Density-aware Single Image De-raining using a Multi-stream Dense Network

He Zhang   Vishal M. Patel

Department of Electrical and Computer Engineering

Rutgers University, Piscataway, NJ 08854

{he.zhang92,vishal.m.patel}@rutgers.edu

Abstract

Single image rain streak removal is an extremely challenging problem due to the presence of non-uniform rain densities in images. We present a novel density-aware multi-stream densely connected convolutional neural network-based algorithm, called DID-MDN, for joint rain density estimation and de-raining. The proposed method enables the network itself to automatically determine the rain-density information and then efficiently remove the corresponding rain-streaks guided by the estimated rain-density label. To better characterize rain-streaks with different scales and shapes, a multi-stream densely connected de-raining network is proposed which efficiently leverages features from different scales. Furthermore, a new dataset containing images with rain-density labels is created and used to train the proposed density-aware network. Extensive experiments on synthetic and real datasets demonstrate that the proposed method achieves significant improvements over the recent state-of-the-art methods. In addition, an ablation study is performed to demonstrate the improvements obtained by different modules in the proposed method. Code can be found at: https://github.com/hezhangsprinter/DID-MDN

1. Introduction

In many applications such as drone-based video surveillance and self driving cars, one has to process images and videos containing undesirable artifacts such as rain, snow, and fog [8, 7] or other distortion such as blur and light [25]. Furthermore, the performance of many computer vision systems often degrades when they are presented with images containing some of these artifacts. Hence, it is important to develop algorithms that can automatically remove these artifacts. In this paper, we address the problem of rain streak removal from a single image. Various methods have been proposed in the literature to address this problem [19, 7, 39, 21, 3, 16, 11, 2, 41, 36, 6].
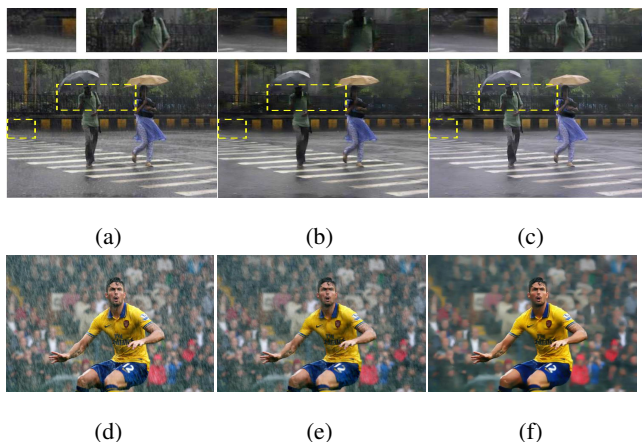
One of the main limitations of the existing single im-

Figure 1: Image de-raining results. (a) Input rainy image. (b) Result from Fu *et al*. [7]. (c) DID-MDN. (d) Input rainy image. (e) Result from Li *et al*. [36]. (f) DID-MDN. Note that [7] tends to over de-rain the image while [36] tends to under de-rain the image.

age de-raining methods is that they are designed to deal with certain types of rainy images and they do not effectively consider various shapes, scales and density of rain drops into their algorithms. State-of-the-art de-raining algorithms such as [36, 7] often tend to over de-rain or under de-rain the image if the rain condition present in the test image is not properly considered during training. For example, when a rainy image shown in Fig. 1(a) is de-rained using the method of Fu *et al*. [7], it tends to remove some important parts in the de-rained image such as the right arm of the person, as shown in Fig. 1(b). Similarly, when [36] is used to de-rain the image shown in Fig. 1(d), it tends to under de-rain the image and leaves some rain streaks in the output de-rained image. Hence, more adaptive and efficient methods, that can deal with different rain density levels present in the image, are needed.

One possible solution to this problem is to build a very large training dataset with sufficient rain conditions containing various rain-density levels with different orientations and scales. This has been achieved by Fu *et al*. [7] and

Yang *et al.*[36], where they synthesize a novel large-scale dataset consisting of rainy images with various conditions and they train a single network based on this dataset for image de-raining. However, one drawback of this approach is that a single network may not be capable enough to learn all types of variations present in the training samples. It can be observed from Fig. 1 that both methods tend to either over de-rain or under de-rain results. Alternative solution to this problem is to learn a density-specific model for de-raining. However, this solution lacks flexibility in practical de-raining as the density label information is needed for a given rainy image to determine which network to choose for de-raining.

In order to address these issues, we propose a novel Density-aware Image De-raining method using a Multi-stream Dense Network (DID-MDN) that can automatically determine the rain-density information (i.e. heavy, medium or light) present in the input image (see Fig. 2). The proposed method consists of two main stages: rain-density classification and rain streak removal. To accurately estimate the rain-density level, a new residual-aware classifier that makes use of the residual component in the rainy image for density classification is proposed in this paper. The rain streak removal algorithm is based on a multi-stream densely-connected network that takes into account the distinct scale and shape information of rain streaks. Once the rain-density level is estimated, we fuse the estimated density information into our final multi-stream densely-connected network to get the final de-rained output. Furthermore, to efficiently train the proposed network, a large-scale dataset consisting of 12,000 images with different rain-density levels/labels (i.e. heavy, medium and light) is synthesized. Fig. 1(c) & (d) present sample results from our network, where one can clearly see that DID-MDN does not over de-rain or under de-rain the image and is able to provide better results as compared to [7] and [36].

This paper makes the following contributions:

1. A novel DID-MDN method which automatically determines the rain-density information and then efficiently removes the corresponding rain-streaks guided by the estimated rain-density label is proposed.
2. Based on the observation that residual can be used as a better feature representation in characterizing the rain-density information, a novel residual-aware classifier to efficiently determine the density-level of a given rainy image is proposed in this paper.
3. A new synthetic dataset consisting of 12,000 training images with rain-density labels and 1,200 test images is synthesized. To the best of our knowledge, this is the first dataset that contains the rain-density label information. Although the network is trained on our synthetic dataset, it generalizes well to real-world rainy images.

4. Extensive experiments are conducted on three highly challenging datasets (two synthetic and one real-world) and comparisons are performed against several recent state-of-the-art approaches. Furthermore, an ablation study is conducted to demonstrate the effects of different modules in the proposed network.

2. Background and Related Work

In this section, we briefly review several recent related works on single image de-raining and multi-scale feature aggregation.

2.1. Single Image De-raining

Mathematically, a rainy image $\mathbf{y}$ can be modeled as a linear combination of a rain-streak component $\mathbf{r}$ with a clean background image $\mathbf{x}$, as follows

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

In single image de-raining, given $\mathbf{y}$ the goal is to recover $\mathbf{x}$. As can be observed from (1) that image de-raining is a highly ill-posed problem. Unlike video-based methods [26, 32, 28], which leverage temporal information in removing rain components, prior-based methods have been proposed in the literature to deal with this problem. These include sparse coding-based methods [16, 11, 47], low-rank representation-based methods [3, 39] and GMM-based (gaussian mixture model) methods [19]. One of the limitations of some of these prior-based methods is that they often tend to over-smooth the image details [16, 39].

Recently, due to the immense success of deep learning in both high-level and low-level vision tasks [10, 34, 44, 23, 35, 37, 25], several CNN-based methods have also been proposed for image de-raining [4, 6, 36, 7]. In these methods, the idea is to learn a mapping between input rainy images and their corresponding ground truths using a CNN structure.

2.2. Multi-scale Feature Aggregation

It has been observed that combining convolutional features at different levels (scales) can lead to a better representation of an object in the image and its surrounding context [9, 45, 13, 38]. For instance, to efficiently leverage features obtained from different scales, the FCN (fully convolutional network) method [20] uses skip-connections and adds high-level prediction layers to intermediate layers to generate pixel-wise prediction results at multiple resolutions. Similarly, the U-Net architecture [27] consists of a contracting path to capture the context and a symmetric expanding path that enables the precise localization. The HED model [33] employs deeply supervised structures, and automatically learns rich hierarchical representations that are fused to resolve the challenging ambiguity in edge and object boundary detection. Multi-scale features have also been
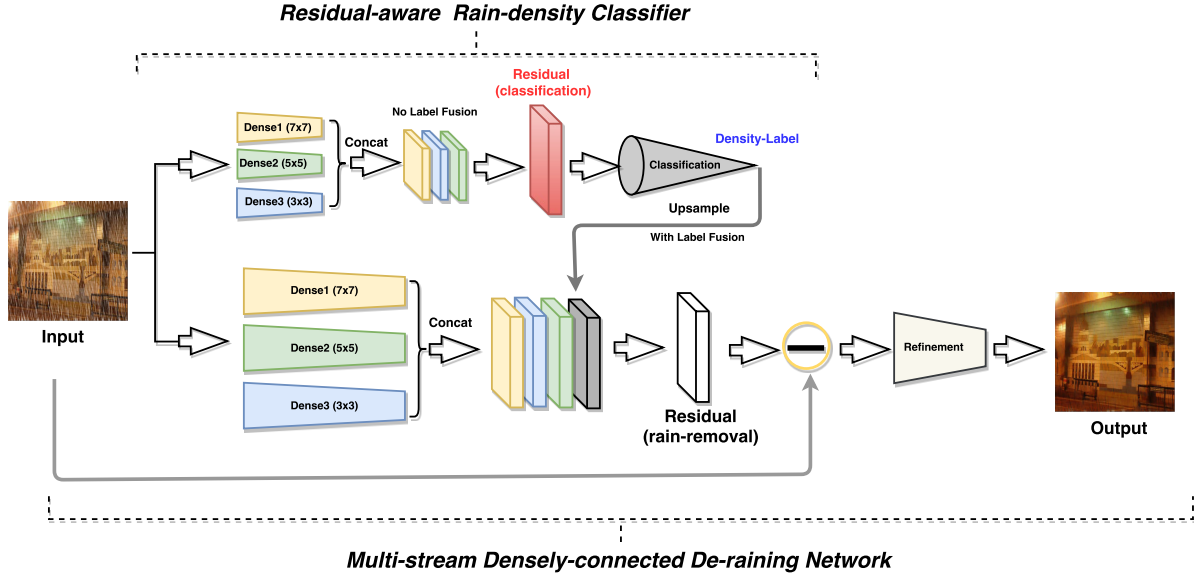
Figure 2: An overview of the proposed DID-MDN method. The proposed network contains two modules: (a) residual-aware rain-density classifier, and (b) multi-stream densely-connected de-raining network. The goal of the residual-aware rain-density classifier is to determine the rain-density level given a rainy image. On the other hand, the multi-stream densely-connected de-raining network is designed to efficiently remove the rain streaks from the rainy images guided by the estimated rain-density information.

leveraged in various applications such as semantic segmentation [45], face-alignment [22], visual tracking [18] crowd-counting [30], single image super-resolution[43], face anti-Spoofing [1], action recognition [48], depth estimation [5], single image dehazing [24, 42, 40] and also in single image de-raining [36]. Similar to [36], we also leverage a multi-stream network to capture the rain-streak components with different scales and shapes. However, rather than using two convolutional layers with different dilation factors to combine features from different scales, we leverage the densely-connected block [13] as the building module and then we connect features from each block together for the final rain-streak removal. The ablation study demonstrates the effectiveness of our proposed network compared with the structure proposed in [36].

3. Proposed Method

The proposed DID-MDN architecture mainly consists of two modules: (a) residual-aware rain-density classifier, and (b) multi-stream densely connected de-raining network. The residual-aware rain-density classifier aims to determine the rain-density level given a rainy image. On the other hand, the multi-stream densely connected de-raining network is designed to efficiently remove the rain streaks from the rainy images guided by the estimated rain-density information. The entire network architecture of the proposed DID-MDN method is shown in Fig. 2.

3.1. Residual-aware Rain-density Classifier

As discussed above, even though some of the previous methods achieve significant improvements on the de-raining performance, they often tend to over de-rain or under de-rain the image. This is mainly due to the fact that a single network may not be sufficient enough to learn different rain-densities occurring in practice. We believe that incorporating density level information into the network can benefit the overall learning procedure and hence can guarantee better generalization to different rain conditions [26]. Similar observations have also been made in [26], where they use two different priors to characterize light rain and heavy rain, respectively. Unlike using two priors to characterize different rain-density conditions [26], the rain-density label estimated from a CNN classifier is used for guiding the de-raining process. To accurately estimate the density information given a rainy input image, a residual-aware rain-density classifier is proposed, where the residual information is leveraged to better represent the rain features. In addition, to train the classier, a large-scale synthetic dataset consisting of 12,000 rainy images with density labels is synthesized. Note that there are only three types of classes (i.e. labels) present in the dataset and they correspond to low, medium and high density.

One common strategy in training a new classifier is to fine-tune a pre-defined model such as VGG-16 [29], Res-net [10] or Dense-net [13] on the newly introduced dataset. One of the fundamental reasons to leverage a fine-tune strategy for the new dataset is that discriminative features encoded

in these pre-defined models can be beneficial in accelerating the training and it can also guarantee better generalization. However, we observed that directly fine-tuning such a 'deep' model on our task is not an efficient solution. This is mainly due to the fact that high-level features (deeper part) of a CNN tend to pay more attention to localize the discriminative objects in the input image [46]. Hence, relatively small rain-streaks may not be localized well in these high-level features. In other words, the rain-streak information may be lost in the high-level features and hence may degrade the overall classification performance. As a result, it is important to come up with a better feature representation to effectively characterize rain-streaks (i.e. rain-density).

From (1), one can regard $\mathbf{r} = \mathbf{y} - \mathbf{x}$ as the residual component which can be used to characterize the rain-density. To estimate the residual component ($\hat{\mathbf{r}}$) from the observation $\mathbf{y}$, a multi-stream dense-net (without the label fusion part) using the new dataset with heavy-density is trained. Then, the estimated residual is regarded as the input to train the final classifier. In this way, the residual estimation part can be regarded as the feature extraction procedure [1], which is discussed in Section 3.2. The classification part is mainly composed of three convolutional layers (Conv) with kernel size 3×3, one average pooling (AP) layer with kernel size 9×9 and two fully-connected layers (FC). Details of the classifier are as follows:

Conv(3,24)-Conv(24,64)-Conv(64,24)-AP-
FC(127896,512)-FC(512,3),

where (3,24) means that the input consists of 3 channels and the output consists of 24 channels. Note that the final layer consists of a set of 3 neurons indicating the rain-density class of the input image (i.e. low, medium, high). An ablation study, discussed in Section 4.3, is conducted to demonstrate the effectiveness of proposed residual-aware classifier as compared with the VGG-16 [29] model.

Loss for the Residual-aware Classifier:. To efficiently train the classifier, a two-stage training protocol is leveraged. A residual feature extraction network is firstly trained to estimate the residual part of the given rainy image, then a classification sub-network is trained using the estimated residual as the input and is optimized via the ground truth labels (rain-density). Finally, the two stages (feature extraction and classification) are jointly optimized. The overall loss function used to train the residual-aware classier is as follows:

$$\mathcal{L} = \mathcal{L}_{E,r} + \mathcal{L}_C, \qquad (2)$$

where $\mathcal{L}_{E,r}$ indicates the per-pixel Euclidean-loss to estimate the residual component and $\mathcal{L}_C$ indicates the cross-entropy loss for rain-density classification.
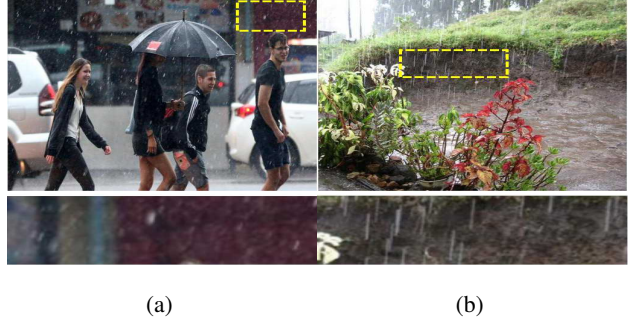

(a)   (b)

Figure 3: Sample images containing rain-streaks with various scales and shapes.(a) contains smaller rain-streaks, (b) contains longer rain-streaks.

3.2. Multi-stream Dense Network

It is well-known that different rainy images contain rain-streaks with different scales and shapes. Considering the images shown in Fig. 3, the rainy image in Fig. 3 (a) contains smaller rain-streaks, which can be captured by small-scale features (with smaller receptive fields), while the image in Fig. 3 (b) contains longer rain-streaks, which can be captured by large-scale features (with larger receptive fields). Hence, we believe that combining features from different scales can be a more efficient way to capture various rain streak components [12, 36].

Based on this observation and motivated by the success of using multi-scale features for single image de-raining [36], a more efficient multi-stream densely-connected network to estimate the rain-streak components is proposed, where each stream is built on the dense-block introduced in [13] with different kernel sizes (different receptive fields). These multi-stream blocks are denoted by Dense1 (7×7), Dense2 (5×5), and Dense3 (3×3), in yellow, green and blue blocks, respectively in Fig. 2. In addition, to further improve the information flow among different blocks and to leverage features from each dense-block in estimating the rain streak components, a modified connectivity is introduced, where all the features from each block are concatenated together for rain-streak estimation. Rather than leveraging only two convolutional layers in each stream [36], we create short paths among features from different scales to strengthen feature aggregation and to obtain better convergence. To demonstrate the effectiveness of our proposed multi-stream network compared with the multi-scale structure proposed in [36], an ablation study is conducted, which is described in Section 4.

To leverage the rain-density information to guide the de-raining process, the up-sampled label map [2] is concatenated with the rain streak features from all three streams. Then,

---

[1]Classificaiton network can be regarded as two parts: 1.Feature extractor and 2. Classifer

[2]For example, if the label is 1, then the corresponding up-sampled label-map is of the same dimension as the output features from each stream and all the pixel values of the label map are 1.

the concatenated features are used to estimate the residual ($\hat{\mathbf{r}}$) rain-streak information. In addition, the residual is subtracted from the input rainy image to estimate the coarse de-rained image. Finally, to further refine the estimated coarse de-rained image and make sure better details well preserved, another two convolutional layers with ReLU are adopted as the final refinement.

There are six dense-blocks in each stream. Mathematically, each stream can be represented as

$$\mathbf{s}_j = cat[DB_1, DB_2, ..., DB_6], \qquad (3)$$

where cat indicates concatenation, $DB_i, i = 1, \cdots 6$ denotes the output from the ith dense block, and $\mathbf{s}_j, j = 1, 2, 3$ denotes the jth stream. Furthermore, we adopt different transition layer combinations[3] and kernel sizes in each stream. Details of each stream are as follows:

Dense1: three transition-down layers, three transition-up layers and kernel size 7×7.

Dense2: two transition-down layers, two no-sampling transition layers, two transition-up layers and kernel size 5×5.

Dense3: one transition-down layer, four no-sampling transition layers, one transition-up layer and kernel size 3×3.

Note that each dense-block is followed by a transition layer. Fig 4 presents an overview of the first stream, **Dense1**.
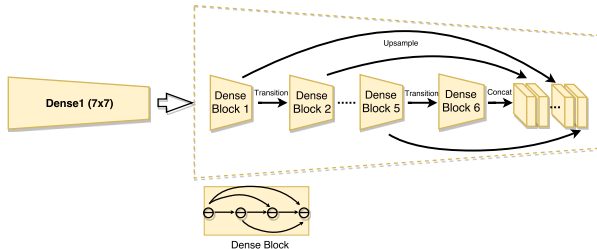

Figure 4: Details of the first stream *Dense1*.

Loss for the De-raining Network:. Motivated by the observation that CNN feature-based loss can better improve the semantic edge information [15, 17] and to further enhance the visual quality of the estimated de-rained image [41], we also leverage a weighted combination of pixel-wise Euclidean loss and the feature-based loss. The loss for training the multi-stream densely connected network is as follows

$$\mathcal{L} = \mathcal{L}_{E,r} + \mathcal{L}_{E,d} + \lambda_F \mathcal{L}_F, \qquad (4)$$

where $\mathcal{L}_{E,d}$ represents the per-pixel Euclidean loss function to reconstruct the de-rained image and $\mathcal{L}_F$ is the feature-based loss for the de-rained image, defined as

$$\mathcal{L}_F = \frac{1}{CWH} \| F(\hat{\mathbf{x}})^{c,w,h} - F(\mathbf{x})^{c,w,h} \|_2^2, \qquad (5)$$

---

[3]The transition layer can function as up-sample transition, down-sample transition or no-sampling transition [14].

where F represents a non-linear CNN transformation and $\hat{\mathbf{x}}$ is the recovered de-rained image. Here, we have assumed that the features are of size $w \times h$ with c channels. In our method, we compute the feature loss from the layer relu1_2 of the VGG-16 model [29].

3.3. Testing

During testing, the rain-density label information using the proposed residual-aware classifier is estimated. Then, the up-sampled label-map with the corresponding input image are fed into the multi-stream network to get the final de-rained image.

4. Experimental Results

In this section, we present the experimental details and evaluation results on both synthetic and real-world datasets. De-raining performance on the synthetic data is evaluated in terms of PSNR and SSIM [31]. Performance of different methods on real-world images is evaluated visually since the ground truth images are not available. The proposed DID-MDN method is compared with the following recent state-of-the-art methods: (a) Discriminative sparse coding-based method (DSC) [21] (ICCV'15), (b) Gaussian mixture model (GMM) based method [19] (CVPR'16), (c) CNN method (CNN) [6] (TIP'17), (d) Joint Rain Detection and Removal (JORDER) method [36] (CVPR'17), (e) Deep detailed Network method (DDN) [7] (CVPR'17), and (f) Joint Bi-layer Optimization (JBO) method [47] (ICCV'17).

4.1. Synthetic Dataset

Even though there exist several large-scale synthetic datasets [7, 41, 36], they lack the availability of the corresponding rain-density label information for each synthetic rainy image. Hence, we develop a new dataset, denoted as *Train1*, consisting of 12,000 images, where each image is assigned a label based on its corresponding rain-density level. There are three rain-density labels present in the dataset (e.g. light, medium and heavy). There are roughly 4,000 images per rain-density level in the dataset. Similarly, we also synthesize a new test set, denoted as *Test1*, which consists of a total of 1,200 images. It is ensured that each dataset contains rain streaks with different orientations and scales. Images are synthesized using Photoshop. We modify the noise level introduced in step 3 of [4] to generate different rain-density images, where light, medium and heavy rain conditions correspond to the noise levels $5\% \sim 35\%$, $35\% \sim 65\%$, and $65\% \sim 95\%$, respectively [5]. Sample syn-

---

[4]http://www.photoshopessentials.com/photo-effects/photoshopweather-effects-rain/

[5]The reason why we use three labels is that during our experiments, we found that having more than three rain-density levels does not significantly improve the performance. Hence, we only use three labels (heavy, medium and light) in the experiments.

Table 1: Quantitative results evaluated in terms of average SSIM and PSNR (dB) (SSIM/PSNR).

	Input	DSC [21] (ICCV'15)	GMM [19] (CVPR'16)	CNN [6] (TIP'17)	JORDER [36] (CVPR'17)	DDN [7] (CVPR'17)	JBO [47] (ICCV'17)	DID-MDN
Test1	0.7781/21.15	0.7896/21.44	0.8352/22.75	0.8422/22.07	0.8622/24.32	0.8978/ 27.33	0.8522/23.05	**0.9087/ 27.95**
Test2	0.7695/19.31	0.7825/20.08	0.8105/20.66	0.8289/19.73	0.8405/22.26	0.8851/25.63	0.8356/22.45	**0.9092/ 26.0745**

Table 2: Quantitative results compared with three baseline configurations on *Test1*.

	Single	Yang-Multi [36]	Multi-no-label	DID-MDN
PSNR (dB)	26.05	26.75	27.56	**27.95**
SSIM	0.8893	0.8901	0.9028	**0.9087**

Table 3: Accuracy of rain-density estimation evaluated on *Test1*.

	VGG-16 [29]	Residual-aware
Accuracy	73.32 %	85.15 %

- **Single**: A single-stream densely connected network (**Dense2**) without the procedure of label fusion.
- **Yang-Multi [36]**[6]: Multi-stream network trained without the procedure of label fusion.
- **Multi-no-label**: Multi-stream densely connected network trained without the procedure of label fusion.
- **DID-MDN (our)**: Multi-stream Densely-connected network trained with the procedure of estimated label fusion.

The average PSNR and SSIM results evaluated on *Test1* are tabulated in Table 2. As shown in Fig. 6, even though the single stream network and Yang's multi-stream network [36] are able to successfully remove the rain streak components, they both tend to over de-rain the image with the blurry output. The multi-stream network without label fusion is unable to accurately estimate the rain-density level and hence it tends to leave some rain streaks in the de-rained image (especially observed from the derained-part around the light). In contrast, the proposed multi-stream network with label fusion approach is capable of removing rain streaks while preserving the background details. Similar observations can be made using the quantitative results as shown in Table 2.

thesized images under these three conditions are shown in Fig 5. To better test the generalization capability of the proposed method, we also randomly sample 1,000 images from the synthetic dataset provided by Fu [7] as another testing set, denoted as *Test2*.

| Heavy | Medium | Light |

Figure 5: Samples synthetic images in three different conditions.

4.2. Training Details

During training, a 512×512 image is randomly cropped from the input image (or its horizontal flip) of size 586×586. Adam is used as optimization algorithm with a mini-batch size of 1. The learning rate starts from 0.001 and is divided by 10 after 20 epoch. The models are trained for up to 80×12000 iterations. We use a weight decay of 0.0001 and a momentum of 0.9. The entire network is trained using the Pytorch framework. During training, we set $\lambda_F = 1$. All the parameters are defined via cross-validation using the validation set.

4.3. Ablation Study

The first ablation study is conducted to demonstrate the effectiveness of the proposed residual-aware classifier compared to the VGG-16 [29] model. The two classifiers are trained using our synthesized training samples *Train1* and tested on the *Test1* set. The classification accuracy corresponding to both classifiers on *Test1* is tabulated in Table 3. It can be observed that the proposed residual-aware classifier is more accurate than the VGG-16 model for predicting the rain-density levels.

In the second ablation study, we demonstrate the effectiveness of different modules in our method by conducting the following experiments:

4.3.1 Results on Two Synthetic Datasets

We compare quantitative and qualitative performance of different methods on the test images from the two synthetic datasets - *Test1* and *Test2*. Quantitative results corresponding to different methods are tabulated in Table 1. It can be clearly observed that the proposed DID-MDN is able to achieve superior quantitative performance.

To visually demonstrate the improvements obtained by the proposed method on the synthetic dataset, results on two sample images selected from *Test2* and one sample chosen from our newly synthesized *Test1* are presented in Figure 7. Note that we selectively sample images from all three con-

---

[6]To better demonstrate the effectiveness of our proposed muli-stream network compared with the state-of-the-art multi-scale structure proposed in [36], we replace our multi-stream dense-net part with the multi-scale structured in [36] and keep all the other parts the same.
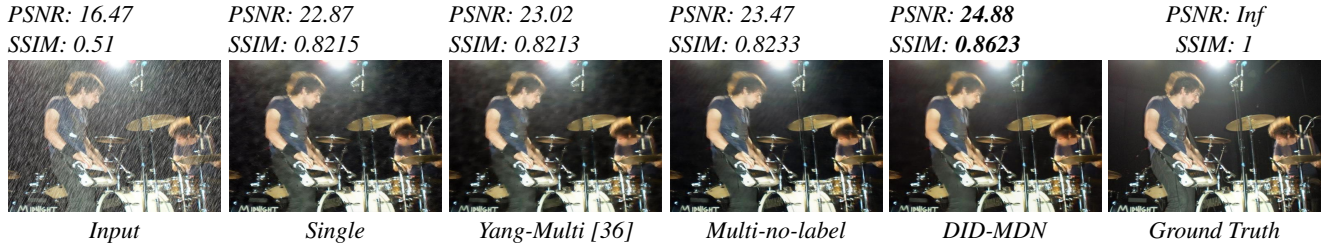
PSNR: 16.47	PSNR: 22.87	PSNR: 23.02	PSNR: 23.47	PSNR: **24.88**	PSNR: Inf
SSIM: 0.51	SSIM: 0.8215	SSIM: 0.8213	SSIM: 0.8233	SSIM: **0.8623**	SSIM: 1
Input	*Single*	*Yang-Multi [36]*	*Multi-no-label*	*DID-MDN*	*Ground Truth*

Figure 6: Results of ablation study on a synthetic image.

| PSNR: 17.27 | PSNR:21.89 | PSNR: 25.30 | PSNR: 20.72 | PSNR: **25.95** | PSNR: Inf |
| SSIM: 0.8257 | SSIM: 0.9007 | SSIM:0.9455 | SSIM: 0.8885 | SSIM: **0.9605** | SSIM: 1 |

| PSNR:19.31 | PSNR:22.28 | PSNR:26.88 | PSNR: 21.42 | PSNR: **29.88** | PSNR: Inf |
| SSIM: 0.7256 | SSIM: 0.8199 | SSIM:0.8814 | SSIM:0.7878 | SSIM:**0.9252** | SSIM:1 |

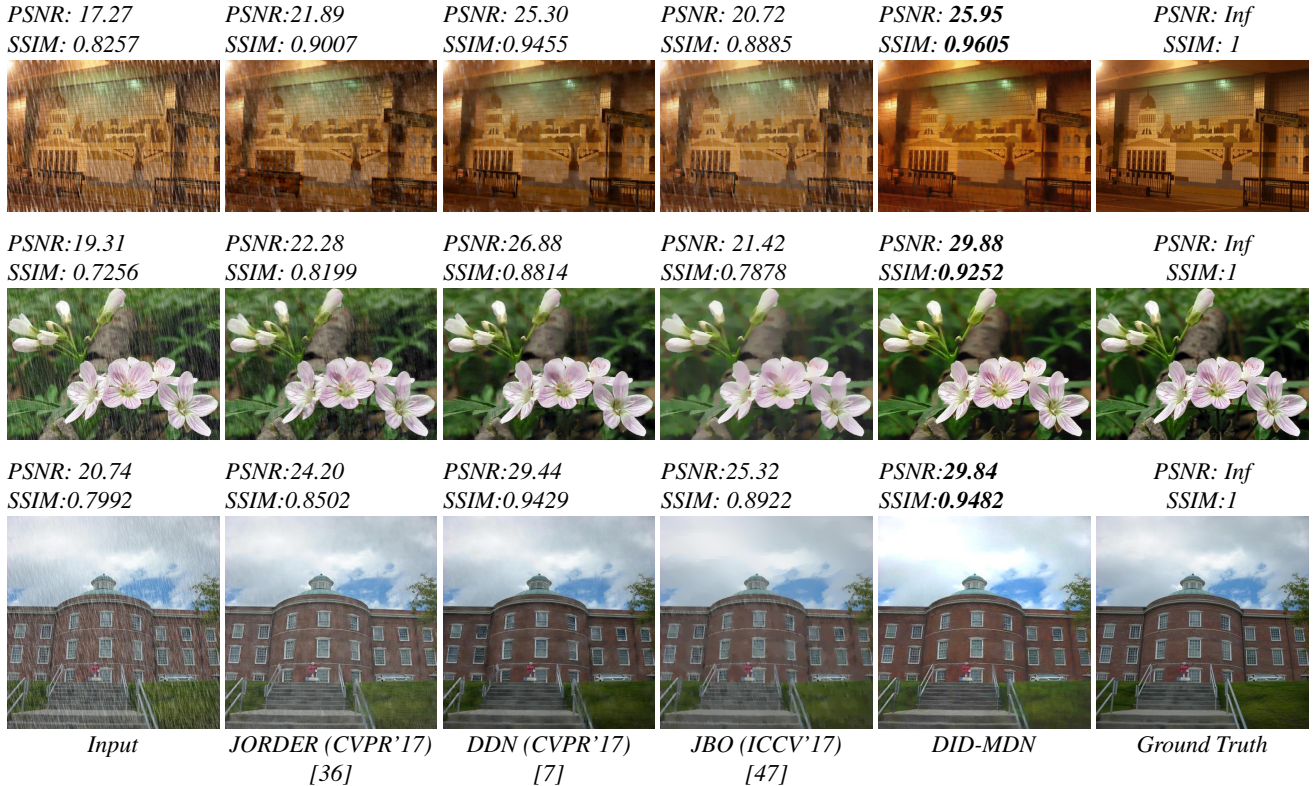| PSNR: 20.74 | PSNR:24.20 | PSNR:29.44 | PSNR:25.32 | PSNR:**29.84** | PSNR: Inf |
| SSIM:0.7992 | SSIM:0.8502 | SSIM:0.9429 | SSIM: 0.8922 | SSIM:**0.9482** | SSIM:1 |
| *Input* | *JORDER (CVPR'17)* [36] | *DDN (CVPR'17)* [7] | *JBO (ICCV'17)* [47] | *DID-MDN* | *Ground Truth* |

Figure 7: Rain-streak removal results on sample images from the synthetic datasets *Test1* and *Test2*.

ditions to show that our method performs well under different variations [7]. While the JORDER method [36] is able to remove some parts of the rain-streaks, it still tends to leave some rain-streaks in the de-rained images. Similar results are also observed from [47]. Even though the method of Fu *et al.* [7] is able to remove the rain-streak, especially in the medium and light rain conditions, it tends to remove some important details as well, such as flower details, as shown in the second row and window structures as shown in the third row (Details can be better observed via zooming-in the figure). Overall, the proposed method

is able to preserve better details while effectively removing the rain-streak components.

4.3.2 Results on Real-World Images

The performance of the proposed method is also evaluated on many real-world images downloaded from the Internet and also real-world images published by the authors of [41, 7]. The de-raining results are shown in Fig 8.

As before, previous methods either tend to under de-rain or over de-rain the images. In contrast, the proposed method achieves better results in terms of effectively removing rain streaks while preserving the image details. In addition, it can be observed that the proposed method is able to deal with different types of rain conditions, such as heavy rain

---

[7]Due to space limitations and for better comparisons, we only show the results corresponding to the most recent state-of-the-art methods [36, 7, 47].

Input	*JORDER (CVPR'17)* [36]	*DDN (CVPR'17)* [7]	*JBO (ICCV'17)* [47]	*DID-MDN*

Figure 8: Rain-streak removal results on sample real-world images.

shown in the second row of Fig 8 and medium rain shown in the fifth row of Fig 8. Furthermore, the proposed method can effectively deal with rain-streaks containing different shapes and scales such as small round rain streaks shown in the third row in Fig 8 and long-thin rain-streak in the second row in Fig 8. Overall, the results evaluated on real-world images captured from different rain conditions demonstrate the effectiveness and the robustness of the proposed *DID-MDN* method. More results can be found in *Supplementary Material*.

4.3.3 Running Time Comparisons

Running time comparisons are shown in the table below. It can be observed that the testing time of the proposed DID-MDN is comparable to the DDN [7] method. On average, it takes about 0.2s to de-rain an image of size 512×512.

Table 4: Running time (in seconds) for different methods averaged on 1000 images with size 512×512.

	DSC	GMM	CNN (GPU)	JORDER (GPU)	DDN (GPU)	JBO (CPU)	DID-MDN (GPU)
512X512	189.3s	674.8s	2.8s	600.6s	0.3s	1.4s	**0.2s**

5. Conclusion

In this paper, we propose a novel density-aware image deraining method with multi-stream densely connected network (DID-MDN) for jointly rain-density estimation and deraining. In comparison to existing approaches which attempt to solve the de-raining problem using a single network to learn to remove rain streaks with different densities (heavy, medium and light), we investigated the use of estimated rain-density label for guiding the synthesis of the de-rained image. To efficiently predict the rain-density label, a residual-aware rain-density classier is proposed in this paper. Detailed experiments and comparisons are performed on two synthetic and one real-world datasets to demonstrate

that the proposed DID-MDN method significantly outperforms many recent state-of-the-art methods. Additionally, the proposed DID-MDN method is compared against baseline configurations to illustrate the performance gains obtained by each module.

Acknowledgement

References

[1] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu. Face anti-spoofing using patch and depth-based cnns. In *In Proceeding of International Joint Conference on Biometrics*, Denver, CO, October 2017.

[2] D.-Y. Chen, C.-C. Chen, and L.-W. Kang. Visual depth guided color image rain streaks removal using sparse coding. *IEEE transactions on circuits and systems for video technology*, 24(8):1430–1455, 2014.

[3] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *IEEE ICCV*, pages 1968–1975, 2013.

[4] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *ICCV*, pages 633–640, 2013.

[5] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, 2014.

[6] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017.

[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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1715–1723, July 2017.

[8] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Trans. on PAMI*, 33(12):2341–2353, 2011.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] 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(1):83–93, 2014.

[12] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang. Context-aware single image rain removal. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 164–169. IEEE, 2012.

[13] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[14] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1175–1183. IEEE, 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*, pages 694–711. Springer, 2016.

[16] L.-W. Kang, C.-W. Lin, and Y.-H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE TIP*, 21(4):1742–1755, 2012.

[17] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2017.

[18] K. Li, Y. Kong, and Y. Fu. Multi-stream deep similarity learning networks for visual tracking. In *IJCAI*, 2017.

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

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[21] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015.

[22] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas. A recurrent encoder-decoder network for sequential face alignment. In *European Conference on Computer Vision*, pages 38–56. Springer International Publishing, 2016.

[23] X. Peng, X. Yu, K. Sohn, D. Metaxas, and M. Chandraker. Reconstruction for feature disentanglement in pose-invariant face recognition. In *ICCV*, 2017.

[24] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, pages 154–169. Springer, 2016.

[25] W. Ren, J. Pan, X. Cao, and M.-H. Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1077–1085, 2017.

[26] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang. Video desnowing and deraining based on matrix decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4210–4219, 2017.

[27] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[28] V. Santhaseelan and V. K. Asari. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision*, 112(1):71–89, 2015.

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

[30] V. A. Sindagi and V. M. Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *ICCV*, 2017.

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[32] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2516–2525, 2017.

[33] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[34] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

[35] J. Xue, H. Zhang, K. Dana, and K. Nishino. Differential angular imaging for material recognition. In *CVPR*, 2017.

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

[37] H. Zhang and K. Dana. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*, 2017.

[38] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[39] H. Zhang and V. M. Patel. Convolutional sparse and low-rank coding-based rain streak removal. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 1259–1267. IEEE, 2017.

[40] H. Zhang and V. M. Patel. Densely connected pyramid dehazing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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

[42] H. Zhang, V. Sindagi, and V. M. Patel. Joint transmission map estimation and dehazing using deep networks. *arXiv preprint arXiv:1708.00581*, 2017.

[43] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual Dense Network for Image Super-Resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[44] Z. Zhang, Y. Xie, F. Xing, M. McGough, and L. Yang. Mdnet: A semantically and visually interpretable medical image diagnosis network. In *CVPR*, 2017.

[45] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2017.

[46] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[47] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng. Joint bilayer optimization for single-image rain streak removal. In *Proceedings of the IEEE international conference on computer vision*, pages 2526–2534, 2017.

[48] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann. Hidden two-stream convolutional networks for action recognition. *arXiv preprint arXiv:1704.00389*, 2017.