

Article

Restoring Raindrops Using Attentive Generative Adversarial Networks

Suhan Goo and Hee-Deok Yang * 

Department of Computer Engineering, Chosun University, 309 Pilmun-Daero, Dong-Gu, Gwangju 61452, Republic of Korea

* Correspondence: heedeok_yang@chosun.ac.kr

Abstract: Artificial intelligence technologies and vision systems are used in various devices, such as automotive navigation systems, object-tracking systems, and intelligent closed-circuit televisions. In particular, outdoor vision systems have been applied across numerous fields of analysis. Despite their widespread use, current systems work well under good weather conditions. They cannot account for inclement conditions, such as rain, fog, mist, and snow. Images captured under inclement conditions degrade the performance of vision systems. Vision systems need to detect, recognize, and remove noise because of rain, snow, and mist to boost the performance of the algorithms employed in image processing. Several studies have targeted the removal of noise resulting from inclement conditions. We focused on eliminating the effects of raindrops on images captured with outdoor vision systems in which the camera was exposed to rain. An attentive generative adversarial network (ATTGAN) was used to remove raindrops from the images. This network was composed of two parts: an attentive-recurrent network and a contextual autoencoder. The ATTGAN generated an attention map to detect rain droplets. A de-rained image was generated by increasing the number of attentive-recurrent network layers. We increased the number of visual attentive-recurrent network layers in order to prevent gradient sparsity so that the entire generation was more stable against the network without preventing the network from converging. The experimental results confirmed that the extended ATTGAN could effectively remove various types of raindrops from images.

Keywords: raindrops; attentive generative adversarial network; convolutional neural networks



Citation: Goo, S.; Yang, H.-D. Restoring Raindrops Using Attentive Generative Adversarial Networks. *Appl. Sci.* **2021**, *11*, 7034. <https://doi.org/10.3390/app11157034>

Academic Editor: Amjad Anvari-Moghaddam

Received: 3 July 2021
Accepted: 29 July 2021
Published: 30 July 2021
Corrected: 28 March 2024

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Vision systems are often used in various devices, such as automotive navigation systems, object-tracking systems, and intelligent closed-circuit televisions. In particular, external vision systems are widely used in various analytical fields. Despite their widespread use, current systems only work well under good atmospheric conditions. They cannot account for inclement conditions, such as rain, fog, mist, and snow. Images captured under inclement conditions degrade the performance of vision systems. Vision systems have to automatically detect, recognize, and remove noise due to rain, snow, and mist in order to enhance the performance of the algorithms utilized in image processing. Several studies have focused on removing noise resulting from inclement conditions, such as rain, fog, and snow. Figure 1 shows the ground-truth images and the images generated by adding raindrop effects to the ground-truth images.

In this paper, we propose a new method for restoring raindrops based on an attentive generative adversarial network. An attentive generative adversarial network (ATTGAN) [1,2] was used to remove raindrops from images. It was composed of two parts: an attentive-recurrent network and a contextual autoencoder. The ATTGAN generated an attention map to detect the rain droplets. We increased the number of visual attentive-recurrent network layers in order to prevent gradient sparsity so that the entire generation would be more stable against the network without preventing the network from converg-

ing. A de-rained image was generated by increasing the number of attentive-recurrent network layers.

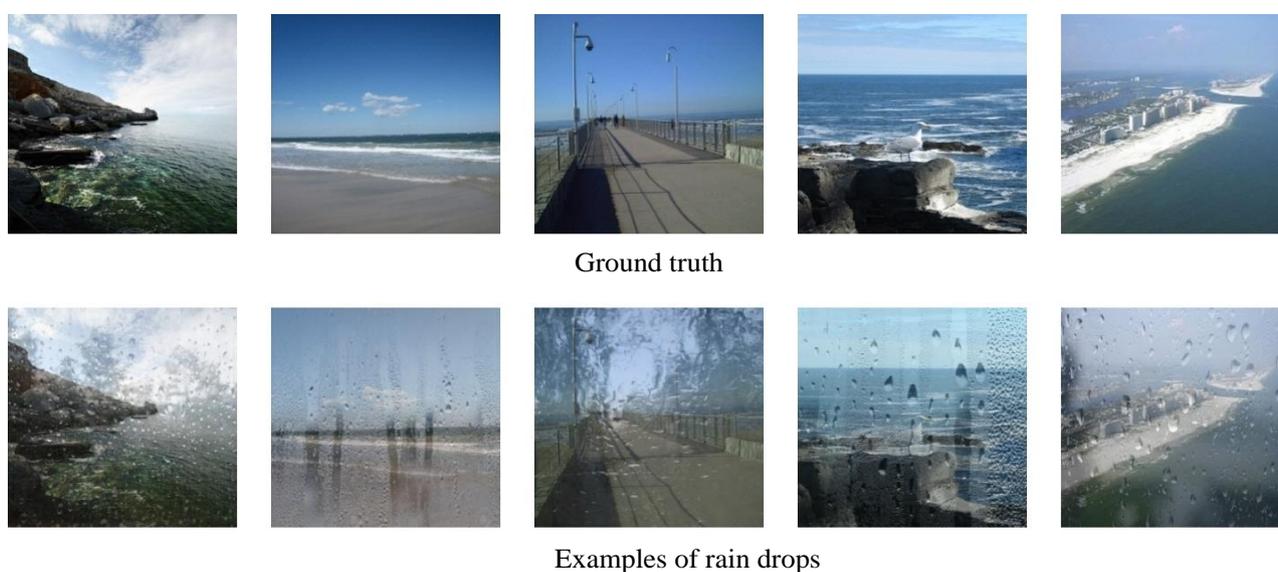


Figure 1. Examples of raindrop images and ground-truth images.

2. Related Works

A few techniques, including strategies using the time and frequency domains, low-rank representation and sparsity-based strategies, Gaussian mixture model strategies, and deep learning techniques, have been utilized to address issues of lucidity in camera images [3–5]. Many rain-removal techniques have been developed. The representative strategies are briefly discussed in this section. For a comprehensive review of downpour-removal strategies, please refer to the overview papers of [3–5].

2.1. Time- and Frequency-Domain-Based Methods

Garg and Nayar examined the impacts of downpour on a vision framework [6]. They utilized a space–time relationship model and movement data to capture the elements of rain and to clarify the photometry of these elements individually.

Zhang et al. applied a histogram model to recognize and eliminate raindrops in an image by utilizing the spatio-temporal properties of rain streaks [7]. They utilized the *K*-means algorithm to construct a histogram model.

Barnum et al. introduced a spatio-temporal frequency-based method to recognize rain and snow [8]. They utilized a physical model and a blurred Gaussian model to estimate the obstruction effects caused by raindrops. However, their proposed blurred Gaussian model could generally not deal with rain streaks.

2.2. Low-Rank Representation and Sparsity-Based Methods

Chen et al. utilized the similarity and repeatability of rain streaks [9]. They proposed the use of a low patch rank before catching rain streak patterns. In addition, they proposed a movement-segmentation-based technique in order to deal with rain streaks.

Hu et al. proposed an iterative layer-separation technique [10]. They separated noisy images into background layers and rain streaks. They eliminated the rain streaks from the background layers.

Zhu et al. proposed an iterative layer-separation technique [11]. They separated the images into rain streaks and background layers. In addition, they eliminated the textures of the background layers and rain streaks with layer-explicit priors.

Deng et al. proposed a sparse directional group model to model rain streaks' sparsity and directions [12].

2.3. Gaussian Mixture Model

Li et al. demonstrated the detection of rain streaks and background layers using Gaussian mixture models (GMMs) [13]. The GMMs of the background layer were acquired from images with different background scenes. A rain patch chosen from an input image that had no background areas was utilized to prepare the GMMs of the rain streaks. Li et al.'s model was able to eliminate rain streaks at small and moderate scales.

2.4. Deep-Learning-Based Methods

The success of convolutional neural networks (CNNs) in several research fields has inspired researchers to develop CNN-based image-denoising methods [14–22].

Yang et al. constructed a joint rain detection and removal network. It could handle heavy rain, overlapping rain streaks, and rain accumulation [14]. The network could detect rain locations by predicting a binary rain mask and using a recurrent framework to remove rain streaks and progressively clear up the accumulation of rain. This network achieved good results in heavy rain cases. However, it could falsely remove vertical textures and generate underexposed illumination. Yang et al. improved and proposed several CNN-based methods [13–16].

Following Yang et al. [14–16] and Fu et al. [17], several other authors proposed CNN-based methods [14–18]. These methods employed more advanced network architectures and the injection of new rain-related priors. They achieved better quantitative and qualitative results.

Fu et al. [17] utilized a two-step technique in which the information of a blurry picture was decayed into a foundation-based layer and an independent detailed layer. At this point, indirect CNN-based planning was used to eliminate the downpour streaks from the detail layer.

Qian et al. [1] built an ATTGAN by infusing visual attention into both the generative and discriminative organizations. The visual attention did not just guide the discriminative organization to zero, but in addition to the nearby consistency of the reestablished raindrop locales, it also caused the generative organization to focus harder on the relevant data encompassing the raindrop territories.

Lee et al. [18] proposed a deep learning method for rain removal in videos based on a recurrent neural network (RNN) architecture. Pseudo-ground truth was generated from real rainy video sequences by temporally filtering through supervised learning instead of focusing on various shapes of rain streaks like conventional methods. They focused on the changes in the behaviors of the rain streaks.

Zhang et al. [19] took one step forward by investigating the construction of feed-forward denoising convolutional neural networks (DnCNNs) in order to embrace the progress in very deep architectures, learning algorithms, and regularization methods for image denoising. Residual learning and batch normalization were utilized in order to speed up the training process, unlike existing discriminative denoising models, which usually train a specific model for additive white Gaussian noise at a certain noise level.

Chen et al. [20] proposed the HIN Block (Half Instance Normalization Block) to boost the performance of image-restoration networks. They proposed a multi-stage network called HINet based on the HIN Block. They applied instance normalization for half of the intermediate features and kept the content information at the same time.

Wang et al. [21] proposed Uformer, an effective and efficient transformer-based architecture, in which they built a hierarchical encoder–decoder network by using the transformer block for image restoration. In contrast to existing CNN-based structures, Uformer built upon the main component, the LeWin transformer block, which can not only handle local context, but can also efficiently capture long-range dependencies.

3. Raindrop Removal with an ATTGAN

3.1. Formation of a Single Waterdrop Image

A rainy image is defined as [1,2]:

$$I = (1 - M) \odot B + W \tag{1}$$

where I , B , and M are the rainy image, background image, and binary mask image, respectively; W is the effect of the water droplets; \odot represents the multiplication operation. M is obtained by subtracting image B from image I . I is generated by adding waterdrop noise. M is the noise region, and B is the other region. The goal is to obtain the background image B from a given input rainy image I . In the mask image,

$$M(x) = \begin{cases} 1 & \text{raindrop region} \\ 0 & \text{background region} \end{cases} \tag{2}$$

where x is a pixel.

3.2. Generative Network

Generative adversarial networks (GANs) are a class of strategies for modeling data distributions, and they consist of two networks: the generator G , which translates an example from an arbitrary uniform distribution into a data distribution, and the discriminator D , which measures the likelihood of whether a given example has a place in the data distribution or not. In light of the hypothetical min-max standards, the generator and discriminator are normally mutually trained by exchanging the preparation of D and G , despite the fact that GANs can produce visually engaging images by preserving high-frequency details [23,24].

Figure 2 shows the overall architecture of the ATTGAN method. The network is composed of two parts: the generative network and the discriminative one. Given an image with raindrops, the generative network generates an image that looks as real as possible and is free from raindrops. The generative network is composed of two parts: an attentive-recurrent network and a contextual autoencoder [23,24]. The aim of an attentive-recurrent network is to find regions of interest in an input image. These regions are the raindrop regions. The discriminative network determines whether the image produced by the generative network looks real or not.

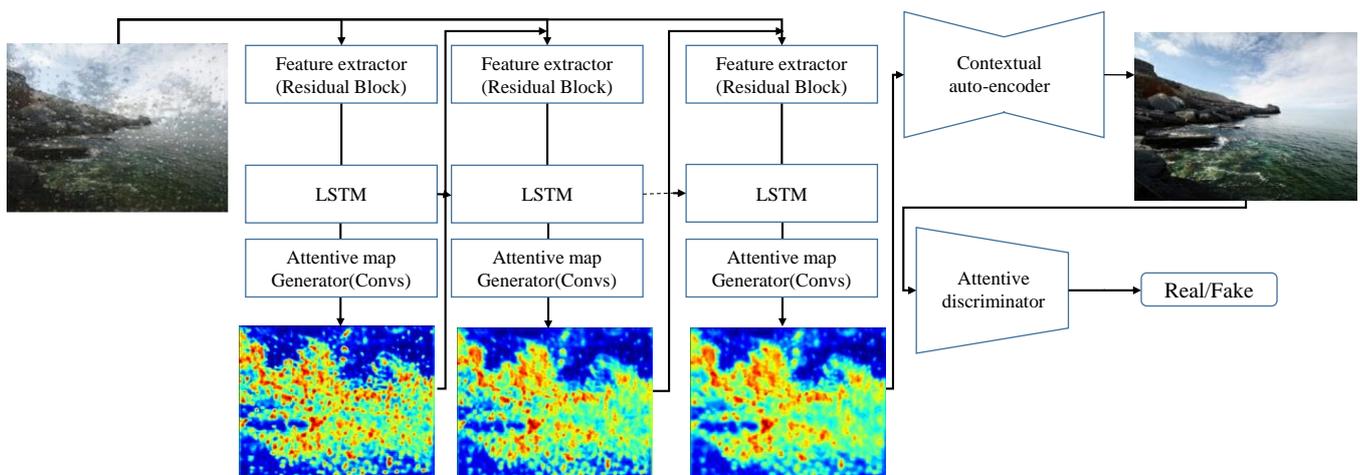


Figure 2. Architecture of the attentive generative adversarial network method.

The overall loss function for the adversarial loss is defined as [1,2]:

$$\min_G \max_D V(D, G) = E_{W \sim p_{clean}} [\log D(W)] + E_{I \sim p_{raindrop}} [\log(1 - D(G(I)))] \quad (3)$$

where W is the de-rained image generated by the generation network and I is a sample drawn from our pool of images that have been degraded by raindrops, which are the inputs of the generative network's truth image.

3.2.1. Attentive-Recurrent Network

A visual attention network was applied to discover the regions of rain droplets in the rainy image inputs [1,2]. To create a visual attention network, we applied a recurrent network. Each layer of the recurrent network was composed of a five-layer-deep residual neural network (ResNet) [22,23], a convolutional long short-term memory (ConvLSTM) network [24], and standard convolutional layers. The ResNet was applied to extract the features from the input image and the mask of the previous block [23]. Each residual block incorporated a two-layer convolution kernel of size 3×3 with a rectified linear unit (ReLU) nonlinear activation function.

The extracted feature map and the initialized attention map were transferred to the ConvLSTM for training. The ConvLSTM unit consisted of an input gate i_t , a forget gate f_t , an output gate o_t , and a cell state C_t . The interactions between the states and the gates along the time dimension are described in detail in [1,2].

The consideration map, which was learned at each time step, was a matrix going from 0 to 1, where the greater its value was, the better the attention map generated would be. In contrast to the binary mask M , the attention map was a non-binary map, and it addressed the expanding attention from the non-raindrop areas to the raindrop areas; the quality changed even inside the raindrop areas. This consideration of the expansion was necessary because the encompassing locales of the raindrops additionally needed the consideration, and the straightforwardness of a raindrop zone actually changed (a few parts did not absolutely block the background and, accordingly, passed on some background data) [1,2].

Pairs of images with and without raindrops that contained the very same background scene were used to train the generative network. The loss function in each recurrent block was characterized as the mean squared error (MSE) between the output attention map at time step t (or A_t) and the binary mask M . We applied this process in N time steps [1,2]. The prior attention maps had more modest qualities and became larger when moving toward the N th time step, demonstrating the increment in certainty.

The loss function in each recurrent block is expressed as [1,2]:

$$L_{ATT}(\{A\}, M) = \sum_{t=1}^N \theta^{N-t} L_{MSE}(M, A_t) \quad (4)$$

where M is the binary mask, A_t is the attention map generated by the recurrent network at time t , N is the number of interactions of the recurrent block, and θ is the weight and was set to 0.8.

3.2.2. Generative Autoencoder

The objective of the generative autoencoder was to produce a refined and clean image that was free from raindrop occlusions and that looked like a genuine picture. The autoencoder consisted of Conv-ReLu blocks, and skip associations were added to prevent a blurred output.

Figure 3 illustrates the autoencoder’s perceptual loss. Perceptual loss measures the global discrepancy between the image created by the autoencoder and the corresponding ground-truth image [1,2].

$$L_{MS}(\{S\}, \{T\}) = \sum_{i=1}^M \lambda_i L_{MSE}(S_i, T_i) \tag{5}$$

where S_i represents the output extracted from the decoder layers and T_i represents the ground truth with the same scale as that of S_i . $\lambda_{i=1}^M$ is the weight for the different scales.

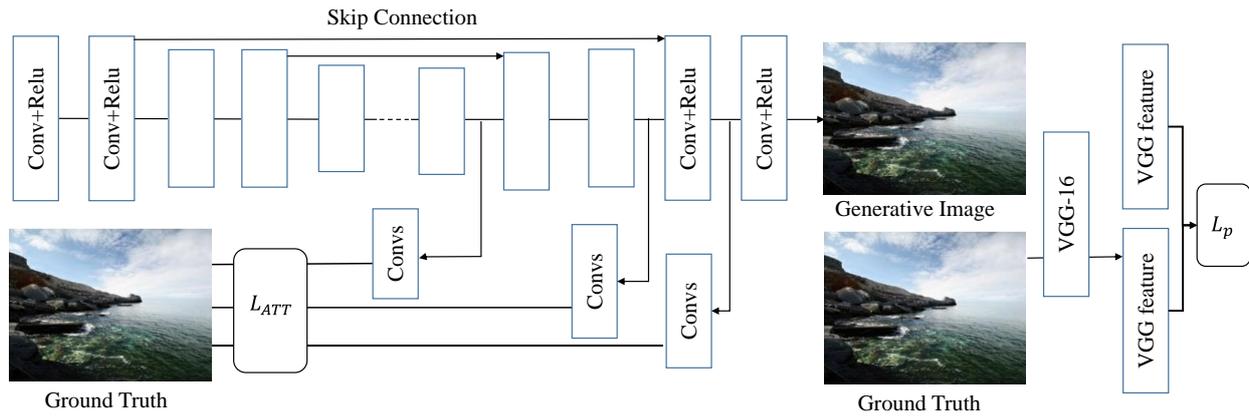


Figure 3. Architecture of the contextual autoencoder. Multiscale loss and perceptual loss are used to help train the autoencoder.

The global features were extracted using a VGG16 model pretrained on the ImageNet dataset. The perceptual loss function is expressed as [1,2]:

$$L_P(O, T) = L_{MSE}(VGG(O), VGG(T)) \tag{6}$$

where $VGG(O)$ and $VGG(T)$ are the features of the output of the autoencoder and the ground-truth image extracted by the pretrained VGG16 model, respectively; O is the output image of the autoencoder, i.e., $O = G(I)$, where I is an input image.

The discriminator loss function of the generative network is expressed as [1,2]:

$$L_{context} = \lambda_g L_{GAN}(G(O)) + L_{ATT}(\{A\}, M) + L_{MS}(\{S\}, T) + L_P((G(O), T)) \tag{7}$$

where $\lambda_g = 10^{-2}$ and $L_{GAN}((G(O)) = L_{MSE}[\log(1 - D(O))]$.

Figure 3 shows the architecture of the contextual autoencoder.

3.3. Discriminative Network

To use the local and global features, we produced an attention map from an attentive-recurrent network. The loss function of the discriminator is expressed as:

$$L_D(O, R, A_N) = -\log(D(R)) - \log(1 - D(O)) + \gamma L_{map}(O, R, A_N) \tag{8}$$

where L_{map} is defined as:

$$L_{map}(O, R, A_N) = L_{MSE}(D_{map}(O) - A_N) + L_{MSE}(D_{map}(R) - 0) \tag{9}$$

where D_{map} represents the process of producing a two-dimensional map using the discriminative network.

The discriminant network consisted of nine convolutional layers. Each layer was associated with the ReLU nonlinear activation function. A 5×5 convolution kernel was

utilized to extract and fuse the texture features. The first six output channels were 8, 16, 32, 64, 128, and 128 [1,2].

4. Experimental Results and Analysis

4.1. Experimental Environment

To train the generative network, we needed pairs of images with and without raindrops. We generated the training data by adding the raindrop effect to the original image and used the public dataset in [25].

We also used a subset of ImageNet. We generated a total of 2500 images and used 10-fold cross-validation for the evaluation. To synthesize the raindrop images, we used 25 filters, and, as shown in Table 1, we divided the waterdrop images into five types according to the raindrop levels.

Table 1. Waterdrop types.

Type	Description
1	medium water mist
2	weak water stream and small water mist
3	strong water drop
4	strong water stream
5	large water drop and strong water fog

The median filter, bilateral filter, cycle GAN (CGAN), and attentive CGAN methods were implemented and compared in the raindrop-removal application. The proposed method was implemented by extending the software in [26].

To measure the accuracy of the proposed method, we used the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

The experiment in this study was carried out on a computer with a 64-bit operating system (Ubuntu v. 18.04), Intel®Core™ i7-6800K CPU at 3.40 GHz, 64 GB of RAM, and GeForce GTX1080 Ti GPU. The TensorFlow 1.10.0 deep learning framework was used for network training.

4.2. Experimental Analysis

Figures 4–8 show the results of the removed waterdrops according to the waterdrop types described in Table 1. Figure 6 shows the results of the waterdrops for type 3. As shown in the results, the proposed method removed most of the waterdrop noise and maintained a high background texture.

Figure 9 shows the waterdrop results according to the waterdrop types. As shown in the results, the PSNR of the proposed method was lower than those of the other methods. Table 2 shows the PSNRs and SSIMs. As shown in the evolution table, the attentive GAN performed better than the other methods.

Table 2. SSIM results of the waterdrops of type 5 (large waterdrops and strong water fog).

Name	Bilateral Filter	Cycle GGAN	ATTGAN	Proposed Method
SSIM	0.562	0.8752	0.9018	0.9124



Figure 4. Results for type 1 waterdrops (medium water mist).



Figure 5. Results for type 2 waterdrops (weak water stream and small water mist).

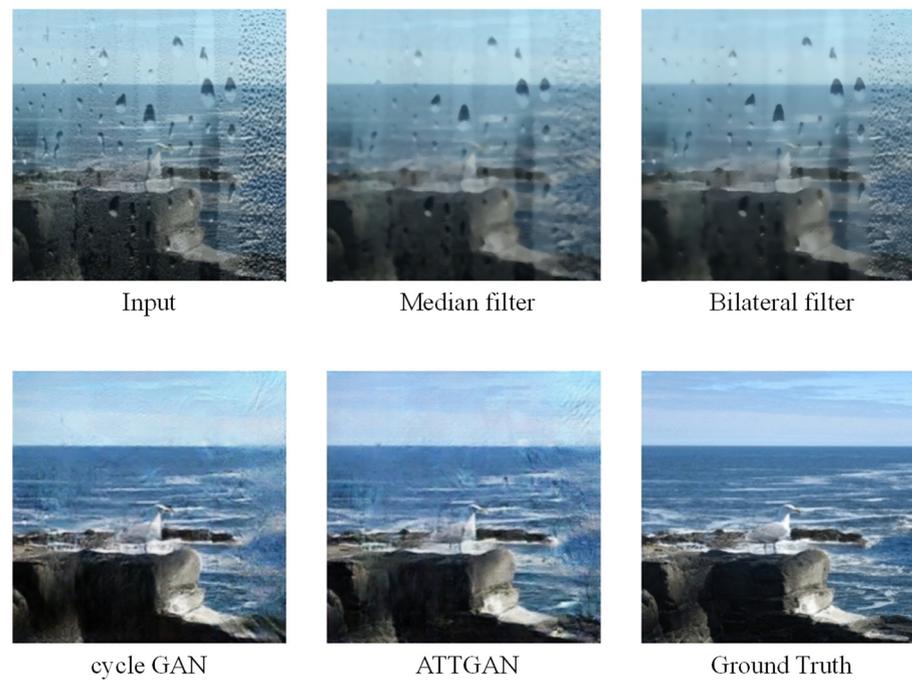


Figure 6. Results for type 3 waterdrops (strong waterdrops).

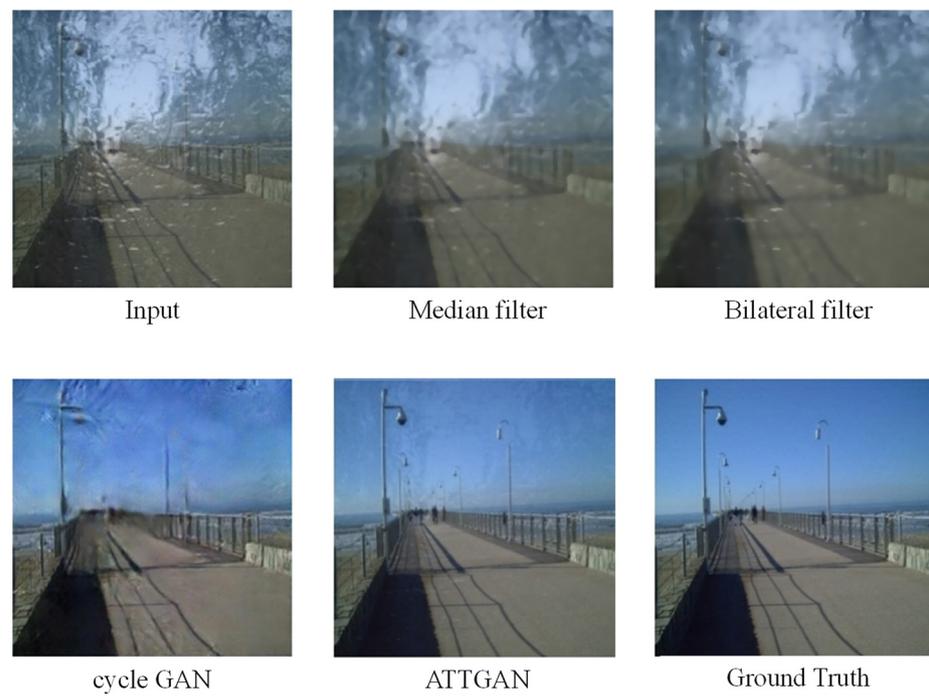


Figure 7. Results for type 4 waterdrops (strong water stream).

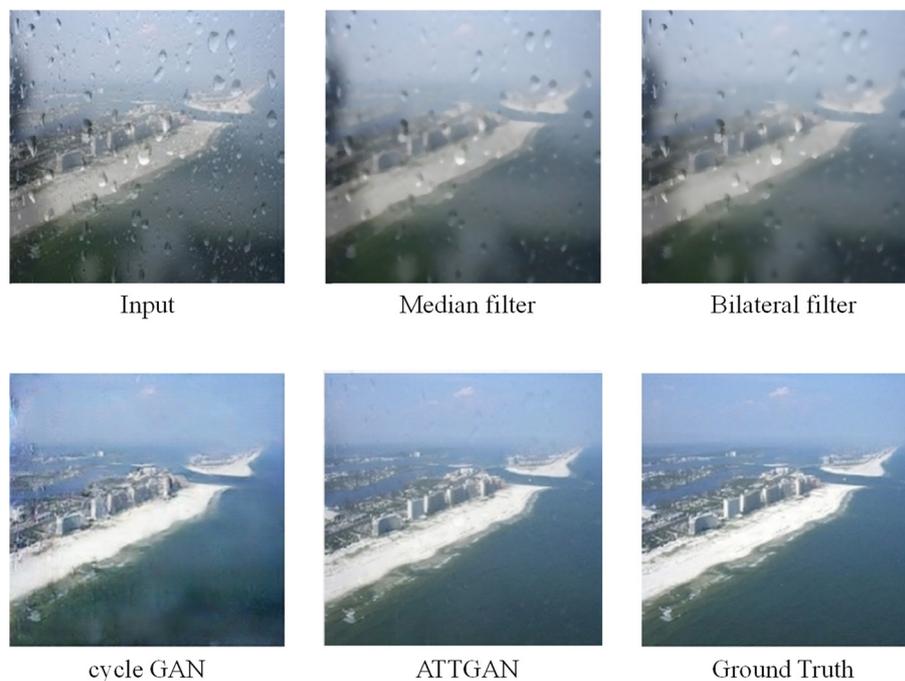


Figure 8. Results for type 5 waterdrops (large waterdrops and strong water fog).

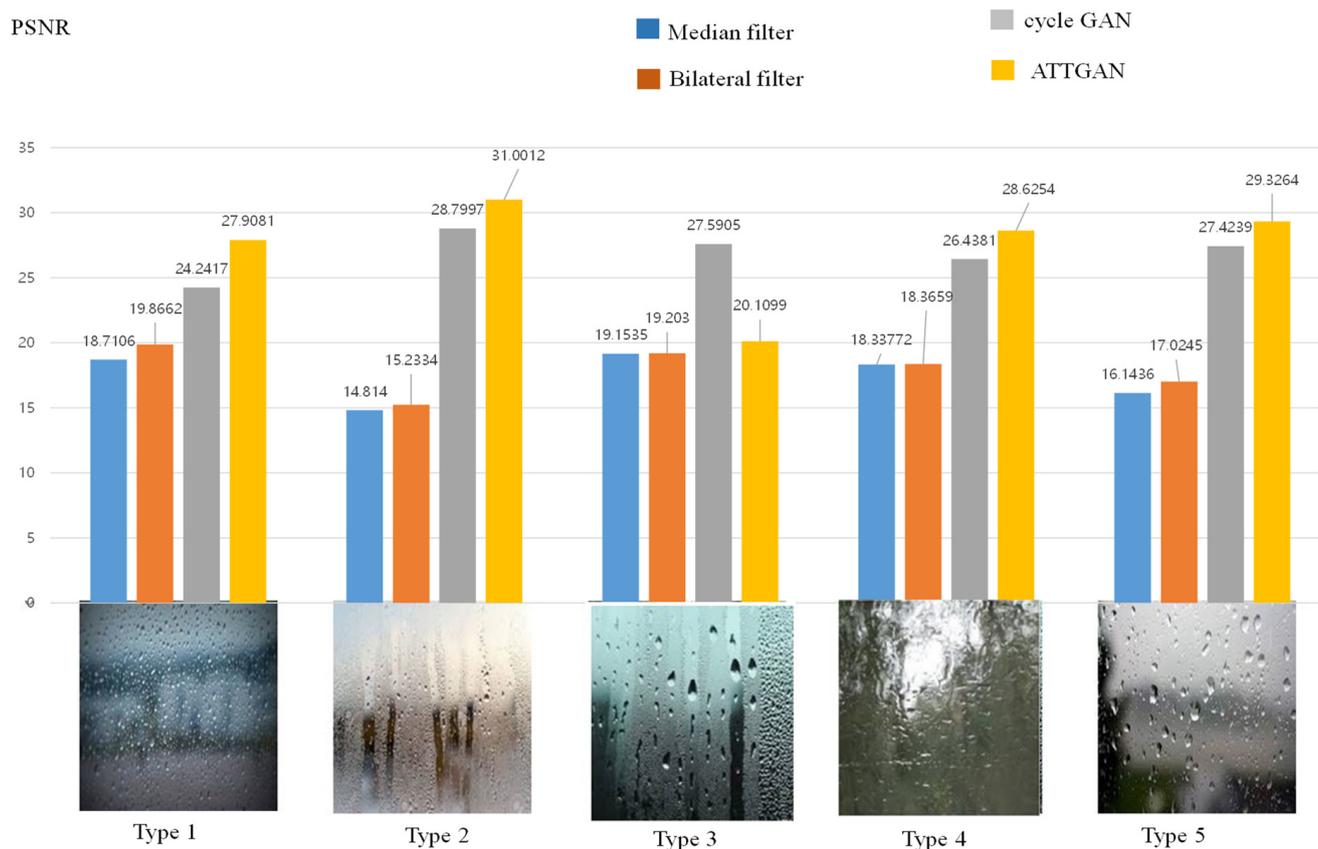


Figure 9. PSNRs according to the waterdrop types.

The proposed method had a better effect on the removal of both large and small water droplets with different shapes by changing the attentive map. On the other hand, the

modified attributes were not prominent, although the raindrops were well preserved. This degraded the performance of the system.

5. Conclusions

We proposed a single-image-based raindrop-removal method. The method utilizes a generative adversarial network, where the generative network produces an attention map via an attentive-recurrent network and applies this map along with the input image to generate a raindrop-free image through a contextual autoencoder. Our discriminative network then assesses the validity of the generated output both globally and locally. For local validation, we inject the attention map into the network. The novelty lies in the use of the attention map in both the generative and discriminative network. Our experiments demonstrated that the proposed method could effectively remove various water drops.

Author Contributions: Conceptualization, H.-D.Y.; Methodology, S.G. and H.-D.Y.; Software, S.G. and H.-D.Y.; Validation, S.G. and H.-D.Y.; Data curation, S.G. and H.-D.Y.; Supervision, H.-D.Y.; Project administration, H.-D.Y.; writing—original draft preparation, S.G. and H.-D.Y.; writing—review and editing, H.-D.Y.; funding acquisition, H.-D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant from the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) (No. NRF-2017R1A2B4005305, No. NRF-2019R1A4A1029769).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This paper was modified and developed from the master's thesis of the first author [27].

Conflicts of Interest: The author declares no conflict of interest.

References

1. Qian, R.; Tan, R.T.; Yang, W.; Su, J.; Liu, J. Attentive generative adversarial network for raindrop removal from a single image. In Proceedings of the IEEE International Conference on Computer Vision, Salt Lake City, UT, USA, 23 June 2018; pp. 2482–2491.
2. Li, X.; Liu, Z.; Li, B.; Feng, X.; Liu, X.; Zhou, D. A novel attentive generative adversarial network for waterdrop detection and removal of rubber conveyor belt image. *Math. Probl. Eng.* **2020**, *2020*, 1–20. [[CrossRef](#)]
3. Wang, H.; Wu, Y.; Li, M.; Zhao, Q.; Meng, D. A survey on rain removal from video and single image. *arXiv* **2019**, arXiv:1909.08326.
4. Yang, W.; Tan, R.T.; Wang, S.; Fang, Y.; Liu, J. Single image deraining: From model-based to data-driven and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, 1–18. [[CrossRef](#)]
5. Li, S.; Ren, W.; Wang, F.; Araujo, I.B.; Tokuda, E.K.; Junior, R.H.; Cesar, J.R., Jr.; Wang, Z.; Cao, X. A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives. *Int. J. Comput. Vision* **2021**, *129*, 1301–1322. [[CrossRef](#)]
6. Garg, K.; Nayar, S.K. Detection and removal of rain from videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 July 2004; pp. 526–535.
7. Zhang, X.; Li, H.; Qi, Y.; Leow, W.K.; Ng, T.K. Rain removal in video by combining temporal and chromatic properties. In Proceedings of the IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 9–12 July 2006; pp. 461–464.
8. Barnum, P.; Kanade, T.; Narasimhan, S. Spatio-temporal frequency analysis for removing rain and snow from videos. In Proceedings of the International Workshop on Photometric Analysis for Computer Vision, Rio de Janeiro, Brazil, 14 October 2007; pp. 1–8.
9. Chen, Y.-L.; Hsu, C.-T. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1968–1975.
10. Hu, X.; Fu, C.W.; Zhu, L.; Heng, P.A. Depth-attentional features for single-image rain removal. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, California, CA, USA, 15–20 June 2019; pp. 8022–8031.
11. Zhu, L.; Fu, C.; Lischinski, D.; Heng, P. Joint bilayer optimization for single-image rain streak removal. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2545–2553.
12. Deng, L.J.; Huang, T.Z.; Zhao, X.L.; Jiang, T.X. A directional global sparse model for single image rain removal. *Appl. Math. Model.* **2018**, *59*, 662–679. [[CrossRef](#)]

13. Li, Y.; Tan, R.T.; Guo, X.; Liu, J.; Brown, M.S. Rain streak removal using layer priors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 30 June 2016; pp. 2736–2744.
14. Yang, W.; Tan, R.T.; Feng, J.; Liu, J.; Guo, Z.; Yan, S. Deep joint rain detection and removal from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 21–26 July 2017; pp. 1685–1694.
15. Yang, W.; Tan, R.T.; Feng, J.; Guo, Z.; Yan, S.; Liu, J. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 1377–1393. [[CrossRef](#)] [[PubMed](#)]
16. Yang, W.; Liu, J.; Feng, J. Frame-consistent recurrent video deraining with dual-level flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, California, CA, USA, 15–20 June 2019; pp. 1661–1670.
17. Fu, X.; Huang, J.; Ding, X.; Liao, Y.; Paisley, J. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Trans. Image Process.* **2017**, *26*, 2944–2956. [[CrossRef](#)] [[PubMed](#)]
18. Lee, K.H.; Ryu, E.; Kim, J.O. Progressive rain removal via a recurrent convolutional network for real rain videos. *IEEE Access* **2020**, *8*, 203134–203145. [[CrossRef](#)]
19. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [[CrossRef](#)]
20. Chen, L.; Lu, X.; Zhang, J.; Chu, X.; Chen, C. HINet: Half instance normalization network for image restoration. *arXiv* **2021**, arXiv:2105.06086.
21. Wang, Z.; Cun, X.; Bao, J.; Liu, J. Uformer: A general U-shaped transformer for image restoration. *arXiv* **2021**, arXiv:2106.03106.
22. Choi, Y.; Choi, M.; Kim, M.; Ha, J.W.; Kim, S.; Choo, J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8789–8797.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Xingjian, S.H.I.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015; pp. 802–810.
25. ImageNet. Available online: <http://image-net.org/> (accessed on 3 May 2021).
26. Qian, R.; Tan, R.T.; Yang, W.; Su, J.; Liu, J. rui1996/DeRaindrop. Available online: <https://github.com/rui1996/DeRaindrop> (accessed on 20 May 2021).
27. Goo, S. Restoring Water Drop on Window using on Conditional Generative Adversarial Network. Master’s Thesis, Department of Computer Engineering, Chosun University, Gwangju, Republic of Korea, 2018.