This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Removing Raindrops and Rain Streaks in One Go

Ruijie Quan[1,2], Xin Yu[2], Yuanzhi Liang[1], Yi Yang[2]
[1]Baidu Research* [2]ReLER Lab, University of Technology Sydney
ruijie.quan@student.uts.edu.au v_liangyuanzhi@baidu.com {xin.yu,yi.yang}@uts.edu.au

(a) Input (b) AttentGAN→PReNet (c) PReNet → AttentGAN (d) Ours

Figure 1: Joint removal of raindrops and rain streaks. Combining a raindrop removal method AttentGAN [30] and a rain streak removal method PReNet [35] cannot remove rain thoroughly. Our method achieves promising derained results.

Abstract

Existing rain-removal algorithms often tackle either rain streak removal or raindrop removal, and thus may fail to handle real-world rainy scenes. Besides, the lack of real-world deraining datasets comprising different types of rain and their corresponding rain-free ground-truth also impedes deraining algorithm development. In this paper, we aim to address real-world deraining problems from two aspects. First, we propose a complementary cascaded network architecture, namely CCN, to remove rain streaks and raindrops in a unified framework. Specifically, our CCN removes raindrops and rain streaks in a complementary fashion, i.e., raindrop removal followed by rain streak removal and vice versa, and then fuses the results via an attention based fusion module. Considering significant shape and structure differences between rain streaks and raindrops, it is difficult to manually design a sophisticated network to remove them effectively. Thus, we employ neural architecture search to adaptively find optimal architectures within our specified deraining search space. Second, we present a new real-world rain dataset, namely RainDS, to prosper the development of deraining algorithms in practical scenarios. RainDS consists of rain images in different types and their corresponding rain-free ground-truth, including rain streak only, raindrop only, and both of them. Extensive experimental results on both existing benchmarks and RainDS demonstrate that our method outperforms the state-of-the-art.

1. Introduction

Rainy weather would severely degrade the performance of outdoor vision systems. Rain streaks in the air severely impair the visibility of captured scenes. Concurrently, raindrops falling on camera lenses or windshields further reduce the image quality as images are captured through raindrenched glasses. Hence, removing rain from images plays an important role in outdoor vision applications, such as autonomous driving.

Existing deraining works have achieved promising progresses and they can be divided into two major categories: rain streak removal and raindrop removal. Rain streak removal methods [29, 42, 41, 53, 54, 44, 11, 5, 9] remove rain streaks mainly based on their sparse line-shape nature. Existing raindrop removal methods [48, 12, 6, 30, 33] remove raindrops by identifying their various shapes, positions and sizes. Previous deraining methods usually assume only one type of rain exhibits in images. However, in real-world rainy weather, rain streaks and raindrops often co-occur during image capture. Therefore, this phenomenon poses a critical challenge to the existing deraining algorithms.

In this paper, we develop a novel complementary cascaded network, dubbed CCN, to remove raindrops and rain streaks jointly in a unified framework. Our CCN consists of two branches to remove rain in different orders, *i.e.*, raindrop removal followed by rain streak removal and rain streak removal followed by raindrop removal. In this complementary fashion, our network removes both types of rain more thoroughly, as illustrated in Fig. 1. Moreover, we present an attention based fusion to merge the outputs from the two branches to achieve satisfactory deraining results.

Considering different shapes, sizes and optical effects of

*This work was done when Ruijie Quan, Yuanzhi Liang interned at Baidu Research. Yi Yang is the corresponding author.

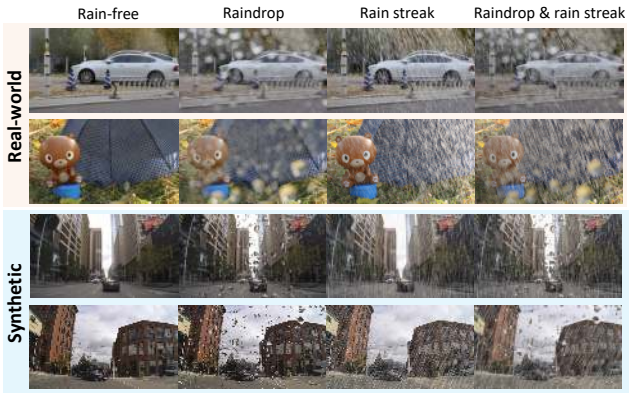

Figure 2: **Top:** Examples of real-world image pairs in RainDS. **Bottom:** Examples of the synthetic image pairs in RainDS. (Best view on screen)

rain, it might be difficult to manually design a powerful network that is able to remove different types of rain. Therefore, we resort to a neural architecture search (NAS) method to find an optimal architecture for deraining. In particular, we also design a specific deraining search space that takes several effective deraining operations into account, such as cascaded convolutions with large- and small-size kernels [36] and spatial attention modules [8]. After architecture search, our network can fully explore global and local information of the raindrops and rain streaks and restore clean images.

In order to train CCN and enable it to generalize well on real images, we manage to collect a real-world rain dataset, dubbed RainDS, including images of rain streaks, raindrops and both of them as well as their corresponding clean images[1]. Moreover, to enrich the diversity of our dataset, we include some synthetic data generated in autonomous driving scenes. Examples of RainDS are illustrated in Fig. 2. The training set of RainDS includes both real and synthetic data. In this way, we can not only train our network on real data to reduce the domain gap between real and synthetic data but also evaluate it on real data quantitatively. Experimental results on RainDS demonstrate that our method outperforms the state-of-the-art in real-world scenarios, thus making our method more favorable.

Overall, our contributions are summarized as follows:

- We propose a new complementary cascaded deraining network, named CCN, to simultaneously remove both raindrops and rain streaks in a complementary manner, thus removing rain more thoroughly.
- We design a specific deraining search space that takes different rain characteristics into account, and then search an optimal architecture within this space for our generic deraining task.
- To the best of our knowledge, our introduced dataset RainDS is the first real-world deraining dataset including different types of rain captured in various lighting conditions and scenes. RainDS significantly facilitates bridging the domain gap between real and synthetic data and improving the model generalization ability.
- Our method achieves state-of-the-art performance on both existing datasets (only rain streak or raindrop) and our proposed benchmark.

[1] All the images are captured by a Cannon EOS D60 camera in real life.

2. Related Work

In the past few years, rain image restoration techniques achieved substantial progresses. As we focus on single image based rain removal, including rain streaks and raindrops, the most related literature will be reviewed.

2.1. Rain removal

Rain streak removal. Single image based rain streak removal methods can be categorized into two groups: prior-based and deep learning-based approaches.

Prior-based methods: Kang *et al*. [20] clustered rain and non-rain dictionaries based on the histogram of oriented gradients (HOGs) features of rain streaks as priors and then reconstructed clean images. Luo *et al*. [29] proposed a dictionary learning based approach to separate the background layer from the rain one. Then Li *et al*. [25] introduced Gaussian mixture model based patch priors to model different orientations and scales of rain streaks and then remove them from rain images. Later, Zhu *et al*. [55] fully exploited three priors, *i.e.*, a centralized sparse representation, a dominant rain streak direction and patch similarity, to extract rain streaks for removal. As rain streaks exhibit obvious line patterns, Chang *et al*. [1] developed a low-rank image decomposition framework for rain streaks removal. Deng *et al*. [3] removed rain streaks by taking the intrinsic structural and directional knowledge of rain streaks into account.

Deep learning-based methods: Recently, deep learning based methods [41, 42, 40, 35, 21, 45, 38, 7, 2] have demonstrated their superiority in removing rain streaks. Yang *et al*. [43] decomposed a rain layer into a series of sub-layers representing rain streaks of different directions, and then removed rain streaks by a recurrent network. Fu *et al*. [10] decomposed a rain image into a low-frequency structure layer and a high-frequency detail layer and then removed rain streaks from the detail layer by a convolutional neural network. More recently, Yasarla *et al*. [46] proposed a Gaussian process based semi-supervised learning framework to remove rain using unlabeled real-world images. Jiang *et al*. [19] proposed a multi-scale collaborative representation to remove rain streaks. Deng *et al*. [4] introduced two parallel sub-networks that synergize to derain and recover lost details caused by deraining. Wang *et al*. [39] proposed a

model-driven deep neural network and utilized the proximal gradient descent to optimize the deraining network.

Raindrop removal. Raindrop removal is also a challenging task as adherent raindrops have many variations in terms of shapes, positions, and sizes. Existing approaches [6, 47, 48, 30, 33] focus on either detecting or removing adherent raindrops. Eigen *et al*. [6] firstly adopted a deep neural network for raindrop removal. You *et al*. [48] used rich temporal information to remove raindrops from videos. However, their method cannot be applied in single image based deraining. Qian *et al*. [30] proposed a generative adversarial network with attention modules to remove raindrops. Quan *et al*. [33] introduced a dual attention mechanism, *i.e.*, a shape-driven attention and a channel re-calibration, to remove effects of raindrops.

Existing deraining methods address either of rain streaks or raindrops using specific networks. Those networks might not be suitable for generic rain removal. In this work, we propose a unified network to remove rain streaks and raindrops simultaneously. Recently, Wang *et al*. [40] proposed a real-world rain streak removal dataset. In their dataset, ground-truth clean images are computed by a video deraining method and image pairs are extracted from local regions ($\sim$2.9K patches). In contrast, our dataset contains different combinations of rain and provides entire images.

2.2. Neural architecture search

Neural architecture search aims to design neural network architectures automatically and allows a searched network to achieve optimal performance. Recently, gradient-based NAS methods have been applied in many computer vision tasks [15, 49, 32, 37, 14]. To the best of our knowledge, there are two works using NAS for rain removal. Specifically, Qin *et al*. [31] employed NAS to search an architecture to remove either rain streaks or raindrops. In other words, Qin *et al*.'s method still addresses raindrops and rain streaks individually. Li *et al*. [22] employed separate encoders to extract features from different bad-weather images and used a shared decoder to reconstruct clean images. However, Li *et al*. did not consider the co-occurrence of rain streaks and raindrops in an image.

Inspired by these works, we propose a new neural architecture search method so as to search optimal sub-network architectures for removal of rain streaks and raindrops, respectively. In particular, we design a new search space that fully exploits the attributes of rain components, such as visibility in different scales and local and global similarities.

3. Proposed Method

In order to remove rain streaks and raindrops in one go, we propose a novel complementary cascaded deraining network. In this section, we first introduce the formulation of rain streaks and raindrops in Sec. 3.1. Then, we present our unified deraining framework, *i.e.*, our complementary cascaded network architecture, in Sec. 3.2. Moreover, we introduce our designed search space and employ NAS to find an optimal network architecture for removing different types of rain simultaneously in Sec. 3.3.

3.1. Formulation of rain streaks and raindrops

Rain streak. A rain streak image R_s is defined as the addition of a clean background scene B and accumulated rain streaks S:

$$R_s = B + S. \tag{1}$$

Rain streaks S impair scene visibility of the background scene B. Therefore, we aim to obtain the clean image B by removing rain streaks S.

Raindrop. A raindrop-drenched image R_d can be disassembled into a clean background B and blurry or obstruction effects of the raindrops D in scattered small regions [30, 23]:

$$R_d = (1 - M) \odot B + D, \tag{2}$$

where M is a binary mask and $\odot$ indicates element-wise multiplication. $M(x) = 1$ when pixel x belongs to a raindrop region, and otherwise $M(x) = 0$ indicates x belongs to background regions. Raindrop removal aims to obtain the rain-free image B by removing raindrops D.

Rain streak and raindrop. In real-world rainy weather scenarios, rain streaks and raindrops may co-occur during outdoor image capture. A rain streak and raindrop image R_{ds} can be modeled as:

$$R_{ds} = (1 - M) \odot (B + S) + \eta D, \tag{3}$$

where η is the global atmospheric lighting coefficient. Note that rain streaks may change the lighting conditions that significantly affect the transparency of raindrops during image capture. Because of this, removing rain streaks and raindrops cannot be simply cast as the combination of a rain streak removal and a raindrop removal.

3.2. Unified deraining network

As indicated in Eq. (3), rain streaks and raindrops mutually affect each other. Therefore, we develop a novel complementary cascaded network architecture, named CCN, to remove rain streaks and raindrops in a unified framework. Figure 3 illustrates our framework.

3.2.1 Deraining blocks

Rain streak and raindrop removal require different network architectures as they have different physical characteristics. For instance, rain streaks are line-shaped while raindrop

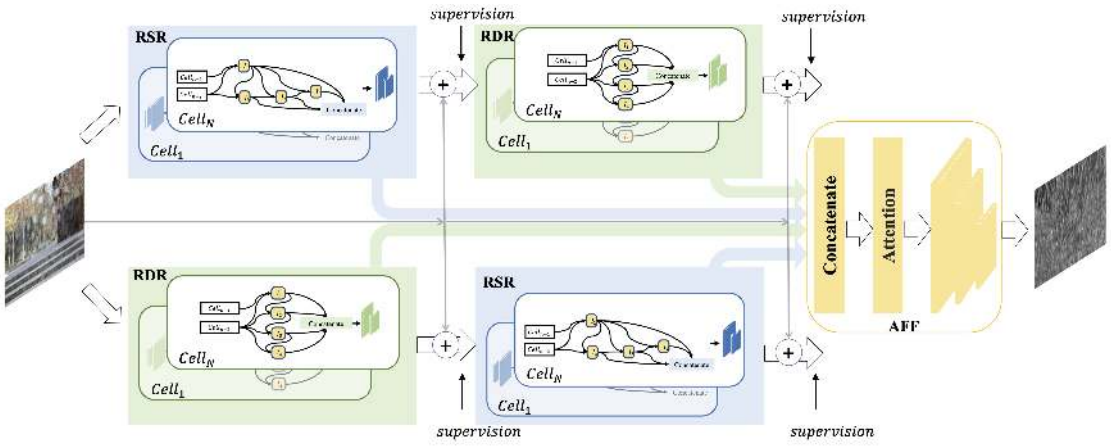

Figure 3: Overview of our complementary cascaded based deraining network, named CCN. CCN consists of rain streak removal (RSR), raindrop removal (RDR) and attention based feature fusion (AFF) blocks. Their architectures are searched by a neural architecture search method [27]. After block structures are found, CCN is trained in an end-to-end fashion.

shapes are elliptical. Hence, we propose a rain streak removal block (RSR) and a raindrop removal block (RDR) for removing rain collaboratively. However, it might be difficult to manually design a powerful yet compact block architecture for the above two blocks since a block with too many layers will make the entire network overweight, which cannot be deployed in practice, and a shallow block may not fully remove rain. Moreover, it is hard to know whether a network architecture is effective for rain streak removal or raindrop removal before training. We thus use NAS to find optimal architectures for RSR and RDR blocks. The block architecture search method will be introduced in Sec. 3.3.

3.2.2 Two-branch cascaded network

We further design a two-branch network equipped with the cascaded deraining blocks to remove rain in different orders. As shown in Fig. 3, the top branch removes rain streaks first and then raindrops while the bottom branch tackles raindrop removal first and then rain streak removal. Specifically, the RSR block in the top branch receives the input rain image (including rain streaks or both rain streaks and raindrops), while the input of the RSR block in the bottom branch is a raindrop-free image processed by its preceding RDR block. Similarly, RDR blocks learn to remove the raindrop layers from both the origin input image in the bottom branch and a rain streak-free image produced by the RSR block in the top branch. In such a complementary fashion, our network will be aware of the effects of both types of rain and then remove rain more thoroughly.

3.2.3 Attention feature fusion

Since the complementary information has been learned by RSR and RDR blocks from two branches, we further propose an attention based feature fusion (AFF) module to obtain a better representation of the rain layer. The basic structure of the proposed AFF is shown in the right side of Fig. 3. AFF takes four penultimate layer features of the four rain removal blocks as inputs, and then concatenates them into a feature. We apply three cascaded layers (*i.e.*, $conv$-bn-$relu$) to the concatenated feature and then exploit an SE-ResNet module [16] as a channel-wise attention to enhance rain layers. At last, we apply a convolutional layer to acquire the final learned rain layer.

3.3. Optimal network block search

Preliminary. We employ a gradient-based architecture search strategy [27] in our block search process and search for the topology structure of neural cells in our blocks. A neural cell is regarded as a directed acyclic graph (DAG) with N nodes. The i-th node in the n-th neural cell (1) receives two tensors (${}^1\mathcal{I}_i^n$ and ${}^2\mathcal{I}_i^n$) as inputs, where ${}^1\mathcal{I}_i^n$ and ${}^2\mathcal{I}_i^n$ represent the output tensors from the previous two neural cells, *i.e.*, $(n-1)$-th cell and $(n-2)$-th cell, or previous two nodes *i.e.*, $(i-1)$-th node and $(i-2)$-th node, in the current cell, (2) applies two operations (${}^1\mathcal{O}_i^n$ and ${}^2\mathcal{O}_i^n$) on those two tensors respectively, and (3) sums those two tensors. Therefore, an output tensor $\mathcal{I}_i^n$ of the i-th node in the n-th neural cell is formulated as:

$$\mathcal{I}_i^n = {}^1\mathcal{O}_i^n({}^1\mathcal{I}_i^n) + {}^2\mathcal{O}_i^n({}^2\mathcal{I}_i^n). \quad (4)$$

The applied operations ${}^1\mathcal{O}_i^n$ and ${}^2\mathcal{O}_i^n$ are selected from a specific deraining search space $\mathcal{O}_{rain}$. Following [27], we relax the categorical choices of a particular operation by a softmax over all possible operations:

$${}^1\mathcal{O}_i^n({}^1\mathcal{I}_i^n) = \sum_{M \in \mathcal{I}_i^c} \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(M,i)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(M,i)})} o(M), \quad (5)$$

where α=$\alpha_o^{(M,i)}$ indicates the topology structure for our neural cell (architecture parameters). The parameters of all operations (weight parameters) in our proposed search space are denoted as ω. During searching, we alternately train network architecture parameters α and weight parameters ω, and we also adopt an early stop search strategy which has been proved effective in [50].

Deraining search space. The search space consists of all possible candidate network architectures. A standard search space in NASNet [31] includes multiple standard convolutional layers and different pooling layers. However, those standard operations especially the pooling layers are not suitable for the rain removal task because the output and input should have the same resolution and image spatial details are very important. Hence, we design a specific deraining search space $\mathcal{O}_{rain}$ consisting of the following operations: (1) *cascaded convolution* $5 \circ 3$, (2) *cascaded convolution* $7 \circ 5$, (3) *cascaded convolution* $11 \circ 7$, (4) *cascaded dilated convolution* $7 \circ 5$, (5) *cascaded dilated convolution* $11 \circ 5$, (6) *spatial attention module*, (7) zero operation, and (8) identity mapping. As seen in Fig. 4, cascaded convolution $k1 \circ k2$ indicates a pair of convolutional layers with large-size ($k1$) and small-size ($k2$) kernels. The cascaded convolutions have been proved effective in the noise removal task [36, 28]. Additionally, a spatial attention module [8] is used to capture the spatial contextual information, thus facilitating rain feature learning.

Search algorithm. The macro structure of our rain removal block is illustrated in Fig. 3. Apart from several convolutional layers as the image encoder and decoder, neural cells are stacked in the middle of our rain removal block. We search for the topology structure of a neural cell for different types of rain, and then stack it into a rain removal block. In the optimization process, we apply two losses:

$$\mathcal{L} = -\mathcal{L}_{ssim} + \mathcal{L}_{\ell 1}, \tag{6}$$

where $\mathcal{L}_{ssim}$ and $\mathcal{L}_{\ell 1}$ indicate pxiel-wise SSIM objective and ℓ_1 loss between an reconstructed image and its ground-truth counterpart, respectively.

3.4. Training details

As shown in Fig. 3, given an input rain image R, our rain streak removal block RSR=$\mathcal{S}(\cdot)$ and our raindrop removal block RDR=$\mathcal{D}(\cdot)$ remove rain in a sequential fashion. In the top branch, we firstly obtain feature maps of rain streaks r_1=$\mathcal{S}_u(R)$, and then extract raindrop layers r_2=$\mathcal{D}_u(\mathcal{S}_u(R))$ from the rain streak-free features processed by the preceding RSR block. Similarly, in the bottom branch, we firstly achieve a raindrop-free features r_3=$\mathcal{D}_l(R)$ and then estimate the rain streak layers r_4=$\mathcal{S}_l(\mathcal{D}_l(R))$. The subscripts u, l of $\mathcal{S}$ and $\mathcal{D}$ indicate the top and bottom branches, respectively In training our unified network, the negative SSIM loss is applied to each block:

$$^t\mathcal{L}_{ssim} = -\mathcal{L}_{ssim}(r_t, B^t), \tag{7}$$

where B^t represents the corresponding rain streak-free, raindrop-free, or rain-free image.

The topology structures of the two rain removal blocks are searched by a NAS method on rain streak data and raindrop data, respectively. Specifically, the same kind of rain blocks (RSR or RDR) in different branches have the same architecture but do not share parameters. Their parameters are optimized according to different input data, resulting in the complementary learning fashion. Finally, we fuse the results from different rain blocks of the two branches to further improve deraining results. Denote $\mathcal{S}^{pre}(R)$ and $\mathcal{D}^{pre}(R)$ as the features from the penultimate layers of the RSR and RDR blocks in our network. Thus, our AFF module takes $\{\mathcal{S}_u^{pre}(R), \mathcal{D}_u^{pre}(\mathcal{S}_u(R)), \mathcal{D}_l^{pre}(R), \mathcal{S}_l^{pre}(\mathcal{D}_l(R))\}$ as inputs and outputs the final rain layer r_5 including both rain streak and raindrop layers. We use the same loss in Eq. (6) to supervise the final output r_5, denoted as $\mathcal{L}_{out}$. The final objective to train our network is expressed as:

$$\mathcal{L}_{train} = \sum_{t=1}^{4} {}^t\mathcal{L}_{ssim} + \mathcal{L}_{out}. \tag{8}$$

Table 1: Summary of existing deraining datasets.

Dataset		Images	Real/Syn	Rain Category
Rain12 [25]		12	syn	rain streak
Rain200H [43]		2,000	syn	rain streak
Rain800 [52]		800	syn	rain streak
Rain1200 [51]		12,000	syn	rain streak
RainDrop [30]		2,238	syn	raindrop
RainDS	Real-world	1,000	real	rain streak & raindrop,
	Synthetic	4,800	syn	rain streak & raindrop

4. RainDS Dataset

In order to train the proposed CCN network and enable it to generalize well in real scenes, we manage to collect a real-world rain dataset, named RainDS, including numerous image pairs in various lighting conditions and different scenes. Each pair contains four images: a rain streak image, a raindrop image, and an image including both types of rain, as well as their rain-free counterparts. To further enrich the diversity of our dataset, we also incorporate a synthetic subset generated in autonomous driving scenes. An overview of our RainDS dataset can be found in Table 1.

4.1. Synthetic Data

As image deraining is highly demanded in autonomous driving, we generate synthetic rainy data based on the first-person view driving scenes collected from two public autonomous driving datasets, *i.e.*, PIE [34] and KITTI [13].

As shown in Table 1, our synthetic dataset consists of 1.2k image pairs, a total of 4.8k images. In our synthetic data, there are rich driving scenes including urban roads, streets, and highways. To simulate the real-world scenarios as much as possible, we carefully control the densities, main directions, brightness, and lengths of rain streaks to be various. Also, raindrops exhibit large variances in terms of their shapes, densities, sizes and locations.

4.2. Real-world Data

Although efforts have been made to simulate rain streaks and raindrops in our synthetic data, there still exists a domain gap between the synthetic and the real data. For example, raindrops in synthetic data are always transparent but actually their transparency and brightness will change along with different lighting conditions and background scenes. Hence, we construct a subset of real-world data that contains 250 image pairs, 1,000 captured images in total. We use a DSLR camera to capture rain images. To be specific, we insert a piece of glass with adherent waterdrops in front of the camera to mimic the raindrop cases. We spray water using sprinklers to generate rain streaks, which is a widely-used technique to mimic rainy scenes in Hollywood film industry. We also carefully control exposure time and the ISO parameter to capture different lengths of rain streaks in different illumination conditions. By removing the glass with raindrops and stop spraying water, we obtain the clean background images. Our real-world rain images are captured in various scenes, including parking lots, parks and urban areas. Moreover, we collect the data at different time, *i.e.*, morning, noon, and afternoon, to obtain different lighting conditions in real scenarios. Therefore, such complex scenes and backgrounds make our real-world data subset more challenging and desirable.

5. Experiments

In this section, we evaluate our CCN deraining network on both existing rain streak datasets and raindrop datasets. Moreover, we also compare our CCN with the state-of-the-art methods on the proposed RainDS.

5.1. Dataset and evaluation metrics

Rain streak dataset. Rain200H and Rain200L, collected by Yang *et al*. [43], are two widely used datasets with "heavy" and "light" rain streaks. In each dataset, there are 1.8k synthetic image pairs used for training and 200 image pairs for testing.

Raindrop dataset. Qian *et al*. [30] collected a raindrop dataset which consists of 1.1k corrupted and clean image pairs. 861 of them are used for training while the rest are for evaluation.

Evaluation metrics. Following previous deraining methods, the quantitative results are evaluated based on two commonly used metrics: PSNR and SSIM.

Table 2: Quantitative comparisons with the state-of-the-art on rain streak datasets.

Methods \ Dataset	Rain200L		Rain200H	
	PSNR	SSIM	PSNR	SSIM
GMM [25]	27.16	0.90	13.04	0.47
DSC [29]	25.68	0.88	13.17	0.43
DDN [10]	33.01	0.97	24.64	0.85
RESCAN [24]	37.07	0.99	26.60	0.90
DAF-Net [17]	32.07	0.96	24.65	0.86
SPA-Net [40]	31.59	0.97	23.04	0.85
PReNet [35]	36.76	0.98	28.08	0.89
DRD-Net [4]	37.15	0.99	28.16	0.92
Ours	**37.94**	**0.99**	**29.12**	**0.92**

5.2. Implementation details

Search Configuration. We split the training set $\mathcal{D}$ into a search training set $\mathbb{D}_{train}$ (70%) and a search validation set $\mathbb{D}_{val}$ (30%). The architecture parameters are optimized on $\mathbb{D}_{val}$ while the network parameters are optimized on $\mathbb{D}_{train}$. The macro structure of each deraining block is shown in Fig. 3. Each block contains N stacked neural cells, and we set N=2 in our experiments due to GPU memory limitations. The channel is set to 64. During searching, we use Adam to optimize α and ω with the initial learning rate of 0.002 and 0.001, respectively, and we adopt a cosine scheduler for 100 epochs. The batch size is 16 and the weight decay is 0.0005. RSR blocks are searched using Rain200H while RDR blocks are found using the raindrop dataset.

Train Configuration. After searching powerful architectures for RSR and RDR blocks respectively, we incorporate them into our unified network CCN. We train our CCN for 100 epochs using Adam as the optimizer with a momentum of 0.9 and a weight decay of 0.0005. The initial learning rate is set to 0.001 and we also adopt the cosine scheduler for training. For both searching and training, we apply a patch-based training strategy [35, 26] and randomly crop patches of 128×128 pixels from each image. Data augmentation [33] is used by flipping images horizontally. The supervision for the RSR/RDR block in our CCN is the version of removing rain streaks/raindrops from its input. Therefore, our CCN is able to tackle different rainy situations.

5.3. Comparison with the State-of-the-Arts

For fair comparisons with state-of-the-art methods, we use the same training and testing sets of existing datasets.

Comparison with rain streak removal algorithms. We conduct extensive experiments on several rain streak datasets to evaluate the deraining performance of our CCN. We compare with eight state-of-the-art rain streak removal

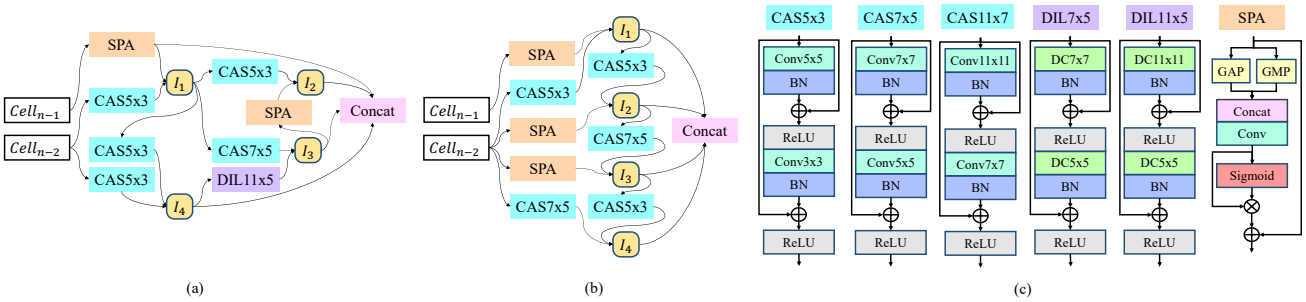

Figure 4: Illustration of one searched cell for RSR (a) and RDR (b) as well as our specified deraining search space (c).

Table 3: Quantitative comparisons with the state-of-the-art on our RainDS. All the networks are re-trained on RainDS.

Methods \ RainDS	Real-world						Synthetic					
	RS		RD		RDS		RS		RD		RDS	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
SPA-Net [40]	0.7016	22.17	0.6628	20.43	0.6318	19.93	0.9129	31.74	0.9251	28.97	0.8737	26.64
PReNet [35]	0.7691	24.56	0.7171	22.33	0.6657	21.20	0.9528	33.92	0.9647	31.99	0.9195	29.62
DRD-Net [4]	0.7203	23.83	0.6569	21.14	0.6043	20.10	0.8970	30.70	0.9160	29.34	0.6987	22.91
CCN	**0.8207**	**26.83**	**0.7576**	**24.81**	**0.7049**	**23.09**	**0.9697**	**35.12**	**0.9749**	**33.29**	**0.9501**	**32.16**

Table 4: Quantitative comparisons with the state-of-the-art on raindrop datasets.

Method	PSNR	SSIM
Eigen's [6]	23.74	0.79
Pix2pix [18]	28.15	0.85
AttentGAN [30]	30.55	0.90
Quan's [33]	30.86	0.93
Ours	**31.34**	**0.95**

Table 5: Ablation study of our network.

RainDS-Syn	RS	RD	RDS
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
CCN-RAND	30.72/0.92	29.43/0.90	28.17/0.86
CCN-1	34.67/0.96	33.56/0.97	32.47/0.95
CCN-2	35.12/0.97	33.89/0.97	32.27/0.95
CCN-3	33.85/0.95	33.14/0.96	31.79/0.94
CCN-4	35.04/0.97	34.05/0.98	32.53/0.96
CCN-RSR	33.42/0.95	31.78/0.94	30.07/0.92
CCN-RDR	32.34/0.94	31.42/0.94	29.39/0.92

methods. The quantitative results are shown in Table 2. We can see that our CCN outperforms all the other methods, especially on PSNR, *e.g.*, CCN surpasses the state-of-the-art method DRD-Net [4] by 0.79dB on Rain200L, and 0.96dB on Rain200H.

Comparison with raindrop removal algorithms. In Table 4, we compare CCN with several existing raindrop removal methods quantitatively. CCN outperforms the state-of-the-art method Quan's by 0.48dB on PSNR. More quantitative and qualitative results on more rain streak datasets are provided in the supplementary material.

5.4. Comparison results on RainDS

We conduct extensive experiments on RainDS dataset to evaluate the performance of the proposed CCN and the state-of-the-art, *i.e.*, PReNet [35], DRD-Net [4] and AttentGAN [30]. For fair comparisons, all the methods used the same training set, including rain streak, raindrop as well as rain streak and raindrop images. Although such real rain/rain-free image pairs are difficult to acquire, those data are important to reduce the domain gap. The evaluation results on real-world data and synthetic data are reported in Table 3. As indicated by Table 3, our CCN outperforms the state-of-the-art in all scenarios. Visual comparisons are shown in Fig. 5.

5.5. Ablation study

In this section, we investigate the impact of each design in our proposed CCN.

(1) Neural architecture search algorithm. To evaluate the influence of the neural architecture search algorithm, we compare the performance of our found rain block architectures with randomly generated ones. We randomly select two operations (*zero operation is excluded*) for each node in the neural cells to construct both the RSR and RDR block. Then, the CCN with these two blocks, denoted as CCN-RAND, is evaluated on our RainDS dataset. As shown in Table 5, CCN-RAND obtains worse results than the searched networks. This indicates that both our search algorithm and the specific search space are effective. Furthermore, we run our searching algorithm for four

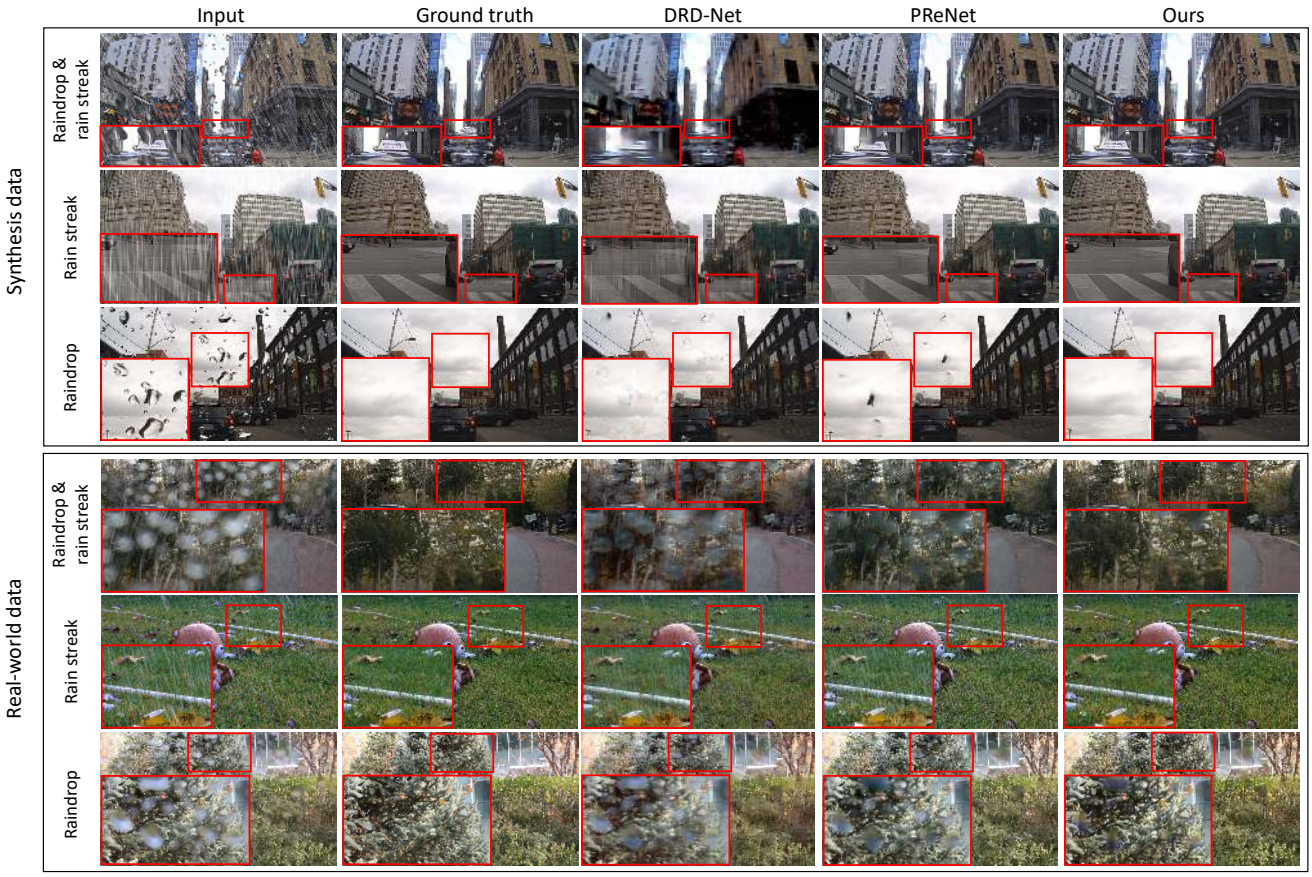

Figure 5: Comparisons with the state-of-the-art methods. DRD-Net [4] and PReNet [35] are retrained on RainDS.

times with different initialization seeds following existing NAS approaches. Our CCN equipped with the searched four groups of RSR and RDR blocks are denoted as CCN-1, CCN-2, CCN-3 and CCN-4 respectively. Their evaluation results can be found in the Table 5. Although there are subtle differences among the results of the four CCNs, they all outperform other competing methods. CCN-2 obtains the highest accuracy among all 4 runs, and we use this model as our CCN.

(2) RSR and RDR blocks. In order to explore whether RSR or RDR has more influences to the performance, we construct two new models, To dissect the impacts of RSR and RDR, we construct two models,*i.e.*, CCN-RSR and CCN-RDR, whose inner blocks are all RSR or RDR respectively. Table 5 shows CCN-RSR performs better than CCN-RDR, and this may be because RDR fails to perceive thin and line-shaped rain streaks while the rain streak block RSR can. Therefore, it is necessary to cascade them into a unified network and remove different types of rain in a complementary fashion.

(3)The effect of different input data. Since the parameters of our RSR or RDR blocks are optimized according to different input data, our proposed CCN is able to tackle various rainy situations. Benefiting from our complementary learning, our CCN still works well when inputs only contain rain streaks or raindrops, as indicated by Table 3. This is because RDR and RSR blocks learn to compensate for each other and are aware of different types of rain during training. Thus, our CCN consistently outperforms state-of-the-art.

6. Conclusion

In this paper, we proposed a novel complementary cascaded network architecture, namely CCN, to remove rain streaks and raindrops in a unified framework. Taking advantage of neural architecture search and our specifically designed deraining search space, we achieved an effective deraining network to remove various types of rain. We present a new real-world rain dataset RainDS to bridge the domain gap between real and synthetic rain. RainDS provide an ideal testbed for evaluating real-world deraining performance of our CCN method. Extensive experiments on our RainDS demonstrate the effectiveness and superiority of our unified deraining network. We believe our dataset would significantly advance the research of rain removal tasks in the future.

References

[1] Yi Chang, Luxin Yan, and Sheng Zhong. Transformed low-rank model for line pattern noise removal. In *ICCV*, pages 1726–1734, 2017. 2

[2] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Real-world image denoising with deep boosting. *TPAMI*, 42(12):3071–3087, 2019. 2

[3] Liang-Jian Deng, Ting-Zhu Huang, Xi-Le Zhao, and Tai-Xiang Jiang. A directional global sparse model for single image rain removal. *Applied Mathematical Modelling*, 59:662–679, 2018. 2

[4] Sen Deng, Mingqiang Wei, Jun Wang, Yidan Feng, Luming Liang, Haoran Xie, Fu Lee Wang, and Meng Wang. Detail-recovery image deraining via context aggregation networks. In *CVPR*, pages 14560–14569, 2020. 2, 6, 7, 8

[5] Yingjun Du, Jun Xu, Xiantong Zhen, Ming-Ming Cheng, and Ling Shao. Conditional variational image deraining. *TIP*, 29:6288–6301, 2020. 1

[6] David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. In *ICCV*, pages 633–640, 2013. 1, 3, 7

[7] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Huang, and Xinghao Ding. Residual-guide network for single image deraining. In *ACM MM*, pages 1751–1759, 2018. 2

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 2, 5

[9] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *TIP*, 26(6):2944–2956, 2017. 1

[10] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *CVPR*, pages 3855–3863, 2017. 2, 6

[11] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. Lightweight pyramid networks for image deraining. *TNNLS*, 31(6):1794–1807, 2019. 1

[12] Kshitiz Garg and Shree K Nayar. Detection and removal of rain from videos. In *CVPR*, volume 1, pages I–I, 2004. 1

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 5

[14] Yuanbiao Gou, Boyun Li, Zitao Liu, Songfan Yang, and Xi Peng. Clearer: Multi-scale neural architecture search for image restoration. *NeurIPS*, 33:17129–17140, 2020. 3

[15] Jianyuan Guo, Kai Han, Yunhe Wang, Chao Zhang, Zhaohui Yang, Han Wu, Xinghao Chen, and Chang Xu. Hit-detector: Hierarchical trinity architecture search for object detection. In *CVPR*, pages 11405–11414, 2020. 3

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 4

[17] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *CVPR*, pages 8022–8031, 2019. 6

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 7

[19] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *CVPR*, pages 8346–8355, 2020. 2

[20] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *TIP*, 21(4):1742–1755, 2011. 2

[21] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *CVPR*, pages 1633–1642, 2019. 2

[22] Ruoteng Li, Robby T Tan, and Loong-Fah Cheong. All in one bad weather removal using architectural search. In *CVPR*, pages 3175–3185, 2020. 3

[23] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *CVPR*, pages 3838–3847, 2019. 3

[24] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, pages 254–269, 2018. 6

[25] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *CVPR*, pages 2736–2744, 2016. 2, 5, 6

[26] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *ECCV*, 2020. 6

[27] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019. 4

[28] Xing Liu, Masanori Suganuma, Zhun Sun, and Takayuki Okatani. Dual residual networks leveraging the potential of paired operations for image restoration. In *CVPR*, pages 7007–7016, 2019. 5

[29] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015. 1, 2, 6

[30] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *CVPR*, pages 2482–2491, 2018. 1, 3, 5, 6, 7

[31] Xu Qin and Zhilin Wang. NASNet: A neuron attention stage-by-stage net for single image deraining. *arXiv preprint arXiv:1912.03151*, 2019. 3, 5

[32] Ruijie Quan, Xuanyi Dong, Yu Wu, Linchao Zhu, and Yi Yang. Auto-reid: Searching for a part-aware convnet for person re-identification. In *ICCV*, pages 3750–3759, 2019. 3

[33] Yuhui Quan, Shijie Deng, Yixin Chen, and Hui Ji. Deep learning for seeing through window with raindrops. In *ICCV*, pages 2463–2471, 2019. 1, 3, 6, 7

[34] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K. Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *ICCV*, 2019. 5

[35] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, pages 3937–3946, 2019. 1, 2, 6, 7, 8

[36] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, pages 1874–1883, 2016. 2, 5

[37] Tianqi Tang, Xin Yu, Xuanyi Dong, and Yi Yang. Autonavigator: Decoupled neural architecture search for visual navigation. In *WACV*, pages 3743–3752, 2021. 3

[38] Guoqing Wang, Changming Sun, and Arcot Sowmya. Erl-net: Entangled representation learning for single image deraining. In *ICCV*, pages 5644–5652, 2019. 2

[39] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, pages 3103–3112, 2020. 2

[40] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, pages 12270–12279, 2019. 2, 3, 6, 7

[41] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *CVPR*, pages 3877–3886, 2019. 1, 2

[42] Wenhan Yang, Robby T Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *TPAMI*, 42(6):1377–1393, 2019. 1, 2

[43] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *CVPR*, pages 1357–1366, 2017. 2, 5, 6

[44] Wenhan Yang, Robby T Tan, Shiqi Wang, Yuming Fang, and Jiaying Liu. Single image deraining: From model-based to data-driven and beyond. *TPAMI*, 2020. 1

[45] Rajeev Yasarla and Vishal M Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *CVPR*, pages 8405–8414, 2019. 2

[46] Rajeev Yasarla, Vishwanath A Sindagi, and Vishal M Patel. Syn2real transfer learning for image deraining using gaussian processes. In *CVPR*, pages 2726–2736, 2020. 2

[47] Shaodi You, Robby T Tan, Rei Kawakami, and Katsushi Ikeuchi. Adherent raindrop detection and removal in video. In *CVPR*, pages 1035–1042, 2013. 3

[48] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Adherent raindrop modeling, detectionand removal in video. *TPAMI*, 38(9):1721–1733, 2015. 1, 3

[49] Zitong Yu, Chenxu Zhao, Zezheng Wang, Yunxiao Qin, Zhuo Su, Xiaobai Li, Feng Zhou, and Guoying Zhao. Searching central difference convolutional networks for face anti-spoofing. In *CVPR*, pages 5295–5305, 2020. 3

[50] Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox, and Frank Hutter. Understanding and robustifying differentiable architecture search. In *ICLR*, 2020. 5

[51] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, pages 695–704, 2018. 5

[52] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *TCSVT*, 2019. 5

[53] Yupei Zheng, Xin Yu, Miaomiao Liu, and Shunli Zhang. Residual multiscale based single image deraining. In *BMVC*, page 147, 2019. 1

[54] Yupei Zheng, Xin Yu, Miaomiao Liu, and Shunli Zhang. Single-image deraining via recurrent residual multiscale networks. *TNNLS*, 2020. 1

[55] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *ICCV*, pages 2526–2534, 2017. 2