Intensity-Aware Single-Image Deraining with Semantic and Color Regularization

Ke Xu, Xin Tian, Xin Yang†, Baocai Yin, Rynson W.H. Lau†

Abstract—**Rain degrades image visual quality and disrupts object structures, obscuring their details and erasing their colors. Existing deraining methods are primarily based on modeling either visual appearances of rain or its physical characteristics (*e.g.*, rain direction and density), and thus suffer from two common problems. First, due to the stochastic nature of rain, they tend to fail in recognizing rain streaks correctly, and wrongly remove image structures and details. Second, they fail to recover the image colors erased by heavy rain. In this paper, we address these two problems with the following three contributions. First, we propose a novel PHP block to aggregate comprehensive spatial and hierarchical information for removing rain streaks of different sizes. Second, we propose a novel network to first remove rain streaks, then recover objects structures/colors, and finally enhance details. Third, to train the network, we prepare a new dataset, and propose a novel loss function to introduce semantic and color regularization for deraining. Extensive experiments demonstrate the superiority of the proposed method over state-of-the-art deraining methods on both synthesized and real-world data, in terms of visual quality, quantitative accuracy, and running speed.**

Index Terms—**Rain removal, image reconstruction, neural networks.**

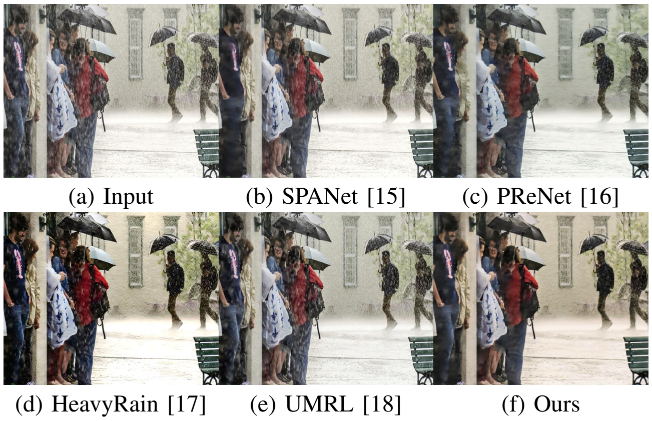

(a) Input (b) SPANet [15] (c) PReNet [16]
(d) HeavyRain [17] (e) UMRL [18] (f) Ours

Fig. 1: While state-of-the-art deraining methods (b to e) fail to correctly separate rain from image structures/details and recover image colors, the proposed method (f) produces a better rain-free image via learning intensity-aware deraining features with semantic and color regularization.

I. Introduction

ALTHOUGH rain is crucial to our ecosystem, it is undesirable for many vision tasks, such as object detection [1], [2], visual tracking [3], [4] and image editing [5], [6]. It fails these tasks by disrupting the image content and degrading the image quality. There are many methods proposed to address this problem. Some help remove rain from a video [7], [8], [9], [10], [11], [12], [13], [14] and rely on inter-frame priors as additional constraints. Others remove rain from a single image, which is more challenging due to the lack of temporal constraints. In this work, we focus on the single-image rain streak removal problem.

Existing single-image deraining methods focus on modeling the physical characteristics of rain or its visual appearances. For example, they model the sparsity of rain streaks via sparse coding [19], [20], [21], Gaussian mixture model [22], convolutional networks [23], [24]. Latest works also consider the locations [15] and directions [25] of rain streaks, rain densities [26], or rain accumulation effects [17], [27], [28], [16]. Despite all these efforts, the stochastic nature of rain still challenges state-of-the-art methods in two ways. First, the unknown intensities of rain streaks make it difficult for existing methods to correctly separate rain streaks from the background, causing them to mis-recognize image structures/details as rain and vise versa. Second, heavy rain can wash away image colors. Existing methods try to address this problem with dehazing priors [25], [17], [27] or recurrent networks [28], [16]. They primarily aim at enhancing the contrasts of heavy rain images, instead of recovering their colors.

In this paper, we aim to address the above two problems. Our method is based on two insights. First, knowing the intensities of rain streaks can help detect and remove them more accurately. However, the intensities of rain streaks are difficult to predict, as they depend on the background color, scene radiance and exposure time [7]. We note that the sizes of rain streaks can provide some valuable cues of their intensities. As described in [29], rain intensities decrease as the distances of rain streaks from the camera increase. We observe that rain streaks of larger sizes tend to be closer to the camera, which implies that these rain streaks tend to have higher intensities than those further away from the camera (*i.e.*, of smaller sizes). This inspires us to aggregating multi-scale spatial information for detecting rain of different sizes, and multi-scale hierarchical information for removing them. Second, we observe that humans can see through heavy rain,

Ke Xu is with Shanghai Jiao Tong University and City University of Hong Kong. Email: kkangwing@gmail.com

Xin Tian is with Dalian University of Technology and City University of Hong Kong. Email: xtian@mail.dlut.edu.cn

Xin Yang and Baocai Yin are with Dalian University of Technology, China. Email: xinyang@dlut.edu.cn, ybc@dlut.edu.cn

Rynson W.H. Lau is with City University of Hong Kong, Hong Kong. E-mail: Rynson.Lau@cityu.edu.hk

† Xin Yang and Rynson W.H. Lau are joint corresponding authors. Rynson W.H. Lau leads this project.

e.g., we can easily imagine the complete shape of an object partially-covered by rain as we have strong semantic contexts about objects. This suggests that semantic information is a strong cue for recovering object structures and colors.

Based on these two insights, we first propose a Parallel and Hierarchical context Pyramid (PHP) block to capture comprehensive spatial and hierarchical information for removing rain streaks of different sizes. We then propose a network, which is based on the PHP block, to adaptively remove rain streaks, recover object structures/colors, and enhance details. Finally, we propose a new dataset for training the network, and a novel loss function to enrich the semantics as well as preventing color distortion. As shown in Figure 1, our method produces a cleaner image, with better recovered image colors, structures and details. Extensive experiments on both synthesized and real-world data show that the proposed method produces state-of-the-art deraining performance, in terms of visual quality, quantitative accuracy, and running speed.

To summarize, this work has the following contributions:

- We propose a novel Parallel and Hierarchical context Pyramid (PHP) block to capture comprehensive spatial and hierarchical information for removing rain streaks of different sizes.
- We propose a novel neural network to adaptively remove rain streaks, recover object structures/colors, and enhance details.
- We propose a new dataset for training, and propose a novel loss function to enrich the semantics and to prevent color distortion.

II. Related Work

Single-image deraining methods typically model a rain image Y by superimposing a rain layer R (containing rain streaks) on a background layer X (rain-free image) as:

$$Y = X + R, \tag{1}$$

and formulate the deraining task as a signal separation problem. Kang *et al.* [20] propose to pre-filter the rain image into low- and high-frequency layers, and then separate rain streaks from the image details in the high-frequency layer via dictionary learning. Luo *et al.* [21] extend the additive rain model to a non-linear formulation by adding an element-wise multiplication of the rain and background layers, and model the rain layer via sparse coding. Li *et al.* [22] use Gaussian mixture models to model the rain and background layers separately. Zhu *et al.* [30] propose a joint optimization method to iteratively remove rain streaks from the background layer and suppress the background details in the rain layer, with the estimation of rain streak directions.

Recent deraining methods achieve state-of-the-art performances using deep learning. Fu *et al.* [24] pre-filter the input image into a base layer and a detail layer, and use a CNN to model rain streaks as high-frequency "residuals" between the detail layers of rain and rain-free images. Yang *et al.* [25] propose a deep recurrent convolutional network with dilations to jointly detect and remove rain streaks. Zhang and Patel [26] propose to classify rain density and use it to remove rain streaks via a multi-branch feature aggregation network. Yasarla and Patel [18] propose to learn confidence maps for separating rain streaks from high-frequency details. Wang *et al.* [15] propose a spatial attentive network to learn deraining features in a direction-aware manner. Wang *et al.* [31] introduce residual learning in the embedding space in order to learn entangled features for improving deraining performance. Wei *et al.* [32] propose to learn domain adapted rain features for deraining by using both synthesized and real rain images in a semi-supervised manner. Yasarla *et al.* [33] propose a semi-supervised rain removal method by formulating the joint learning from labeled synthetic dataset and unlabeled real rainy images in the Gaussian process. Wang *et al.* [34] propose a convolutional dictionary learning method to iteratively separate rain and clean background layers. Liang *et al.* [35] propose to model rain removal and detail recovery as two separate tasks, and design a network with two parallel branches to address these two tasks. To remove heavy rain streaks, some methods use deep recurrent convolutional networks [28], [16], [36], while others propose to synthesize both haze and rain streaks and learning depth-aware features for joint deraining and dehazing [27], [17], [37].

Despite all these efforts, as shown in Figure 1(b-e), existing methods still fail to separate dense rain streaks with varying intensity from the background and to restore image colors erased in heavy rain. In this paper, we propose to learn an intensity-aware deraining model by aggregating multi-scale spatial and hierarchical information from rain streaks of different sizes, and exploit semantic and color regularization to recover object structures, details and colors.

Multi-image deraining methods typically leverage inter-frame information to detect and remove rain streaks, based on the assumption that rain streaks do not occlude the same background location all the time. Garg and Nayar [7], [8] propose an appearance model to detect and remove rain, assuming that rain drops have equal falling speed. Chen and Hsu [38] propose a low-rank rain appearance model to remove rain streaks by considering spatial-temporal redundancy of rain patches. Santhaseelan and Asari [12] propose the phase congruency features to detect rain, and reconstruct the image using spatial and temporal neighborhood information. Jiang *et al.* [9] propose to jointly model the vertical smoothness/sparsity of rain and horizontal consistence of image textures. Ren *et al.* [11] propose a matrix decomposition based method to process heavy rain and moving objects in different layers. Wei *et al.* [13] propose a patch-based Gaussian mixture model to describe the variation of rain in a stochastic manner. Temporal redundancy has also been explored in [15], for semi-automatically generating one rain-free image from a sequence of rain frames, with human supervision. Apart from using the rich temporal information, Liu *et al.* [10] recently propose a deep recurrent convolutional network to explicitly classify rain degradation for deraining. Yang *et al.* [14] propose to exploit motion information to regularize inter-frame consistency for deraining via a two-step recurrent network. Zhang *et al.* [39] propose to exploit the different appearances of each rain streak in the input stereo images for rain removal. They also propose to incorporate scene semantics for stereo rain removal via the semantic segmentation task. Yang *et al.* [40] propose a self-learning method for video rain removal, which leverages temporal correlation/consistency of adjacent frames and optical flow information.

All the above works exploit temporal constraints to help derain. In contrast, we aim to address the more challenging single-image deraining problem in this work.

III. Proposed Method

A. Motivation

We address the single image deraining problem based on two observations. First, we note that the sizes of rain streaks can provide some valuable cues of their intensities, *e.g.*, rain streaks with a larger size tend to be closer to the camera, which implies that they tend to have higher intensity [29]. This motivates us to design the PHP block to aggregate multi-scale spatial information for detecting rain of different sizes, and multi-scale hierarchical information for removing them in an intensity-aware manner. Second, we observe that strong semantic contexts about objects enable human to not only recognize an object partially-covered by rain, but also to depict its complete contour. This suggests that semantic information is a strong cue for recovering object structures and colors.

Based on these two insights, we propose an intensity-aware deraining network that utilizes the PHP block to remove rain streaks of different sizes, and leverage semantic and color regularization to recover object structures, details and to prevent color distortion.

To introduce semantic regularization, we propose to explicitly model the pixel-wise semantics in deraining results via an auxiliary segmentation network. On the one hand, as the semantic segmentation task inherently demands for object recognition and boundary delineation, we can leverage its ability of intra-object regularization to help separate rain streaks from local semantically meaningful textures and inter-object regularization to preserve the image structures. In other words, we interpret the function of the semantic segmentation task as an attention mechanism to guide the deraining network to learn adaptive features depending on local semantics. On the other hand, we note that deep semantic segmentation networks rely heavily on their intrinsic invariance to feature deformations [41]. Such invariance encourages the network to learn an abstract feature representation for each object category, suggesting the network to prefer some shared information across objects within each category and suppress the information specific to individual objects in the category.

These observations further inspire us to separate the sub-task of instance-level detail recovery from other sub-tasks, *i.e.*, rain streaks removal, structure and color recovery. Hence, we apply the semantic and color regularization to the output of the first sub-network (X^c in Figure 2), and prepare ground truth images of low-frequency to supervise its learning process.

B. Network Details

Overview. The overall pipeline of the proposed network is illustrated in Figure 2. It consists of two sub-networks. The first sub-network is for removing rain streaks and recovering the obscured object structures and colors. The second sub-network is for recovering details.

PHP block. Rain streaks appear in the image with different sizes depending on their distances from the camera. To be able to capture rain streaks of different sizes, we propose to leverage multi-scale spatial information by constructing parallel pyramids of the receptive fields, to exploit rain streak appearances of different sizes. Although this is explored in previous deraining works by using parallel convolutions with different dilation rates [25] or kernel sizes [26], these methods typically neglect the negative effects, *e.g.*, change in intensity, caused by the different sizes of the rain streaks. In [25], parallel rain features are combined for deraining, which may lead to over-deraining as the network learns different rain intensities from the summed features. In [26], parallel rain features are concatenated to avoid over-estimating the rain intensities. However, as it does not consider the correlation among different rain intensities, they often fail to remove rain streaks with low intensities (usually of small sizes). Unlike these previous works, we further consider the correlations between rain sizes and their intensities to learn discriminative features for rain streak removal, by first constructing hierarchical pyramids for the captured rain streak features and then adaptively combining the hierarchical pyramids via a spatial reweighting mechanism. Constructing the separate hierarchical pyramids aims to gradually incorporate more contextual information for removing rain streaks of certain sizes, and the spatial reweighting mechanism is to reweight these deraining features in a global attentive manner.

As shown in Figure 3, we first feed the input rain features to a set of parallel dilated convolutions [42] to identify rain streaks of different sizes. We then gradually aggregate these rain features within the pyramid in a top-down manner, *i.e.*, we aggregate the deraining features to other features of larger receptive fields. This ensures that rain streaks of smaller sizes would not be missed, as their features would be aggregated to those of larger receptive fields. On the other hand, intensity of rain streaks with larger sizes would not be under-estimated, since their features from smaller receptive fields would be aggregated. These rain features are then concatenated and spatially adjusted via an attention function $f_{out} = f * Sigmoid(T(f))$, where f is the *weight features* and T represents two repeated groups of operations: depth-wise convolution and SELU activation [43]. It learns to pay self-attention to the correlation among these separately extracted rain features. We then form the hierarchical pyramid from these separately extracted rain features, by stacking the parallel pyramids with increasing dilation rates. This hierarchical pyramid augments the reweight features f_{out} by the contextual information from the increasingly abstract rain representations, producing better deraining features in the deeper layers.

Contrast-aware pooling. Rain pixels are usually brighter than their surrounding background pixels due to the reflection and refraction properties of rain drops. This brightness contrast motivates us to design the contrast-aware pooling layer to (1) detect rain streaks and (2) identify rain-free pixels. This allows the network to exploit the self-similarity property of a single input image to separately learn rain streak and

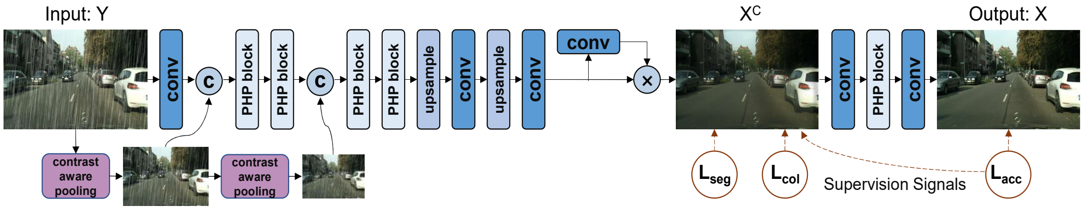


Fig. 2: Overview of the proposed model. Given a rain image Y, Our method first outputs a rain-free image X^c by using the parallel and hierarchical context pyramids to remove rain streaks of different sizes. This learning process is supervised by the background accuracy as well as semantic and color regularization. The rain-free image X^c is then fed into a refinement network to output final rain-free image X with details enhanced.

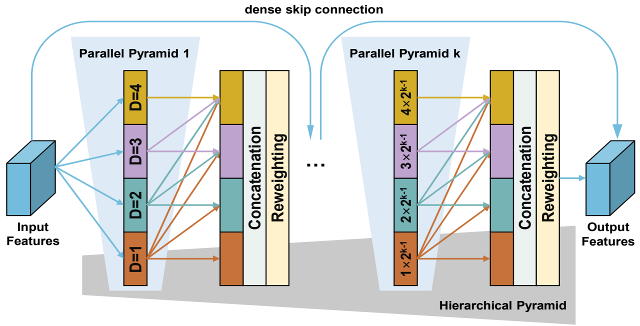


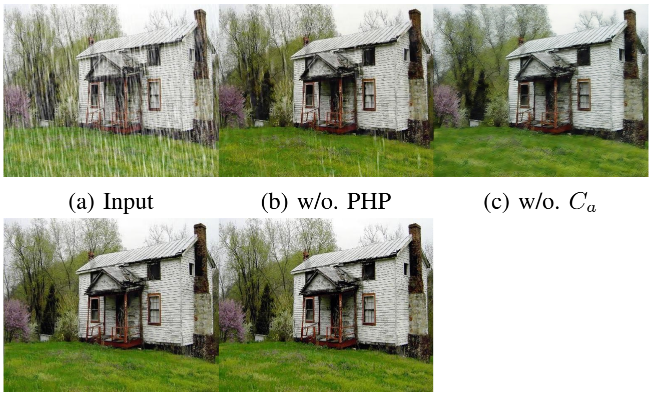
Fig. 3: The proposed PHP block. The input features are first encoded via a set of parallel convolutions of different dilation rates to form a *parallel pyramid* (light blue region), which output features are gradually aggregated in a top-down manner before concatenation and re-weighting. This top-down aggregation within one pyramid aims to enhance features of rain streaks in different sizes. We then stack a group of operations (*parallel pyramid formation* → *spatial aggregation* → *concatenation* → *reweighting*) with increasing dilation rates to form the *hierarchical pyramid* (gray region). k is set to 4 in our implementation.

background representations better. Specifically, the detection of rain streaks facilitates the rain removal process by providing guidance to the network to focus on modeling rain streak appearances of different sizes. It also facilitates the rain-free image reconstruction process, as it allows the network to exploit the spatial redundancy of background textures (*i.e.*, we may leverage patterns of non-rain background regions to help reconstruct the textures of rain-free pixels after deraining).

Specifically, given the input rain image Y, we first use two groups of dilated convolutions (with kernel size/dilation rate of 1/1 and 3/2), denoted as f_{d1} and f_{d2}, to extract features in different receptive fields. We then compute an attention map C_a between these two features as:

$$C_a = sigmoid(f_{d1}(Y) - f_{d2}(Y)). \tag{2}$$

C_a indicates the pixel-wise relative contrast information, where pixels of high contrasts can be considered as rain-affected. We then compute the inverse map $\overline{C}_a = 1 - C_a$ to select clean background pixels from Y as: $Y_c = \overline{C}_a \cdot Y$. Y_c is then average-pooled and concatenated with low-level features extracted by the first convolution layer (Figure 2), for further learning of deraining features. A visual example is shown in Figure 4, which illustrates the effectiveness of the PHP block as well as the contrast-aware pooling layer.


(a) Input (b) w/o. PHP (c) w/o. C_a

(d) Ours (e) Ground truth

Fig. 4: Visualization of the ablation study on the proposed network, which shows the advantages of the proposed PHP block on detecting rain streaks and contrast-aware pooling on preserving details while deraining.

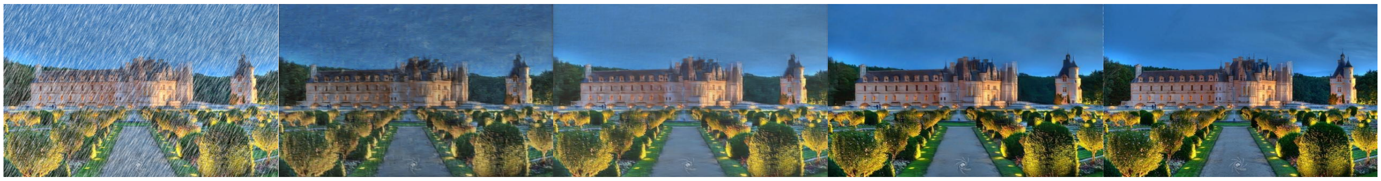
C. Loss Function

To enrich semantics and prevent color distortion, we propose a novel loss function consisting of three loss terms, *i.e.*, the *background accuracy*, *color alignment* and *semantic regularization* terms, for training the proposed network.

Background accuracy term. To encourage the first sub-network to focus on removing rain streaks and reconstructing background without considering the confusing image details, we prepare the corresponding ground truth, denoted as X_g^{gt}, by using the Gaussian filter to filter out image details while maintaining the main structures and contents of the ground truth image. We adopt the approximated L_1 distance [44] to measure the *background accuracy* for both sub-networks (see Figure 2), as L_2 may have the blur effect.

$$L_{acc} = \sqrt{(X^c - X_g^{gt})^2 + \epsilon^2} + \sqrt{(X - X^{gt})^2 + \epsilon^2}, \tag{3}$$

where X^c, X, X_g^{gt} and X^{gt} are the deraining results from the first and second sub-networks, the ground truth clean images



(a) Input (b) L_{acc} (c) $L_{acc} + L_{seg}$ (d) $L_{acc} + L_{seg} + L_{col}$ (e) Ground truth

Fig. 5: Visualization of the proposed loss function. It verifies the effectiveness of each term on removing rain streaks, cleaning background and restoring colors.

for the first and second sub-networks, respectively. ϵ is a small constant value to prevent the loss being zero.

Color alignment term. The non-uniformity of rain intensity can easily cause over-/under-deraining, which cannot be completely addressed by the L_1 term. To maintain background color stability, we propose to use the cosine similarity for background color alignment. The cosine similarity measures the similarity of two vectors by computing the angle between them as:

$$L_{col} = 1 - \frac{1}{N}\sum_{i=1}^{N}\frac{X_i \cdot X_i^{gt}}{\left\|X_i\right\|_2 \left\|X_i^{gt}\right\|_2}, \tag{4}$$

where X and X^{gt} are the deraining result and the ground truth clean image. Since the per-pixel L_1 loss can be dominated by the channels with large difference values, background colors can be easily distorted if the rain intensity is misjudged. In contrast, L_{col} forces the predicted pixels to point to the same directions as the ground truth pixels in the RGB space, without considering the magnitude. Hence, it helps provide color stability to the deraining process.

Semantic regularization term. We use the standard cross entropy loss (denoted as L_{seg}) for semantic regularization. Additionally, to encourage the second network to focus on recovering high-frequency details, we decouple the learning process of these two sub-networks: we prepare the corresponding ground truth images (denoted as X_f^{gt}) for supervising the background accuracy of the first sub-network by using a Gaussian low-pass filter to filter out the high-frequency details of the ground truth images X^{gt}.

The final loss function used in the proposed model is then:

$$\begin{aligned} Loss = {} & \lambda_1 L_{acc}(X^c, X_f^{gt}) + \lambda_2 L_{acc}(X, X^{gt}) \\ & + \lambda_3 L_{col}(X^c, X^{gt}) + \lambda_4 L_{seg}(\Phi(X^c), S^{gt}), \end{aligned} \tag{5}$$

where λ_1, λ_2, λ_3, and λ_4 are constant values to balance the loss terms. $\Phi(\cdot)$ and S^{gt} are the semantic segmentation network and ground truth semantic map, respectively.

Sensitivity on hyper-parameters. Our method is robust to the first two balancing hyper-parameters (*i.e.*, λ_1 and λ_2, which are both set to 1) in Eq. 5. We have also followed the classic PSPNet [45] to set the λ_1 to 0.4, but do not observe obvious performance changes. This is because our model is shallower than PSPNet, and the gradients could be back propagated through the whole network well. For the other two balancing hyper-parameters (*i.e.*, λ_3 and λ_4) in Eq. 5, we empirically set them to 0.1 and 0.05, to make these four loss terms to have the same magnitude after one training epoch. We have also explored automatical learning of the weights of these loss terms by modeling the task uncertainties as introduced in [46]. However, such a strategy does not work in our cases (the network cannot converge in training), due to the differences between their task and ours. Their method explores three highly-related tasks, *i.e.*, semantic segmentation, instance segmentation and depth prediction, while ours focuses on using the semantic segmentation task to provide pixel-wise semantics for rain removal. Figure 5 visualizes the effectiveness of the proposed loss function, where incorporating more terms (*i.e.*, L_{acc}, L_{seg}, L_{col}) help remove rain streaks, recover background details and colors.

D. Training Details

Dataset. Our motivations of constructing the dataset are: (1) to provide careful control of rain streak sizes, (2) to provide pixel-wise semantic information, and (3) to supervise our network for removing rain streaks in a low-to-high frequency manner. Previous works do not consider these three factors together. In [24] and [47], only rain/rain-free image pairs are provided. Their datasets focus on involving different background (*i.e.*, rain-free) scenes to cover real-world scenes. In [27], a depth-related rain model is leveraged to synthesize both rain and fog effects. It assumes the co-existences of rain and fog, and may cause over-deraining in the non-fog regions. In [15], rain-free images are derived from a sequence of real rain images. However, their proposed dataset does not consider dense rain scenes, as their rain-free image generation method would fail. Other works construct rain datasets with additional rain streak binary masks [25] and rain density labels [26]. Most recently, [39] provides semantic segmentation labels in their stereo rain dataset. However, their rain synthesis model does not consider any rain characteristics.

We synthesize the rain images based on the Cityscapes dataset [48], which has about 3K high-resolution urban street scene images with fine pixel-wise annotations on 19 classes of common objects. The high image resolution also allows our model to learn fine details of individual instances. We use the general rain formation model (Eq. 1) and Adobe Photoshop [49] to synthesize the rain images. Assuming the negative y-axis to be $0'$, we synthesize the rain streaks of 7 directions (*i.e.*, $0'$, $\pm 15'$, $\pm 30'$, $\pm 40'$). We use Gaussian blur kernel to blur the randomly generated noise to synthesize rain streaks. We manipulate the filter radius ([0.2, 1.0], at interval of 0.2) to control the blurred rain streak sizes. To decompose the ground truth images into low and high frequency components, we first

Method	Cityscapes dataset		Rain800 dataset		DIDMDN dataset		DDN dataset		GPU Time
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	SECOND↓
LP [22]	20.61	0.6927	21.47	0.7825	19.95	0.7910	18.40	0.4577	–
DDN [24]	24.32	0.7414	21.33	0.7918	25.49	0.8491	26.21	0.8401	0.091
JORDER [25]	23.30	0.7261	21.08	0.8053	24.32	0.8022	22.24	0.7741	0.178
RESCAN [28]	24.30	0.7862	22.57	0.8121	18.29	0.7543	26.45	0.8458	0.365
DIDMDN [26]	25.63	0.8488	21.42	0.8126	26.20	0.8978	26.35	0.8384	0.051
HeavyRain [17]	23.14	0.7174	17.59	0.6825	20.12	0.7590	24.10	0.8313	0.150
SPANet [15]	25.45	0.8371	20.47	0.7569	22.07	0.7914	26.61	0.8517	0.459
PReNet [16]	25.24	0.8218	20.49	0.7855	24.72	0.8581	27.46	0.8612	0.101
UMRL [18]	26.29	0.8613	22.55	0.8323	26.96	0.8951	27.53	0.8631	0.525
MSPFN [36]	26.47	0.8845	23.18	0.8415	27.45	**0.9018**	**28.14**	0.8732	0.308
Ours	**26.53**	**0.8975**	**23.29**	**0.8521**	**27.59**	0.9015	27.95	**0.8825**	**0.008**

TABLE I: Quantitative comparison between the proposed method and 10 state-of-the-art single-image deraining methods on four datasets (*i.e.*, Cityscapes, Rain800, DIDMDN and DDN) and three metrics (*i.e.*, PSNR, SSIM and inference time). Top two performances are marked in **bold** and underlined.

apply four Gaussian filters with kernel sizes of 3×3, 5×5, 7×7 and 9×9, to generate four versions of low-frequency images. We then ask users to judge in which version they could tell the main contents of the filtered image well, while not seeing the image details. We try different kernel sizes as high-resolution images typically contain multi-scale high-frequency details.

Implementation details. The proposed model is implemented on the Pytorch framework [50], and tested on a PC with an i7 4GHz CPU and a GTX 1080Ti GPU. As we train our model from scratch, the network parameters are initialized randomly, and standard augmentation strategies, *i.e.*, scaling, cropping, flipping and rotation, are adopted. Batch size is set to 1. For loss minimization, we adopt the ADAM optimizer [51] for 250 epochs, with an initial learning rate of $3e^{-4}$ and divided by 10 at the 150^{th} and 200^{th} epochs. We use a 3×3 kernel size for all the convolutional operations except for the last convolution layer, which has a 1×1 kernel size. We share the same network structure of the segmentation network $\Phi(\cdot)$ to our deraining network, to avoid being biased to any existing segmentation method.

IV. Experiments

A. Experimental Setting

Evaluation methods. We compare the proposed model to 10 state-of-the-art single image deraining methods: LP [22], DDN [24], JORDER [25], DIDMDN [26], RESCAN [28], SPANet [15], PReNet [16], HeavyRain [17], UMRL [18] and MSPFN [36]. Among them, LP [22] relies on handcrafted features to remove rain streaks, while the others learn deep features for rain removal. For a fair comparison on the proposed dataset, we fine-tune all deep learning based methods from their original pre-trained models.

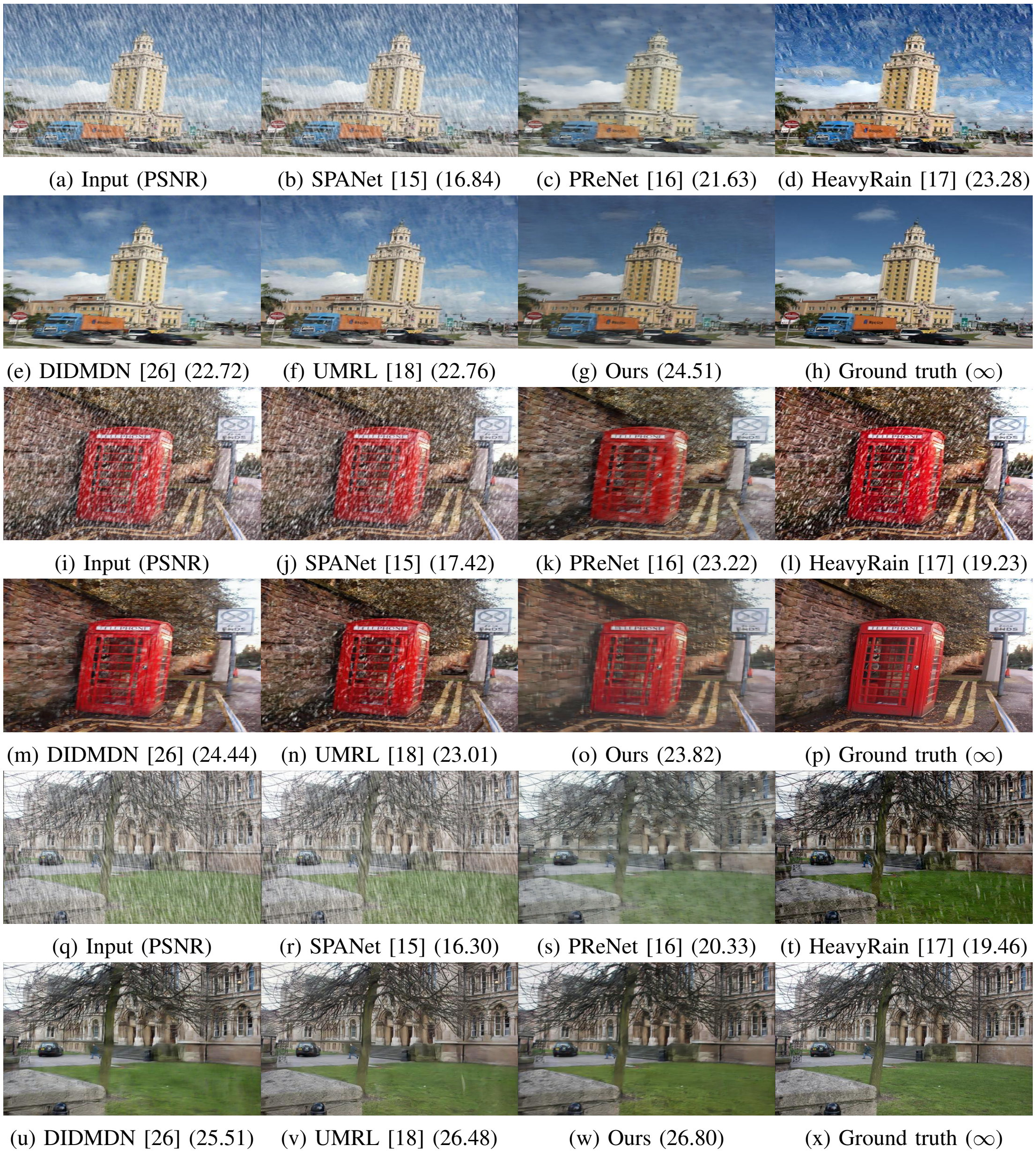
Evaluation datasets and metrics. We quantitatively evaluate our model on four datasets: Cityscapes validation set (500 images), Rain800 testing set [47] (100 images), DIDMDN testing set [26] (1200 images), and DDN [24] (1000 images). These datasets are with diverse rain sizes, *e.g.*, the Rain800 dataset has shorter and wider rain streaks, while the DDN dataset has longer rain streaks. We also provide qualitative evaluation on real-world rain images collected by previous works [52], [53] and [26]. We use two popular image quality metrics, PSNR and SSIM [54], for perceptual evaluation.

B. Comparison with State-of-the-arts

Evaluation on synthesized rain images. Table I reports the quantitative results on the proposed dataset and three existing datasets. For a fair comparison on the proposed Cityscapes test set, we fine-tune other deep learning based methods on our training set. We can see that our method performs consistently better than other methods on both PSNR and SSIM metrics. This shows that, by learning comprehensive spatial and hierarchical information, our method performs better in removing rain streaks with varying sizes and intensities. Further, by incorporating semantic and color regularization in the rain-free image reconstruction process, our method learns robust deraining features across existing datasets. Figures 6 and 7 show the comparisons on some visual examples with different rain characteristics (*e.g.*, directions and sizes) from two existing datasets. In each group of images, the first image shows the input rain images and the next six images show the derained images produced by five latest state-of-the-arts: SPANet [15], PReNet [16], HeavyRain [17], DIDMDN [26], UMRL [18] and ours. While existing methods either fail to remove rain streaks (*e.g.*, Figure 6(b,d,f) and Figure 7(j,k,l)) or generate blurry background images (*e.g.*, in Figure 6(c,e,s,u) and Figure 7(c,e,f,m)), our method can successfully remove rain streaks and produce visually pleasing rain-free images.

Evaluation on real-world rain images. We show a variety of real-world rain images for qualitative comparison in Figures 8 and 9. The first row in Figure 8 shows one image with broad rain streaks that existing methods fail, while our method can detect and remove them. The next two rows provide challenging examples with dense rain streaks laying on complex background objects. While existing methods tend to leave rain streaks unremoved or generate blurry background, our method can remove rain streaks located nearby as well as at distant. The last two rows show rain images with people. Exiting methods fail to remove rain streaks nor preserve the background well, while our method is more sensitive to rain streaks of different sizes and can produce better rain-free images. These results demonstrate the effectiveness of the proposed PHP block on capturing rain streaks of varying sizes and removing them in intensity-aware manner.

Although heavy rain can erase the image colors, our method can recover them well (see Figure 9 for illustration). Note that



(a) Input (PSNR) (b) SPANet [15] (16.84) (c) PReNet [16] (21.63) (d) HeavyRain [17] (23.28)

(e) DIDMDN [26] (22.72) (f) UMRL [18] (22.76) (g) Ours (24.51) (h) Ground truth (∞)

(i) Input (PSNR) (j) SPANet [15] (17.42) (k) PReNet [16] (23.22) (l) HeavyRain [17] (19.23)

(m) DIDMDN [26] (24.44) (n) UMRL [18] (23.01) (o) Ours (23.82) (p) Ground truth (∞)

(q) Input (PSNR) (r) SPANet [15] (16.30) (s) PReNet [16] (20.33) (t) HeavyRain [17] (19.46)

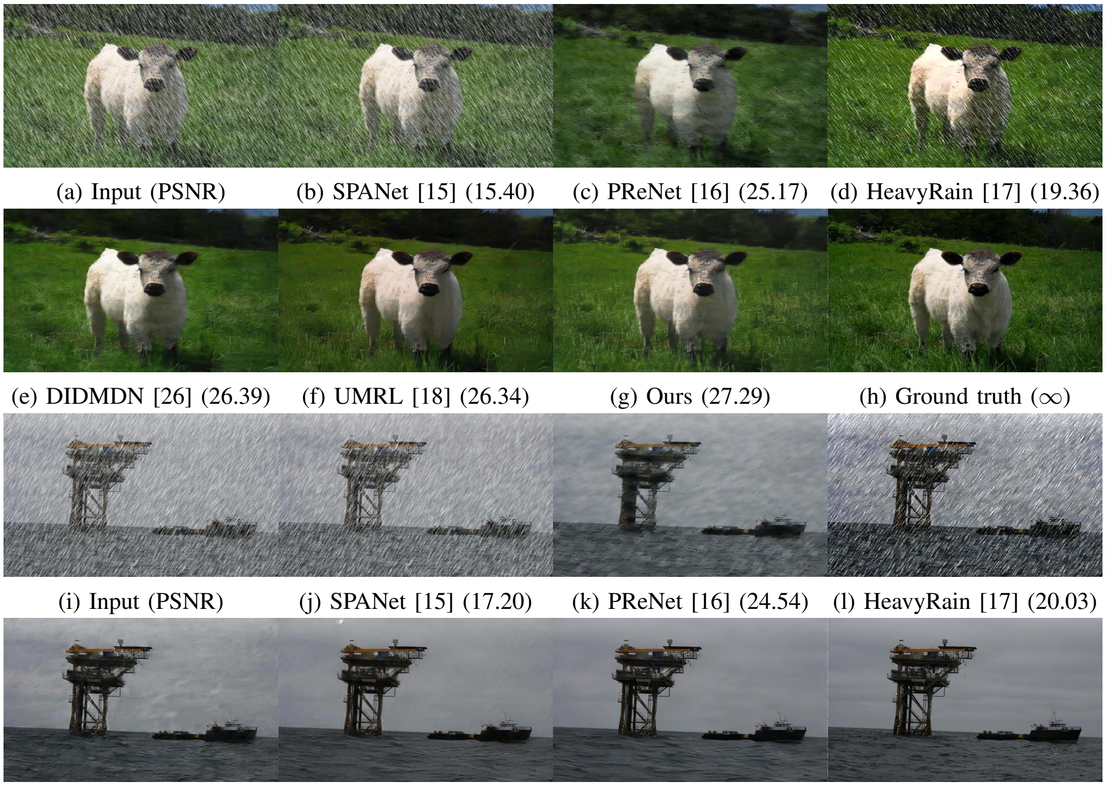
(u) DIDMDN [26] (25.51) (v) UMRL [18] (26.48) (w) Ours (26.80) (x) Ground truth (∞)

Fig. 6: Visual comparison on synthesized rain images of Rain800 dataset [47]. Our method can produce better derained images with background structures, details and colors recovered.

HeavyRain [17] incorporates dehazing priors in their model for heavy rain image restoration, and can produce rain-free images of high contrasts. However, it tends to enhance both rain streaks and image details. In contrast, our method models object semantics and learns to align object colors in heavy rain to their original colors, resulting in better derained images.

Runtime evaluation. When the deraining task is served as a pre-processing step to its subsequent task, fast inference is necessary since practical outdoor applications usually demand for real-time performances ($>$ 24 fps). The last column of Table I shows the GPU time costs of 8 deep learning based state-of-the-art deraining methods and ours. Note that LP [22] is CPU-based and requires around $60s$ for a 512×512 image. We can see that our method is the fastest deraining method, with consistently better deraining performance, which again demonstrates the superiority of the proposed method.

Dataset evaluation. We further evaluate the generalization capacity of the proposed dataset over previous datasets in Table II. We train the proposed network on our proposed intensity-aware dataset (denoted as "Ours-Intensity" in Table II), the density-aware dataset [26] (denoted as "Ours-Density"), the depth-aware dataset [27] (denoted as "Ours-Depth"), and the rain spatial distribution-aware dataset (denoted as "Ours-Spatial") [15], and test these four models on the Rain800 dataset [47]. We can see that the model trained on our dataset performs better than those trained on existing datasets. This shows that learning the rain intensity features is more effective for rain removal, compared to learning to



(a) Input (PSNR) (b) SPANet [15] (15.40) (c) PReNet [16] (25.17) (d) HeavyRain [17] (19.36)

(e) DIDMDN [26] (26.39) (f) UMRL [18] (26.34) (g) Ours (27.29) (h) Ground truth (∞)

(i) Input (PSNR) (j) SPANet [15] (17.20) (k) PReNet [16] (24.54) (l) HeavyRain [17] (20.03)

(m) DIDMDN [26] (27.83) (n) UMRL [18] (30.34) (o) Ours (31.22) (p) Ground truth (∞)

Fig. 7: Visual comparison on synthesized rain images of DIDMDN dataset [26]. Our method can produce better derained images with background structures, details and colors recovered.

Method	PSNR↑	SSIM↑
Ours-Density	20.51	0.8143
Ours-Spatial	19.08	0.7741
Ours-Depth	23.15	0.8211
Ours-Intensity	**23.29**	**0.8521**

TABLE II: Proposed dataset evaluation. We train our network on the Density-aware dataset [26], rain Spatial distribution-aware dataset [15], depth-aware dataset [27], and the proposed Intensity-aware dataset, and test these four models on the Rain800 dataset [47].

Method	PSNR↑	SSIM↑	Acc.↑	mIoU↑
L_{acc}	25.59	0.8419	0.597	0.155
$L_{acc} + L_{col}$	25.64	0.8572	0.671	0.206
$L_{acc} + L_{col} + L_{seg}$	**26.53**	**0.8975**	**0.701**	**0.288**

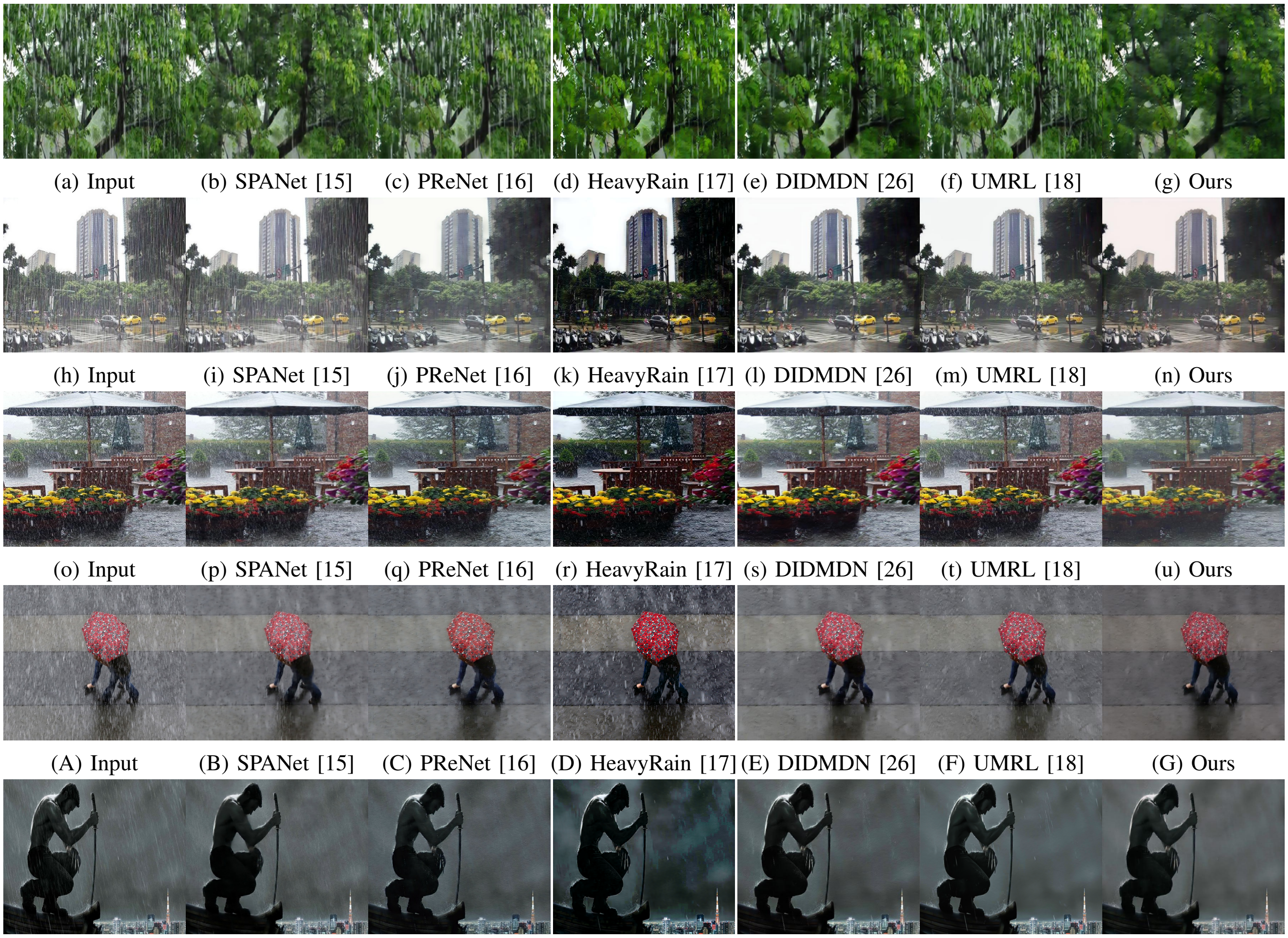
TABLE III: Ablation study on the Cityscapes dataset. Performance is being continuously improved by incorporating more terms.

encode other factors (*i.e.*, rain density and spatial distributions, and scene depth).

C. Internal Analysis and Discussion

Loss function study. Table III studies the contribution of each term in the proposed loss function (Eq. 5). While the model performance is being continuously improved by incorporating more terms, introducing L_{col} obtains a notable performance gain in terms of per-pixel accuracy. This shows that the color alignment term helps correct the image details. Besides, introducing the semantic regularization term L_{seg} mainly improve mIoU, due to the enriched semantics. One visual example is shown in Figure 5, where we can see that using $L_{acc} + L_{seg}$ helps clear the rain in the sky (Figure 5(c)), due to the semantic contextual information introduced by L_{seg}. In addition, object colors are restored by using $L_{acc} + L_{seg} + L_{col}$ (Figure 5(d)), due to the magnitude-invariant color regularization L_{col}. We further evaluate the effectiveness of proposed color alignment term L_{col} on removing rain streaks from gray-scale images. Figure 10 shows two visual examples. While we can see that both models can remove rain streaks, results of removing L_{col} tend to be darker and greenish, and adding L_{col} produces more accurate and visually pleasing rain-free images. Note that although the input images are in gray-scale (*i.e.*, single channel), our method produces three-channel results. Without the color regularization provided by L_{col} during training, the model may not learn the background intensities well and the results would be improperly painted with colors from the training dataset.

Network design study. To verify the effectiveness of our network design, we then study thirteen baselines as listed in Table IV. Particularly, "Standard Convolution" represents a normal densely-connected network without considering both spatial and hierarchical information aggregation. "Parallel



(a) Input (b) SPANet [15] (c) PReNet [16] (d) HeavyRain [17] (e) DIDMDN [26] (f) UMRL [18] (g) Ours

(h) Input (i) SPANet [15] (j) PReNet [16] (k) HeavyRain [17] (l) DIDMDN [26] (m) UMRL [18] (n) Ours

(o) Input (p) SPANet [15] (q) PReNet [16] (r) HeavyRain [17] (s) DIDMDN [26] (t) UMRL [18] (u) Ours

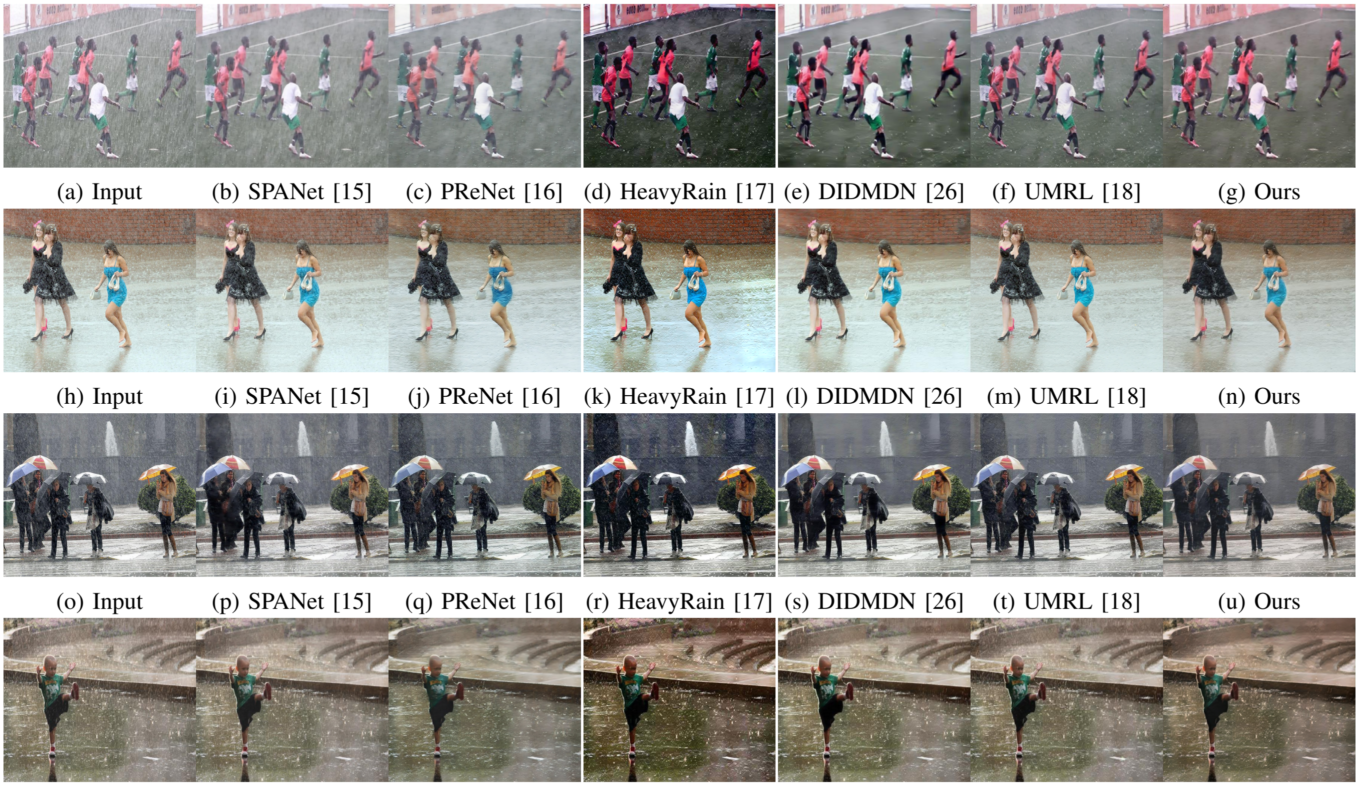
(A) Input (B) SPANet [15] (C) PReNet [16] (D) HeavyRain [17] (E) DIDMDN [26] (F) UMRL [18] (G) Ours

(H) Input (I) SPANet [15] (J) PReNet [16] (K) HeavyRain [17] (L) DIDMDN [26] (M) UMRL [18] (N) Ours

Fig. 8: Visual comparison on real-world rain images where rain streaks of varying characteristics appear. Our method produces better deraining results with cleaner backgrounds, by learning intensity-aware and semantic contextual features.

Only" is a baseline that only considers spatial-varying contextual information for detecting rain streaks. This baseline corresponds to the exploration of spatial contexts in previous deraining methods [25], [26]. "Hierarchical Only" is a baseline that only uses hierarchical dilation convolutions, which aim to aggregate contextual information along the network depth in the feature space. We also design a baseline that combines both spatial and hierarchical context aggregations, by simply concatenating features, indicated as "PHP-concatenation". We further replace the proposed PHP block with the non-local block [55] (indicated as "PHP → Non-local"). To study the contrast-aware information on the deraining performance, we design another baseline by removing the contrast-aware pooling layer (indicated as "w/o. Contrast-aware pooling"), and a further baseline by replacing the proposed Contrast-aware pooling with the Max-pooling (indicated as "Contrast-aware pooling → Max-pooling"). We then study the necessity of preparing different ground truth images for our two sub-networks, by replacing the pre-filtered ground truth images with the original ground truth images, denoted as "$X_g^{gt} \rightarrow X^{gt}$", and by removing the second sub-network (indicated as "$w/o.\ X$"). We also replace the first sub-network with one existing deraining method SPANet [15], denoted as "$X^c \rightarrow SPANet$ [15]". We choose this method as it is the best-performing one of all single-stage deraining methods, according to Table I. To further understand the effectiveness of the proposed method, we remove training images of different rain sizes. We consider rain streaks generated with a filter radius of 0.2 as small size, filter radii of 0.4 and 0.6 as medium size, and the rest are of large size (indicated as "$w/o.\ S$", "$w/o.\ M$" and "$w/o.\ L$").

The results from Table IV generally verify our design choices. It shows that contextual information of different spatial scales and level of abstraction both help the model learn discriminative deraining features (by comparing the results in the 2nd or 3rd row to the 1st row), and a simple combination would combine these advantages (comparing the result in the 4th row to that in the 1st row). In addition, our PHP block can further boost the deraining performance, by applying gradual aggregations on features within the same parallel pyramid, and learning the spatial reweighting function (see last row). We also show that the proposed PHP block performs better than



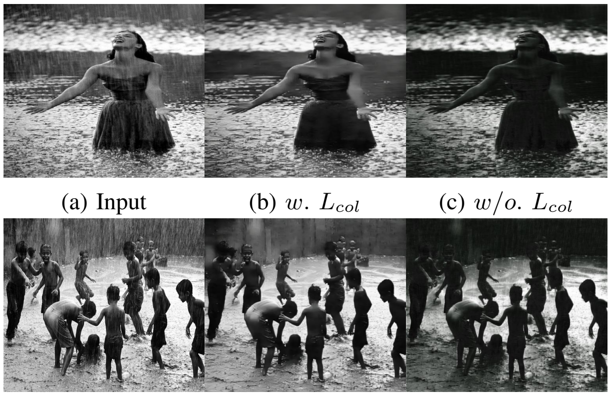
(a) Input (b) SPANet [15] (c) PReNet [16] (d) HeavyRain [17] (e) DIDMDN [26] (f) UMRL [18] (g) Ours

(h) Input (i) SPANet [15] (j) PReNet [16] (k) HeavyRain [17] (l) DIDMDN [26] (m) UMRL [18] (n) Ours

(o) Input (p) SPANet [15] (q) PReNet [16] (r) HeavyRain [17] (s) DIDMDN [26] (t) UMRL [18] (u) Ours

(A) Input (B) SPANet [15] (C) PReNet [16] (D) HeavyRain [17] (E) DIDMDN [26] (F) UMRL [18] (G) Ours

Fig. 9: Visual comparison on real-world rain images where heavy rain attenuates image colors. Our method not only learns to remove rain streaks but also restores colors.



(a) Input (b) $w.\ L_{col}$ (c) $w/o.\ L_{col}$

(d) Input (e) $w.\ L_{col}$ (f) $w/o.\ L_{col}$

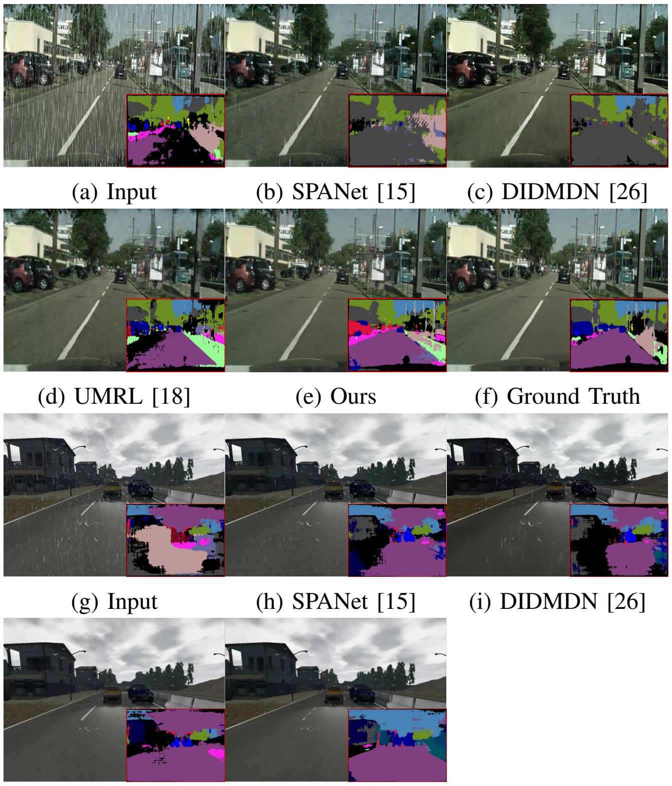
Fig. 10: Ablation study of the proposed color alignment term on real-world gray-scale rain images.

Method	PSNR↑	SSIM↑
Standard Convolution	20.41	0.7545
Parallel Only	22.61	0.7911
Hierarchical Only	24.34	0.8207
PHP-concatenation	25.17	0.8710
PHP → Non-local [55]	24.91	0.8431
w/o. Contrast-aware pooling	25.47	0.8817
Contrast-aware pooling → Max-pooling	25.10	0.8563
$X_g^{gt} \rightarrow X^{gt}$	26.23	0.8906
$w/o.\ X$	20.19	0.7126
$X^c \rightarrow SPANet$ [15]	25.87	0.8620
$w/o.\ S$	25.82	0.8841
$w/o.\ M$	24.71	0.8701
$w/o.\ L$	25.58	0.8820
Ours	**26.53**	**0.8975**

TABLE IV: Evaluation of the method design with 13 baselines.

the well-known Non-local block. This is because the Non-local operation tends to ignore the spatial property of rain streaks (*e.g.*, vertical smoothness) as it always relates the current pixel to all pixels. Note that the performance drops due to removing the contrast-aware pooling layer and replacing it with max-poolling (the 6th and 7th rows) verify the necessity of incorporating contrast-aware information for removing sharp rain streaks. We can see that applying the pre-filtered ground truth images for supervising the first sub-network works better than using the original ground truth images for it, since this avoids the first sub-network from learning confusing features of image high-frequency details (the 8th row). Removing the second sub-network does not make sense, as high-frequency image details cannot be recovered without it (the 9th row). On the contrary, high-frequency information learned in the second sub-network can further boost the performance of SPANet [15] from PSNR/SSIM: $25.45/0.8371$ (Table I) to PSNR/SSIM: $25.87/0.8620$ (the 10th row). Finally, we can see that learning comprehensive deraining features of different rain streak sizes boost the deraining performance.

Semantic study. We are also interested in knowing whether



(a) Input (b) SPANet [15] (c) DIDMDN [26]

(d) UMRL [18] (e) Ours (f) Ground Truth

(g) Input (h) SPANet [15] (i) DIDMDN [26]

(j) UMRL [18] (k) Ours

Fig. 11: Semantic evaluation on derained results using the Cityscapes dataset (a-f) and the SYNTHIA dataset (g-k).

the "semantic structures" introduced by semantic regularization can be generalized to existing semantic segmentation networks. To this end, we feed our deraining results into pre-trained semantic segmentation networks and evaluate their semantic segmentation performances (measured in terms of intersection over union (IoU) and overall pixel accuracy). We use two existing semantic segmentation models: ESPNet [56] and PSPNet [45]. While ESPNet is a light-weight model that achieves a good balance between prediction accuracy and real-time inference, PSPNet has a heavier architecture with a state-of-the-art segmentation performance on various metrics. We compare our method to 3 best performing deraining methods (according to Table I): SPANet [15], DIDMDN [26] and UMRL [18].

Table VI reports the segmentation performance on the Cityscapes dataset measured by mIOU and overall pixel accuracy. While the ill-corrected image structures and details from existing deraining methods severely deteriorate the segmentation performance, our method outperforms existing deraining methods with a notable margin. This shows that our method can successfully inject the semantic contexts into the deraining process in the image space, so that more image structures and details can be recovered. Note that as we do not optimize our deraining process for one specific model (ESPNet or PSPNet), our method is robust to existing segmentation models of different design philosophies.

Method	PSNR↑	SSIM↑	Acc.↑	mIoU↑
(a) shared obj. pred.	17.74	0.7211	-	-
(b) cascaded obj. pred.	20.69	0.7007	-	-
(c) shared sem. seg.	24.32	0.8336	**0.730**	**0.301**
Ours	**26.53**	**0.8975**	0.701	0.288

TABLE V: Evaluation on semantic regularization choices. We compare our semantic regularization choice against three methods: (a) using object prediction in a shared manner, (b) using object prediction in a cascaded manner, and (c) using semantic segmentation in a shared manner. Ours yields better deraining performance.

We further finetune the segmentation network [56] on existing deraining results (as well as ours) (right part of Table VI). It is worth noting that the segmentation network [56] finetuned on existing derained results does not show advantages over the one finetuned on rain images. It demonstrates the importance of fine structures and details to the semantic segmentation task, and verifies the necessity of preserving them in the deraining task. Figure 11 shows two examples, of which (a) is a rain image from Cityscapes dataset and (g) is a rain image from the SYNTHIA dataset [57], which does not provide the corresponding rain-free image. Comparing the results of these two rain images, it demonstrates the good generalization ability of our method on preserving image structures and details when deraining images from other domain.

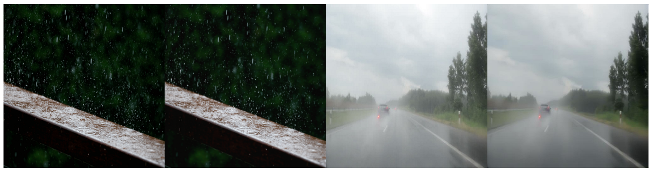
Semantic regularization choices. Lastly, we study other possible methods for providing semantic regularization on deraining process. We have considered three candidate strategies: (a) performing object prediction (as in [58]) and deraining using a shared encoder and separate decoders; (b) performing deraining and then object detection in a cascaded manner, and (c) performing deraining and semantic segmentation using a shared encoder and separate decoders. Involving the object prediction task, however, does not boost the deraining performance. This is mainly because the object prediction task only concerns about the main discriminative features for object classification. It does not aim at recovering the object structures and boundaries. See Table V first two rows. Involving the semantic segmentation task as in strategy (c) produces a better segmentation performance but poorer deraining performance (third row in Table V). This is because the semantic segmentation task tends to suppress the features that are specific to individual objects while preferring the features that are common to all objects in the category. It helps regularize the object structures and boundaries but negatively affects image texture/color reconstructions. This experiment demonstrates the effectiveness of leveraging semantic segmentation to modeling semantic contextual information for deraining.

V. Conclusion and Future Work

In this paper, we have presented a new method for single-image deraining. Specifically, we leverage a new PHP block to capture comprehensive spatial and hierarchical information for removing rain streaks of different sizes. We then design a new network to first remove rain streaks, then recover objects structures/colors, and finally recover details. To train

Method	ESPNet [56]		PSPNet [45]		Finetuned [56]	
	Acc.↑	mIoU↑	Acc.↑	mIoU↑	Acc.↑	mIoU↑
Rain	0.262	0.061	0.115	0.436	0.625	0.225
Clean	0.905	0.351	0.939	0.692	–	–
SPANet [15]	0.515	0.147	0.658	0.225	0.633	0.257
DIDMDN [26]	0.544	0.164	0.706	0.307	0.629	0.231
UMRL [18]	0.602	0.203	0.739	0.346	0.641	0.262
Ours	**0.701**	**0.288**	**0.869**	**0.472**	**0.726**	**0.311**

TABLE VI: Semantic evaluation in terms of overall pixel accuracy and mIOU, tested using ESPNet [56] (left), PSPNet [45] (middle), and the finetuned ESPNet [56] (right) using rain images and different derained results.



(a) Input (b) Ours (c) Input (d) Ours

Fig. 12: Failure cases. Our method may fail in scenes where limited semantics can be captured for providing the contextual information to help the deraining task (b). Our method may also fail in extreme dense rain scenes (d), where rain streaks are accumulated into haze and completely obscure the background contents.

our model, we have prepared a dataset and design a new loss function to introduce semantic and color regularization for deraining. We have conducted extensive experiments to demonstrate the superiority of the proposed method over state-of-the-art deraining methods on both synthesized and real-world data, in terms of visual quality, quantitative accuracy, and running speed. We will release our model and our dataset.

Our method does have some limitations, as show in Figure 12. As our method exploits semantic contextual information for separating rain streaks from the background, it may fail in scenes that contain limited semantic information, as demonstrated in Figure 12(b). Another limitation is that when objects are far away from the camera, and the scene has extremely dense rain streaks, the haze effects may completely obscure the image contents. In this case, our method may fail to recover the objects, as shown in the region around the car.

We observe that in real driving scenes under rain conditions, rain streaks can be blurred by fast motion. This phenomenon is fundamentally different from that of the normal rain scenes. A possible future work may be to study the feasibility of combining a haze removal algorithm with a deraining algorithm to address this problem.

VI. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant 61632006, 61972067), the Innovation Technology Funding of Dalian (Project No. 2018J11CY010, 2020JJ26GX036), and a General Research Fund from RGC of Hong Kong SAR (Ref.: 11205620).

References

[1] X. Yang, H. Mei, K. Xu, X. Wei, B. Yin, and R. W. Lau, "Where is my mirror?" in *ICCV*, 2019.

[2] X. Tian, K. Xu, X. Yang, B. Yin, and R. W. Lau, "Weakly-supervised salient instance detection," *BMVC*, 2020.

[3] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M.-H. Yang, "Crest: Convolutional residual learning for visual tracking," in *ICCV*, 2017.

[4] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *CVPR*, 2019.

[5] X. Yang, K. Xu, S. Chen, S. He, B. Y. Yin, and R. Lau, "Active matting," in *NIPS*, 2018.

[6] X. Yang, K. Xu, Y. Song, Q. Zhang, X. Wei, and R. W. Lau, "Image correction via deep reciprocating hdr transformation," in *CVPR*, 2018.

[7] K. Garg and S. Nayar, "Detection and removal of rain from videos," in *CVPR*, 2004.

[8] ——, "When does a camera see rain?" in *ICCV*, 2005.

[9] T. Jiang, T. Huang, X. Zhao, L. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *CVPR*, 2017.

[10] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *CVPR*, 2018.

[11] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *CVPR*, 2017.

[12] V. Santhaseelan and V. Asari, "Utilizing local phase information to remove rain from video," *IJCV*, 2015.

[13] 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 *ICCV*, 2017.

[14] W. Yang, J. Liu, and J. Feng, "Frame-consistent recurrent video deraining with dual-level flow," in *CVPR*, 2019.

[15] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *CVPR*, 2019.

[16] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *CVPR*, 2019.

[17] R. Li, L.-F. Cheong, and R. T. Tan, "Heavy rain image restoration: Integrating physics model and conditional adversarial learning," in *CVPR*, 2019.

[18] R. Yasarla and V. M. Patel, "Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining," in *CVPR*, 2019.

[19] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *ICCV*, 2017.

[20] L. Kang, C. Lin, and Y. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE TIP*, 2012.

[21] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *ICCV*, 2015.

[22] Y. Li, R. Tan, X. Guo, J. Lu, and M. Brown, "Rain streak removal using layer priors," in *CVPR*, 2016.

[23] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain streaks removal," *IEEE TIP*, 2017.

[24] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *CVPR*, 2017.

[25] W. Yang, R. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *CVPR*, 2017.

[26] H. Zhang and V. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *CVPR*, 2018.

[27] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, "Depth-attentional features for single-image rain removal," in *CVPR*, 2019.

[28] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *ECCV*, 2018.

[29] K. Garg and S. K. Nayar, "Vision and rain," *IJCV*, 2007.

[30] L. Zhu, C. Fu, D. Lischinski, and P. Heng, "Joint bi-layer optimization for single-image rain streak removal," in *ICCV*, 2017.

[31] G. Wang, C. Sun, and A. Sowmya, "Erl-net: Entangled representation learning for single image de-raining," in *ICCV*, 2019.

[32] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *CVPR*, 2019.

[33] R. Yasarla, V. A. Sindagi, and V. M. Patel, "Syn2real transfer learning for image deraining using gaussian processes," in *CVPR*, 2020.

[34] H. Wang, Q. Xie, Q. Zhao, and D. Meng, "A model-driven deep neural network for single image rain removal," in *CVPR*, 2020.

[35] S. Deng, M. Wei, J. Wang, Y. Feng, L. Liang, H. Xie, F. L. Wang, and M. Wang, "Detail-recovery image deraining via context aggregation networks," in *CVPR*, 2020.

[36] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *CVPR*, 2020.

[37] Y. Wang, Y. Song, C. Ma, and B. Zeng, "Rethinking image deraining via rain streaks and vapors," in *ECCV*, 2020.

[38] Y. Chen and C. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *ICCV*, 2013.

[39] K. Zhang, W. Luo, W. Ren, J. Wang, F. Zhao, L. Ma, and H. Li, "Beyond monocular deraining: Stereo image deraining via semantic understanding," in *ECCV*, 2020.

[40] W. Yang, R. T. Tan, S. Wang, and J. Liu, "Self-learning video rain streak removal: When cyclic consistency meets temporal correspondence," in *CVPR*, 2020.

[41] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[42] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv:1412.7062*, 2014.

[43] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *NIPS*, 2017.

[44] N. Xu, B. Price, S. Cohen, and T. Huang, "Deep image matting," in *CVPR*, 2017.

[45] H. Zhao, J. S. andXiaojuan Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017.

[46] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018.

[47] H. Zhang, V. Sindagi, and V. Patel, "Image de-raining using a conditional generative adversarial network," *arXiv:1701.05957*, 2017.

[48] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[49] Adobe, "Adobe photoshop cs6."

[50] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS Workshop*, 2017.

[51] P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[52] S. Li, I. B. Araujo, W. Ren, Z. Wang, E. K. Tokuda, R. H. Junior, R. Cesar-Junior, J. Zhang, X. Guo, and X. Cao, "Single image deraining: A comprehensive benchmark analysis," in *CVPR*, 2019.

[53] S. Li, W. Ren, F. Wang, I. Araujo, E. Tokuda, R. Hirata-Junior, R. Cesar-Junior, Z. Wang, and X. Cao, "A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives," *IJCV*, 2020.

[54] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, 2004.

[55] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018.

[56] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *ECCV*, 2018.

[57] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *CVPR*, 2016.

[58] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *CVPR*, 2018.

Author Biographies:

Ke Xu is with the Department of Computer Science and Engineering at Shanghai Jiao Tong University, and City University of Hong Kong. He obtains the dual Ph.D. degrees from Dalian University of Technology and City University of HongKong. His research interests include deep learning, image restoration and enhancement.

Xin Tian is a PhD student in the Department of Computer Science at Dalian University of Technology and City University of HongKong. His research interests include salient object detection and image restoration.

Xin Yang is a professor in the Department of Computer Science at Dalian University of Technology, China. Xin received his B.S. degree in Computer Science from Jilin University in 2007. From 2007 to June 2012, he was a joint Ph.D. student in Zhejiang University and UC Davis for Graphics, and received his Ph.D. degree in July 2012. His research interests include computer graphics and robotic vision.

Baocai Yin is a professor of computer science department at the Dalian University of Technology and the dean of Faculty of Electronic Information and Electrical Engineer. His research concentrates on digital multimedia and computer vision. He received his B.S. degree and Ph.D. degree in computer science, both from Dalian University of Technology.

Rynson W.H. Lau received his Ph.D. degree from University of Cambridge. He was on the faculty of Durham University and is now with City University of Hong Kong.

Rynson serves on the Editorial Board of International Journal of Computer Vision (IJCV) and Computer Graphics Forum. He has served as the Guest Editor of a number of journal special issues, including ACM Trans. on Internet Technology, IEEE Trans. on Multimedia, IEEE Trans. on Visualization and Computer Graphics, and IEEE Computer Graphics & Applications. He has also served in the committee of a number of conferences, including Program Co-chair of ACM VRST 2004, ACM MTDL 2009, IEEE U-Media 2010, and Conference Co-chair of CASA 2005, ACM VRST 2005, ACM MDI 2009, ACM VRST 2014. Rynson's research interests include computer graphics and computer vision.