Uncertainty Guided Multi-Scale Residual Learning-using a Cycle Spinning CNN for Single Image De-Raining

Rajeev Yasarla and Vishal M. Patel
Johns Hopkins University
Department of Electrical and Computer Engineering, Baltimore, MD 21218, USA
ryasarl1@jhu.edu, vpatel36@jhu.edu

Abstract

Single image de-raining is an extremely challenging problem since the rainy image may contain rain streaks which may vary in size, direction and density. Previous approaches have attempted to address this problem by leveraging some prior information to remove rain streaks from a single image. One of the major limitations of these approaches is that they do not consider the location information of rain drops in the image. The proposed Uncertainty guided Multi-scale Residual Learning (UMRL) network attempts to address this issue by learning the rain content at different scales and using them to estimate the final de-rained output. In addition, we introduce a technique which guides the network to learn the network weights based on the confidence measure about the estimate. Furthermore, we introduce a new training and testing procedure based on the notion of cycle spinning to improve the final de-raining performance. Extensive experiments on synthetic and real datasets to demonstrate that the proposed method achieves significant improvements over the recent state-of-the-art methods.

1. Introduction

Many practical computer vision-based systems such as surveillance and autonomous driving often require processing and analysis of videos and images captured under adverse weather conditions such as rain, snow, haze etc. These weather-based conditions adversely affect the visual quality of images and as a result often degrade the performance of vision systems. Hence, it is important to develop algorithms that can automatically remove these artifacts before they are fed to a vision-based system for further processing.

In this paper, we address the problem of removing rain streaks from a single rainy image. Rain streak removal or image de-raining is a difficult problem since a rainy image may contain rain streaks which may vary in size, direction and density. A number of different techniques have been developed in the literature to address this problem. These algorithms can be clustered into two main groups - (i) video based algorithms [36, 9, 25, 20, 16], and (ii) single image-based algorithms [35, 8, 18, 31, 37]. Algorithms corresponding to the first category assume temporal consistency among the image frames, and use this assumption for de-raining. On the other hand, single image de-raining methods attempt to use some prior information to remove rain components from a single image [18, 13, 37, 35]. Priors such as sparsity [33, 21] and low-rank representation [4] have been used in the literature. In particular, the method proposed by Fu et al. [8] uses a priori image domain knowledge by focusing on high frequency details during training to improve the de-raining performance. However, it was shown in [35], that this method tends to remove some im-

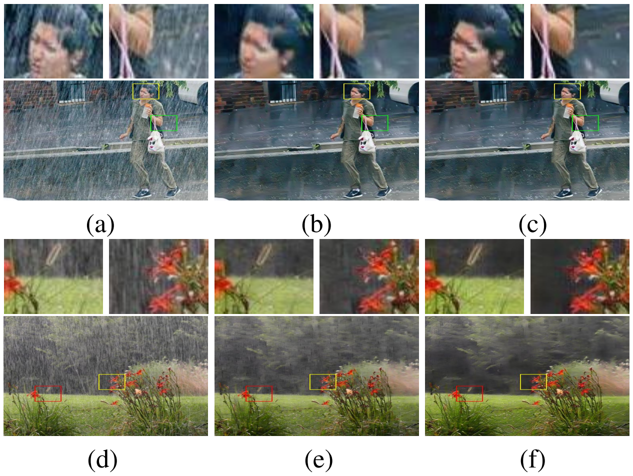


Figure 1: Sample image de-raining results. (a) Rainy image. (b) De-rained using DID-MDN [35] where zoomed in part shows the blurry effects on face and various rain streaks near the elbow. (c) De-rained using UMRL. (d) Rainy image. (e) De-rained using Fu et al. [8] where zoomed in part shows under de-raining of the image. (f) De-rained using UMRL, zoomed in highlighted parts show the clear differences between UMRL and other compared methods.

portant parts in the de-rained image (see Figure 1(e)). Similarly, a recent work by Zhang and Patel [35] uses the image-level priors to estimate the rain density information which is then used for de-raining. Although their approach provides the state-of-the-art results, they estimate image level priors which do not consider the location information of rain drops in the image. As a result, their algorithm tends to introduce some artifacts in the final de-rained images. These artifacts can be clearly seen from the de-rained results shown in Figure 1(b).

In this paper we take a different approach to image de-raining where we make use of the observation that rain streak density and direction does not change drastically with different scales. Rather than relying on the rain density information (i.e. heavy, medium or light) present in the rainy image [35], we develop a method in which the rain streak location information is taken in to consideration in a multi-scale fashion to improve the de-raining performance. While providing the estimated rain content (i.e. residual map) to the subsequent layers of the network, we may end-up propagating the errors in estimations. To block the flow of incorrect estimation in rain streaks, we estimate an uncertainty metric along with the rain streak information. We use an Unet architecture with skip connections [24] as our base network. The proposed network learns the residue at each level in the decoder of Unet with an uncertainty map, which indicates how confident the network is about the rain content it learned. Say there are L layers in the decoder network, the uncertainty map generated at layer "l" is given to layer "$l+1$" so that the subsequent layers of "l" can discard the rain content learned by layer "l" if the confidence value is low in the uncertainty map.

Another important contribution of our work is that we propose to incorporate the cycle spinning framework of Coifman and Donoho [5] into our de-raining method. Cycle spinning was originally proposed to remove the artifacts introduced by orthogonal wavelets in image de-noising. Similar to wavelets, deep learning-based methods also introduce some artifacts near the edges of the de-rained images (see Figure 1). In cycle spinning, the data is first shifted by some amount, the shifted data are then de-noised, the de-noised data are then un-shifted, and finally the un-shifted data are averaged to obtain the final de-noised result. Cycle spinning has been successfully applied to reduce the artifacts introduced near the edges in many applications including image de-blurring [6] and de-noising [5], [2]. Hence, we adopt it in our de-raining framework. In fact, we show that cycle spinning is a generic method that can be used to improve the performance of any deep learning-based image de-raining method.

Figure 1 (c) and (f) present sample results from our Uncertainty guided Multi-scale Residual Learning using cycle spinning (UMRL) network, where one can clearly see that UMRL is able to remove the noise artifacts and provides better results as compared to [35] and [8].

To summarize, this paper makes the following contributions:

- A novel method called UMRL is proposed which generates the rain streak content at each location of the image along with the uncertainty map that guides the subsequent layers about the rain streak information at each location.
- We incorporate cycle spinning in both training and testing phases of our network to improve the final de-raining performance.
- We run extensive experiments to show the performance of UMRL against the several recent state-of-the-art approaches on both synthetic and real rainy images. Furthermore, an ablation study is conducted to demonstrate the effectiveness of different parts of the proposed UMRL network.

2. Background and Related Work

An observed rainy image y can be modeled as the superposition of a rain component (i.e. residual map) r with a clean image x as follows

$$y = x + r. \tag{1}$$

Given y the goal of image de-raining is to estimate x. This can be done by first estimating the residual map r and then subtracting it from the observed image y. Various methods have been proposed in the literature for image de-raining [13, 3, 21, 10, 18] including dictionary learning-based [1], Gaussian mixture-model (GMM) based [23], and low-rank representation based [19] methods. In recent years, deep learning-based single image de-raining methods have also been proposed in the literature. Fu et al. [7] proposed a convolutional neural network (CNN) based approach in which they directly learn the mapping relationship between rainy and clean image detail layers from data. Zhang et al. [34] proposed a generative adversarial network (GAN) based method for image de-raining. Furthermore, to minimize the artifacts introduced by GANs and ensure better visual quality, a new refined loss function was also introduced in [34]. Fu et al. [8] presented an end-to-end deep learning framework for removing rain from individual images using a deep detail network which directly reduces the mapping range from input to output. Zhang and Patel [35] proposed a density-aware multi-stream densely connected CNN for joint rain density estimation and de-raining. Their network automatically determines the rain-density information and then efficiently removes the corresponding rain-streaks using the estimated rain-density label. Note the methods proposed in [8], and [35] showed the benefits of using multi-scale networks for image de-raining. Recently, Wang et al.

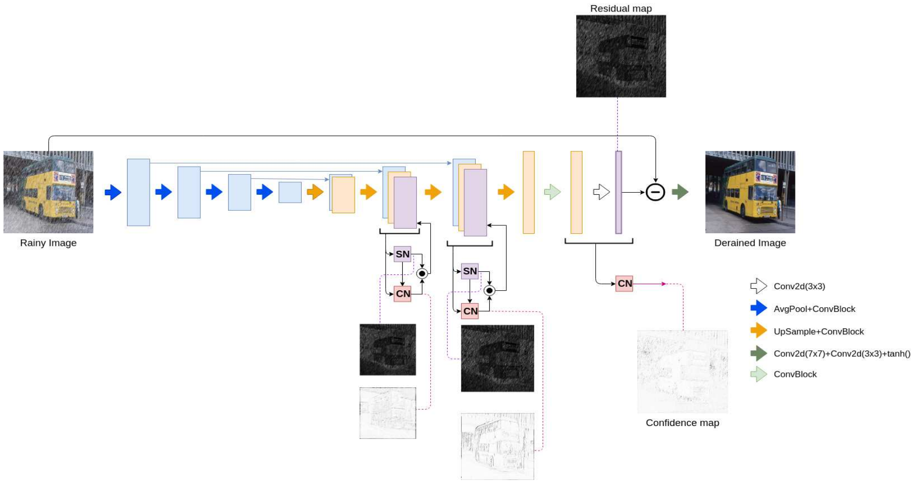


Figure 2: An overview of the proposed UMRL network. The aim of the UMRL network is to estimate the clean image given the corresponding rainy image. To address that, UMRL learns the residual maps and computes the confidence maps to guide the network. To achieve this, we introduce RN and CN networks and feed their outputs to the subsequent layers.

[28] proposed a hierarchical approach based on estimating different frequency details of an image to get the de-rained image. The method proposed by Qian et al. [22] generates attentive maps using the recurrent neural networks, and then uses the features from different scales to compute the loss for removing the rain drops on glasses. Note that this method was specifically designed for removing rain drops from a glass rather than removing rain streaks from an image. [27, 30, 17] illustrated the importance of attention based methods in low-level vision tasks. In a recent work [17], Li et al. proposed a convolutional and recurrent neural network-based method for single image de-raining which makes use of the contextual information for rain removal. It was observed in [35], that some of the recent deep learning-based methods tend to under de-rain or over de-rain the image if the rain condition present in the rainy image is not properly considered during training.

3. Proposed Method

Unlike many deep learning-based methods that directly estimate the de-rained image from the noisy observation, we take a different approach in which we first estimate the rain streak component $\hat{r}$ (i.e. residual map) and then use it to estimate the de-rained image as $\hat{x} = y - \hat{r}$. We define $\hat{c}$ as the confidence score which is an uncertainty map about the estimation of $\hat{r}$. The confidence score at each pixel is a measure of how much the network is certain about the residual value computed at each pixel. Qian et al. [22] estimate an attentive map based on the rainy image using a recurrent network and then use it as a location-based information to the de-raining network. In contrast, our method combines the residual and confidence information judiciously and uses them as input to subsequent layers at higher scales. In this way, it passes the location-based rain information to the rest of the network. We estimate the residual map and its corresponding uncertainty map at three different scales, $\{\hat{r}_{\times 1}, \hat{c}_{\times 1}\}$ (at the original input size), $\{\hat{r}_{\times 2}, \hat{c}_{\times 2}\}$ (at 0.5 scale of input size), and $\{\hat{r}_{\times 4}, \hat{c}_{\times 4}\}$ (at 0.25 scale of input size).

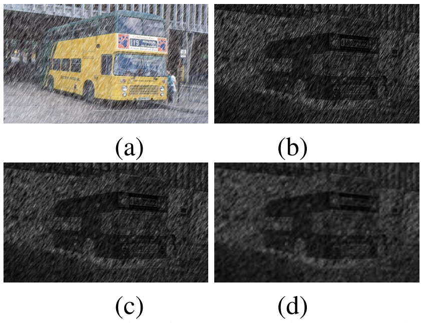

(a) (b)
(c) (d)

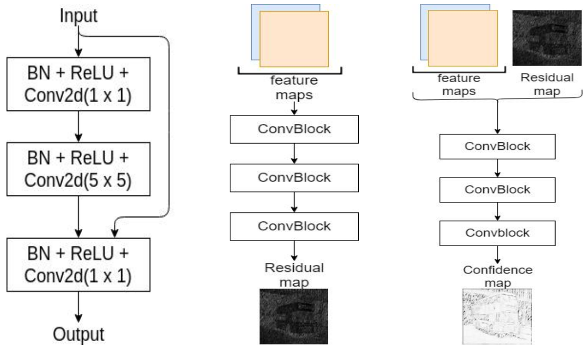
Figure 3: (a) Input rainy image, y. (b), (c), and (d) are the residual maps $r_{\times 1}$, $r_{\times 2}$, $r_{\times 4}$ at scales 1.0, 0.5 and 0.25, respectively. Note that the residual maps at different scales have the same direction and density.

Let $r_{\times 2}$ (0.5 scale size of r) and $r_{\times 4}$ (0.25 scale size of r) be the residual maps at different scales. As can be seen from Figure 3, the residual maps $r_{\times 1}$, $r_{\times 2}$, and $r_{\times 4}$ have the same direction and density at each location of the image. To estimate these residual maps, we start with the Unet architecture [24] as the base network. We use the convolutional block (ConvBlock as shown in Figure 4(a)) as the building block of our base network. The base network can be described as follows:

ConvBlock(3,32)-AvgPool-ConvBlock(32,32)-AvgPool-Convblock(32,32)-AvgPool-ConvBlock(32,32)-AvgPool-ConvBlock(32,32)-UpSample-ConvBlock(64,32)-UpSample-ConvBlock(67,32)-UpSample-ConvBlock(67,16)-ConvBlock(16,16)-Conv2d(3×3), where AvgPool is the average pooling layer, UpSample is the upsampling convolution layer, and ConvBlock(i, j) indicates ConvBlock with i input channels and j output channels. A Refinement Network(RFN) is used at the end of Unet to produce de-rained image. The Refinement Network(RFN) consists of the following blocks

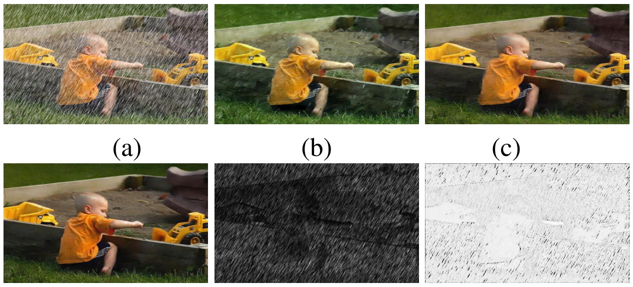
Conv2d(7×7)-Conv2d(3×3)-tanh(),

which takes $y - \hat{r}_i$ as the input and generates $\hat{x}_i$ (i.e. de-rained image) as the output. Here, Conv2d($m \times m$) represents 2D convolution using the kernel of size $m \times m$.



(a) (b) (c)

Figure 4: (a) Convolutional block (ConvBlock). BN - batchnormalization, ReLU - Rectified Linear Units, Conv2d($m \times m$) - convolutional layer with kernel of size $m \times m$. (b) Residual Network (RN). (c) Confidence map Network (CN).



(a) (b) (c)

(d) (e) (f)

Figure 5: (a) Input rainy image, y. (b) De-rained image using the base network. (c) De-rained using [35]. (d) De-rained using the proposed UMRL method. (e) The residual map. (f) The confidence map at scale 1.0($\times 1$).

3.1. UMRL Network

Rainy streaks are high frequency components and existing de-raining methods either tend to remove high frequencies that are not rain streaks or do not remove the rain near high frequency components of the clean image like edges as shown in the Figure 5. To address this issue, one can use the information about the location in image where network might go wrong in estimating the residual value. This can be done by estimating a confidence value corresponding to the estimated residual value and guide the network to remove the artifacts, especially near the edges. For example, we can observe clearly from Figure 5 that the residual map and its corresponding confidence map were able to capture the regions where there is high probability of incorrect estimates. We estimate the residual value and its corresponding confidence map at different scales (1.0($\times 1$), 0.5($\times 2$) and 0.25($\times 4$)) of the input size. This information is then fed back to the subsequent layers so that the network can learn the residual value at each location, given the computed residual value and confidence value at a lower scale.

3.1.1 Residual and Confidence Map Networks

Feature maps at different scales such as $\times 2$ and $\times 4$ are given as input to the Residual Network (RN) to estimate the residual map at the corresponding scale as shown in the Figure 2. RN consists of the following sequence of convolutional layers,

Convblock(64,32)-Convblock(32,32)-Convblock(32,3)

as shown in Figure 4(b). We use the estimated residual map and the feature maps as input to the Confidence map Network (CN) to compute the confidence measure at every pixel, which indicates how sure the network is about the residual value at each pixel. CN consists of the following sequence of convolutional layers,

Convblock(67,16)-Convblock(16,16)-Convblock(16,3)

as shown in the Figure 4(c). Given the estimated residual map and the corresponding feature maps as input to the confidence map network, it estimates $\hat{c}_{\times 4}$ and $\hat{c}_{\times 2}$. The element wise product of $\hat{r}_i$ and $\hat{c}_i$ is computed, and up-sampled to pass it as an input to the subsequent layer of the UMRL network as shown in Figure 2 for $i \in \{\times 2, \times 4\}$. Given the output residual map $r_{\times 1}$ and the feature maps of the final layer of UMRL as input to CN, we get $\hat{c}_{\times 1}$. We compute the de-rained image at different scale as

$$\hat{x}_i = \text{RFN}(y_i - \hat{r}_i), \tag{2}$$

where RFN is the Refinement Network, y_i and $\hat{x}_i$ are the input rainy image and the output de-rained image at scales, $i \in \{\times 1(1.0), \times 2(0.5), \times 4(0.25)\}$. We use the confidence guided loss and the preceptual loss to train our network.

3.1.2 Loss for UMRL

We use the confidence to guide the residual learning in the training stage of UMRL network. We define the confidence guided loss as,

$$\begin{aligned}
\mathcal{L}_l &= \sum_{i \in \{\times 1, \times 2, \times 4\}} \|(c_i \odot \hat{x}_i) - (c_i \odot x_i)\|_1, \\
\mathcal{L}_c &= \sum_{i \in \{\times 1, \times 2, \times 4\}} \Big(\sum_j \sum_k \log(c_{i_{jk}})\Big), \qquad (3) \\
\mathcal{L}_u &= \mathcal{L}_l - \lambda_1 \mathcal{L}_c,
\end{aligned}$$

where $\odot$ is the element wise product. Here, $\mathcal{L}_l$ tries to minimize the L1-norm between $\hat{x}_i$ and x_i and also the value of $c_{i_{jk}}$. On the other hand, $\mathcal{L}_c$ tries to increase $c_{i_{jk}}$ by making it close to 1. A trivial solution for $\mathcal{L}_l$ can be seen as $c_{i_{jk}} = 0 \; \forall \; i, \; j, \; k$. To avoid this, we construct $\mathcal{L}_u$ as a linear combination of $\mathcal{L}_l$ and $\mathcal{L}_c$, where $\mathcal{L}_c$ acts as a regularizer to avoid the trivial solution. Similar loss has been used for classification and regression tasks in methods [14, 15]. However, to the best of our knowledge ours is the first attempt to use this kind of loss in image restoration tasks. Inspired by the importance of the perceptual loss in many image restoration tasks [11, 33], we use it to further improve the visual quality of the de-rained images. The perceptual loss is feature based loss, and in our case, extracted features from layer $relu1\_2$ of pretrained network VGG-16[26], and computed perceptual loss similar to method proposed in [12, 32]. Let $F(.)$ denote the features obtained using the VGG16 model [26], then the perceptual loss is defined as follows

$$\mathcal{L}_p = \frac{1}{NHW} \sum_i \sum_j \sum_k \| F(\hat{x}_1)^{i,j,k} - F(x_1)^{i,j,k} \|_2^2, \quad (4)$$

where N is the number of channels of $F(.)$, H is the height and W is the width of feature maps. The overall loss used to train the UMRL network is,

$$\mathcal{L} = \mathcal{L}_l - \lambda_1 \mathcal{L}_c + \lambda_2 \mathcal{L}_p, \quad (5)$$

where λ_1 and λ_2 are two parameters.

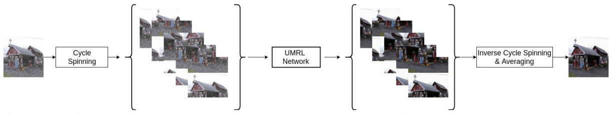


Figure 6: The idea behind cycle spinning using the UMRL network.

3.2. Cycle Spinning

As discussed earlier, cycle spinning was originally proposed to minimize the artifacts near the edges introduced by the orthogonal wavelets when de-noising images [5]. In this work, we adapt this idea to further improve the de-raining performance of UMRL. Figure 6 gives an overview of cycle spinning using UMRL. Let $T_{cs}(., p, q)$ be the function to shift an image cyclically by p rows and q columns. Given an image of size $m \times n$, we shift the image cyclically in steps of p rows and q columns to get the shifted images as shown in the Figure 6. We then de-rain the shifted images using the UMRL network, inverse shift and average them to get the final de-rained image during testing. Figure 7 shows an example of cyclically spun input images and the corresponding de-rained images. By applying cycle spinning to our method, we are able to remove some artifacts introduced by the original UMRL network. In particular, as will be shown later, cycle spinning can be applied to any CNN-based de-raining method to further improve its performance.

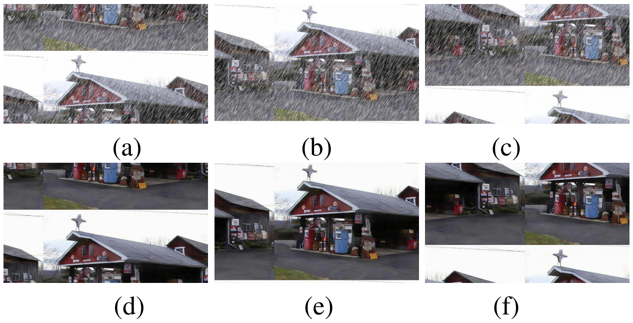

(a) (b) (c)
(d) (e) (f)

Figure 7: Cyclically spinned images with (a) $p = 100$, $q = 200$, (b) $p = 0$, $q = 200$, and (c) $p = 300$, $q = 400$. (d),(e),(f) are the corresponding de-rained images using UMRL.

4. Experimental Results

In this section, we evaluate the performance of our method on both synthetic and real images. Peak-Signal-to-Noise Ratio (PSNR) and Structural Similarity index (SSIM) [29] measures are used to compare the performance of different methods on synthetic images. We visually inspect the performance of different methods on real images, as we don't have the ground truth clean images. The performance of the proposed UMRL method is compared against several recent state-of-the-art algorithms such as (a) Gaussian mixture model (GMM) based [18] (CVPR16) (b) Fu et al.[7] CNN method (TIP'17), (c) Joint Rain Detection and Removal (JORDER) [31](CVPR17), (d) Deep detailed Network (DDN)[8] (CVPR'17) (e) Zhu et al. [37] (JBO) (ICCV17) (f) Density-aware Image Deraining method using a Multistream Dense Network (DID-MDN) [35] (CVPR'18).

4.1. Training and Testing Details

The UMRL network is trained using the synthetic image datasets created by the authors of [35, 34]. The dataset in [35] consists of 12000 images with different rain levels like low, medium and high. The dataset in [34] contains 700 training images. The (y, x) rainy-clean image pairs are shifted randomly p rows and q columns using $T_{cs}(., p, q)$ to obtain y_s, x_s, respectively. The shifted pairs (y_s, x_s) are used to train UMRL using the loss $\mathcal{L}$. The Adam optimizer with the batch size of 1 is used to train the network. Learning rate is set to 0.001 for first 10 epochs and 0.0001 for the remaining epochs. During training initially λ_1 and λ_2 are set equal to 0.1 and 1.0, respectively, but when the mean of all values in the confidences maps $\hat{c}_{\times 1}$, $\hat{c}_{\times 2}$ and $\hat{c}_{\times 4}$ is greater than 0.8 then λ_1 is set equal to 0.03. UMRL is trained for 30 epochs that is a total of 30×12700 iterations.

Similar to the previous approaches [35], we evaluate the performance of UMRL using the datasets *Test-1* containing 1200 images from [35], and *Test-2* containing 1000 images from [7]. We use the real-world rainy images pro-

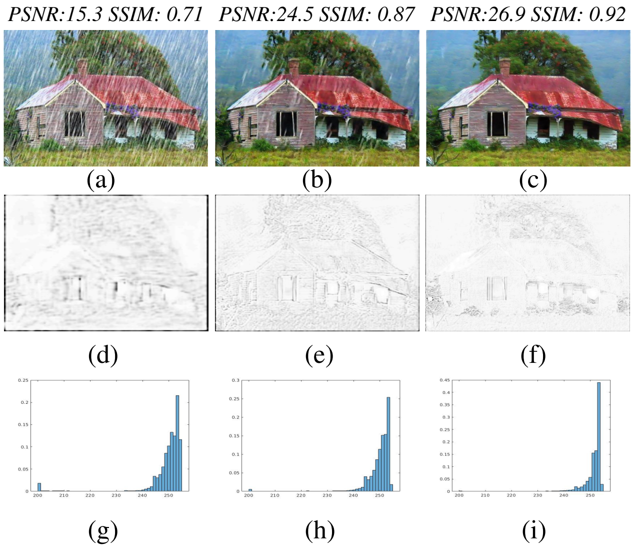


Figure 8: (a) Input rainy image. (b) De-rained image using BN + RN. (c) De-rained using BN + RN + CN (UMRL). (d),(e), and (f) are the corresponding confidence maps at scales $\times 4, \times 2, \times 1$. (g),(h), and (i) are the corresponding normalized histograms, that is sum of all bin values is equal to 1.

vided by Zhang et al.[34] and Yang et al. [31] for testing UMRL based cycle spinning method. The testing images are cyclically shifted in steps of 50 rows and 50 columns using $T_{cs}(.,.,.)$ and fed as input to UMRL for de-raining, further these de-rained images are inverse shifted and averaged to get the final output.

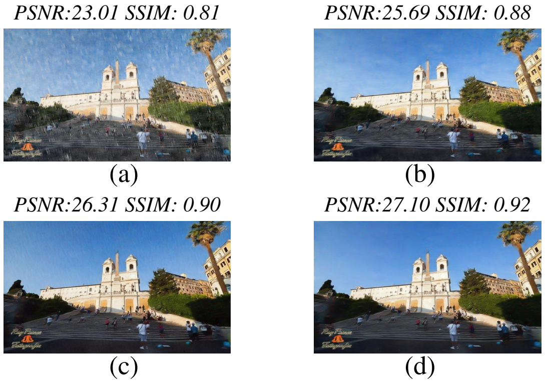


Figure 9: De-rained images using (a) DDN [8], (b) DID-MDN [35], (c) UMRL, and (d) UMRL + cycle spinning.

4.2. Ablation Study

We study the performance of each block's contribution to the UMRL network by conducting extensive experiments on the test datasets. We start with the Unet-based base network (BN) and then add one component at a time to see the significance each component brings to the network in estimating the final de-rained image. Table 1, shows the contribution of each block on the UMRL network. Note that BN and BN+RN are trained using a linear combination of L1-norm and $\mathcal{L}_p$ as loss (L1+$\mathcal{L}_p$). The UMRL is trained using the overall loss, $\mathcal{L}$. It can be seen from Table 1 that as more components (i.e RN and CN) are being added to the base network, the performance improves significantly. The base network, BN itself produces poor results. However, when RN is added to BN, the performance improves significantly. In particular, BN+RN is already able to produce results that are comparable to DID-MDN [35]. The combination of BN, RN and CN (i.e UMRL) produces the best results. Furthermore, by comparing the last two columns of Table 1 we see that cycle spinning further improves the performance of UMRL. Using cycle spinning, we are able to gain the performance improvement of approximately 0.3 dB on both datasets as it was able to remove the artifacts near edges. From the Figure 9 by zooming-in, we can clearly observe the cycle spinning is helping the method to remove small rain streaks in the sky and on the edges of building.

Table 1: PSNR and SSIM (PSNR|SSIM) results corresponding to the ablation study.

Dataset	Rainy Image	DID-MDN [35]	BN	BN+RN	BN+RN+CN (UMRL)	UMRL+ cycle spinning
Test-1	21.15|0.77	27.95|0.91	24.25|0.83	27.65|0.87	29.42|0.91	**29.77**|**0.92**
Test-2	19.31|0.77	26.08|0.90	23.32|0.83	25.88|0.87	26.47|0.91	**26.67**|**0.92**

We preformed similar experiments to see how much improvement cycle spinning brings over DDN [8] and DID-MDN [35]. In general, we observe approximately 0.25 dB gain in the performance with cycle spinning compared to without cycle spinning as shown in Table 2.

Table 2: PSNR and SSIM (PSNR|SSIM) results corresponding to the ablation study regarding the use of cycle spinning.

Dataset	Rainy Image	DDN [8]	DDN [8] + cycle spinning	DID-MDN [35]	DID-MDN [35] + cycle spinning	UMRL	UMRL+ cycle spinning
Test-1	21.15|0.77	27.33|0.90	27.52|0.91	27.95|0.91	28.19|0.91	29.42|0.91	**29.77**|**0.92**
Test-2	19.31|0.77	25.63|0.88	25.90|0.89	26.08|0.90	26.37|0.91	26.47|0.91	**26.67**|**0.92**

Figure 8 illustrates that confidence map is guiding the network to learn the rain content at the edges and texture regions clearly by imposing low confidence values. From Figure 8 by looking at the histograms of confidence maps at different scales, we can observe that as the scale is increasing the confidence values are approaching 1 at most of the pixels. This behavior is expected since at lower scales, the rain streaks will be blurry (see Figure 3) and the network is less confident about the values it estimates. This explains why UMRL tries to increase the confidence value by estimating accurate residual maps, in return CN is computing and feeding back the possible areas where UMRL goes wrong.

4.3. Results on Synthethic Test Images

The proposed UMRL method based on cycle spinning is compared against the state-of-the-art algorithms qualitatively and quantitatively. Table 3 shows the quantitative performance of our method. As it can be seen from this table, our method clearly out-performs the present state-of-the-art algorithms. Furthermore, we compare our method against a

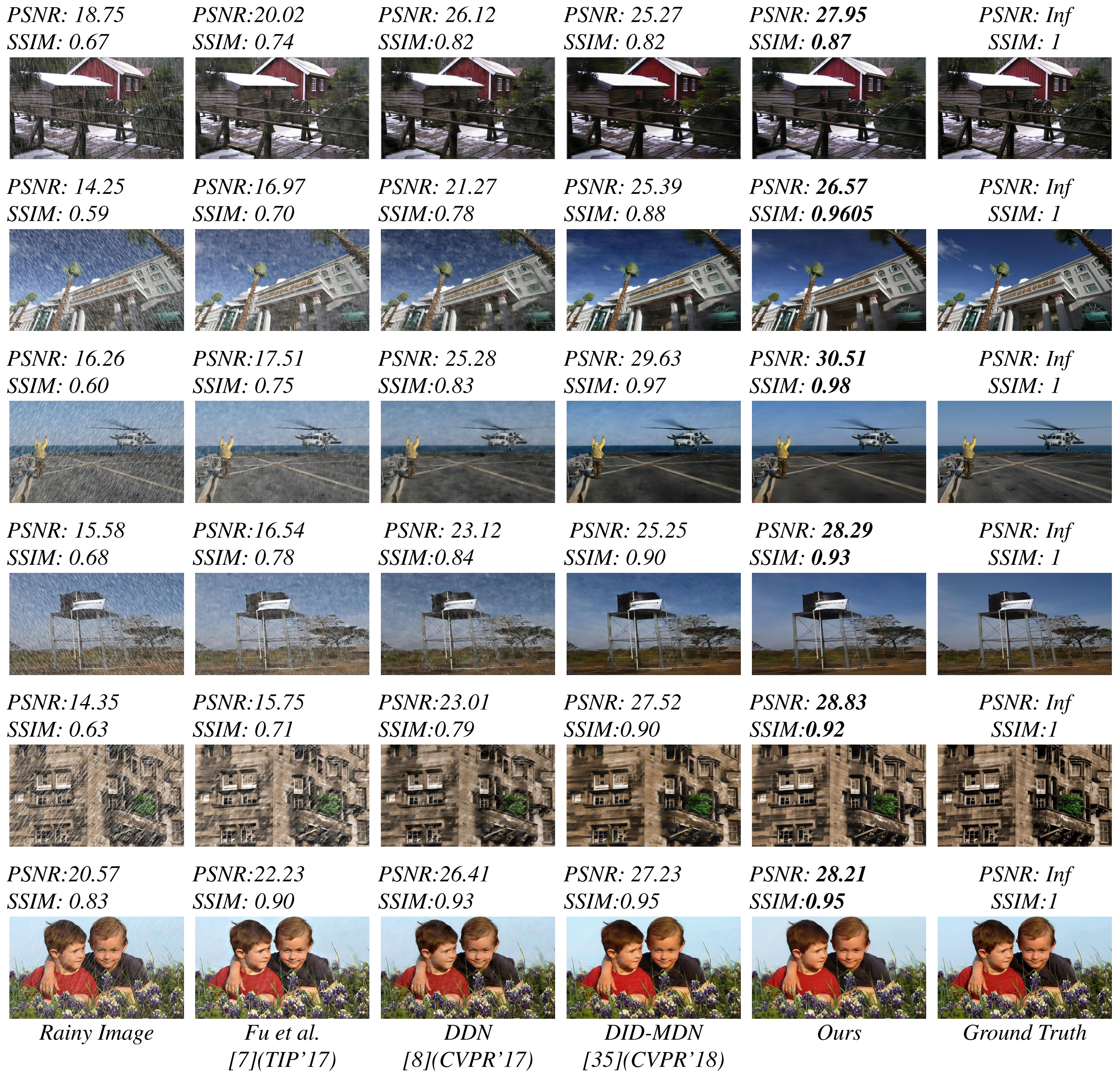


Figure 10: De-rained results on synthetic datasets *Test-1* and *Test-2* consisting different rain levels (low, medium and heavy) and different directions.

Table 3: PSNR and SSIM comparison of UMRL against state-of-art methods (PSNR|SSIM))

Dataset	Rainy Image	GMM based [18](CVPR'16)	Fu et al. [7](TIP'17)	JORDER [31](CVPR'17)	DDN [8](CVPR'17)	JBO [37](ICCV'17)	DID-MDN [35](CVPR'18)	UMRL+ cycle spinning
Test-1	21.15|0.77	22.75|0.84	22.07|0.84	24.32|0.86	27.33|0.90	23.05|0.85	27.95|0.91	**29.77**|**0.92**
Test-2	19.31|0.77	22.60|0.81	19.73|0.83	22.26|0.84	25.63|0.88	22.45|0.84	26.08|0.90	**26.67**|**0.92**

recent ECCV'18 method called REcurrent SE Context Aggregation Net (RESCAN) [17] using the *Rain800* dataset containing 100 images from [34]. The PSNR and SSIM values achieved by RESCAN [17] are 24.37 and 0.84, whereas our method achieved 24.59 and 0.87, respectively.

Figure 10 shows the qualitative performance of different methods on three sample images from *Test-1* and *Test-2* datasets. Though Fu et al. (TIP'17) [7] is able to remove some rain streaks, it is unable to remove all the rain components. DDN [8] is over de-raining on some images and on others it is slightly under de-raining as shown in the third column of Figure 10. DID-MDN [35] is over de-raining as

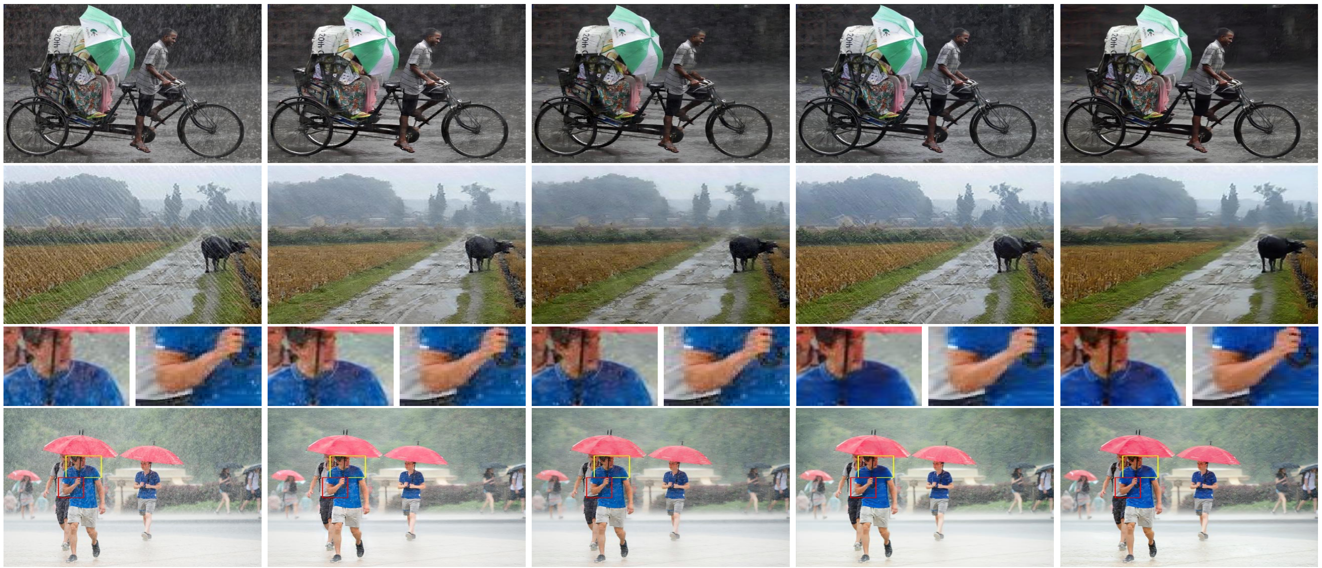


Figure 11: De-rained results on sample real-world images.

shown in the fourth column of Figure 10 where it removes the texture on wooden wall, edges of the building in second image. Furthermore, it blurs the edges of water tank in the fourth image. By comparing third and fourth images of the fourth column, we see that the outputs of DID-MDN [35] has a small blurred version of the residual streaks in the sky of those images. Visually we can see in the fifth column of Figure 10, our method produces images without any artifacts. For example in (i) it is able to recover the texture on wooden wall, in (ii) it is able to produce images with clear sky in the third and fourth images of fifth column, and in (iii) it is able to produce the sharp edges in second and fourth images.

To de-rain an image of size 512×512, on average UMRL takes about 0.05 seconds, and UMRL with cycle spinning takes about 5.1 seconds.

4.4. Results on Real-World Rainy Images

We conducted experiments on the real-world images provided by [34, 7, 35]. Results are shown in Figure 11. Similar to the results obtained on synthetic images, we observe the same trend of either over de-raining or under de-raining by the other methods. On the other hand, our method is able to remove rain streaks while preserving details of objects in the resultant output images. For example, the background and man's face in the first image of the fifth column is more clear than the outputs from other methods. Also, Trees and plants in the second image of the fifth column, front man's face and t-shirt collar in the third image are visually more clear than the results from other method. All of these experiments clearly show that our method can handle different levels of rain (low, medium and high) with different shapes and scales. More results on synthetic and real-world images are provided in the supplementary material.

5. Conclusion

We proposed a novel UMRL method based on cycle spinning to address the single image de-raining problem. In our approach, we introduced uncertainty guided residual learning where the network tries to learn the residual maps and the corresponding confidence maps at different scales which were then fed back to the subsequent layers to guide the network. In addition to UMRL, we analyzed the benefits of using cycle spinning in de-raining using various recently proposed deep de-raining networks. Extensive experiments showed that UMRL is robust enough to handle different levels of rain content for both synthetic and real-world rainy images.

Acknowledgements

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government.

References

[1] H S Bhadauria and M L Dewal. Online dictionary learning for sparse coding. *In: International Conference on Machine*

Learning(ICML), pages 689–696, 2009.

[2] H S Bhadauria and M L Dewal. Analysis of effect of cycle spinning on wavelet- and curvelet-based denoising methods on brain ct images, 2014.

[3] Duan-Yu Chen, Chien Cheng Chen, and Li Wei Kang. Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1430 – 1455, 2014.

[4] Y. Chen and C. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *2013 IEEE International Conference on Computer Vision*, pages 1968–1975, Dec 2013.

[5] R R Coifman and D L Donoho. Translation-invariant denoising. *In: Antoniadis A., Oppenheim G. (eds) Wavelets and Statistics. Lecture Notes in Statistics, vol 103. Springer, New York, NY*, pages 125–150, 1995.

[6] D. L. Donoho and M. E. Raimondo. A fast wavelet algorithm for image deblurring. In Rob May and A. J. Roberts, editors, *Proc. of 12th Computational Techniques and Applications Conference CTAC-2004*, volume 46, pages C29–C46, Mar. 2005. `http://anziamj.austms.org.au/V46/CTAC2004/Dono`.

[7] X Fu, J Huang, X Ding, Y Liao, and J Paisley. Clearing the skies a deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26:2944–2956, 2017.

[8] X Fu, J Huang, D Zeng, X Ding, Y Liao, and J Paisley. Removing rain from single images via a deep detail network. *In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1715–1723, 2017.

[9] K Garg and S K Nayar. Vision and rain. *In: International Journal of Computer Vision*, 75:3–27, 2007.

[10] D A Huang, L W Kang, Y C F Wang, and C W Lin. Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on multimedia*, 16:83–93, 2014.

[11] J Johnson, A Alahi, and L Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *In European Conference on Computer Vision(ECCV)*, pages 694–711, 2016.

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. 2016.

[13] 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*, 21:1742–1755, 2012.

[14] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017.

[15] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video rain streak removal by multiscale convolutional sparse coding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[17] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. *In: European Conference on Computer Vision(ECCV)*, pages 262–277, 2018.

[18] Y Li, R T Tan, X Guo, J Lu, and M S Brown. Rain streak removal using layer priors. *In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2736–2744, 2016.

[19] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:171–184, 2013.

[20] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming 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.

[21] Y Luo, Y Xu, and H Ji. Removing rain from a single image via discriminative sparse coding. *In:IEEE International Conference on Computer Vision(ICCV)*, pages 3397–3405, 2013.

[22] Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying 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.

[23] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.

[24] O. Ronneberger, P.Fischer, and T. Brox". U-net: Convolutional networks for biomedical image segmentation". In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)"*, volume "9351" of *"LNCS"*, pages "234–241", "2015".

[25] Varun Santhaseelan and Vijayan Asari. Utilizing local phase information to remove rain from video. *In: International Journal of Computer Vision*, 112, 2015.

[26] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[27] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *CVPR*, 2018.

[28] Y Wang, S Liu, C Chen, and B Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE Transactions on Image Processing*, 26:3936–3950, 2017.

[29] Zhou 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.

[30] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] W Yang, R T Tan, J Feng, J Liu, and S Yan. Deep joint rain detection and removal from a single image. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017.

[32] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*, 2017.

[33] H. Zhang and Vishal M Patel. Convolutional sparse and lowrank coding-based rain streak removal. *7 IEEE Winter Conference In Applications of Computer Vision(WACV)*, pages 1259–1267, 2017.

[34] He Zhang and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017.

[35] H. Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, abs/1802.07412, 2018.

[36] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. *In: IEEE International Conference on Multimedia and Expo*, pages 461–464, 2006.

[37] L Zhu, C W Fu, D Lischinski, and P A Heng. Joint bilayer optimization for single-image rain streak removal. *In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2534, 2017.