IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 24, 2020, accepted March 17, 2020, date of publication March 24, 2020, date of current version April 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983087

A^2Net: Adjacent Aggregation Networks for Image Raindrop Removal

HUANGXING LIN, CHANGXING JING, YUE HUANG, AND XINGHAO DING
School of Information, Xiamen University, Xiamen 361000, China

Corresponding author: Xinghao Ding (dxh@xmu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 81671766, Grant 61971369, Grant U19B2031, and Grant 61671309, in part by the Open Fund of Science and Technology on Automatic Target Recognition Laboratory under Grant 6142503190202, and in part by the Fundamental Research Funds for the Central Universities under Grant 20720180059, Grant 20720190116, and Grant 20720200003.

ABSTRACT Existing methods for single images raindrop removal either have poor robustness or suffer from parameter burdens. In this paper, we propose a new Adjacent Aggregation Network (A^2Net) with lightweight architectures to remove raindrops from single images. Instead of directly cascading convolutional layers, we design an adjacent aggregation architecture to better fuse features for rich representations generation, which can lead to high quality images reconstruction. To further simplify the learning process, we utilize a problem-specific knowledge to force the network focus on the luminance channel in the YUV color space instead of all RGB channels. By combining adjacent aggregating operation with color space transformation, the proposed A^2Net can achieve state-of-the-art performances on raindrop removal with significant parameters reduction.

INDEX TERMS Raindrop removal, feature aggregation, YUV space, deep learning.

I. INTRODUCTION

Severe weather conditions, such as rain [1], haze [2], [3] and snow [4], impact not only human visual perception but also outdoor computer vision systems [5]. In rainy days, both dynamic rain streaks and static raindrops adhered to the camera lens can significantly degrade the captured image's quality. Raindrops in particular, can easily blur and block objects in the image due to light refraction and scattering. Since many computer vision systems are designed based on the assumption of clean inputs, their performance is easily affected by adhered raindrops. Thus, developing an effective method for raindrop removal is desirable for a wide range of practical applications.

Many methods have been proposed for rain streaks [1], [6]–[9] or snow [10], [11] removal. These methods usually rely on the physical properties specific to rain streaks or snow. For example, Fu *et al.* [1] exploit the high-frequency prior of rain streaks to help remove rain streaks. Intuitively, if we apply these derain/desnow methods directly to raindrop removal, the results can be disappointing. The reason is that the appearance and physical imaging process of raindrops are completely different from those of rain streaks and snow. Compared with rain streaks and snow, raindrops have more irregular geometric structures, which makes this problem more challenging. To address the restoration of raindrop degradation, some sophisticated convolutional neural networks have recently been developed. These CNNs [12], [13] employ very deep and complex architectures to effectively remove raindrops. While the visual results are often decent, the heavy computational burden prevents these CNNs from being applied to real systems.

In this paper, we make effort to develop a lightweight network that can effectively remove raindrops from a single raindrop-degraded image. To better learn the mapping function between the degraded image and the clean one, not only high-level semantic features but also rich spatial information [14] are required. Therefore, we design a new feature aggregating operation to better fuse spatial and semantic information across adjacent layers. Furthermore, we find that raindrops are short focal length convex lenses with the function of converging light. The light passing through a raindrop and converge to a point, causing a significant change in the luminance of the raindrop occluded area, while the chrominance will not be affected much. Based on this observation, we force our network to focus on the luminance channel of the YUV color space. Compared with the conventional way of processing all the RGB channels, this color space transformation can significantly simplify the learning process. The contributions of this work are threefold:

1) We propose a simple yet effective adjacent aggregation structure which fuses adjacent features for gen-

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

erating more informative representations. By directly deploying this simple aggregation into existing network architectures, the raindrop removal performance can be significantly improved without increasing parameters.
2) We observe a new phenomenon and utilize it for this specific raindrop removal problem. Specifically, instead of directly processing RGB, we intentionally force the network to focus on the luminance (Y channel), but less on the chrominance (UV channels). This divide-and-conquer strategy can effectively simplify the learning process.
3) By combining the aggregating operation with color space transformation, our adjacent aggregation networks (A^2Net) can achieve state-of-the-art performance on single images raindrop removal. The A^2Net exerts less pressure on system resources (*e.g.* CPU and memory), which makes it more suitable for practical applications.

II. RELATED WORKS

In this section, we briefly review several recent works related to image restoration in bad weather.

A. RAIN STREAKS REMOVAL

Traditional deraining methods are designed by using handcrafted image features to describe physical characteristics of rain streaks, or exploring prior knowledge to make the problem easier. These methods include sparse coding [9], [15], low-rank approximations [16], [17], and Gaussian mixture models (GMMs) [18]. To further improve generalization, several CNN-based methods [1], [19], [20] have also been proposed for image de-raining. In [1], a deep network takes the image detail layer as its input and predicts the negative residual as output. In [19], Zhang *et al.* propose a density-aware multi-stream densely connected convolutional neural network (DID-MDN) for joint rain density estimation and rain removal. In [20], a recurrent dilated network with multitask learning is proposed for joint rain streaks detection and removal. Since these methods are specifically developed for rain streaks, they cannot be expected to be efficient for raindrop removal.

B. SNOW REMOVAL

Most snow removal methods use priors of rainfall-driven features (*e.g.*, HOG) [21], [22], color assumptions [23] and frequency space separation [24], [25] to model falling snow particles. Recently, deep CNNs [10], [11] have also been applied to single image snow removal. Because the appearance of snow particles is similar, the performance of these methods is usually good.

C. RAINDROP REMOVAL

In general, the existing methods for raindrop removal can be categorized as model-based methods and learning-based methods.

Model-based methods are based on physical imaging process or raindrop geometric appearance to model the distribution of raindrops. In [26], Roser *et al.* attempt to model the shape of adherent raindrops by a sphere section. Furthermore, in [27], Bessel curves are used to obtain higher modeling accuracy. Since raindrops have various shapes and sizes, the above models can cover only a small portion of raindrop shapes. To simplify the raindrop removal problem, several hardware constraints, *e.g.*, multiple cameras [28], a stereo camera system [29] and pan-tilt surveillance cameras [30], are exploited. However, these methods cannot work with single cameras. In [31], You *et al.* detect raindrops by using motion and intensity temporal derivatives of input videos. Since this method requires consecutive video frames to extract raindrop features, it is not suitable for processing single images.

Learning-based methods use large amounts of data to learn and explore the characteristics of raindrops. In [32], Kurihata *et al.* detects raindrops on a windshield using raindrop features learned by PCA. When objects in the background are similar to raindrops, PCA cannot effectively extract the characteristics of raindrops and causes false detection. Recently, due to the large amount of available training data and computing resources, deep learning has become the most popular learning-based methods. In [33], Eigen *et al.* build a 3 layers network to extract features of static raindrops and dirt spots from synthetic images. While this method works well on small and sparse rain spots, it cannot remove large and dense raindrops. To tackle this problem, an AttentiveGAN [12] is proposed to simultaneously detect and remove raindrops. This method employs a recurrent network to detect raindrops. Though AttentiveGAN can remove raindrops efficiently, it tends to leave obvious artificials. In addition, AttentiveGAN contains a relatively large number of parameters and requires complex recurrent operations, which limits its potential value in practical applications with limited computing resources. Later, Quan *et al.* [13] propose a double attention guided CNN to restore the raindrop-degraded image. Since they also ignore the computational efficiency, the architecture of the proposed CNN is complicated.

III. OUR METHOD

We show the framework of our proposed A^2Net in Figure 1. Instead of designing a complex network architecture, we focus on how to better aggregate adjacent features to generate informative representation for clean image reconstruction. Furthermore, we utilize a problem-specific knowledge to simplify the learning process, *i.e.*, we learn the mapping function in the YUV color space rather than RGB. In this way, the tough single image raindrop removal problem can be well tackled with much less parameters.

A. ADJACENT AGGREGATION NETWORK

Since raindrops in the image exhibit various shapes, sizes and appearances, it is difficult for a network to identify all raindrops using single-scale features. Naturally, dealing with this problem in a multi-scale fashion is more suitable

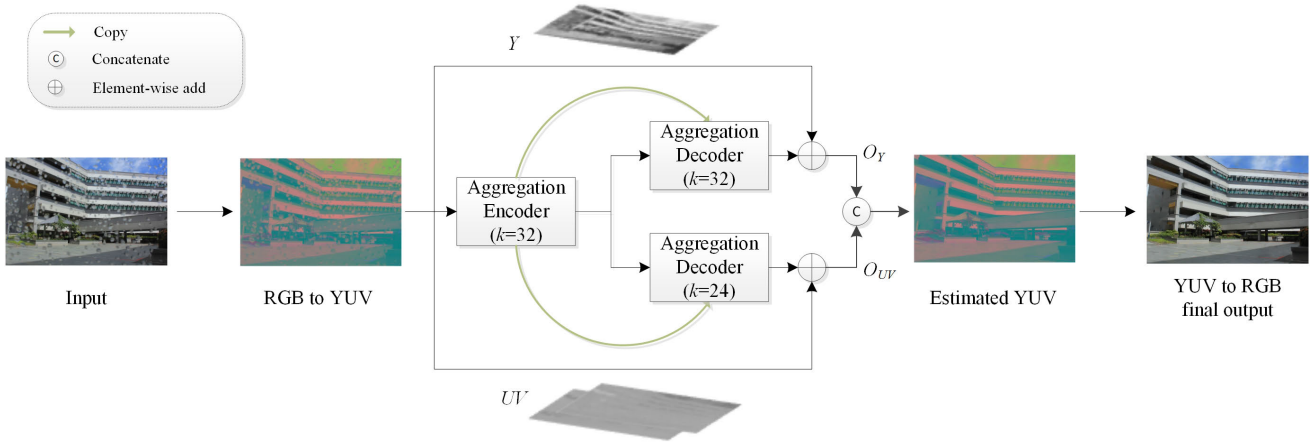


FIGURE 1. The framework of our A²Net. The input is firstly transformed into the YUV image and all YUV components share one encoder. Then the encoded features of Y and UV are separately decoded as O_Y and O_{UV} to estimate the clean YUV image. The final output is obtained by transforming the estimated YUV image into RGB image. k denotes the number of feature maps at each convolutional layer.

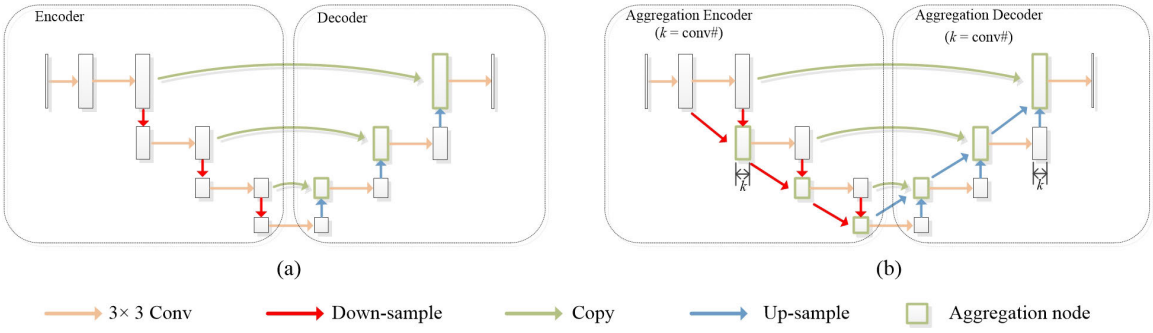


FIGURE 2. Our adjacent aggregation network. (a) A general encoder-decoder model, (b) Our adjacent aggregation network.

to capture complex geometric structures of raindrops. Fortunately, the hourglass structured network, *e.g.*, U-Net [34] and FPN [35], can well extract features at different scales. This classical architecture can benefit the network to identify various raindrops. However, during our experiment, we found that directly using this network cannot generate pleasing results. This is because the hourglass networks are good at extracting high-level semantic features at the cost of detail lost. For the specific raindrop removal problem, the networks not only need to extract strong semantic features to recognize raindrops but also require rich detail information to reconstruct clean images. The lateral connections, indicated by green lines in Figure 2(a), that fuse low- and high-level features can alleviate this problem to some degree. However, this simple fusion cannot help much to the quality of reconstructed images, since the low-level features are too noisy to provide sufficient semantic guidance [36].

To address the aforementioned problems, we design a new adjacent aggregation network based on the classical encoder-decoder architecture, as shown in Figure 2(b). Feature aggregation [14], [37], [38] has been proven to be good at extracting informative representations from its inputs. Here, we conduct a novel combination by introducing adjacent feature aggregation into the encoder-decoder network. The purpose of this combination is to better fuse information

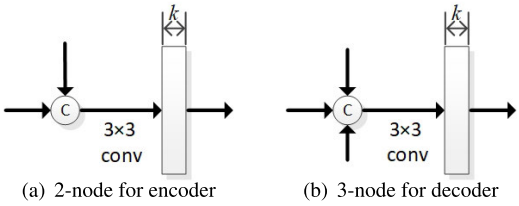


FIGURE 3. Illustration of aggregation node architectures. k denotes the number of feature maps.

across layers and scales. Our aggregating operation merges features extracted from adjacent layers and generate new representations with richer detail and semantics. In the aggregation encoder, the inputs of fusion are two adjacent features with same resolutions after down-sampling. In the aggregation decoder, the inputs of fusion are three features with same resolutions, two are adjacent features after up-sampling and one is from the encoder. These features are merged by using same aggregation nodes, as shown in Figure 3. The aggregation operation is defined by

$$X = \delta(W * C(X_1, \ldots, X_m) + b), \tag{1}$$

where $X_1, \ldots, X_m$ are the inputs of the aggregation node, C denotes the concatenation, $*$ indicates the convolution operation and $\delta(\cdot)$ is the nonlinear activation. W is a kernel

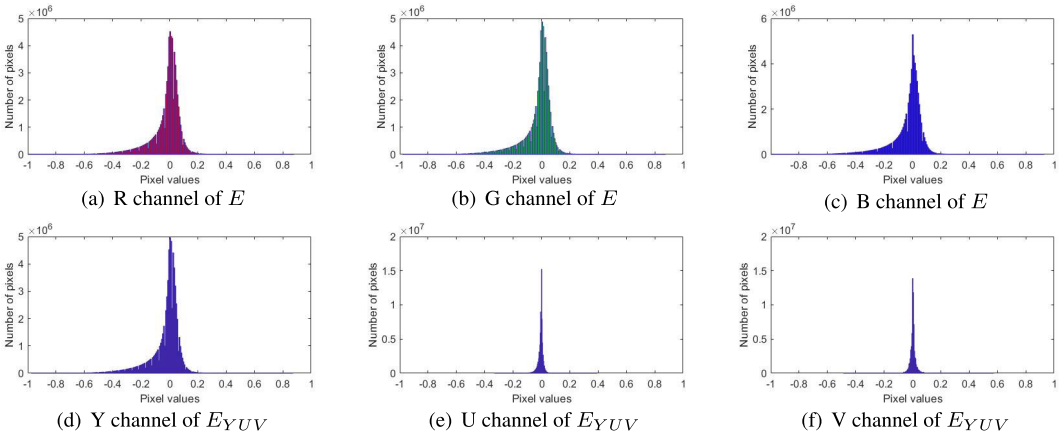


FIGURE 4. Residual histogram distributions of 200 pairs of clean and raindrop-degraded images. (a)-(c) represent histograms of residual distributions in R, G, and B channels, respectively. (d)-(f) represent histograms of residual distributions in Y, U, and V channels, respectively.

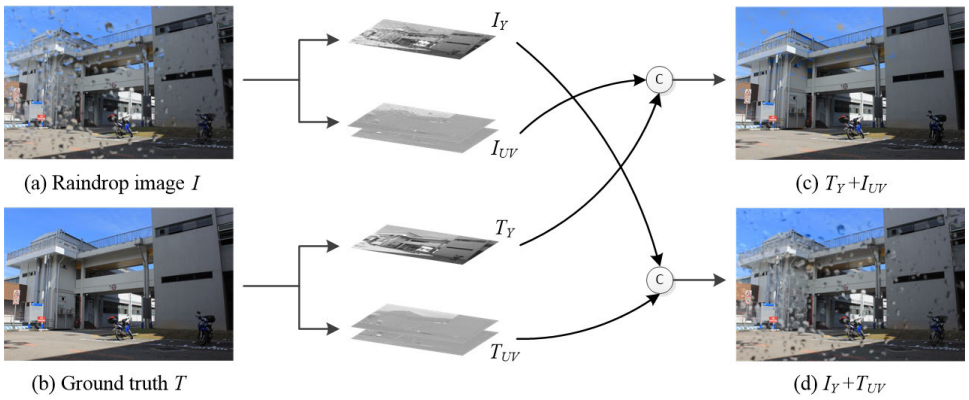


FIGURE 5. (a) Raindrop image (input) *I*. (b) Ground truth *T*. (c) An RGB image converted from a concatenation of the Y channel of *T* and the UV channels of *I*. (d) An RGB image converted from a concatenation of the Y channel of *I* and the UV channels of *T*.

of size 3×3 to fuse $X_1, \ldots, X_m$ and the kernel number is a hyper-parameter for each encoder and decoder. X is the fused feature which has the same resolution as $X_1, \ldots, X_m$. In this paper, we set the number of down-sample three times, and the aggregation decoder will correspondingly up-sample three times. Moreover, we set the kernel numbers to the same for all layers in each encoder or decoder based on our experiments.

B. COLOR SPACE CONVERSION

To further improve the raindrop removal results, we investigate the way raindrops degenerate images and observe an interesting phenomenon: the adherent raindrops are short focal length convex lenses with the function of converging light. The rays of reflected light from a wider environment will converge to a point after passing through a raindrop. According to Grassmann's laws [39], when two lights of different color are mixed, they give rise to a color which lies in between the two, while the total luminance of any mixture light is the sum of each light's luminance. Therefore, there should be obvious changes in the luminance of the raindrop occluded area in the image, while the chrominance will not be affected much. This observation inspires us that if we can remove raindrops from the luminance channel of the image, the quality of the raindrop-degraded image should be greatly enhanced, like shown in Figure 5(c).

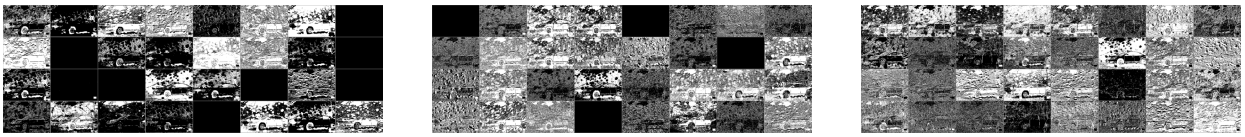
Computer vision tasks are dominated by the RGB color space, which seems to be the most intuitive choice. We hereinafter denote the input raindrop-degraded RGB image as I and the corresponding ground truth as T. Then the residual E can be represented as

$$E = T - I. \tag{2}$$

However, RGB color space does not separate the luminance from the chrominance components [40], and pixel values in RGB color space are highly correlated. This makes it a poor choice for luminance and chrominance analysis. Therefore, we convert I from the RGB color space into the YUV color space, where the luminance (Y channel) and chrominance (UV channels) are separated. The residual in YUV color model can be expressed as

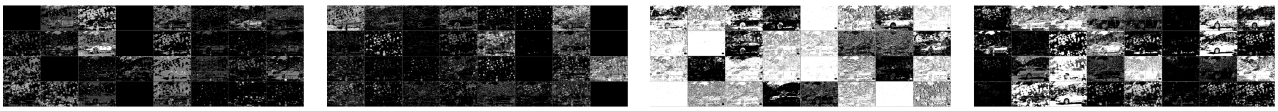
$$E_{YUV} = T_{YUV} - I_{YUV}, \tag{3}$$

where the subscript 'YUV' denotes the YUV color model.



(a) Adjacent features 1 (b) Adjacent features 2 (c) Fusing (a) and (b)

FIGURE 6. The aggregation result of the first aggregation node in the encoder.



(a) Adjacent features 1 (b) Adjacent features 2 (c) Low-level features (d) Fusing (a), (b) and (c)

FIGURE 7. The aggregation result of the last aggregation node in the decoder for Y channel. Please zoom in for a better visualization.

To verify the effectiveness of our color space conversion strategy is, we randomly select 200 pairs of raindrop-degraded images and the corresponding raindrop-free images from the raindrop dataset [12]. Figure 4 shows the residual histogram distributions over all 200 pairs of images. Since channels in RGB space are strongly coupled, raindrops have almost the same effect on each channel in RGB space, as shown in Figures 4(a) to (c). While in the YUV space, the residual components in UV channels have significant range reduction in pixel value compared with Y channel, as shown in Figures 4(d) to (f). This implies that the impact of raindrops on the image is mainly concentrated in the luminance channel. Furthermore, using a simple channel swap operation, we observe that raindrops are rare in the UV channels (as shown in Figure 5(c)), while most of the raindrops remain in the Y channel (as shown in Figure 5(d)).

Given the above, the YUV space should be a better choice for raindrop removal than RGB space. YUV space was used to reduce computation burden in previous methods of rain streaks removal or denoising [18], [41]. To the raindrop removal, utilizing YUV involves basic physical properties. In the YUV space, luminance (Y) and chrominance (UV) information can be processed separately, so we can divide the single image-to-image mapping problem into two easier sub-problems. In addition, since most raindrops exhibit in the luminance channel, we can intentionally force the network to focus on the Y channel to simplify the learning process.

C. PROPOSED A^{2}Net

Combining with the adjacent aggregation and color space conversion strategy, we propose the A^{2}Net as shown in Figure 1. Our A^{2}Net contains one aggregation encoder and two aggregation decoders. One decoder is for Y channel and the other is for UV channels. In this way, the network can achieve better raindrop removal performance than directly learning the mapping between images with only one decoder, as shown in Figure 10. The lateral connections are used to connect the encoder and decoders for information propogation. At the same time, we find that Y and UV channels can share one encoder to guide each other for better raindrop removal and parameters efficiency. Our network adopts global residual learning with the long short-cut to ease the learning process [1]. The output components are represented as O_Y and O_{UV}, which are concatenated to generate the estimated clean YUV image. Finally, we convert the YUV image to the RGB image. We set the kernel number of the decoder in Y channel larger than that in the UV channels, so that our network can mainly concentrate on the Y channel. In the final A^{2}Net, the kernel numbers are set to 32, 32 and 24, as shown in Figure 1.

To illustrate the value of the proposed aggregation operation, we show the learned feature maps in Figure 6 and Figure 7. Figure 6 shows the aggregation result of the first aggregation node in the encoder. Figures 6(a) and (b) are the adjacent feature maps. As can be seen, the two feature maps contain many invalid information, while the fused features are more informative as shown in Figure 6(c). The fused features retain more details while gaining stronger semantics, which can bring benefits for reconstructing images. Figure 7 shows the aggregation result of the last aggregation node in the decoder for Y channel. The fused features (as shown in Figure 7(d)) can better capture the raindrops, which makes it easier for the decoder to estimate the residual component in Y channel. We argue that the performance of raindrop removal can be significantly enhanced by effectively aggregating adjacent features.

D. LOSS FUNCTION

The mean squared error (MSE) is widely used to train a network. However, raindrops are blend with objects and background scene, it is hard to distinguish between raindrops and objects by simply minimizing MSE loss. A model with only MSE loss tends to result in a blurred reconstruction. Therefore, for Y and UV channels we adopt MSE + SSIM loss [42] which can preserve global structure better as well as keeping per-pixel similarity. We minimize the combination of those loss functions to train the encoder and decoders. The loss functions for Y and UV channels are defined by:

$$L_Y = \frac{1}{M}\sum_{i=1}^{M}(L_{MSE}(O_Y^i, T_Y^i) + L_{SSIM}(O_Y^i, T_Y^i)), \tag{4}$$

TABLE 1. Network architecture of aggregation encoder. N_{out} denotes the number of output feature maps for each layer.

NAME	N_{out}	FUNCTION
INPUT	3	
ENC_CONV0	32	Convolution 3 × 3
ENC_CONV1	32	Convolution 3 × 3
DOWNSAMPLE1_1	32	Downsample output of ENC_CONV0
DOWNSAMPLE1_2	32	Downsample output of ENC_CONV1
ENC_CONCAT1	64	Concatenate outputs of DOWNSAMPLE1_1 and DOWNSAMPLE1_2
ENC_CONV2	32	Convolution 3 × 3
ENC_CONV3	32	Convolution 3 × 3
DOWNSAMPLE2_1	32	Downsample output of ENC_CONV2
DOWNSAMPLE2_2	32	Downsample output of ENC_CONV3
ENC_CONCAT2	64	Concatenate outputs of DOWNSAMPLE2_1 and DOWNSAMPLE2_2
ENC_CONV4	32	Convolution 3 × 3
ENC_CONV5	32	Convolution 3 × 3
DOWNSAMPLE3_1	32	Downsample output of ENC_CONV4
DOWNSAMPLE3_2	32	Downsample output of ENC_CONV5
ENC_CONCAT3	64	Concatenate outputs of DOWNSAMPLE3_1 and DOWNSAMPLE3_2
ENC_CONV6	32	Convolution 3 × 3

TABLE 2. Network architecture of aggregation decoder. Number of feature maps k and network output channels m andepend on the experiment. For Y channel, $m = 1, k = 24$, and for UV channels, $m = 2, k = 32$.

NAME	N_{out}	FUNCTION
DEC_CONV6	k	Convolution 3 × 3
UPSAMPLE3_1	k	Upsample output of ENC_CONV6
UPSAMPLE3_2	k	Upsample output of DEC_CONV6
DEC_CONCAT3	$2k + 32$	Concatenate outputs of UPSAMPLE3_1, UPSAMPLE3_2 and ENC_CONV5
DEC_CONV5	k	Convolution 3 × 3
DEC_CONV4	k	Convolution 3 × 3
UPSAMPLE2_1	k	Upsample output of DEC_CONV5
UPSAMPLE2_2	k	Upsample output of DEC_CONV4
DEC_CONCAT2	$2k + 32$	Concatenate outputs of UPSAMPLE2_1, UPSAMPLE2_2 and ENC_CONV3
DEC_CONV3	k	Convolution 3 × 3
DEC_CONV2	k	Convolution 3 × 3
UPSAMPLE1_1	k	Upsample output of DEC_CONV3
UPSAMPLE1_2	k	Upsample output of DEC_CONV2
DEC_CONCAT1	$2k + 32$	Concatenate outputs of UPSAMPLE1_1, UPSAMPLE1_2 and ENC_CONV1
DEC_CONV1	k	Convolution 3 × 3
DEC_CONV1	m	Convolution 3 × 3

$$L_{UV} = \frac{1}{M}\sum_{i=1}^{M}(L_{MSE}(O^i_{UV}, T^i_{UV}) + L_{SSIM}(O^i_{UV}, T^i_{UV})), \quad (5)$$

where M is the number of training data, T_Y and T_{UV} denote the components of ground truth, L_{MSE} and L_{SSIM} are the MSE and SSIM loss, respectively.

The overall loss function can be written as:

$$L = L_Y + \alpha \cdot L_{UV}, \quad (6)$$

where α is the parameter to balance the two losses. We set $\alpha = 0.6$ based on cross validation, which forces the network to focus on the Y channel.

E. PARAMETER SETTING

As discussed, the proposed A²Net consists of one aggregation encoder and two aggregation decoders. Table 1 and 2 show the details of aggregation encoder and aggregation decoder. Each encoder and decoder includes several 3 × 3 convolutional kernels. In the encoder, the down-sampling operations are performed by 4 × 4 convolution with stride 2. In the decoders, we use 2 × 2 deconvolution with stride 2 to upsample the low-resolution features. The activation functions for all convolutional and deconvolution layers are *ReLU* [43], while the activation function of the last layer is *Tanh*.

F. TRAINING DETAILS

We use the dataset from [12], which contains 861 pairs of raindrop/clean images, to train our networks. We randomly select 256 × 256 image patches from training set as inputs. We keep the learning rate to 2×10^{-4} for the first 2000 epochs and linearly decay the rate to 0 for the next 2000 epochs. The mini-batch size is set to 4. All experiments are performed on a server with Inter(R) Core(TM) i7-8700K CPU, 16GB RAM and NVIDIA GeForce GTX 1080 Ti.

TABLE 3. Comparisons on SSIM|PSNR values and parameter numbers.

Method	SSIM	PSNR	Parameters #
DDN [1]	0.8177	25.80	0.06M
DID [19]	0.8013	19.01	0.14M
JORDER-E [20]	0.8616	25.81	4.2M
Eigen [33]	0.7678	22.69	0.75M
Pix2pix [44]	0.8085	24.19	54.41M
AttentiveGAN [12]	0.8960	28.30	6.24M
Our A²Net	**0.9074**	**28.43**	0.40M

TABLE 4. Comparisons on increasing kernel numbers.

Method	SSIM	PSNR	Parameters #
Our A²Net-32	0.9078	28.45	0.45M
Our A²Net	0.9074	28.43	0.40M

IV. EXPERIMENTS

In this section, we present experimental results on synthetic and real-world dataset. We compare our A²Net with several deep learning methods: DDN [1], DID-MDN [19], JORDER-E [20], Eigen [33], Pix2pix [44] and AttentiveGAN [12]. Note that, DDN, DID-MDN and JORDER-E are three state-of-the-art rain streaks removal methods.

A. RESULTS ON SYNTHETIC DATASET

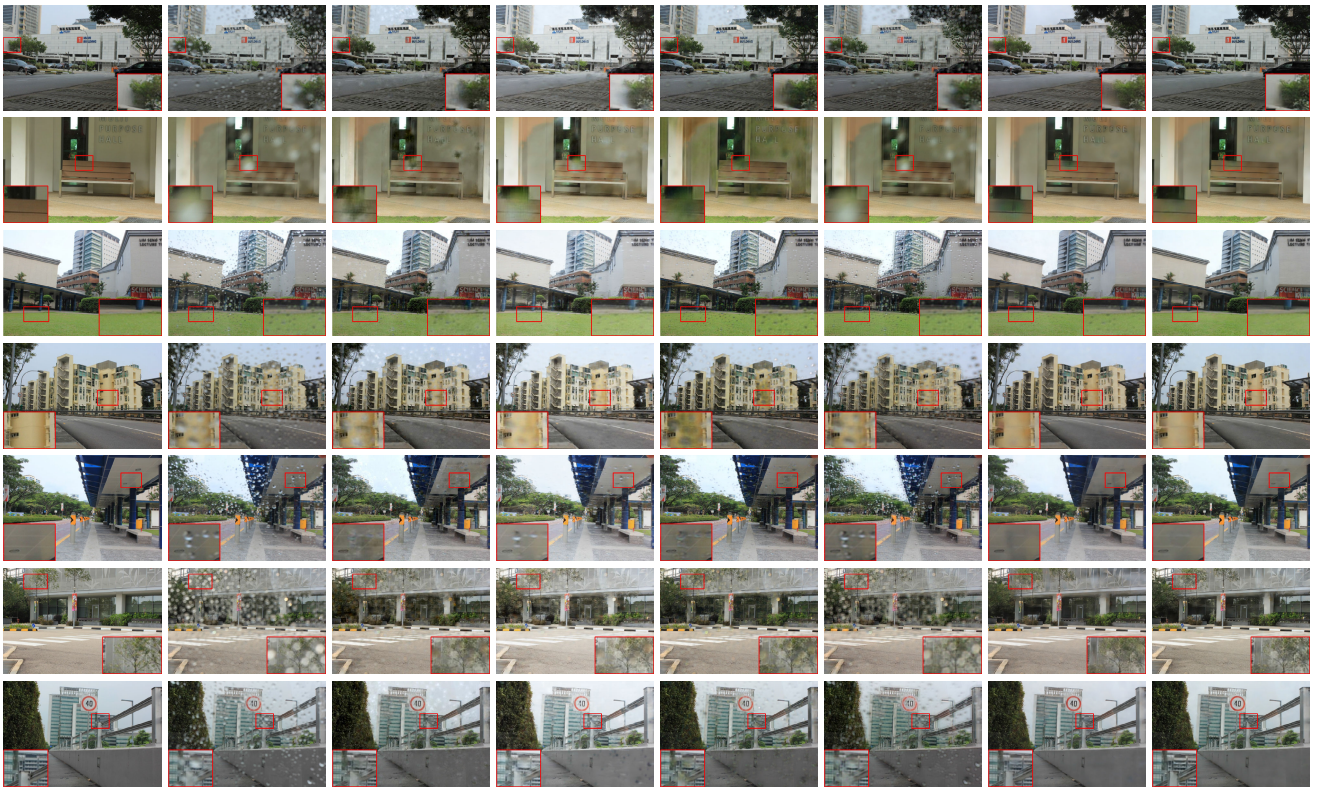
There are two test dataset, testA and testB, provided by [12]. The testA contains only 58 pairs of well aligned images. The testB contains 249 pairs of images that better reflect the complexity of the raindrop distribution, but some of the image pairs are not perfectly aligned. To ensure a fair comparison, we merge testA and testB, then strip out the unaligned images and get 177 pairs of precisely aligned images for testing.[1]

We show the visual comparisons in Figure 8. As can be seen, our method significantly outperforms other methods and is successful in restoring the images. The rain streaks removal methods (*i.e.* DDN, DID-MDN, JORDER-E) fail to remove raindrops. These results are reasonable because the physical properties of raindrops are very different from those of rain streaks. The GAN-based method AttentiveGAN can remove raindrops effectively but always loses some details. In contrast, our method achieves promising results in removing raindrops, enhancing the visibility and preserving details.

SSIM and PSNR are also adopted to perform quantitative evaluations in Table 3. As observed, our method achieves higher SSIM and PSNR than AttentiveGAN and considerably outpeforms other methods. This indicates that our method can generate results more similar to the ground truth. Moreover, our A²Net contains far fewer parameters than AttentiveGAN, potentially making it more suitable for practical applications.

On the other hand, we tune the kernel numbers for all encoder and decoders to 32, and refer to this model as A²Net-32. As shown in Table 4, the performance of A²Net-32 is slightly better than that of the final A²Net, but the number

[1]Our code and datasets will be released soon.



(a) Ground truth (b) Input (c) DDN (d) DID-MDN (e) JORDER-E (f) Eigen (g) AttentiveGAN (h) Our A^{2}Net

FIGURE 8. Visual comparisons on raindrop dataset provided by [12].

of parameters is increased to 0.45M. This experiment further confirms that the UV channels does not have too much impact on the effect of raindrop removal. Therefore, we set the kernel numbers to 32, 32 and 24 as the default setting to balance the performance and parameters burden.

B. RESULTS ON REAL-WORLD DATA

In this section, we show that A^{2}Net, which is trained on the raindrop dataset [12], can still works well on real-world data. We collect 100 real-world raindrop images from the Internet as a new dataset. Since no ground truth exists, we only show the qualitative results on real-world data in Figure 9. As can be seen, our A^{2}Net achieves the best visual performance on removing raindrops and preserving details compared to other methods.

C. RUNNING TIME

Our A^{2}Net can process new images very efficiently. Table 5 shows the average running time of 100 test images for three different image sizes on both CPU and GPU. With the increase of image size, most deep models require substantial amounts of memory and time to work. In contrast, our method has the faster computational speed and lower hardware requirements. As a pre-processing for other high-level vision tasks, the raindrop removal process should be simpler and faster. Our A^{2}Net is a relatively shallow network that requires fewer calculations, so it is more practical, for example, on mobile devices.

D. ABLATION STUDY

We provide an ablation study to demonstrate the advantage of each part of our model.

1) YUV SPACE AND FEATURE AGGREGATION

To validate the necessity of color space conversion strategy and adjacent feature aggregation operations, we design three variants of the A^{2}Net for exhaustive comparison. One is called Net^{YUV} that only keeps one aggregation encoder and one aggregation decoder. The Net^{YUV} directly learns the image-to-image mapping by focusing on all YUV channels in YUV space. The second is called Net^{RGB} which has the same structure as Net^{YUV}, but is trained in RGB space. The last one is a general UNet that only keeps one general encoder and one general decoder, as shown in Figure 2(a). The UNet is also trained in RGB space. We adopt global residual learning for each network structure.

We show the comparison of the above networks in Figure 10. Since these networks have similar numbers of layers, we only tune their kernel numbers to control the parameters. We find that Net^{RGB} can achieve higher SSIM and better parameters efficiency than UNet. This is because the adjacent aggregation operation can effectively enhance the network capacity. On the other hand, the Net^{YUV} cannot



(a) Input (b) DDN (c) DID-MDN (d) JORDER-E (e) Eigen (f) AttentiveGAN (g) Our A^{2}Net

FIGURE 9. Visual comparisons on real-world raindrop images.

TABLE 5. Comparison of running time (seconds). "\" represents "out of memory".

	DDN [1]		DID [19]		JORDER-E [20]		Eigen [33]		Pix2pix [44]		AttentiveGAN [12]		Our A^{2}Net	
Image size	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU
256×256	0.91	0.08	1.77	0.10	18.80	0.12	9.95	0.13	30.7	0.11	11.16	0.09	1.4	0.07
512×512	2.25	0.16	7.26	0.31	211.8	0.52	220.5	0.22	705.1	1.13	114.1	0.22	5.4	0.11
768×768	4.32	0.31	23.10	0.46	\	1.37	\	0.70	\	1.85	\	\	12.5	0.40

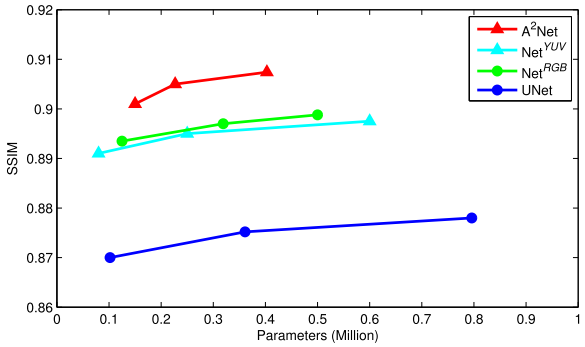
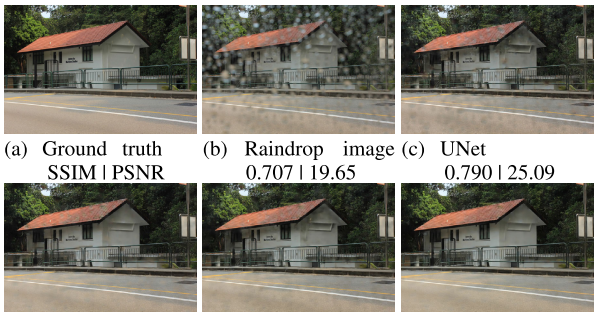


FIGURE 10. Quantitative comparisons on ablation study.



(a) Ground truth SSIM | PSNR (b) Raindrop image 0.707 | 19.65 (c) UNet 0.790 | 25.09

(d) Net^{RGB} 0.839 | 27.41 (e) Net^{YUV} 0.834 | 26.60 (f) A^{2}Net 0.866 | 28.69

FIGURE 11. Visual comparisons on ablation study.

produce satisfactory results since it directly learns the image-to-image mapping by using only one decoder. In contrast, our final A^{2}Net handles the Y and UV channels separately and achieves better raindrop removal performance. This indicates that the raindrop removal problem can be simplified with the color space conversion strategy.

Visual comparisons are shown in Figure 11. As can be seen in Figure 11(c), some of the raindrops in the image can be removed via UNet, but the quality of the reconstructed image is not desirable. The adjacent aggregation operation helps a lot to preserve image details, and the color space conversion strategy can further improve the raindrop removal performance, as shown in Figures 11(d) and (f).

2) OTHER COLOR SPACES

There are some other color spaces (*e.g.* CIE Lab) that can separate the luminance and chrominance of an image. We further test whether our method is still efficient in other color spaces that meet our motivation. We use Lab space to replace YUV space. In Lab space, a color image is split into luminance (L channel) and chrominance (ab channels). We again train the models on the raindrop dataset [12]. The subjective comparisons are presented in Figure 12, where we see that the visual quality of Figure 12(c) and (d) is comparable. On the other hand, our A^{2}Net achieves the average SSIM/PSNR of 0.9055/28.36 dB in Lab space, close to the YUV space of 0.9074/28.43 dB. This experiment proves that our method performs well not only in YUV space, but also in other color spaces where luminance and chrominance are separated.

3) UV CHANNELS

We also test the effect of UV channels on the final results. We train a new model, called Net^{Y}, which contains only one



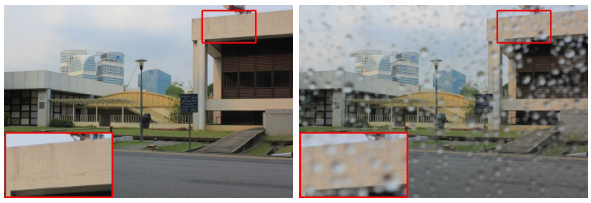
(a) Ground truth
SSIM | PSNR

(b) Raindrop image
0.822 | 20.58

(c) Lab
0.896 | 26.45

(d) YUV
0.898 | 26.73

FIGURE 12. Raindrop removal results using different color space.



(a) Ground truth
SSIM | PSNR

(b) Raindrop image
0.759 | 21.44

(c) Net^Y
0.897 | 30.18

(d) Ours
0.909 | 31.32

FIGURE 13. Raindrop removal example.

decoder for the Y channel, not for the UV channels. This means that the raindrops remain in the UV channels of the resulting image. Net^Y gets an average of 0.9005/28.01 dB in terms of SSIM and PSNR for the test set [12], while our final A^{2}Net gives a slightly better 0.9074/28.43 dB. Subjective comparisons are presented in Figure 13. As can be seen, Net^Y effectively removes all raindrops, but the result still left obvious artifacts. These experiments show that the network can achieve good raindrop removal results by focusing on the Y channel, but the UV channel also slightly affects the image quality.

4) HYPER-PARAMETER α

In eq.(6), the losses for Y and UV channels are weighted by parameter α. We test our model with $\alpha \in \{0.3, 0.6, 0.9, 1.2, 1.5\}$ and the evaluation results are shown in Table 6. As can be seen, our model achieves the best performance when $\alpha = 0.6$. According to this parameter setting, our model focuses on the Y channel, and the UV channels are used to further enhance the visual quality.

TABLE 6. Quantitative comparisons for different α.

α	0.3	0.6	0.9	1.2	1.5
SSIM	0.9041	0.9074	0.9060	0.9042	0.9037
PSNR	28.31	28.43	28.35	28.09	28.11

V. FUTURE WORK

Despite superior results of our methods, there still remain several problems to explore. For example, we only focus on the removal of raindrops, without considering the removal of fog and rain streaks. We can investigate image restoration in more severe weather conditions, in which both rain and fog exist. In addition, to break the limitations of data, unsupervised methods are also worth exploring.

VI. CONCLUSION

In this paper, we have proposed a adjacent aggregation network for single image raindrop removal. Our network employs an encoder-decoder structure with a new adjacent aggregation operation to effectively extract informative features. To simplify the raindrop removal problem, our A^{2}Net learns the raindrop removal function in YUV color space. In this way, the complex image-to-image mapping problem can be transformed into luminance (Y channel) and chrominance (UV channels) processing problems. Most of the network's attention is focused on the Y channel. Our A^{2}Net contains only 0.4M parameters while can still achieve state-of-the-art performance.

REFERENCES

[1] 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)*, Jul. 2017, pp. 3855–3863.

[2] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[3] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 154–169.

[4] Y. Wang, S. Liu, C. Chen, and B. Zeng, "A hierarchical approach for rain or snow removing in a single color image," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3936–3950, Aug. 2017.

[5] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? Deep joint recurrent rain removal and reconstruction in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3233–3242.

[6] 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.

[7] 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)*, Jul. 2017, pp. 1357–1366.

[8] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *Int. J. Comput. Vis.*, vol. 86, nos. 2–3, pp. 256–274, Jan. 2010.

[9] C.-H. Son and X.-P. Zhang, "Rain removal via shrinkage of sparse codes and learned rain dictionary," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–6.

[10] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, "DesnowNet: Context-aware deep network for snow removal," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3064–3073, Jun. 2018.

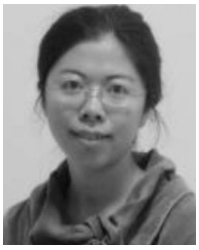
[11] Z. Li, J. Zhang, Z. Fang, B. Huang, X. Jiang, Y. Gao, and J.-N. Hwang, "Single image snow removal via composition generative adversarial networks," *IEEE Access*, vol. 7, pp. 25016–25025, 2019.
[12] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2482–2491.
[13] Y. Quan, S. Deng, Y. Chen, and H. Ji, "Deep learning for seeing through window with raindrops," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2463–2471.
[14] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2403–2412.
[15] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng, "Joint bi-layer optimization for single-image rain streak removal," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2526–2534.
[16] 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.*, Dec. 2013, pp. 1968–1975.
[17] H. Zhang and V. M. Patel, "Convolutional sparse and low-rank coding-based rain streak removal," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 1259–1267.
[18] 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)*, Jun. 2016, pp. 2736–2744.
[19] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 695–704.
[20] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 28, 2019, doi: 10.1109/TPAMI.2019.2895793.
[21] J. Bossu, N. Hautière, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 348–367, Jul. 2011.
[22] S.-C. Pei, Y.-T. Tsai, and C.-Y. Lee, "Removing rain and snow in a single image using saturation and visibility features," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2014, pp. 1–6.
[23] J. Xu, W. Zhao, P. Liu, and X. Tang, "An improved guidance image based method to remove rain and snow in a single image," *Comput. Inf. Sci.*, vol. 5, no. 3, p. 49, 2012.
[24] X. Zheng, Y. Liao, W. Guo, X. Fu, and X. Ding, "Single-image-based rain and snow removal using multi-guided filter," in *Proc. Int. Conf. Neural Inf. Process.* Springer, 2013, pp. 258–265.
[25] D. Rajderkar and P. S. Mohod, "Removing snow from an image via image decomposition," in *Proc. IEEE Int. Conf. Emerg. Trends Comput., Commun. Nanotechnol. (ICECCN)*, Mar. 2013, pp. 576–579.
[26] M. Roser and A. Geiger, "Video-based raindrop detection for improved image registration," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops, ICCV Workshops*, Sep. 2009, pp. 570–577.
[27] M. Roser, J. Kurz, and A. Geiger, "Realistic modeling of water droplets for monocular adherent raindrop recognition using Bézier curves," in *Proc. Asian Conf. Comput. Vis.*, 2010, pp. 235–244.
[28] A. Yamashita, M. Kuramoto, T. Kaneko, and K. T. Miura, "A virtual wiper-restoration of deteriorated images by using multiple cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 4, Oct. 2003, pp. 3126–3131.
[29] Y. Tanaka, "Removal of adherent waterdrops from images acquired with a stereo camera system," *IEICE Trans. Inf. Syst.*, vol. 89, no. 7, pp. 2021–2027, Jul. 2006.
[30] A. Yamashita, T. Harada, T. Kaneko, and K. T. Miura, "Removal of adherent noises from images of dynamic scenes by using a pan-tilt camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2004, pp. 437–442.
[31] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi, "Adherent raindrop modeling, detectionand removal in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1721–1733, Sep. 2016.
[32] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu, and T. Miyahara, "Rainy weather recognition from in-vehicle camera images for driver assistance," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2005, pp. 205–210.
[33] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 633–640.
[34] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
[35] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[36] Z. Zhang, X. Zhang, C. Peng, D. Cheng, and J. Sun, "Exfuse: Enhancing feature fusion for semantic segmentation," in *Proc. ECCV*, 2018, pp. 269–284.
[37] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," 2019, *arXiv:1904.04514*. [Online]. Available: http://arxiv.org/abs/1904.04514
[38] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5693–5703.
[39] D. Malacara, *Color Vision and Colorimetry: Theory and Applications*. Bellingham, WA, USA: SPIE, 2011.
[40] Z. H. Al-Tairi, R. W. Rahmat, M. I. Saripan, and P. S. Sulaiman, "Skin segmentation using YUV and RGB color spaces," *J. Inf. Process. Syst.*, vol. 10, no. 2, pp. 283–299, Jun. 2014.
[41] X. Xue, X. Jin, C. Zhang, and S. Goto, "Motion robust rain detection and removal from videos," in *Proc. IEEE 14th Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2012, pp. 170–174.
[42] 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.
[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.
[44] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image translation with conditional adversarial networks," 2016, *arXiv:1611.07004*. [Online]. Available: http://arxiv.org/abs/1611.07004



HUANGXING LIN received the B.S. degree in tracking technology and instrument from Beijing Jiaotong University, in 2015, and the M.S. degree in control theory and control engineering from Xiamen University, Xiamen, China, in 2018, where he is currently pursuing the Ph.D. degree in signal and information processing. His research interests include MRI reconstruction, medical image segmentation, and machine learning.



CHANGXING JING received the master's degree in control engineering from China Jiliang University, Hangzhou, China, in 2018. He is currently pursuing the Ph.D. degree in signal and information processing with Xiamen University, Xiamen, China. His research interests include image processing and deep learning.



YUE HUANG received the B.S. degree from Xiamen University, Xiamen, China, in 2005, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2010. From 2015 to 2016, she was a Visiting Scholar with Carnegie Mellon University. She is currently an Associate Professor with the Department of Communication Engineering, School of Information Science and Engineering, Xiamen University. Her main research interests include machine learning and image processing.



XINGHAO DING was born in Hefei, China, in 1977. He received the B.S. and Ph.D. degrees from the Hefei University of Technology, Hefei, in 1998 and 2003, respectively. From 2009 to 2011, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Pratt School of Engineering, Duke University, Durham, NC, USA. Since 2011, he has been a Professor with the Department of Communication Engineering, School of Information Science and Engineering, Xiamen University, Xiamen, China. His main research interests include image processing and machine learning.