Hindawi
Security and Communication Networks
Volume 2020, Article ID 8869096, 9 pages
https://doi.org/10.1155/2020/8869096

Research Article

A Robust Image Watermarking Approach Using Cycle Variational Autoencoder

Qiang Wei,[1] Hu Wang,[2] and Gongxuan Zhang[1]

[1] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*
[2] *Beijing Mysleepart Technology Co., Ltd., Beijing, China*

Correspondence should be addressed to Gongxuan Zhang; gongxuan@njust.edu.cn

Received 7 July 2020; Revised 26 August 2020; Accepted 7 September 2020; Published 22 September 2020

Academic Editor: Xiaolong Xu

With the rapid development of Internet and cloud storage, data security sharing and copyright protection are becoming more and more important. In this paper, we introduce a robust image watermarking algorithm for copyright protection based on variational autoencoder networks. The proposed image watermarking embedding and extracting network consists of three parts: encoder subnetwork, decoder subnetwork, and detector subnetwork. In the training process, the encoder and decoder subnetworks learn a robust image representation model and further implement the embedding of 1-bit watermark image to the cover image. Meanwhile, the detector subnetwork learns to extract the 1-bit watermark image from the embedding image. Experimental results demonstrate that the watermarked images generated by the proposed algorithm have better visual effects and are more robust against geometric and noise attacks than traditional approaches in the transform domain.

1. Introduction

In the era of big data and cloud computing, especially with the rapid development of mobile edge computing (MEC) technology, the demand for real-time services from a wide range of mobile terminals and commercial services providers (CSPs) is more and more urgent. On the one hand, many MEC-based services have been provided, such as paper citation network based link prediction and paper recommendation [1, 2], electricity load forecasting [3], and energy efficient dynamic offloading [4]. To fulfill real-time responses of MEC-based services, workflow scheduling and management are very important. In [5–8], many workflow scheduling approaches under different systems and environments (i.e., NSGA-II, edge computing environment, cyber-physical cloud systems, etc.) have been proposed. However, on the other hand, whether in the stage of data collection or application, people can access the required multimedia resources more easily than before, which will pose a serious threat to the privacy and copyright protection of those multimedia resources [9].

Privacy protection and authentication technologies can be divided into two categories. One is at the system level, which means the recommendation algorithms deployed in the service system (i.e., LSH-based recommender systems, multidimensional service recommendation, etc.) can avoid the users' request for obtaining the privacy information [10–13]. The other is at the data level, known as active authentication technology. In this kind of technology, digital image watermarking technology has become an important means of copyright protection of image resources. However, the problems of geometric attack resistance and balance between robustness and imperceptibility are still common problems in the field of digital image watermarking research.

Traditional image watermarking algorithms are often implemented in the transform domain; that is, image is firstly transformed into frequency or spatial-frequency domain (e.g., discrete cosine transform or wavelet transform). Then, appropriate coefficients in transform domain are selected for embedding watermark images. Finally, the modified transform domain coefficients, which are embedded watermarking information, are transformed back to the spatial domain to derive the watermarked digital images [14–16]. Although the watermarked images generated by this kind of approaches have good visual effect, they are not robust against geometric and noise attacks.

In recent years, some studies have introduced deep learning and adversarial learning into the field of watermarking and steganography. For instance, Volkhonskiy et al. have proposed a Steganographic Generative Adversarial Networks (SGAN) model [17], which, for the first time, incorporates the GAN and adversarial learning with information steganography technology. In this approach, an additional information embedding module was added on the basis of the original generative network to produce pseudonatural images after embedding information. Meanwhile, a steganalysis discriminant network is trained to discriminate the original natural image and the watermarked images generated by the generator. Under this framework, Shi et al. used Wasserstein GAN (WGAN) to optimize the training procedure and make the generated watermarked image more realistic with better visual quality [18]. Based on additive distortion cost function, Tang et al. firstly proposed the concept of automatic steganographic distortion learning (ASDL) model, which is called ASDL-GAN [19]. In this algorithm, the probability matrix P of image pixel modification is obtained by deep learning, and then the Syndrome-Trellis Codes (STC) method is used for information embedding. However, in this kind of GAN-based method, the discriminator is only used to distinguish whether the generated image contains hidden information or not and the quality of the generated image is not evaluated. That means its essence is to judge whether the probability distribution in the parameter space of the natural image or generated image is distinguishable. So, it cannot guarantee the visual quality of the generated image. Therefore, Mun et al. proposed a watermark network (WM-Net), which directly uses convolutional neural network (CNN) to fulfill the robust image watermarking and improves the antiattack ability of the watermarking embedding network by adding geometric attacks during the training process of the network [20]. However, the proposed CNN model does not contain any loss function to evaluate the quality of recovered watermark image either.

Therefore, in this paper, we propose a robust image watermarking embedding algorithm based on cycle variational autoencoder (Cycle-VAE) networks. One advantage of VAE model is that it can learn an abstract representation of a particular kind of images (such as face images). Furthermore, we use a convolution network similar to that in [20] to embed a 1-bit watermarking image into the cover image in the representation space. Although this strategy is similar to the WM-Net, they have two main differences. On the one hand, in the WM-Net, quaternion discrete Fourier transform (QDFT) is used before the watermark embedding, which is a fixed transform. But in the Cycle-VAE model, the network tries to learn an image transform that is more suitable for information embedding. On the other hand, in the WM-Net, the images should be partitioned into image blocks before performing the QDFT, like DCT-based watermark techniques. This will affect the ability of watermark algorithm for antigeometric attack. However, our proposed network can deal with the image entirely. In addition, because the dimension of image in the abstract representation space is usually not too high, the embedding and extraction network of watermark can also be small. Finally, to ensure the balance between the reality of the watermarked image and the reliability of the extracted watermark, we adopt a similar mechanism as Cycle Generative Adversarial Network (CycleGAN) [21]. In cycle A, an image is transformed to the representation space via encode network, after watermark embedding, and then transformed back to the image space via decode network. The loss function constrains the consistency between the input and watermarked image. Meanwhile, in cycle B, a watermark is embedded in the representation space by the embedding network, after transforming to the image space and back to the representation space again, and then extracted by the detection network. The loss function constrains the consistency between the input and recovered watermark. A demonstration of the above flow chart is shown in Figure 1.

This paper will be presented by the following parts. Section 2 gives an overview of the related works about CycleGAN and VAE approaches. Next section describes the proposed Cycle-VAE model for image watermarking, including the network structures, loss function, and implementation details. Section 4 has shown the results of robustness of our proposed Cycle-VAE model under geometric and noise attacks. In the end, the conclusion is presented in Section 5.

2. Related Works for VAE and CycleGAN

Currently, there are mainly two popular generation models: Generative Adversarial Nets (GAN) [22] and Variational Automatic Encoder (VAE) [23] and variants based on these two models. In GAN model, a generative model G and a discriminant model D are trained simultaneously. The generative model G captures the distribution of data, while the discriminant model D distinguishes the probability that the sample comes from the training data set rather than from the model G generated. However, there are some drawbacks in the GAN model. For example, it needs to find Nash equilibrium in the training process, which is much more difficult than optimizing an objective function. In addition, it uses a noise z as a prior, but the generative model G cannot control the noise z. That is, the training procedure of GAN is too free, which makes the training process and results of GAN uncontrollable with lack of robustness. In order to stabilize the training process of GAN, researchers have proposed many training techniques from the perspective of model improvement and theoretical analysis, such as Wasserstein GAN (WGAN) [24] and Least Square GAN (LS-GAN) [25].

In addition to the GAN model, Automatic Encoder Neural Network (AENN) is another unsupervised learning algorithm which can be trained by Back Propagation (BP) algorithm [23]. Its biggest characteristic is that the input and output are constrained to be consistent. In fact, a simple self-encoder is a low-dimensional representation of learning data

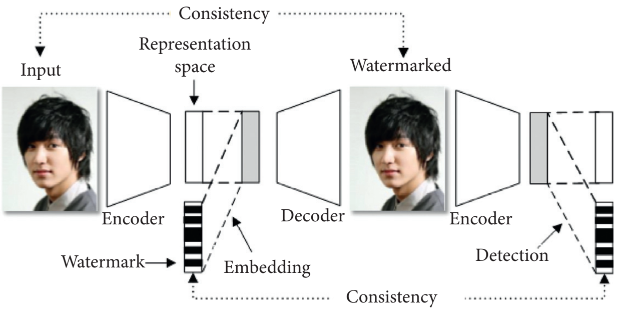

Figure 1: Framework of Cycle-VAE model for image watermarking.

sets, which is similar to Principle Component Analysis (PCA), except that PCA is linear, while self-encoder is nonlinear. However, the performance of standard automatic encoder is limited, mainly because the distribution of output vectors in the hidden layer is unknown and chaotic. Therefore, Kingma and Welling introduced the Variational Automatic Encoder (VAE) [23, 26]. It introduced a hidden variable Z in the hidden layer of standard autoencoder. Through the hidden variable Z, it can generate data automatically and combine the viewpoint of deep learning with that of statistical learning. Besides generating data, VAE can also provide an effective nonlinear data representation approach.

Furthermore, in the image watermarking task, besides an effective data representation approach, we also need to transform the image from the spatial domain to the transform domain that fits for embedding watermarks, which is similar to the image transfer between different domains. The concept of image-to-image translation was first proposed by Hertzman et al. [27]. For pair-matched dataset, several approaches have been proposed to learn a parametric translation function with the help of deep convolutional neural networks in recent years. However, for most real application scenarios, pair-matched data is scarce. To deal with this lack, CycleGAN is a famous model for unpaired image-to-image translation [21]. It is developed from the Conditional GAN (cGAN) [28] and Coupled GAN (CoGAN) [29] with the cycle-consistency loss and its ability of unpaired translation has been proved by many experiments. UNIT-like models [30, 31] are another series of unsupervised image-to-image translation models. They observe the hypothesis of latent space, combine the VAE with CoGAN, and use different codes to represent images content or style. In addition, for another kind of image translation when images are translated from simple to complex or vice versa, Dou et al. proposed an Asymmetric CycleGAN model [32, 33] for improving the CycleGAN model on image translation between domains with different complexity.

This kind of model improves the interpretability of translation model, but it also brings about higher optimization complexity. Therefore, in this paper, we propose a cycle variational autoencoder model to translate spatial domain images into a representation domain and fulfill the watermark embedding, which have low optimization complexity and are robust against noise and geometrical attacks.

3. Proposed Cycle-VAE for Image Watermarking

In this section, we demonstrate our proposed Cycle-VAE model whose goal is to learn a representation space that is suitable for image watermark embedding. To facilitate further illustration of our model, we denote the transformation from image domain to the representation domain as encoder E_I and the transformation from representation domain back to image domain as decoder D_I. The "representation space" or "representation domain" mentioned here denotes the representation feature space in the encoder or decoder network because the explicit explanation of features extracted by the networks is really difficult. In addition, for watermark embedding and detection, we denote the embedding network as E_W and the detection network as D_W. We use $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{m}$ to denote the original image, representation coefficients, and the watermark, respectively. Also we use $\widehat{\mathbf{x}}$, $\widehat{\mathbf{y}}$, and $\widehat{\mathbf{m}}$ to denote the watermarked image, the watermarked representation coefficients, and the detected watermark, respectively. That is, during each step of embedding and detection of the watermark, we have $\widehat{\mathbf{m}}, \widehat{\mathbf{y}} = E_W(\mathbf{m} \mid \mathbf{y})$, $\widehat{\mathbf{x}} = D_I(\widehat{\mathbf{y}})$, and $\widehat{\mathbf{m}} = D_W(E_I(\widehat{\mathbf{x}}))$. As illustrated in Figure 1, our model includes two cycle-consistency losses to constrain the distortion between the original and watermarked images, as $\mathbf{x}$ and $\widehat{\mathbf{x}}$, and between the original and detected watermarks, as $\mathbf{m}$ and $\widehat{\mathbf{m}}$, respectively. More detailed discussion about the model structure and implementations is in the following subsections.

3.1. Model Structure of Cycle-VAE. As shown in Figure 2, our watermarking framework consists of two cycles: an image transformation cycle with encoder and decoder networks, E_I and D_I, respectively, and a watermark embedding cycle with embedding and detection networks, E_W and D_W, respectively. In [34], some theoretical analyses and suggestions on disentangling factors of variation with cycle-consistent structures for variational autoencoders have been provided. Here, in the image transformation cycle (denoted as cycle A), we use VAE loss and identity loss to train E_I and D_I to be an image representation that is suitable for hiding watermark information. In the watermark embedding cycle (denoted as cycle B), we use image and watermark identity loss to enforce the ability of E_W and D_W for watermark hiding and detection, respectively.

In **cycle A**, our image encoding and decoding networks roughly follow the architectural guidelines set forth by [35]. We replace the pooling layers in [35] by using strided convolutions for in-network downsampling and upsampling. Our encoder network E_I comprises five residual blocks [36] with stride 2 convolution, and all nonresidual convolutional layers are followed by batch normalization [37] and ReLU activation layers. All the convolutional layers use 3 × 3 kernels. Therefore, for the encoder network E_I, the

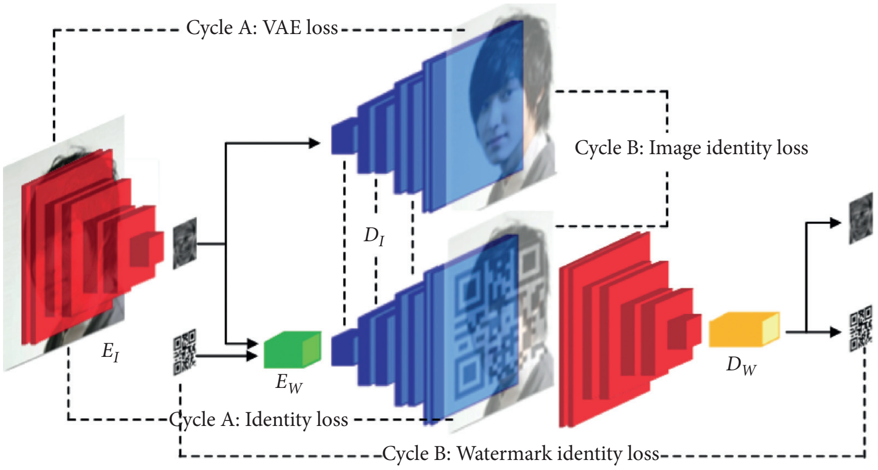
Cycle A: VAE loss
Cycle B: Image identity loss
D_I
E_I
E_W
D_W
Cycle A: Identity loss
Cycle B: Watermark identity loss

Figure 2: A detailed framework of Cycle-VAE model structure for image watermarking.

input and output are color images with shape of 3 × 128 × 128 and representation coefficients with size of 36 × 32 × 32, respectively. Furthermore, the corresponded decoder network D_I consists of 6 upsampling blocks, and each block contains an upsampling layer and a convolutional layer, followed by batch normalization, except for the final output layer, which uses scaled tanh to ensure that the output image has pixels with value between 0 and 255. For the upsampling layer, we use bilinear upsampling with the parameter of scale factor set to be 2. For the convolutional layer, the stride and kernel size are set to be 1 and 3 × 3, respectively. So, the output of decoder network is a three-channel color image with size of 128 × 128.

The loss function in **cycle A** contains a VAE loss and an identity loss of images. Considering some dataset $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ consisting of N i. i. d samples of some continuous or discrete variable $\mathbf{x}$, we assume that the data are generated by some random process, involving an unobserved continuous random variable $\mathbf{y}$. From a coding theory perspective, the unobserved variables y have an interpretation as a latent representation or code. In this paper, VAE is specified by a parametric generative model (as decoder) $p_{D_I}(\mathbf{x} \mid \mathbf{y})$ of the visible variables given the latent variables, a prior $p(\mathbf{y})$ over the latent variables, and an approximate inference model (as encoder) $q_{E_I}(\mathbf{y} \mid \mathbf{x})$ over the latent variables given the visible variables. Then, the marginal likelihood $\log p_{D_I}(\mathbf{x})$ can be rewritten as [26]

$$\begin{aligned}\log p_{D_I}(\mathbf{x}) \geq & -KL\big(p_{E_I}(\mathbf{y} \mid \mathbf{x}), p(\mathbf{y})\big) \\ & + E_{q_{E_I}(\mathbf{y} \mid \mathbf{x})} \log p_{D_I}(\mathbf{x} \mid \mathbf{y}),\end{aligned} \tag{1}$$

where the right-hand side is called the variational lower bound or evidence lower bound (ELBO). However, in general, this lower bound is unattainable. So, when performing maximum-likelihood training, our goal is to optimize the marginal log-likelihood.

$$\arg\max_{D_I, E_I} E\left[\log p_{D_I}(\mathbf{x})\right]. \tag{2}$$

Unfortunately, computing $\log p_{D_I}(\mathbf{x})$ requires marginalizing out $\mathbf{y}$ in $\log p_{D_I}(\mathbf{x}, \mathbf{y})$, which is usually intractable. Thus, based on the inequality in equation (1) and the assumptions used in [23], with the variational Bayes algorithm, our VAE loss can be converted to the following optimization problem:

$$\begin{aligned}\min_{D_I, E_I} \ell_{\mathrm{VAE}}(\mathbf{x}, \mathbf{y}) = E\Big[& KL\big(p_{E_I}(\mathbf{y} \mid \mathbf{x}), p(\mathbf{y})\big) \\ & - E_{q_{E_I}(\mathbf{y} \mid \mathbf{x})} \log p_{D_I}(\mathbf{x} \mid \mathbf{y})\Big].\end{aligned} \tag{3}$$

Because of inequality (1), we still optimize a lower bound to the true maximum-likelihood objective (2). In addition to the VAE loss, which is used for training a good representation of images, we hope that the decoder network D_I can also have the ability of hiding watermarks. So, the identity loss between decoded image and decoded watermarked image is used, which is the squared Frobenius norm of the difference between these two images:

$$\min_{D_I, E_I} \ell_{\mathrm{Ident}-I1}(\mathbf{x}, \mathbf{m}) = \left\| \mathbf{x} - D_I\left(E_W\left(\mathbf{m} \mid E_I(\mathbf{x})\right)\right) \right\|^2. \tag{4}$$

Therefore, during the training process in cycle A, the encoder and decoder networks, E_I and D_I, are generated by solving the problem

$$\left\{\widehat{E}_I, \widehat{D_I}\right\} = \arg\min_{D_I, E_I} \left\{\ell_{\mathrm{VAE}}(\mathbf{x}, \mathbf{y}) + \lambda_1 \ell_{\mathrm{Ident}-I1}(\mathbf{x}, \mathbf{m})\right\}. \tag{5}$$

In **cycle B**, for the embedding network E_W, we simply use 3 blocks, and each contains a 3 × 3 convolutional layer with padding and stride of 1 and a 1 × 1 convolutional layer. The input of embedding network includes an image representation coefficients vector with size of 36 × 32 × 32 and a 1-bit watermark image with size of 32 × 32. The 1-bit watermark image means the value of pixels in watermark can only be 0 or 1. So, we concatenate the watermark to the image coefficients as an additional channel. Then, the watermark and coefficients are sent to the embedding network with an output with size of 36 × 32 × 32 as the watermarked coefficients. For the detection network D_W, we also use a 3-block convolutional neural network but with each block containing a 1 × 1 convolutional layer, a 3 × 3 transpose convolutional layer, and a batch normalization. Finally, to keep the output pixel at 0 or 1, we add a sigmoid activation at the last layer of the detection network. The output of detection network is a one-channel binary image with size of 32 × 32.

The loss function in **cycle B** contains two identity losses: one is for watermarked image and the other is for detected watermark. The identity loss for images is used to train the embedding network E_W for hiding the watermark to a specified image, which is the squared Frobenius norm of the difference between the images with and without a watermark:

$$\min_{E_W} \ell_{\text{Ident}-I2}(\mathbf{y},\mathbf{m})\left\|D_I(\mathbf{y}) - D_I\left(E_W(\mathbf{m}\mid\mathbf{y})\right)\right\|^2. \quad (6)$$

The identity loss for watermark is used to train the detection network D_W for detecting the watermark from embedded image representation coefficients, which is the squared Frobenius norm of the difference between the original watermark and the detected watermark:

$$\min_{D_W} \ell_{\text{Ident}-W}(\hat{\mathbf{x}},\mathbf{m})\left\|\mathbf{m} - D_W\left(E_I(\hat{\mathbf{x}})\right)\right\|^2. \quad (7)$$

Thus, in the training process of cycle B, the embedding and detection networks, E_W and D_W, are generated by minimizing the following objective function:

$$\left\{\widehat{E_W},\widehat{D_W}\right\} = \arg\min_{E_W,D_W}\left\{\ell_{\text{Ident}-I2}(\mathbf{y},\mathbf{m}) + \lambda_2\ell_{\text{Ident}-W}(\hat{\mathbf{x}},\mathbf{m})\right\}. \quad (8)$$

3.2. Implementation Details. Since we use an image representation model, as E_I and D_I, to fulfill our watermark imbedding task, the images should belong to one category rather than any kinds of natural images. So, we study embedding 1-bit QR-code watermark into a specified kind of images, that is, face images. For the image training data, we use 200, 000 24-bit images with size of 128 × 128 from CelebA dataset. For the QR-code watermark data, we also randomly generated 200, 000 1-bit binary images. However, it should be noted that the image and QR-code are not in one-to-one correspondence; they are randomly selected and matched.

For the parameters setting, we set $\lambda_1 = 0.001$ and $\lambda_2 = 0.2$ in equations (5) and (8), respectively, in our training process. We use adaptive moment estimation (Adam) solver [38] with a batch size of 64. All networks were trained from scratch with a learning rate of 0.0001. We keep the same learning rate for the first 100 epochs and linearly decay the rate to zero over the next 100 epochs.

4. Experimental Results and Discussions

To show the effectiveness of our method, we give some comparison results on face image watermarking in this section. We randomly selected 10 face images from CelebA dataset, which were not in the training set. Five of these testing images are shown in the top row in Figure 3. Our watermark images are randomly generated binary QR-code images with size of 32 × 32, which were also not used in the training process. The network was trained using a GPU, NVIDIA GTX 1080Ti, under the PyTorch 0.4.1 environment for two days followed by the instructions as in Section 3.2. The performance of our proposed watermark algorithm is measured from two aspects: visual imperceptibility and robustness against noise and geometric affine transform attack. We compared our algorithm to the state-of-the-art block-based watermark algorithm in the quaternion discrete Fourier transform (QDFT) [39].

For a good watermarking algorithm, the embedded watermarking information should not be visible. So, we use peak signal-to-noise ratio (PSNR) and Structural Similarity (SSIM) [40] to measure the invisibility of the watermarked image, which are defined as

$$\text{PSNR}(\mathbf{x},\hat{\mathbf{x}}) = 10\log_{10}\frac{255^2 \times 3 \times M \times N}{\|\mathbf{x} - \hat{\mathbf{x}}\|^2}, \quad (9)$$

$$\text{SSIM}(\mathbf{x},\hat{\mathbf{x}}) = \frac{\left(2\mu_x\mu_{\hat{\mathbf{x}}}+c_1\right)\left(2\sigma_{x\hat{\mathbf{x}}}+c_2\right)}{\left(\mu_x^2 + \mu_{\hat{\mathbf{x}}}^2+c_1\right)\left(\sigma_x^2 + \sigma_{\hat{\mathbf{x}}}^2+c_2\right)}, \quad (10)$$

respectively. In equations (9) and (10), $\mathbf{x}$, $\hat{\mathbf{x}}$, $\mu_{\mathbf{x}}$, and $\sigma_{\mathbf{x}}$ denote the original image, the watermarked image, the mean, and the standard deviation of the image, respectively. From equations (9) and (10), we can find that these two indices, that is, PSNR and SSIM, can reflect, respectively, the pixel level and structural difference of two images, which means the higher PSNR and SSIM values, the smaller the difference between two images and the better the visual invisibility of watermarks. Ordinarily, when the PSNR (resp., SSIM) value is greater than 35 dB (resp., 0.95), we cannot distinguish the difference between two images by our naked eyes directly. Figure 3 shows five original images (top row) and their watermarked equivalents (bottom row), from which we can find that, in the watermarked images, the embedded watermarking information is invisible; that is, our proposed algorithm has strong imperceptibility. For quantitative comparison, the PSNR and SSIM values of ten watermarked test images embedded by our proposed model and the QDFT algorithm are shown in Table 1. Note that the PSNR and SSIM values of these ten test images in Table 1 are derived by averaging watermarked images with embedding five different watermarks as shown in Figure 4. Besides the index comparison about the visual imperceptibility, the computational efficiency is another important index for practical applications. Since the QDFT is a traditional transform based algorithm, its computational complexity is $O(n^2)$, where n is the number of pixels of an image. However, our proposed watermark algorithm is a deep

Figure 3: Visual impact comparison before and after watermark embedding. (a) The top row shows the original test images from CelebA dataset and (b) the bottom row shows the watermarked test images.

Table 1: Comparison of PSNR and SSIM of watermarked images derived by different methods.

Image ID		200863	201822	200364	200528	200292	200273	201739	201160	201100	200290
QDFT	PSNR	32.09	33.52	37.10	31.59	33.95	32.09	36.01	33.73	32.74	34.33
	SSIM	0.948	0.932	0.978	0.956	0.955	0.925	0.959	0.935	0.938	0.965
Proposed	PSNR	32.91	33.99	37.91	32.54	34.44	32.97	36.85	34.14	33.23	34.79
	SSIM	0.952	0.942	0.979	0.976	0.964	0.936	0.974	0.951	0.959	0.979

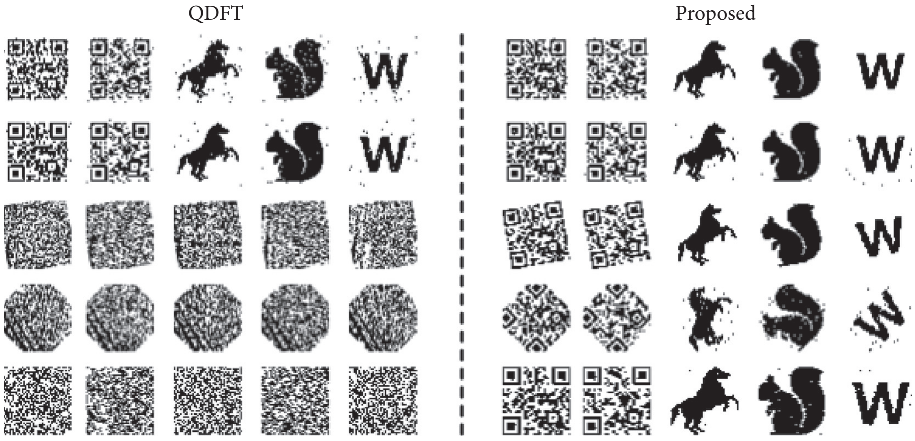

Figure 4: Comparison of the robustness of QDFT and our proposed algorithm against the noise and geometric attacks. From top to bottom are attacks including Gaussian noise, salt and pepper noise, rotation 10°, rotation 45°, and zooming 20%. In each side of the image, from left to right are images with ID 200273, 200290, 201100, 201160, and 201739.

neural network based model whose computational complexity is hard to calculate explicitly. But, for numerical evaluation, our model can process images with size of 128 × 128 pixels at 40 FPS (frames per second) under our experimental environment (NVIDIA GTX 1080Ti, PyTorch 0.4.1). Although the computational time of the proposed model will increase linearly according to the size of input images, with the help of CUDA and AI chip technology, the proposed model is still possible for practical applications.

To show the robustness of our proposed watermark approach, we also conduct the noise and geometric attacks experiment. For noise attacks, we add two kinds of noise, that is, Gaussian noise and pepper noise, to the watermarked images. For geometric attacks, we apply two types of affine transformations, that is, rotation and resize, to the watermarked images. After these attacks, we use the detection network D_W to extract the 1-bit watermark image from the attacked images. Figure 5 shows an example of noise and

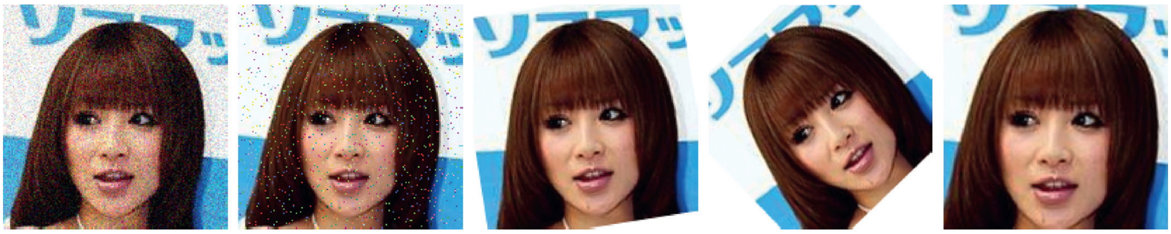

Figure 5: Demonstration of noise and geometric attack used in the experiments. From left to right are Gaussian noise 0.05, salt and pepper noise 0.02, rotation 10°, rotation 45°, and zooming 20% attacks.

geometric attack to the test image. The parameters of Gaussian noise and pepper noise are set to be 0.05 and 0.02 under the scale with image pixel values normalized between 0 and 1. The extracted watermark images derived by QDFT and our proposed algorithm are shown in Figure 4.

From Figure 4, we can find that both QDFT and our proposed algorithm are quite robust against the Gaussian noise and salt & pepper noise attacks. But it seems that our proposed algorithm can extract more clean watermarks compared to the QDFT algorithm. This might be because the autoencoder network itself has a certain ability of image denoising. Then, for the geometric attacks, including rotation and zooming, our proposed algorithm can still extract the correct watermark images, while QDFT algorithm cannot derive satisfactory results. This is because we use the entire image as the input to train our autoencoder and embedding network, but QDFT algorithm is a block-based watermark algorithm. Thus, our proposed algorithm is also quite robust against the geometric attacks.

As shown in Table 1, we can find that, compared to the QDFT method, the proposed approach can achieve an average 0.5 ~ 1.0 dB improvement in PSNR in the aspect of visual quality of watermarked images. Meanwhile, as shown in Figure 4, the proposed approach is more robust than the QDFT method in the aspect of geometric attacks for watermarked images. Therefore, the proposed approach is more robust to attacks and better for watermark information hiding, which means that it has great potential value for practical applications.

5. Conclusion

We propose a new framework for robust image watermarking embedding using cycle variational autoencoder networks. Since the VAE model can learn an abstract representation of a specific kind of images, we use face images to validate our proposed algorithm in this paper. In addition, we train a convolution network to embed a 1-bit watermarking image into the face image in the representation space. Unlike block-based algorithm, that is, QDFT, and DCT-based techniques, our algorithm processes the input image entirely. Therefore, as validated in the experimental section, the proposed algorithm can preserve a better visual quality and is more robust against the noise and geometric attacks compared to those block-based algorithms. However, since we process the input image as a whole, the size of our network will be too big to be practically used for large images directly. So, developing lightweight autoencoder network for large images is an important issue that warrants further study. Moreover, in many real applications, we need to embed watermark information to many different kinds of images, not just face images. That means, compared to the traditional transform-based watermark algorithm, the versatility of our proposed deep learning based model needs to be tested and discussed. To extend our watermark embedding approach to natural images is another issue that merits further study.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61773206).

References

[1] H. Liu, H. Kou, C. Yan, and L. Qi, “Link prediction in paper citation network to construct paper correlated graph,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[2] H. Liu, H. Kou, C. Yan, and L. Qi, “Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph,” *Complexity*, vol. 2020, pp. 1–15, Article ID 2085638, 2020.

[3] L. Qi, W. Dou, W. Wang, G. Li, H. Yu, and S. Wan, “Dynamic mobile crowdsourcing selection for electricity load forecasting,” *IEEE Access*, vol. 6, pp. 46926–46937, 2018.

[4] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, “Energy efficient dynamic offloading in mobile edge computing for internet of things,” *IEEE Transactions on Cloud Computing*, p. 1, 2019.

[5] X. Xu, S. Fu, W. Li, F. Dai, H. Gao, and V. Chang, “Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–11, 2020.

[6] X. Xu, H. Cao, Q. Geng, X. Liu, F. Dai, and C. Wang, “Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment,” *Concurrency and Computation-Practice & Experience*, 2020.

[7] X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, and S. Yu, "A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems," *Future Generation Computer Systems*, vol. 105, pp. 789–799, 2020.

[8] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

[9] Q. Wei, H. Shao, and G. Zhang, "Flexible, secure, and reliable data sharing service based on collaboration in multicloud environment," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–16, Article ID 5634561, 2018.

[10] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified locality-sensitive hashing-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, 2020.

[11] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.

[12] C. Zhou, Li Ali, A. Hou, Z. Zhang, Z. Zhang, and F. Wang, "Modeling methodology for early warning of chronic heart failure based on real medical big data," *Expert Systems with Applications*, vol. 151, Article ID 113361, 2020.

[13] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, and W. Dou, "A blockchain-powered crowdsourcing method with privacy preservation in mobile environment," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1407–1419, 2019.

[14] S. D. Lin and C.-F. Chen, "A robust DCT-based watermarking for copyright protection," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 415–421, 2000.

[15] N. Kashyap and G. Sinha, "Image watermarking using 3-level discrete wavelet transform (DWT)," *International Journal of Modern Education and Computer Science*, vol. 4, pp. 1–7, 2012.

[16] X. Hu, S. Peng, and W. Hwang, "EMD revisited: A new understanding of the envelope and resolving the mode-mixing problem in am-fm signals," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1075–1086, 2012.

[17] D. Volkhonskiy, I. Nazarov, B. Borisenko, and E. Burnaev, "Steganographic generative adversarial networks," in *Proceedings of the NIPS 2017 Workshop on Adversarial Training*, pp. 201–208, Long Beach, CA, USA, November 2017.

[18] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "Secure steganography based on generative adversarial networks," in *Proceedings of the Advances in Multimedia Information Processing - PCM 2017*, pp. 534–544, Harbin, China, September 2017.

[19] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.

[20] S.-M. Mun, S.-H. Nam, H. Jang, D. Kim, and H.-K. Lee, "Finding robust domain from attacks: A learning framework for blind watermarking," *Neurocomputing*, vol. 337, pp. 191–202, 2019.

[21] J. Zhu, T. Park, P. Isola, and A. A. Efros, ""Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, Venice, Italy, October 2017.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2672–2680, Montreal, Canada, December 2014.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," http://arxiv.org/abs/1312.6114.

[24] A. Martin, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223, Sydney, Australia, August 2017.

[25] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821, Venice, Italy, October 2017.

[26] L. Mescheder, S. Nowozin, and A. Geiger, "Adversarial variational bayes: unifying variational autoencoders and generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 2391–2400, Sydney, Australia, August 2017.

[27] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 327–340, Los Angeles, CA, USA, August 2001.

[28] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *Computer Science*, pp. 2672–2680, 2014.

[29] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proceedings of the 29th Advances in Neural Information Processing Systems*, pp. 379–390, Barcelona, Spain, December 2016.

[30] M.-Yu Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proceedings of the 30th Advances in Neural Information Processing Systems*, pp. 700–708, Long Beach, CA. USA, December 2017.

[31] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1–18, Munich, Germany, September 2018.

[32] H. Dou, C. Chen, X. Hu, and S. P. Peng, "Asymmetric cyclegan for unpaired NIR-to-RGB face image translation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1757–1761, Brighton, United Kingdom, May 2019.

[33] H. Dou, C. Chen, X. Hu, L. Jia, and S. Peng, "Asymmetric cyclegan for image-to-image translations with uneven complexities," *Neurocomputing*, vol. 415, pp. 114–122, 2020.

[34] A. H. Jha, S. Anand, M. Singh, and V. S. R. Veeravasarapu, "Disentangling factors of variation with cycle-consistent variational auto-encoders," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 829–845, Munich, Germany, September 2018.

[35] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–16, San Juan, Puerto Rico, May 2016.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[37] Sergey Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference*

on Learning Representations (ICLR), San Diego, CA, USA, May 2015.

[39] X.-Y. Wang, C.-P. Wang, H.-Y. Yang, and P.-P. Niu, "A robust blind color image watermarking in quaternion fourier transform domain," *Journal of Systems and Software*, vol. 86, no. 2, pp. 255–277, 2013.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.