

Article

Convolutional Neural Network-Based Digital Image Watermarking Adaptive to the Resolution of Image and Watermark

Jae-Eun Lee ^{†,‡}, Young-Ho Seo [‡] and Dong-Wook Kim ^{*}

Department of Electronic Materials Engineering, Kwangwoon University, Seoul 01897, Korea; jelee@kw.ac.kr (J.-E.L.); yhseo@kw.ac.kr (Y.-H.S)

* Correspondence: dwkim@kw.ac.kr

† Current address: 913 Chambit Hall, Kwangwoon-ro 20, Nowon-gu, Seoul 01897, Korea.

‡ These authors contributed equally to this work.

Received: 24 August 2020; Accepted: 24 September 2020; Published: 29 September 2020



Abstract: Digital watermarking has been widely studied as a method of protecting the intellectual property rights of digital images, which are high value-added contents. Recently, studies implementing these techniques with neural networks have been conducted. This paper also proposes a neural network to perform a robust, invisible blind watermarking for digital images. It is a convolutional neural network (CNN)-based scheme that consists of pre-processing networks for both host image and watermark, a watermark embedding network, an attack simulation for training, and a watermark extraction network to extract watermark whenever necessary. It has three peculiarities for the application aspect: The first is the host image resolution's adaptability. This is to apply the proposed method to any resolution of the host image and is performed by composing the network without using any resolution-dependent layer or component. The second peculiarity is the adaptability of the watermark information. This is to provide usability of any user-defined watermark data. It is conducted by using random binary data as the watermark and is changed each iteration during training. The last peculiarity is the controllability of the trade-off relationship between watermark invisibility and robustness against attacks, which provides applicability for different applications requiring different invisibility and robustness. For this, a strength scaling factor for watermark information is applied. Besides, it has the following structural or in-training peculiarities. First, the proposed network is as simple as the most profound path consists of only 13 CNN layers, which is through the pre-processing network, embedding network, and extraction network. The second is that it maintains the host's resolution by increasing the resolution of a watermark in the watermark pre-processing network, which is to increase the invisibility of the watermark. Also, the average pooling is used in the watermark pre-processing network to properly combine the binary value of the watermark data with the host image, and it also increases the invisibility of the watermark. Finally, as the loss function, the extractor uses mean absolute error (MAE), while the embedding network uses mean square error (MSE). Because the extracted watermark information consists of binary values, the MAE between the extracted watermark and the original one is more suitable for balanced training between the embedder and the extractor. The proposed network's performance is confirmed through training and evaluation that the proposed method has high invisibility for the watermark (WM) and high robustness against various pixel-value change attacks and geometric attacks. Each of the three peculiarities of this scheme is shown to work well with the experimental results. Besides, it is exhibited that the proposed scheme shows good performance compared to the previous methods.

Keywords: digital watermark; neural networks; invisibility; robustness; digital images

1. Introduction

With the general use of digital data and the widespread use of the Internet, there are frequent acts of infringement of intellectual property rights, such as illegal use, copying, and digital content theft. Digital images are high value-added contents such that their intellectual property rights must be protected. A recent technique for this is digital watermarking [1].

Watermarking embeds the owner's information (watermark (WM)) into the content, and the result is stored or distributed. This technology is to claim ownership by extracting the embedded WM information when it is necessary. Various technologies have been researched/developed according to the occupied technologies, application field, etc. Until recently, the methods have been proposed to perform WM embedding algorithmically, extract the WM algorithmically according to the embedding process, or modify it [2–8]. For invisibility, a typical method embeds the WM in a discrete cosine transform (DCT) domain [2], a discrete wavelet transform (DWT) domain [3,4], a discrete Fourier transform (DFT) domain [5], or a quantization index modulation (QIM) domain [6–8].

In general, watermarking might suffer from a malicious attack intended to damage or remove the embedded WM information or a non-malicious attack by inevitable processes to store or distribute the content. Therefore, WM embedding can be performed algorithmically or deterministically. However, WM extraction has a different situation. Because of the malicious or non-malicious attacks, the WM embedded host data is damaged that the embedded WM data may also be damaged. Therefore, it may not be appropriate to extract the WM algorithmically or deterministically, and a more statistical scheme might show better performance.

With this and other reasons, recent studies have been tending to perform watermarking with a neural network (NN) [9–15], which are explained separately in Section 2. The purpose of them is the same as protecting intellectual property rights or ownership. In this technique, the embedder must embed a WM for the extractor to easily extract the WM with high invisibility; the extractor must extract the WM by accurately analyzing the host image's characteristics and the embedded WM. With deep learning, this relationship can be organized into a loss function that is used in back-propagation. Usually, it is performed by separating the embedding network and the extraction processes.

In this paper, we investigate a NN to perform invisible watermarking that hides the insertion of WM information for digital image content as much as possible, robust watermarking that loses WM information as small as possible despite malicious and non-malicious attacks, and blind watermarking that does not use original content information when extracting WM information. The structure uses a convolutional neural network (CNN) and is implemented as simply as possible by incorporating minimum CNN layers. It consists of pre-processing networks for both the host image and the WM, a WM embedding network, an attack simulation for robustness training, and a WM extraction network. In the WM pre-processing network, the resolution of the WM data increases to that of the host image for the embedding network to maintain the host image's resolution during the process. This is to retain the amount of information of the host image to increase the watermarked image quality. Also, this network uses average pooling for each layer. This is to smoothen the discrete characteristics of the binary WM values to combine with the host image smoothly to increase the WM invisibility. In training, mean absolute error (MAE) between the extractor WM and the original WM is used, while the embedder uses mean square error (MSE) as its loss function. This is because MAE is more suitable for discrete values than MSE. It also helps train the extractor network and balance the losses for the two networks more efficiently. The proposed NN is adaptive to the watermark information for a user to use his watermark information without any further training conducted by training the NN with random patterns as the watermark. It also has the adaptability to the host image's resolution to apply any the host image resolution by not including any resolution-dependent layer or component in the NN. This network can also control the WM's invisibility and the robustness against attacks, which have trade-off relationship by incorporating a strength scaling factor for the WM information as a hyper-parameter inside the NN.

This paper's composition is as follows: Section 2 introduces the relevant previous studies; the proposed network structure is explained in Section 3; Section 4 discusses the training technique and the experimental results, and this paper is concluded in Section 5.

2. Analysis of Previous Methods

NN-based watermarking schemes have been proposed [9–15], and their characteristics are described in this section, which is summarized in Table 1 except [9] because it is different from other schemes in that it is a non-blind scheme and only a part of embedder was implemented by NN. The characteristics in Table 1 include the domain of the data used by NNs (Domain), whether the method has a restriction on the resolution of the host image (Host image resolution adaptability), whether it is limited by a specific WM data (WM adaptability), the characteristics of the embedding and extractor networks (Embedding network and extractor network), attack simulation, attacks included in a mini-batch in the attack simulation process, training characteristics, and invisibility–robustness controllability.

First, we briefly explain the method by H. Kandi [9]. It is the first method to propose a digital watermarking method using deep learning. This method uses a codebook scheme that a codebook is generated in the WM embedding process, and it is used in the WM extraction process. That means it is a non-blind scheme. It uses the normalized original host image (Posimg) and its inverted image (Negimg) consisting of the pixels by subtracting the normalized pixel values from '1'. Posimg and Negimg are processed to reduce the resolution, and then the results are up-sampled to the original resolution to form the positive codebook and negative codebook, respectively. It uses a binary WM data, and the watermarked image is formed by taking the corresponding pixel group from the positive codebook when the WM bit is '1' and from the negative codebook by changing each pixel value by subtracting the pixel value of the negative codebook from '1' when the WM bit is '0'. NN is used only the process to reduce the image resolution for both Posimg and Negimg. It is the encoder type of an autoencoder (AE) because the resolution must be reduced. The extraction process consists of determining the embedded WM value by taking the corresponding pixel group and calculating which images their values are close to, the normalized watermarked, and attacked the host image or its inverted one. It experimented only for two images, Lena and Mandrill, for various pixel-value change attacks and geometric attacks, and it showed various metrics for invisibility and robustness.

Since the study of [9], most afterward works are blind watermarking methods. J. Zhu proposed a method named 'HiDDeN', which consists of a WM embedding network, noise layer (similar to the attack simulation in our scheme), WM extracting network, and an adversary network. The adversary network is for the steganographic process, which is an additional function. However, it is used for watermarking function, too. The adversary network's loss function is an adversarial loss, while the embedding and extraction networks use L2 norm loss. The watermarking process's loss function is the scaled combination of the three, but the adversary network uses only the adversarial loss. All the layers except the final layer of the extractor network consist of CONV (convolution)-BN (batch normalization)-ReLU (rectified linear activation) combination with mostly 64 channels. The WM data is re-structured to 1-dimensional data, and the result is replicated as many as the resolution of the host image, which is to affect the WM data to the whole host image. The replicated WM data is concatenated to the host image to enter to the WM embedding network. The WM embedding network's result enters both the WM extraction network through the noise layer and the adversary network. The extraction network reduces the resolution and finally processes with a global pooling and a FC (fully connected) layer, which means it is dependent on the host image resolution. It uses random WM data, but specific attacks and their strengths are used in training. One more thing to note is that it proposed a scheme to make the JPEG compression differentiable with two schemes, JPEG mask, and JPEG drop.

M. Ahmadi proposed a scheme to use the DCT frequency domain by implementing and training a network to perform a DCT separately [11]. The network consists of a WM embedding network, attack simulation, and WM extraction network. Before entering the network, the host image is reduced

to the resolution of WM and DCTed by a DCT network that has been constructed and trained already. The WM data is multiplied by a scaling factor and concatenated to the DCTed host image data, the result of which is processed in the WM embedding network that consists of convolution layers with ELU (exponential linear unit) activation. The middle three layers of the embedding network perform the circular convolution for a global convolution. It increases the resolution to that of the original host image, and finally, an inverse DCT layer is processed to form the watermarked image. The WM extraction network also includes the DCT layer and the inverse DCT layer used in the embedding network and several convolution layers, including the circular convolution layers. It reduces the resolution to that of the WM data. It also used specific kinds and strengths of attacks in training with only one kind of attack in a mini-batch, by which it cannot guarantee the performance for other WM information. The SSIM value is used as the loss of the WM embedding network and the cross-entropy for the extraction network. For the cost function of the training, a trade-off combination of the two values by multiplying A and $1-A$, respectively, is used, where A is determined empirically. It also proposed a scheme to use the DCT within the network.

S. M. Mun proposed a watermarking method with an AE-structured NN, consisting of residual blocks [12]. All residual blocks are composed of a unit that performs ReLU after adding CONV(1×1)-ReLU-CONV(3×3)-ReLU-CONV(1×1) and CONV(1×1). For the embedding, the host image is reduced to the WM's resolution by the AE encoding process. To each of the resulting layers, the WM information is added to form the embedding AE's encoded data, which is entered into the decoder of the embedding AE. This decoder increases the resolution to that of the host image and reduces the resulting number of images to the original host image's channels. The watermarked image is output by accumulating the WM information multiplied by different strength factors to the host image several times. The extractor has the AE encoding structure only. It does not use any pooling layer in the network. The attack simulation maintains a constant distribution for all the specified attacks in each mini-batch. It also uses only specific WM information, even though it includes the inverted WM information, to avoid overfitting WM.

X. Zhong proposed a scheme to replace the attack simulation with a Frobenius norm [13]. Each of its embedder and extractor networks consists of four connected function networks, each other and one layer (invariance layer) connect the two networks. Thus, each pair network forms a loss function, and the final cost function for training is constructed with the linear combination of the four by determining the four-loss functions by determining the scaling factors empirically. The host image is fixed to 128×128 color image, but the binary random data is used as the WM data. Each network consists of convolutional layers, but the invariance layer connects the embedder network, and the extractor network consists of a sparse FC layer with tanh activation. The WM information is up-sampled (network μ) to the host image's resolution, and the result is concatenated to the host image to enter into the embedding network. The embedder reduces the resolution to that of the WM data (network γ), increases it to the host image (network σ), and additionally processes it without changing resolution (network ϕ). Both embedder and extractor consist of conventional CNN layers.

Bingyang used the same network structure as J. Zhu [10] but incorporated an adaptive attack simulation such that it selects more the attacks for which the network shows weak robustness [14]. The method to consider the weak robustness is to include the extractor loss for the worst-case attack result. However, in a mini-batch, only one kind of attack with different strengths is included. It processes in the spatial domain and uses the FC layer to extract the WM. For WM embedding, it uses an adversarial loss, not only for the steganography layer, for training with using random patterns as the WM data.

Y. Liu proposed a two-stage training scheme TSDL, two-stage separable deep learning), in which the entire NN with adversary network is trained without any attack (FEAT, free end-to-end adversary training), at first, then in the next train only the extractor without the adversary network is re-trained (ADOT, noise-aware decoder-only training) by adding the attack simulation [15]. In this scheme, the duplicated binary (here, 1, and -1) to the resolution of the host image is concatenated in each

convolution layer in the embedder network except the last two layers performing 3×3 convolution and 1×1 convolution, in order. WM extraction network and the adversary network also consist of 1×1 and 3×3 convolution layers. The final layer of the extractor network consists of a FC layer after average pooling and ReLU activation. The loss function for the embedder network and the extractor network are MSE (mean square error) loss, but the adversary network uses an adversarial loss. For each of the two training processes, the appropriate combinations of the loss functions are used by multiplying scaling factors determined empirically. In training, it included only one kind of attack that is included in a mini-batch. It also used specific kinds and strengths of attacks in training and showed the test results for only the kinds and strengths of attacks used in training.

Most of the methods are blind schemes [10–15], use spatial domain data [9–15], and target a specific resolution of host images and WM information [9–15]. Through analysis of previous studies that performed watermarking using deep learning, we drew three as follows.

- When a specific WM is used for training, additional training is essential whenever one wants to use a new WM [9,10,12].
- Using FC layer(s) restricts the applicable resolution of the host images [10,14,15]. These methods cannot guarantee their usefulness in general applications with other host image resolution.
- Some methods did not realize the controllability of the tradeoff relationship between invisibility and robustness [9,10,13,14]. Because they cannot provide options for the user’s preference, their practicality may be restricted.

Therefore, for the three problems we have raised, we propose a new neural network structure with three goals, which are resolution adaptability of host images, content adaptability of watermark, and controllability of invisibility and robustness, for deep learning as follows.

- Resolution adaptability of host image: Deep learning network capable of embedding watermarks in host images of all resolutions.
- Content adaptability of watermark: Deep learning network that can change the content of the watermark to be inserted without re-training.
- Controllability of invisibility and robustness: Deep learning network with controllable visual visibility and watermarking intensity.

Table 1. Characteristics of state-of-the-art networks and ours.

Items	[10] HiDDeN	[11] ReDMark	[12]	[13]	[14] ROMark	[15]	Ours
Domain	spatial	frequency	spatial	spatial	spatial	spatial	spatial
Host resolution adaptability	specific (FC layer)	no result	no result	no result	specific (FC layer)	specific (FC layer)	general
Host resolution adaptability	specific	general	specific	general	general	general	general
Embedding network	adversarial loss	circular convolution and DCT layer	AE using residual block	CNN	adversarial loss	adversarial loss	Simple CNN and average pooling
Extractor network	global pooling and FC layer	DCT layer	residual block	CNN	global pooling and FC layer	FC layer	Simple CNN
Embedder simulation	specific attack and strength	specific attack and strength	specific attack and strength	Frobenius norm	variety attack and strength	specific attack and strength	variety attack and strength
Attack per mini-batch	one kind	one kind	all kinds	-	one kind	one kind	all kinds
Training characteristic	differentiable JPEG attack	differentiable DCT layer	residual block	losses before and after embedding	differentiable JPEG attack	two-stage training	-
Invisibility–robustness controllability	×	○	○	×	×	○	○

Accordingly, we propose a blind, invisible, and robust watermarking NN that adapts to the resolution of the host image and WM information. This method uses spatial domain data and consists

of CNN layers with average pooling layers. In its attack simulation, all the attacks are included in each mini-batch, with random strength, but maintain a balanced distribution. Moreover, randomly generated data is used as the WM information in the training process that any data can be used as the WM. This method also includes a strength factor which adjusts the tradeoff relationship between invisibility and robustness.

The next section describes the proposed deep learning network architecture for watermarking to achieve our three goals.

3. Proposed Watermarking Framework

In the previous section, we derived three items that deep learning-based watermarking should overcome through analysis of previous studies and selected functional goals of the deep learning network for the three items. We intend to solve the problems presented in this section and propose a new network structure for watermarking to achieve the goals. Since the network we are proposing contains various functions, the entire network is composed of four sub-networks as follows.

- Host image pre-processing network for resolution adaptability of host images.
- WM pre-processing network for content adaptability of watermarks.
- Embedding network, Attack Simulation, Extraction network for controllability of invisibility and robustness.

Next, the detailed structure and operation of these sub-networks and a method of improving the watermarking performance of the proposed network through their combination will be described.

3.1. Overall Watermarking Scheme

Figure 1 shows the overall structure of the proposed digital watermarking scheme: (a) for the embedder, (b) for the extractor, both relatively conventional. Our scheme is designed to process only one channel that Y component after converting RGB image to YCbCr components is used. Before entering it, it is normalized to the range of $[-1, 1]$. Meanwhile, the WM data is binary data, and it is scrambled with a key. Both normalized host image data and scrambled WM data are preprocessed, and the results are concatenated. Here, the WM data is multiplied by a strength scaling factor (s), to adjust the trade-off between invisibility and robustness. The concatenated result is processed in the embedding network to output the watermarked data, which is de-normalized and converted to RGB format with the converted Cb and Cr components to form the watermarked host image.

The extraction process receives a watermarked and attacked RGB image, which is converted to YCbCr format. Only the Y component is taken and normalized to the $[-1, 1]$ ranged data processed in the extractor network. It extracts the WM information as the output, and the result is de-scrambled with the key used in the scrambling process. The de-scrambled data is the final extracted WM.

3.2. Structure of Watermarking Network to Be Trained

As mentioned before, our intentions with the proposed NN-based watermarking scheme are simplicity in the network structure and depth, host resolution adaptability, WM adaptability, and controllability between invisibility and robustness of WM. Also, to increase the quality of the watermarked image, WM invisibility, and balanced training, we use several techniques such as maintaining the host image's resolution, average pooling for processing the WM data, MAE loss for the extractor network. Those are more focused in the following explanations.

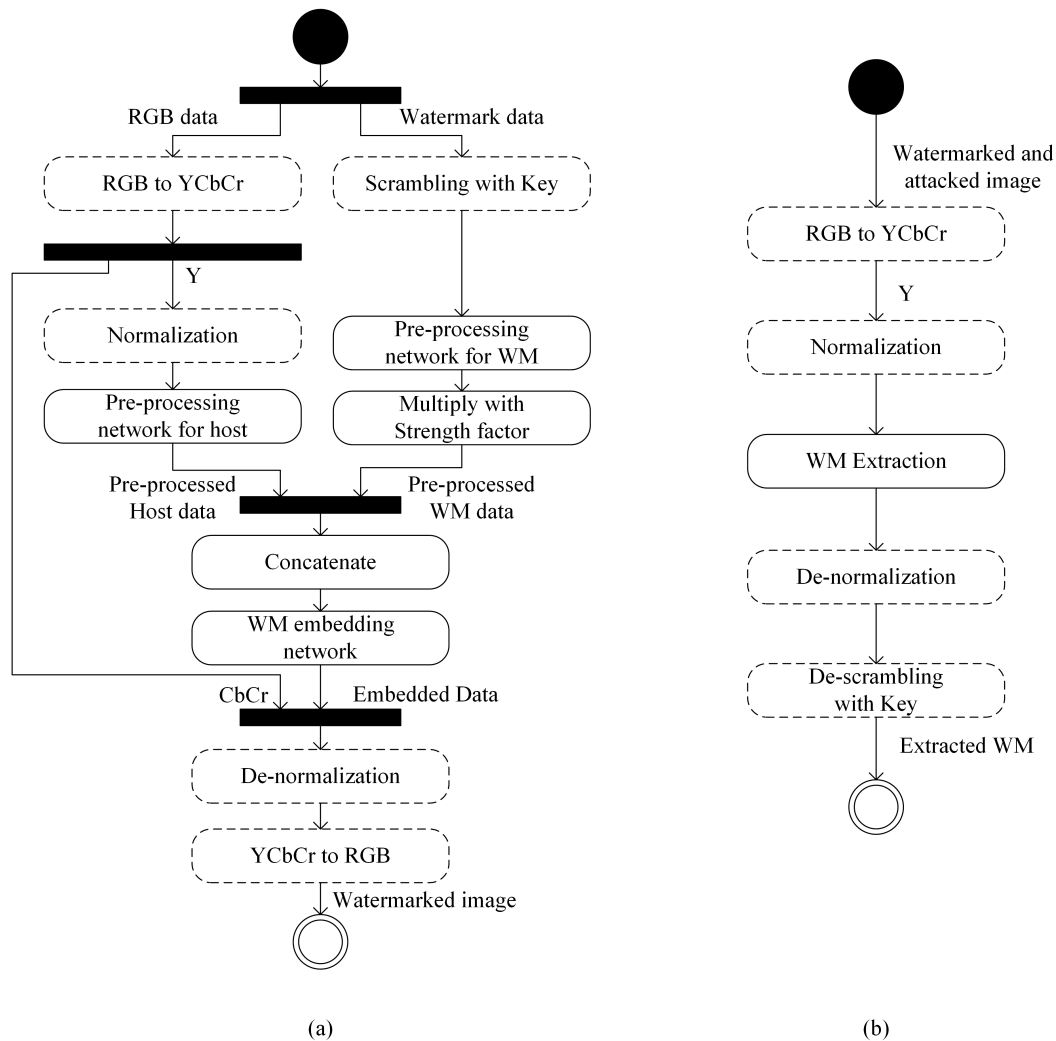


Figure 1. Proposed digital watermarking system: (a) watermark embedding, and (b) watermark extraction.

Among the functional blocks in Figure 1, only the solid-lined ones are implemented in the network; those in the dotted blocks are processed in off-the-network. This network structure is shown in Figure 2, which includes pre-processing networks for host and WM, WM embedding network, and WM extraction network. Here, the attack simulation process is included for training. This network’s first structural feature is that it consists of only the simple CNNs with a relatively shallow depth that the highest depth has only 13 CNNs. Each of the consisting networks and their components was determined empirically based on the tremendous experiments’ data. The second feature is that the WM pre-processing network increases the resolution to that of the host image, while most previous works reduce the host image’s resolution to that of the WM. This is to maintain the host image’s information to increase the invisibility of the WM, which is based on our experimental results that it is more challenging to obtain invisibility performance than robustness, and the scheme maintaining the host resolution was superior in invisibility with the same robustness.

Other features of our network are dealt with in detail in the following sub-sections to explain each network.

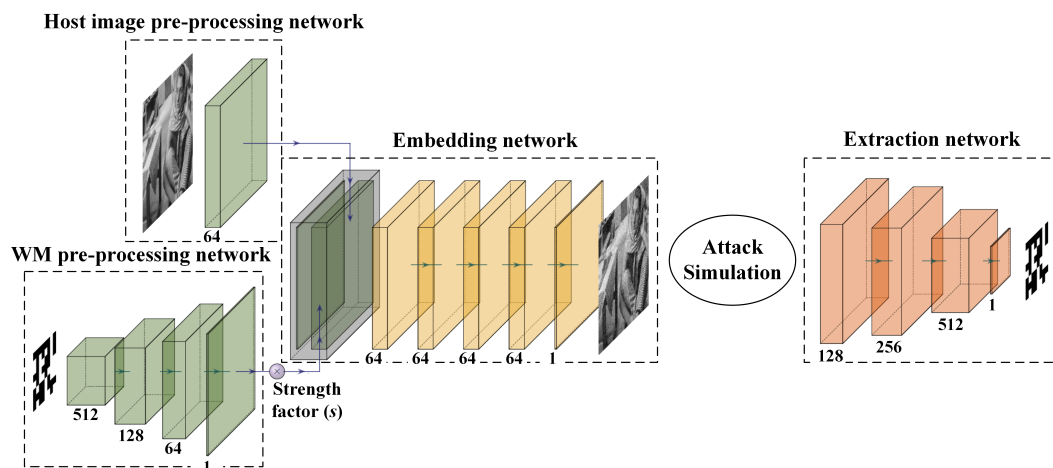


Figure 2. Structure of proposed network for training.

3.2.1. Pre-Processing Network for Host Image

First, the host image pre-processing network maintains the original image's resolution and is composed of one convolutional layer (CL) with 64 filters whose strides are the same as 1. Since the embedding network's output should be similar to the host image, the input image, this network should not damage the host significantly. Thus, it consists of only one CL with 3×3 filters. Nevertheless, it uses 64 filters (it means 64 channels are produced) to extract as many characteristics of the host image as possible.

3.2.2. Pre-Processing Network for WM

The WM pre-processing network is configured to gradually increase the resolution to match the host image pre-processing network's resolution. This is to increase the WM invisibility, as explained before. It has been confirmed by our experiments that the case maintaining the resolution of the host image in WM embedding has high WM invisibility than the case reducing the resolution to that of the WM and increasing the resolution to output the watermarked image. This network includes four network blocks: the first three consist of the CL, batch normalization (BN), activation function (AF), an average pooling (AP), but the last block consists of CL and AP. All CLs have a 0.5 stride for up-sampling. The corresponding number of filters is 512, 256, 128, and 1, respectively. AF is the rectified linear unit (ReLU), and AP is a 2×2 filter with a stride of 1. AP is used because WM is a binary data that the values are discrete, but the host image data is real and continuous; it is necessary to smoothen the WM data with the APs to combine with the host image data retaining the continuous characteristics. It also has been confirmed with our experiments. The WM pre-processing network output is multiplied by the strength scaling factor to control the invisibility and robustness of the WM.

3.2.3. WM Embedding Network

The WM embedding network concatenates the results of the 64 channels of the pre-processed host information and one channel of the pre-processed WM information and uses them as the input to output the watermarked image information. The network consists of CL-BN-AF (ReLU) for the front four blocks, and the last block consists of CL-AF (tanh). The tanh activation maintains the positive and negative values to meet the data range of $[-1, 1]$ to the input host information. All CL strides are set to 1 to maintain the resolution of the host image for invisibility. All blocks, except the last one, have 64 CL filters; the last block has the same number of filters as the number of channels in the host image, which is one here. Because we are aiming for invisible watermarking, we use the mean square error (MSE) between the watermarked image (I_{WMed}) and the host image (I_{host}) as a loss function (L_1) of the pre-processing network and the embedded network. This is shown in Equation (1).

$$L_1 = \frac{1}{MN} \sum_{i,j} [I_{host}(i,j) - I_{WMed}(i,j)]^2 \quad (1)$$

Here, $M \times N$ is the resolution of the host image.

3.2.4. Attack Simulation

For high robustness, the watermarked image is intentionally suffered from preset attacks in the attack simulation. It comprises seven pixel-value change attacks and 3 geometric attacks, which are considered the malicious and non-malicious attacks are occurring in the distribution process. Table 2 shows the types, strengths, and the ratio of each attack used in one mini-batch [10,11] in training. We maintain the ratios for each mini-batch, including the ones not attacked ('identity' in the table).

Table 2. Attacks for attack simulation used in training.

Attack Type	Attack	Strength	Ratio
No attack	Identity	-	1/12
Pixel-value change attack	Gaussian filtering	$3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9$	2/12
	Average filtering	$3 \times 3, 5 \times 5$	2/12
	Median filtering	$3 \times 3, 5 \times 5$	1/12
	Salt & Pepper	$p = 0.1$	0.5/12
	Gaussian noise	Sigma = 0.1	0.5/12
	Sharpening	5-point stencil, 9-point stencil	1/12
	JPEG	Quality factor = 50	1/12
Geometric attack	Rotation	$0 \sim 90^\circ$ (random)	1/12
	Crop	$0.5 \sim 0.8$ (random)	1/12
	Dropout	$0.3 \sim 0.9$ (random)	1/12

3.2.5. WM Extractor Network

The extraction network is a reversely symmetrical structure to the WM pre-processing network except for the number of filters used. It reduces the resolution of the watermarked and attacked image and extracts the WM information. It consists of three CL-BN-AF (ReLU) blocks and one CL-AF (tanh) block, that is the last block. We set the stride of all CLs to 2 for down-sampling. The number of filters used in the CLs is 128, 256, 512, and 1, respectively. This network uses mean absolute error (MAE) between the extracted WM (WM_{ext}) and the original WM (WM_o) as a loss function (L_2). The reason why MAE is used for the extraction network is that the WM information consists of binary values (compare to Equation (1) that uses MSE for the host image information). It is determined empirically based on the data from lots of experiments. This is shown in Equation (2).

$$L_2 = \frac{1}{XY} \sum_{i,j} |WM_o(i,j) - WM_{ext}(i,j)| \quad (2)$$

Here, $X \times Y$ is the resolution of the WM information.

3.2.6. Loss Function of the Network for Training

With the two-loss terms of Equations (1) and (2) for host information and WM information, respectively, the loss function of the whole network for training is constructed as Equations (3) and (4) for WM embedding and WM extraction, respectively.

$$L_{emb} = \lambda_1 L_1 + \lambda_2 L_2 \quad (3)$$

$$L_{ext} = \lambda_3 L_2 \quad (4)$$

In these two equations, λ_1 , λ_2 , and λ_3 are set to the hyper-parameters that control invisibility and robustness. λ_1 represents the strength of the L_1 loss applied to the embedding network, λ_2 represents the strength of the L_2 loss applied to the embedding network, and λ_3 is the strength of the L_2 loss applied to the extraction network. Because the L_1 loss and the L_2 loss have different properties, it is not easy to analytically determine the three hyper-parameters. Therefore, they are obtained empirically.

4. Experimental Results and Discussion

Qualitative and quantitative evaluations from various experiments were performed to evaluate the invisibility and robustness of the proposed digital watermarking scheme. First, the dataset used and the environment set for implementation are described, and the measurement method for quantitative evaluation is described. The invisibility, robustness, WM adaptability, host image adaptability, and controllability of the invisibility and robustness are then verified. Finally, the results are compared with the state-of-the-art methods.

4.1. Dataset

4.1.1. Host Image

We used the BOSS dataset [16], which consists of 10,000 grayscale images with 512×512 resolution, as the training dataset. The first reason that we chose the BOSS dataset is that it is used broadly in deep learning for various applications and techniques. Also, it contains only gray images that are more convenient to use in our network because it uses only the Y component, although Figure 1 and its explanation assumed more available RGB color images. Besides, a standard dataset [17], which has 49 grayscale images with 512×512 resolution and is used broadly as the evaluation dataset, was used as the evaluation dataset. We have down-sampled images in both datasets to 128×128 resolution to use as the host images.

4.1.2. Watermark

Binary images having a resolution of 8×8 was used as the WM. A random WM was generated for each iteration in the training process, and it was scrambled with a corresponding key. These randomly generated WMs are to adapt the network to any WM information. Also, they reduce overfitting in the training process.

4.2. Training

The proposed watermarking network was trained in a PC with an Intel (R) Core (TM) i7-9700 CPU @ 3.00 GHz, 64 GB RAM, and the RTX 2080ti GPU. The hyper-parameters and their values used in training are listed in Table 3, set empirically. A mini-batch includes 100 host images, and a newly generated random-pattern WM data was used for each mini-batch. The training was continued until the loss value is stable, which was 4000 epochs. It used Adam optimizer [18] with learning rates 0.0001 and 0.00001 for the embedding network and the extraction network, respectively. During the training, the strength factor was set to 1, and the weight decay rate was 0.01.

The values of the λ s were set by separate experiments after determining the other parameters. The finally determined λ s values were 45, 0.2, and 20 for λ_1 , λ_2 , and λ_3 , respectively. The values of λ_1 and λ_2 are to balance the invisibility and robustness for embedder, while the value of λ_3 is to balance the training speed of the embedder and the extractor with the weight decay rate. All three values are correlated that we have experimented for the large ranges of values for them.

With the hyper-parameters in Table 3, the training took about four days with the BOSS dataset.

Table 3. Hyper-parameters used in training.

Hyper-Parameter	Value
Hyper-parameter	Value
Epoch	4000
Mini-batch	100
Optimization	Adam
λ_1	45
λ_2	0.2
λ_3	20
Embedding network learning rate	0.0001
Extraction network learning rate	0.00001
s	1

4.3. Performance Assessment Metrics

For the quantitative evaluation of invisibility, the peak-signal-to-noise-ratio (PSNR) of Equation (5) has been used primarily in the previous works, and it is thus used in this study, too. As previously mentioned, the pixel value of the WM embedder's output is ranged to $[-1, 1]$ because of normalization. It is converted to an integer in the range of $[0, 255]$ as a final watermarked image used for the invisibility evaluation.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (5)$$

Robustness is evaluated by the bit error ratio (BER) of the extracted WM information. Because the WM information comprises binary images, the original and extracted, WM information's pixel value is measured as one if they are the same, and 0 if they are different. The resulting values are averaged for the number of pixels. It is shown in Equation (6).

$$BER(\%) = 100 \frac{1}{XY} \sum_{i,j} \delta(WM_o(i,j), WM_{ext}(i,j)) \quad (6)$$

$$\delta(A, B) = \begin{cases} 0, & A \neq B \\ 1, & A = B \end{cases}$$

Besides, the WM capacity is shown in Equation (7), which is the ratio of the WM resolution to the host image resolution. In this equation, the resolution of the WM and the host image is (X, Y) and (M, N) , respectively. In this study, the WM capacity was fixed at 0.0039, without the loss of generality and practicality.

$$capacity = \frac{XY}{MN} \quad (7)$$

4.4. Results

4.4.1. Invisibility of Watermarked Image

When $s = 1$, the watermarked image's average PSNR to the original host image from the training result showed 43.23 dB. We also applied the trained weight set to the evaluation dataset, the result of which showed the PSNR range of [37.46 dB, 42.13 dB]. The average was 40.58 dB. Figure 3 shows three example pairs of the host image (a), watermarked image (b), and the 100 times magnified difference image (c) from the test dataset. They are the ones showing the lowest (1st row), middle (2nd row), and the highest (3rd row) invisibility, respectively. As you can see, it is not easy to distinguish the original image and the watermarked image with the naked eyes, even for one with the lowest invisibility. Therefore, it can be said that the invisibility of our scheme is very high.



Figure 3. Examples showing invisibility: (a) host images, (b) watermarked images, and (c) magnified difference image, peak-signal-to-noise-ratios (PSNRs) of the first, second, and third row are 37.46 dB, 40.87 dB, and 42.13 dB, respectively.

4.4.2. Robustness for Various Attacks

With the trained weight set, robustness experiments were conducted on various types and strengths of attacks on the evaluation dataset. Figure 4 shows some examples of the attacked images. The purpose of the attack is to use the image without ownership by malicious or non-malicious weakening or removing WM. However, as you can see from the figures, some attacks damage the image too much to reuse that those attacks are entirely meaningless. However, we included them to compare with previous works that included them.

The experimental robustness results are shown in Table 4 (right now, the first column of the three BER columns), in which the BER values are the average values for all the images in the evaluation dataset. Note that the values in Table 4 are when $s = 1$. As you can see in Table 4, the BER values tend to increase as the attack strength increases for each kind of attack. Note that the rotation attack disturbs the image information most at the 45 degrees. So the BER increases as the rotation angle increases, but after 45 degrees, it decreases as the angle increases more. It means that the proposed network was trained well without overfitting to a specific kind of strength.

As values in the table, most pixel-value change attacks showed high robustness as less BERs than 10% except Gaussian filtering attacks with 7×7 and higher filters, Gaussian noise attacks with σ larger than 0.08, and JPEG attack with higher compression than quality 40. Especially, it showed strong robustness for the salt-and-pepper noise addition attacks. For the geometric attacks, it is quite vital for the rotation attacks but shows high BERs against more than 50% of crop and cropout attacks and higher dropout attacks of 30%. However, those attacks for which the proposed scheme shows

higher than 10% of BER are potent attacks that damage the host image a lot. Therefore, we believe the proposed scheme would be robust enough for meaningful attacks.

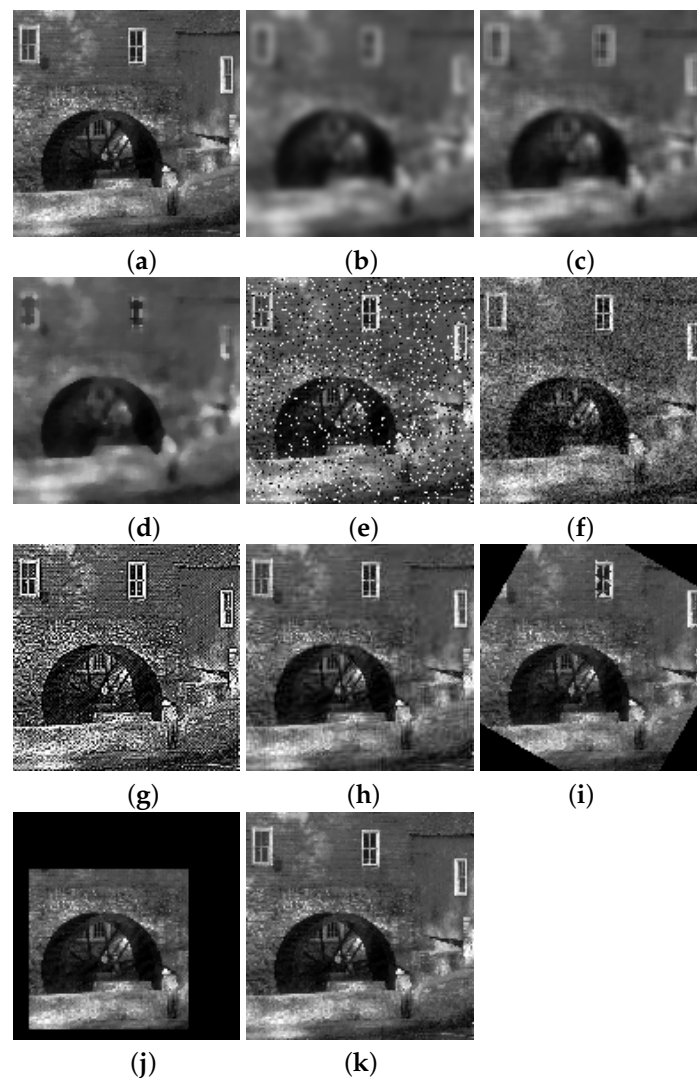




Figure 4. Examples of attacked images: (a) host image, (b) Gaussian filtering (9×9), (c) average filtering (5×5), (d) median filtering (5×5), (e) salt and pepper noise addition ($p = 0.1$), (f) Gaussian noise addition ($\sigma = 0.1$), (g) Laplacian sharpening (5-point stencil), (h) JPEG compression (quality factor = 50), (i) rotation (30°), (j) crop ($p = 0.5$), (k) dropout ($p = 0.5$).

For reference, Figure 5 shows some examples of the extracted watermarks according to their BERs. From the figures, it is quite clear that the extracted WM with a higher BER value than 10% cannot guarantee to protect the host image's intellectual property rights.

Table 4. Average bit error ratio (BER) values of extracted watermarks resulting from various attacks.

Attack Type	Attack	Strength	BER (%)		
			WM1 Random (Average)	WM2 	WM3 
	No attack	-	0.7015	0.6696	0.6696
Pixel-value change attacks	Gaussian filtering	3 × 3	1.5944	1.7538	2.0089
		5 × 5	7.5255	7.3023	8.4503
		7 × 7	11.5115	11.2883	11.5115
		9 × 9	18.463	19.1964	17.5064
	Arverage filtering	3 × 3	4.273	3.9541	4.1135
		5 × 5	5.2296	5.3571	4.7832
	Median filtering	3 × 3	8.4184	7.8763	7.8763
		5 × 5	10.5548	10.8418	11.0969
	Salt and pepper noise addition	0.01	0.861	0.7972	0.7972
		0.03	1.1798	1.0204	1.1161
		0.05	1.4031	1.2436	1.3393
		0.07	1.8176	1.5306	1.8495
		0.09	2.5829	3.1250	2.0408
	Gaussian noise addition	$\sigma = 0.01$	0.8291	0.7972	0.7334
		$\sigma = 0.03$	1.977	1.6582	1.977
		$\sigma = 0.05$	6.0906	6.8878	5.6441
		$\sigma = 0.08$	11.9898	12.8508	13.3291
	Sharpening	5-point stencil	3.2844	3.6671	3.5714
		9-point stencil	3.9222	4.6237	4.3686
	JPEG	90	0.9566	0.8610	0.7653
70		4.2411	3.9860	4.2411	
50		8.0676	7.9082	9.0561	
30		14.8916	15.1148	14.8278	
10		31.4732	31.8240	33.4184	
Rotation	15	2.0727	1.8814	1.9133	
	30	4.9107	4.8151	5.2296	
	45	5.0383	5.1339	5.7398	
	60	3.8265	3.6671	4.7194	
	75	1.7857	1.8176	1.9452	
Crop	0.9	0.7015	1.1798	0.9247	
	0.7	2.1365	4.3367	13.4566	
	0.5	14.6365	16.7411	11.4796	
	0.3	24.9681	21.3967	29.1773	
	0.1	39.6365	48.5013	38.361	
Geometric attacks	Cropout	0.1	2.2003	3.0293	1.7538
		0.3	9.0561	12.4362	9.088
		0.5	17.0281	24.2666	20.9184
		0.7	24.0434	33.4184	25.4783
		0.9	34.8533	44.9298	37.3724
Dropout	0.9	0.9247	0.9247	1.0523	
	0.7	2.4554	2.2003	2.3278	
	0.5	6.25	5.4528	5.5166	
	0.3	14.8916	15.5612	15.1148	
	0.1	34.1199	37.3087	34.7577	

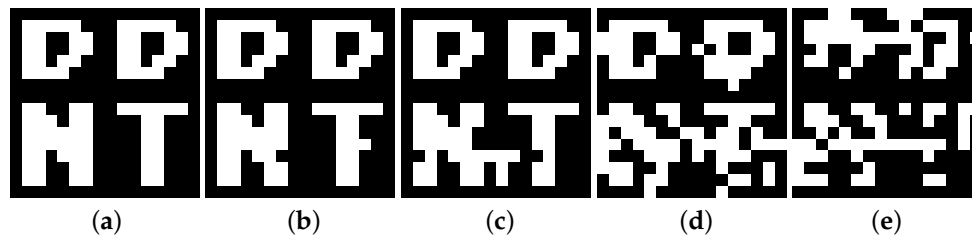


Figure 5. Examples of the extracted watermarks (WMs) according to BER: (a) 0, (b) 0.78, (c) 3.51, (d) 10.93, (e) 21.09.

4.4.3. WM Adaptability

As mentioned before, a watermarking scheme's capability to accommodate any WM data is essential because any user can use the scheme with his WM data, even though some of the previous works do not have this WM adaptability [10,12]. Our scheme makes it possible by using newly generated random data as the WM information for every mini-batch in training. We have verified this adaptability of our scheme by experimenting with various WM information. Table 4 shows two examples of the results. The one marked as 'Random (average)' means the average values for all the WM data used, while WM2 and WM3 are the two examples of the case using two specific WM data described in the table. From those columns' values, it is confirmed that our scheme applies to any WM data without losing similar robustness.

4.4.4. Host Image Adaptability

Because the proposed method does not use any layers that are dependent on the host image's resolution, such as the FC layer, it is adaptable to the resolution of the host image. The WM invisibility and the robustness against attack were evaluated by changing the host image's resolution from 64×64 to 512×512 , as shown in Table 5. Here, we fixed the WM capacity to about 0.0039. Note that the network has been trained with 128×128 host images and 8×8 WM data. As shown in Table 5, the WM invisibility increased as the host image resolution increased.

Figure 6 shows the results from robustness experiments as graphs, in which each graph includes the results for one kind of attack with the different attack strengths and the different resolution of the host images. Note that the legends in other graphs for the resolution are the same as (a). According to Figure 6, the robustness decreases as the resolution increases for the most pixel-value change attacks, except the Gaussian noise addition and the high-compression JPEG attacks. Those two attacks showed not much difference in robustness for different resolutions and did not follow the tendency. For most of the geometric attacks, the robustness tends to decrease as the resolution increases, but the dropout attack showed increasing robustness as the resolution increases.

Even in the cases that the robustness decreases as the resolution increases, the proportion was not large, or the increased BER values are not so high. That is, the reduced robustness by resolution change can still be regarded as high robustness. Therefore, we can conclude that the proposed method applies to various resolutions of the host image. Especially most pixel-value change attacks, the proposed scheme is more suitable to the high-resolution trend because mostly it increases the robustness as the resolution increases.

Table 5. Watermark resolution, host image resolution, measured invisibility.

Host Image Resolution	Watermark Resolution	Invisibility [dB]
64×64	4×4	39.97
128×128	8×8	40.58
256×256	16×16	41.23
512×512	32×32	42.35

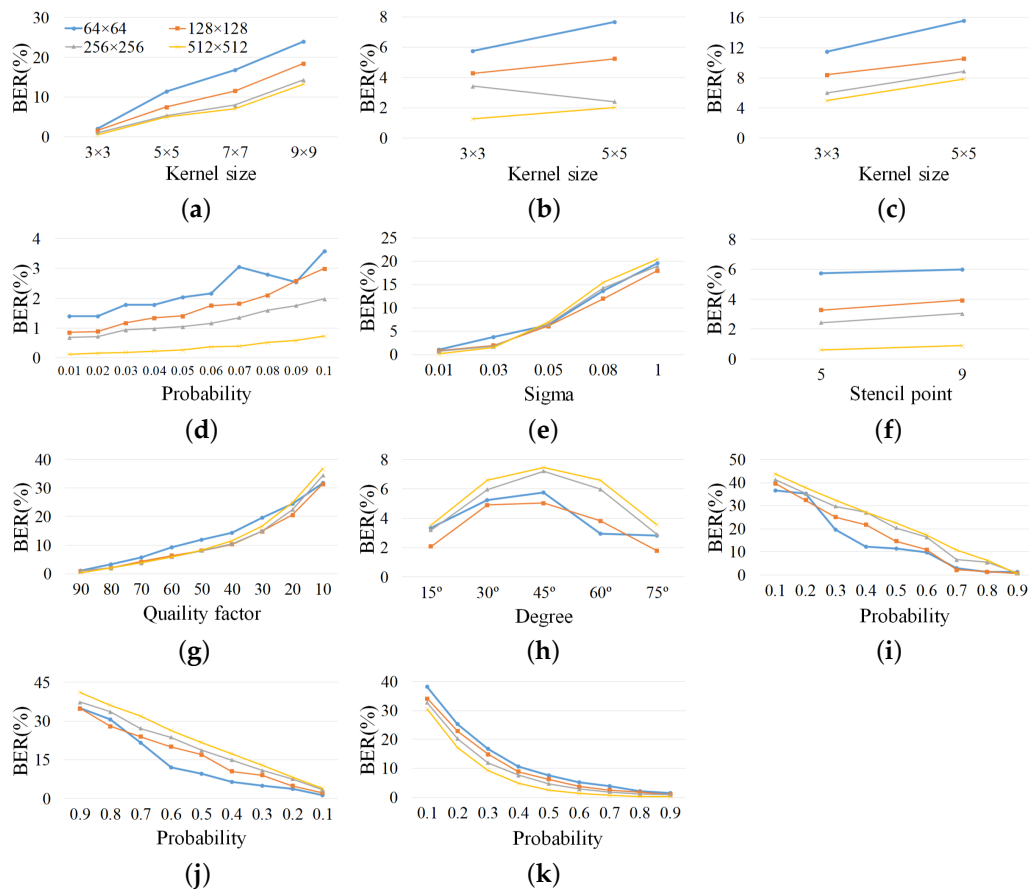


Figure 6. BER values resulting from the experiments for the various sizes of host image (the legend in (a) is applied to all the figures): (a) Gaussian Filtering, (b) average Filtering, (c) median Filtering, (d) salt and pepper noise addition, (e) Gaussian noise addition, (f) sharpening, (g) JPEG, (h) rotation, (i) crop, (j) cropout, (k) dropout.

4.4.5. Invisibility–Robustness Controllability

Invisibility–robustness controllability, which can control the complementary relationship between these two characteristics, is required according to the user’s need in using a watermarking system, a more robust scheme by sacrificing invisibility or a more invisible scheme by sacrificing robustness. In our scheme, the strength scaling factor is used to control this complementary relationship. When invisibility is more critical than robustness, s is set to a lower value. However, when higher robustness is needed, s is set to a higher value.

Controllability, invisibility, and robustness for the various attacks with increasing s from 0.5 to 2 are measured, and the results are shown in Figure 7. This figure includes the invisibility change in (a) and robustness changes for all considered kinds and strengths of the attacks and their strengths in from (b) to (m). As shown in Figure 6a, the WM invisibility decreases as s increases, as expected. For each of the attacks, the robustness increases consistently as s increases while maintaining the performance tendency for the change in the attack’s strength. This shows that the proposed scheme has the firm capability to control the trade-off relationship between invisibility and robustness.

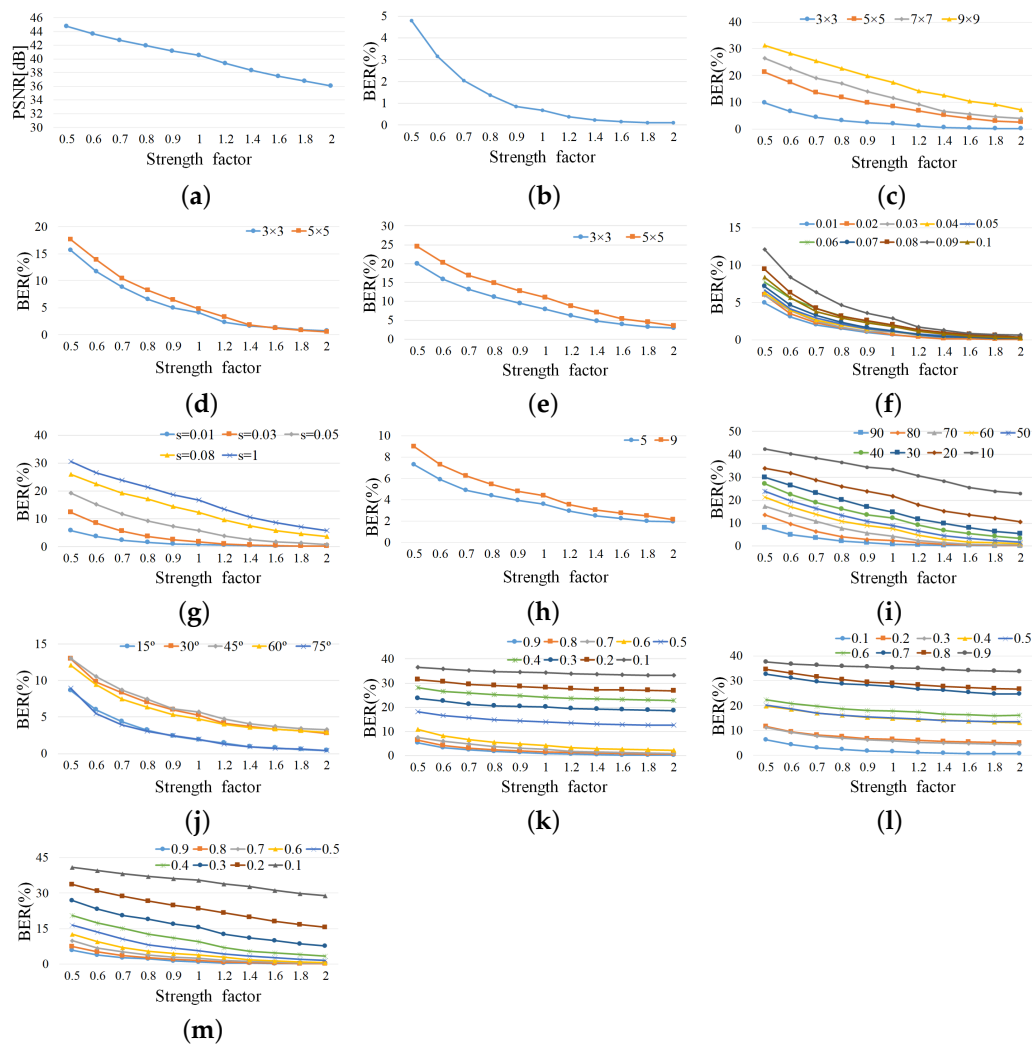


Figure 7. Invisibility and robustness according to a change in the strength factor: (a) Invisibility, (b) No attack, (c) Gaussian Filtering, (d) average Filtering, (e) median Filtering, (f) salt and pepper noise addition, (g) Gaussian noise addition, (h) sharpening, (i) JPEG, (j) rotation, (k) crop, (l) cropout, (m) dropout.

4.5. Comparison with State-of-the-Arts Methods

The performances of the proposed scheme are compared with the state-of-the-art methods (HiDDeN [10], ReDMark [11], and [15]). For a fair comparison, we adjusted s to fit the PSNR similar to the other methods. Because ReDMark [11] showed precise numerical results, we first compare it with ours separately for various attacks. We adjusted the PSNR to 40.58 dB by adjusting $s = 1$. The results are shown in Table 6. The kinds and the strengths of attacks are from [11]. From the values in this table, it is clear that the proposed method performs better for all attacks except the Gaussian noise addition attacks.

The other state-of-the-art methods did not show the clear numerical data. Therefore, we use the data presented in [15], which is the result of comparing [15] with [10] and [11], for a specific set of attacks. The comparison results with used training and test dataset are shown in Table 7. For this comparison, s was set to 2.75 for our scheme, to adjust the PSNR to 33.5 dB. As a result, the proposed method showed excellent results, except for the crop (0.035) attack, compared with [10] and [11]. However, when compared with [15], our method only showed better results for the JPEG compression attack.

The crop (0.035) attack uses only 3.5% of the watermarked image to extract the WM information, which is unrealistic because 3.5% of the image is not useful. Also, the method of [15] used the same kinds and the same strengths of attacks in training. That means the only trained attacks were evaluated. Therefore, it cannot guarantee the result for other kind or other strength of attack that it cannot be said to have good robustness results in a real application. Our scheme shows high utility from the comparisons because it demonstrates exemplary performance in all attacks, except the Gaussian noise addition attack and high-strength crop attack, which also result in valueless images.

Table 6. Comparison of the proposed method with ReDMark [11].

Attack	Strength	ReDMark	Proposed
PSNR		40.24 [dB]	40.58 [dB]
No attack	-	-	0.7015
Gaussian filtering	Radius = 1	8.6	7.1429
	Radius = 1.6	39	9.7258
	Radius = 2	-	12.7232
Median filtering	3 × 3	13.4	8.4184
	5 × 5	-	10.5548
Salt and pepper noise addition	0.02	2.9	1.0204
	0.6	4.5	1.5306
	0.1	9.1	3.1888
Gaussian noise addition	5%	2.4	5.9949
	15%	14.5	27
	25%	25.6	38.1696
Sharpening	Radius = 1	0.9	0.9885
	Radius = 5	2.4	1.7217
	Radius = 10	3.2	2.0089
JPEG	90	1.6	0.9566
	70	4.2	4.24
	50	11.8	8.0676
Cropout	0.1	7.7	2.1365
	0.2	13.1	5.3253
	0.3	18.8	8.6735

Table 7. Comparison of the proposed method with recent studies.

Attack	Strength	[10]	[11]	[15]	Proposed (s = 2.75)
Training dataset	-	COCO	Cifar-10, Pascal VOC	COCO	BOSS
Training dataset	-	BOSS	Standard dataset	COCO	Standard dataset
PSNR		-	-	33.5	33.5
JPEG	50	37	25.4	23.8	0.6696
Cropout	0.3	6	7.5	2.7	5.8355
Dropout	0.3	7	8	2.6	4.7194
Crop	0.035	12	0	11	44.1327
Gaussian filtering	$\sigma = 2$	4	50	1.4	4.3048

5. Conclusions

In this paper, we proposed a digital image watermarking method using CNN that does not limit the resolution of the host image and WM information. This method adjusts the complementary relationship between invisibility and robustness using the strength factor. The pre-processing network for watermark increases the WM's resolution to that of the host image for the invisibility of the WM. The embedding network processes using CNNs that maintain the resolution to output the watermarked image. The extraction network also consists of CNNs to output the WM information by reducing the resolution. We performed attack simulations on the same distribution in each mini-batch to verify the robustness of the WM. This network is composed of a simple CNN and does not use

any resolution-dependent layer, such as the FC layer. It is, therefore, adaptive to the resolution of the input image. It is also independent of the WM information because it uses the newly and randomly generated WM information for each mini-batch in training.

Invisibility and robustness were measured for various pixel value change attacks and geometric attacks, for various WM information and host image resolutions. The results showed excellent performance and showed better performance for meaningful attacks in comparison with the state-of-the-art works. Therefore, our scheme has been proven to be very practical and universal. Besides, by adjusting the strength factor, we confirmed that our scheme could effectively control the complementary relationship between invisibility and robustness.

Therefore, we think the proposed method would be a beneficial watermarking scheme for a digital image because it enables the embedding and extraction of WMs without restrictions on the host image and WM information, that is, and without any additional training. The usefulness can be further improved by adequately controlling the invisibility and the robustness to obtain proper performance, as per the user requirements.

Author Contributions: Conceptualization, J.-E.L. and D.-W.K.; methodology, J.-E.L.; software, J.-E.L.; validation, J.-E.L., Y.-H.S. and D.-W.K.; formal analysis, J.-E.L., Y.-H.S. and D.-W.K.; investigation, J.-E.L.; resources, J.-E.L.; data curation, J.-E.L.; writing—original draft preparation, J.-E.L.; writing—review and editing, Y.-H.S. and D.-W.K.; visualization, J.-E.L.; supervision, D.-W.K.; project administration, D.-W.K.; funding acquisition, D.-W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2019R1F1A105455212).

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

References

1. Cox, I.J.; Miller, M.; Bloom, J.; Fridrich, J.; Kalker, T. *Digital Watermarking and Steganography*; Morgan Kaufmann Publisher: Burlington, MA, USA, 2007.
2. Kang, X.; Huang, J.; Shi, Y.Q.; Lin, Y. A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 776–786. [[CrossRef](#)]
3. George, J.; Varma, S.; Chatterjee, M. Color image watermarking using DWT-SVD and Arnold transform. In Proceedings of the 2014 Annual IEEE India Conference (INDICON), Pune, India, 11–13 December 2014; pp. 1–6.
4. Lee, Y.S.; Seo, Y.H.; Kim, D.W. Blind image watermarking based on adaptive data spreading in n-level DWT subbands. *Secur. Commun. Netw.* **2019**, *2019*, 8357251. [[CrossRef](#)]
5. Li, C.; Zhang, Z.; Wang, Y.; Ma, B.; Huang, D. Dither modulation of significant amplitude difference for wavelet based robust watermarking. *Neurocomputing* **2015**, *166*, 404–415. [[CrossRef](#)]
6. Ouyang, J.; Coatrieux, G.; Chen, B.; Shu, H. Color image watermarking based on quaternion Fourier transform and improved uniform log-polar mapping. *Comput. Electr. Eng.* **2015**, *46*, 419–432. [[CrossRef](#)]
7. Mehta, R.; Vishwakarma, V.P.; Rajpal, N. Lagrangian support vector regression based image watermarking in wavelet domain. In Proceedings of the 2015 2nd International Conference on SPIN, Noida, India, 19–20 February 2015; pp. 854–859.
8. Hu, H.; Chang, Y.; Chen, S. A progressive QIM to cope with SVD-based blind image watermarking in DWT domain. In Proceedings of the 2014 IEEE China Summit & International Conference on Signal and Information Processing, Xi'an, China, 9–13 July 2014; pp. 421–425.
9. Kandi, H.; Mishra, D.; Gorthi, S.R.S. Exploring the learning capabilities of convolutional neural networks for robust image watermarking. *Comput. Secur.* **2017**, *65*, 247–268. [[CrossRef](#)]
10. Zhu, J.; Kaplan, R.; Johnson, J.; Fei-Fei, L. HiDDeN: Hiding data with deep networks. In Proceedings of the European Conference on Computer Vision (ECCV), Xi'an, China, 9–13 July 2018; pp. 657–672.
11. Ahmadi, M.; Norouzi, A.; Soroushmehr, S.M.; Karimi, N.; Najarian, K.; Samavi, S.; Emami, A. ReDMark: Framework for residual diffusion watermarking on deep networks. *arXiv* **2018**, arXiv:1810.07248.
12. Mun, S.M.; Nam, S.H.; Jang, H.; Kim, D.; Lee, H.K. Finding robust domain from attacks: A learning framework for blind watermarking. *Neurocomputing* **2019**, *337*, 191–202. [[CrossRef](#)]

13. Zhong, X.; Shih, F.Y. A robust image watermarking system based on deep neural networks. *arXiv* **2019**, arXiv:1908.11331.
14. Wen, B.; Aydore, S. ROMark, a robust watermarking system using adversarial training. *arXiv* **2019**, arXiv:1910.01221.
15. Liu, Y.; Guo, M.; Zhang, J.; Zhu, Y.; Xie, X. A novel two-stage separable deep learning framework for practical blind watermarking. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 1509–1517.
16. Bas, P.; Filler, T.; Pevny, T. Break our steganographic system: The ins and outs of organizing BOSS. In *International Workshop on Information Hiding*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 59–70.
17. Dataset of Standard 512 × 512 Grayscale Test Images. Available online: <http://decsai.ugr.es/cvg/CG/base.htm> (accessed on 6 August 2019).
18. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).