Single Image Reflection Removal Using DeepLabv3+

Keisuke Hamamoto(✉), Naoya Hideshima, Huimin Lu, and Seiichi Serikawa

Kyushu Institute of Technology, Kitakyushu, Fukuoka, Japan
hamamoto.keisuke806@mail.kyutech.jp

Abstract. When photographing near transparent objects such as glass, things in the area often appear in the image (reflected light). This reflected light degrades the image information and affects computer vision task such as object detection and segmentation. Separating reflected light in an image is a challenging task in computer vision. In this study, we create synthetic images and increase the number of training data. We proposed a reflection rejection method that used DeepLabv3+ as a deep learning model and measured the accuracy of the proposed method using PSNR and SSIM, commonly used evaluation methods for reflection rejection. The accuracy of the proposed method is improved compared to the conventional method due to the feature of DeepLabv3+ to obtain information in a wide range of contexts efficiently.

Keywords: Deep Learning · SIRR · DeepLabv3

1 Background

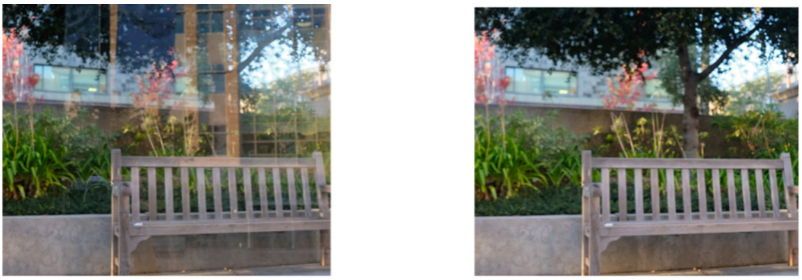
When photographing near-transparent objects such as glass, reflected light can appear in the image, as shown in Fig. 1. This loss of image information due to reflected light can frequently occur, making SIRR a challenging problem that has attracted considerable attention in the computer vision community. The image containing reflected light consists mathematically of the image I, a linear combination of a transmitted image layer T and a reflected image layer R, as in Eq. (1).

$$I = T + R \tag{1}$$

Hence, reflected light rejection can achieve its goal by estimating the transmitted image layer T. Many researchers have tackled the technical challenge of reflected light rejection. Many solutions have been proposed. However, many currently have limitations in performance, robustness, and versatility.

In the early statistical models of SIRR, removing the reflective layer from a single image was avoided because it was impossible to separate the transmission layer from the reflective layer. Conversely, multiple images have been used to estimate the transmission layer T [1–3]. The problem has been solved by adding and formulating constraints to the images. And even when only a single image is used, the transmission layer T has been estimated using the formulated equation [4–6].

H. Lu and J. Cai (Eds.): ISAIR 2023, CCIS 1998, pp. 181–188, 2024.
https://doi.org/10.1007/978-981-99-9109-9_18



(a) Images with reflected light (b) Images without reflected light

Fig. 1. Examples of images containing reflected light

However, it is difficult to construct a versatile light-reflectance removal model by simply adding these simple constraint conditions to image processing since various assumed situations are possible. Against this background, research on constructing deep learning models has been active in recent years [7–9].

There are two problems with SIRR using deep learning models [10, 11]. One is that extracting background images without reflection is illogical, and the other is that the training data is tiny. For the former, the performance of the model is limited. The latter problem arises because of the difficulty in obtaining paired datasets of images with and without reflections. Because of the rare case of reflected light, rather than a simple dataset such as Image Net or MNIST. Therefore, in SIRR, synthetic images obtained by merging images are often used because it is difficult to get true values.

2 Methods

2.1 Proposed Model

The network model proposed in this study is shown in Fig. 1. Six-layer convolution is performed in the encoder part. The bottleneck part employs Deeplabv3+ [12], followed by six-layer convolution in the decoder part to obtain the estimated transmitted image layer $\breve{T}$ and the estimated reflected image layer $\breve{R}$ as outputs.

The bottleneck part, the Deeplabv3+ module, uses MobileNetv2 [13] as the backbone. This is followed by ASPP (Atlas Spatial Pyramid Pooling), which uses Image Pooling with atlas convolution rates of 1, 6, 12, and 18, respectively (Figs. 2 and 3).

2.2 Loss Function

As the basis of many neural networks, in image restoration techniques, the loss function is generally optimized for the network using the mean squared error (MSE) between the output and the true value.

$$L_{MSE} = \|F(I) - T\|_2^2 \tag{2}$$

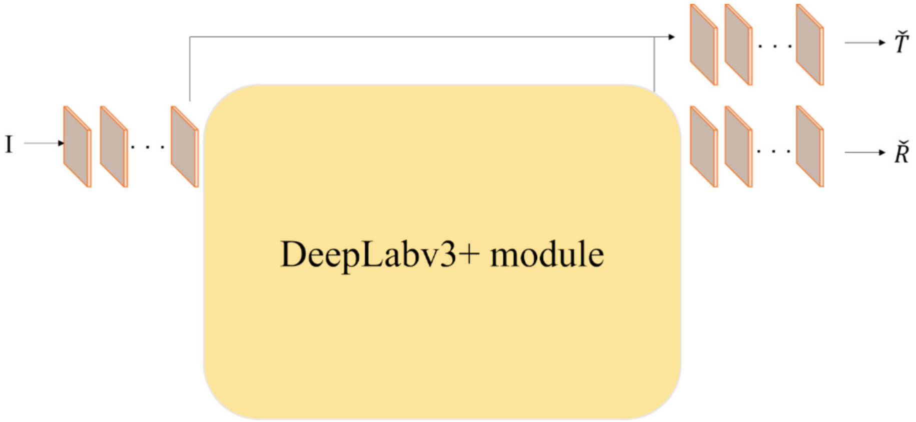


Fig. 2. Proposal Network

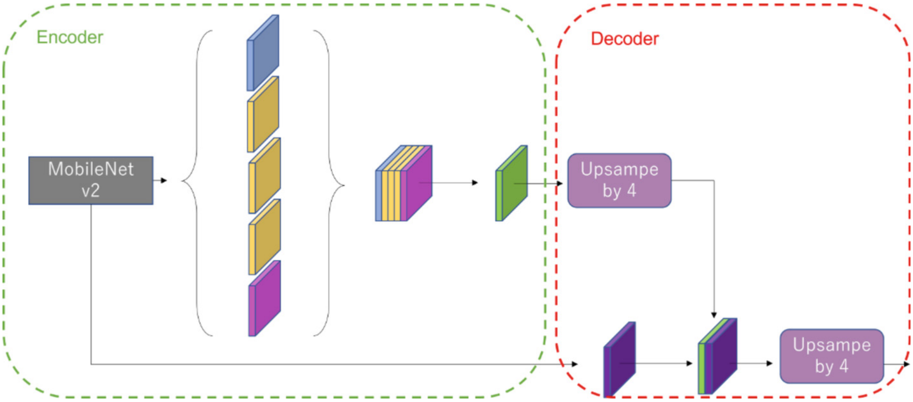


Fig. 3. DeepLabv3+ module

However, F is the data after processing. However, models optimized using only L_{MSE} often fail to retain the high-frequency content. In the case of de-reflection, both the reflective and the transparent layers are natural images with different characteristics. To obtain the best restoration results, the network needs to learn the perceptual properties of the transmission layer. Therefore, we adopt a loss function close to a high-level feature abstraction. The VGG loss is calculated as the difference between the layer representation of the restored transmission and the actual transmission image on the pre-trained 19-layer VGG network proposed by Simonyan and Zisserman [14].

$$L_{VGG} = \frac{1}{W_i H_i} \sum_{i=1}^{M} \|\varphi_i(T) - \varphi_i(F(I))\|_2^2 \tag{3}$$

where φ_i is the feature map obtained by the i-th convolutional layer (after activation) in the VGG19 network, M is the number of convolutional layers used, W_i and H_i are the dimensions of the i-th feature map.

This study uses the loss function consisting of the sum of these two loss functions. The equation is expressed as follows.

$$L = L_{MSE} + \lambda L_{VGG} \tag{4}$$

λ is a parameter of L_{VGG} and is set to 0.1 in this study.

2.3 Dataset Creation

Equation (1) will be used in this study as well. The Eq. (1) multiplied by the transmittance α is given below.

$$I = \alpha T + R \tag{5}$$

In this study, two R patterns in Eq. (5) are set and used.

The first equation is,

$$R = \beta G * R' \tag{6}$$

R' is the reflective image layer, β is the reflectance, and G is the Gaussian kernel.

The second equation is,

$$R = G * R' - \gamma \tag{7}$$

R' is the reflective image layer, γ is a constant, and G is the Gaussian kernel.

The values of α, β, γ, and G are varied as there are various patterns of reflected light in the real image. In some cases, gamma correction is used to darken R.

3 Experiments and Results

3.1 Dataset

3890 images from the MIT-67 Dataset [15] and 17000 images from the PASCAL VOC 2012 Dataset [16] were collected to generate images with pseudo-reflected light using Eqs. (5), (6), and (7) as the training dataset. The datasets used for the evaluation were Object, Post, and Wild from the SIR2 Dataset [17] and the Real20 Dataset [18]. These datasets are real images, not pseudo-synthesized images containing reflected light. The number of each type of data used in the evaluation is shown in Table 1.

3.2 Experimental Procedure

In this study, training was carried out using the same learning setup under the same conditions to perform a control experiment. The training was carried out with a batch size of 16, 100 epochs, a learning rate of 0.0003, and Adam as optimizer. PSNR and SSIM were used as evaluation metrics [19–21]. The GPU used for training was an NVIDIA GeForce RTX 3090, the CPU was an Intel Core i9, the RAM was 64GB and the OS was Ubuntu 20.04 [22]. The combinations used in the experiments are listed in Table 2.

Table 1. Number of data used in the evaluation.

Datasets	Number of datasets
Object	20
Post	20
Wild	101
Real20	109

Table 2. Proposed methods

	Proposed method 1	Proposed method 2
Proposed Network	○	○
Conventional reflectance generation model	○	
Proposed reflection generation model		○

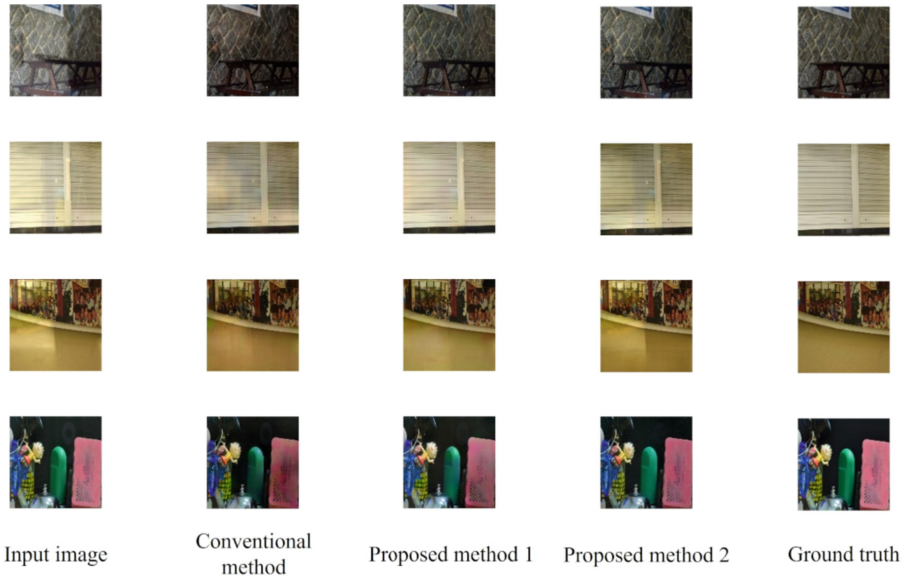


Fig. 4. Comparison of output images

3.3 Experimental Results

The experimental results are shown in Fig. 4. The evaluation indices are given in Table 3. Note that the values shown in the table are average.

Table 3. Results of experiments

Data set	Evaluation metrics	Conventional method	Proposed method 1	Proposed method 2
Object	PSNR	21.57	22.98	23.57
	SSIM	0.82	0.84	0.86
Post	PSNR	15.93	21.54	16.91
	SSIM	0.72	0.79	0.77
Wild	PSNR	23.60	25.23	24.91
	SSIM	0.83	0.86	0.87
Real20	PSNR	17.79	19.30	18.85
	SSIM	0.65	0.70	0.72

4 Discussion

The results in Fig. 4 and Table 3 are discussed. We also discuss each of the proposed methods listed in Table 2.

First, a comparison is made between the conventional and proposed methods 1. These are controlled experiments because the network models are different, and the reflection formation model is the same. Table 3 shows that the accuracy of the proposed method 1 is superior. The resultant images in Fig. 4 show that the reflections are removed. It can also be seen that the pixel values do not drop much. It is considered that the network model (DeepLabv3+) of the proposed method 1 has a significant influence on the removal of reflected light. It is also considered that the ASPP in the DeepLabv3+ structure plays a role.

Next, we compare the proposed method 1 with the proposed method 2. These are contrasting experiments under the condition that the network models are the same and the reflection formation models are different. From Table 3, it is impossible to say which method is better. Similarly, the resultant images in Fig. 4 show different results depending on the image. The reflection formation model influences these. The reflection formation model affects the results by the similarity with the real image.

The proposed method is superior to the conventional methods [23]. However, even with the proposed method, there were some images where the reflection could not be removed. There are two possible reasons for this. The first is the network model. As can be seen from this study, the results vary greatly depending on the network model. Constructing and improving a network model suitable for reflection removal is necessary. The second is the reflection formation model. As the images created by the reflection formation model are used for training, the real data must approximate it.

Moreover, as can be seen from the results, there are various patterns of reflected light in the real data [24], and images must be created for each. In the future, solving these two causes will lead to the removal of reflected light. Furthermore, in this study, learning was carried out using only synthetic data, but learning using real data is also considered one of the measures.

5 Conclusion

DeepLabv3+ is proposed as a deep learning model for single-image reflection removal in this paper. A reflection formation model is also proposed, and a synthetic image is generated.

Experiments are conducted on four datasets commonly used in the SIRR field to compare the proposed methods. The proposed method shows better results through the experiments than the conventional methods. It is also confirmed that the results are affected by the different reflection formation models used in the synthetic data. Although the proposed method in this paper was trained only on the synthetic data, it gave excellent results on the real data.

Future tasks are to study the construction of a model more suitable for removing reflections, to study a reflection formation model similar to real data, and to study learning with real data. Learning using real data is the most effective method, and transfer learning, and meta-learning can be used for this purpose.

References

1. Xue, T., et al.: A computational approach for obstruction-free photography. ACM Trans. Graph. **34**(4), 1–11 (2015)
2. Guo, X., et al.: Robust separation of reflection from multiple images. In: CVPR, pp. 2195–2202 IEEE Computer Society (2014)
3. Lu, H., Li, Y., Nakashima, S., et al.: Underwater image super-resolution by descattering and fusion. IEEE Access **5**, 670–679 (2017)
4. Levin, A., et al.: Separating reflections from a single image using local features. In: CVPR (1). pp. 306–313 (2004)
5. Li, Y., Brown, M.S.: Single image layer separation using relative smoothness. In: CVPR, pp. 2752–2759. IEEE Computer Society (2014)
6. Lu, H., Yang, R., Deng, Z., et al.: Chinese image captioning via fuzzy attention-based DenseNet-BiLSTM. ACM Trans. Multimedia Comput. Commun. Appl. **17**(1s), 1–18 (2021)
7. Chi, Z., et al.: Single Image Reflection Removal Using Deep Encoder-Decoder Network. CoRR. abs/1802.00094 (2018)
8. Wen, Q., et al.: Single Image Reflection Removal Beyond Linearity. In: CVPR, pp. 3771–3779. Computer Vision Foundation/IEEE (2019)
9. Li, T., et al.: Improved multiple-image-based reflection removal algorithm using deep neural networks. IEEE Trans. Image Process. **30**, 68–79 (2021)
10. Lu, H., Li, Y., Chen, M., et al.: Brain Intelligence: go beyond artificial intelligence. Mobile Netw. Appl. **23**, 368–375 (2018)
11. Lu, H., Tang, Y., Sun, Y.: DRRS-BC: decentralized routing registration system based on blockchain. IEEE/CAA J. Autom. Sinica **8**(12), 1868–1876 (2021)
12. Chen, L.C., et al.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
13. Sandler, M., et al.: MobileNetV2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018)
14. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/abs/1409.1556 (2014)

15. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: CVPR, pp. 413–420 IEEE Computer Society (2009)
16. Everingham, M., et al.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results (2012)
17. Wan, R., et al.: Benchmarking single-image reflection removal algorithms. In: ICCV, pp. 3942–3950. IEEE Computer Society (2017)
18. Zhang, X.C., et al.: Single image reflection separation with perceptual losses. In: CVPR, pp. 4786–4794. IEEE Computer Society (2018)
19. Lu, H., Qin, M., Zhang, F., et al.: RSCNN: a CNN-based method to enhance low-light remote-sensing images. Remote Sens. **13**(1), 62 (2020)
20. Lu, H., Wang, D., Li, Y., et al.: CONet: a cognitive ocean network. IEEE Wirel. Commun. **26**(3), 90–96 (2019)
21. Xu, X., Lu, H., Song, J., et al.: Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval. IEEE Trans. Cybern. **50**(6), 2400–2413 (2020)
22. Lu, H., Zhang, Y., Li, Y., et al.: User-oriented virtual mobile network resource management for vehicle communications. IEEE Trans. Intell. Transport. Syst. **22**(6), 3521–3532 (2021)
23. Lu, H., Li, Y., Mu, S., et al.: Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. IEEE Internet Things J. **5**(4), 2315–2322 (2018)
24. Lu, H., Zhang, M., Xu, X., et al.: Deep fuzzy hashing network for efficient image retrieval. IEEE Trans. Fuzzy Syst. **29**(1), 166–176 (2021)