sensors

MDPI

Article

Convolutional Neural Network Architecture for Recovering Watermark Synchronization

Wook-Hyung Kim [1], Jihyeon Kang [2], Seung-Min Mun [3] and Jong-Uk Hou [4,*]

1 Visual Display Division, Samsung Electronics, Suwon 16677, Korea; whkim@mmc.kaist.ac.kr
2 Graduate School of Information Security, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea; kangji@kaist.ac.kr
3 School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea; qkqhd222@kaist.ac.kr
4 School of Software, Hallym University, Chuncheon 24252, Korea
* Correspondence: juhou@hallym.ac.kr

Received: 10 August 2020; Accepted: 17 September 2020; Published: 22 September 2020

Abstract: In this paper, we propose a convolutional neural network-based template architecture that compensates for the disadvantages of existing watermarking techniques that are vulnerable to geometric distortion. The proposed template consists of a template generation network, a template extraction network, and a template matching network. The template generation network generates a template in the form of noise and the template is inserted into certain pre-defined spatial locations of the image. The extraction network detects spatial locations where the template is inserted in the image. Finally, the template matching network estimates the parameters of the geometric distortion by comparing the shape of spatial locations where the template was inserted with the locations where the template was detected. It is possible to recover an image in its original geometrical form using the estimated parameters, and as a result, watermarks applied using existing watermarking techniques that are vulnerable to geometric distortion can be decoded normally.

Keywords: digital watermark; depth-image-based rendering; copyright protection; template watermark; deep neural network

1. Introduction

Digital content has always been subject to copyright infringement due to its ease of duplication. Recently, as the market for real-time content services such as web-comics (also called webtoons) and video streaming services such as YouTube has grown very rapidly, the problem of copyright infringement is increasing. These services, unlike traditional paid services, offer free content and earn revenue by advertising to users. As a result, these services are more vulnerable to illegal copying because they are easier to access than services that are provided for a fee. In addition, these services are provided on Internet browsers or as smartphone applications, making it easier to replicate the content with the Internet browsers' or applications' downloading or capturing functions.

Watermarking techniques have emerged to reduce the loss caused by piracy. These techniques minimize loss through the insertion of invisible information in the content to enable tracking of illegal distribution routes and copyright authentication. As mentioned above, in real-time content provided by Internet browsers and applications, geometric distortion occurs very frequently due to capturing. Figure 1 shows an illegal distribution scenario that occurs in a real-time content service and a watermark extraction process suitable for such a scenario. The piracy process, such as the capturing shown in this figure, makes it difficult to decode the watermark by breaking the synchronization of the watermark. Correction of the geometric distortion must be performed to extract the watermark

accurately. Therefore, there is a need for a method that can effectively recover the image after such geometric distortion.

In this paper, we propose a convolutional neural network (CNN)-based template that can effectively correct geometric distortion. The proposed method divides the image into blocks of a specific size, inserts bit information into half of the blocks using the conventional watermarking technique, and inserts a CNN-based template to correct the geometric distortion in the other half. By inserting these block-based templates, it is possible to compensate for the disadvantages of geometric distortion while preserving the advantages of existing watermarking techniques except for the bit capacity.

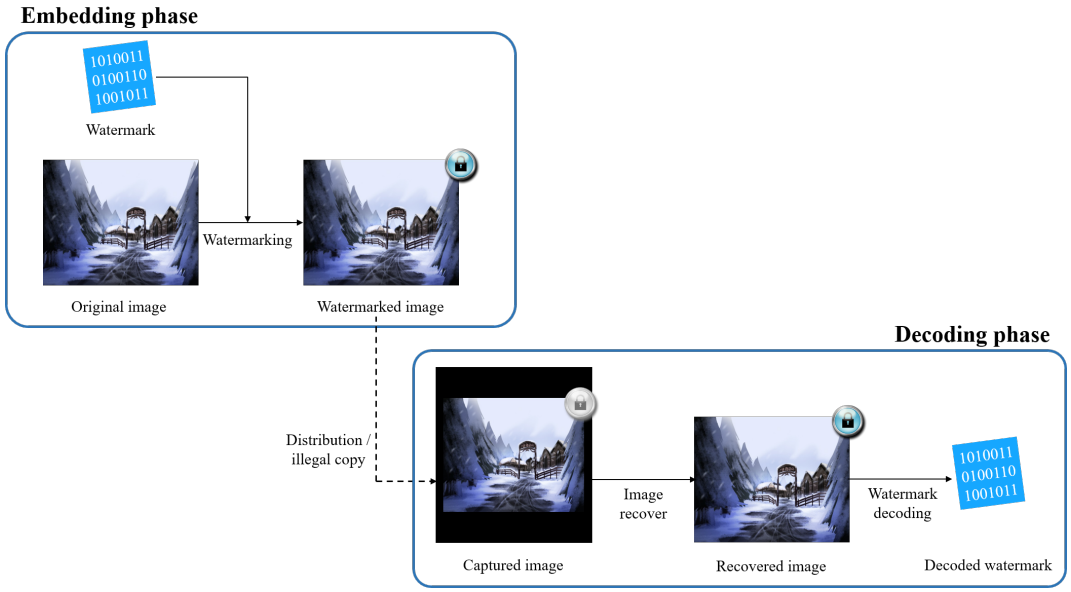

Figure 1. Illegal distribution scenario for real-time web-comics and watermark extraction process suitable for this scenario. Illegal distribution is frequently caused by screen capturing on Internet browsers and smartphone applications. Scaling and translation occur in this process and these distortions must be corrected before watermark decoding.

The contributions of the proposed method are as follows:

(1) A learning-based template network that restores geometric distortion. Through learning, a robust template against various attacks can be designed.
(2) Block-unit template. This makes it easy to design a multi-bit watermarking system and does not interfere with the watermark signal because it can be inserted independently with the watermark.
(3) Due to the above two characteristics, the template can be easily applied to other watermarking techniques. This complements the vulnerability to geometric distortion of new forms of watermarks such as the DIBR watermarking method as well as existing watermarks.

The remainder of this paper is organized as follows. Section 2 summarizes related works of robust watermarking and template-based watermarking. Section 3 presents the main concept of the proposed method, and Section 4 discusses the proposed method. Section 5 presents experimental results and Section 6 concludes the paper.

2. Related Work

To date, numerous watermarking techniques have been proposed to protect the copyright of content. Methods of inserting watermarks into various transform domains such as discrete cosine transform (DCT), discrete Fourier transform (DFT), radon transform, curvelet transform, Dual-tree complex wavelet transform (DT-CWT), and contourlet transform have been proposed [1–6]. Various insertion methods such as spread-spectrum, quantization index modulation (QIM), angle QIM, and absolute angle QIM have also been proposed [7–10]. However, these watermarking

techniques often show weaknesses after geometric distortion. Because geometric distortion causes synchronization errors, it is not easy to ensure robustness of these watermarks to geometric attacks. In addition, the print-scan process, which is regarded as a watermark removal attack, is commonly used for image reproduction and distribution. Print-scan resilient data hiding provides an authentication method of an important document, which is becoming more significant issue because of the security problems [11–13].

To reduce the vulnerability to geometric attacks, many studies have been carried out to identify the transform domain and insertion methods with invariant characteristics against geometric distortions [14–17]. However, these algorithms have drawbacks such as low invisibility, lack of bit capacity such as zero-bit watermarking, vulnerability to specific geometric distortion such as translation, or necessity of additional information.

Many template-based watermarking methods for decoding watermarks using template matching techniques have been proposed [18–21], but these methods also have a drawback in that it is difficult to insert sufficient copyright information because the bit capacity is low. In addition, since these templates are additionally inserted over the watermarked image in which the copyright information is embedded, the invisibility is further reduced, and the template and the watermark signal may interfere with each other.

These watermarking techniques robust to geometric attacks have many disadvantages compared to watermarking techniques that do not consider geometric attacks. In addition, if the synchronization problem is solved from the geometrically distorted image, the watermark can be normally decoded. For this reason, watermarking techniques that do not consider geometric attacks are used in many cases. If a geometric attack occurs, watermark decoding is performed after the image is recovered into the geometric characteristics of the original form. However, this approach requires a process for finding the original image information, comparing the original image with the geometrically distorted image, and then recovering the geometrical characteristics. This process is often inefficient because it is done manually or using heuristic search.

Conventional template-based watermarking techniques usually use whole image-unit transform rather than block-unit transform. This is because it is advantageous to use the image-unit transform when searching for invariant domains and insertion methods against rotation, scaling and translation (RST) attacks. When using the block-unit transform, it is necessary to solve the problem of correcting the block synchronization after geometrical distortion, which is difficult. However, the image-unit template has a problem in that the template cannot be inserted independently with the watermark containing copyright information unless they use the same transform domain. Therefore, there is a limitation that the template and watermark signal can interfere with each other. Also, it is difficult to design a multi-bit watermarking system in comparison with the block-unit method.

3. Main Concept of Proposed Method

The proposed method consists of a preparation step, insertion step, and decoding step as shown in Figure 2. In the preparation step, a random binary code is generated using the key, and this code is used to generate a 2D binary template matrix K of $M \times N$ size. This matrix determines whether each block is a template block or a watermark block in an image divided into blocks. It also serves as the ground truth for estimating the RST parameters in the template matching step.

In the insertion step, the image is divided into blocks, and then watermark insertion and template insertion are performed. First, the image is spatially divided into $M \times N$ blocks. The set of generated blocks is defined as B. The following rules distinguish the roles of the blocks.

$$B(x,y) = \begin{cases} \text{Template block}, & \text{if } K(x,y) = 1 \\ \text{Watermark block}, & \text{if } K(x,y) = 0 \end{cases} \quad (1)$$

where x and y are the horizontal and vertical coordinates of B and K, $0 \leq x < M$, and $0 \leq y < N$.

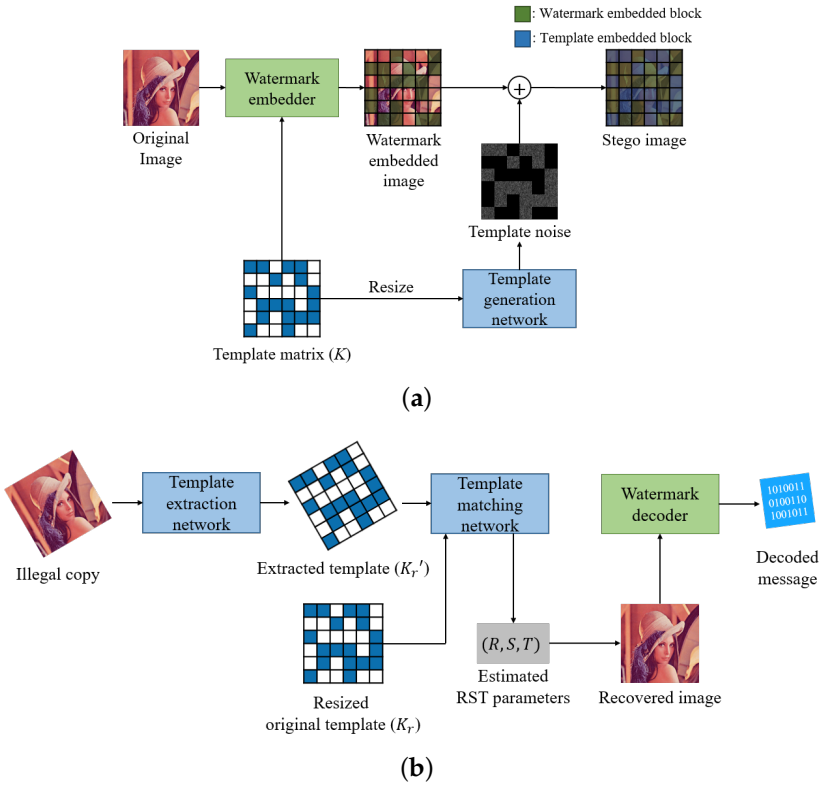

Figure 2. An overview of the proposed method. (**a**) insertion step, and (**b**) decoding step. A Stego image is an image in which both a watermark and a template are inserted.

Then, a block-based watermark is inserted into all the watermark blocks through the watermark embedder. The watermark embedder consists of an image transform, watermark insertion, and inverse transform in the same manner as conventional watermarking techniques.

As the final step of the insertion, a template is inserted into the template blocks. Template insertion is completed by simply adding the $w \times h$ sized noise output from the template generation network to the image, where w and h are the width and height of the image. The template generation network is responsible for generating a specific form of noise that can be detected in the template extraction network.

The decoding step consists of extracting the template, estimating the distorted geometric information of the image, recovering the image, and decoding the bits from the watermark. First, in the template extraction step, the extraction network finds the location where the template is inserted in the image. The template extraction network outputs a matrix K_r' with a value of 1 where the template is inserted and 0 where the watermark is inserted. Inputting the K_r' and the original template K_r, which can be obtained from the key, into the template matching network yields estimated RST parameters. K_r is simply a resize of K, which is used to increase the resolution of the template.

The geometrically distorted image is recovered using the estimated RST parameters. This step solves the problem of block synchronization, which is the biggest problem with block-based watermarking techniques. Therefore, the block-based watermark can be extracted blindly by using the block size and position information used in the watermark insertion step.

As the last step of decoding, the watermark extraction proceeds in the order of image transform and watermark extraction in the transformed domain as in conventional techniques.

The template generation network, template extraction network, and template matching network, which require a more detailed description, are described in Sections 4.1 and 4.2, and the watermark embedder and decoder are described in Section 4.3.

4. Proposed Method

In this section, the description of the template and the watermark are presented separately without considering the insertion and extraction order. The insertion and extraction order is detailed in Section 3.

In each step of template insertion and extraction, a resized K is used. Resizing is performed with the nearest neighbor filter, and K resized into $r \times r$ is defined as K_r. For example, K_{64} means that the K of size $M \times N$ is resized to 64×64. The reason for resizing K is to increase the resolution of the template to increase the accuracy in the matching step.

4.1. Template Embedding and Extraction Network

Figure 3 presents a template insertion and extraction scheme. Template embedding is performed as a simple sum as,

$$I_T = I + T_n, \tag{2}$$

where I_T is the template inserted image, I is the original image, and T_n is the template noise generated from the template generation network.

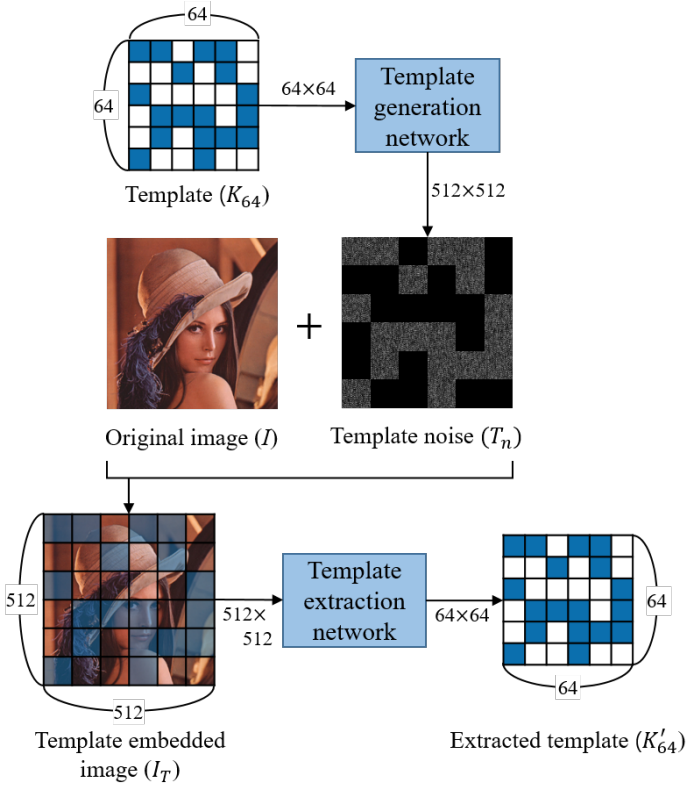

Figure 3. Template insertion and extraction overview. Networks are trained so that the extracted template K'_{64} and the inserted template K_{64} have the same shape.

In the template extraction step, a template is extracted from I_T. The goal is to learn the template generation network and extraction network so that the extracted template K'_{64} becomes the same as the template K_{64} used at the template insertion step.

The detailed template generation and extraction network structures are shown in Figure 4. We set the kernel size of all convolutional layers in the network to (3, 3). The reason is to extract the local features of the image similar to discrete wavelet transform rather than global features similar to DFT. In order to estimate the RST parameters, the same geometric transformation as that occurring in the image must occur in the extracted template. If the template is perfectly invariant and the template is

extracted without coordinate distortion, the RST parameters cannot be estimated. Thus, by exploiting local features, we induce the template to have semi-invariant properties. A semi-invariant template preserves its value from geometric distortion, but the coordinates are transformed according to the degree of geometric distortion.

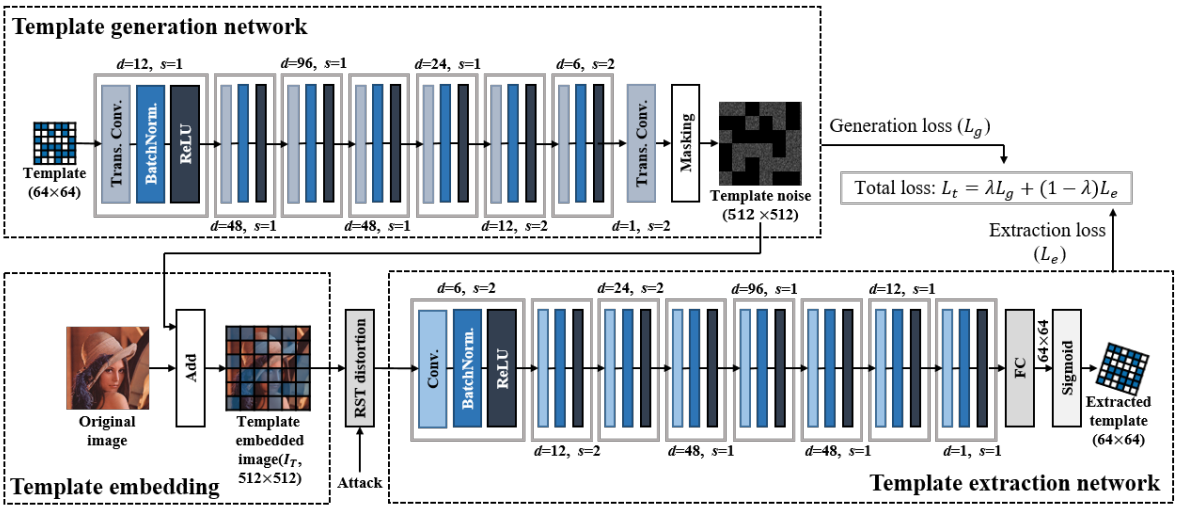

Figure 4. Proposed template generation and extraction network architectures. All kernel sizes of convolutional layers are set to (3, 3), *d* and *s* denote the depth and stride, respectively.

The template generation network is an inverse process to the template extraction network. This template generation network generates a template noise (T_n) of size 512 × 512 from a template (K_r) of size 64 × 64 using a transposed convolutional layer. The process of expanding the dimension is similar to the insertion method for conventional watermarking techniques. In conventional watermarking techniques, when the watermark inserted in the middle frequency of the transformed domain passing through the inverse transform, the watermark signal spreads to the entire image of the spatial domain. The proposed template generation network also spreads the low-dimensional signal to the high-dimensional spatial domain similar to the conventional technique. At the end of the network, the generated template noise is masked so that the template noise does not interfere with the watermark block. The loss of the template generation network is defined as,

$$L_g = \frac{\sum_{i,j=1}^{U}(T_n^o(i,j))^2}{U^2}, \tag{3}$$

T_n^o is the template noise generated from the generation network, i and j denote the horizontal and vertical coordinates of the generated template noise, respectively, U represents the width and height of the network output and is set to 512, i.e., since L_g corresponds to the average energy of the template to be inserted, the generation network is trained so as to improve the template invisibility. L_g is combined with the loss of the extraction network described below, and the generation network and the extraction network are trained together.

As in (2), simply adding the template noise generated from the template generation network to the original image will complete the template embedding. The template embedded image is then sent to the template extraction network after RST distortion.

The template extraction network extracts 64 × 64 templates that are the same size as the input of the generation network from the attacked image. The loss of the template extraction network is defined as,

$$L_e = \frac{1}{V^2}\sum_{i,j=1}^{V}[K'_V(i,j) - G^T(K_V(i,j))]^2, \tag{4}$$

where V is the size of the output of the extraction network and is set to 64. K'_V denotes the output of the network, i.e., the extracted template, and K_V denotes the original template. i and j denote the horizontal and vertical coordinates of the template, respectively. G^T indicates a geometric transformation using the ground-truth parameters. Since the geometric distortion that occurs in the image occurs equally in the inserted template, we define the loss function so that the extracted template is also subjected to this geometric distortion. This loss, L_e, trains the extraction network to improve the extraction accuracy of the template.

The total loss using the losses of the template generation and extraction networks is defined as,

$$L_t = \lambda L_g + (1 - \lambda) L_e, \tag{5}$$

where λ is the trade-off parameter between invisibility and template extraction accuracy. The larger the λ, the higher the invisibility but the lower the extraction accuracy.

To use the middle frequency as the template noise similar to the existing watermarking technique, the template generation network is pre-trained before the whole network training. First, as in the spread-spectrum watermark [1], we generate a middle frequency noise with a size of 512×512. This noise is generated by substituting a pseudo-random sequence from $1/4$ to $3/4$ of the zigzag-scanned DCT coefficients, and is defined as T_m. The pseudo-random sequence is set with an average value of 0 and a variance of 1. Then the template generation network is pre-trained until $\sum_{i,j=1}^{U}(T_n^o(i,j) - T_m(i,j))^2/U^2$ is less than 0.1. The template extraction network is initialized by the Xavier uniform initializer [22].

4.2. Template Matching Network

The template matching network shown in Figure 5 compares the extracted template with the original template and estimates the RST parameters. First, the feature extraction network extracts features from the extracted template and the original template. The feature extraction network mimics domain transformation methods such as DFT. Domain transforms, such as DFT, are the sum of all pixels multiplied by different weights in the image. Similar to DFT, we use the kernel size as the template size to compute the global features of the template. We compute 256 global features and reshape it by 16×16. This is similar to calculating 16×16 DFT coefficients from an image. As we can estimate the translation degree from the phase of the DFT, these global features will facilitate RST parameter estimation.

The extracted features are then matched to estimate the RST parameters. These matching layers refer to the structure in [23–25]. Feature extraction networks are Siamese networks that share weights with each other. After the feature extraction network, a concatenation layer, which is fast and has good matching accuracy, is used to combine features extracted from the feature extraction networks. Later layers are identical to the structure in [25]. The final result is a five-dimensional matrix $[R, S_x, S_y, T_x, T_y]$, which indicates the rotation, scaling of x and y, and the translation of x and y parameters. The loss of the matching network is defined as,

$$L_d = \frac{1}{S^2} \sum_{i,j=1}^{S} d[G^e(x_i, y_j), G^T(x_i, y_j)]^2, \tag{6}$$

where d is the Euclidean distance between two points, G_e is the geometric transformation using the estimated RST parameters, and G^T is the geometric transformation using the ground-truth RST parameters. x_i and y_i correspond to the S equally divided points of the image in the horizontal and vertical directions, respectively, and S is set to 10, i.e., L_d corresponds to the mean squared error between the estimated points and the true points. The reason for using the Euclidean distance as a loss without using the RST parameters directly is that each RST parameter has different weights of distortion on the image. For example, when a parameter of the same value is used, the distortion of rotation is larger than the distortion of translation.

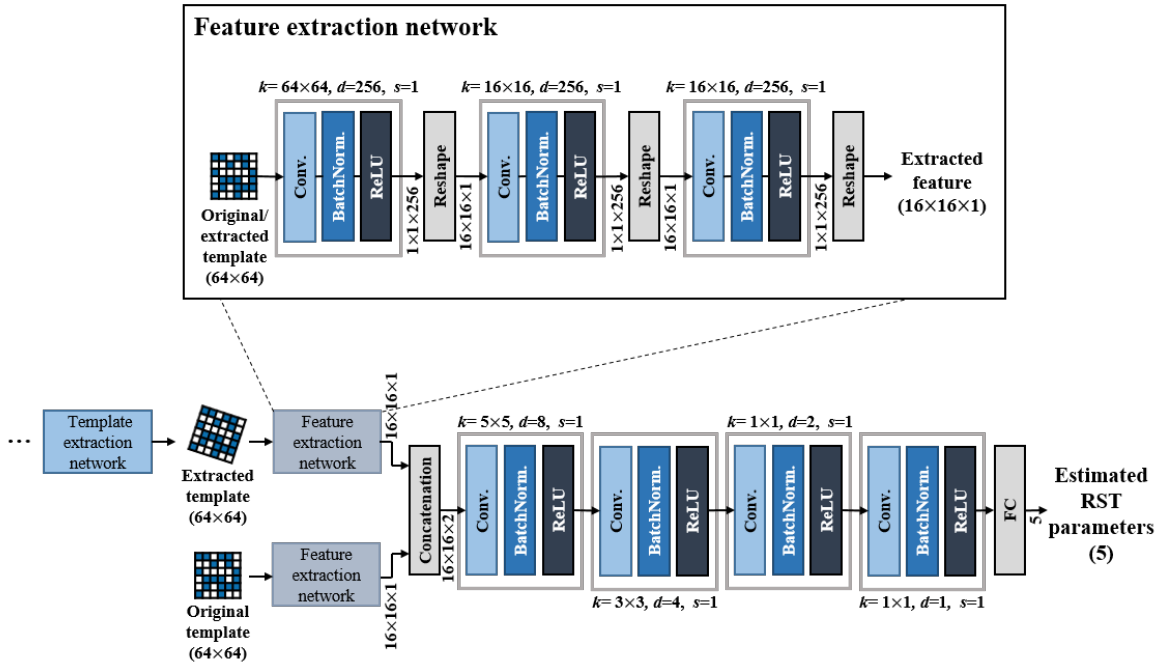

Figure 5. Template matching network architecture. Two feature extraction networks share weights with each other. k means kernel size, and zero-padding is not used in matching networks.

Although the template matching network has been described separately here, the template generation and extraction network described above are attached to the template matching network, and then end-to-end learning is performed. End-to-end loss is defined as,

$$L_{end-to-end} = \lambda L_g + (1 - \lambda) L_d, \tag{7}$$

where λ is the trade-off parameter between invisibility and matching accuracy. The template matching network also uses a Xavier uniform initializer to initialize the network. Since L_d includes training of the template extraction network, the template generation network, template extraction network, and template matching network are trained all at once by $L_{end-to-end}$.

4.3. Watermark Embedder and Decoder

The watermark has the role of inserting and decoding bit information. Similar to conventional techniques, the watermark is inserted/decoded in the transformed domain, and the curvelet is used as the transform domain. The reason for using this domain is that it is easy to insert multiple bits into one block through parameter adjustment while the watermark in this domain remains invisible and robust.

The curvelet transform is a multi-scale decomposition-like wavelet transform, and the curvelet represents the curve shape for various directions in the spatial domain [26–29]. In the image, the frequency domains are decomposed into various scales and directions by the curvelet transform as shown in Figure 6, and the curvelet coefficients are expressed as $C^{s,l}(i, j)$, where s means scale and l means direction. i and j represent the horizontal and vertical coordinates of the coefficients in each scale and direction, respectively.

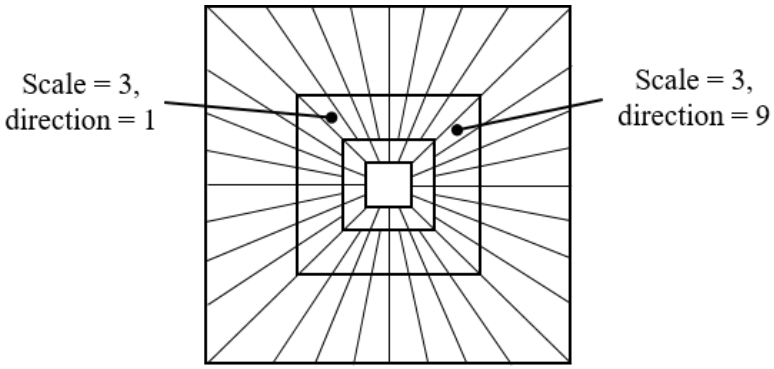

Figure 6. Frequency spectrum coverage of curvelet transform. The curvelet decomposes the frequency domain into various scales and directions.

In this paper, we divide the watermark block into 5 scales and 8 directions in total. The watermark is inserted using the QIM method as in Algorithm 1 on a scale of 3 levels.

Algorithm 1 QIM-based watermark embedding procedure.

Input: curvelet coefficients $C^{s,l}(i,j)$ over scale s and direction l
Output: modified curvelet coefficients $C_m^{s,l}(i,j)$
q : quantization step
b : bit to be inserted

$A^{s,l} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} abs(C^{s,l}(i,j))$
if mod[round$(A^{s,l}/q), 2] = b$ **then**
    $c = 0$
else
    if mod$(abs(A^{s,l})/q, 1) \leq 0.5$ **then**
        $c = 0.5$
    else if mod$(abs(A^{s,l})/q, 1) > 0.5$ **then**
        $c = -0.5$
    end if
end if
$Q^{s,l} = (\text{round}(abs(A^{s,l})/q + c) \cdot q)/A^{s,l}$
$C_m^{s,l}(i,j) = C^{s,l}(i,j) \cdot Q^{s,l}$

In Algorithm 1, m and n denote the horizontal and vertical sizes of the curvelet coeffcient, respectively, and i and j denote the horizontal and vertical coordinates of the curvelet coefficient. Q is a quantization operation, q is a quantization step, and b is the bit to be inserted. This algorithm is the same as in other QIM techniques and is designed to find the nearest quantization level corresponding to the bit to be inserted.

Because the curvelet coefficient expresses a curved shape with directionality, it is related to the surrounding coefficients. As a result, when one coefficient is modified, the modification spreads to the surrounding coefficients [30]. Therefore, when quantization is performed, the entire $C^{s,l}(i,j)$ should be adjusted by a multiplication operation as,

$$C_m^{s,l}(i,j) = C^{s,l}(i,j) \cdot Q^{s,l}, \tag{8}$$

where C_m denotes the modified curvelet coefficients. $A^{s,l}$ is adjusted by applying (8) to all coefficients.

The curvelet has a symmetry property similar to DFT. Therefore, the same modification should be applied in opposite directions. Since scale 3 has a total of 16 directions, a total of 8 bits can be inserted

into one block considering the symmetry property. If this process is applied to all watermark blocks, the total bit capacity becomes the number of 'watermark blocks' × 8.

The watermark decode proceeds as shown in Algorithm 2. C_d denotes the curvelet coefficient of the image to be decoded, and b_d denotes the decoded bit. This process is repeated for all watermark blocks to decode all the bits.

Algorithm 2 Watermark decoding procedure.

1: Input: watermarked curvelet coefficients $C^{s,l}(i,j)$
2: Output: decoded bit pattern b_d
3: $A_d^{s,l} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} abs(C_d^{s,l}(i,j))$
4: $b_d = \mathrm{mod}[\mathrm{round}(A_d^{s,l}/q),2]$

4.4. Application Method for Images of Various Sizes

Because the input size is fixed in CNN, all descriptions are based on 512 × 512 images according to the network input size. In the real world, however, there are images of various sizes, so all images must be resized to 512 × 512 to insert and extract the template and watermark. However, if the image is resized for watermark and template insertion, image quality degradation caused by the resizing cannot be avoided.

To avoid image degradation due to resizing, the embedding process is performed as shown in Figure 7. First, the image is resized to 512 × 512 and a template and watermark are inserted into the resized image to create a stego image. Subtracting the 512 × 512 image before the embedding step will leave only the stego signal, which contains the template and watermark signals. By resizing the stego signal into the original image size and adding it to the original image, the template and watermark can be inserted into the image without image quality degradation.

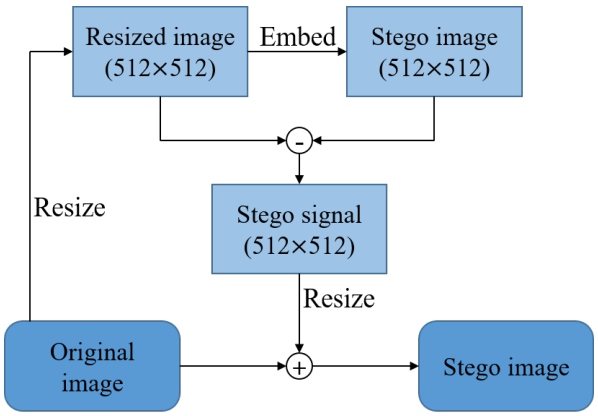

Figure 7. The process of inserting a template and watermark into images of various sizes.

5. Experimental Results

This section reports the performance of the proposed method in terms of invisibility and robustness. A comparison experiment is also conducted using the method described by Zhang [4], which uses the same domain and insertion method as the proposed method. Unlike the proposed watermarking method, Zhang's method transforms the whole image into a curvelet domain to insert a watermark. Whereas Zhang's method inserts multi-bits by dividing the curvelet directions, the proposed watermarking method inserts multi-bits by dividing the image into spatial blocks.

We also experimented with the proposed template for the new type of DIBR watermarking [31]. This DIBR watermarking method exploits the DT-CWT domain, which is robust against DIBR attacks and signal distortions, but vulnerable to geometric distortion. By comparing the performance before and after applying the template to the DIBR watermarking method, we show that the

proposed template can compensate the vulnerability of the geometric distortion for the new watermarking method.

5.1. Experiment Setting

BOSSBase [32], Middlebury [33,34], and Microsoft Research 3D Video [35] datasets with 5000 images with resolutions ranging from 512 × 512 to 1800 × 1500 were used for the experiment. A total of 3500 images were used for training, and the remaining 1500 images were used for performance testing. For network learning, 70,000 images were synthesized from the 3500 images with geometric and signal distortions. The geometric distortion used random parameters of 0 to 90-degree rotation, 0.7 to 1.5 times scaling, and 0 to 30% translation. The signal distortion also randomly applied a Gaussian noise of 0 to 200 variance and a JPEG compression factor of 30 to 100 to the geometrically distorted image.

The proposed method was implemented with a tensorflow library [36] and Python. The network was trained using the adam optimizer [37], with a learning rate of 10^{-3}, epsilon of 10^{-8}, batch size of 32, and λ in (7) of 0.2. Until convergence occurred, the end-to-end network was trained and convergence typically occurred after 15 epochs. The training took about an hour per epoch using the Nvidia GTX 1080 single GPU.

For fair comparison, all experiments were performed on a gray channel. The bit capacity for Zhang's method and the proposed watermarking method were set to 256 bits. For the proposed watermarking method, the image was divided by 8 × 8 to create 64 blocks. Among these, 32 blocks were used to insert bit information and the remaining blocks were used for the template. The proposed watermark was inserted on scale 3 in the curvelet domain divided by scale 5 and direction 8. Zhang's method inserts the watermark on scales 3 and 4 in the curvelet domain divided by scale 5 and direction 128. The quantization step q for the proposed method and Zhang's method was set to 3.

In the experiment with the DIBR watermarking method [31], all other parameter values were set to default, and only the block size was adjusted. In the DIBR watermarking method with the template, the image was divided by 8 × 8 to create 64 blocks. Among these, 32 blocks were used to insert bit information and the rest were used for the template. In the method without a template, the image was divided by 5 × 6 to create a total of 30 blocks, and all 30 blocks were inserted with bit information. With this block size adjustment, the bit capacity of the DT-CWT method with a template and the DT-CWT method without a template were set to a similar level.

5.2. Image Quality

Figure 8 shows the original image and the template/watermark-embedded image. As can be seen, the degradation of quality due to the proposed watermark and template insertion is hardly noticeable. We also measured the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [38] for more objective image quality measurements. As can be seen from Table 1, the 'proposed template + proposed watermark' shows a similar image quality to that obtained using Zhang's method.

The watermarking method using DT-CWT shows a lower visual quality than other methods because the DT-CWT watermarking method greatly modifies the image to have robustness to DIBR. 'DT-CWT + proposed template' has a slightly better visual quality than the DT-CWT only method because it uses half the image area as a template, which has relatively lower energy than the DIBR watermark.

Table 1. Average PSNR and SSIM.

	PSNR	**SSIM**
Proposed template + watermark	43.6668	0.9859
Zhang's method	43.2523	0.9850
DT-CWT (DIBR watermark)	40.3532	0.9788
DT-CWT + proposed template	41.1286	0.9804

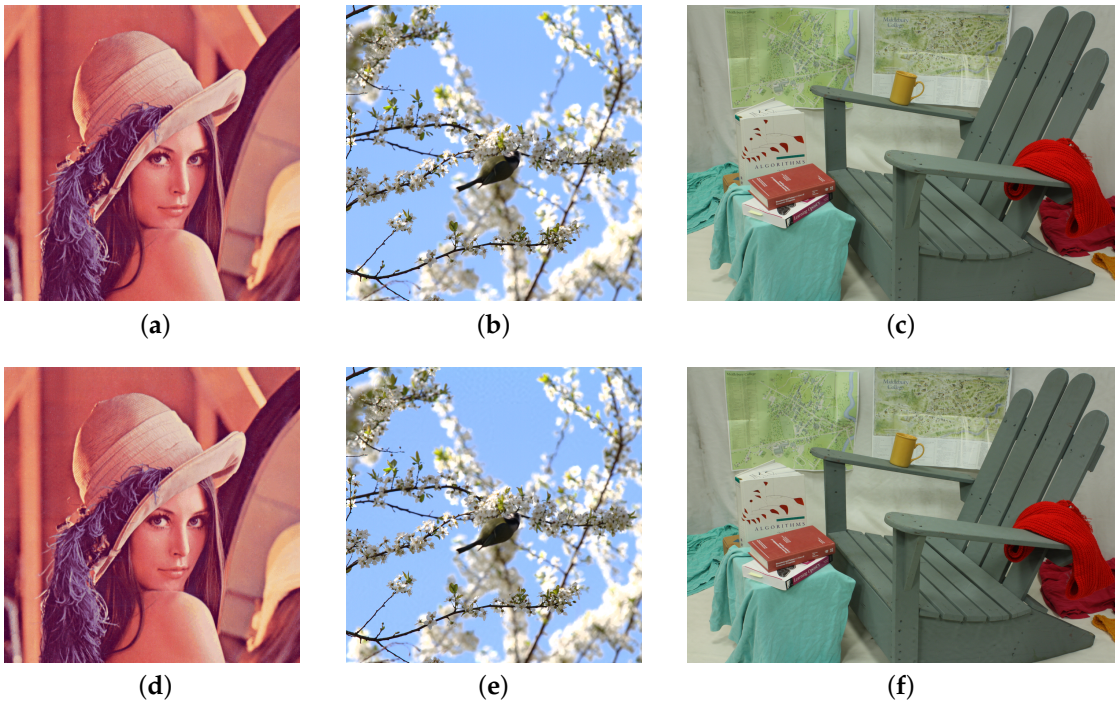

Figure 8. (**a**–**c**) Original image, (**d**–**f**) Proposed template and watermark-embedded image.

5.3. Robustness Test for RST Attack

Figure 9 shows the robustness test results for rotation, scaling, and translation. The following four methods were tested for robustness: (1) proposed watermarking method without recovery, (2) proposed watermarking method with proposed template recovery, (3) proposed watermarking method with ground-truth recovery and (4) Zhang's method. The robustness performance was measured based on the bit-error-rate (BER).

Zhang's method shows good robustness to low-level geometric distortion. This is because the absolute values of curvelet coefficients do not have a large variation in weak geometric distortions. Especially, this method shows low BER for weak scaling and translation. However, it shows a relative weakness in rotation.

The proposed watermarking method without recovery showed lower performance despite using a similar insertion method in the same domain as Zhang's method. This is because the watermark is inserted with block-units unlike with Zhang's method. Since block synchronization is broken due to geometric distortion, block-based watermarking is not effective without correction for geometric distortion. On the other hand, low BER is obtained for scaling even without recovery because the proposed watermarking method uses resized images in fixed sizes of 512×512 for the watermark insertion/decoding process to fit the image size to the network size.

If an image is recovered using the proposed template, the proposed watermark shows a low error even for strong geometric distortion. This result shows that the robustness is almost identical to the results of recovery with ground-truth, which means that the template almost completely recovers the image from the geometric distortion. The error that occurs even when the image is completely recovered is because the watermark information is cropped together with the image information. Except for the cropped part, the watermark is normally decoded.

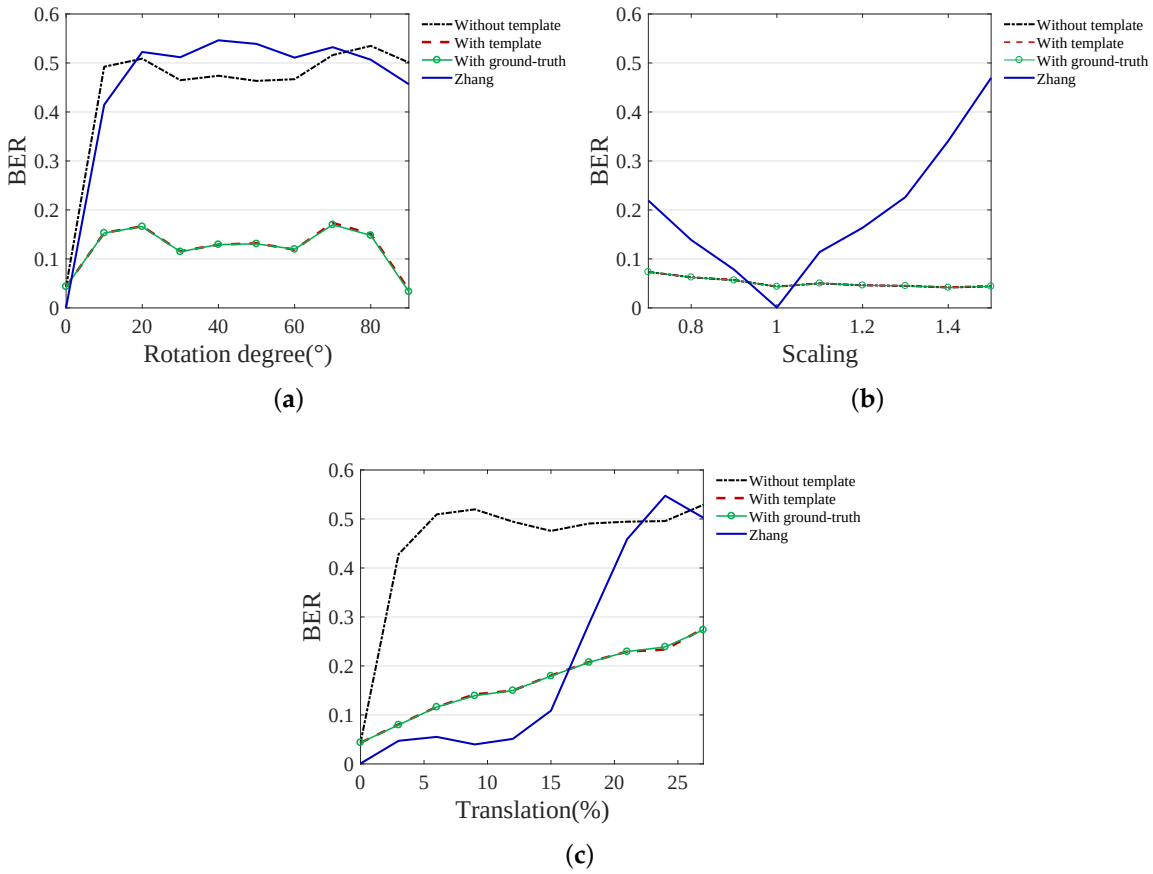

Figure 9. Robustness results for geometric distortions. (**a**) rotation, (**b**) scaling, (**c**) translation. The method with the proposed template recovery and the method with ground-truth recovery show almost the same performance.

5.4. Robustness Test for Simultaneous Attack

To test the robustness of the template for signal distortion, we measured the BER when signal distortion and RST distortion occurred simultaneously. We compared the BER when recovering an image with the proposed template and ground-truth.

Tables 2–4 show the results of BER when Gaussian noise and RST distortion occur simultaneously. As can be seen from these tables, the larger the noise variance, the greater the error in the method with a template compared to the method with ground-truth. At 200 noise variance, the BER is very high in the method with a template. This is because the noise has corrupted the template as well as the watermark signal, and the image has not been correctly recovered. However, considering that the variance of the inserted template noise is less than 10, the noise variance of 200 is very strong, and in practice, such a large amount of noise barely occurs. In addition, there is little difference between the template-recovered method and the ground-truth-recovered method for scaling. This is because the watermark is inserted/decoded at a fixed image size of 512 $\times$ 512 as mentioned above.

Table 2. Average BER for rotation attack with Gaussian noise. GT denotes ground-truth.

		Noise Variance			
Rot.	**Recover**	**25**	**50**	**100**	**200**
10°	With template	0.174	0.233	0.268	0.304
	With GT	0.17	0.22	0.265	0.281
30°	With template	0.138	0.205	0.258	0.288
	With GT	0.132	0.189	0.223	0.244
50°	With template	0.169	0.241	0.28	0.314
	With GT	0.145	0.2	0.232	0.25
70°	With template	0.204	0.297	0.333	0.387
	With GT	0.189	0.248	0.296	0.32
90°	With template	0.081	0.204	0.254	0.33
	With GT	0.054	0.159	0.194	0.243

Table 3. Average BER for scaling attack with Gaussian noise.

		Noise Variance			
Scaling	**Recover**	**25**	**50**	**100**	200
0.7	With template	0.089	0.173	0.235	0.275
	With GT	0.089	0.168	0.226	0.264
0.9	With template	0.07	0.16	0.201	0.249
	With GT	0.061	0.163	0.197	0.244
1.2	With template	0.068	0.161	0.207	0.241
	With GT	0.068	0.161	0.207	0.241
1.5	With template	0.065	0.152	0.204	0.243
	With GT	0.065	0.152	0.204	0.243

Table 4. Average BER for translation attack with Gaussian noise.

		Noise Variance			
Trans.	**Recover**	**25**	**50**	**100**	**200**
3%	With template	0.119	0.229	0.273	0.338
	With GT	0.1	0.198	0.23	0.275
9%	With template	0.163	0.249	0.289	0.354
	With GT	0.15	0.227	0.253	0.301
15%	With template	0.203	0.291	0.329	0.387
	With GT	0.19	0.263	0.289	0.324
21%	With template	0.255	0.328	0.364	0.427
	With GT	0.234	0.294	0.319	0.353
27%	With template	0.304	0.369	0.4	0.445
	With GT	0.284	0.338	0.359	0.378

5.5. Application of Proposed Template to DIBR Watermarking Method

We tested the robustness of the DIBR watermarking method against RST when the proposed template is applied. The experiment was conducted in three ways: (1) DT-CWT with template recovery, (2) DT-CWT with ground-truth recovery, and (3) Only DT-CWT method. All experiments used the right-view image rendered by DIBR. In other words, we measured the BER of images with DIBR

distortion and RST distortion simultaneously. DIBR parameters used in this test were the recommended values in [39].

As shown in Figure 10, the DT-CWT-based DIBR watermarking method has a low BER for weak geometric distortions, similar to curvelet watermarking methods. However, as the degree of geometric distortion increases, the BER increases sharply.

On the other hand, if the image is recovered using a template with the DT-CWT method, the watermark can be decoded well after the geometric distortion. The DT-CWT method recovered with a template has a slightly higher BER than that recovered with ground-truth because the template is disturbed by DIBR. However, since the error increase rate is insignificant, the proposed template can be considered robust to the new type of distortion, DIBR. These results show that the proposed template can give robustness against geometric attacks to newly proposed watermarking techniques.

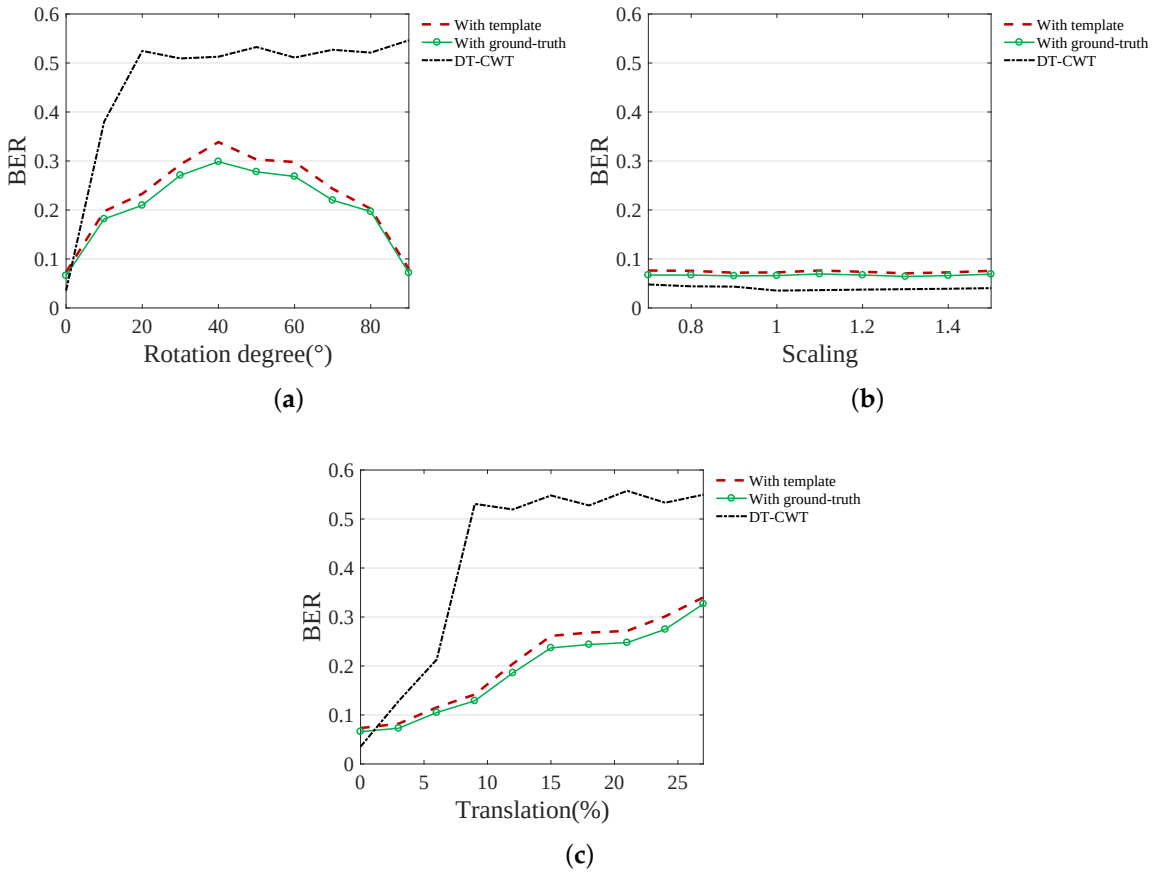

Figure 10. Average BER for RST attack with DIBR rendering. (**a**) rotation, (**b**) scaling, (**c**) translation.

6. Discussion and Conclusions

Conventional template-based watermarking techniques usually use whole image-unit transform rather than block-unit transform. However, the image-unit template has a problem in that the template cannot be inserted independently with the watermark containing copyright information unless they use the same transform domain. Therefore, there is a limitation that the template and watermark signal can interfere with each other. Also, it is difficult to design a multi-bit watermarking system in comparison with the block-unit method. On the other hand, since the proposed template is a block-unit method and can be separated spatially, it does not cause interference with the watermark. Moreover, because it is a learning-based method, it has the advantage of being able to respond quickly to new types of distortion. Due to these advantages, the proposed template can be applied not only to conventional watermarking methods but also to newly proposed watermarking methods for various purposes.

For example, the proposed template can be applied to a new form of watermarks such as the recently proposed depth-image-based rendering (DIBR) watermarking technique. DIBR is a rendering method to give a stereoscopic effect to images [40,41], but it cannot be protected by conventional watermarks and templates because it causes horizontal non-linear distortion. To cope with this, various DIBR watermarking techniques have been proposed [31,42–44], but they show weaknesses in geometric distortion since there have been few studies on the topic. The proposed learning-based and block-based template can easily solve the problem of geometric distortion for the DIBR watermark.

However, the proposed template-based watermarking system also has an inherent problem. Instead of acquiring robustness, there is an increase in computational cost for matching the embedded template. For example, compared to the case of applying only the curvelet transform, the watermark detection time increases by about 0.6 s on average when the proposed template is used. In the case of watermark, about 0.6 s of CPU time comes out when decoding a 512 $\times$ 512 video (based on Intel 6700 K). The proposed method uses only half of the image area, resulting in around 0.3 s of CPU time. In the case of the template, it takes around 0.25 s of GPU time to decode (based on GTX 1080). This can be a weakness when processing large-scale image data (e.g., web-scale image database) and improvement in template processing time will be needed in future studies. In terms of cost-efficiency, model simplification of the deep neural network [45,46] should be helpful to improve the performance.

Recently, Webcomic companies and content providers of IPTV have started actively using these techniques to track and punish illegal distribution. Watermarking techniques and templates based on CNN are still in the early stage for these applications. Further research is needed to improve watermark performance such as masking techniques that will increase invisibility and security enhancement. As the market for real-time content services such as web-comics (also called webtoons) and video streaming services such as YouTube has grown very rapidly, the demands of the watermarking framework have to be not only user friendly but also requires computing efficiency. There are some potential implementation of the proposed method based on the other machine learning techniques including graphical models, and different CNN architectures such as dilated convolution [47]. Also, we will extend the scope of our research into video content watermarking using CNN.

Author Contributions: Conceptualization, W.-H.K.; Methodology, W.-H.K.; Software, W.-H.K.; Writing-Original Draft Preparation, W.-H.K.; Writing-Review and Editing, J.K., S.-M.M. and J.-U.H.; Validation, J.K.; Software, S.-M.M.; Conceptualization, Supervision and Funding Acquisition, J.-U.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1C1C1013433), and in part by Hallym University Research Fund, 2020 (HRF-202005-017).

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

References

1. Barni, M.; Bartolini, F.; Cappellini, V.; Piva, A. A DCT-domain system for robust image watermarking. *Signal Process.* **1998**, *66*, 357–372. [CrossRef]
2. Solachidis, V.; Pitas, L. Circularly symmetric watermark embedding in 2-D DFT domain. *IEEE Trans. Image Process.* **2001**, *10*, 1741–1753. [CrossRef]
3. Simitopoulos, D.; Koutsonanos, D.E.; Strintzis, M.G. Robust image watermarking based on generalized Radon transformations. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 732–745. [CrossRef]
4. Zhang, C.; Cheng, L.L.; Qiu, Z.; Cheng, L.M. Multipurpose watermarking based on multiscale curvelet transform. *IEEE Trans. Inf. Forensics Secur.* **2008**, *3*, 611–619. [CrossRef]
5. Coria, L.E.; Pickering, M.R.; Nasiopoulos, P.; Ward, R.K. A video watermarking scheme based on the dual-tree complex wavelet transform. *IEEE Trans. Inf. Forensics Secur.* **2008**, *3*, 466–474. [CrossRef]
6. Akhaee, M.A.; Sahraeian, S.M.E.; Marvasti, F. Contourlet-based image watermarking using optimum detector in a noisy environment. *IEEE Trans. Image Process.* **2010**, *19*, 967–980. [CrossRef]
7. 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]

8. Chen, B.; Wornell, G.W. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Trans. Inf. Theory* **2001**, *47*, 1423–1443. [CrossRef]
9. Ourique, F.; Licks, V.; Jordan, R.; Pérez-González, F. Angle QIM: A novel watermark embedding scheme robust against amplitude scaling distortions. In Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), Philadelphia, PA, USA, 23 March 2005; Volume 2, p. ii-797.
10. Nezhadarya, E.; Wang, Z.J.; Ward, R.K. Robust image watermarking based on multiscale gradient direction quantization. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 1200–1213. [CrossRef]
11. He, D.; Sun, Q. A practical print-scan resilient watermarking scheme. In Proceedings of the IEEE International Conference on Image Processing 2005, Genova, Italy, 14 September 2005; Volume 1, p. I-257.
12. Kang, X.; Huang, J.; Zeng, W. Efficient general print-scanning resilient data hiding based on uniform log-polar mapping. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 1–12. [CrossRef]
13. Hou, J.U.; Kim, D.G.; Lee, H.K. Blind 3D mesh watermarking for 3D printed model by analyzing layering artifact. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2712–2725. [CrossRef]
14. Ruanaidh, J.J.O.; Pun, T. Rotation, scale and translation invariant spread spectrum digital image watermarking1. *Signal Process.* **1998**, *66*, 303–317. [CrossRef]
15. Kim, H.S.; Lee, H.K. Invariant image watermark using Zernike moments. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 766–775.
16. Xin, Y.; Liao, S.; Pawlak, M. A multibit geometrically robust image watermark based on Zernike moments. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, UK, 26 August 2004; Volume 4, pp. 861–864.
17. Wang, S.; Cui, C.; Niu, X. Watermarking for DIBR 3D images based on SIFT feature points. *Measurement* **2014**, *48*, 54–62. [CrossRef]
18. Stankovic, S.; Djurovic, I.; Pitas, I. Watermarking in the space/spatial-frequency domain using two-dimensional Radon-Wigner distribution. *IEEE Trans. Image Process.* **2001**, *10*, 650–658. [CrossRef]
19. Lu, W.; Lu, H.; Chung, F.L. Feature based watermarking using watermark template match. *Appl. Math. Comput.* **2006**, *177*, 377–386. [CrossRef]
20. Bas, P.; Chassery, J.M.; Macq, B. Geometrically invariant watermarking using feature points. *IEEE Trans. Image Process.* **2002**, *11*, 1014–1028. [CrossRef]
21. Pereira, S.; Pun, T. Robust template matching for affine resistant image watermarks. *IEEE Trans. Image Process.* **2000**, *9*, 1123–1129. [CrossRef]
22. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
23. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Deep image homography estimation. *arXiv* **2016**, arXiv:1606.03798.
24. Kanazawa, A.; Jacobs, D.W.; Chandraker, M. Warpnet: Weakly supervised matching for single-view reconstruction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27 June 2016; pp. 3253–3261.
25. Rocco, I.; Arandjelovic, R.; Sivic, J. Convolutional neural network architecture for geometric matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; Volume 2.
26. Candes, E.J.; Donoho, D.L. *Curvelets: A Surprisingly Effective Nonadaptive Representation for Objects with Edges*; Technical Report; Stanford University Department of Statistics: Stanford, CA, USA, 2000.
27. Candès, E.J.; Guo, F. New multiscale transforms, minimum total variation synthesis: Applications to edge-preserving image reconstruction. *Signal Process.* **2002**, *82*, 1519–1543. [CrossRef]
28. Candès, E.J.; Donoho, D.L. New tight frames of curvelets and optimal representations of objects with piecewise C2 singularities. *Commun. Pure Appl. Math.* **2004**, *57*, 219–266. [CrossRef]
29. Candes, E.; Demanet, L.; Donoho, D.; Ying, L. Fast discrete curvelet transforms. *Multiscale Model. Simul.* **2006**, *5*, 861–899. [CrossRef]
30. Kim, W.H.; Nam, S.H.; Lee, H.K. Blind curvelet watermarking method for high-quality images. *Electron. Lett.* **2017**, *53*, 1302–1304. [CrossRef]

31. Kim, H.D.; Lee, J.W.; Oh, T.W.; Lee, H.K. Robust DT-CWT watermarking for DIBR 3D images. *IEEE Trans. Broadcast.* **2012**, *58*, 533–543. [CrossRef]
32. Bas, P.; Filler, T.; Pevnỳ, 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.
33. Hirschmuller, H.; Scharstein, D. Evaluation of cost functions for stereo matching. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07), Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
34. Scharstein, D.; Hirschmüller, H.; Kitajima, Y.; Krathwohl, G.; Nešić, N.; Wang, X.; Westling, P. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 31–42.
35. Zitnick, C.L.; Kang, S.B.; Uyttendaele, M.; Winder, S.; Szeliski, R. *High-Quality Video View Interpolation Using a Layered Representation*; ACM Transactions on Graphics (TOG); ACM: New York, NY, USA, 2004; Volume 23, pp. 600–608.
36. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, GA, USA, 2–4 November 2016; Volume 16, pp. 265–283.
37. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *Image Process. IEEE Trans.* **2004**, *13*, 600–612. [CrossRef]
39. Zhang, L.; Tam, W.J. Stereoscopic image generation based on depth images for 3D TV. *IEEE Trans. Broadcast.* **2005**, *51*, 191–199. [CrossRef]
40. Fehn, C.; Kauff, P.; De Beeck, M.O.; Ernst, F.; Ijsselsteijn, W.; Pollefeys, M.; Van Gool, L.; Ofek, E.; Sexton, I. An evolutionary and optimised approach on 3D-TV. In Proceedings of the IBC, Amsterdam, The Netherlands, 13–17 September 2002; Volume 2, pp. 357–365.
41. Fehn, C. Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. In *Stereoscopic Displays and Virtual Reality Systems XI*; International Society for Optics and Photonics: Washington, DC, USA, 2004; Volume 5291, pp. 93–105.
42. Lin, Y.H.; Wu, J.L. A digital blind watermarking for depth-image-based rendering 3D images. *IEEE Trans. Broadcast.* **2011**, *57*, 602–611. [CrossRef]
43. Nam, S.H.; Kim, W.H.; Mun, S.M.; Hou, J.U.; Choi, S.; Lee, H.K. A SIFT features based blind watermarking for DIBR 3D images. *Multimed. Tools Appl.* **2017**, *77*, 7811–7850. [CrossRef]
44. Al-Haj, A.; Farfoura, M.E.; Mohammad, A. Transform-based watermarking of 3D depth-image-based-rendering images. *Measurement* **2017**, *95*, 405–417. [CrossRef]
45. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, London, UK, 8–10 July 2019; pp. 6105–6114.
46. Sun, X.; Ren, X.; Ma, S.; Wei, B.; Li, W.; Xu, J.; Wang, H.; Zhang, Y. Training simplification and model simplification for deep learning: A minimal effort back propagation method. *IEEE Trans. Knowl. Data Eng.* **2018**, 32, 374–387. [CrossRef]
47. Rekabdar, B.; Mousas, C. Dilated Convolutional Neural Network for Predicting Driver's Activity. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3245–3250.

© 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/).