IET Image Processing

Research Article

Two-stage visible watermark removal architecture based on deep learning

ISSN 1751-9659
Received on 23rd March 2020
Revised 24th September 2020
Accepted on 19th October 2020
E-First on 16th February 2021
doi: 10.1049/iet-ipr.2020.0444
www.ietdl.org

Pei Jiang[1]*, Shiwen He*[1,2] ✉*, Hufei Yu*[1]*, Yaoxue Zhang*[1]

[1]*School of Computer Science and Engineering, Central South University, Changsha 410083, People's Republic of China*

[2]*Purple Mountain Laboratories, Nanjing 210096, People's Republic of China*

✉ *E-mail: shiwen.he.hn@csu.edu.cn*

Abstract: With the rapid development of the Internet, watermarks are widely used in images to protect copyright. This implies that the robustness of watermark is very important. In recent years, there have been some studies to evaluate watermark performance by removing the watermark. Among them, some methods need to mark the watermark position in advance, and some require multiple images with the same watermark. Moreover, when the colour of thewatermark is similar to that of the background, the existing methods can hardly remove the watermark from the watermarked image. In the proposed work, the authors presented a watermark removal structure consisting of watermark extraction and image inpainting to address the aforementioned issues. In particular, the extraction network is used to extract the watermark in the watermarked image, and the inpainting network is used to inpainting image for a better watermark removal image, respectively. Finally, the authors train and test the developed network architecture by constructing two data sets, i.e. white watermarked image data set (WW-data set) and colour watermarked image data set (CW-data set). The proposed method not only has better performance on the WW-data set than the current latest methods (on the CW-data set, other methods have almost failed) but also effectively removes the watermarks.

1 Introduction

With the rapid development of the Internet, image copyright has become an increasingly prominent problem. Usually, people add a visible or invisible watermark to the image to claim the image copyright [1]. However, the watermarks are often attacked and destroyed, such that the image copyright is stolen. In order to evaluate and improve the robustness of visible watermarks, in recent years, more attention has been paid to the research of visible watermark removals.

For visible watermark removal, there are two main problems with traditional methods. The first is the demand of manually marking the watermarked areas and non-watermarked areas in the watermarked image before performing watermark removal [2]. Second, in order to efficiently remove the watermark in the watermarked image, multiple wartermarked images that have the same watermark are needed for watermark removal [3]. In recent years, some deep learning-based visible watermark removal methods are proposed to solve the aforementioned two problems [4, 5]. Cheng *et al.* [6] proposed a watermark removal method, which detects the watermark position by using the existing object detection methods first, and then removes the watermark through a U-Net network. However, the results obtained from their model were not ideal due to the simple network structure they conducted and an inefficient loss function they exploited. Li *et al.* [5] introduced adversarial loss to the loss function defined in [6], such that the generated image becomes more realistic. However, for a white watermarked image, as show in Fig. 1, there are still residual watermark after removing watermark. Furthermore, the method proposed by Li and colleagues has no ability to remove the visible watermark whose colour is very similar to the background colour, as shown in Fig. 2.

Motivated by these observations, in our work, we proposed a visible watermark removal network architecture based on conditional generative adversarial networks (CGANs) [7] and least-squares generative adversarial networks (LSGANs) [8]. This architecture includes two main components, i.e. watermark extraction and image inpainting. Specifically, first, the watermark in a watermarked image is extracted through the extraction network, which mainly focuses on the watermark areas of the watermarked image. Then, a preliminary watermark removal image is obtained by subtracting the extracted watermark from the watermarked image. Finally, the preliminary watermark removal

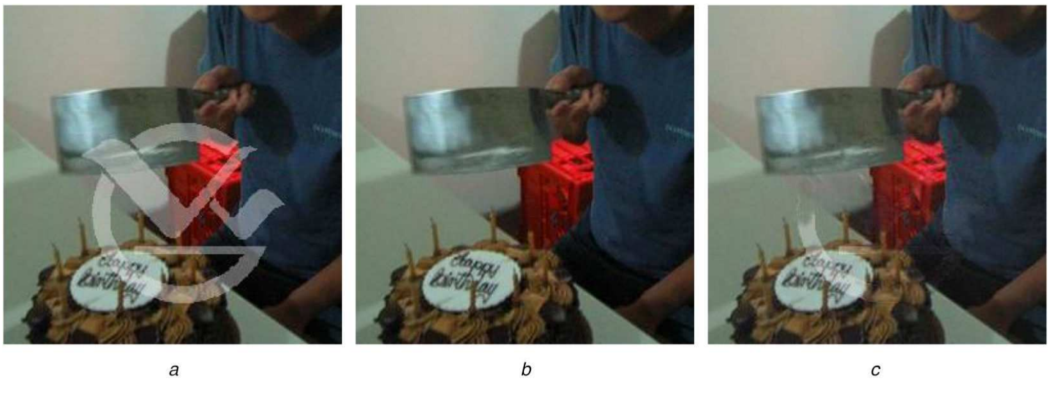

Fig. 1 *Example results of Li et al.'s model on visible white watermarked images*
(a) Image with visible common white watermark, ***(b)*** Original watermarkless image, ***(c)*** Watermark removal image [5]

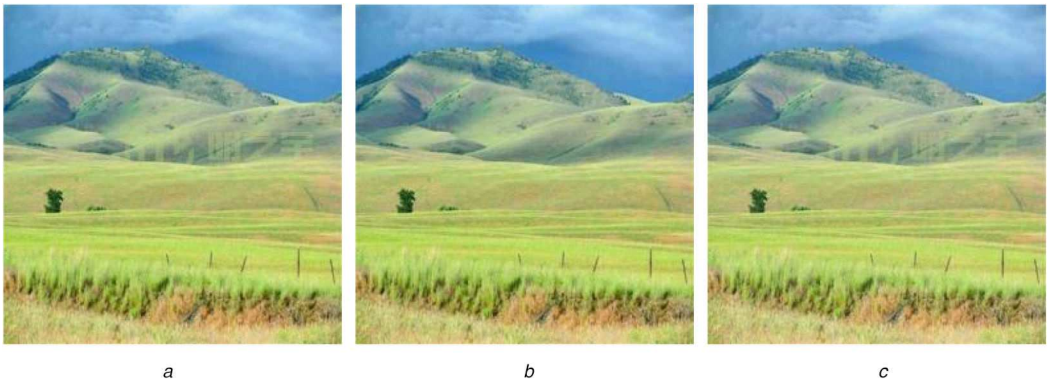

Fig. 2 *Example results of Li et al.'s model on visible colour watermarked images*
(a) Image with visible colour watermark, ***(b)*** Original watermarkless image, ***(c)*** Watermark removal image [5]

image is input into the inpainting network to obtain a more consistent and authentic watermark removal image. To the best of the author's knowledge, it is the first deep learning-based watermark removal method combining the methods which focus on watermark region of the watermarked image with the methods of image inpainting to remove watermark in the watermarked image. Compared with the traditional methods, the proposed method does not need to manually select the watermark area, and can also effectively remove the watermark for a single watermarked image. In addition, the end-to-end training of our network architecture can be easily implemented than that of the network architecture developed in [6]. Our proposed method also has stronger capability than the method given in [5] in terms of the watermark removal, which directly inpaints the area covered by the watermark without explicitly learning the features of the watermark. In order to evaluate the proposed network architecture's performance, we established two watermarked image data sets, i.e. a white watermarked image data set (WW-data set) and a colour watermarked image data set (CW-data set). Experimental results show that the proposed method has the ability to remove the residual watermarks and to remove colour watermarks that are close to the background colour.

2 Related work

In the existing literature, some existing watermark removal methods focus on extracting watermarks from the watermarked image by learning watermark features [3]. In addition, some methods focus on the entire watermarked image, in which the watermarked region is regarded as a missing region and is filled via image inpainting [2].

2.1 Watermark removal

In recent years, the studies on visible watermark removal have attracted extensive concern in both academia and industry. Huang and Wu [2] treated the watermarking attack as an image inpainting problem requiring the attacker to select the watermark region to be repaired. Pei and Zeng [9] divided the visible watermark removal into three stages: watermarked region segmentation, reference image generation, and image recovery. Watermarked region segmentation requires manually indicating watermarked and non-watermarked pixels. Both watermark removal models need to manually label the watermarked and non-watermarked regions before removing watermark [2, 9]. To release the demand on manually labelling watermark regions, Santoyo-Garcia *et al.* [10] proposed an automatic visible watermark detection method. Firstly, they decompose the visible watermark into a structure image and a texture image using Total Variation method. Then, they extract the watermark edge from the structure image. Finally, they use the inpainting method developed in [2] to remove the watermark. However, the inpainting attack is only effective to remove simple watermarks composed of thin lines or symbols. Xu *et al.* [3] hypothesised that the watermark images in the test set have the same resolution and the same watermark region, in order to detect the watermark region via statistical method. Then, the watermark region is regarded as a missing region and is filled using image inpainting methods. Dekel *et al.* [11] proposed a general multi-image matting algorithm to remove the watermark with high precision when the original image was added a watermark with a consistent manner. The watermark removal methods in [3, 11] are based on multiple images having the same watermark.

With the application and development of deep learning [4] in the field of image processing, some scholars have applied deep learning to watermark removal. The first deep learning-based watermark remove work was finished by Cheng *et al.* in [6]. In their method, the watermark is first detected via an object detection method and then is removed by using the image-to-image conversion model. Although Cheng and colleagues solved the two issues raised earlier, however, it is difficult to train the generator only by pixel loss. Li *et al.* [5] proposed a watermark processing framework based on CGAN with a loss function including adversarial loss and content loss. The result of watermark removal is more photo-realistic for the white watermarked image but is not good enough for colour watermarked images.

2.2 Image inpainting

The existing image inpainting methods can be divided into two categories: non-learning image inpainting methods and learning-based image inpainting methods. The former is based on low-level features, while the latter exploits high-level semantic information.

Non-learning image inpainting methods include diffusion-based image inpainting methods and patch-based image inpainting methods. Diffusion-based image inpainting calculates the unknown pixels from neighbouring known pixels [12]. This method is only suitable for inpainting small and narrow missing regions. The patch-based image inpainting method iteratively inpaints the image by searching for similar patches from non-missing regions [13]. This method is not only computationally expensive but also cannot semantically inpaint image.

Learning-based image inpainting methods are usually based on convolutional networks and generative adversarial network (GAN) [14–16]. The first neural network for image inpainting is context encoder, which uses an encoder as a generator to predict the missing region through adversarial training [17]. However, it does not fill well with texture details. Yang *et al.* [18] added a texture network on the basis of Context Encoder to obtain clearer results at the cost of large number of computations. Yu and Koltun [19] and Iizuka *et al.* [20] replaced the channel convolution layer in the Context Encoder with dilated convolution, and used jointly local and global discriminators to make the inpainting result more realistic. Yu *et al.* [21] proposed a two-stage network architecture via adding a contextual attention module to make the repair region more coherent, but ignoring the correlations between patchs inside the missing region. In order to solve this problem, Liu *et al.* [22] adopted the coherent semantic attention to evaluate the correlation of deep features in the missing region, and introduced a feature patch discriminator to obtain a better repair effect.

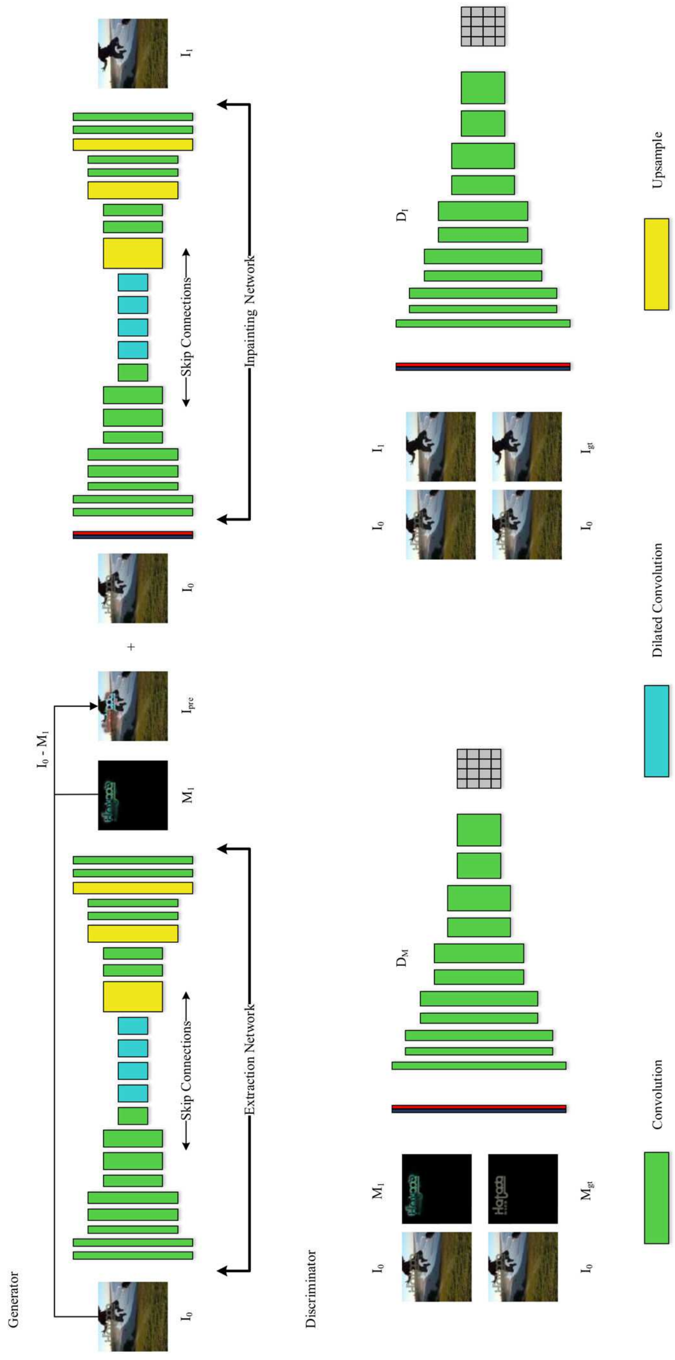

Fig. 3 *Structure of the proposed model*

3 Method

In this section, we first introduce two watermarked image data sets. One is WW-data set in which each image added a white watermark, and the other is CW-data set which is a CW-data set. Specifically, each watermarked image is obtained by adding a watermark to an original image. For the CW-data set, the colour of watermarks in the watermarked image is determined by the average pixel value of the region to be watermarked in the original image. Second, we narrate network architecture based on conditional LSGAN [8] (CGAN and LSGAN), which is mainly composed of two coordinated networks, i.e. extraction network and inpainting network. The network architecture is shown in Fig. 3. Finally, we describe the loss function adopted by the proposed model.

3.1 Data sets

Each watermarked image data set contains 53,300 watermarked images, which are generated from 53,300 watermarkless images and 100 kinds of original watermarks. The 533,00 watermarkless images obtained from two public data sets, i.e. PASCAL VOC2012 and places2 [23]. The 100 kinds of original watermarks are logos of brands, websites etc. and collected from the Internet. Some watermark examples are shown in Fig. 4*a*. To get the WW-data set and CW-data set, first, we divide equally the 53,300 watermarkless images into 100 groups (80 groups are the training set and 20 groups are the test set.). Then, the images in each group are added with the same watermark, and the images in different groups are added with a different watermark. The only difference between the two data sets is the colour of watermarks.

As shown in Fig. 5, when combining each watermarkless image (real image) with a watermark, we first determine, randomly, the size of watermark, which cannot exceed the size of real image. Then, we created a blank image with the same size as the real image and placed the determined watermark in any position of the blank image. In addition, we need to set the colour of the watermark. Specifically, for the WW-data set, the colour of the watermark is white. For the CW-data set, the colour of the watermark is determined by the average value of pixels included in the corresponding region of real image. At this point, we get the

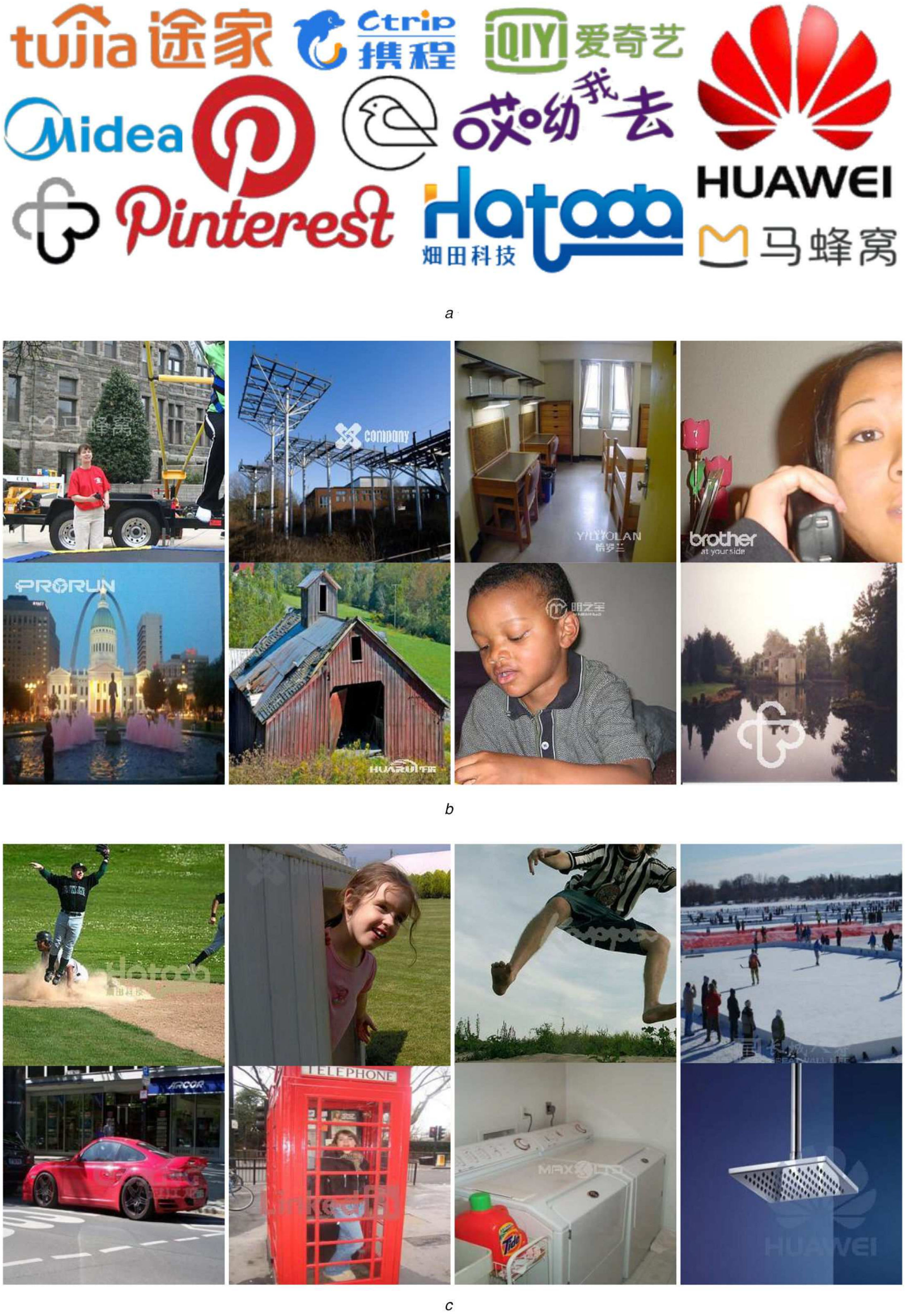

Fig. 4 *Example of our data sets*
(a) Some original watermarks, ***(b)*** Examples of the WW-data set, ***(c)*** Examples of the CW-data set

real watermark M_{gt}. Finally, the watermark M_{gt} is added to the real image I_{gt} to obtain a watermarked image. Figs. 4*b* and *c* illustrate some examples of the WW-data set and the CW-data set, respectively. In Fig. 4*c*. One can see the watermark colour is very similar to the background image.

3.2 Architectures

In order to completely remove watermark and to ensure the authenticity of the whole image, we construct a two-stage visible watermark removal model by extracting the features of watermark and image, respectively. The first stage is to extracts the watermark through identifying the watermark feature and then removes them. The second stage is image inpainting, which directly obtains watermarkless image through encoding and decoding from the first stage watermark removal image. In particular, the generator of the proposed model includes two parts, i.e. Extraction Network and Inpainting Network, as shown in Fig. 3, in which two discriminators are used to identify the generated images and real images.

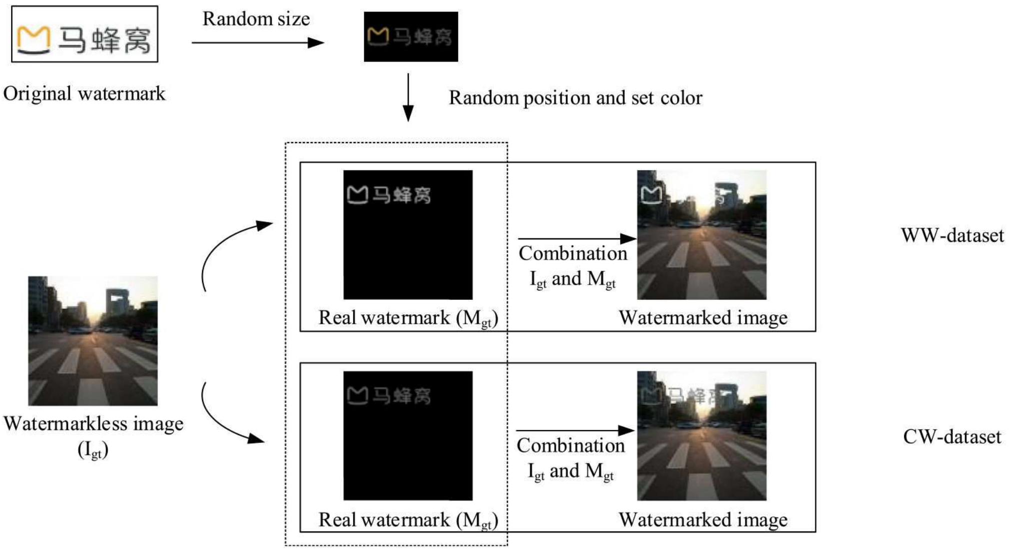

Fig. 5 *Process of watermarking M_{gt} to the real image I_{gt}*

Generator: The extraction network is based on U-Net [24], which consists of three encoder layers, three decoder layers, and four dilated convolution layers that connect the encoder and decoder. The encoder is used to extract significant features of inputting images. The decoder is designed to decode the output of dilated convolution. The dilated convolution layers aim to expand the size of the receptive field. Specifically, each encoder layer includes three convolution layers, in which the first two convolution layers with stride equals 1 and the last convolution layer with stride equals 2 to achieve downsampling. The corresponding decoder layer includes an upsampling layer and two convolution layers with stride equals 1. In addition, there is an additional convolution layer with stride equals 1 at the end of the last decoder layer to make the output size consistent with the size of the input image. The network structure of inpainting network is the same as that of extraction network.

In summary, the extraction network mainly focuses on the watermark areas of watermarked image, taking watermarked image I_0 as input and extracting watermark M_1 as output. After getting watermark M_1, we subtract watermark M_1 from watermarked image I_0 to get a preliminary watermark removal image I_{pre}. The inpainting network focuses on the entire watermarked image, which takes preliminary watermark removal image I_{pre} and watermarked image I_0 as input and outputs a watermarkless image I_1. As far as we know that it is the first visible watermark removal model that is based on deep learning and combines the methods focusing on the watermark region of watermarked image and the methods of image inpainting.

Discriminator: In order to make the extracted watermark closer to the real watermark, and the generated watermarkless image closer to the real image, we use two conditional PatchGAN discriminators, D_M and D_I [25]. Unlike traditional discriminators, they can judge the true probability of each region in the image. Specifically, D_M only has effect on extraction network, and D_I has effect on extraction network and inpainting network, because we use end-to-end training. The input of each discriminator is a pair of images, and the output is a map of real/fake probabilities. They have the same structure and consist of ten convolutional blocks and a dense layer at the end. Each block contains a convolutional layer concatenated with activation and batch normalisation.

The proposed model overcomes the shortcomings of traditional watermark removal methods. By the extraction network, the proposed model can automatically detect the watermark position, unlike [2] where the watermark area needs to be manually marked. In addition, some images added with the same watermark are required to extract the watermark features and then remove the watermark [3]. Meanwhile, the proposed model can still effectively remove the visible watermark for a single watermarked image. Compared with the method based on deep learning, the method developed in [6] relies on the existing object detection method to detect the watermark position and has a simple structure and an inefficient loss function in its watermark removal network. The proposed model, through end-to-end training, can directly output the watermarkless image for the input watermarked image, and the output watermarkless image is more realistic due to the increased adversarial loss. Furthermore, compared with [5], the proposed model pays more attention to the watermark region and obtains more stable performance.

3.3 Loss function

In this work, we adopt reconstruction loss and adversarial loss to guide the proposed model for training. Specifically, the reconstruction loss includes ℓ_2 loss and perceptual loss. The ℓ_2 loss is used to evaluate the mean square of the pixel distance between the generated image and the real image. Different from the ℓ_2 loss, the perceived loss is feature level and is obtained via convolution, which can capture the image's semantic information and make the generated image more realistic. In addition, the adversarial loss is used to improve the subjectively perceived image quality.

Reconstruction Loss: In the proposed model, the ℓ_2 loss includes L_M and L_I. In particular, L_M is the ℓ_2 distance between the real watermark and generated watermark. L_I is the ℓ_2 distance between the real watermarkless image and generated watermarkless image. L_M is given by

$$L_M(f) = \| f(I_0) - M_{gt} \|_2, \tag{1}$$

Where I_0 is a watermarked image, $f(\cdot)$ denotes the output of extraction network, M_{gt} is the real watermark. L_I is defined as

$$L_I(f,g) = \| g(I_0 - f(I_0)) - I_{gt} \|_2, \tag{2}$$

where, $g(\cdot)$ denotes the output of inpainting network, I_{gt} is the real watermarkless image. In order to extract high-level features and effectively capture the semantic information of image, we use the perceptual loss as part of reconstruction loss, defined as

$$L_{\text{per}}^{\Phi,j}(f,g) = \frac{1}{C_jH_jW_j}\| \Phi_j(g(I_0 - f(I_0))) - \Phi_j(I_{gt}) \|_2, \tag{3}$$

where $\Phi_j(\cdot)$ represents the jth layer feature map of an convolution transformation pre-trained for extracting advanced features. In our work, $\Phi_j(\cdot)$ corresponds to feature map from layer Relu2-2 of the VGG19 network pre-trained on ImageNet data set [26]. The size of feature map is $C_j \times H_j \times W_j$.

Adversarial Loss: extraction network and inpainting network are both trained based on conditional LSGAN. The conditional LSGAN objective function defined as follows [8]

$$\begin{aligned}\min_{D} V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{x \sim p_{\text{data}}(x)}\left[(D(x|\Phi(y)) - 1)^2\right] \\ + \frac{1}{2}\mathbb{E}_{x \sim p_{\text{data}}(x)}\left[(D(G(z)|\Phi(y)))^2\right],\end{aligned} \tag{4}$$

$$\min_{G} V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{z \sim p_{\text{data}}(z)}\left[(D(G(z)|\Phi(y)) - 1)^2\right], \tag{5}$$

where D is the discriminator, G is the generator, x is the real data, z is the input of the generator, and y is the condition. Here, the label of the real data is 1, and the label of the generated data is 0. Through training, they hope that the discriminator can accurately distinguish between real data and generated data, that is, the discriminator can output 1 for real data and 0 for generated data. On the other hand, they hope that the data generated by the training generator can fool the discriminator, that is, the discriminator can output 1 when the input is generated data.

In the proposed watermark removal model, the input of discriminator of extraction network (respectively, extraction network and inpainting network) needs not only the output of generator ($f(I_0)$ (respectively., $g(I_0 - f(I_0))$)), but also I_0 as condition. In addition, the discriminators we adopt are patch-based, so the output of the discriminator is a matrix rather than a single number. The objective functions of the discriminator D_M and extraction network are defined as follows and L_f is the adversarial loss for extraction network:

$$\begin{aligned}\min_{D_M} L_{D_M} = \min_{D_M} \frac{1}{2}\mathbb{E}_{M_{gt} \sim P_{M_{gt}}, I_0 \sim P_{I_0}}\left[D_M(I_0, M_{gt}) - I\right]^2 \\ + \frac{1}{2}\mathbb{E}_{I_0 \sim P_{I_0}}\left[D_M(I_0, f(I_0))\right]^2.\end{aligned} \tag{6}$$

$$\min_{f} L_f = \min_{f} \frac{1}{2}\mathbb{E}_{I_0 \sim P_{I_0}}\left[D_M(I_0, f(I_0)) - I\right]^2. \tag{7}$$

Because D_M is a patch based discriminator, its output is a matrix. Each element of the matrix represents the true probability of the corresponding region of input image. when the element value is close to 1, the discriminator considers it to be a real image, and the element value is close to 0 indicates that the discriminator thinks it is the generated image. In (6) and (7), I is a matrix whose elements are all one. f wants to trick the D_M into thinking that the generated watermark is a real watermark, which is achieved by minimising (7). In addition, D_M can distinguish the generated fake watermark from the real watermark by minimising (6). In the end, the discriminator D_M cannot distinguish whether the generated watermark is true or false, and the f can effectively extract the watermark. Similar to (6) and (7), The objective functions of the discriminator D_I and the entire generator G (extraction network and inpainting network) are defined as follows and $L_{f,g}$ is the adversarial loss for extraction network and inpainting network:

$$\begin{aligned}\min_{D_I} L_{D_I} = \min_{D_I} \frac{1}{2}\mathbb{E}_{I_{gt} \sim P_{I_{gt}}, I_0 \sim P_{I_0}}\left[D_I(I_0, I_{gt}) - I\right]^2 \\ + \frac{1}{2}\mathbb{E}_{I_0 \sim P_{I_0}}\left[D_I(I_0, g(f(I_0)))\right]^2,\end{aligned} \tag{8}$$

$$\min_{f,g} L_{f,g} = \min_{f,g} \frac{1}{2}\mathbb{E}_{I_0 \sim P_{I_0}}\left[D_I(I_0, g(f(I_0))) - I\right]^2. \tag{9}$$

Similarly, in the end, the discriminator D_I cannot distinguish whether the generated watermarkless image is true or false, and the generator can effectively removal watermark.

Our total loss function is:

$$L_G = \underbrace{\lambda_1 L_M + \lambda_2 L_I + \lambda_3 L_{\text{per}}^{\Phi,j}}_{\text{Reconstruction Loss}} + \underbrace{\lambda_4 L_f + \lambda_5 L_{f,g}}_{\text{Adversarial Loss}}, \tag{10}$$

where λ_1, λ_2, λ_3, λ_4, and λ_5 are the corresponding weight of different losses, which are used to measure the impact of different losses on the entire loss function. The value of λ_1, $_2$, λ_3, λ_4, λ_5 needs to be set manually during the experiment.

4 Experiments

In order to evaluate the performance of the proposed watermark removal method, we perform the training and testing on the data sets mentioned in Section 3.1. The experiments are mainly divided into two parts, one is four control experiment on CW-data set, and the other is to compare with other existing visible watermark removal methods on two watermarked image data sets. In our experiments, under the limitation of server performance and in order to obtain a relatively accurate gradient, the batch size is set to be 4. In addition, we train the model via 10000 iterations, 30,000 iterations, 50,000 iterations, 70,000 iterations, and 100,000 iterations, respectively. The indexes of the test results are shown in Table 1. As always, in this article, we show the best results in the table in bold. Finally, we choose to iterate 5000 times, because it achieves relatively better results.

We use Adam optimiser [27] with learning rate of 0.00025 to train the proposed models. After the model is trained, some examples of watermark removal results are provided to illustrate the effectiveness and advantage of the proposed method. In order to better reflect the effect of watermark removal, we use three evaluation indexes: watermark mean square error (wMSE), watermark structural similarity (wSSIM) index, and watermark peak signal to noise ratio (wRNSP). wMSE, wSSIM, and wRNSP only calculate the MSE, SSIM, and RNSP of the watermark area, respectively.

Set loss weight (λ_1 to λ_5): In deep learning, the hyperparameter setting of the model has no mature theoretical basis. We determine the weight of loss based on experience and experiment. First, we set the two ℓ_2 losses and the two adversarial losses to have the same weight (that is, $\lambda_1 = \lambda_2$ and $\lambda_4 = \lambda_5$). Second, we determine the weight ratio of reconstruction loss and adversarial loss through comparative experiments. As shown in Table 2, when the weight ratio of reconstruction loss and adversarial loss is $1:1 \times 10^{-4}$ (that is, $(\lambda_1 + \lambda_2 + \lambda_3):(\lambda_4 + \lambda_5) = 1:1 \times 10^{-4}$), the proposed model works best. Finally, through comparative experiments, as shown in Table 3, all indicators are better when the weight ratio of ℓ_2 loss to perceptual loss is 1:1 (that is, $(\lambda_1 + \lambda_2) = \lambda_3$). Accordingly, in the following experiments, the values of λ_1–λ_5 are $\lambda_1 = 0.25$, $\lambda_2 = 0.25$, $\lambda_3 = 0.5$, $\lambda_4 = 0.5 \times 10^{-4}$, $\lambda_5 = 0.5 \times 10^{-4}$.

Effect of dilated U-Net: Our generator based on the U-Net structure is added four dilated convolution layers in the middle of the U-Net. As shown in Table 4, compare with the network without the dilated convolution layers, the proposed network with the dilated convolution layers performs better. Specifically, wMSE reduces by 0.000029, wSSIM and wPSNR increase by 0.000909 and 1.586726, respectively. It should be noted that the smaller the wMSE is, the larger the wSSIM and wPSRN are, which means the better the quality of the generated image. Fig. 6 shows the watermark removal images obtained by these two methods. It is can be see that without the dilated layers the watermark removal image remains watermark. This is because the dilated convolution can extract higher-level features by expanding the receptive field. Correspondingly, Fig. 7 shows the details of the watermark area of each picture. For example, in the first row of Fig. 7, we can clearly

Table 1 Index of experimental results of different training iterations

Iterations	wMSE	wSSIM	wPSNR
10,000	0.000258	0.991453	38.271956
30,000	0.000132	0.995064	40.972816
50,000	**0.000074**	**0.997065**	**43.999067**
70,000	0.000071	0.997142	43.999539
100,000	0.000099	0.996484	43.038464

Table 2 Indexes of experimental results with different ratios of reconstruction loss and adversarial loss

$(\lambda_1+\lambda_2+\lambda_3):(\lambda_4+\lambda_5)$	wMSE	wSSIM	wPSNR
$1:1\times10^{-1}$	0.000077	0.996901	43.633028
$1:1\times10^{-2}$	0.000118	0.995361	41.513909
$1:1\times10^{-3}$	0.000107	0.995808	42.021092
$1:1\times10^{-4}$	**0.000074**	**0.997049**	**43.844612**
$1:1\times10^{-5}$	0.000088	0.996645	43.060845

Table 3 Indexes of experimental results with different ratios of ℓ_2 loss and perceptual loss.

$(\lambda_1+\lambda_2):\lambda_3$	wMSE	wSSIM	wPSNR
1:3	0.000097	0.995865	41.924797
2:2	**0.000074**	**0.997049**	**43.844612**
3:1	0.000102	0.995973	42.188555

Table 4 Comparison of watermark removal results with dilated convolution layers and without dilated convolution layers

Structure	CW-data sets		
	wMSE	wSSIM	wPSNR
without dilated convolution	0.000103	0.996140	42.257886
with dilated convolution	**0.000074**	**0.997049**	**43.844612**

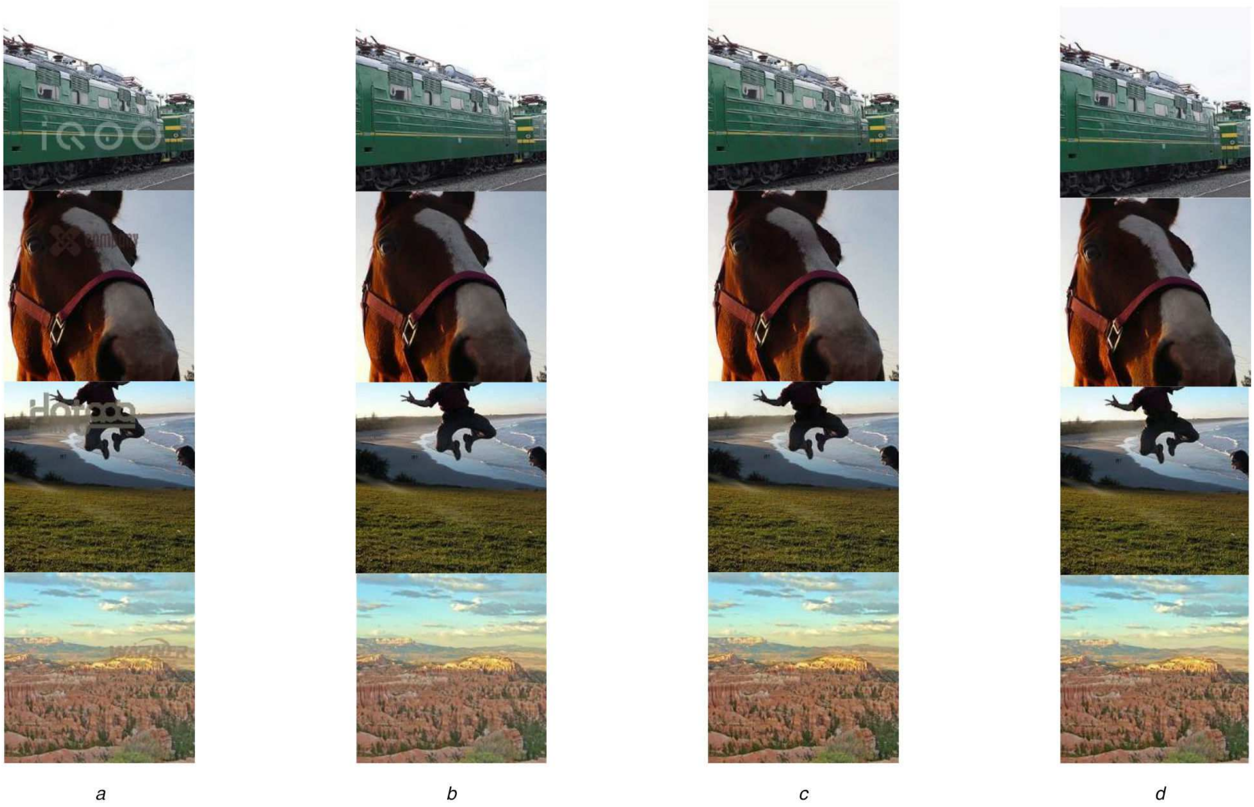

Fig. 6 *Effect of dilated convolution layers*
(a) Colour watermarked images, ***(b)*** Original watermarkless images, ***(c)*** Proposed network architecture without dilated convolution layers, ***(d)*** Proposed network architecture

see that the residual watermark in the image is obvious when there no dilated layers.

Effect of extraction network: The generator of the proposed network architecture is composed of two parts, extraction network and inpainting network. We designed control experiments to compare the proposed model with a model that only contains inpainting network to verify the importance of extraction network. The experiment results are shown in Fig. 8, where the red frame area in the first row of images is enlarged and displayed in the second row. It can be seen clearly that the model with the extraction network has a better watermark removal effect. As the extraction network only focuses on the watermark area and can extract watermarks very well, which is a key step for the proposed network architecture to remove watermarks.

Effect of taking I_0 and I_{pre} as the input of inpainting network: We find that the extraction network also affects the non-watermarked area. As shown in Fig. 9, the non-watermarked area of the preliminary watermark removal image I_{pre} is destroyed. In order to inpaint it, we use watermarked image I_0 and the preliminary watermark removal image I_{pre} as the input of the inpainting network. As can be seen from Fig. 10, compared to inputting a single image I_{pre} to the inpainting network, we use I_0 as a condition and input it to the inpainting network with I_{pre}, and achieved perfect watermark removal image.

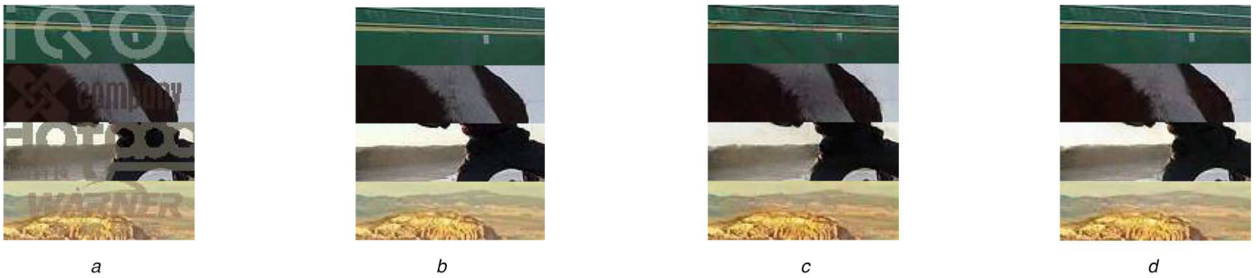

Fig. 7 *Details of the watermark area of all images in Fig. 6*
(a) The enlarged detail of the watermark area in Fig. 6(a), ***(b)*** The enlarged detail of the watermark area in Fig. 6(b), ***(c)*** The enlarged detail of the watermark area in Fig. 6(c), ***(d)*** The enlarged detail of the watermark area in Fig. 6(d)

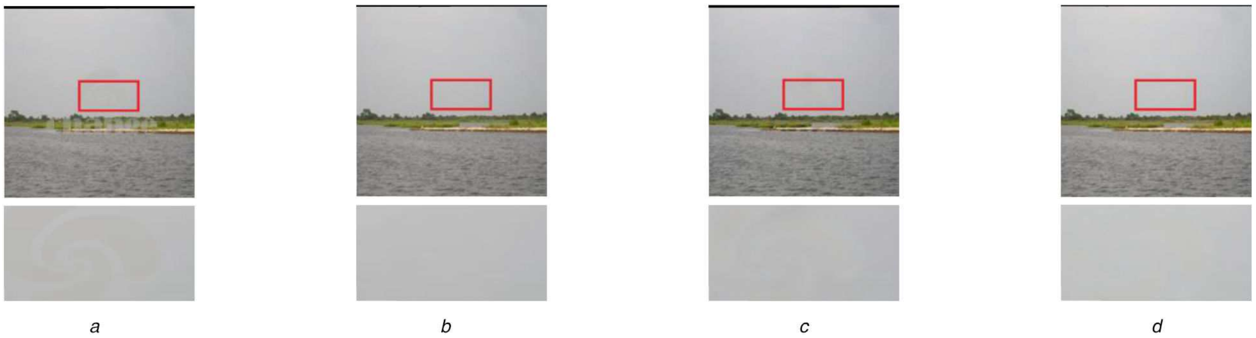

Fig. 8 *Effect of extraction network*
(a) Input image with colour watermark, ***(b)*** Original watermarkless image, ***(c)*** Proposed network architecture without extraction network, ***(d)*** Proposed network architecture

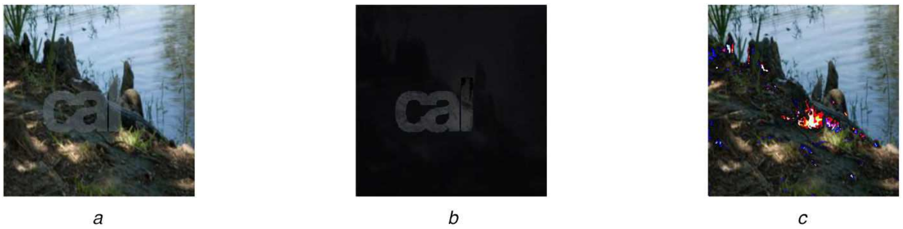

Fig. 9 *inpainting network destroy the non-watermarked area of the watermarked image*
(a) Colour watermarked image, ***(b)*** Output of inpainting network, ***(c)*** I_{pre} obtained by subtracting the extracted watermark from the watermarked image

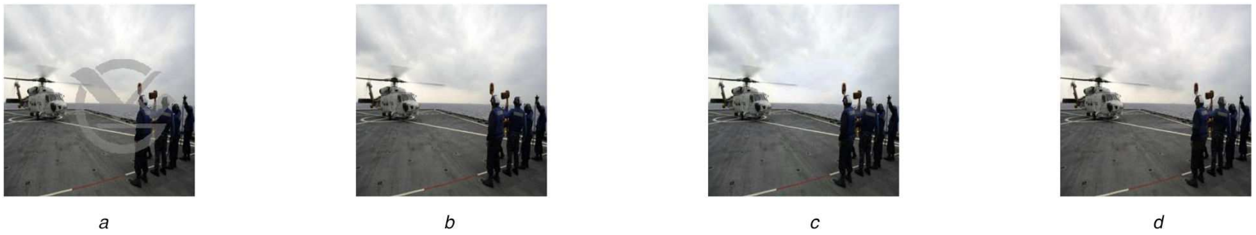

Fig. 10 *Effect of taking I_0 and I_{pre} as input of Inpainting Network*
(a) Colour watermarked image, ***(b)*** Original watermarkless image, ***(c)*** Proposed network architecture Input a single image I_{pre} to inpainting network, ***(d)*** Proposed network architecture

Table 5 Comparison with Li and colleagues on WW-data set and CW-data set

Structure	WW-data sets			CW-data sets		
	wMSE	wSSIM	wPSNR	wMSE	wSSIM	wPSNR
input	0.003739	0.962240	26.512192	0.000698	0.980169	33.689828
Pix2pix [28]	0.000318	0.988719	37.253956	0.000618	0.982728	34.488141
attention-GAN [29]	0.000099	0.996249	43.416783	0.000331	0.988769	37.041123
PRN [30]	0.000365	0.992222	38.733818	0.000344	0.990152	37.302824
Li *et al.* [5]	0.000471	0.985684	35.950049	0.000892	0.977649	32.708849
ours	**0.000049**	**0.997786**	**46.007274**	**0.000074**	**0.997049**	**43.844612**

Comparison with state-of-the-act: In order to prove the validity of our method, We compare with other methods on the WW-data set and CW-data set, respectively, in terms of wMSE, wSSIM, and wPSNR, as shown in Table 5. Obviously, the proposed work has obtained exceptional results on each indicator.

Figs. 11 and 12 show the comparison of watermark removal results of other methods on the WW-data set and CW-data set,

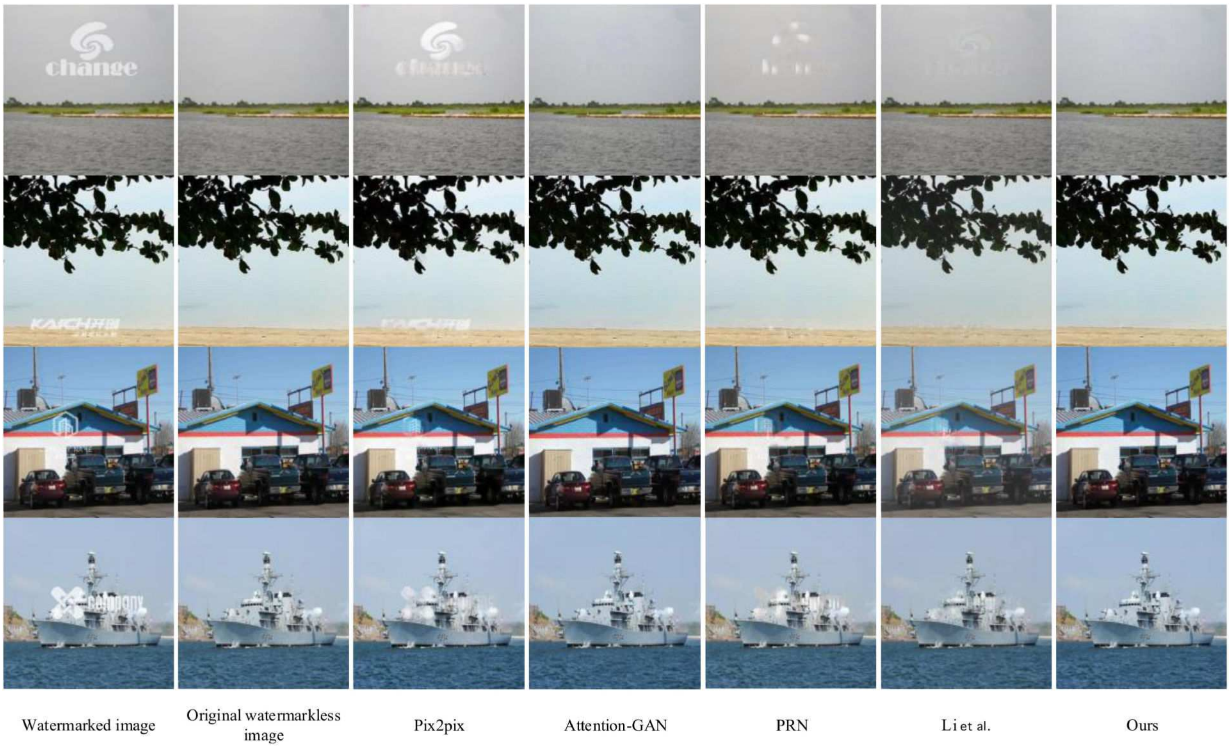

Fig. 11 *Comparison of watermark removal results on WW-data sets*

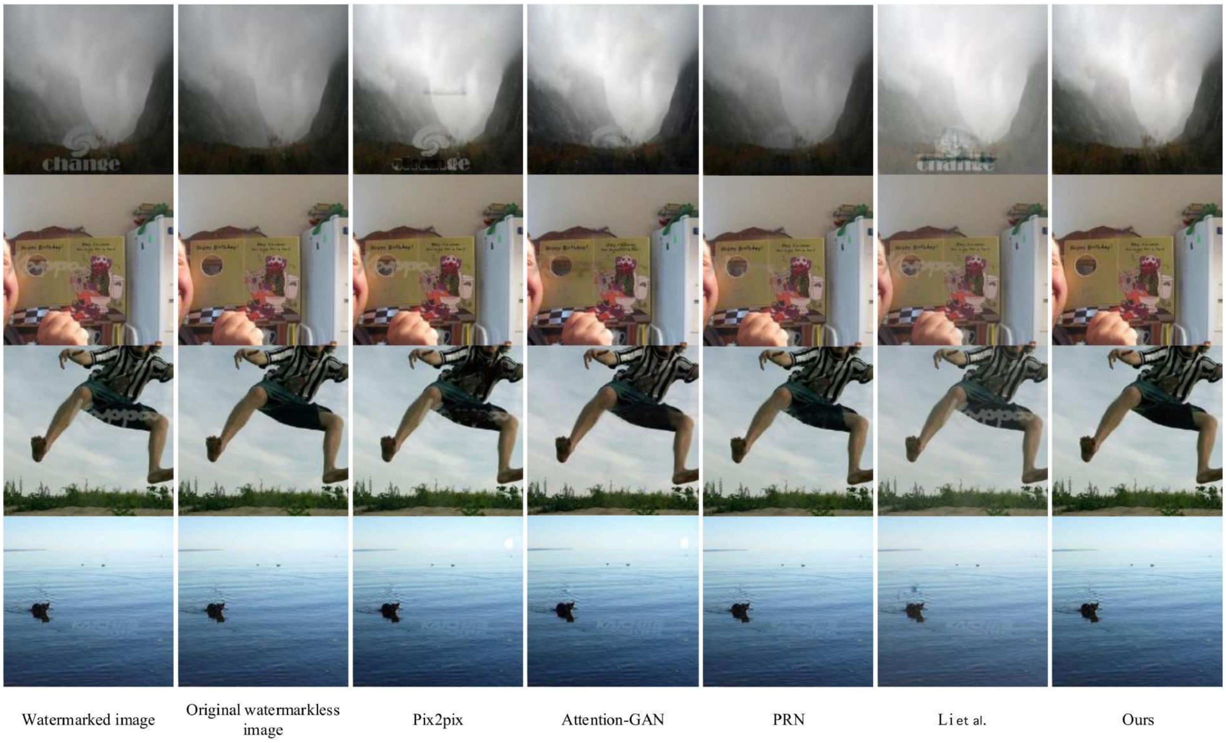

Fig. 12 *Comparison of watermark removal results on CW-data set*

respectively. On the WW-data set, as shown in Fig. 11, the watermark removal results of Pix2pix [28] and PRN [30] have obvious residual watermarks. The results of Li and colleagues are slightly better, but the shape of the watermark can still be seen. Attention-GAN's [29] watermark removal effect is the only one comparable to the proposed model, but when the watermark colour and background are very similar, as shown in Fig. 12, attention-GAN cannot successfully remove the watermark. It can also be seen from Fig. 12 that other methods that do not work well on WW-data set also cannot remove the watermarks on CW-data set. In conclusion, the experiment shows that the proposed model performs better on both WW-data sets and CW-data sets.

5 Conclusion

In this paper, we proposed a visible watermark removal network architecture including watermark extraction and image inpainting. The structure of these two parts is based on the U-Net, and is trained through ℓ_2 loss, perceptual loss, and adversarial loss. In addition, we created two watermark image data sets, i.e. WW-data set and CW-data set for training and testing the network architecture. Experiments have shown that this method can effectively remove colour watermarks with similar colours to the background, and the effect is better than the existing methods. This also means that the colour watermarks similar to the background cannot protect image copyrights well, and developing a robust watermarking technology to protect image copyrights remains a challenge. Recently, research on the harmonisation of composite images is increasing. It can be considered from this perspective to make the watermark compatible with the image to counter the watermark removal method.

6 Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61471120 and the Natural Science Foundation of Hunan Province under Grant 2020JJ4745.

7 References

[1] Liu, S., Pan, Z., Song, H.: 'Digital image watermarking method based on DCT and fractal encoding', *IET Image Process.*, 2017, **11**, (10), pp. 815–821

[2] Huang, C.H., Wu, J.L.: 'Attacking visible watermarking schemes', *IEEE Trans. Multimedia*, 2004, **6**, (1), pp. 16–30

[3] Xu, C., Lu, Y., Zhou, Y.: 'An automatic visible watermark removal technique using image inpainting algorithms'. Int. Conf. on Systems and Informatics (ICSAI), Hangzhou, China, November 2017, pp. 1152–1157

[4] LeCun, Y., Bengio, Y., Hinton, G.: 'Deep learning', *Nature*, 2015, **521**, (7553), pp. 436–444

[5] Li, X., Lu, C., Cheng, D.*, et al.*: 'Towards photo-realistic visible watermark removal with conditional generative adversarial networks'. Proc. IEEE Conf. Computer Vision Pattern Recognition (CVPR), Long Beach, CA, USA, 2019

[6] Cheng, D., Li, X., Li, W.H.*, et al.*: 'Large-scale visible watermark detection and removal with deep convolutional networks'. Conf. Pattern Recognition Computer Vision (PRCV), Guangzhou, China, November 2018, pp. 27–40

[7] Mirza, M., Osindero, S.: 'Conditional generative adversarial nets', arXiv:14111784, 2014

[8] Mao, X., Li, Q., Xie, H.*, et al.*: 'Multi-class generative adversarial networks with the L2 loss function', CoRR, abs/1611.04076, 2016

[9] Pei, S.C., Zeng, Y.C.: 'A novel image recovery algorithm for visible watermarked images', *IEEE Trans. Inf. Forensic Secur.*, 2006, **1**, (4), pp. 543–550

[10] Santoyo-Garcia, H., Fragoso-Navarro, E., Reyes-Reyes, R.*, et al.*: 'An automatic visible watermark detection method using total variation'. Proc. Int. Workshop Workshop on Biometrics and Forensics (IWBF), Coventry, United kingdom, April 2017

[11] Dekel, T., Rubinstein, M., Liu, C.*, et al.*: 'On the effectiveness of visible watermarks'. Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR), Honolulu, United states, November 2017, pp. 6864–6872

[12] Ballester, C., Bertalmio, M., Caselles, V.*, et al.*: 'Filling-in by joint interpolation of vector fields and gray levels', *IEEE Trans. Image Process.*, 2001, **10**, (8), pp. 1200–1211

[13] Liu, G., Reda, F.A., Shih, K.J.*, et al.*: 'Image inpainting for irregular holes using partial convolutions'. European Conf. on Computer Vision, Munich, Germany, September 2018, pp. 89–105

[14] Gu, J., Wang, Z., Kuen, J.*, et al.*: 'Recent advances in convolutional neural networks', *Pattern Recognit.*, 2018, **77**, pp. 354–377

[15] Fukushima, K.: 'Neocognitron: a hierarchical neural network capable of visual pattern recognition', *Neural Netw.*, 1988, **1**, (2), pp. 119–130

[16] Goodfellow, I., Pouget-Abadie, J., Mirza, M.*, et al.*: 'Generative adversarial nets'. Advances in Neural Information Processing Systems, Montreal, Canada, 2014, pp. 2672–2680

[17] Pathak, D., Krahenbuhl, P., Donahue, J.*, et al.*: 'Context encoders: feature learning by inpainting'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, USA, June 2016, pp. 2536–2544

[18] Yang, C., Lu, X., Lin, Z.*, et al.*: 'High-resolution image inpainting using multi-scale neural patch synthesis'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, USA, July 2017, pp. 4076–4084

[19] Yu, F., Koltun, V.: 'Multi-scale context aggregation by dilated convolutions', arXiv:1511.07122, 2015

[20] Iizuka, S., Simo-Serra, E., Ishikawa, H.: 'Globally and locally consistent image completion', *ACM Trans. Graph.*, 2017, **36**, (4), pp. 1–14

[21] Yu, J., Lin, Z., Yang, J.*, et al.*: 'Generative image inpainting with contextual attention'. Proc. IEEE Computer Society on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, USA, June 2018, pp. 5505–5514

[22] Liu, H., Jiang, B., Xiao, Y.*, et al.*: 'Coherent semantic attention for image inpainting'. Proc. IEEE Int. Conf. Computer Vision, Seoul, Korea, October 2019, pp. 4169–4178

[23] Zhou, B., Lapedriza, A., Khosla, A.*, et al.*: 'Places: a 10 million image database for scene recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, **40**, (6), pp. 1452–1464

[24] Ronneberger, O., Fischer, P., Brox, T.: 'U-net: convolutional networks for biomedical image segmentation'. 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 2015, (Lecture Notes Computer Science, vol. 9351), pp. 234–241

[25] Ali, S., Abbas, W., Hassan, N.U.*, et al.*: 'Cpgan: conditional patchbased generative adversarial network for retinal vessel segmentation', *IET Image Process.*, 2020, **14**, pp. 1081–1090

[26] Russakovsky, O., Deng, J., Su, H.*, et al.*: 'Imagenet large scale visual recognition challenge', *Int. J. Comput. Vis.*, 2015, **115**, (3), pp. 211–252

[27] Kingma, D.P., Ba, J.: 'Adam: a method for stochastic optimization', arXiv:14126980, 2014

[28] Isola, P., Zhu, J.Y., Zhou, T.*, et al.*: 'Image-to-image translation with conditional adversarial networks'. Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR), Honolulu, United states, July 2017, pp. 5967–5976

[29] Qian, R., Tan, R.T., Yang, W.*, et al.*: 'Attentive generative adversarial network for raindrop removal from a single image'. Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR), Salt Lake City, United states, June 2018, pp. 2482–2491

[30] Ren, D., Zuo, W., Hu, Q.*, et al.*: 'Progressive image deraining networks: a better and simpler baseline'. Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR), Long Beach, United states, June 2019, pp. 3932–3941