

Multi-Decoding Deraining Network and Quasi-Sparsity Based Training

Yinglong Wang[†] Chao Ma^{§*} Bing Zeng^{†*}

[†]University of Electronic Science and Technology of China

[§]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University

ylwanguestc@gmail.com, chaoma@sjtu.edu.cn, eezeng@uestc.edu.cn

Abstract

Existing deep deraining models are mainly learned via directly minimizing the statistical differences between rainy images and rain-free ground truths. They emphasize learning a mapping from rainy images to rain-free images with supervision. Despite the demonstrated success, these methods do not perform well on restoring the fine-grained local details or removing blurry rainy traces. In this work, we aim to exploit the intrinsic priors of rainy images and develop intrinsic loss functions to facilitate training deraining networks, which decompose a rainy image into a rain-free background layer and a rainy layer containing intact rain streaks. To this end, we introduce the quasi-sparsity prior to train network so as to generate two sparse layers with intact textures of different objects. Then we explore the low-value prior to compensate sparsity, forcing all rain streaks to enter into one layer while non-rain contents into another layer to restore image details. We introduce a multi-decoding structure to specially supervise the generation of multi-type deraining features. This helps to learn the most contributory features to deraining in respective spaces. Moreover, our model stabilizes the feature values from multi-spaces via information sharing to alleviate potential artifacts, which also accelerates the running speed. Extensive experiments show that the proposed deraining method outperforms the state-of-the-art approaches in terms of effectiveness and efficiency.

1. Introduction

The rainy artifacts on images usually include visible rain streaks and the haze-like veiling effect. This work focuses on rain streaks removal as it is still an unsolved challenging task. Conventional methods usually remove rain via learning an over-complete dictionary [26, 1, 10]. The rain-free image is reconstructed by the non-rain dictionary atoms, which are identified from all the learned dictionary atoms

by means of the heuristic appearance characteristics of rain. These methods do not perform well when the patterns of rain streaks are complex, e.g., rain streaks overlap and interweave. Without using high-level features, complex rain streaks cannot be identified, causing heavy performance bottleneck.

Recently, deep learning based methods explore high-level features of rainy images through pretrained CNN models. They achieve the state-of-the-art deraining performance in not only restoration quality but also the running speed [27, 19, 8, 31, 15, 5, 4, 30, 35, 17, 25]. However, the limitation appears when rain streaks are wide or blurry. Fig. 1 shows an example that, when the middle parts of rain streaks become brighter, while the edges are blurry, existing deep learning based methods only remove the bright parts, but the blurry edges still remain. The reason lies in that these CNN models are trained by means of minimizing the statistical differences between deraining results and ground truth images, as formulated by the MSE [35, 5, 30] or MAE [14] loss functions. MSE and MAE converge at the arithmetic mean and median of the observations respectively, which correlate poorly with image details restoration [12, 37]. Therefore, exploring losses beyond the traditional formulation is urged for further performance improvement.

In this paper, we introduce the intrinsic priors to construct loss functions. We first study the sparsity of rainy images to achieve sparse image decomposition, i.e., making each sparse layer contain as intact (not split) textures of objects as possible [13, 3, 20]. Based on such property, rain streaks will not be split during deraining, as well as other image contents. Note that the concept of sparsity here is an intrinsic property of images, which is different from the one indicating the majority of elements in a matrix are zero or close to zero [29]. However, existing sparse stochastic distributions lead to too complex derivation of maximum likelihood (ML) to train CNNs [13].

To obtain tractable loss functions via ML, we relax the sparsity degree and develop *quasi-sparsity prior* to approximate the sincere sparsity. Quasi-sparsity with simpler formula keeps the property of sparsity and can be used to train

*C. Ma and B. Zeng are corresponding authors.

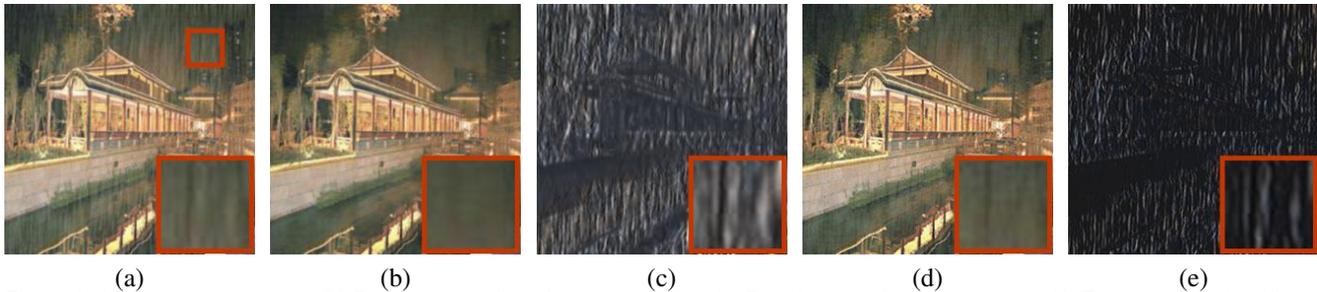


Figure 1. (a) Input rainy image. (b) Deraining result with quasi-sparsity. (c) Rain layer with quasi-sparsity. (d) Deraining result without quasi-sparsity. (e) Rain layer without quasi-sparsity. Without quasi-sparsity, a few traces of rain streaks remain in the deraining result.

CNN to decompose images sparsely, meaning that the textures belonging to one object will not appear in different layers. Moreover, an auxiliary decoder loss and a low-value prior loss are proposed to force rain streaks to enter into the rainy layer during deraining. Fig. 1 compares our networks with and without quasi-sparsity. We observe that quasi-sparsity produces clearer deraining result and removes rain more thoroughly.

Deep features from different spaces have not been fully studied before [35, 30]. We introduce novel auxiliary decoders to respectively decode deep features in multi-spaces. Auxiliary decoders play roles in two aspects: 1) they generate rain-free images from multi-spaces respectively, which helps to compare the effectiveness of different types of deep features; 2) they boost encoder to generate optimal features via respectively imposing supervision in each space. Features from different spaces usually possess large numerical gaps that may cause undesired artifacts [31]. We propose to implement information sharing among multi-type features before feeding them into the main decoder. Ablation studies illustrate that information sharing alleviates the undesired artifact by stabilizing the feature values from different spaces.

In summary, the main contributions are listed below:

- We explore the intrinsic properties of rainy images, i.e., quasi-sparsity and low-value priors, for deraining.
- We introduce a multi-decoding structure that enables each feature space to generate optimal deraining features by respectively imposing supervision. We introduce information sharing to stabilize feature values of multi-spaces to alleviate artifacts.
- We conduct extensive experiments in comparison with the state of the arts. We push forward the state-of-the-art deraining performance by large margins in terms of both effectiveness and efficiency.

2. Related Work

Dictionary learning [18] is first used to remove rain streaks from a single image by decomposing the image content into multiple different layers [10, 1, 26, 24]. Recently,

deep learning improves the deraining performance substantially. Based on the rain model, existing deep learning based deraining methods typically fall into three categories: direct learning methods, residual model methods and scattering model methods. Direct learning methods learn to output a rain-free image directly from the input rainy image. In [23], Wang et al. first build a dataset with temporal priors and human supervision, and train a SPANet to remove random rain streaks in a local-to-global way.

Residual methods decompose the rainy image into a rainy layer and a rain-free layer. In [4], a DerainNet is trained in the high-frequency domain to restore image details, so that the interference from the background can be reduced. A deep network based on ResNet [6] is also trained with high-pass to reduce the mapping range from input to output for training efficiency [5]. In [30, 31], a new rain model is introduced to model the apparent rain streaks and the veiling effect. But the atmospheric light and transmission of the veiling effect are not explicitly predicted. The rainy image is finally decomposed into a rain layer and the background layer by their JORDER network. Moreover, a binary map is learnt to locate rain streaks to guide the JORDER training. In [35], the density of rain streaks is evaluated by a multi-stream densely connected DID-MDN structure, which characterizes the rain streaks with various shape and size. Li et al. decompose rain streaks in single images into a number of rain layers. Then a recurrent neural network RESCAN is trained to remove rain streaks state-wisely [17]. Hu et al. study the relationship between the visual effect of rain and scene depth, based on which fog that contains depth information is introduced to model the formation of rainy images and to guide the end-to-end training of their network [8]. In [19], Ren et al. rethink the network structure, input and output, and the loss functions, and propose a simpler deraining baseline.

For the scattering model methods [15], Li et al. render the ground truth for atmospheric light, rain streaks and the transmission of vapor to remove rain streaks as well as the vapor effect. In [32], self-supervision is introduced to remove rain streaks across scales. Due to the correlation of deraining to textures, this work creates a fractal band learn-



Figure 2. (a) Rainy image. (b) Log-histogram after (a) is filtered. (c) Non-sparse rainy image. (d) Log-histogram after (c) is filtered. (e) Log-probability of some stochastic distributions. According to [13], Gaussian distribution is non-sparse as it is above the straight line. Laplace distribution results in the straight line, which is the demarcation line of sparsity and non-sparsity.

ing (FBL) network to capture discriminative features for deraining. Besides, cross-scale self-supervision is introduced to improve the generalization of FBL. Based on the previous studies of deraining and dehazing, Yang et al. implement comprehensive experiments to verify how low-level image enhancement tasks contribute to the high-level visual recognition [33]. In [34], an UMRL network is proposed to predict the rain content at different scales with uncertainty. Such rain content is used to estimate the final deraining results. Different from existing approaches, we pay attention to the intrinsic priors of rainy images and design novel cost functions to facilitate deraining.

3. Quasi-Sparsity Prior

Sparsity keeps intact object textures during image decomposition, i.e., textures belong to one object will appear in one image layer [13]. Therefore, sparsity has great potential to separate rain and non-rain textures during deraining. However, not all images possess sparsity, e.g., blurry images. Hence we first determine the sparsity of rainy images to ensure the generalization of our algorithm. Existing stochastic distributions fitting sparsity are too complex to derive an effective maximum likelihood to train CNNs [13, 21], which hinders the applications of sparsity in computer vision tasks. We develop a quasi-sparse distribution to approximate sparsity to obtain a feasible loss function.

3.1. Sparsity of Rainy Images

In [13], Levin and Weiss show a property that the log-histograms of filtered natural images are below the straight line connecting the minimal and maximal values, and name this property as *sparsity prior*. We collect 1000 real rainy images and 9000 synthetic rainy images and compute their log-histogram after filtering. We find that 93.7% of these images maintain sparsity prior, as an example shown in Fig. 2(a) and (b). An example of the remaining non-sparse part is displayed in Fig. 2(c) and (d). We experimentally find that these minority of non-sparse images are of low contrast. Ruling out such small part, sparsity prior is applicable to rainy images from the statistical perspective.

3.2. Quasi-Sparsity Formulation

According to the definition of sparsity [13], single Gaussian and Laplace distributions are non-sparse as shown in Fig. 2(e). Single Gaussian distribution is above the straight line, hence infinite Gaussian distributions with different parameters are added together to fit the sparsity [21]. This yields complex derivation process during the ML estimation. Single Laplace distribution results in the straight line. Levin and Weiss [13] fit the sparsity prior by combining two different Laplace distributions as follows:

$$P(x) = \frac{\pi_1}{2s_1} e^{-|x|/s_1} + \frac{\pi_2}{2s_2} e^{-|x|/s_2}, \quad (1)$$

where π_1 , π_2 , s_1 and s_2 are parameters. The complexity decreases substantially in comparison to the combination of infinite Gaussian distributions, but it is still intractable when used to train a network through maximum likelihood estimation based on Eq. (1). Because Eq. (1) is an addition of two exponential functions, it introduces the hybrid operation of exponential and logarithmic functions during ML estimation, which is time-consuming especially applied to the whole images.

To train a CNN via sparsity, we relax the sparsity by setting $s_1 = s_2 = s$ and $\pi_1 = \pi_2 = \pi$ in Eq. (1):

$$P_q(x) = \frac{\pi}{s} e^{-|x|/s}, \quad (2)$$

i.e., we apply single Laplace distribution to approximate the sincere sparsity defined in [13]. As shown in Fig. 2(e), single Laplace distribution is located at the demarcation line of sparsity and non-sparsity. We define single Laplace distribution as *quasi-sparsity distribution*, as it is not true sparsity according to [13]. When quasi-sparsity distribution in Eq. (2) is used to approximate the true sparsity, the quasi-sparsity distribution of a whole rainy image \mathbf{I} could be formulated as:

$$P_q(\mathbf{I}) = \prod_{i,k} P_q(\omega_{i,k} * \mathbf{I}) \quad (3)$$

where $\omega_{i,k}$ is the k^{th} filter centered at the i^{th} pixel. $*$ is the convolution operation. The filters with horizontal and vertical gradients and the first and the second derivatives

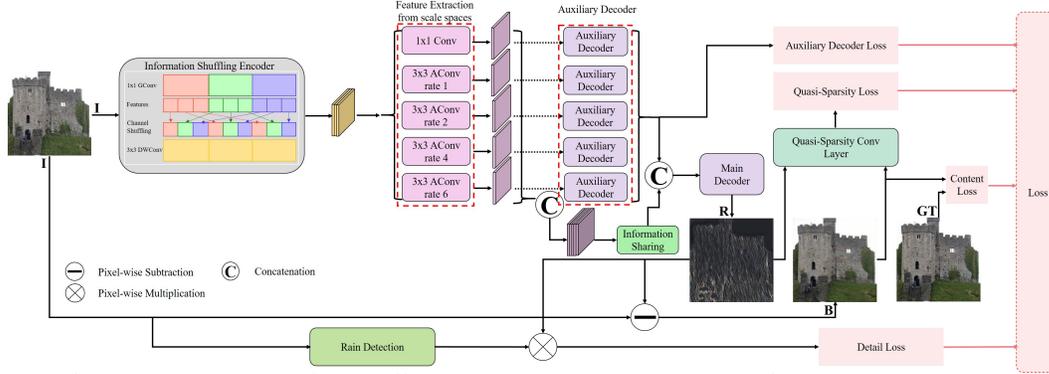


Figure 3. Pipeline of our method. A stack of 12 ShuffleNet units constitute the backbone of our network. An multi-decoding structure (auxiliary decoder) is introduced to decode deraining results from different scale spaces. The intrinsic quasi-sparsity loss, detail loss, auxiliary decoder loss and content loss work complementarily to train our network.

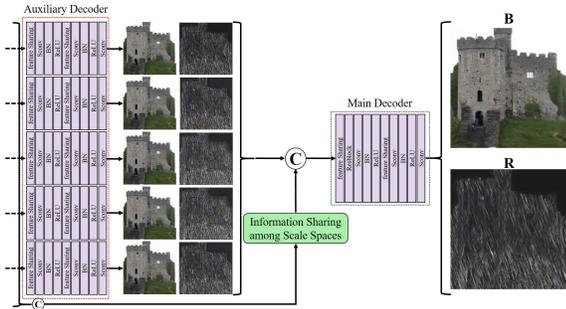


Figure 4. Architecture of the auxiliary and main decoders. Different convolution groups exchange information in the single auxiliary decoder, as well as the features from different scale spaces.

are used here to constitute the quasi-sparsity prior of rainy images [13].

Quasi-sparsity contains one exponential function, which substantially simplifies the derivative process of the logarithm maximum likelihood estimation to obtain a concise sparse loss function. This loss function helps to conveniently apply sparsity for training a CNN. More importantly, quasi-sparsity preserves the property of the sincere sparsity, which helps our network better separate different objects during image decomposition.

4. Proposed Method

Under the constraints of several intrinsic properties of rainy images, we train the CNN network to decompose the rainy image I into a sparse rain layer R and rain-free background layer B as follows:

$$R = \mathcal{S}(I) \quad (4)$$

$$B = I - \mathcal{S}(I), \quad (5)$$

where $\mathcal{S}(\cdot)$ denotes the network inference. Because our multi-decoding network can generate sparse rain and non-rain decomposition via quasi-sparsity prior, we call our

network **Quasi-Sparse Multi-Decoding** deraining network (QSDM).

4.1. Network Structure

We show the pipeline of our network structure in Fig. 3. A stack of 12 ShuffleNet units constitute the backbone of our QSDM, which is called Information Shuffling Encoder. ShuffleNet units have been verified to have a fast feature extraction speed [36]. Besides, the lightweight group convolution and deep separable convolution in ShuffleNet units reduce the parameters of our QSDM. Following [30], we also use a parallel connection of \hat{a} -trous convolutions [2] to generate multi-type features from different scale spaces, by which the obtained features keep the same size as original rainy image, avoiding the down-sampling and up-sampling operations when using spatial pyramid pooling [7]. Moreover, point-wise convolution acts as a shortcut path to preserve the original feature map.

The decoding module consists of the auxiliary decoders and the main decoder. Each auxiliary decoder decodes out a rain-free image from the corresponding scale space. The generated rain-free images are concatenated together as the input of the main decoder as shown in Figs. 3 and 4. Direct concatenation of deep features from multi-spaces might cause artifacts, for example the black artifacts in the result of [31]. Inspired by [36], we let different feature spaces share information by mutually exchanging feature maps before being fed into the main decoder. Both quantitative and qualitative results demonstrate the effectiveness of our method in alleviating the artifacts.

4.2. Training Loss

We present how our QSDM is trained via the proposed quasi-sparsity prior based on a logarithmic ML estimation which favors the separation of objects during the decomposition of rainy images into rain and non-rain layers. A low-value prior of rainy layer R and the auxiliary decoder

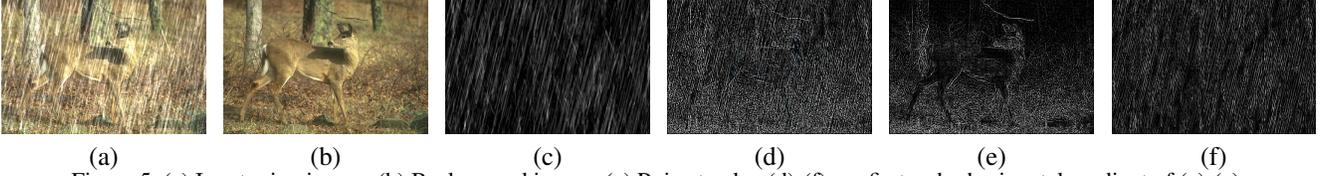


Figure 5. (a) Input rainy image. (b) Background image. (c) Rain streaks. (d)-(f) are first order horizontal gradient of (a)-(c).

Table 1. PSNR/SSIM comparisons on our testing datasets

Methods	MSPFN[9]	RESCAN[17]	RCDNet[22]	JORDER[31]	PYM+GAN[15]	SPANet[23]	Ours
Test-I	32.50	27.21	31.69	27.44	17.93	28.47	33.15
	0.913	0.835	0.866	0.885	0.677	0.858	0.923
Test-II	31.70	34.81	35.51	28.39	18.48	35.17	35.66
	0.941	0.944	0.945	0.902	0.747	0.944	0.950

work together to force rain streaks enter into the rain layer \mathbf{R} , and non-rain contents into the background layer \mathbf{B} .

Quasi-sparsity loss. We assume that the rain layer \mathbf{R} and the background layer \mathbf{B} are independent to simplify our algorithm. By substituting Eq. (2) into Eq. (3), the quasi-sparsity prior of \mathbf{I} in Eq. (3) could be rewritten as:

$$P_q(\mathbf{I}) = P_q(\mathbf{R})P_q(\mathbf{B}) = \prod_{i,k} \frac{\pi}{s} e^{-(|\omega_{i,k} * \mathbf{R}| + |\omega_{i,k} * \mathbf{B}|)/s}. \quad (6)$$

By applying logarithm, Eq. (6) becomes:

$$\log(P_q(\mathbf{I})) = -\frac{1}{s} \sum_{i,k} (|\omega_{i,k} * \mathbf{R}| + |\omega_{i,k} * \mathbf{B}|) + \beta. \quad (7)$$

β is a constant produced during calculating logarithm. s is also constant. Hence, maximizing Eq. (7) is equal to minimizing the following loss function:

$$\mathcal{L}_q = \sum_{i,k} |\omega_{i,k} * \mathbf{R}| + |\omega_{i,k} * \mathbf{B}|. \quad (8)$$

For all images $\{\mathbf{I}_t\}_{t=1}^N$ in our training dataset $\{(\mathbf{I}_t, \mathbf{B}_t)\}_{t=1}^N$, \mathcal{L}_q is rewritten as:

$$\mathcal{L}_q = \sum_{t=1}^N \sum_{i,k} |\omega_{i,k} * \mathcal{S}(\mathbf{I}_t)| + |\omega_{i,k} * [\mathbf{I}_t - \mathcal{S}(\mathbf{I}_t)]|. \quad (9)$$

Similar formulation of Eq. (8) used in traditional optimizations [20] belongs to our quasi-sparsity according to the strict definition of sparsity [13]. The maximum likelihood estimation based on our quasi-sparsity generates a concise loss function which could be conveniently utilized to train a CNN. If we utilize the true sparsity in Eq. (1), the derivative process, i.e., Eqs. (6), (7), (8) and (9), will become very complex and the obtained loss function is also time-consuming. Moreover, though our quasi-sparsity loss function is formulated with L_1 norm, it is no longer the

commonly used losses which measure the differences of deraining results and ground truth. Behind our loss function is the intrinsic sparsity prior of rainy images.

Content loss. We utilize MAE to measure the difference between the deraining results from the main decoder and ground truth to restore image contents:

$$\mathcal{L}_C = \sum_{t=1}^N |\mathbf{I}_t - \mathcal{S}(\mathbf{I}_t) - \mathbf{B}_t|. \quad (10)$$

Detail loss. As shown in Fig. 5(c), the non-rain areas have very low values in \mathbf{R} , which is formulated as our low-value prior of rainy images to restore image details as follows:

$$\mathcal{L}_D = |(\mathbb{1} - \mathbf{L}) \circ \mathbf{R}|, \quad (11)$$

where \mathbf{L} is the location map of rain streaks obtained by [25] to filter out the high values at the rain location in \mathbf{R} . $\mathbb{1}$ is all-one matrix. \circ is element-wise multiplication.

Auxiliary decoder loss. Each feature space could generate optimal deraining features by imposing supervision on the auxiliary decoders. Let $\{\mathcal{A}_i(\cdot)\}_{i=1}^5$ denote the inference of the five auxiliary decoders. Our auxiliary decoder loss is defined as follows:

$$\mathcal{L}_A = \frac{1}{5} \sum_{i=1}^5 \sum_{t=1}^N \|\mathbf{I}_t - \mathcal{A}_i(\mathbf{F}_{t,i}) - \mathbf{B}_t\|_F^2, \quad (12)$$

where $\mathbf{F}_{t,i}$ is the feature map of \mathbf{I}_t and an input of $\mathcal{A}_i(\cdot)$. Our whole loss function is:

$$\mathcal{L} = \lambda_Q \mathcal{L}_q + \lambda_C \mathcal{L}_C + \lambda_A \mathcal{L}_A + \lambda_D \mathcal{L}_D \quad (13)$$

5. Experiments

To evaluate the performance, PSNR and SSIM [28] are selected as quantitative metrics. Six state-of-the-art methods [9, 17, 22, 31, 15, 23] are used for comparisons.

Table 2. Average running time comparisons. The image size is 512×512

Methods	MSPFN[9]	RESCAN[17]	RCDNet[22]	JORDER[31]	PYM+GAN[15]	SPANet[23]	w/o Sharing	Ours
Time	11.4s	0.47s	0.99s	1.39s	0.45s	0.66s	0.007s	0.005s

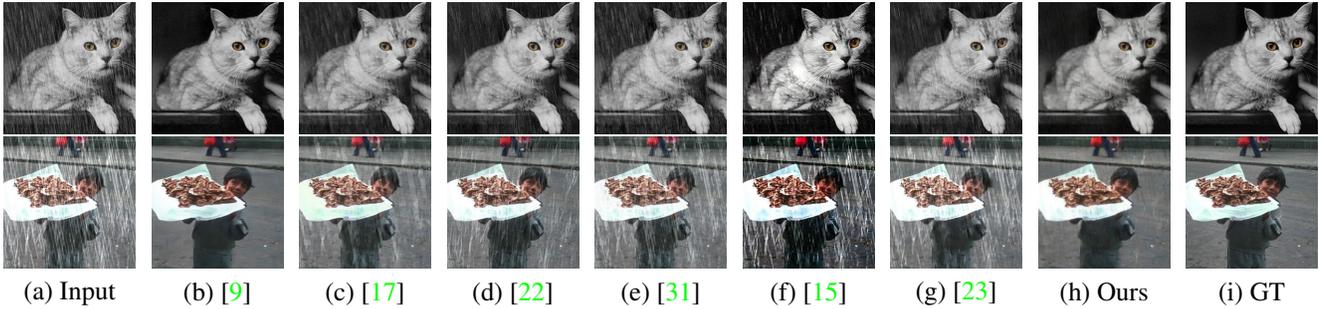


Figure 6. Qualitative comparisons on synthetic rainy images. (a) Input rainy images. (b)-(h) Deraining results of MSPFN [9], RESCAN [17], RCDNet [22], JORDER [31], PYM+GAN [15], SPANet [23] and our method. (i) Ground Truth.

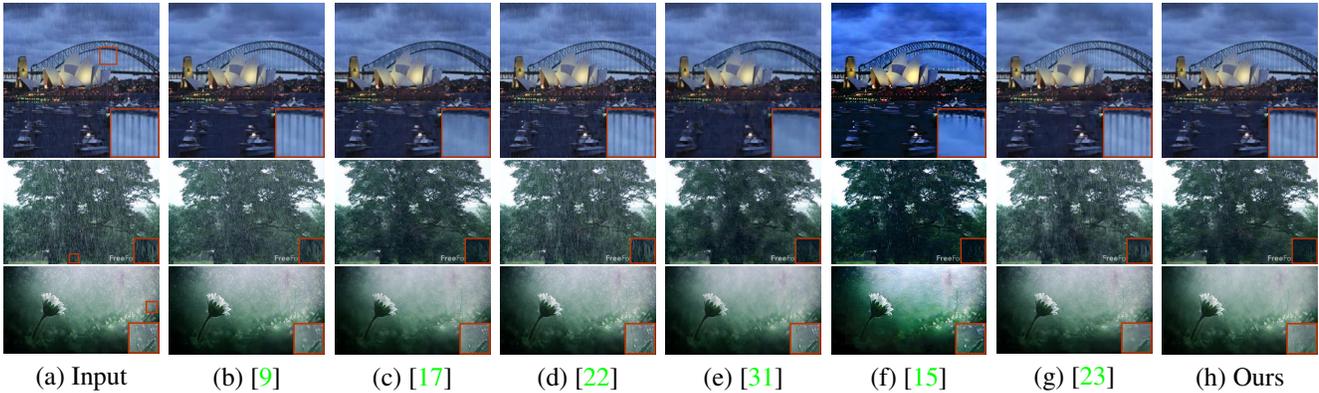


Figure 7. Qualitative comparisons on real-world rainy images. (a) Input rainy images. (b)-(h) Deraining results of MSPFN [9], RESCAN [17], RCDNet [22], JORDER [31], PYM+GAN [15], SPANet [23] and our method.

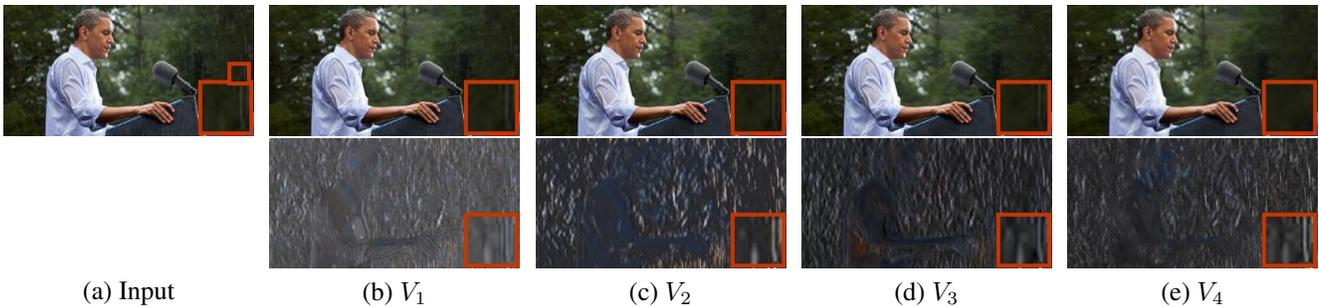


Figure 8. Visual results of ablation studies. (a) Input rainy image. (b)-(e) Results of V_1 , V_2 , V_3 and V_4 . The second line is the rain layers.

5.1. Implementation Details

The training pairs are randomly cropped from our training dataset with fixed size of 256×256 . Adam [11] is utilized as our optimizer. The learning rate is set to 0.001 initially and decays by multiplying 0.1 when the loss does not update. Our code is implemented on a NVIDIA 1080Ti GPU based on Pytorch. The parameters λ_Q , λ_C , λ_A and

λ_D in Eq. (13) are 10^{-3} , 1, 0.01 and 10^{-4} respectively.

5.2. Dataset Constructions

We follow [16] to prepare our training dataset which contains 20800 pairs. The synthetic rainy images are synthesized with rendered rainy layer and the ground truth by using the screen blend mode. In order to cover more base-

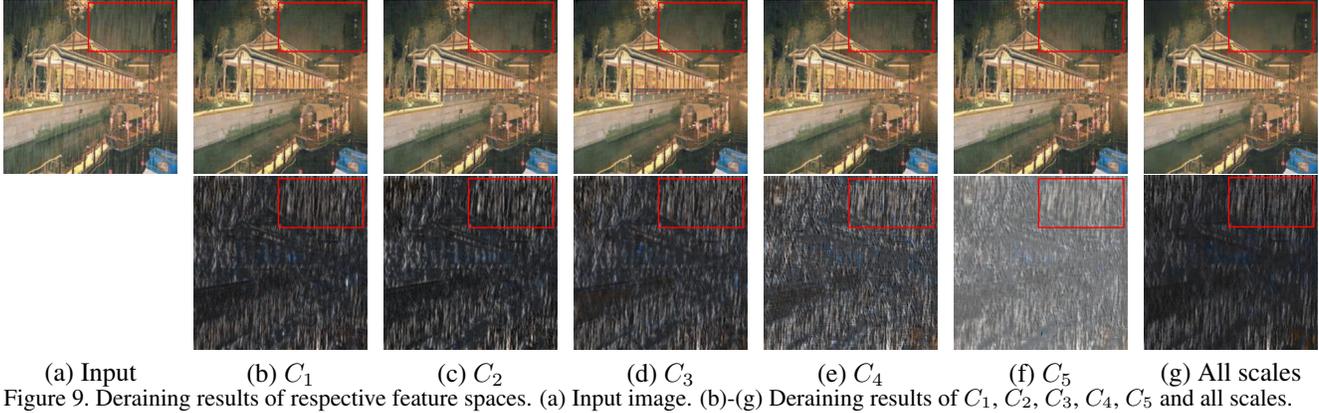


Figure 9. Deraining results of respective feature spaces. (a) Input image. (b)-(g) Deraining results of C_1 , C_2 , C_3 , C_4 , C_5 and all scales.

Table 3. PSNR/SSIM of the different variants of our method

Variants	V_1	V_2	V_3	V_4
\mathcal{L}_C	✓	✓	✓	✓
\mathcal{L}_Q	w/o	✓	✓	✓
\mathcal{L}_D	w/o	w/o	✓	✓
\mathcal{L}_A	w/o	w/o	w/o	✓
Test-I	29.94 0.858	31.70 0.898	32.03 0.908	33.15 0.923
Test-II	34.15 0.826	34.50 0.873	34.85 0.891	35.66 0.950

Table 4. PSNR/SSIM of respective feature spaces

Scales	C_1	C_2	C_3	C_4	C_5	All scales
Test-I	31.68 0.902	32.14 0.907	32.18 0.902	30.75 0.862	30.69 0.870	33.15 0.923
Test-II	33.86 0.868	34.85 0.901	34.91 0.914	33.96 0.887	32.66 0.869	35.66 0.950

line testing datasets to compare the generalization of different methods, we follow [27] and randomly select 100 pairs from the testing datasets of [35, 17, 5] respectively to constitute our Test-I. We follow the real-world dataset [23] as our second testing dataset named as Test-II.

5.3. Evaluations with State-of-the-arts

We compare our method with existing deraining methods on our testing datasets. The comparisons are classified as numerical and visual evaluations. The details are reported in the following:

Quantitative Evaluation. Table 1 shows the comparisons to the state-of-the-art deraining methods under the PSNR and SSIM metrics. Our method achieves more favorable results. Running time comparison is shown in Table 2. Our method achieves the fastest running speed and is two order of magnitude faster than the second fastest method [15].

Qualitative Evaluation. We show the visual comparison from the aspects of synthetic and real-world data. Fig. 6

Table 5. PSNR/SSIM of our QSMD w/o or w/ information sharing

Datasets	Test-I	Test-II
w/o Sharing	31.91/0.897	34.51/0.920
QSMD	33.15/0.923	35.66/0.950

shows two synthetic cases where our method is able to restore effectively, especially for the second heavy condition.

Besides the synthetic case, comparisons on the real-world rainy images are shown in Fig. 7. Light rain streaks can be handled well by existing methods, as shown in the first line of Fig. 7. But, some image details may be mistreated as rain and removed, such as the results of RESCAN [17], JORDER [31] and PYM+GAN [15]. When rain is heavy, some apparent rainy streaks still remain in the results of PYM+GAN [15], MSPFN [9], RCDNet [22] and SPANet [23], such as the second rainy image. The works [17, 31, 23] performs well on the bright grain-like rain streaks, but our method achieves clearer result.

5.4. Ablation Studies

The training of our network consists of intrinsic quasi-sparsity prior, low-value prior, a decoding loss indicating the auxiliary decoders and a content loss for the main decoder. We show how these loss functions work together to gradually improve image restoration results. Our four configurations to train our QSMD are: 1) only the content loss \mathcal{L}_C is used and named as V_1 ; 2) $\mathcal{L}_C + \mathcal{L}_Q$ is used and named as V_2 ; 3) $\mathcal{L}_C + \mathcal{L}_Q + \mathcal{L}_D$ is used and named as V_3 ; 4) the whole loss $\mathcal{L}_C + \mathcal{L}_Q + \mathcal{L}_D + \mathcal{L}_A$ is used and named as V_4 .

Fig. 8 and Table 3 show the qualitative and quantitative results. We observe that rain streaks become less and less after the loss functions are introduced one-by-one. In comparison, image details decrease step-wisely and rain streaks increase accordingly in the rainy layer. The quantitative indexes indicate the same laws, where the decoding loss improve the performance more apparently.

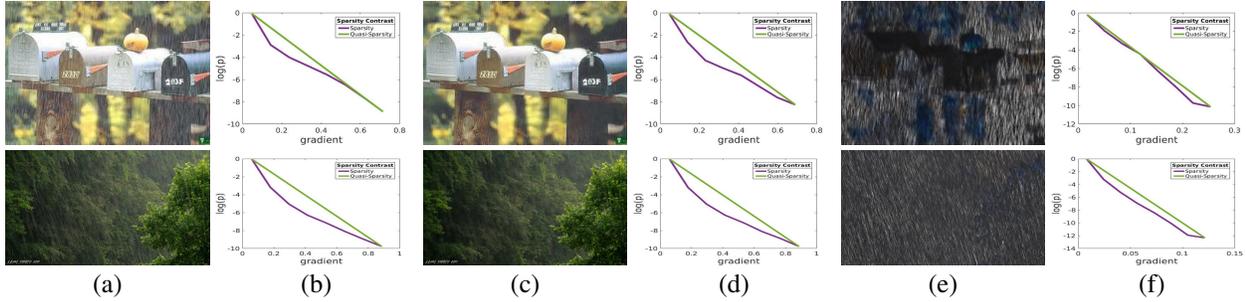


Figure 10. Sparsity of rainy image, the deraining result and the decomposed rainy layer.

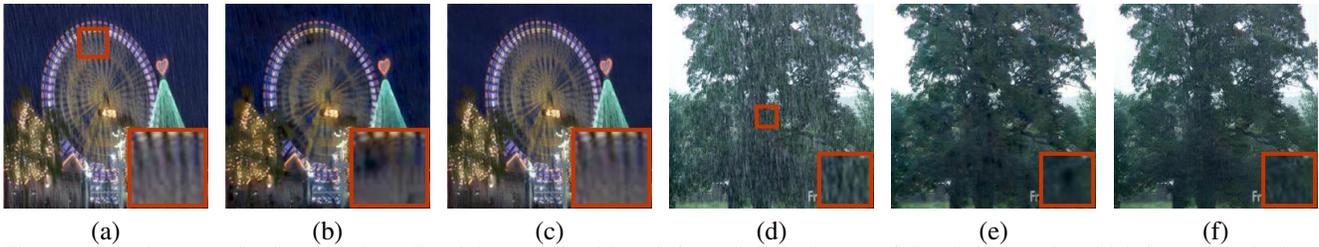


Figure 11. (a)(d) Input rainy images. (b)(e) Deraining results without information sharing. (c)(f) Deraining results with information sharing.

5.5. Discussions

Our method involves decoding in multi-spaces, information sharing and quasi-sparsity prior to improve network performance. We discuss their merits in the following:

Decoding in multi-spaces. We first symbolize multi-spaces as: C_1 denotes the space spanned by the convolution with 1×1 kernel. Similarly, C_2, C_3, C_4, C_5 are the spaces produced by à-trous convolutions with à-trous rate equalling to 1, 2, 4, 6 respectively. Auxiliary decoders decode deraining images from each space, as visually shown in Fig. 9. We observe that traces of wide streaks may remain in the results decoded from small-scale feature spaces, e.g., Fig. 9(b), and vice versa, e.g., Fig. 9(e). This shows specific convolution captures rainy features with corresponding size. To remove multi-type rain streaks, features from multi-spaces should be combined and work together.

Table 4 shows the quantitative evaluation to the different scales of features. We observe that C_2 and C_3 produce higher PSNR/SSIM values, indicating that deep features from them contribute the most to the deraining performance, which also indicates that the majority of rain streaks possess the width of 3 or 5 pixels in common images.

Sparse decomposition of QSMD. Quasi-sparsity lets our QSMD better separate objects during image decomposition, so that each layer is sparse and contains intact textures of objects. Fig. 10 shows the sparsity of rainy images, decomposed rainy layers and backgrounds. We observe that sparse rainy images are decomposed by our QSMD into sparse background layer and sparse rainy layer.

Information Sharing. Direct concatenation of different-type features may lead to undesired artifacts, for example

[31], which is also verified by our auxiliary decoders. This is due to large numerical gap among features from multi-spaces. We solve this problem via exchanging features from different scale spaces. Table 5 and Fig. 11 show the performance improvement. All the black artifacts disappear with the use of information sharing. Table 2 also indicates that the running speed improves by 0.002s.

6. Conclusions

Sparsity prior is an intrinsic property of images with sharp textures and clear contrast. When acting as a constraint for image decomposition, it helps each image layer contain intact textures of objects. In this paper, we determine the sparsity prior of rainy images from the perspective of statistics. Allowing for complexity, we exploit quasi-sparsity prior to approximate the true sparsity for training the proposed QSMD network, so that our network can produce sparse image decomposition. Additionally, the low-value prior and auxiliary decoders are introduced to enhance the separation of rain streaks and background contents. Through our multi-decoding structure, we study the deraining performance of different features and find the most favorable deraining features. Quantitative and qualitative evaluations illustrate the favorable performance of the proposed method over the state of the arts.

Acknowledgements

This work was supported by NSFC (61906119, U19B2035) and Shanghai Pujiang Program.

References

- [1] Duanyu Chen, Chiencheng Chen, and Liwei Kang. Visual depth guided color image rain streaks removal using sparse coding. *IEEE Trans. Circuit Syst. Video Technol.*, 24(8):1430–1455, Aug. 2014. [1](#), [2](#)
- [2] Liangchieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected arfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017. [4](#)
- [3] Taegsang Cho, Charles L. Zitnick, Neel Joshi, Sing Bing Kang, Richard Szeliski, and William T. Freeman. Image restoration by matching gradient distributions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):683–694, April 2012. [1](#)
- [4] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: a deep network architecture for single-image rain removal. *IEEE Trans. Image Process.*, 26(6):2944–2956, July 2017. [1](#), [2](#)
- [5] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2017. [1](#), [2](#), [7](#)
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2015. [2](#)
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1904–1916, 2015. [4](#)
- [8] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2019. [1](#), [2](#)
- [9] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, Virtual, Jun. 2020. [5](#), [6](#), [7](#)
- [10] Liwei Kang, Chiawen Lin, and Yuhsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans. Image Process.*, 21(4):1742–1755, Apr. 2012. [1](#), [2](#)
- [11] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2015. [6](#)
- [12] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *Int. Conf. Mach. Learn.*, 2018. [1](#)
- [13] Anat Levin and Yair Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(9):1647–1654, Sep. 2007. [1](#), [3](#), [4](#), [5](#)
- [14] Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. Non-locally enhanced encoder-decoder network for single image de-raining. In *ACM Int. Conf. Multimedia*, Oct. 2018. [1](#)
- [15] Ruoteng Li, Loong-Fah Cheong, and Robby T. Tan. Heavy rain image restoration: integration physics model and conditional adversarial learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2019. [1](#), [2](#), [5](#), [6](#), [7](#)
- [16] Siyuan Li, Wenqi Ren, Jiawan Zhang, Jinke Yu, and Xiaojie Guo. Fast single image rain removal via a deep decomposition-composition network. *arXiv:1804.02688*, 2018. [6](#)
- [17] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Eur. Conf. Comput. Vis.*, Sep. 2018. [1](#), [2](#), [5](#), [6](#), [7](#)
- [18] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, pages 19–60, Mar. 2010. [2](#)
- [19] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: a better and simpler baseline. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2019. [1](#), [2](#)
- [20] YiChang Shih, Dilip Krishnan, Fredo Durand, and William T. Freeman. Reflection removal using ghosting cues. In *IEEE Conf. Comput. Vis. Pattern Recog.*, volume 2, June 2015. [1](#), [5](#)
- [21] M.J. Wainwright, E.P. Simoncelli, and A.S. Willsky. Random cascades of gaussian scale mixtures for natural images. In *IEEE Int. Conf. Image Process.* IEEE, Sep. 2000. [3](#)
- [22] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2020. [5](#), [6](#), [7](#)
- [23] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2019. [2](#), [5](#), [6](#), [7](#)
- [24] Yinglong Wang, Chen Chen, Shuyuan Zhu, and Bing Zeng. A framework of single-image deraining method based on analysis of rain characteristics. In *IEEE Int. Conf. Image Process.*, Sep. 2016. [2](#)
- [25] Yinglong Wang, Dong Gong, Jie Yang, Qinfeng Shi, Anton van den Hengel, Dehua Xie, and Bing Zeng. Deep single image deraining via modeling haze-like effect. *IEEE Trans. Multimedia*, Aug. 2020. [1](#), [5](#)
- [26] Yinglong Wang, Shuaicheng Liu, Chen Chen, and Bing Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE Trans. Image Process.*, 26(8):3936–3950, August 2017. [1](#), [2](#)
- [27] Yinglong Wang, Yibing Song, Chao Ma, and Bing Zeng. Re-thinking image deraining via rain streaks and vapors. In *Eur. Conf. Comput. Vis.*, Aug. 2020. [1](#), [7](#)
- [28] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, April 2004. [5](#)
- [29] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. Deep networks for image super-resolution with sparse prior. In *Int. Conf. Comput. Vis.*, Dec. 2015. [1](#)
- [30] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2017. [1](#), [2](#), [4](#)
- [31] Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Shuicheng Yan, and Zongming Guo. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, January 2019. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [32] Wenhan Yang, Shiqi Wang, Dejia Xu, Xiaodong Wang, and Jiaying Liu. Towards scale-free rain streak removal via self-supervised fractal band learning. In *AAAI*, Feb. 2020. [2](#)
- [33] W. Yang, Y. Yuan, W. Ren, J. Liu, W. Scheirer, Z. Wang, T. Zhang, Q. Zhong, D. Xie, S. Pu, Y. Zheng, Y. Qu, Y. Xie, L. Chen, Z. Li, C. Hong, H. Jiang, S. Yang, Y. Liu, X. Qu, P. Wan, S. Zheng, M. Zhong, T. Su, L. He, Y. Guo, Y. Zhao, Z. Zhu, J. Liang, and J. Wang. Advancing image understanding in poor visibility environments: A collective benchmark study. *IEEE Trans. Image Process.*, 29:5737–5752, March 2020. [3](#)
- [34] Rajeev Yasarla and Vishal M. Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. [3](#)
- [35] He Zhang and Vishal M. Patel. Density-aware single image de-raining using a multi-stream dense network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2018. [1](#), [2](#), [7](#)
- [36] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: a extremely efficient convolutional neural network for mobile devices. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2018. [4](#)
- [37] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Comput. Image.*, 3(1):47–57, Jan. 201. [1](#)