

# Heavy Rain Image Restoration: Integrating Physics Model and Conditional Adversarial Learning\*

Ruoteng Li<sup>1</sup>, Loong-Fah Cheong<sup>1</sup>, and Robby T. Tan<sup>1,2</sup>

<sup>1</sup>National University of Singapore

<sup>2</sup>Yale-NUS College

## Abstract

Most deraining works focus on rain streaks removal but they cannot deal adequately with heavy rain images. In heavy rain, streaks are strongly visible, dense rain accumulation or rain veiling effect significantly washes out the image, further scenes are relatively more blurry, etc. In this paper, we propose a novel method to address these problems. We put forth a 2-stage network: a physics-based backbone followed by a depth-guided GAN refinement. The first stage estimates the rain streaks, the transmission, and the atmospheric light governed by the underlying physics. To tease out these components more reliably, a guided filtering framework is used to decompose the image into its low- and high-frequency components. This filtering is guided by a rain-free residue image — its content is used to set the passbands for the two channels in a spatially-variant manner so that the background details do not get mixed up with the rain-streaks. For the second stage, the refinement stage, we put forth a depth-guided GAN to recover the background details failed to be retrieved by the first stage, as well as correcting artefacts introduced by that stage. We have evaluated our method against the state of the art methods. Extensive experiments show that our method outperforms them on real rain image data, recovering visually clean images with good details.

## 1. Introduction

As one of the commonest dynamic weather phenomena, rain causes significant detrimental impacts on many computer vision algorithms [30]. A series of rain removal methods have been proposed to address the problem (e.g. [16, 14, 41, 7, 38, 22, 36, 43, 23, 6, 29, 21]). Principally,

\*This work is supported by the DIRP Grant R-263-000-C46-232. R.T. Tan's research is supported in part by Yale-NUS College Start-Up Grant.

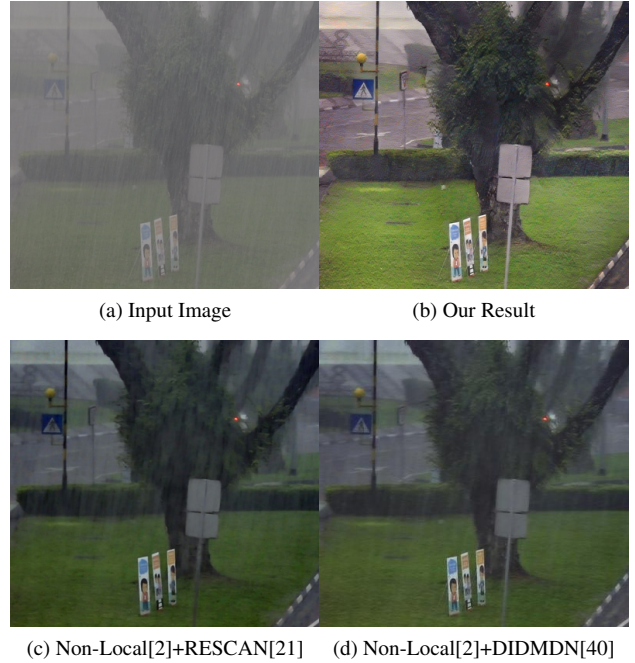


Figure 1: A comparison of our algorithm with combined state of the art dehazing/defogging [2] and deraining [21][40]. (Zoom-in to view details.)

these methods rely on the following model:

$$\mathbf{I} = \mathbf{J} + \sum_i^n \mathbf{S}_i, \quad (1)$$

where  $\mathbf{I}$  is the observed input image.  $\mathbf{J}$  is the background scene free from rain.  $\mathbf{S}_i$  is the rain layer, with  $n$  as the total number of rain-streak layers.

While the model in Eq. (1) is widely used, it crudely represents the reality. In real rain, particularly in relatively heavy rain, aside from the rain streaks, there is also a strong

veiling effect, which is the result of rain-streak accumulation in the line of sight. This important rain veiling effect (also known as rain accumulation) is ignored in the model. Hence, most of the existing methods do not perform adequately when dense rain accumulation is present (shown in Fig. 1). As one can observe in the figure, a state of the art method of rain-streak removal [21] combined with a state of the art dehazing/defogging method [2] still retains some rain streaks and veiling effect in the output. Note, zooming in the image will reveal the streaks and veiling effect.

The density of rain, both rain streaks and accumulation, is a spectrum. Thus, there is no clear dividing line between what light and heavy rain are. In this paper, we associate heavy rain to the severity of its visual degradation, namely when the rain streaks are strongly visible, the veiling effect significantly washes out the image, the distant background scenes are slightly blurry (due to multiflux scattering), and the physical presence of the rain streaks and rain accumulation is entangled with each other. The purpose of using the term “heavy rain” is to differentiate our method from other methods that do not address the mentioned problems.

To achieve our goal of restoring an image degraded by heavy rain, we need to address a few problems related to it. First, we can no longer utilize the widely used model (Eq. (1)), since it does not accommodate rain accumulation. We need a model that can represent both rain streaks and rain accumulation, like the one introduced by [38]:

$$\mathbf{I} = \mathbf{T} \odot (\mathbf{J} + \sum_i^n \mathbf{S}_i) + (\mathbf{1} - \mathbf{T}) \odot \mathbf{A}, \quad (2)$$

where  $\mathbf{T}$  is the transmission map introduced by the scattering process of the tiny water particles,  $\mathbf{A}$  is the global atmospheric light of the scene.  $\mathbf{1}$  is a matrix of ones, and  $\odot$  represents element-wise multiplication.

Second, aside from the model, existing methods tend to fail in handling heavy rain because, when dense rain accumulation (dense veiling effect) is present, the appearance of the rain streaks is different from the training data of the existing methods [7, 40, 38]. In the real world, rain streaks and rain accumulation can entangle with each other, which is intractable to be rendered using simple physics models. Hence, a sequential process (e.g, rain-streak removal followed by rain-accumulation removal) as suggested in [22, 38] cannot solve the problem properly. Moreover, unlike in fog images, estimating the atmospheric light,  $\mathbf{A}$ , in rain images is more complex, due to the strong presence of rain streaks. Note that, the proper estimation of the atmospheric light is critical, since it affects the restoration outputs significantly.

Third, particularly in heavy rain, the visual information of the background scene can be severely damaged. This is due to both rain streaks and rain accumulation as described

in Eq. (2). Unfortunately, some of the damages are not represented by the model. One of them is multiflux scattering effect in the form of blurriness of the scenes, particularly the further scenes [26]. In other words, the model cannot fully represent what happens in the real world. This creates performance problems, especially for methods that rely on the model, like most of the methods do.

To address these existing problems resulted by heavy rain, we introduce a novel CNN method to remove rain streaks as well as rain accumulation simultaneously with the following contributions:

1. We introduce an integrated two-stage neural network: a physics-based subnetwork and a model-free refinement subnetwork, to address the gap between physics-based rain model (Eq. (2)) and real rain. The first stage estimates  $\mathbf{S}$ ,  $\mathbf{A}$ ,  $\mathbf{T}$  and produces reconstructed image  $\mathbf{J}$  strictly governed by the rain model. The second stage contains a conditional GAN (cGAN) [25] that is influenced strongly by the outputs of the first stage.
2. We propose novel streak-aware decomposition to adaptively separate the image into high-frequency component containing rain streaks and low-frequency component containing rain accumulation. This addresses the problem of entangled appearance of rain streaks and rain accumulation. Also, since we can have a low frequency component, we can utilize it to resolve the problem of estimating the atmospheric light,  $\mathbf{A}$ .
3. We provide a new synthetic data generation pipeline that synthesizes the veiling effect in a manner consistent with the scene depth. For more realism, we also add Gaussian blur on both the transmission map and the background to simulate the effect of scattering in heavy rain scenarios.

Using these ideas, our experimental results show the superiority of our method compared to the state of the art methods qualitatively and quantitatively.

## 2. Related Works

Most existing deraining methods are not designed for heavy rain scenes, therein lies the main difference with our work. This applies to all the image-based [16, 24, 14, 22, 38, 7, 40, 21] and video-based works [41, 8, 1, 3, 23, 17, 6, 19, 31, 5, 32, 33, 39]. In the following, we focus our review on the image-based works.

Kang et al.’s [16] introduces the very first single image deraining method that decomposes an input image into its low frequency component and a high-frequency component using bilateral filter. The main difference with our decomposition method lies in that its high-frequency layer contains both rain streaks and high-frequency background

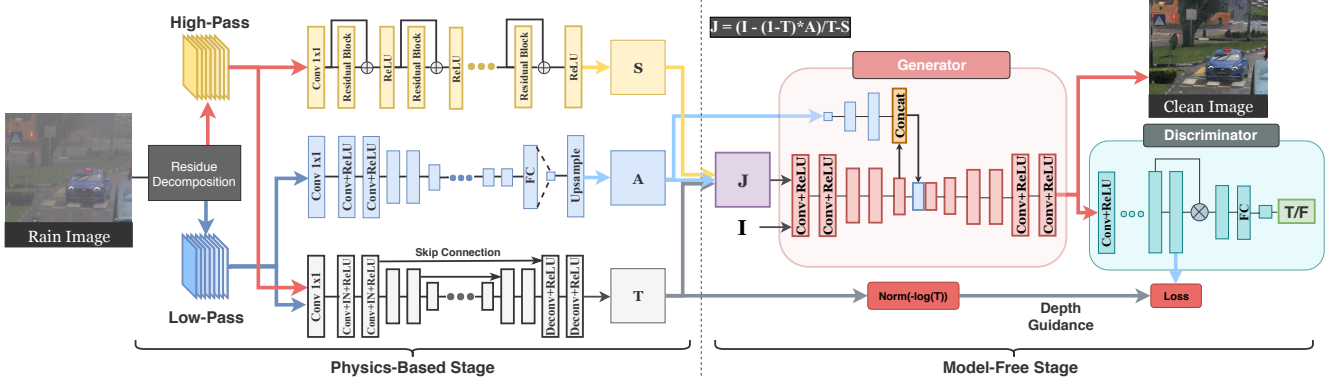


Figure 2: The overall architecture of the proposed network. The details of the residue decomposition module is shown in Fig. 3. The image  $\mathbf{J}$  is reconstructed according to Eq. (3) during training.

details—its sparse-coding based method using dictionary cannot differentiate genuine object details from the rain streaks. Li et al.’s [22] decomposes the rain image into a rain-free background layer and a rain streak layer, by utilizing Gaussian Mixture Models (GMMs) as a prior for the background and rain streak layers. This paper also attempts to address rain accumulation using a pre-processing dehazing step [4]. However, the dehazing step enhances clear rain streak further, causing the rain streak’s contrast and intensity much higher than that of the training data. Thus, the subsequent rain streak removal method cannot effectively remove boosted rain streaks. Fu et al. [7] proposes a deep convolutional network solution that is based on an image decomposition step similar to [16] and the details layer again contain both rain streaks and background details, which hampers the learning of rain streaks. Yang et al.’s [38] removes the rain accumulation using a dehazing method [4] as an iteration step in his recurrent framework. However in heavy rain scenes, a large number of noise hidden in the atmospheric veils will be boosted by dehazing method, which cannot be handled by Yang et al.’s rain streak removal module. Without treating the rain accumulation problem in an integral manner like our approach, it can only work well for the veiling effect produced in light rain, but not the heavy rain discussed in this paper. Both [40] and [21] are deep learning approaches that attempt to deal with the complex overlaying of rain layers in heavy rain scenes (by being density-aware and by having a recurrent network, respectively) but they do not deal with rain accumulation, and they also fail to remove the rain streaks cleanly in our experiments.

### 3. Network Design

Before describing the proposed 2-stage network, we first discuss the overall input and output of the network, as well as the intermediate output by the first stage. Referring to

Fig. 2, the first stage, the physics-based network, takes in a single rain image as input and extracts the physical parameters of rain, including the rain streak intensity  $\mathbf{S}$ , atmospheric light  $\mathbf{A}$  and transmission  $\mathbf{T}$ . The output of this first stage is the clean background image  $\mathbf{J}$  computed by the following equation (derived from Eq. (2)):

$$\mathbf{J} = \frac{\mathbf{I} - (\mathbf{1} - \mathbf{T}) \odot \mathbf{A}}{\mathbf{T}} - \sum_i^n \mathbf{S}_i. \quad (3)$$

The cGAN in the second stage refines the estimated  $\mathbf{J}$  to produce the clean background image  $\mathbf{C}$  as our final output.

The reason of proposing the 2-stage network is as follows. The physics model (Eq. (2)) is an approximated representation of real rain scenes, and thus can provide constraints to our network, such as rain-streaks ( $\mathbf{S}$ ), atmospheric light ( $\mathbf{A}$ ), and transmission ( $\mathbf{T}$ ). However, there is a significant disadvantage of using the physics model alone to design the network, since the model is only a crude approximation of the real world. Therefore, using a network that is purely based on the model will not make our method robust, particularly for heavy rain. As mentioned in the introduction, the damages induced by rain streaks and rain accumulation cannot be fully expressed by the model (Eq. (2)). For this reason, we add another network, the model-free network, which does not assume any model. Hence, unlike the first network, this network has less constraints and adapts more to the data. However, we cannot use this network alone either, since there is no proper guidance to the network in transforming a rain image to its clean image.

#### 3.1. Stage 1: Physics-based Restoration

The outline of our physics-based network is as follows. First, it decomposes the input image into high and low frequency components, where from the high frequency component, the network estimates the rain-streaks map,  $\mathbf{S}$ , and from the low frequency component, it estimates the atmo-

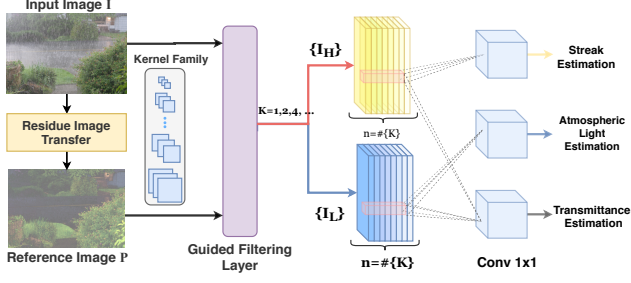


Figure 3: The schematic view of the structure of colored-residue image guided decomposition module.

spheric light,  $\mathbf{A}$ , and the transmission map,  $\mathbf{T}$ , as shown in Fig. 2. The details of these processes are discussed in these subsequent sections.

**Residue Channel Guided Decomposition** In rain images, particularly heavy rain, the visual appearances of rain streaks and rain accumulation are entangled in each other. This entanglement causes complexity in estimating the rain parameters:  $\mathbf{S}$ ,  $\mathbf{A}$ , and  $\mathbf{T}$ . Estimating  $\mathbf{A}$  and  $\mathbf{T}$  from the input image directly will be complex due to the strong presence of rain streaks. Similarly, estimating  $\mathbf{S}$  from the raw input image is intractable, due to the strong presence of dense rain accumulation. For this reason, we propose a process to decompose the input image into high and low frequency components, to reduce the complexity of the estimations and thus increase the robustness.

Our decomposition is adopted from [37], where we create a decomposition CNN layer that is differentiable during training (details shown in Fig.3). Specifically, we first perform image smoothing on the input image  $\mathbf{I}$ . The smoothed image is considered as the low-frequency component  $\mathbf{I}_L$  while the subtraction  $\mathbf{I}_H = \mathbf{I} - \mathbf{I}_L$  provides the high-frequency component. In each component, Eq. (2) becomes:

$$\begin{aligned} \mathbf{I}_H &= (\mathbf{1} - \mathbf{T}_H)(\mathbf{J}_H + \mathbf{S}_H) + \mathbf{T}_H \mathbf{A}_H, \\ \mathbf{I}_L &= (\mathbf{1} - \mathbf{T}_L)(\mathbf{J}_L + \mathbf{S}_L) + \mathbf{T}_L \mathbf{A}_L, \end{aligned} \quad (4)$$

where  $(\cdot)_H, (\cdot)_L$  represent the high-frequency component and low-frequency component respectively. Assuming the atmospheric light  $\mathbf{A}$  is constant throughout the image, we can assume that  $\mathbf{A}_H = \mathbf{0}$ . In addition, we also assume that low-frequency component of rain streak  $\mathbf{S}_L$  is negligible, i.e.,  $\mathbf{S}_L = \mathbf{0}$ . In other words, the low frequency of rain streaks mainly manifests itself as a veil (rain accumulation), and is modeled by  $\mathbf{A}_L$ . Hence, Eq. (4) reduces to:

$$\begin{aligned} \mathbf{I}_H &= (\mathbf{1} - \mathbf{T}_H)(\mathbf{J}_H + \mathbf{S}_H), \\ \mathbf{I}_L &= (\mathbf{1} - \mathbf{T}_L)(\mathbf{J}_L) + \mathbf{T}_L \mathbf{A}_L. \end{aligned} \quad (5)$$

The most important difference in our frequency decomposition lies in the use of the residue image [20] as a ref-

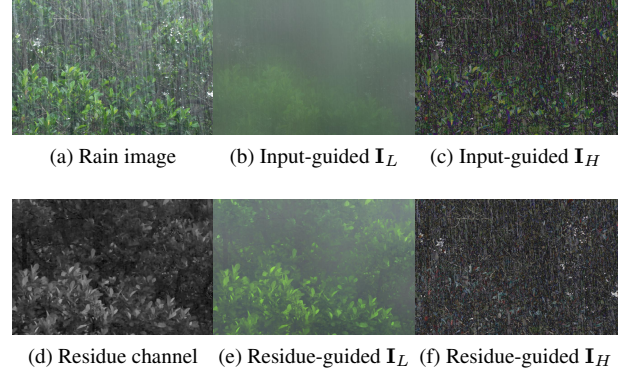


Figure 4: Input rain image decomposition using (a) input image itself and (d) its residue channel (kernel size  $k = 64 \times 64$ ) as guidance image. One can observe that more background details are left in the low-frequency channel.

erence image to guide the filtering during the aforementioned low-pass smoothing process. This guided filtering allows us to have a spatially variant low-frequency pass-band that selectively retains the high-frequency background details in the low-frequency channel. As a result, the high-frequency channel contains only rain streaks unmarred by high-frequency background details, which greatly facilitates the learning of rain streaks. The residue image is defined in [20] as follows:

$$\mathbf{I}_{res}(x) = \max_{c \in r, g, b} \mathbf{I}^c(x) - \min_{d \in r, g, b} \mathbf{I}^d(x), \quad (6)$$

where  $\mathbf{I}^c, \mathbf{I}^d$  are the color channels of  $\mathbf{I}$ . This residue channel is shown to be invariant to rain streaks, i.e., it is free of rain streaks and contains only a transformed version of the background details (see Fig. 4 (d)). It can thus provide information to guide and vary the passband in the low-frequency smoothing so that the background details are not smoothed away. In practice, we use the colored-residue image [20] as shown in Fig. 3.

To handle the large variation in the rain streak size present in our rain images, the decomposition uses a set of smoothing kernels  $K$ , with size given by  $k = 2^i$ ,  $i = 0, 1, \dots$ . In each of the frequency channels, we concatenate these images and send them to a  $1 \times 1$  convolutional kernel, which behaves as a channel-wise feature selector.

**Learning Rain Streaks** From the high-frequency component  $\mathbf{I}_H$ , we learn the rain streaks  $\mathbf{S}$  from the ground-truth streaks map using a fully convolutional network containing 12 residual blocks [11]:

$$\mathcal{L}_S = \mathcal{L}_{MSE}(\mathbf{S}, \mathbf{S}_{gt}), \quad (7)$$

where  $\mathcal{L}_S$  represents the loss for learning rain streaks and  $\mathbf{S}_{gt}$  is the groundtruths of a rain-streaks map.

**Learning Atmospheric Light** The atmospheric light sub-network learns to predict the global atmospheric light  $\mathbf{A}$  only from the low-frequency component  $\{\mathbf{I}_L\}^k$ . This is because the low-frequency component does not contain rain streaks, where its specular reflection may significantly change the brightness of the input image and adversely affects the estimation of  $\mathbf{A}$ . This subnetwork is composed of 5 Conv+ReLU blocks appended with 2 fully-connected layers. The output vector  $\mathbf{A}$  is then upsampled to the size of the input image for the estimation of  $\mathbf{J}$  in Eq. (3). The loss function for learning  $\mathbf{A}$  is defined by:

$$\mathcal{L}_A = \mathcal{L}_{MSE}(\mathbf{A}, \mathbf{A}_{gt}), \quad (8)$$

where  $\mathbf{A}_{gt}$  is the groundtruth of the atmospheric light.

**Learning Transmission** We use an auto-encoder with skip connection to learn the transmission map  $\mathbf{T}$ . We adopt the instance normalization [34] instead of batch normalization in the first two convolutional layers, as in our experiment, the latter performs poorly when the testing data has a significant domain gap from the training data. The loss function for learning  $\mathbf{T}$  is defined as:

$$\mathcal{L}_T = \mathcal{L}_{MSE}(\mathbf{T}, \mathbf{T}_{gt}), \quad (9)$$

where  $\mathbf{T}_{gt}$  refers to the ground-truth transmission map.

**Loss functions** Based on the preceding, the overall loss function for the physics-based network to predict the physical parameters  $\Theta$  is:

$$\mathcal{L}_\Theta = \lambda_S \mathcal{L}_S + \lambda_A \mathcal{L}_A + \lambda_T \mathcal{L}_T, \quad (10)$$

where  $\lambda_S$ ,  $\lambda_A$  and  $\lambda_T$  are weighting factors for each loss. In our experiment, they are all set to 1 since they are all MSE losses with the same scale.

### 3.2. Stage 2: Model-Free Refinement

The model-free refinement stage contains a conditional generative adversarial network. The generative network takes in the estimated image  $\mathbf{J}$  and rain image  $\mathbf{I}$  as input and produces the clean image  $\mathbf{C}$  to be assessed by the discriminative network. The overall loss function for the cGAN is:

$$\begin{aligned} \min_G \max_D V(D, G) &= \mathbb{E}_{\mathbf{C} \sim p(\mathbf{C}, \mathbf{I})} [\log D(\mathbf{C}|\mathbf{I})] \\ &+ \mathbb{E}_{\mathbf{J} \sim p(\mathbf{J}, \mathbf{I})} [\log(1 - D(G(\mathbf{J}|\mathbf{I})))] \end{aligned} \quad (11)$$

where  $D$  represents the discriminative network and  $G$  represents the generative network.

**Generative Network** The generative network is an autoencoder that contains 13 Conv-ReLU blocks, and skip connections are added to preserve more low-level image details.

The goal of the generative network is to generate a refined clean version  $\mathbf{C}$  that looks real and free from rain effect and artefacts produced by the previous stage. The input of this generator is  $\mathbf{I}$  and  $\mathbf{J}$ . Since  $\mathbf{J}$  is considerably sensitive to the estimation errors in the atmospheric light  $\mathbf{A}$ , the generator may not be able to learn effectively. To improve the training, we inject the estimated atmospheric light  $\mathbf{A}$  into the generator as shown in Fig. 2. In particular, we first embed  $\mathbf{A}$  into a higher dimensional space using two convolutions before concatenating the result with the encoder output of the generative network. This is done at the highest layer of the encoder where more global features are represented, because  $\mathbf{A}$  itself is a global property of the scene.

We also add MSE and perceptual losses [15] for the training of the generative network. They are given by the first and second terms in the following loss function:

$$\begin{aligned} \mathcal{L}_C &= \mathcal{L}_{MSE}(\mathbf{C}, \mathbf{C}_{gt}) \\ &+ \lambda_p \mathcal{L}_{MSE}(VGG(\mathbf{C}), VGG(\mathbf{C}_{gt})), \end{aligned}$$

where  $\lambda_p = 8$  in our experiment, and the perceptual loss is based on VGG16 pretrained on the ImageNet dataset.

Overall, the loss function for the generative network is:

$$\mathcal{L}_G = \mathcal{L}_C + \lambda_{GAN} \mathcal{L}_{GAN}(\mathbf{C}), \quad (12)$$

where  $\mathcal{L}_{GAN}(\mathbf{C}) = \log(1 - D(\mathbf{C}))$  and the weighting parameter  $\lambda_{GAN}$  is set to 0.01.

**Discriminative Network** The discriminative network accepts the output of generative network and checks if it looks like a realistic clear scene. Since it is usually the distant scene that suffers loss of information, we want to make sure that the GAN focuses on these faraway parts of the scene. We first leverage the transmission map  $\mathbf{T}$  produced from the physics-based network and convert it to a relative depth map according to the relationship:

$$\mathbf{T}(x) = \exp^{-\beta \mathbf{d}(x)}, \quad (13)$$

where  $\mathbf{d}$  represents the scene depth and  $\beta$  indicates the intensity of the veil or rain accumulation (in our experiment,  $\beta$  is randomly sampled from a uniform distribution in [3, 4.2]). Then, we take the features from the 6<sup>th</sup> Conv-ReLU layer of the discriminator and compute the MSE loss between these features and the depth map  $-\log(\mathbf{T})$  normalized to [0, 1]:

$$\mathcal{L}_{depth}(\mathbf{C}, \mathbf{T}) = \mathcal{L}_{MSE}(Conv(D(\mathbf{C}))_6, Norm(-\log \mathbf{T})), \quad (14)$$

where  $D(\mathbf{C})_m$  represents the features at the  $m^{\text{th}}$  layer of the discriminator. We use the learnt depth map to weigh the features from the previous layer by multiplying them in an element-wise manner:

$$D(\mathbf{C})_7 = \mathbf{d} \odot D(\mathbf{C})_6. \quad (15)$$

**Algorithm 1** Algorithm for *Outdoor-Rain* Rendering

- 1: **Input:** Clean Image  $\mathbf{C}$  and its depth map  $\mathbf{D}$
- 2:  $\mathbf{C}_{blur}(\mathbf{x}) = \text{imgaussfilt}(\mathbf{C}(\mathbf{x}), \sigma_{\mathbf{C}}(\mathbf{x}))$ . The smooth kernel varies according to depth:  $\sigma_{\mathbf{C}}(\mathbf{x}) = 1.5\mathbf{D}(\mathbf{x})$ .
- 3: Generate 2D Noise map  $\mathbf{N}$  with  $\mu \sim -U(0, 0.2) - 0.8$ ,  $\sigma \sim U(0, 0.3) + 0.7$
- 4: Rain Streaks map  $\mathbf{S} = \text{immotionfilt}(\mathbf{N}, l, \theta)$ , parameter  $l \sim U(0, 40) + 20$ ,  $\theta \sim U(80, 100)$
- 5: Obtain Rain image  $\mathbf{I}_S = \mathbf{S} + \mathbf{C}_{blur}$
- 6: Obtain Transmission  $\mathbf{T} = \exp^{-\beta\mathbf{D}}$ ,  $\beta \sim U(3, 4.2)$
- 7: Obtain  $\mathbf{T}_{blur} = \text{imgaussfilt}(\mathbf{T}, \sigma_T)$ ,  $\sigma_T \sim \mathcal{N}(5, 1.5)$ .
- 8: Obtain global atmospheric light  $\mathbf{A} \sim U(0.3, 0.8)$
- 9: **Output:** Rain Image  $\mathbf{I} = \mathbf{T}_{blur}\mathbf{I}_S + (1 - \mathbf{T}_{blur})\mathbf{A}$

Since faraway objects have higher depth values  $\mathbf{d}$ , the errors coming from these objects will be back-propagated to the generative network with greater weights during training.

The whole loss function of the discriminative network can be expressed as :

$$\begin{aligned} \mathcal{L}_D = & -\log(D(\mathbf{C}_{gt})) - \log(1 - D(\mathbf{C})) \\ & + \mathcal{L}_{depth}(\mathbf{C}, \mathbf{T}) + \mathcal{L}_{depth}(\mathbf{C}_{gt}, \mathbf{T}_{gt}). \end{aligned} \quad (16)$$

## 4. Implementation

### 4.1. Data Generation

There are several large-scale synthetic datasets available for training deraining networks; however none of them contains rain accumulation effects. Hence, for the training of the physics-based stage, we create a new synthetic rain dataset named *NYU-Rain*, using images from NYU-Depth-v2 [27] dataset as background. We render synthetic rain streaks and rain accumulation effects based on the provided depth information. These effects include the veiling effect caused by the water particles, as well as image blur (for details of the rain rendering process, see Algorithm 1). This dataset contains 16,200 image samples, out of which 13,500 images are used as the training set. For the training of the model-free refinement stage, we create another outdoor rain dataset on a set of outdoor clean images from [28], denoted as *Outdoor-Rain*. In order to render proper rain streaks and rain accumulation effects as above, we estimate the depth of the scene using the state of the art single image depth estimation method [9]. This dataset contains 9000 training samples and 1,500 validation samples.

### 4.2. Training Details

The proposed network is first trained in a stage-wise manner and then fine-tuned on an end-to-end basis. To train the physics-based stage on the *NYU-Rain* dataset, we use Adam [18] optimizer with weight decay  $10^{-4}$  and only supervise  $\mathcal{L}_{\Theta}$ . The learning rate is set to 0.001 initially and is

Table 1: A comparison on performance of estimated  $\mathbf{S}$ ,  $\mathbf{A}$ ,  $\mathbf{T}$  and  $\mathbf{J}$  among three different architectures on *Test 1* data.

Method	Guidance Image	$\mathbf{J}$	$\mathbf{S}$	$\mathbf{T}$	$\mathbf{A}$
Metric		PSNR	PSNR	PSNR	Error
No Decomposition	-	10.87	23.65	14.95	0.212
Decomposition	Input Image	11.30	23.42	15.85	0.151
Decomposition	Residue Channel	<b>13.83</b>	<b>23.70</b>	<b>19.48</b>	<b>0.150</b>
Improvement over “No Decomposition”		27.23 %	0.21 %	30.30 %	29.25 %

divided by 2 after every 10 epochs until the 60<sup>th</sup> epoch. To train the model-free refinement stage, we fix the parameters of the physics-based network and use the same optimizer and learning rate schedule as above. This model-free network is trained up to the 100<sup>th</sup> epochs in this stage. Finally, we unfreeze the parameters in the physics-based network and fine-tune the entire model for a few thousand iterations. The entire network is implemented in Pytorch framework and will be made publicly available. <sup>1</sup>

## 5. Experimental Results

In this section, we evaluate our algorithm with a few baseline methods on both the synthetic rain data and real rain data. For synthetic rain evaluation, we created a test datasets based on the test images from [28] using the same rendering techniques in Algorithm 1, denoted as *Test 1*. For a fair comparison with baselines, we combine the state of the art dehazing method [2] with a series of state of the art rain streaks removal methods: (a) Deep detailed Network (DDN) [7], (b) DID-MDN method [40], (c) RESCAN [21] method, and (d) JCAS [10] method. In addition, we also compare with Pix2Pix GAN [13] and CycleGAN [42] trained on the *Outdoor-Rain* dataset.

### 5.1. Ablation Study

**Derain + Dehaze or Dehaze + Derain?** The first ablation study evaluates the performance of combined dehazing and deraining methods in different order. We denote DeHaze First as *DHF* and DeRain First as *DRF*. We test these methods on *Test 1* dataset and Table 2 shows the quantitative results of these baseline methods in PSNR [12] and SSIM [35] metric. We will henceforth compare our method with the better pipeline.

**Decomposition Module** To study the effectiveness of the decomposition module, we compared three different network architectures: (a) No decomposition module in the first stage, denoted as “No Decomposition”. (b). Decomposition module using input image as guidance image, denoted as “Input-guided Decomposition”. (c). We use the architecture proposed in this paper, named as “Residue-guided Decomposition”. We run these three methods on the testing dataset *Test 1* and evaluate the estimated  $\mathbf{S}$ ,  $\mathbf{T}$

<sup>1</sup><https://github.com/liruoteng/HeavyRainRemoval>



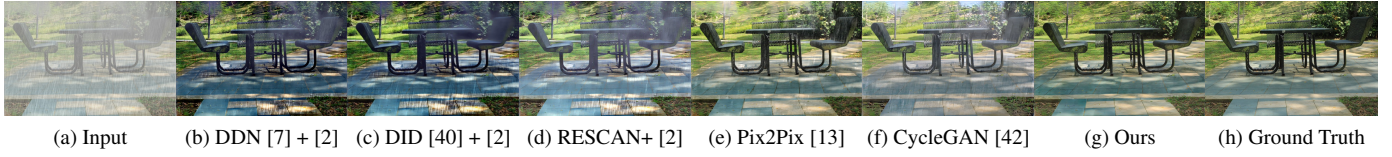


Figure 5: A comparison of our algorithm with the baseline methods performed on *Test 1* dataset.

Table 2: A comparison of our algorithm with the baseline methods performed on *Test 1* dataset.

Method		Test 1	
Metric		PSNR	SSIM
JCAS [10] + Dehaze	DHF	14.95	0.590
	DRF	16.44	0.599
DDN [7] + Dehaze	DHF	13.36	0.583
	DRF	15.68	0.640
DID-MDN [40] + Dehaze	DHF	14.17	0.577
	DRF	12.58	0.471
RESCAN [21] + Dehaze	DHF	14.72	0.587
	DRF	15.91	0.615
Pix2Pix [13]		19.09	0.710
CycleGAN [42]		17.62	0.656
No Decomposition + Stage 2		20.82	0.832
Ours-J		20.05	0.779
Ours-C		<b>21.56</b>	<b>0.855</b>

and the reconstructed image  $\mathbf{J}$  in PSNR [12] metric. For atmospheric light  $\mathbf{A}$ , we evaluated the sum error against the ground-truth  $\mathbf{A}_{gt}$ :  $Error = \sum_{i \in r, g, b} |\mathbf{A}^i - \mathbf{A}_{gt}^i|$ . From the quantitative results shown in Table 1, the decomposition operation significantly increases the accuracy of transmission estimation and thus improves the reconstructed image  $\mathbf{J}$ . Since the decomposition guided by input image cannot fully separate rain streaks from the low-frequency component, the estimated  $\mathbf{S}$  does not gain advantage. However, using the streak-free residue channel as guidance image, the transmission and atmospheric light will benefit from the streak-free low-frequency component, leading to further improvement on estimation.

**Study of Refinement Stage** Fig. 6 shows the comparison between reconstructed image  $\mathbf{J}$  and final output  $\mathbf{C}$  produced by our network on real-world rain image. One can observe that there are dark regions around the distant tree are on image  $\mathbf{J}$ . The darkened result is one of the common problems in dehazing methods. Our refinement network is able to identify these areas and restore the contextual details of the distant tree with visually fine color according to the relative depth map  $\mathbf{d}$  converted from estimated transmission map  $\mathbf{T}$  using Eq. (13).

## 5.2. Synthetic Rain Analysis

Table 2 demonstrates the quantitative performance of our algorithm compared with the baseline methods in PSNR



Figure 6: The reconstructed image  $\mathbf{J}$  produces darkened result on distant objects. The refinement network restores the details according to normalized depth map  $\mathbf{d}$ .

[12] and SSIM [35] metrics. Fig. 5 shows the qualitative results produced by our algorithm and other baseline methods. Here, we choose the better performed result between dehaze+derain and derain+dehaze for those rain streaks removal methods. [10][7][21][40]. Note that directly using GAN method such as [13] [42] does not produce appropriate solution for this image enhancement problem since these generative models can sometimes generate fake results as shown in the first example (top part) of Fig.5.

## 5.3. Real-world Rain Analysis

**Qualitative Result** Fig. 7 shows the qualitative comparison between our method and other baseline methods. For the baseline methods under moderate rain scenes, the haze removal component usually produces dark results and the rain removal components inevitably damage the background details, resulting in blurred image. (e.g. the tree leaves and the lamp poles in Fig. 7 Row 1,2). In the case of heavy rain, these baseline methods fail to remove the rain streaks effectively due to the presence of strong rain accumulation (Fig. 7 Row 5). In addition, the state of the art haze removal method cannot effectively remove the veiling effect. One can still observe hazy effect at the remote area of the baseline results (row 4 of Fig. 7). Thanks to the depth guided GAN, our method is able to identify the remote areas and remove the proper amount of veiling effect.

**Application** In order to provide the evidence that our image restoration method will benefit outdoor computer vision applications, we employ Google Vision API object recognition system to evaluate our results. Fig. 8 shows the screenshots of the results produced by Google API. We test 20 sets of real rain images and derained images of our method and baseline methods [7, 21]. We report the classification

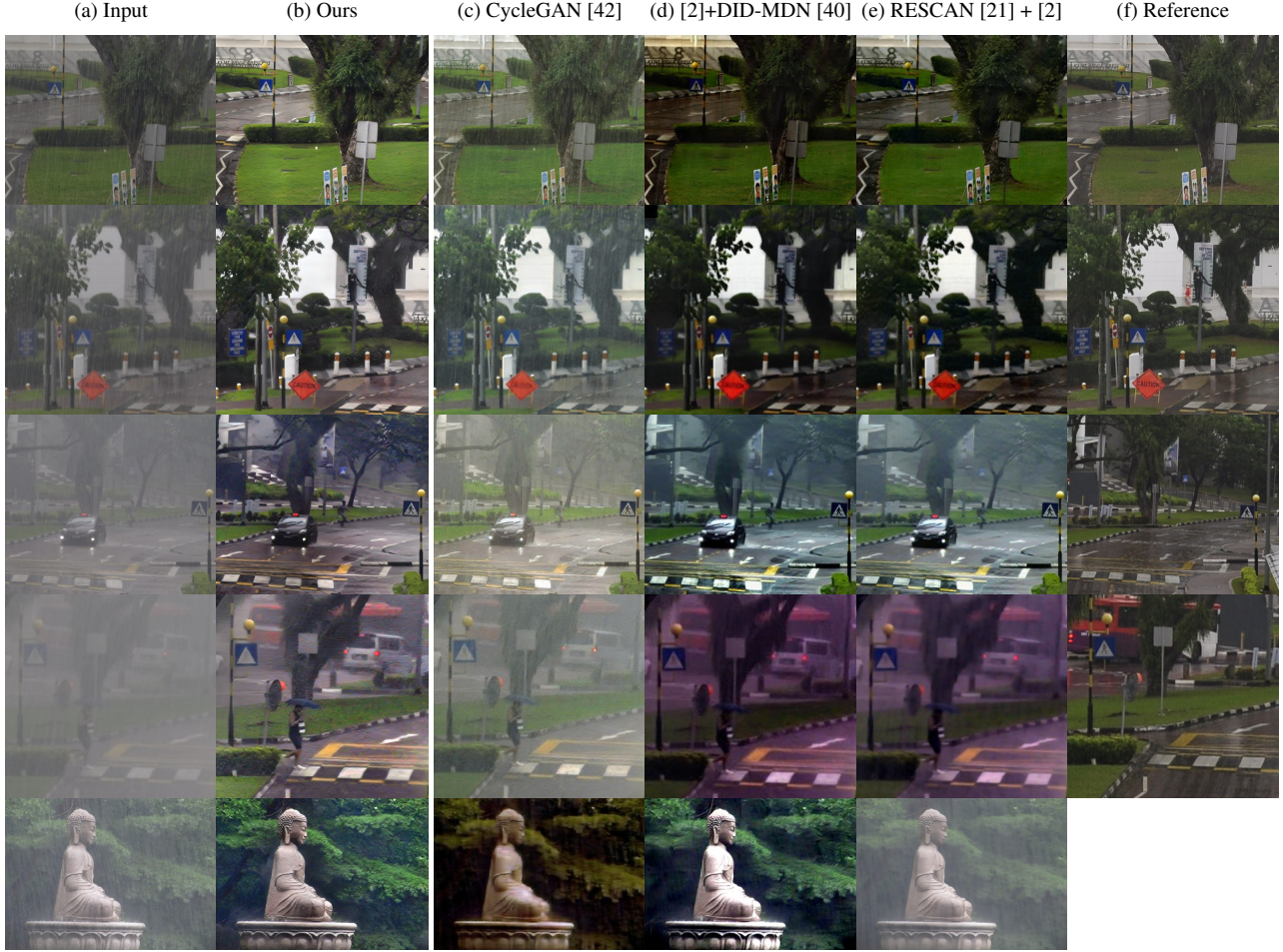


Figure 7: A comparison of our algorithm with baseline methods on real-world rain scenes. The reference images are other pictures taken just after rains. From top to bottom, the rain becomes more and more severe. (Zoom-in to view details).

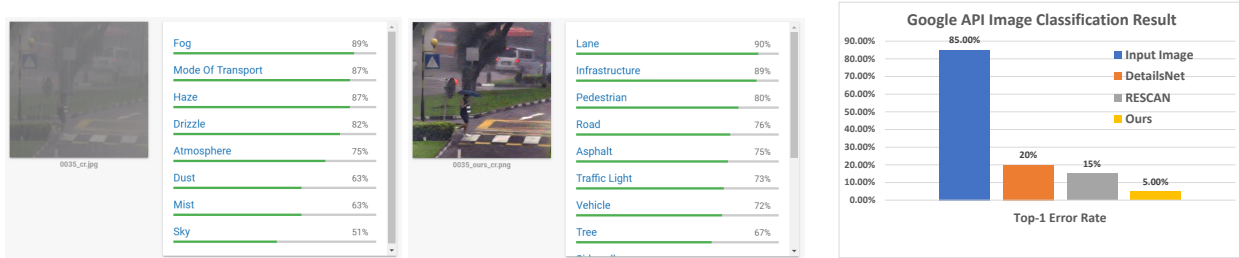


Figure 8: Object recognition results for the input rain image and our results respectively. We test 20 sets of rain and derain images of ours and baseline methods [21, 7]. We record the top-1 error rate on the right bar chart.

results of top-1 error rate. As one can see, our method significantly improve the recognition results and outperforms other baseline methods.

## 6. Conclusion

We propose a novel 2-stage CNN that is able to remove rain streaks and rain accumulation simultaneously. In the first physics-based stage, a new streak-aware decomposition module is introduced to decompose the entangled rain

streaks and rain accumulation for better joint feature extraction. Scene transmission and atmospheric light are also estimated to provide necessary depth and light information for second stage. We propose a conditional GAN in the refinement stage that takes in the reconstructed image from previous level and produce the final clean images. Comprehensive experimental evaluations show that our method outperforms the baselines on both synthetic and real rain data.



## References

- [1] P. Barnum, T. Kanade, and S. Narasimhan. Spatio-temporal frequency analysis for removing rain and snow from videos. In *Proceedings of the First International Workshop on Photometric Analysis For Computer Vision-PACV 2007*, pages 8–p. INRIA, 2007.
- [2] D. Berman, T. Treibitz, and S. Avidan. Non-local image dehazing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [3] J. Bossu, N. Hautière, and J.-P. Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International Journal of Computer Vision*, 93(3):348–367, Jul 2011.
- [4] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *Trans. Img. Proc.*, 25(11):5187–5198, Nov. 2016.
- [5] J. Chen and L. Chau. A rain pixel recovery algorithm for videos with highly dynamic scenes. *IEEE Transactions on Image Processing*, 23(3):1097–1104, March 2014.
- [6] J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li. Robust video content alignment and compensation for rain removal in a cnn framework. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [7] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [8] K. Garg and S. K. Nayar. Vision and rain. *Int. J. Comput. Vision*, 75(1):3–27, Oct. 2007.
- [9] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.
- [10] S. Gu, D. Meng, W. Zuo, and L. Zhang. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.
- [12] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13):800–801, June 2008.
- [13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016.
- [14] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [15] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.
- [16] L. W. Kang, C. W. Lin, and Y. H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, April 2012.
- [17] J. H. Kim, J. Y. Sim, and C. S. Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing*, 24(9):2658–2670, Sept 2015.
- [18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [19] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, and D. Meng. Video rain streak removal by multiscale convolutional sparse coding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [20] R. Li, R. T. Tan, and L.-F. Cheong. Robust optical flow in rainy scenes. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- [21] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- [22] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [23] J. Liu, W. Yang, S. Yang, and Z. Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [24] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3397–3405, Dec 2015.
- [25] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.
- [26] S. G. Narasimhan and S. K. Nayar. Shedding light on the weather. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR’03*, pages 665–672, Washington, DC, USA, 2003. IEEE Computer Society.
- [27] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.
- [28] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [29] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang. Video desnowing and deraining based on matrix decomposition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [30] S. R. Richter, Z. Hayder, and V. Koltun. Playing for benchmarks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2232–2241, 2017.
- [31] V. Santhaseelan and V. K. Asari. A phase space approach for detection and removal of rain in video. In *Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*, volume 8301, page 830114, Jan. 2012.

- [32] V. Santhaseelan and V. K. Asari. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision*, 112(1):71–89, Mar 2015.
- [33] A. K. Tripathi and S. Mukhopadhyay. Video post processing: low-latency spatiotemporal approach for detection and removal of rain. *IET Image Processing*, 6(2):181–196, March 2012.
- [34] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016.
- [35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.
- [36] 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 *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [37] H. Wu, S. Zheng, J. Zhang, and K. Huang. Fast end-to-end trainable guided filter. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [38] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Joint rain detection and removal via iterative region dependent multi-task learning. *CoRR*, abs/1609.07769, 2016.
- [39] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Adherent raindrop modeling, detection and removal in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1721–1733, Sept 2016.
- [40] H. Zhang and V. M. Patel. Density-aware single image de-raining using a multi-stream dense network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [41] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng. Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE International Conference on Multimedia and Expo*, pages 461–464, July 2006.
- [42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.
- [43] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng. Joint bi-layer optimization for single-image rain streak removal. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.