IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 6, 2019, accepted October 9, 2019, date of publication October 14, 2019, date of current version October 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947266

Single Image Reflection Removal Based on GAN With Gradient Constraint

RYO ABIKO, (Student Member, IEEE), AND MASAAKI IKEHARA, (Member, IEEE)
Department of Electronics and Electrical Engineering, Keio University, Yokohama 223-8522, Japan
Corresponding author: Ryo Abiko (abiko@tkhm.elec.keio.ac.jp)

ABSTRACT When we take a picture through glass windows, the photographs are often degraded by undesired reflections. To separate reflection layer and background layer is an important problem for enhancing image quality. However, single-image reflection removal is a challenging process because of the ill-posed nature of the problem. In this paper, we propose a single-image reflection removal method based on generative adversarial networks. Our network is an end-to-end trained network with four types of losses. It includes pixel loss, feature loss, adversarial loss, and gradient constraint loss. We propose a novel gradient constraint loss in order to separate the background layer and the reflection layer clearly. Gradient constraint loss is applied in a gradient domain and it minimizes the correlation between the background and reflection layer. Owing to the novel loss and our new synthetic dataset, our reflection removal method outperforms state-of-the-art methods in PSNR and SSIM, especially in real world images.

INDEX TERMS Image restoration, deep learning, reflection removal, image separation, generative adversarial network.

I. INTRODUCTION

When taking photographs through transparent material such as glass or windows, undesired reflections often ruin the images. To obtain clear images, users may make dark situation or change the camera position but it is not effective for removing reflections because of the limitation on space. The reflection does not only degrade the image quality but also affects the results of applications such as segmentation or classification. Thus, removing reflections from an image is an important task in computer vision. The example of single-image reflection removal task is shown in Fig. 1.

Separating background layer and reflection layer is an ill-posed problem because the photographing situation is not fixed. The thickness of the glass, the number of the glass, the transparent rate, and the reflection rate could change and we cannot model them in an appropriate manner. To solve this ill-posed problem, multiple input images or a video is used in many methods [1]–[8]. Inputting multiple images makes the ill-posed problem easier to solve but in actual cases, it is difficult to prepare adequate multiple images. It is because additional devices such as polarizing filter [2] are sometimes required but they cannot be obtained readily.

The associate editor coordinating the review of this manuscript and approving it for publication was You Yang.

(a) Input image (b) Generated image (c) Reflection layer

FIGURE 1. A visualization of single-image reflection removal. Fig. 1a is a synthetic input image which includes reflections. Fig. 1b is a generated background image of our method and Fig. 1c is a reflection layer.

Even if additional devices are not required, users have to take several photographs from different view and this is difficult when the space is limited or the object is not stationary. In addition, most photographs which are already taken do not contain multiple view images so the study of single-image reflection removal is important.

Recently, some methods were proposed to remove reflection without using multiple images [9]–[17]. Methods in [9]–[11] are based on solving optimization problem. Since understanding the image structure is important in removing reflections, Convolutional Neural Network (CNN) was applied in some methods [12]–[14]. In particular, Generative Adversarial Networks (GAN)-based methods [15]–[17] have produced good results. Generally, It is said that GAN-based

methods can generate more realistic images than other methods since adversarial loss encourages networks to generate images which follow natural image distribution. Though in the case of reflection removal, GAN-based methods still have problem with achieving both reflection-free and natural image. BDN [15] and ERR Net [17] sometimes generate an image with unnatural color tone. This is occured because they do not consider the correlation between background and reflection layers. Since the color tone is degraded when the structure of background layer appears in the reflection layer, it is important to keep the structure of background layer and reflection layer independent. PL Net [16] considers the correlation between two layers but this approach does not deal with the case when overexposure is occurred.

In this paper, we propose a novel single-image reflection removal method based on generative adversarial networks. We propose a new training loss called ''gradient constraint loss'' in order to preserve texture and structure information effectively while separating background and reflection layer. Gradient constraint loss keeps the correlation between background and reflection layer low. Since the background layer and reflection layer have no relevance, it is important not to share the information in these two layers. In addition, we use Tanhshrink function when calculating gradient constraint loss in order to compress small gradients. Owing to this function, our network can keep training effectively even if overexposure is occurred in training images. The detail is described in Sec. IV-B.4. To train our network, we use combination of four kinds of losses including pixel loss, feature loss, adversarial loss, and gradient constraint loss. Feature loss is applied to both background layer and reflection layer so it is possible to separate the image into two layers while retaining the image features. When training the networks, we used several reflection models and applied many conditions to the synthetic reflection images. It leads our network to remove a real world reflection which has perplexing conditions. The contributions of our paper are summarized below:

- We propose a new Gradient Constrained Network (GCNet) for single-image reflection removal. When training our network, we use four types of losses including pixel loss, feature loss, adversarial loss, and gradient constraint loss. Since the gradient constraint keeps the correlation between background and reflection layer low, the output background layer preserves texture information well and the visual quality is high.
- We applied many kinds of terms to the training dataset, which enable our trained network to remove reflections in many challenging real conditions.

The overview of our method is shown in Fig. 2. Our code is available at https://github.com/ryo-abiko/GCNet.

II. RELATED WORK

Since reflection separation is an ill-posed problem, many methods use multiple images [1]–[5], [7], [8] or video [6] as an input. Multiple images make the ill-posed problem easier to solve but they are difficult to obtain and additional operation will be required. Thus, single-image reflection removal methods are mainly considered in this paper.

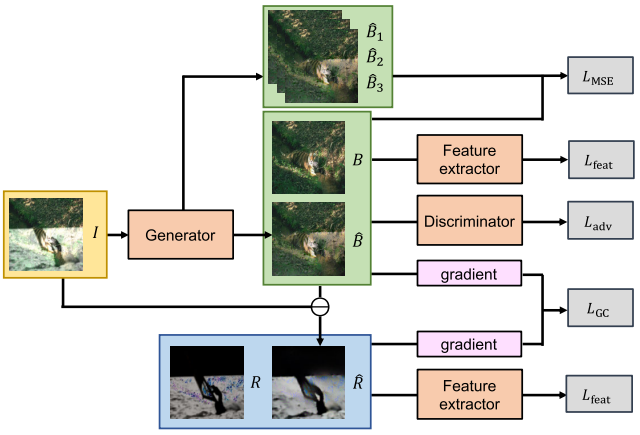


FIGURE 2. The overview of our method. *B* is ground truth background layer and *R* is ground truth reflection layer. Generator, discriminator, and feature extractor are based on CNN. The architecture of the generator is shown in Fig. 3.

A. OPTIMIZATION BASED METHODS

Several methods use optimization to suppress the reflection in a single image. To solve an optimization problem, additional prior such as gradient sparsity [10], [11] or gaussian mixture models [9] is needed. These methods can suppress reflections effectively when the input image follow the assumption but when the assumption cannot be applied, the result will be catastrophic.

B. DEEP LEARNING BASED METHODS

The first method which uses deep convolutional neural networks for reflection removal was proposed by Fan et al. in [12]. Two networks are cascaded and they first predict edges of background layer by using the first CNN. The predicted edge is used for the guide when reconstructing background layer by using the second CNN. Since they only use pixel-wise loss function in the training process, the semantic structure is not considered. In particular, Generative Adversarial Networks (GAN)-based methods have produced outstanding results in reflection removing task, likewise in other computer vision tasks such as inpainting [18] and super-resolution [19]. Zhang et al. proposed PL Net [16] which is trained by loss function composed of feature loss, adversarial loss, and exclusion loss. The network architecture and loss function are tuned to focus on both low-level and high-level image information. The network is weak in processing overexposed images because their training method does not cope with those problem. Yang *et al.* [15] proposed a network which predicts the background layer and reflection layer alternately. They use L_2 loss and adversarial loss to train the networks. Wei et al. proposed ERR Net [17] which can be trained by misaligned data. The image features which are obtained by pre-trained VGG19 network [20] are used as input data and they are also used in calculating feature loss. Since these two methods do not focus on the correlation

between background and reflection layer, they sometimes generate an image with unnatural color tone.

III. SUPPORTING METHODS

A. SYNTHETIC REFLECTION IMAGE

In this paper, we denote I as an image with reflections. The background layer is denoted as B and the reflection layer is denoted as R. In this case, I can be modeled as a linear combination of B and R as below:

$$I = B + R. \tag{1}$$

In previous work, R is generated by using several reflection models. In our training method, we use three of them. Since people usually focus on background layer when taking a photograph, reflection layer tends to be less focused and blurred [10]. This can be synthesized by applying Gaussian filter to the reflection layer. When we denote R_o as an original reflection image, first reflection model can be expressed as:

$$R_1 = \alpha K * R_o, \tag{2}$$

where R_1 is a synthetic reflection layer, K is a Gaussian kernel, and α is a reflection rate. When the glass is thick or the window is double-paned, ghosting effects will appear in the captured image [9], [13]. Reflection layer with ghosting effect R_2 can be expressed as:

$$R_2 = \beta K * H * R_o, \tag{3}$$

where H is a random kernel with two pulses and β is a reflection rate. Third model is proposed by [12], which can be computed by subtracting a value from reflection layer. This model can be expressed as:

$$R_3 = K * R_o - \gamma, \tag{4}$$

where R_3 is a synthetic reflection layer and γ is an amount of shift. In our method, γ is computed in the same way which is described in [12]. Restoring B from I is the final goal of single-image reflection removal methods but it is difficult because solving B from I is an ill-posed problem.

B. GENERATIVE ADVERSARIAL NETWORKS

Generative adversarial networks (GAN) [21] is a learning method which maps noise to an image. Generator G is trained to create a real-like image and discriminator D is trained to judge whether the discriminator input is real or not. When training GAN, min-maximizing process between generator and discriminator is applied and it can be expressed as:

$$\min_G \max_D V(G, D) = \mathbb{E}_x[logD(x)] + \mathbb{E}_z[log(1 - D(G(z)))], \tag{5}$$

where x is an image and z is a noise variable. When GAN is applied to image restoration tasks, z should be a deteriorated image. Since GAN is good at solving inverse problems, it has shown remarkable results in image processing such as inpainting [18], colorization [22], denoising [23], and super-resolution [19].

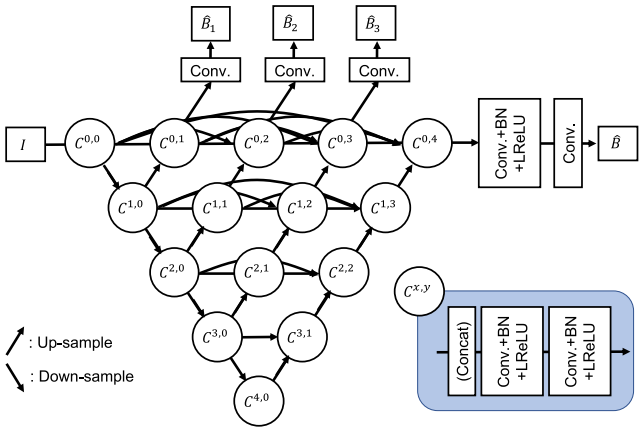


FIGURE 3. The architecture of our proposed generator. It is based on UNet++ L^4 [24].

IV. PROPOSED METHOD

Our method removes reflections from a single image with a training-based algorithm using GAN. We represent that our GAN-based method with gradient constraint can remove reflections effectively. The overview of our method is shown in Fig. 2.

A. NETWORK MODEL

We illustrate the proposed generator architecture in Fig. 3. The network structure of our proposed method is based on UNet++ L^4 [24]. It is a combination of convolutional layer, batch normalization layer [25], leaky ReLU layer [26], max pooling layer, and bilinear interpolation layer. Since we adapt deep supervision structure [24], there are four outputs. We use $\hat{B}$ as a main output and the other outputs are used for computing pixel loss. The filter size of the convolutional layers are set to 3×3. The number of the channels in the convolutional layers in $C^{x,y}$ are set to 2^{x+5}.

Our discriminator is composed of the enumeration of convolutional layer, batch normalization layer, and leaky ReLU layer. The stride of convolutional layers is set to 2 in every two convolutional layers. Since the final output size of our discriminator is 16×16, L_2 difference is applied to compute the adversarial loss.

B. LOSS FUNCTIONS FOR GENERATOR

In our method, we applied four kinds of losses to separate background and reflection layer effectively. Let G, D, F be generator, discriminator, and feature extractor, respectively. Generated background image $\hat{B}_i$ can be obtained by inputting image I_i into generator G. In our method, we do not estimate reflection layer directly so the reflection layer $\hat{R}_i$ is estimated by subtracting generated background image from the input image. Thus, the estimation of $\hat{B}_i$ and $\hat{R}_i$ can be expressed as:

$$\begin{aligned} \hat{B}_i &= G(I_i; \theta_G) \\ \hat{R}_i &= I_i - \hat{B}_i \end{aligned} \tag{6}$$

where θ_G is the set of weights of Generator G. The main purpose in the training process is to minimize the loss $\mathcal{L}_G(\theta_G)$.

(a) Background layer (b) Synthetic reflection layer (c) Synthetic reflected image (d) Ground truth reflection layer

FIGURE 4. The example of overexposure. The ground truth reflection layer contains a structure of the background layer since clipping is caused.

Our loss $\mathcal{L}_G(\theta_G)$ is a combination of four kinds of losses and can be defined as:

$$\mathcal{L}_G(\theta_G) = \mu_1\mathcal{L}_{\text{MSE}} + \mu_2\mathcal{L}_{\text{feat}} + \mu_3\mathcal{L}_{\text{adv}} + \mu_4\mathcal{L}_{\text{GC}}. \quad (7)$$

$\mathcal{L}_{\text{MSE}}$ is a pixel loss which computes the L_2 difference and $\mathcal{L}_{\text{feat}}$ is a feature loss which is applied in feature domain. $\mathcal{L}_{\text{adv}}$ is an adversarial loss and $\mathcal{L}_{\text{GC}}$ is a novel loss which is effective for separating background and reflection layers.

1) PIXEL LOSS

Pixel loss is applied to compare the pixel-wise difference between generated image and ground truth image. Since minimizing the mean squared error (MSE) is effective for avoiding vanishing gradient problem in training GAN [27], we use MSE loss function to calculate the pixel loss. Our generator generates four images including one main generated image $\hat{B}$ and three supporting images $\hat{B}_1$, $\hat{B}_2$, and $\hat{B}_3$. $\hat{B}_1$, $\hat{B}_2$, and $\hat{B}_3$ are used only for calculating the pixel loss and they have a good influence in training process [24]. To emphasize the optimization of main generated image, the four output images are weight-averaged when the pixel loss is computed. The additional information is shown in Fig. 3. From the above, our pixel loss is computed by calculating L_2 difference and it is expressed as:

$$\hat{B}_{1i} = G_1(I_i;\theta_G),\ \hat{B}_{2i} = G_2(I_i;\theta_G),\ \hat{B}_{3i} = G_3(I_i;\theta_G)$$
$$\mathcal{L}_{\text{MSE}} = \sum_i^N ||\frac{1}{8}(5*\hat{B}_i + \hat{B}_{1i} + \hat{B}_{2i} + \hat{B}_{3i}) - B_i||_2 \quad (8)$$

where G_1, G_2, G_3 are the part of generator G and B_i is a ground truth background image.

2) FEATURE LOSS

In the reflection removing task, it is important to preserve the structure of the image. Since the pixel loss cannot optimize the semantic feature of the image, we adopted feature loss in our method. Pretrained VGG-19 network [20] is applied for the feature extracting network and the outputs from the layer 'conv5_2' are used for the computation. We calculate the L_1 difference between the feature vector of generated image and ground truth image. Since background layer and reflection layer have different image structure, the feature loss is applied to both background and reflection layer. Our feature loss $\mathcal{L}_{\text{feat}}$ is expressed as:

$$\mathcal{L}_{\text{feat}} = \sum_i^N(||F(\hat{B}_i) - F(B_i)||_1 + ||F(\hat{R}_i) - F(R_i)||_1). \quad (9)$$

3) ADVERSARIAL LOSS

It is known that simple CNN-based networks with MSE loss tend to generate blurry and unnatural images. It is because the images generated by those methods are the average of several natural solutions [19]. To avoid this problem, adversarial loss was proposed in [21]. The adversarial loss is applied to encourage generator to generate images which follows natural image distribution. In the reflection removing task, the deterioration of color tone is a common problem but in our method, applying the adversarial loss restrained this problem. The adversarial loss in our method is expressed as:

$$\mathcal{L}_{\text{adv}} = \sum_i^N ||1 - D(\hat{B}_i;\theta_D)||_2. \quad (10)$$

4) GRADIENT CONSTRAINT LOSS

The main task in a single-image reflection removal is to separate a single image into two layers including background layer and reflection layer. In most cases, background layer and reflection layer have no correlation so minimizing the correlation between two layers is effective in this task. To minimize the correlation, we applied a novel loss function called gradient constraint loss. It is applied in a gradient domain in order to make the task easier. Our gradient constraint loss is composed of two terms: $\mathcal{L}_{\text{GCM}}$ and $\mathcal{L}_{\text{GCS}}$. $\mathcal{L}_{\text{GCM}}$ is a term to keep the correlation between two layers low and $\mathcal{L}_{\text{GCS}}$ works as a constraint of $\mathcal{L}_{\text{GCM}}$. However, in the early stage of training, we find that the effect of gradient constraint loss is too strong and the network cannot be trained effectively. Thus, the gradient constraint loss is multiplied by the number of epochs in order to keep the effect of the loss low in the early stage of training. Finally, the gradient loss can be described as:

$$\mathcal{L}_{\text{GC}} = (\text{epoch} - 1) * (\mathcal{L}_{\text{GCM}} + \mathcal{L}_{\text{GCS}}). \quad (11)$$

Since the edge information of background layer and reflection layer should be independent, $\mathcal{L}_{\text{GCM}}$ calculates the element-wise product of these two edge layers. $\mathcal{L}_{\text{GCS}}$ is

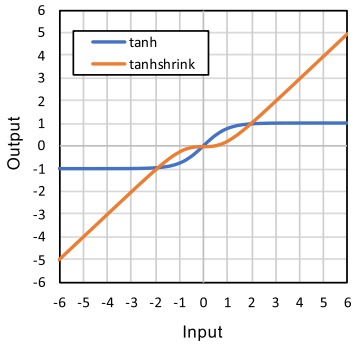


FIGURE 5. The comparison of Tanh and Tanhshrink function.

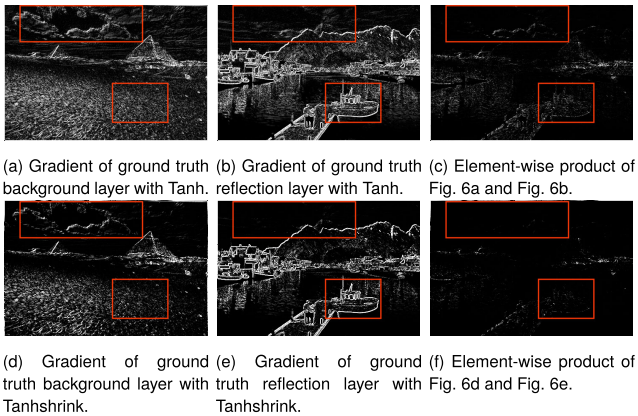
applied for giving a constraint to $\mathcal{L}_{\mathrm{GCM}}$ and it helps network to separate layers effectively. $\mathcal{L}_{\mathrm{GCM}}$ and $\mathcal{L}_{\mathrm{GCS}}$ can be expressed as:

$$\hat{B}_{gi} = \mathrm{Tanhshrink}(\nabla_x \hat{B}_i + \nabla_y \hat{B}_i)$$
$$\hat{R}_{gi} = \mathrm{Tanhshrink}(\nabla_x \hat{R}_i + \nabla_y \hat{R}_i)$$
$$\mathcal{L}_{\mathrm{GCM}} = \sum_{i}^{N} ||\hat{B}_{gi} \odot \hat{R}_{gi}||_1 \tag{12}$$
$$\mathcal{L}_{\mathrm{GCS}} = \sum_{i}^{N} ||(\hat{B}_{gi} + \hat{R}_{gi}) - (\nabla_x I_i + \nabla_y I_i)||_1. \tag{13}$$

The basic idea of minimizing correlation between two layers is proposed in [16] but in our method, we applied a new active function and added a constraint. The main purpose of our gradient constraint loss is to focus on large edges and separate layers effectively. Since the input and ground truth images are normalized into the range $[-2.5, 2.5]$, which is also used in VGG19 network [20], the conventional Tanh function is not suitable for our network. In addition, to separate layers by mainly using large edges, we want to reduce the impact of small edge regions. Thus, we applied Tanhshrink function as an activation function. The comparison of Tanh and Tanhshrink function is shown in Fig. 5 and the formula can be described as:

$$\mathrm{Tanh}(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$
$$\mathrm{Tanhshrink}(x) = x - \mathrm{Tanh}(x). \tag{14}$$

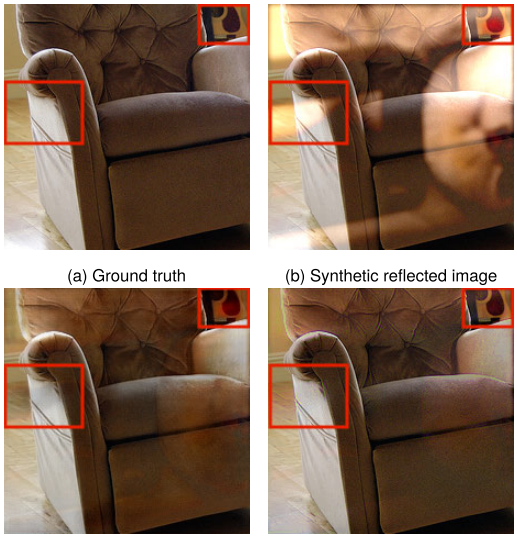
By using Tanhshrink, the robustness against blown out highlights is also obtained. When overexposure is occurred, the structure of the reflection layer will be corrupted with the background layer as shown in Fig. 4. In this case, the correlation between background and reflection layer does not become zero. Since Eq. (12) encourage the element-wise product of the gradient layers to be zero, the training will not perform well in this situation. To overcome this problem, Tanhshrink function is effective. It is because the gradients of the artifacts generated by the overexposure tend to be smaller than the desired gradients. As shown in Fig. 6, we can see that Fig. 6f contains fewer structures from background layer than Fig. 6d. Owing to this effect, when Tanhshrink is applied as an



(a) Gradient of ground truth background layer with Tanh. (b) Gradient of ground truth reflection layer with Tanh. (c) Element-wise product of Fig. 6a and Fig. 6b.

(d) Gradient of ground truth background layer with Tanhshrink. (e) Gradient of ground truth reflection layer with Tanhshrink. (f) Element-wise product of Fig. 6d and Fig. 6e.

FIGURE 6. The comparison of Tanh and Tanhshrink in gradient domain.



(a) Ground truth (b) Synthetic reflected image

(c) PL [16] (d) GCNet (ours)

FIGURE 7. The recovering example when overexposure is caused. Our method can remove blown out highlights effectively.

activation function, the ground truth of element-wise product layer gets closer to zero even if overexposure is occurred. This is important when applying Eq. (12) during the training process.

We also apply $\mathcal{L}_{\mathrm{GCS}}$ as a constraint of $\mathcal{L}_{\mathrm{GCM}}$. Since we use Tanhshrink for the activation function, small gradients are compressed into even smaller values. The training process may be affected by this feature when large gradients are wrongly divided into small gradients. This problem often occurs when the global tone of the generated image is changed. Thus, we apply $\mathcal{L}_{\mathrm{GCS}}$ in order to help generator to separate image not by deteriorating the color tone but by focusing on the structure of the image. We experimentally apply Tanhshrink function only to the generated gradient values since the training did not perform well when the Tanhshrink function is applied to the input gradient values. The effectiveness of the gradient constraint loss in processing real image is shown in Sec. V-B and in Fig. 11.

(a) Ground truth image (b) Gray-scaled reflection (c) Color reflection

FIGURE 8. A visualization of the types of synthetic reflection.

C. LOSS FUNCTION FOR DISCRIMINATOR

Since our method is based on GAN, discriminator has to be trained while generator is trained. The discriminator is trained by minimizing the loss $\mathcal{L}_D(\theta_D)$ which is based on LSGAN [27] and it is described as below:

$$\mathcal{L}_D(\theta_D) = \sum_i^N (||D(\hat{B}_i; \theta_D)||_2 + ||V - D(B_i; \theta_D)||_2), \quad (15)$$

where V is a random valued matrix which follows Gaussian distribution with an average of 1. We did not set V as a constant value in order to train discriminator effectively.

D. TRAINING DATASET

To create the training dataset, we use PASCAL VOC 2012 dataset [28] which includes 17K images. We exclude grayscale and pale colored images since they affect the training. The images are first resized into 256×256 by using bicubic interpolation. After that, the images are randomly flipped and one image is used for the background layer B and another image is used for the reflection layer R. The background layer image is randomly shifted darker in order to deal with the dark real situations. The color reflection image is randomly converted into grayscale image and blurred with Gaussian filter ($\sigma \in [0.2, 7]$ in the case of grayscale, $\sigma \in [2, 4]$ in the case of RGB) and the tone is modified randomly. The visualization of synthetic gray-scaled reflection image and color reflection image are shown Fig. 8. The reflection model is selected randomly from Eq. (2)-(4):

$$R = \begin{cases} R1, & \text{with probability } 0.1 \\ R2, & \text{with probability } 0.1 \\ R3, & \text{otherwise.} \end{cases} \quad (16)$$

The synthetic reflection images I for the training are generated by using Eq. (1) and clipped to the range [0, 1]. Since the image pairs for training are generated before every iteration, 3.4M image pairs are finally used in our training.

E. TRAINING

Since to remove gray-scaled reflection is easier than removing color reflection, we first trained our network by using the images only include gray-scaled reflections. After training the network for 50 epochs, we initialize a new network with the trained weights. The new network is trained for 100 epochs by using the dataset in Sec. IV-D. We train our generator by minimizing Eq. (7) where μ_1, μ_2, μ_3, and μ_4 is set to 2, 1, 0.001, and 0.01, respectively. The implementation of our model is based on PyTorch [29] and an Adam solver [30] is used for the optimization. The initial learning rate is set to 0.0002 and the batch size is set to 8. It takes about 40 hours to train the network on a single GeForce GTX 1080 Ti.

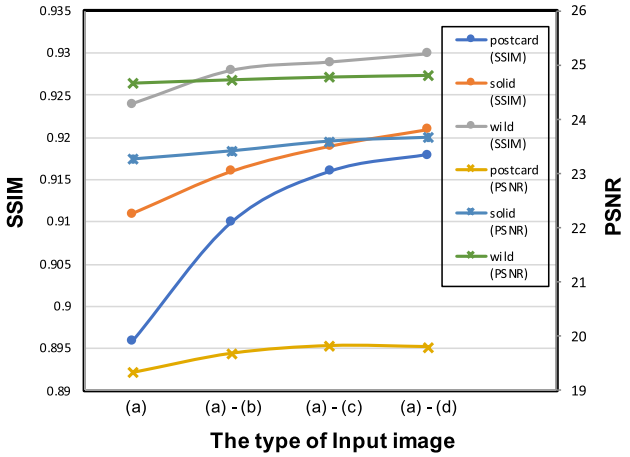


FIGURE 9. The comparison of generated results in PSNR and SSIM. (a)-(c) means that we use (a), (b), and (c) as the input image of our network and we average all of the output images to get the final image.

F. ROTATE AVERAGING PROCESS

Since the proposed network is not rotationally invariant, the results will change when the rotated image is processed. In our method, we propose a rotate averaging process, which averages the several output images generated from the rotated input images. Images in SIR^2 benchmark dataset [31] are used for the evaluation. We prepared four kinds of input images: (a) unprocessed image, (b) 90 degrees rotated image, (c) 180 degrees rotated image, and (d) 270 degrees rotated image. The comparison of generated results in PSNR and SSIM is shown in Fig. 9. We can see that when we use all the four images, the recovered image quality is the highest. Thus, in our method, we use four rotated images for the input and average all of the output images to get the final image.

V. EXPERIMENTAL RESULTS

In this section, we compare our Gradient Constrained Network (GCNet) with other notable methods, including CEIL Net [12], BDN [15], PL [16], ERR [17]. Images in SIR^2 benchmark dataset [31] are used for the objective evaluation. We use PSNR [32] and SSIM [32] to assess the performance. PSNR value provides the numerical differences between two images and SSIM value provides the structural differences between two images. Since reflection removal is an ill-posed problem and the transmittance rate cannot be decided, SSIM value is more important to measure the background image quality. We use real images provided by the authors of [12], [17], [31], [33] for the subjective evaluation. All the comparison methods are implemented by the original authors.

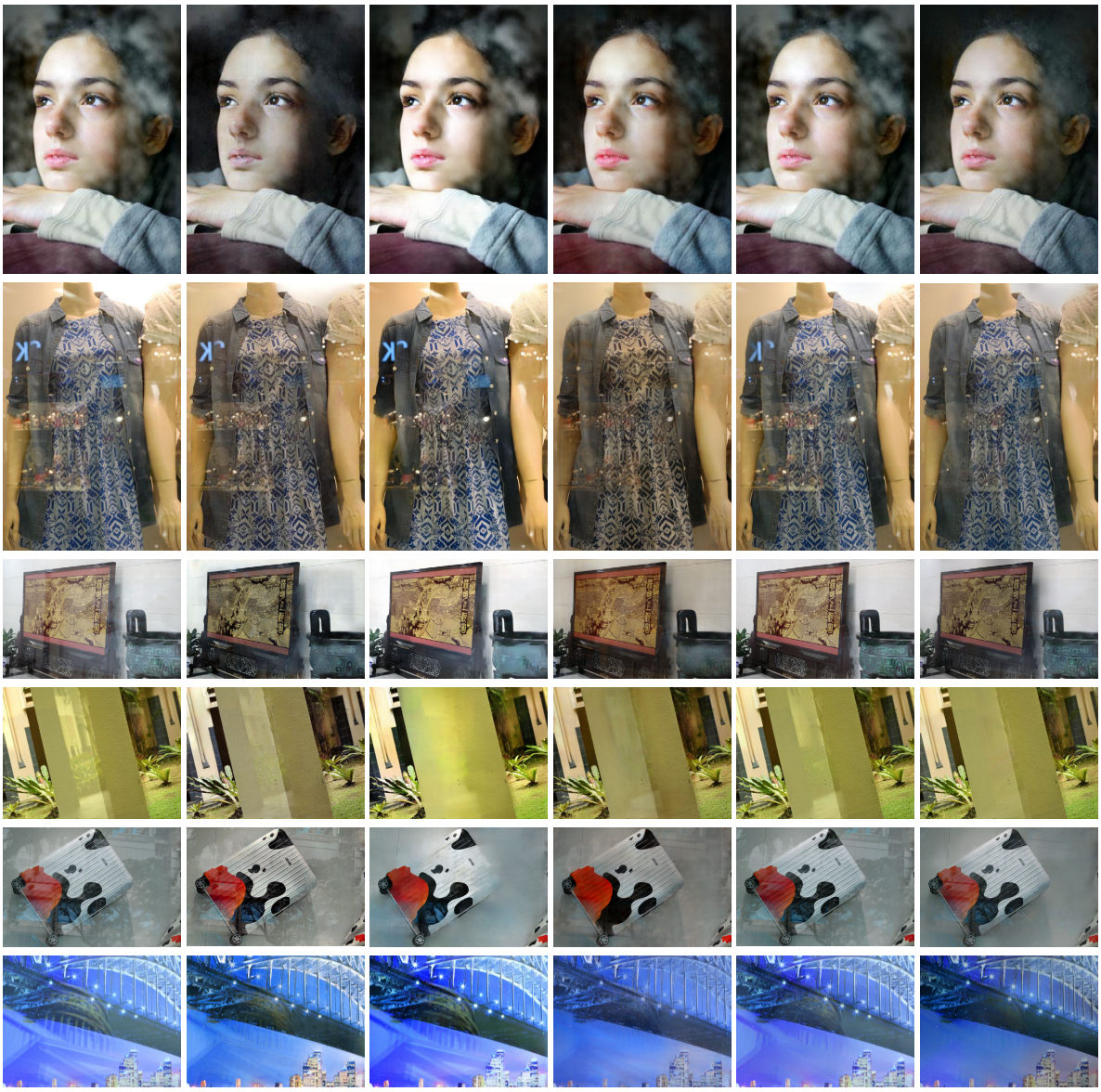


FIGURE 10. Reflection removal results on real images. Images are from [12], [17], [33], and [31]. The top five images do not have ground truth images. Best viewed on screen with zoom.

TABLE 1. Comparison on restoration result in PSNR and SSIM. Images in SIR^2 benchmark dataset [31] are used for the benchmark. *Postcard* includes 199 images, *Solid* includes 200 images, and *Wild* includes 55 images. Higher is better. All the comparison methods are implemented by the original authors.

	Methods (PSNR / SSIM)				
Dataset	CEIL [12]	BDN [15]	PL [16]	ERR [17]	GCNet (ours)
Postcard	19.98 / 0.800	20.54 / 0.849	15.80 / 0.597	**21.81** / 0.866	19.64 / **0.918**
Solid	23.17 / 0.831	22.70 / 0.830	22.14 / 0.775	**24.77** / 0.868	23.87 / **0.928**
Wild	20.84 / 0.794	22.00 / 0.825	21.15 / 0.828	23.73 / 0.865	**24.97** / **0.932**

A. RESULTS

Images in SIR^2 benchmark dataset [31] are used for the benchmark. SIR^2 includes two types of real reflection images: controlled scenes and wild scenes. Controlled scenes are collected in a controlled environment such as in a laboratory. Postcards and daily solid objects are selected as subjects for photography and the datasets includes 199 and 200 images, respectively. Images in wild scenes dataset are collected in a real world out of a lab. Since wild scenes dataset includes complex reflectance, various distances, and different illumination, it is more difficult to remove reflections than the controlled scenes dataset. Table 1 shows the comparison on restoration results in PSNR and SSIM. From Table 1, we can see that our proposed method achieves much higher

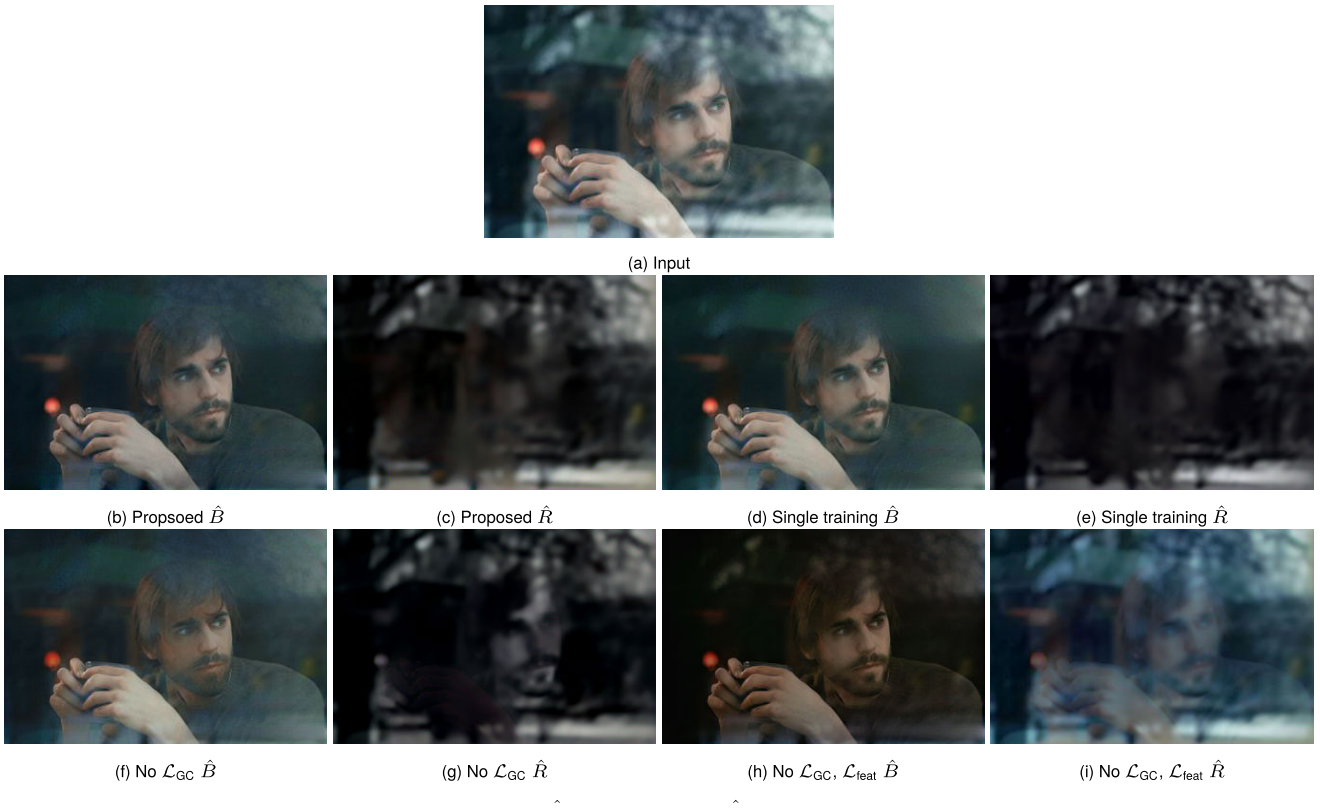


FIGURE 11. Comparison on reflection removal results. Background layers $\hat{B}$ and reflection layers $\hat{R}$ are shown. (a) Input image. (b-c) Our proposed method. (d-e) Trained without pre-training. (f-g) Trained without using $\mathcal{L}_{GC}$. (h-i) Trained without using $\mathcal{L}_{GC}$ and $\mathcal{L}_{feat}$. The ground truth image is not provided.

SSIM than conventional methods in all datasets. In particular, our method shows good results when the wild scenes are processed. This is because our method uses various synthetic reflection images for the training. In addition, the high SSIM shows that the gradient constraint is effective for separating background layer and reflection layer while preserving the image structure.

The subjective evaluation on real images is performed in Fig. 10. We compare our method with CEIL Net [12], BDN [15], PL [16], and ERR [17]. The real images are provided by the authors of [12], [17], [33], and [31]. We can see that BDN and ERR are not good at removing real reflections. CEIL Net and PL can remove reflections effectively but the global tone of the images are changed in some situations. Since we use gradient constraint loss and feature loss in the training process, our proposed network is good at removing reflections effectively while preserving the textures well. The results which include ghosting artifacts are shown in the lower end of Fig. 10. Since we use Eq. (3) for the training data generation, our method can remove ghosting artifacts effectively.

B. THE EFFECTIVENESS OF OUR LOSS FUNCTION AND TRAINING METHOD

When we train our generator, a combination of four kinds of losses is minimized as remarked in Section IV-B. In addition, our proposed network is trained in two steps as remarked in Section IV-E. To show the effectiveness of our loss function and training method, we trained our network in several situations.

TABLE 2. Comparison on restoration result of our methods. Images in SIR^2 benchmark dataset [31] and synthetic reflection images generated by using Eq. (4) are used for the benchmark. *Synthetic* includes 100 images, *Postcard* includes 199 images, *Solid* includes 200 images, and *Wild* includes 55 images. Higher is better. (a) Our proposed method. (b) Trained without pre-training. (c) Trained without using $\mathcal{L}_{GC}$. (d) Trained without using $\mathcal{L}_{GC}$ and $\mathcal{L}_{feat}$.

	Methods (PSNR / SSIM)			
Dataset	Proposed	Single training	No $\mathcal{L}_{GC}$	No $\mathcal{L}_{GC}$, $\mathcal{L}_{feat}$
Synthetic	**23.45 / 0.896**	19.82 / 0.872	19.29 / 0.867	16.12 / 0.824
Postcard	19.64 / **0.918**	**21.60 / 0.918**	20.52 / 0.915	11.68 / 0.757
Solid	**23.87 / 0.928**	23.25 / 0.917	22.62 / 0.910	15.73 / 0.817
Wild	**24.97 / 0.932**	23.70 / 0.925	22.17 / 0.908	17.63 / 0.825

As remarked in Section IV-E, we train our network by changing the dataset in the first step and the second step. Hence, we trained our network without dividing in two steps and used single dataset in order to validate the effectiveness of our training method. We show this restoration result as "Single training". We also trained our generator by ablating some loss functions. "No $\mathcal{L}_{GC}$" indicates that gradient constraint loss is removed from the loss function when the training is performed. "No $\mathcal{L}_{GC}$, $\mathcal{L}_{feat}$" indicates that gradient constraint loss and feature loss are removed from the loss function. In the other words, loss function is composed of pixel loss and adversarial loss in this case.

TABLE 3. Comparison on the execution time. N/A means that the memory of the GPU is lacking to process an image.

Image size	Methods				
	CEIL [12]	BDN [15]	PL [16]	ERR [17]	GCNet (ours)
512×512 (GPU)	**0.01 s**	0.06 s	-	0.89 s	0.17 s
1MP (GPU)	N/A	**0.09 s**	-	N/A	0.66 s
512 × 512 (CPU)	-	**0.68 s**	5.47 s	13.24 s	9.13 s

The comparison of the restoration result is shown in Table 2 and Fig. 11. We can see that the proposed training method achieves the highest PSNR and SSIM in most situation. By the visual result of "No $\mathcal{L}_{GC}$" and "No $\mathcal{L}_{GC}$, $\mathcal{L}_{feat}$", we can say that the gradient constraint loss is effective to separate the background layer and the reflection layer. The texture of the background layer should not appear in the reflection layer but in Fig. 11g and Fig. 11i, we can recognize that the texture of the face appears in the reflection layer. Hence, we can say that minimizing the correlation between the background layer and the reflection layer by using gradient constraint loss is efficacious. By the result of "Single training", we can say that considering the character of the reflection is meaningful in the stage of training. In other words, when solving challenging problem by using learning-based method, finetuning of the network by considering the behavior of the problem is an effective way to train the network.

C. RUNNING TIME

Since our method is based on a deep CNN, we suppose that our method is run on GPU. Our execution environment is Intel Xeon CPU E5-1650 v4 @ 3.60GHz, 64 GB RAM and GeForce GTX 1080 Ti. The comparison on the execution time is shown in Table 3. When the rotating process is not performed, the execution time will be 4 × faster.

VI. CONCLUSION

In this paper, we have proposed a novel Gradient Constrained Network (GCNet) for single-image reflection removal. Four kinds of loss functions are combined to train the network and gradient constraint loss is a new loss function which we have proposed. Since the independence between background layer and reflection layer should be considered, the gradient constraint loss which minimizes the correlation between these two layers improves the performance for reflection removal. Owing to the novel loss, new synthetic dataset, and training method, our method can remove reflection more clearly than state-of-the-art methods. Both quantitative and qualitative evaluation results show that our proposed network preserves the background textures well and the image structure is not corrupted.

REFERENCES

[1] R. Szeliski, S. Avidan, and P. Anandan, "Layer extraction from multiple images containing reflections and transparency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2000, pp. 246–253.

[2] B. Sarel and M. Irani, "Separating transparent layers through layer information exchange," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2004, pp. 328–341.

[3] A. Agrawal, R. Raskar, S. K. Nayar, and Y. Li, "Removing photography artifacts using gradient projection and flash-exposure sampling," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 828–835, 2005.

[4] S. N. Sinha, J. Kopf, M. Goesele, D. Scharstein, and R. Szeliski, "Image-based rendering for scenes with reflections," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 100:1–100:10, Jul. 2012. doi: 10.1145/2185520.2185596.

[5] X. Guo, X. Cao, and Y. Ma, "Robust separation of reflection from multiple images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2187–2194.

[6] T. Xue, M. Rubinstein, C. Liu, and W. T. Freeman, "A computational approach for obstruction-free photography," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 79.

[7] B.-J. Han and J.-Y. Sim, "Reflection removal using low-rank matrix completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5438–5446.

[8] A. Punnappurath and M. S. Brown, "Reflection removal using a dual-pixel sensor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 1556–1565.

[9] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman, "Reflection removal using ghosting cues," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3193–3201.

[10] N. Arvanitopoulos, R. Achanta, and S. Susstrunk, "Single image reflection suppression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4498–4506.

[11] Y. Yang, W. Ma, Y. Zheng, J.-F. Cai, and W. Xu, "Fast single image reflection suppression via convex optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 8141–8149.

[12] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf, "A generic deep architecture for single image reflection removal and image smoothing," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3238–3247.

[13] Z. Chi, X. Wu, X. Shu, and J. Gu, "Single image reflection removal using deep encoder-decoder network," 2018, *arXiv:1802.00094*. [Online]. Available: https://arxiv.org/abs/1802.00094

[14] Y. Chang and C. Jung, "Single image reflection removal using convolutional neural networks," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1954–1966, Apr. 2019.

[15] J. Yang, D. Gong, L. Liu, and Q. Shi, "Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 654–669.

[16] X. Zhang, R. Ng, and Q. Chen, "Single image reflection separation with perceptual losses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4786–4794.

[17] K. Wei, J. Yang, Y. Fu, D. Wipf, and H. Huang, "Single image reflection removal exploiting misaligned training data and network enhancements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 8178–8187.

[18] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5485–5493.

[19] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4681–4690.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[22] K. Nazeri, E. Ng, and M. Ebrahimi, "Image colorization using generative adversarial networks," in *Proc. Int. Conf. Articulated Motion Deformable Objects*. Cham, Switzerland: Springer, 2018, pp. 85–94.

[23] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3155–3164.
[24] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2018, pp. 3–11.
[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167
[26] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*. [Online]. Available: https://arxiv.org/abs/1505.00853
[27] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2794–2802.
[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Proc. NIPS*, 2017, pp. 1–4.
[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[31] R. Wan, B. Shi, L.-Y. Duan, A.-H. Tan, and A. C. Kot, "Benchmarking single-image reflection removal algorithms," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3922–3930.
[32] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.
[33] Y. Li and M. S. Brown, "Exploiting reflection change for automatic reflection removal," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2432–2439.

RYO ABIKO received the B.E. degree in electrical engineering from Keio University, Yokohama, Japan, in 2018, where he is currently pursuing the M.E. degree, under the supervision of Prof. M. Ikehara. His research interests include image filtering, denoising, and image processing using deep learning.

MASAAKI IKEHARA received the B.E., M.E., and Dr. Eng. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1984, 1986, and 1989, respectively. He was appointed as a Lecturer at Nagasaki University, Nagasaki, Japan, from 1989 to 1992. In 1992, he joined the Faculty of Engineering, Keio University, where he is currently a Full Professor with the Department of Electronics and Electrical Engineering. From 1996 to 1998, he was a Visiting Researcher with the University of Wisconsin–Madison and Boston University, Boston, MA, USA. His research interests include multi-rate signal processing, wavelet image coding, and filter design problems.