IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 17, 2020, accepted October 16, 2020, date of publication November 9, 2020,
date of current version November 19, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3036680*

# Progressive Rain Removal via a Recurrent Convolutional Network for Real Rain Videos

**KYU-HO LEE, EUNJI RYU, AND JONG-OK KIM, (Member, IEEE)**
School of Electrical Engineering, Korea University, Seoul 02841, South Korea

Corresponding author: Jong-Ok Kim (jokim@korea.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF), Ministry of Science and Information and Communication Technologies (ICT) (MSIT), South Korea, funded by the Korea Government, under Grant 2019R1A2C1005834, and in part by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) supervised by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2020-0-01749-001.

**ABSTRACT** Rain removal in videos is a problem that has attracted tremendous interest of researchers within the field of deep learning. Although deep-learning-based rain removal methods outperform large number of conventional vision methods, some technical issues that need to be resolved remain. In this article, we propose a new deep learning method for video rain removal based on recurrent neural network (RNN) architecture. Pseudo groundtruth was generated from real rainy video sequence by temporal filtering for supervised learning. Instead of focusing on various shapes of rain streaks similar to conventional methods, in this article, we focused on the changing behaviors of rain streaks. To accomplish this, images of progressive rain streaks were generated from the real rain videos and are sequentially fed to the network in a decreasing rain order. Multiple images with different amounts of rain streaks were used as RNN inputs to more efficiently identify rain streaks and then remove them. Experimental results demonstrate that our method is suitable for a wide range of rainy images. Moreover, experiments performed on both real-world and synthetic images demonstrate that our proposed method can achieve competitive results in comparison with the benchmarked and conventional approaches for rain streak removal from images.

**INDEX TERMS** Progressive rain removal, real rain dataset, video rain removal, image restoration, recurrent convolutional network.

## I. INTRODUCTION

Rain causes a series of visibility degradation and alters the content and color of digital images. Raindrops can not only annoy or confuse a human viewer but also degrade the performance of any computer vision algorithm that depends on small features. When raindrops are close to the camera lens, each drop causes reflection highlight, scattering, and blur, thus significantly reducing the visibility of rainy images.

Although the removal of rain, haze and snow from images has been extensively studied for years [1]–[9], [29], [47]–[57], it remains challenging because it is an inherently ill-posed problem. Several non-deep-learning-based rain removal methods, such as frequency domain representation [10], Gaussian mixture model [11], and sparse representation [12], have been proposed and demonstrated to lead to significant quality improvements. Owing to increasing interest in deep learning recently, few deep-learning-based rain removal methods have been proposed [1]–[5], [13], [14], [13], [27]–[31], [29], [47]–[54]. Deep networks allow us to easily learn the correlation between rain streaks and background and typically achieve better performance than non-deep-learning approaches.

As a pair of clean and rainy images for the exact same scene and background is difficult to obtain, general learning-based methods commonly resort to synthetic dataset creation. These pairs of rainy and clear images are fed as inputs and labels in a supervised training of a CNN. However, as groundtruth image is not available in the case of actual rainfall image sequences, training a neural network in a supervised manner is difficult. The network probably learns with the pseudo groundtruth on the right in Figure 1, which is obtained from multiple rainy images via temporal filtering. A derained image close to the actual groundtruth can be generated by temporarily filtering multiple images. However, this pseudo groundtruth still suffers from the residues of rain, such as faint white

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Moinul Hossain.

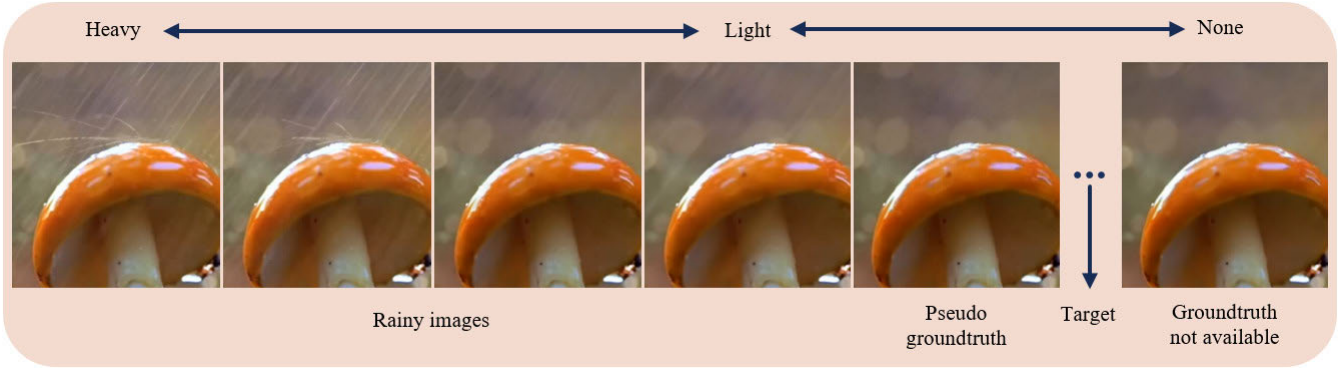


**FIGURE 1.** Example of progressive rain images (including pseudo groundtruth) whose rain streaks gradually decrease from heavy to light. Under the assumption that the difference between neighboring rainy images is only rain streaks, we attempted to learn the rain behaviors between progressive rain images, leading to successful training without groundtruth. Although the proposed network is trained with pseudo groundtruth, it can achieve a clearer image (located in the target of the figure), rather than the pseudo groundtruth. Note that the pseudo groundtruth image is generated by temporal filtering with multiple rain images.

dots. Thus, imperfect groundtruth cannot guarantee excellent network output. This motivates us to learn rain behaviors with a sequence of progressive rain images, instead of a single or multiple raw rain images. Figure 1 shows a sequence of rainy images where the amount of rain streaks progressively decreases from left to right.

Most conventional methods use a well-known rain model to generate rainy images with various amounts of rain and background components. However, these synthetic raindrops and streaks appear different in terms of shapes, sizes, and directions, compared to real raindrops. It is hard to perform supervised learning for real rain due to the unavailable clean version of a rainy image. Therefore, in order to overcome this challenge, a training dataset with pairs of pseudo-groundtruth and rainy images was generated using real rainy images. Also, rather than synthetic raindrops, we focused on their changing behaviors.

In this article, we propose a new deep learning method for video rain removal. Rather than feeding a network with synthetic pairs of clean and rainy datasets, real-world rainy images are used to train it in a supervised manner. The following are the key contributions of the study. First, we create training and evaluation datasets based on real rain videos. Several progressive rainy images are generated using temporal filtering, and their rain streaks are gradually decreased. The resulting image with the least raindrops is regarded as the pseudo groundtruth, and the other images are fed into a network as an input. For a real rainy image, its clean version cannot be almost obtained. Thus, rather than using a pair of clean and rainy images, we utilized many gradually rain decreasing images (hereinafter referred to as progressive rain images), and we attempted to learn the decreasing behaviors of raindrops. Second, we adopted a recurrent neural network (RNN) that handles sequential inputs and/or outputs as a base architecture. We observed that real rain video scenes have varying rain streaks with different shapes, sizes, and directions. Most conventional methods adopted the approach of extracting rain streaks themselves and subsequently removing them. However, the shape and size of rain streaks vary, and, thus, accurately removing them is difficult. Rather than concentrating on the raindrops, we focused on their changing behaviors. Our progressive rain images have the strongest correlation of rain streaks among them owing to the lighter rainy inputs being a subset of the heavier ones in terms of rain streaks only. Multiple rainy images with different rain streaks are used as the RNN inputs to more efficiently identify rain streaks.

The rest of this article is organized as follows. Section II introduces related works on image and/or video rain removal. Section III describes the generation of real rain datasets from rainy videos. Section IV describes our RNN-based rain removal architecture. Section V verifies the effectiveness of our proposed rain removal method visually and quantitatively. Finally, Section VI provides the concluding statements.

## II. RELATED WORKS

In this section, we briefly review the existing image and video rain removal algorithms.

### A. SINGLE IMAGE RAIN REMOVAL

For the removal of a single image rain streak, Kang *et al.* [10] proposed a method that decomposes an input image into its low-frequency component as a structure layer and its high-frequency component as a texture layer. Their method attempts to separate rain streak frequencies from the high-frequency layer via sparse coding-based dictionary learning with HoG features. Although the decomposition concept is refined, its results tend to blur the background. In a similar manner, several methods address the problem by separating the signals into a rain layer and a structure layer, based on texture appearances. Kim *et al.* first detected rain streaks that are removed with the nonlocal mean filter [20]. Li *et al.* exploited the Gaussian mixture model to extract rain streaks. Deep learning promoted the development of single-image deraining [11]. Further, Luo *et al.* proposed a discriminative

sparse coding method to separate rain streaks from the background [12].

Owing to the success of convolutional neural network (CNN) in classification, segmentation, and recognition [22]–[26], [36]–[40], CNN has been applied to rain removal. Deep networks in [13], [14] are adopted as a basic model to separate the structure and texture layers from rainy images. Fu *et al.* proposed a deep network that uses the image detail layer as its input and predicts the negative residues as rain streaks [13]. The network has a good capacity to retain the texture details; however, it cannot handle cases of heavy rain where the rain streaks are dense. Yang *et al.* proposed a deep learning-based joint rain detection and removal to recurrently remove rain streaks and accumulations [14], [52], obtaining impressive results in heavy rain cases. However, the rain streaks and textures of the background are intrinsically overlapped in the feature space.

### B. VIDEO RAIN REMOVAL

For the image restoration of rain videos, earliest methods use a temporal median filter for each pixel [15], [16]. Temporal median filtering exploits the fact that, in all but the heaviest storms, each pixel is more often clear than corrupted. Garg *et al.* proposed a method that can control a video camera's operational parameters when capturing a rainy scene [17]. They suggested using temporal and spatial blurring either by increasing the exposure time or reducing the depth of field. Thus, adjusting these camera parameters while capturing the video will reduce the appearance of rain streaks. These early attempts primarily rely on the linear space–time correlation of raindrops and, thus, fail when the rain streaks are diversified in terms of its scales and densities. Subsequent studies formulate rain streaks with more flexible and intrinsic models. Zhang *et al.* added an additional constraint called the chromaticity constraint [7]. They observed that intensity changes in RGB color channels are identical to pixels that represent rain streaks. This method is an improvement over simple median filtering. Recently, several methods further utilized temporal dynamics, including continuity of background motions, explicit motion modeling, and random appearance of rains between frames, to facilitate video rain removal [18]–[21]. Chen and Chen proposed embedding motion segmentation by using a Gaussian mixture model into rain detection and removal [19]. Tripathi *et al.* trained a Bayes rain detector based on spatial and temporal features [41]. Kim *et al.* trained an SVM to refine the roughly detected rain maps [20]. Further, Wei *et al.* encoded rain streaks as patch-based mixtures of the Gaussian model that can finely adapt a wider range of rain variations [42].

## III. DATASET GENERATION

The synthesized rain datasets widely used in several supervised rain removal methods have different characteristics than actual rain. The length and thickness of rain streaks in the synthesized image are not similar to real rain. Therefore, synthesized datasets do possibly not reflect the actual rain characteristics to CNN hyper-parameter learning. Therefore, in this article, we constructed a real rain dataset based on the following observations.

We observed that the amount and characteristics of rain streaks differ across real rain images, which vary on weather conditions at the time when the image was captured. For example, under heavy rain conditions, rain streaks closer to camera lens are thicker and long, and those farther away from the camera are thinner and shorter. Moreover, if the distance of the rain streaks from the camera is farther away, rain is captured similar to fog. Another observation is that, in most cases, the colors of background objects become brighter due to rain streaks. The light reflected by raindrops is mixed with the light of the background objects such that the background is captured brighter than the original color. Our final observation is that raindrops are random and fast regardless of the amount of rain; this implies that a clean background without rain can be captured at least several times in several video frames. These observations indicate that simply generating pseudo groundtruth images from real rain image sequences using temporal filtering is possible.

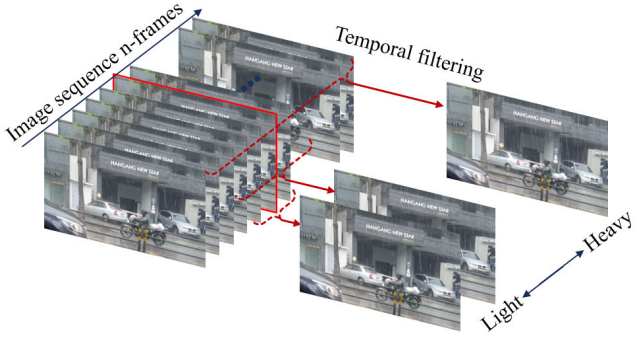


**FIGURE 2. Overview of our dataset generation with the actual rain image sequence. We generated several progressive rain images to include different amounts of rain by controlling the window length for temporal filtering.**

Figure 2 shows the overview of our dataset generation process from a real rain video. We created a real rain dataset using the following steps. The pixels on the temporal axis in an image sequence were grouped as a vector. The elements (temporal samples) of each vector can be one among rain, background, and their mixture. Based on our second observation, we adopted a minimum operation to effectively obtain a rain-free pixel, which was used as the pseudo groundtruth.

Subsequently, we obtained progressive rain images, wherein the rain streaks gradually increase. Based on the second observation mentioned previously, all rain streaks in an image sequence can be extracted by the maximum operation. In other words, a maximum pixel in each group of temporal pixels is selected as a rain pixel. The amount of rain can be easily controlled by the number of pixels to be temporarily filtered; that is, the wider the temporal filtering window, the higher the probability that a rain pixel will be selected (Figure 2). By adjusting the window length of temporal filtering, we can generate few progressive rain

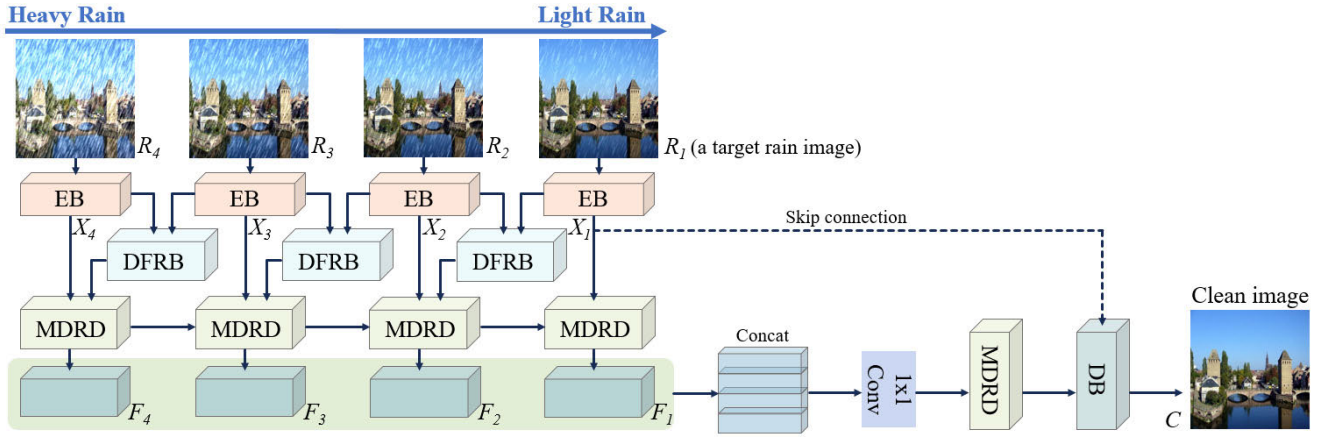


**FIGURE 3.** Network structure of the proposed rain removal method.

images ranging from light to heavy (Figure 1). Further, our RNN network takes four different progressive rain images as inputs (Figure 3).

## IV. PROPOSED METHOD

The key concept of the proposed network is that the rain streaks are progressively removed by feeding rainy inputs in a decreasing rain order. In this section, we describe our method of removing rain streaks through the RNN structure, taking advantage of the strong correlation between input images.

### A. RNN ARCHITECTURE

Recurrent neural networks (RNNs) handle sequential inputs and/or outputs. RNN architectures that are widely used: many-to-one and many-to-many. Figure 3 shows the overview of our proposed network structure. A common RNN architecture involves using a series of images as inputs that become progressive rain streak images (described in the previous section), including a target one herein.

Given four input images $R_i$, $(i = 1, 2, 3, 4)$ corresponding to progressive four rain images, they first go through an encoding block to extract feature maps $X_i$, $(i = 1, 2, 3, 4)$. For clarity, $i = 1$ in the notations $R_i$ and $X_i$ indicates a target rain image and its feature map, respectively. The input sequence of rain frames is constructed as $\{R_1, R_2, R_3, R_4\}$, where a higher $n$ represents a heavier rain image.

Our proposed RNN architecture comprises four blocks: initial feature extraction (encoding block), difference residual block, multi-dilated residual dense block, and reconstruction (decoding block). They are sequentially described in the remainder of this section.

### B. ENCODING BLOCK

Image information is extracted from each input using an encoder block. As we used various numbers of rain images as inputs, the extracted feature tensors contain different detail information for the same background scene. This sufficient detail information can be useful when reconstructing the final output. During the extraction of the feature of each input, a nonlocal block is added to the encoding module to extract both local and global information simultaneously. Local information is easily extracted by a general convolution layer. For the extraction of global details, a nonlocal block is used instead of a convolution layer.

### C. DIFFERENCE RESIDUAL BLOCK

Here, we aimed to progressively remove rain streaks by sequentially feeding from heavy to light rain images into RNN. Several conventional methods adopted an approach that extracts rain streaks and subsequently remove them. However, the shape and size of the rain streaks vary, and accurately removing them is therefore difficult. In this study, we focused on the decreasing behaviors of rain streaks in few progressive rain images. Therefore, we added the difference residual block (DFRB), as shown in Figure 4 (a). DFRB is composed of three ResBlocks and two regular convolution layers. A current input is enhanced by combining it with the next input. The difference between the two input features represents the decreased rain streaks in the image sequence, and it helps efficiently train various shapes of rain streaks.

### D. MULTI-DILATED RESIDUAL DENSE BLOCK

The multi-dilated residual dense block (MDRD) is designed for extracting global features, as illustrated in Figure 4 (b). The MDRD comprises eight dilated convolution layers, seven regular convolution layers and two $1 \times 1$ convolution layers. To obtain more global details, we increased the receptive field via dilated convolutions with increasing dilation factors. Each regular convolution layer uses the concatenation of two neighbor dilated convolution feature maps from previous layers as an input to create a richer nonlinear representation. Thus, our network is developed as a cascade of four multi-dilated residual dense blocks; each block uses the previous MDRD output feature as an input. The dilation factors

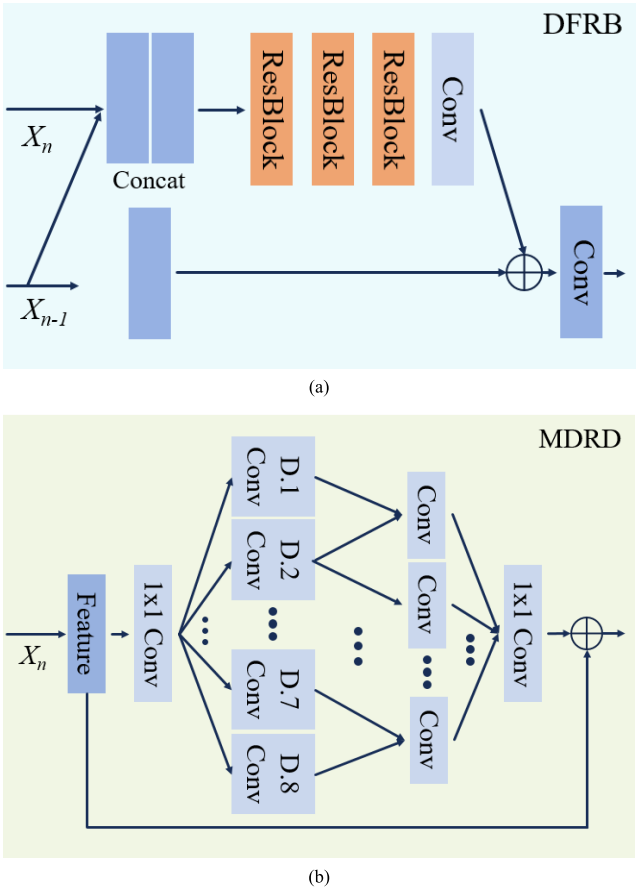


**FIGURE 4.** (a) Difference residual block (DFRB) structure and (b) multi-dilated residual dense block (MDRD) structure. D.n (n = 1–8) in (b) indicates dilated convolution layers.

increase from one to the next, and all regular and dilated convolutions are followed by ReLU nonlinearity [24].

### E. DECODER BLOCK

The reconstruction of the output derained image was performed by decoding the auto encoder. To propagate finer details to the output, we incorporated additional ResNet-style [25] skip connections between the input and the output of each block. The skip connection is a simple concatenation of the input feature blocks to the output ones.

### F. LOSS FUNCTION

Our objective is to optimize network parameters that minimize the following loss function:

$$L = \mu_1 L_{L1} + \mu_2 R_{TV}, \tag{1}$$

where $\mu_1$ and $\mu_2$ are the coefficients that are empirically determined. The first term, $L_{L1}$, is a $L_1$ loss between the derained image $C$ and pseudo groundtruth. The second term $R_{TV}$ is a total variation (TV) regularizer [43]. As mentioned in Section III, rain streaks mostly have higher pixel values than the surrounding pixels; thus, TV regularization is applied to smoothen the rain streak pulse. The TV loss is given by

$$R_{TV}(x) = \sum_{i,j} ((x_{i,j+1} - x_{ij})^2 + (x_{i+1,j} - x_{ij})^2)^{\frac{\beta}{2}}, \tag{2}$$

where $x$ is a vectorized and mean-subtracted image and $i$ and $j$ denote the location of the pixels and $\beta = 2$.

## V. EXPERIMENTAL RESULTS

### A. IMPLEMENTATION DETAILS

Our network was trained using real rain dataset pairs of 66,432 images, with a fixed resolution of 768 × 512. It is implemented using the PyTorch framework on a PC with 2 NVIDIA RTX 2080ti GPUs [36]. For loss optimization, we adopted the Adam optimizer with a batch size of 32 [37]. The initial learning rate is 0.0001 and is divided by 10 for every 20K iterations.

### B. TRAINING DATASET

As real rainy images and the corresponding sunny images are not simultaneously available, we created a real rain dataset using our own steps, as discussed in Section III. We captured several rainy videos on a rainy day as well as searched for many rainy videos on YouTube. We generated progressive rain inputs and pseudo groundtruth via temporal filtering. Our real rain dataset includes 66,432 frames for 120 different scenes. During a training phase, they are randomly cropped into 64 × 64 × 3 patches.

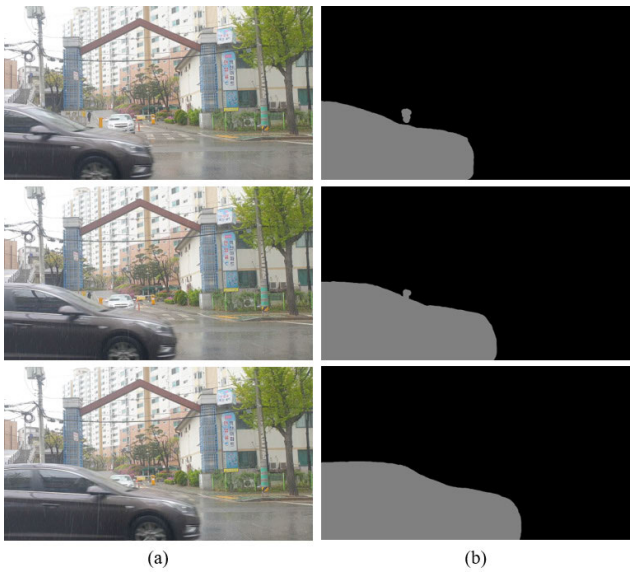


**FIGURE 5.** (a) Example of rainy image sequence, including a moving object and (b) its object mask maps.

To generate progressive four inputs in Figure 2, temporal filtering window (image sequence lengths) was set to 1, 3, 5, and 7. If the image sequence contains moving objects, ghosting artifacts frequently occur in video deraining. The ghost artifact is one of the obstacles to accurately derain, and preprocessing is necessary to prevent artifacts. Prior to


**FIGURE 6.** Rain removal results of real rain images include a moving object. Upper images are n-1$^{th}$ frames in image sequence, lowers are n$^{th}$ frames: (a) rainy images, (b) results using the method in [31], and (c) results using our proposed method.

**TABLE 1.** Quantitative average results on 20 synthetic DID-MDN dataset images. Red indicates the best result, and blue indicates the second best

| | Rainy | Ground Truth | Method [c] | Method [d] | Method [e] | Method [f] | Method [g] | Method [h] | Method [i] | Method [j] | Method [k] | Ours |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PSNR** | 20.77 | Inf | 29.23 | 30.40 | 23.65 | 34.38 | 25.82 | 26.85 | 29.31 | 25.20 | 29.40 | 34.99 |
| **SSIM** | 0.4794 | 1 | 0.6972 | 0.7232 | 0.5850 | 0.8000 | 0.5788 | 0.5897 | 0.6985 | 0.6395 | 0.6462 | 0.8296 |
| **UQI** | 0.4822 | 1 | 0.7156 | 0.7584 | 0.6128 | 0.8247 | 0.5866 | 0.6103 | 0.7129 | 0.6487 | 0.6644 | 0.8591 |
| **VIF** | 0.2583 | 1 | 0.5535 | 0.6659 | 0.5092 | 0.7589 | 0.4317 | 0.4474 | 0.6146 | 0.4907 | 0.5906 | 0.7949 |

the generation of the input images, we first searched for image regions with motion, and an optical flow algorithm was used to generate the moving object mask. Figure 5 shows an example of the image sequence and its moving object masks. Temporal filtering for input images was performed only on static regions, except for the moving region specified in the moving object mask.

### C. REAL-WORLD IMAGE RESULTS

Conventional methods are also evaluated on real-world images. Figure 6 shows the qualitative results on our real dataset that includes a moving object. As also shown in the figure, the method [31] still suffers from artifacts after deraining. In particular, in the moving object of a car, some rain can be removed; however, a considerable number of artifacts are produced. Our proposed method provides better visual performance by effectively removing rain streaks and preserving details.

Figure 7 shows another qualitative result on real-world images. For another method [28], color distortion happens in the deraining results, whereas the methods in [13], [29], [30] cannot thoroughly remove rain streaks. Another method [28] tends to remove image details such as leaves on the tree as well as rain streaks, as shown in Figure 7 (e1). The raindrops on the ground remain in the methods used in [31], [34]. In the methods used in [27], [13], [29], [30], we can observe artifacts near the edge region. Conversely, our method evidently removes rain streaks better and improves the visual quality.

### D. SYNTHESIS IMAGE RESULTS

We also quantitatively and qualitatively compared performances on synthetic images. Four quality measures were used for quantitative evaluation: peak signal-to-noise ratio, structural similar index [44], universal quality index (UQI) [45], and visual information fidelity [46]. Table 1 presents the quantitative results on 12 synthetic images. The proposed method achieves competitive performance compared to the conventional methods. In particular, our method achieves the highest UQI values. For other measures, we also quantitatively achieved the best and the highest scores.

Figure 8 also visually demonstrates the effectiveness of our proposed method for a synthetic rain sequence. The derained images of the methods used in [29], [30], [32], [34] contain rain artifacts and streaks, but some rain streaks remain, especially for heavy rain, as shown in Figure 8 (c1–c2, f1–f2, g1–g2, j1–j2). The methods used in [21], [31] successfully remove bright and thick rain streaks; however, thin and low-intensity rain streaks remain, as shown in Figure 8 (i1–i2, k1–k2). The method used in [28] tends to remove even some image details, and rain artifacts remain and the color is also degraded, as observed in Figure 8 (e1–e2). They may be caused by the insufficient learning ability of parameter layers.

The method used in [27] achieves good deraining results, but some thick and spotted rain streaks cannot be removed in the background, as shown in Figure 8 (d1–d2). The method used in [13] tends to blur image details, as shown in Figure 8


**FIGURE 7.** Rain removal results on real-world images: (a1–a2) rainy images, (b1–b2) pseudo groundtruth, (c1–c2) results using the method in [29], (d1–d2) results using the method in [27], (e1–e2) results using the method in [28], (f1–f2) results using the method in [30], (g1–g2) results using the method in [33], (h1–h2) results using the method in [13], (i1–i2) results using the method in [21], (j1–j2) results using the method in [32], (k1–k2) results using the method in [31], and (l1–l2) results using our proposed method.

(h1, h2), as the rain-density levels of input rainy images are difficult to determine. In contrast to the nine methods, our proposed method is capable of removing more rain streaks while preserving image details.

### E. PROGRESSIVE RAIN REMOVAL

Our progressive rain images have a strong correlation of rain streaks between neighboring input images. It is expected that the correlation contributes to better removal of rain streaks.

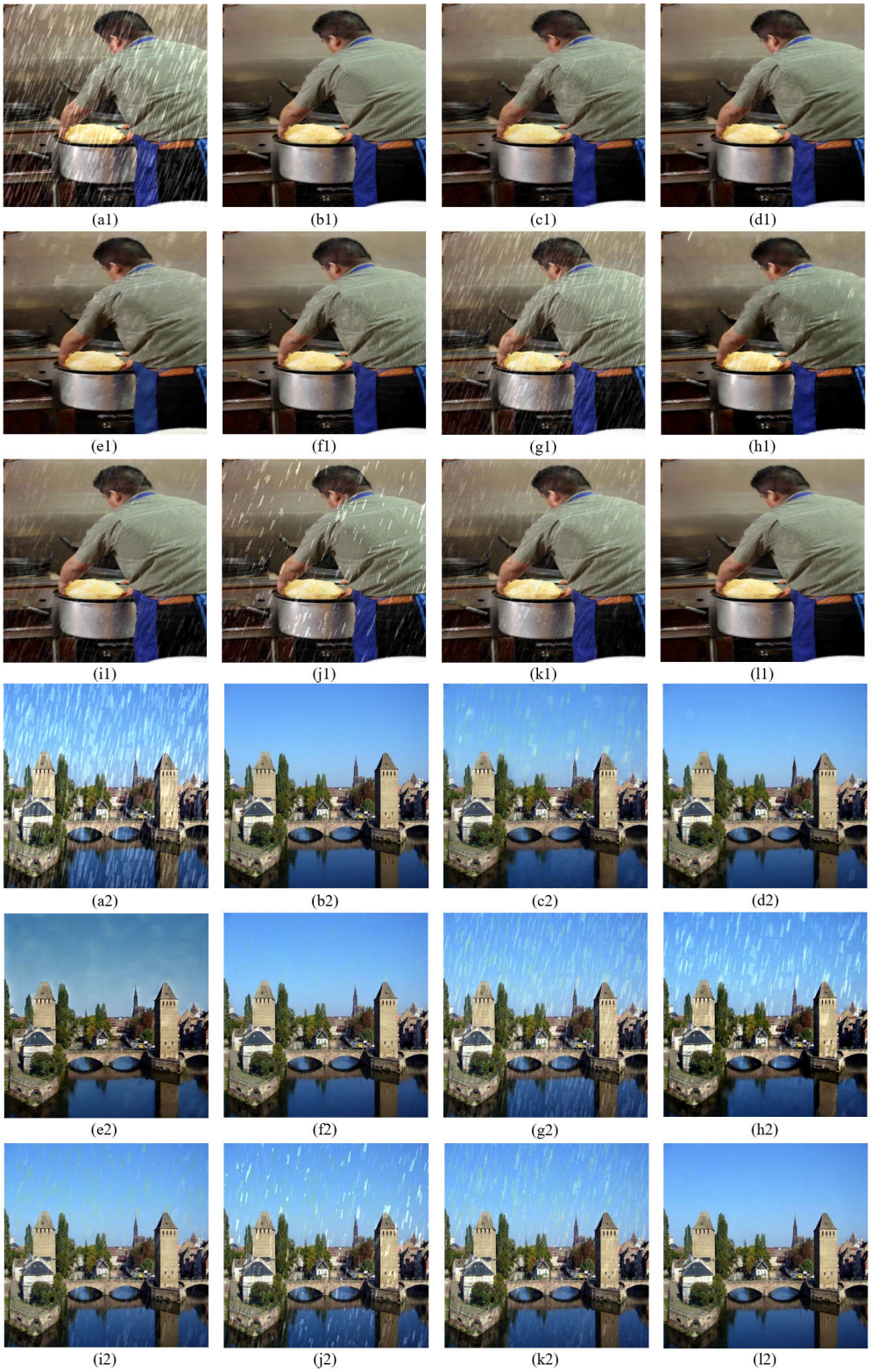


**FIGURE 8.** Rain removal results on synthetic images: (a1–a2) rainy images, (b1–b2) groundtruth, (c1–c2) results using the method in [29], (d1–d2) results using the method in [27], (e1–e2) results using the method in [28], (f1–f2) results using the method in [30], (g1–g2) results using the method in [33], (h1–h2) results using the method in [13], (i1–i2) results using the method in [21], (j1–j2) results using the method in [32], (k1–k2) results using the method in [31], and (l1–l2) results using our proposed method.

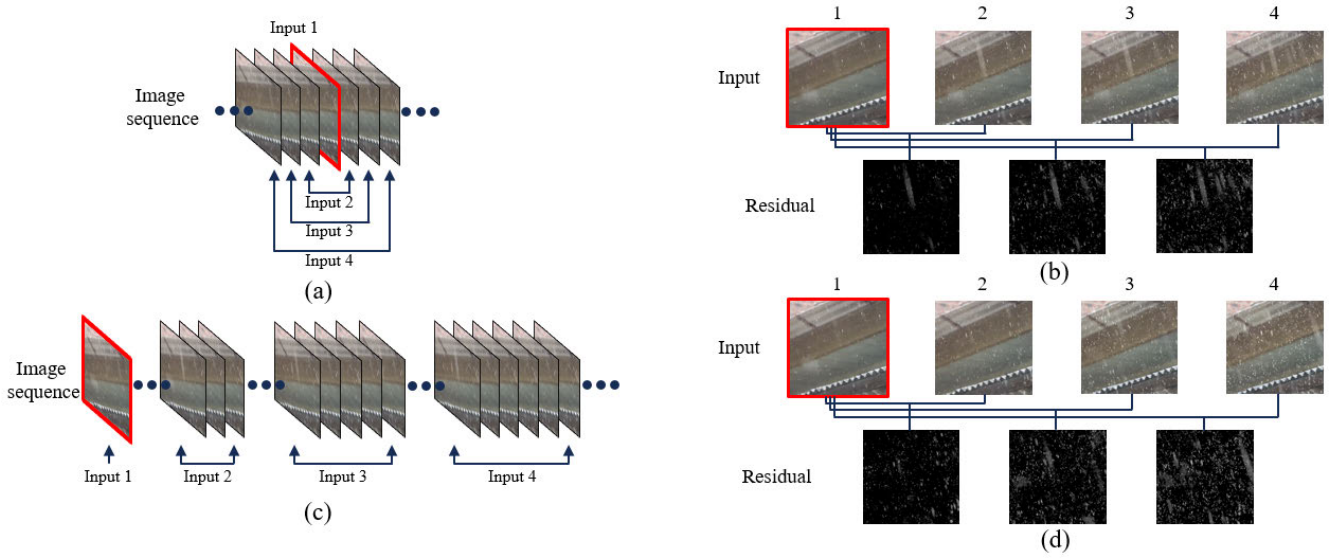


**FIGURE 9.** Comparison of the two different ways of generating progressive inputs. (a) is the method of generating rain correlated inputs in the proposed method; (c) generates rain non-correlated inputs. The red solid box indicates a target frame. (b) and (d) show the residuals between the input images for the methods (a) and (c), respectively, for correlation comparison.

This is easily confirmed by comparing it with the result of uncorrelated progressive rain inputs.

Figures 9 (a) and (c) illustrate the generations of two types of input images. Figure 9 (a) shows the generation of the proposed method, as already shown in Section III. Meanwhile, Figure 9 (c) shows the generation of uncorrelated progressive inputs, indicating ''no-correlation of rain streaks'' between neighboring inputs. In the case of (a), the temporal filtering window of ''input $i$'' is a subset of ''input $i+1$''. This indicates that the rain streaks of ''input $i$'' probably belong to those of ''input $i+1$''. Conversely, in the case of (c), temporal filtering was performed on a separate window for the generation of four inputs so that there is no correlation between neighboring inputs. For this experiment, the rainy image sequence was synthetically generated by repeating a static image with random rain streaks.

Figures 9 (b) and (d) show the residuals between the input images of (a) and (c), respectively. In (b), the intensity of rain streaks progressively increases from light to heavy, and the lighter rainy inputs are a subset of the heavier rainy input, thereby increasing the correlation. Thus, the progressive reduction of rain streaks is learned, and this more effectively removes rain streaks. Meanwhile, in Figure 9 (d), rain streaks in each input image gradually increase, but the rain streaks in each image are independent and have a low correlation.

Subsequently, we evaluated the capability of the proposed RNN to gradually improve rain removal as rainy inputs increase. Figure 10 shows the decoding result of each subnet feature map, corresponding to $F_n$ ($n = 4, 3, 2, 1$) in Figure 3. In Figure 10, *dc* indicates decoded feature maps. For a visualization of feature maps, decoders were installed on each feature map. As shown in Figure 10, more inputs can lead to further enhance the performance of rain removal.

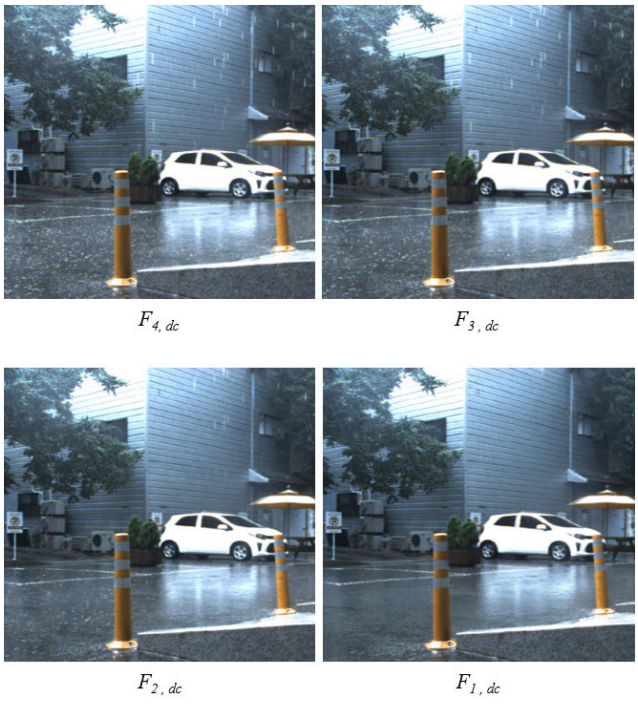


**FIGURE 10.** Example of the proposed RNN capability to progressively reduce rain streaks. For visualization, we attached decoders on feature maps ($F_4$ – $F_1$) to the bottom of Figure 3.

### F. ABLATION STUDIES

We investigated the architecture of our proposed method and validated the importance of individual components in the overall performance. Figure 11 shows the visual results of our proposed method and its baseline variants. Figure 11 (b) shows the derained result without the DFRB module in our proposed method. As can be observed, rain streaks have

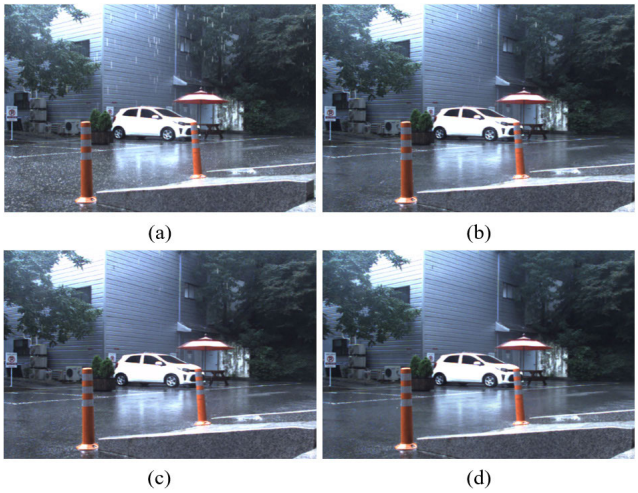


**FIGURE 11.** Visual results of the proposed method and its baseline variants: (a) rainy image, (b) without DFRB module, (c) use four input images with random picks in image sequence, and (d) the result of the proposed method with full architecture.

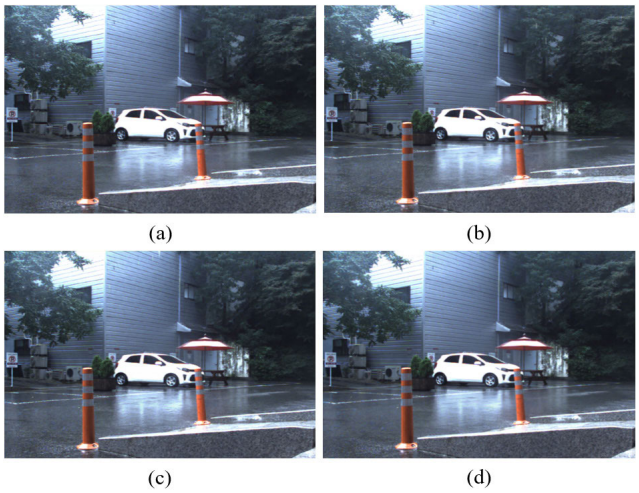


**FIGURE 12.** Visual results of the proposed method and its limited number of input images. (a)–(c) network architecture with 3-to-1 input, and (d) the result of the proposed method with full architecture.

been removed from many regions, including the floor in the image, but the rain streaks in the front of the building remain. Figure 11 (c) shows the result of using randomly selected four input images in the image sequence. As can be observed, the proposed RNN could not be effectively trained, and rain streaks could not be properly removed. Figures 12 (a–c) show the results of limiting the number of input images to 3, 2, and 1 in our overall network structure. The input images were sequentially removed from heavy input ($R_4$ in Figure 3) to light. Even if the number of input modules is reduced, it demonstrates good rain removal performance; however, compared to our proposed method with full architecture, many rain streaks remain.

Figure 13 visualizes some feature maps of $F_4$-to-$F_1$ in Figure 3. As shown in Figure 13 (a), the feature maps of $F_4$ extracted from the first input show the mixture between rain streaks and objects. The second input further enhances

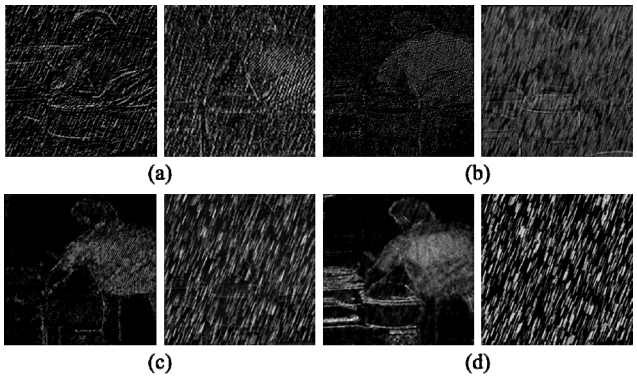


**FIGURE 13.** (a) ∼ (d) are the visualizations of some feature maps at $F_4 \sim F_1$ in Figure 3, respectively.

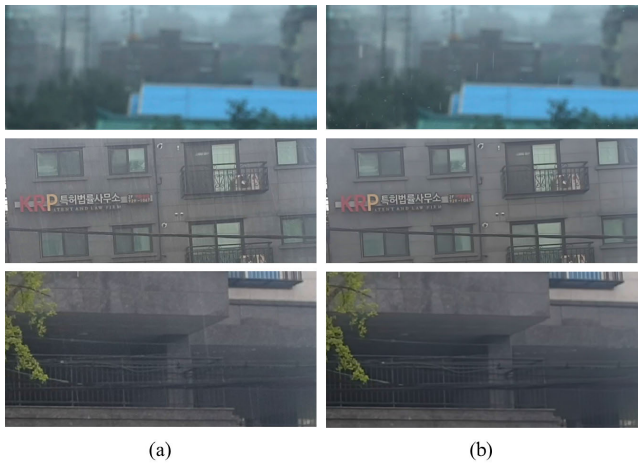


**FIGURE 14.** Visual quality of the proposed method to jointly remove rain streaks and hazy effects. (a) rainy images and (b) results of the proposed method.

the separation of rain streaks from objects as shown in Figure 13 (b). Thus, it can be confirmed from the feature maps at $F_4 \sim F_1$ that we could gradually separate rain streaks with progressive rainy inputs. Some of conventional methods remove rain streaks by separating a rain layer from a rainy image. However, the proposed method attempts to remove rain streaks by utilizing the changing behaviors of rain streaks between progressive rain images. This is the key difference, compared to conventional approaches.

## VI. CONCLUSION AND DISCUSSION

In this article, we proposed a new deep learning method to remove rain streaks from real rain videos. For supervised learning, we generated pseudo groundtruth from real rainy video sequences via temporal filtering. Several conventional methods attempted to extract rain streaks and subsequently remove them. However, while such methods require groundtruth for learning, it is practically not available in real rainy videos. This prevents a neural network from effectively learning rain removal. In this article, we focused on the learning of rain decreasing behaviors using RNN. Progressive rain streak images were obtained from a real rain video and were sequentially fed to the network in a decreasing

rain order. The experiments on both real-world and synthetic images demonstrate that our proposed method can handle rain removal in a supervised learning manner without groundtruth.

However, some limitations of the proposed method are existed. One of the limitations is that it is weak for fast camera moving scenes. Producing suitable progressive rain images would be difficult as moving regions are wider in the image. Another limitation of the proposed method is that it is not suitable for hazy effect caused by rains. The hazy effect occurs when raindrops are captured close to the camera as particles or very far away from the camera as mist. Figure 14 shows the rainy scenes with occurred hazy effects. Middle and bottom images of Figure 14 (a), small raindrop particles in close to the camera captured as hazy effect. These slight changes of small rain particles could capture in rain image sequence, so the hazy effect occurred by particles is reduced by the proposed method. However, the hazy effect such as mist existing in the far distance as shown in top of Figure 14 (a), cannot be removed by the proposed method because the mist far from the camera is not changed in the image sequence. These limitations would be addressed in a future study.

## REFERENCES

[1] Y. Luo, J. Zhu, J. Ling, and E. Wu, "Fast removal of rain streaks from a single image via a shape prior," *IEEE Access*, vol. 6, pp. 60069–60078, 2018.
[2] Z. Li, J. Zhang, Z. Fang, B. Huang, X. Jiang, Y. Gao, and J.-N. Hwang, "Single image snow removal via composition generative adversarial networks," *IEEE Access*, vol. 7, pp. 25016–25025, 2019.
[3] Y. Wang and T.-Z. Huang, "A tensor-based low-rank model for single-image rain streaks removal," *IEEE Access*, vol. 7, pp. 83437–83448, 2019.
[4] T. Matsui and M. Ikehara, "GAN-based rain noise removal from single-image considering rain composite models," *IEEE Access*, vol. 8, pp. 40892–40900, 2020.
[5] T. Yan, Y. Ding, F. Zhang, N. Xie, W. Liu, Z. Wu, and Y. Liu, "Snow removal from light field images," *IEEE Access*, vol. 7, pp. 164203–164215, 2019.
[6] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, USA, Jul. 2004, p. 1.
[7] X. Zhang, H. Li, Y. Qi, W. Leow, and T. Ng, "Rain removal in video by combining temporal and chromatic properties," in *Proc. IEEE Int. Conf. Multimedia Expo*, Toronto, ON, Canada, Jul. 2006, pp. 461–464.
[8] P. Barnum, T. Kanade, and S. G. Narasimhan, "Spatio-temporal frequency analysis for removing rain and snow from videos," in *Proc. PACV*, Oct. 2007, pp. 8–16.
[9] Y. Sun, X. Duan, H. Zhang, and Z. Yu, "A removal algorithm of rain and snow from images based on fuzzy connectedness," in *Proc. Int. Conf. Comput. Appl. Syst. Modeling*, Taiyuan, China, Oct. 2010, pp. 478–481.
[10] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic Single-Image-Based rain streaks removal via image decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.
[11] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2736–2744.
[12] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 3397–3405.
[13] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3855–3863.
[14] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Kobe, Japan, Jul. 2017, pp. 1357–1366.
[15] H. Hase, K. Miyake, and M. Yoneda, "Real-time snowfall noise elimination," in *Proc. Int. Conf. Image Process.*, Nice, France, 1999, pp. 406–409.
[16] S. Starik and M. Werman, "Simulation of rain in videos," in *Proc. ICCV*, Nice, France, vol. 2, Oct. 2003, pp. 406–409.
[17] K. Garg and S. K. Nayar, "When does a camera see rain?" in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Beijing, China, Oct. 2005, pp. 1067–1074.
[18] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 1968–1975.
[19] J. Chen and L.-P. Chau, "A rain pixel recovery algorithm for videos with highly dynamic scenes," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1097–1104, Mar. 2014.
[20] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2658–2670, Sep. 2015.
[21] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4057–4066.
[22] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2874–2883.
[23] M. Haris, S. Gregory, and U. Norimichi, "Recurrent back-projection network for video super-resolution," in *Proc. CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 3897–3906.
[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, vol. 2012, pp. 1097–1105.
[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[26] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Salt Lake City, UT, USA, Jun. 2018, pp. 864–873.
[27] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin, "Non-locally enhanced encoder-decoder network for single image de-raining," in *Proc. ACM Multimedia Conf. Multimedia Conf.*, San Diego, CA, USA, 2018, pp. 1056–1064.
[28] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3937–3946.
[29] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image deraining," *IEEE Trans. Neural Net. Learn. Sys.*, vol. 31, no. 6, pp. 1794–1807, Jul. 2019.
[30] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 254–269.
[31] J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li, "Robust video content alignment and compensation for rain removal in a CNN framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6286–6295.
[32] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, and D. Meng, "Video rain streak removal by multiscale convolutional sparse coding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6644–6653.
[33] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "FastDeRain: A novel video rain streak removal method using directional gradient priors," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 2089–2102, Apr. 2019.
[34] *PyTorch*. Accessed: Oct. 2016. [Online]. Available: http://pytorch.org
[35] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, USA, May 2015, pp. 1–5.
[36] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.
[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, Munich, Germany, Oct. 2015, pp. 234–241.

[38] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, 2017, pp. 2980–2988.
[39] D.-G. Ko, S.-H. Song, K.-M. Kang, and S.-W. Han, "Convolutional neural networks for character-level classification," *IEIE Trans. Smart Process. Comput.*, vol. 6, no. 1, pp. 53–59, Feb. 2017.
[40] W. Seok and C. Park, "Recognition of human motion with deep reinforcement learning," *IEIE Trans. Smart Process. Comput.*, vol. 7, no. 3, pp. 245–250, Jun. 2018.
[41] A. K. Tripathi and S. Mukhopadhyay, "Removal of rain from videos: A review," *Signal, Image Video Process.*, vol. 8, no. 8, pp. 1421–1430, Nov. 2014.
[42] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Should we encode rain streaks in video as deterministic or stochastic?" in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2516–2525.
[43] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Venice, Italy, Jun. 2015, pp. 5188–5196.
[44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[45] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.
[46] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.
[47] C. Wang, Z. Li, J. Wu, H. Fan, G. Xiao, and H. Zhang, "Deep residual haze network for image dehazing and deraining," *IEEE Access*, vol. 8, pp. 9488–9500, Jan. 2020.
[48] Y. Wang, S. Liu, D. Xie, and B. Zeng, "Removing rain streaks by a linear model," *IEEE Access*, vol. 8, pp. 54802–54815, Mar. 2020.
[49] H. Wang, Q. Xie, Q. Zhao, and D. Meng, "A model-driven deep neural network for single image rain removal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Virtual, Italy, Jun. 2020, pp. 3103–3112.
[50] R. Yasarla, V. A. Sindagi, and V. M. Patel, "Syn2Real transfer learning for image deraining using Gaussian processes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Virtual, Italy, Jun. 2020, pp. 2726–2736.
[51] W. Yang, R. T. Tan, S. Wang, Y. Fang, and J. Liu, "Single image deraining: From model-based to data-driven and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 19, 2020, doi: 10.1109/TPAMI.2020.2995190.
[52] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1377–1393, Jun. 2020.
[53] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, Jun. 2017.
[54] H. Lin, Y. Li, X. Fu, X. Ding, Y. Huang, and J. Paisley, "Rain O'er me: Synthesizing real rain to derain with data distillation," *IEEE Trans. Image Process.*, vol. 29, pp. 7668–7680, Jul. 2020.
[55] B. Li, Y. Gou, J. Z. Liu, H. Zhu, J. T. Zhou, and X. Peng, "Zero-shot image dehazing," *IEEE Trans. Image Process.*, vol. 29, pp. 8457–8466, Aug. 2020.
[56] H. Zhu, X. Peng, V. Chandrasekhar, L. Li, and J.-H. Lim, "DehazeGAN: When image dehazing meets differential programming," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, Jul. 2018, pp. 1234–1240.
[57] H. Zhu, Y. Cheng, X. Peng, J. T. Zhou, Z. Kang, S. Lu, Z. Fang, L. Li, and J.-H. Lim, "Single-image dehazing via compositional adversarial network," *IEEE Trans. Cybern.*, early access, Dec. 31, 2020, doi: 10.1109/TCYB.2019.2955092.

**KYU-HO LEE** received the B.S. degree in electronic engineering from the Hankuk University of Foreign Studies, South Korea, in 2014. He is currently pursuing the joint Master/Ph.D. degree with Korea University, Seoul, South Korea. His current research interests include image processing, computer vision, future displays including head mounted displays, and deep learning.

**EUNJI RYU** received the B.S. degree in display engineering from Hoseo University, Asan-si, Chungcheongnam-do, in 2018, and the M.S. degree in electrical engineering from Korea University, Seongbuk, Seoul, in 2020. Her research interests include image processing algorithms using deep learning-based methods and computer vision applications.

**JONG-OK KIM** (Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Korea University, Seoul, South Korea, in 1994 and 2000, respectively, and the Ph.D. degree in information networking from Osaka University, Osaka, Japan, in 2006. From 1995 to 1998, he served as an Officer with the Korea Air Force. From 2000 to 2003, he was with the SK Telecom Research and Development Center and Mcubeworks Inc., South Korea, where he was involved in research and development on mobile multimedia systems. From 2006 to 2009, he was a Researcher with the Advanced Telecommunication Research Institute International (ATR), Kyoto, Japan. He joined Korea University, Seoul, in 2009, where he is currently a Professor. His current research interests include image processing, computer vision, and intelligent media systems. He was a recipient of Japanese Government Scholarship, from 2003 to 2006.