combination of LAL + L1 + GAN + CS can achieve reasonable results. The possible reason is that the combination of GAN + CS can obtain the most accurate watermark position information.

### *4.7. Limitation*

Although the proposed method is able to eliminate appearance changes, it has some limitations. First, the average PSNR value is not very high. Second, we noticed some undesired pixels in the center of the container images. When the color value of a cover image is relatively single, a container image will have some artifacts in the area where the digital watermark appears. These limitations will be studied in the next paper.

## 5. Conclusions

In this paper, we proposed a layerwise adversarial training method to eliminate the appearance changes caused by the existing DNN-based image steganography. To our knowledge, it is the first time a layerwise adversarial training U-Net type “fountain” architecture has been proposed for image steganography tasks. Besides the output layer, we added a single-layer sub-network and a discriminator behind each layer to capture the representational power of these layers and exploit this representational power to update the weights of each layer and update the weights of the same discriminator. Compared with the most advanced methods, the proposed method has achieved leading scores for image steganography on two datasets. Paying more attention to the impact of visual watermarks on image steganography will further improve the robustness and application range of image steganography.

**Author Contributions:** Conceptualization, B.C.; methodology, B.C. and Z.C.; software, Z.C.; validation, Z.C. and L.S.; formal analysis, L.S. and S.N.; investigation, S.N. and L.S.; resources, Z.C. and S.N.; data curation, B.C. and S.N.; writing—original draft preparation, L.S.; writing—review and editing, S.N. and B.C.; visualization, L.S. and Z.C.; supervision, B.C.; project administration, B.C.; funding acquisition, L.S. 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 (No. 61702158), the Central Guidance on Local Science and Technology Development Fund of Hebei Province (No. 226Z1808G), the Science Foundation of Hebei Normal University (No. L2022B22), the Science Foundation of Hebei Normal University (No. L2021B38), the Guangxi Natural Science Foundation (No.2021GXNSFBA196054), the Fundamental Research Funds for the Central Universities (No.CUC23GY006) and the Humanities and the Social Science Research Project of Hebei Education Department (No. SZ2023161).

**Data Availability Statement:** Data openly available in a public repository. The two datasets of this paper came from COCO [48], which is a large image dataset designed for object detection, segmentation, and person keypoints detection.

**Acknowledgments:** The authors wish to thank the editors and reviewers for their valuable comments and helpful suggestions which have greatly improved the paper's quality.

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

## References

1. Baluja, S. Hiding images in plain sight: Deep steganography. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 2069–2079.
2. Yang, J.; Ruan, D.; Huang, J.; Kang, X.; Shi, Y.Q. An embedding cost learning framework using GAN. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 839–851. [CrossRef]
3. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, USA, 8–13 December 2014; pp. 2672–2680.
4. Hayes, J.; Danezis, G. Generating steganographic images via adversarial training. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1954–1963.
5. Tang, W.; Li, B.; Tan, S.; Barni, M.; Huang, J. CNN-based adversarial embedding for image steganography. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2074–2087. [CrossRef]
6. Liu, X.; Ma, Z.; Ma, J.; Zhang, J.; Schaefer, G.; Fang, H. Image disentanglement autoencoder for steganography without embedding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 2303–2312.
7. Islam, M.A.; Jia, S.; Bruce, N.D. How much Position Information Do Convolutional Neural Networks Encode? In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
8. Sankaranarayanan, S.; Jain, A.; Chellappa, R.; Lim, S.N. Regularizing deep networks using efficient layerwise adversarial training. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
9. Mielikainen, J. LSB matching revisited. *IEEE Signal Process. Lett.* **2006**, *13*, 285–287. [CrossRef]
10. Cox, I.J.; Kilian, J.; Leighton, F.T.; Shamoon, T. Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Process.* **1997**, *6*, 1673–1687. [CrossRef]
11. Lin, W.H.; Horng, S.J.; Kao, T.W.; Fan, P.; Lee, C.L.; Pan, Y. An efficient watermarking method based on significant difference of wavelet coefficient quantization. *IEEE Trans. Multimed.* **2008**, *10*, 746–757.
12. Luo, W.; Huang, F.; Huang, J. Edge adaptive image steganography based on LSB matching revisited. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 201–214.
13. Holub, V.; Fridrich, J.; Denemark, T. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Inf. Secur.* **2014**, *2014*, 1. [CrossRef]
14. 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.
15. Pevny, T.; Bas, P.; Fridrich, J. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 215–224. [CrossRef]

16. Ruanaidh, J.; Dowling, W.; Boland, F.M. Phase watermarking of digital images. In Proceedings of the 3rd IEEE International Conference on Image Processing, Cairo, Egypt, 7–10 November 1996; Volume 3, pp. 239–242.
17. Liu, Q. An approach to detecting JPEG down-recompression and seam carving forgery under recompression anti-forensics. *Pattern Recognit.* **2017**, *65*, 35–46. [CrossRef]
18. Ma, Y.; Luo, X.; Li, X.; Bao, Z.; Zhang, Y. Selection of Rich Model Steganalysis Features Based on Decision Rough Set-Positive Region Reduction. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 336–350. [CrossRef]
19. Ye, J.; Ni, J.; Yi, Y. Deep learning hierarchical representations for image steganalysis. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2545–2557. [CrossRef]
20. Wang, L.; Xu, Y.; Zhai, L.; Ren, Y.; Du, B. A posterior evaluation algorithm of steganalysis accuracy inspired by residual co-occurrence probability. *Pattern Recognit.* **2019**, *87*, 106–117. [CrossRef]
21. Qian, Y.; Dong, J.; Wang, W.; Tan, T. Deep learning for steganalysis via convolutional neural networks. In Proceedings of the Media Watermarking, Security, and Forensics San Francisco, CA, USA, 9–11 February 2015; International Society for Optics and Photonics: Bellingham, WA, USA, 2015; Volume 9409, p. 94090J.
22. Xu, G.; Wu, H.Z.; Shi, Y.Q. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Process. Lett.* **2016**, *23*, 708–712. [CrossRef]
23. Husien, S.; Badi, H. Artificial neural network for steganography. *Neural Comput. Appl.* **2015**, *26*, 111–116. [CrossRef]
24. Brandao, A.S.; Jorge, D.C. Artificial neural networks applied to image steganography. *IEEE Lat. Am. Trans.* **2016**, *14*, 1361–1366. [CrossRef]
25. Boroumand, M.; Chen, M.; Fridrich, J. Deep residual network for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1181–1193. [CrossRef]
26. Zeng, J.; Tan, S.; Li, B.; Huang, J. Large-scale JPEG image steganalysis using hybrid deep-learning framework. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 1200–1214. [CrossRef]
27. Zhang, Y.; Zhang, W.; Chen, K.; Liu, J.; Liu, Y.; Yu, N. Adversarial examples against deep neural network based steganalysis. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security, Innsbruck, Austria, 20–22 June 2018; pp. 67–72.
28. 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; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
29. Hu, R.; Xiang, S. Cover-lossless robust image watermarking against geometric deformations. *IEEE Trans. Image Process.* **2020**, *30*, 318–331. [CrossRef]
30. 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.
31. Sua, J.N.; Lim, S.Y.; Yulius, M.H.; Su, X.; Yapp, E.K.Y.; Le, N.Q.K.; Yeh, H.Y.; Chua, M.C.H. Incorporating convolutional neural networks and sequence graph transform for identifying multilabel protein Lysine PTM sites. *Chemom. Intell. Lab. Syst.* **2020**, *206*, 104171. [CrossRef]
32. Pugliese, R.; Regondi, S.; Marini, R. Machine learning-based approach: Global trends, research directions, and regulatory standpoints. *Data Sci. Manag.* **2021**, *4*, 19–29. [CrossRef]
33. Kha, Q.H.; Ho, Q.T.; Le, N.Q.K. Identifying SNARE Proteins Using an Alignment-Free Method Based on Multiscan Convolutional Neural Network and PSSM Profiles. *J. Chem. Inf. Model.* **2022**, *62*, 4820–4826. [CrossRef]
34. Lu, S.P.; Wang, R.; Zhong, T.; Rosin, P.L. Large-capacity image steganography based on invertible neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10816–10825.
35. Mahdy, A. A numerical method for solving the nonlinear equations of Emden-Fowler models. *J. Ocean. Eng. Sci.* **2022**. [CrossRef]
36. Ray, B.; Mukhopadhyay, S.; Hossain, S.; Ghosal, S.K.; Sarkar, R. Image steganography using deep learning based edge detection. *Multimed. Tools Appl.* **2021**, *80*, 33475–33503. [CrossRef]
37. Mahdy, A.M. Stability, existence, and uniqueness for solving fractional glioblastoma multiforme using a Caputo–Fabrizio derivative. *Math. Methods Appl. Sci.* **2023**. [CrossRef]
38. Xu, Y.; Mou, C.; Hu, Y.; Xie, J.; Zhang, J. Robust invertible image steganography. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7875–7884.
39. Bengio, Y.; Lee, D.H.; Bornschein, J.; Mesnard, T.; Lin, Z. Towards biologically plausible deep learning. *arXiv* **2015**, arXiv:1502.04156.
40. Mostafa, H.; Ramesh, V.; Cauwenberghs, G. Deep supervised learning using local errors. *Front. Neurosci.* **2018**, *12*, 608. [CrossRef]
41. Lillicrap, T.P.; Cownden, D.; Tweed, D.B.; Akerman, C.J. Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.* **2016**, *7*, 13276. [CrossRef]
42. Shaham, U.; Yamada, Y.; Negahban, S. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing* **2018**, *307*, 195–204. [CrossRef]
43. Dong, X.; Chen, S.; Pan, S. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4857–4867.

44. Gao, Y.; Ma, J.; Zhao, M.; Liu, W.; Yuille, A.L. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 3205–3214.
45. Pan, J.; Zi, Y.; Chen, J.; Zhou, Z.; Wang, B. LiftingNet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification. *IEEE Trans. Ind. Electron.* **2017**, *65*, 4973–4982. [CrossRef]
46. Belilovsky, E.; Eickenberg, M.; Oyallon, E. Greedy Layerwise Learning Can Scale To ImageNet. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 583–593.
47. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
48. Chen, X.; Fang, H.; Lin, T.Y.; Vedantam, R.; Gupta, S.; Dollár, P.; Zitnick, C.L. Microsoft COCO captions: Data collection and evaluation server. *arXiv* **2015**, arXiv:1504.00325.
49. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
50. Hore, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
51. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
52. Lerch-Hostalot, D.; Megías, D. Unsupervised steganalysis based on artificial training sets. *Eng. Appl. Artif. Intell.* **2016**, *50*, 45–59. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.