Explore Internal and External Similarity for Single Image Deraining with Graph Neural Networks

Cong Wang[1,2] , **Wei Wang**[1*] , **Chengjin Yu**[3] and **Jie Mu**[4]

[1]Shenzhen Campus, Sun Yat-sen University
[2]The Hong Kong Polytechnic University
[3]Anhui University
[4]Dongbei University of Finance and Economics
supercong94@gmail.com, wangwei29@mail.sysu.edu.cn, 23073@ahu.edu.cn, jiemu@dufe.edu.cn

Abstract

Patch-level non-local self-similarity is an important property of natural images. However, most existing methods do not consider this property into neural networks for image deraining, thus affecting recovery performance. Motivated by this property, we find that there exists significant patch recurrence property of a rainy image, that is, similar patches tend to recur many times in one image and its multi-scale images and external images. To better model this property for image detaining, we develop a multi-scale graph network with exemplars, called MSGNN, that contains two branches: 1) internal data-based supervised branch is used to model the internal relations of similar patches from the rainy image itself and its multi-scale images and 2) external data-participated unsupervised branch is used to model the external relations of the similar patches in the rainy image and exemplar. Specifically, we construct a graph model by searching the k-nearest neighboring patches from both the rainy images in a multi-scale framework and the exemplar. After obtaining the corresponding k neighboring patches from the multi-scale images and exemplar, we build a graph and aggregate them in an attentional manner so that the graph can provide more information from similar patches for image deraining. We embed the proposed graph in a deep neural network and train it in an end-to-end manner. Extensive experiments demonstrate that the proposed algorithm performs favorably against eight state-of-the-art methods on five public synthetic datasets and one real-world dataset. The source codes will be available at https://github.com/supersupercong/MSGNN.

1 Introduction

Images captured in a rainy environment are usually corrupted by rain streaks. These rainy images seriously degrade visibility, which accordingly interferes with the following high-level tasks, e.g., semantic segmentation, object detection, etc.
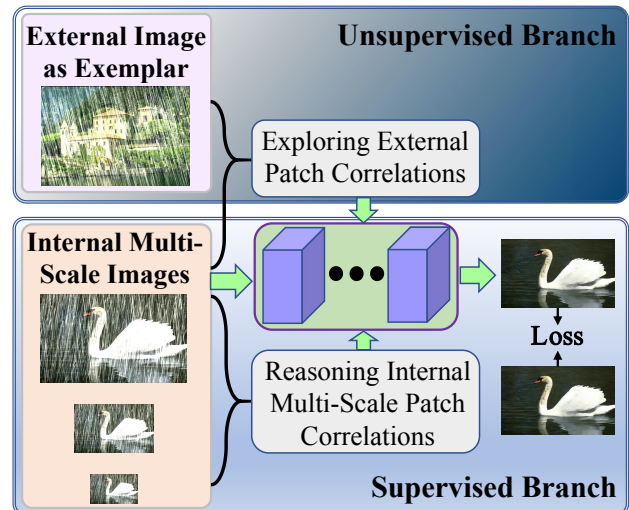
---

*Wei Wang is the corresponding author

Figure 1: Illustration of our main idea. The proposed method consists of two branches: **Supervised Branch** and **Unsupervised Branch**. The Supervised Branch reasons the internal patch correlation on the multi-scale images, while the unsupervised branch utilizes the learning ability of graph neural networks to explore the external patch correlation with unseen images as exemplars to learn more rainy conditions for better rain removal.

Therefore, it is of great need to develop an effective deraining algorithm to improve the quality of the images so that they can facilitate related applications.

Mathematically, the degradation model for image deraining is usually formulated as a linear combination of a rain streaks component R with a clean background image B:

$$O = B + R. \tag{1}$$

According to (1), image deraining is a highly ill-posed problem as the rain streaks R and the clear image B are both unknown. To make this problem well-posed, conventional methods [Luo *et al.*, 2015; Chen and Hsu, 2013; Li *et al.*, 2016] usually impose priors on rain streaks and clear images to constrain the solution spaces. However, these priors-based methods usually lead to complex optimization problems that are difficult to solve and have higher time consumption.
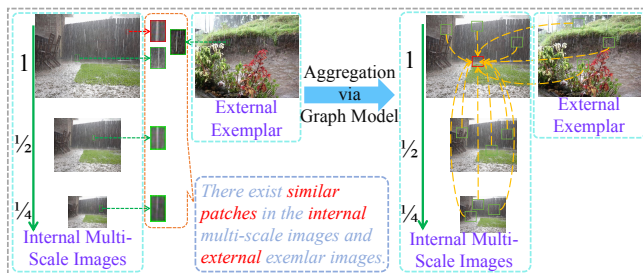
Figure 2: Illustration of the internal similarity in multi-scale images and external similarity in the exemplar. As can be seen, patch recurrence property indeed exists in internal multi-scale images and external exemplars.

Inspired by the success of the deep convolutional neural networks (CNNs) in many vision tasks, recent efforts have been devoted to applying CNNs to solve the image restoration [Jin *et al.*, 2022a; Jin *et al.*, 2023; Jin *et al.*, 2022b; Wang *et al.*, 2021; Wang *et al.*, 2022; Wang *et al.*, 2024a; Wang *et al.*, 2024c; Wang *et al.*, 2024b; Wang *et al.*, 2024d; Cui *et al.*, 2022; Zhu *et al.*, 2020; Xu *et al.*, 2019; Xu and He, 2022; Xu *et al.*, 2023] and also deraining problem [Yang *et al.*, 2017; Li *et al.*, 2018; Zhang and Patel, 2018; Wang *et al.*, 2019a; Wang *et al.*, 2019b; Wei *et al.*, 2019; Ren *et al.*, 2019; Wang *et al.*, 2020a; Yasarla *et al.*, 2020; Wang *et al.*, 2020c; Zamir *et al.*, 2021; Zhou *et al.*, 2021]. These deraining methods generally model the image deraining problem as a pixel-wise image regression process that directly learns the mapping from rain images to clear images by end-to-end trainable networks. Although these CNN-based approaches have achieved decent results, few of them consider patch recurrence into neural networks and hence are less effective in modeling the non-local self-similarity property of natural images. Although using large receptive fields can remedy this problem to some extent, it usually leads to larger or deeper networks that are difficult to train.

To better solve the image deraining task, we propose an effective method that explores the non-local similarity property of rainy images by a graph neural network. Figure 1 illustrates the main idea of our method that contains two branches: 1) internal data-based supervised branch is used to model the internal relations of similar patches from the rainy image itself and its multi-scale images and 2) external data-participated unsupervised branch is used to model the external relations of the similar patches in the rainy image and exemplar.

The proposed method is motivated by the patch recurrence property of natural images. We find that similar patches tend to recur many times in one image and its multi-scaled versions and external images, especially in rainy images as shown in Figure 2, where most image patches in rainy images have similar rain streaks. This property stimulates us to build the non-local graph modules that can aggregate these neighbors that have similar rain streaks into one query patch to achieve better deraining. To model the non-local similarity property of both the rainy image itself and external exem-

plars, we first model the internal relations of similar patches from the rainy image itself by a multi-scale graph network. Then, we explore the property of the exemplars and model the external relations of the similar patches in the rainy images and exemplars. Specifically, we construct a graph model by searching k-nearest neighboring patches in the multi-scale images and exemplar for each query patch of the input rainy image. We then obtain the corresponding k neighboring patches in the multi-scale images and exemplar and aggregate them with a graph attention mechanism. Finally, we formulate the constructed graph into an end-to-end trainable deep neural network to solve image deraining. The main contributions of this paper are summarized as follows:

- We propose a multi-scale graph module to explore internal non-local similarity by aggregating similar patches in multi-scale rainy images to the query patch of the input rainy image.

- We propose to use an exemplar image to explore external non-local similarity for enriching the representation of the query patch to furthermore improve the deraining quality.

- We analyze the proposed algorithm and show that it is able to remove rain streaks and preserve image details. Quantitative and qualitative experiments demonstrate that the proposed algorithm outperforms state-of-the-art methods on both synthetic and real-world datasets.

To the best of our knowledge, this is the first algorithm that explores the non-local property from both the rainy image itself and external exemplars by a graph neural network for single image deraining.

2 Related Work

2.1 Single Image Deraining

Priors-based deraining methods are typically built on some assumption about rain streaks and rainy images [Luo *et al.*, 2015; Chen and Hsu, 2013; Li *et al.*, 2016].

CNNs-based deraining approaches have dominated recent research and achieved great success. [Fu *et al.*, 2017] maps the high-frequency structure of a rainy image to negative residual rain streaks and obtains final rain-free images by (1). After that, a series of deraining methods are proposed [Li *et al.*, 2018; Wang *et al.*, 2019b; Ren *et al.*, 2019; Wang *et al.*, 2020b; Wang *et al.*, 2020a; Wang *et al.*, 2022]

2.2 Non-local Self-similarity for Image Restoration

The non-local self-similarity prior assumes that similar patches frequently recur in a natural image, which has been exploited by many classical image restoration methods [Dabov *et al.*, 2007; Buades *et al.*, 2005; Chang *et al.*, 2004]. Here, we briefly review recent approaches that exploit this prior with graph neural networks. Plötz . [Plötz and Roth, 2018] are the first to introduce a graph-based model for image denoising and image super-resolution. They propose a neural nearest neighbor network that learns to aggregate the neighbors of the query patch. Zhou *et al.* [Zhou *et al.*, 2020] propose a cross-scale graph network for image super-resolution.
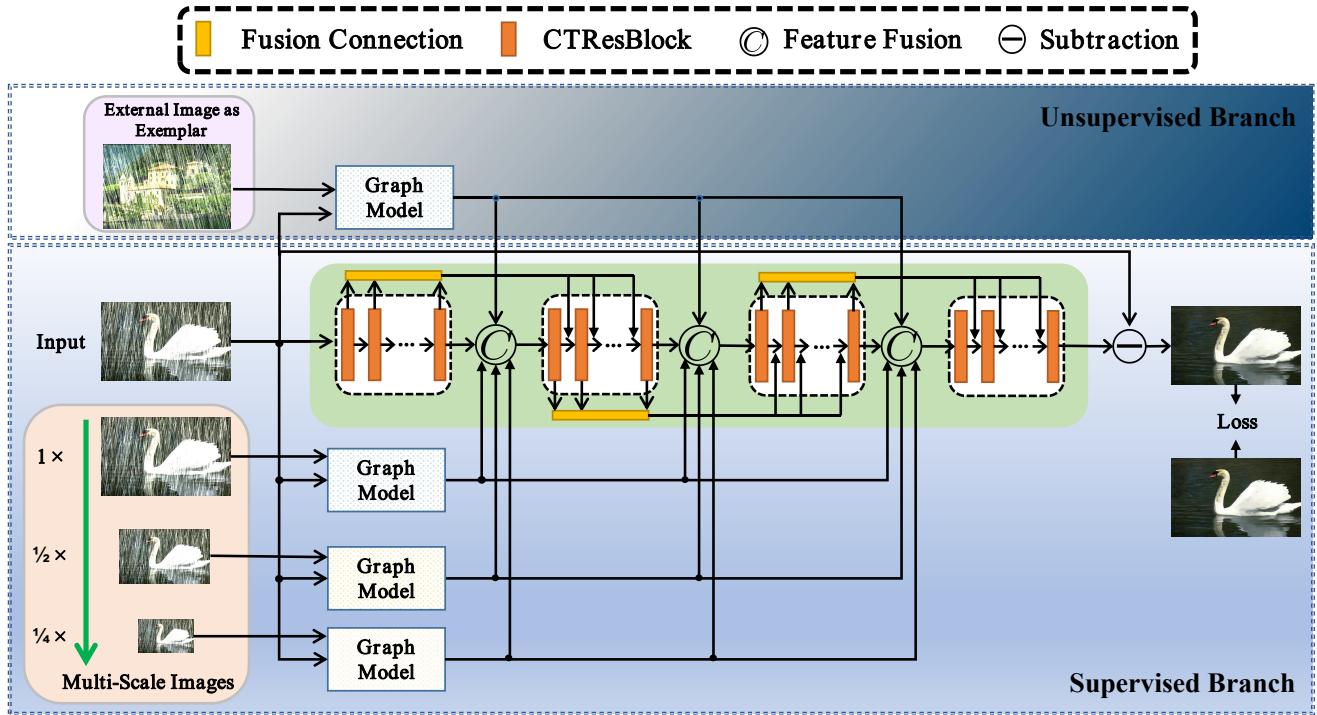
Figure 3: Diagram of the proposed MSGNN. Our method consists of two branches: an internal data-based supervised branch and an external data-participated unsupervised branch. In the supervised branch, we explore the internal property of multi-scale images by reasoning the multi-scale patch correlations to help the original-scale image learn more internal image information. In the unsupervised branch, we build the external patch correlations between input images and another external image as an exemplar by utilizing the superiority of the graph network which can search similar patches between any two samples so that the network is able to learn more rainy conditions for better rain removal. Graph Model is illustrated in Figure 4.

In this paper, we explore the non-local similarity property of rainy images by taking into consideration of multi-scale input images and external exemplars.

3 Proposed Method

3.1 Overall Framework

The overall framework is shown in Figure 3. Our method intends to learn the rain streaks from an input image via a graph-boosted CNN, and then subtract the learned rain streaks from the input image to obtain the deraining result. We express this process as follows:

$$\tilde{B} = \text{MSGNN}\Big(O, \mathcal{R}(O, O), \mathcal{R}(O, O_{\frac{1}{2}}), \mathcal{R}(O, O_{\frac{1}{4}}), \mathcal{R}(O, E)\Big),$$

$$(2)$$

where MSGNN denotes our proposed backbone CNN. $\tilde{B}$ refers to the estimated rain-free image. $O_{\frac{1}{2}}$ and $O_{\frac{1}{4}}$ refers to the $\frac{1}{2}$ and $\frac{1}{4}$ scaled images, respectively, while E represents the external exemplar. $\mathcal{R}(\cdot, \cdot)$ denotes a graph model that is divided into two steps: the nearest neighbor search and attentional aggregation.

MSGNN takes a rainy image O as input and reasons internal multi-scale patch correlation with its multi-scale images and explores the external patch correlation with an exemplar in the network to search the most similar patch information and aggregate them for better rain streaks removal.

The MSGNN contains N sub-networks with an identical structure, and each sub-network consists of M Channel Transformation [Yang *et al.*, 2020] Residual Blocks (CTResBolcks). Two adjacent sub-networks are connected via "fusion connection" (yellow blocks), which concatenates all the outputs of the convolution layers, then applies a 1×1 convolution on them, and sends the results to the next sub-network. The output of the graph model is injected into every direct connection between sub-networks via feature fusion, which concatenates all features together and then applies two convolution layers with kernel size 5×5 and stride size 2×2. The graph model is described later in the rest of this section, and the details of CTResBolck are provided in supplementary material.

To train the network, we minimize the negative structural similarity index measure (SSIM) [Wang *et al.*, 2004] between the estimated rain-free image $\tilde{B}$ and the ground truth B:

$$\mathcal{L} = -\text{SSIM}(\tilde{B}, B). \qquad (3)$$

3.2 Graph Model

Our graph model exploits both internal and external non-local similarity. Internal non-local similarity is utilized to identify similar patches to the query patch within the input image of different scales. The external non-local similarity is to find
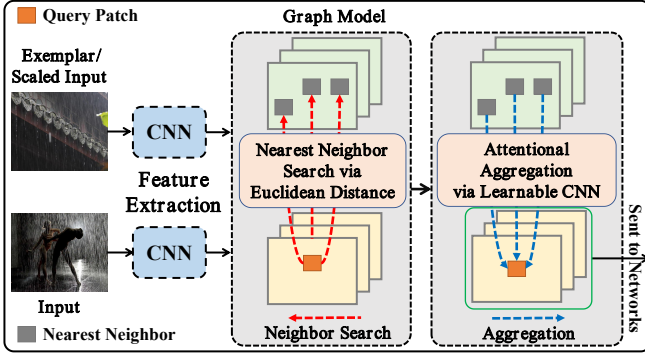
Figure 4: Illustration of the graph model which is divided into two steps: nearest neighbor search and attentional aggregation.

similar patches from another rainy image, the external exemplar.

As shown in Figure 4, the proposed graph model consists of the following steps: 1) extract features from images by a CNN; 2) for each query patch, search for patches with similar feature maps; 3) aggregate the similar patches with attention mechanism and then send the combined features to the backbone CNN to facilitate the deraining of the query patch.

Internal and External Similarity

As mentioned above, non-local self-similarity has been exploited in previous graph-based methods for various restoration tasks [Plötz and Roth, 2018; Zhou *et al.*, 2020]. However, these methods overlook information in multi-scale images. In this paper, we explore the non-local self-similarity property in multi-scale inputs as in [Jiang *et al.*, 2020]. Specifically, patch recurrence not only exists in the original image but also can be found across its different scaled versions due to the perspective nature of images – recurrent patterns may be located at different distances and thus have different visual sizes and are similar to each other only after properly scaled.

In addition to internal self-similarity, external images of the same domain, topic, or theme can also be similar to the input image and thus can provide more information for deraining. These external images are referred to as exemplars. Hence, in our model, we also explore the relation between the input image and an exemplar image for better deraining. Figure 2 provides a better illustration of the internal patch similarity in multi-scale images and external patch similarity in exemplars, which supports our motivation.

Nearest Neighbor Search

For a query patch of the input image, we want to find similar patches from each scaled input image and the exemplar image. Here, we take the exemplar image as an example to illustrate the graph construction process, which can be similarly applied to other images. The graph construction process includes the following steps: we first extract features from the original rainy image and the exemplar image via a three-layer CNN, and denote them by $\mathcal{F}_O$ and $\mathcal{F}_E$ respectively. Then, we divide $\mathcal{F}_O$ and $\mathcal{F}_E$ into patches of size $l \times l$. For each query patch in F_O, which is indexed by q and denoted by $\mathcal{P}_O^q$, we

find k nearest patches $\mathcal{P}_E^{n_r}$ among patches in $\mathcal{F}_E$, based on the Euclidean distance between their CNN-extracted features. Nearest patches $\mathcal{P}_E^{n_r}$ are indexed by n_r and $r = \{1, \cdots, k\}$.

As such, a k-nearest neighbor graph $\mathcal{G}_k(\mathcal{V}, \mathcal{E})$ is constructed. $\mathcal{V}$ is the patch set (vertices in the graph) and equal to $\mathcal{V}_O \cup \mathcal{V}_E$, the union of input rainy image patch set $\mathcal{V}_O$ and an exemplar patch set $\mathcal{V}_E$. Set $\mathcal{E}$ is the edge set with size $|\mathcal{E}| = |\mathcal{V}_O| \times k$, which indicates k nearest neighbors for each query patch in $\mathcal{V}_O$. For each edge, one of its two terminal vertices is the input rainy image patch and the other one is the exemplar patch.

Attentional Aggregation

With the k-nearest neighbor graph, we can perform attentional aggregation to combine the nearest neighbors of the query patch. Notice that the query patch itself is its closest neighbor. Specifically, the aggregation is done by weighted averaging:

$$\widehat{\mathcal{P}}_O^q = \frac{1}{\delta_q} \sum_{n_r \in S_q} \alpha_{n_r \to q} \mathcal{P}_E^{n_r}, \qquad (4)$$

where $\alpha_{n_r \to q}$ is the weight, S_q is the set of the k nearest neighbors of q, $\delta_q = \sum_{n_r \in S_q} \alpha_{n_r \to q}$ is the normalizing factor, and $\widehat{\mathcal{P}}_O^q$ is the aggregated patch representation.

The weight $\alpha_{n_r \to q}$ is computed by an attention mechanism, which is devised to measure the similarity between the query patch and its neighbors in the sense that similar neighbors should have a large weight. It is implemented as follows. First, we calculate the difference between the query and the neighbor:

$$\mathcal{D}^{n_r \to q} = \mathcal{P}_O^q - \mathcal{P}_E^{n_r}. \qquad (5)$$

Then, we obtain the attention weight by a CNN with trainable parameters θ:

$$\alpha_{n_r \to q} = \exp(\text{CNN}_\theta(\mathcal{D}^{n_r \to q})), \qquad (6)$$

where the exponential function $\exp(\cdot)$ ensures all weights are positive. Finally, for all query patches, we assemble the aggregated patch representations $\widehat{\mathcal{P}}_O^q$ into $\widehat{\mathcal{F}}_O$ via `patch2img` [Plötz and Roth, 2018] and send it to downstream networks.

4 Experimental Results

To demonstrate the effectiveness of our proposed method for single image deraining, we evaluate it on both synthetic datasets and real-world datasets. For a fair comparison, we ensure that all the baselines are retrained using the codes provided by the authors.

4.1 Datasets

Synthetic datasets. We conduct deraining experiments on five widely used synthetic datasets: Rain200H [Yang *et al.*, 2017], Rain200L [Yang *et al.*, 2017], Rain1200 [Zhang and Patel, 2018], Rain1400 [Fu *et al.*, 2017], Rain12 [Li *et al.*, 2016]. Note that we use the model trained on Rain200H to test on the Rain12 dataset since Rain12 does not have training images. We use the Rain100H as the analysis dataset.

Real-world dataset. Note that [Yang *et al.*, 2017; Li *et al.*, 2019; Wang *et al.*, 2020a] provide a mass of real-world rainy images. We use them as real-world datasets.

	RESCAN		PreNet		SpaNet		DCSFN		MSPFN		RCDNet		DualGCN		MOSS		MSGNN	
Dataset	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Rain200H	26.661	0.8419	27.640	0.8872	25.484	0.8584	28.469	0.9016	25.553	0.8039	28.698	0.8904	*28.758*	*0.9026*	25.283	0.8418	**29.627**	**0.9178**
Rain200L	36.993	0.9788	36.487	0.9792	36.075	0.9774	37.847	*0.9842*	30.367	0.9219	38.400	0.9841	*38.415*	0.9818	32.138	0.9563	**39.088**	**0.9869**
Rain1200	32.127	0.9028	27.307	0.8712	27.099	0.8082	*32.275*	*0.9228*	30.382	0.8860	32.273	0.9111	32.033	0.9163	31.668	0.9122	**33.110**	**0.9274**
Rain1400	30.969	0.9117	30.608	0.9181	29.000	0.8891	*31.493*	*0.9279*	31.514	0.9203	31.016	0.9164	30.567	0.9148	30.355	0.9177	**31.703**	**0.9299**
Rain12	32.965	0.9545	34.791	0.9644	33.217	0.9546	35.803	0.9683	34.253	0.9469	31.038	0.9069	*35.805*	*0.9687*	30.775	0.9317	**36.799**	**0.9707**
Parameter	0.15M		0.17M		0.28M		6.45M		21.00M		3.67M		2.73M		2.86M		2.46M	

Table 1: Quantitative results on five synthetic datasets. The best and second best results are marked in **bold** and *italic* respectively.

(a) Input   (b) PreNet   (c) SpaNet   (d) DCSFN   (e) MSPFN

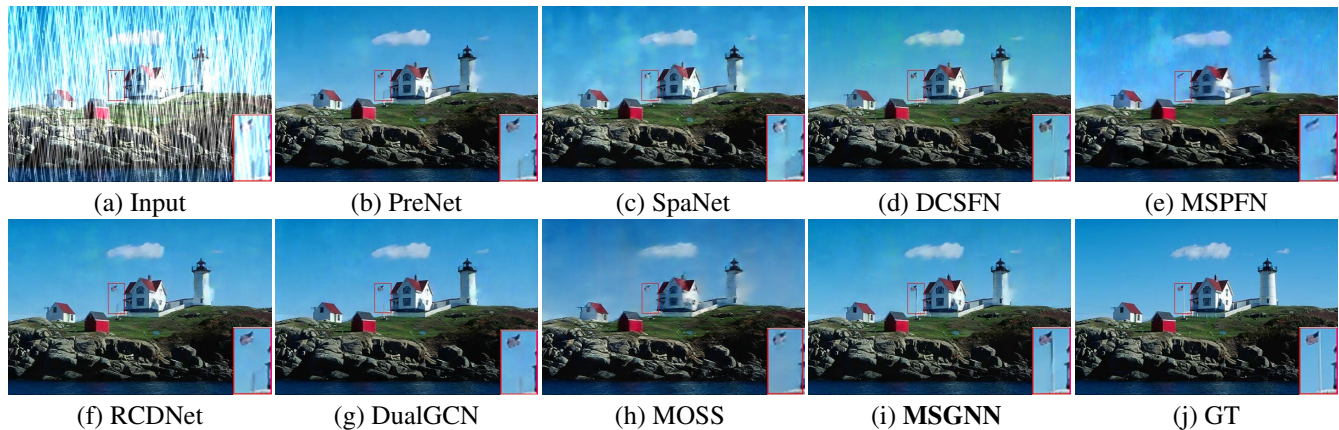(f) RCDNet   (g) DualGCN   (h) MOSS   (i) **MSGNN**   (j) GT

Figure 5: Comparisons with state-of-the-art methods on synthetic dataset. Our proposed MSGNN is able to restore better texture.

4.2 Implementing Details

We randomly select a rainy image as an exemplar from the dataset as input to the network. Although the exemplar is randomly chosen, it is able to improve the deraining performance, as shown in Figure 9. The number of channels is set as 32 and the nonlinear activation we used is LeakyReLU with $\alpha = 0.2$ for all convolution layers except for the last one. For the last layer, the channel is 3 without any activation function. The patch size of input is 64×64, and the batch size is 8. We use ADAM [Kingma and Ba, 2015] to optimize our network parameters. The initial learning rate is 0.0005, and the rate will be divided by 10 at 300 and 400 epochs, and the model training terminates after 500 epochs. The number of sub-network N is 4 and the number of CTResBlocks M in each sub-network is 8. The patch size l is 3 and the stride length s is 3. The number of nearest neighbors k is set as 5.

4.3 Results on Synthetic Datasets

We compare our proposed network with eight state-of-the-art methods on five synthetic datasets, and the results are summarized in Table 1. It can be seen that our proposed method achieves the best results on all the datasets in terms of PSNR and SSIM. Moreover, we note that the number of parameters of our model is 62% less compared with DCSFN [Wang *et al.*, 2020b], while the SSIM is increased by 1.62% on the Rain200H dataset. In Figure 5, we also visualize several deraining results on the synthetic datasets. It can be seen that our method is able to restore better details and textures and obtain cleaner and clearer background images, while other

approaches hand down some rain streaks or lose some details.

4.4 Results on the Real-world Dataset

We further demonstrate the effectiveness of the proposed method by the comparison with state-of-the-art methods on the real-world dataset in Figure 6. It can be seen that the proposed algorithm is able to generate cleaner and clearer deraining results, while other methods hand down some rain streaks. This shows the effectiveness of the proposed method in deraining real-world images.

4.5 Ablation Study

Analysis on the Basic Components. In our model, we design the graph-based deep network using the fusion connection, channel attention (Channel Transformation), and graph model. We analyze them and present the results in Table **??**. First, it can be seen that the results between M_1 and M_2 show that the fusion connection (FC) can improve the deraining performance. Moreover, as a comparison, we also analyze other channel attention modules, ECA [Wang *et al.*, 2020d] and SE [Hu *et al.*, 2018]. We can see that ECA is less effective for our model and SE can improve the deraining results in terms of PSNR and SSIM. Furthermore, CT is the best channel attention module among the three. At last, we can observe that our graph model can boost the deraining results. Hence, the fusion connection, CT, and graph modules applied in our framework are all meaningful for deraining.

Effect of the Stride Length. In Table 3, we present the results of different strides. Since the number of graph patches

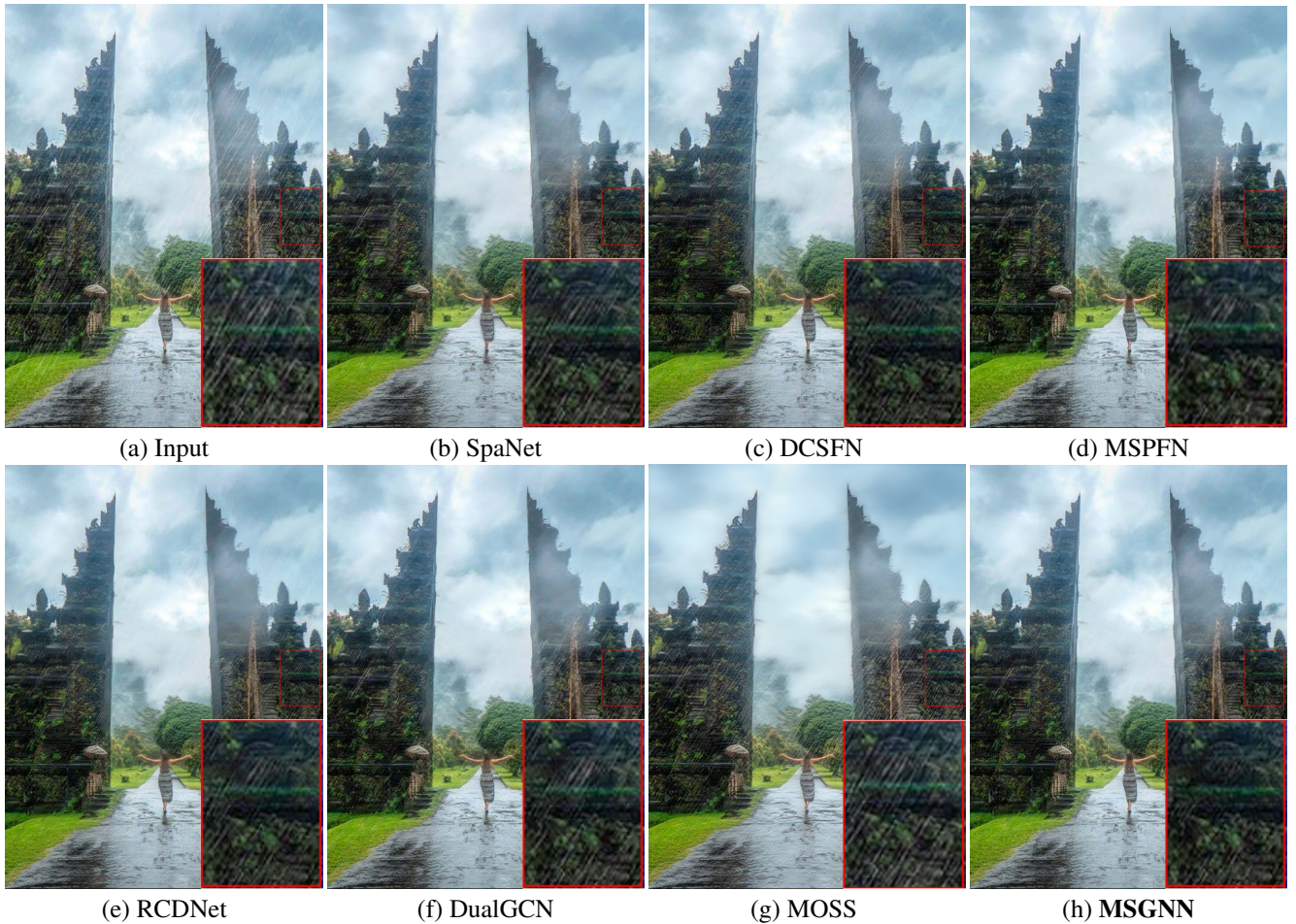| (a) Input	(b) SpaNet	(c) DCSFN	(d) MSPFN
(e) RCDNet	(f) DualGCN	(g) MOSS	(h) **MSGNN**

Figure 6: Comparisons with state-of-the-art methods on real-world dataset. The proposed MSGNN is able to produce a clearer result.

Experiments	M_1	M_2	M_3	M_4	M_5	M_6
FC	✗	✔	✔	✔	✔	✔
ECA	✗	✗	✔	✗	✗	✗
SE	✗	✗	✗	✔	✗	✗
CT	✗	✗	✗	✗	✔	✔
Graph	✗	✗	✗	✗	✗	✔
PSNR	29.161	29.451	27.535	29.397	29.488	29.627
SSIM	0.9111	0.9147	0.8944	0.9149	0.9163	0.9178

Table 2: Ablation results on different components. The ✔ and ✗ symbols denote the corresponding component whether is adopted or not.

	Training Patch is 64×64			Training Patch is 48×48		
s	1	2	3	1	2	3
PSNR	-	29.472	29.627	29.011	28.942	29.016
SSIM	-	0.9168	0.9178	0.9116	0.9112	0.9119

Table 3: Effect of the stride length.

with stride 1 is increased sharply when the training patch size is 64×64, which is out of our computer memory, we adjust the training patch size to 48×48. We can observe that the model achieves the best results when the stride is 3 under the two conditions. The stride being 3 means that the obtained graph patches are not overlapped because the graph patch size is also 3. Moreover, we find an interesting phenomenon from Table 3 that the deraining results are influenced by the train-ing patch size, and the larger the training patch size, the better the results.

Effect of the Patch Size. We analyze the effect of the size of the graph patch, i.e., l, and the results are reported in Table 4. We can see the deraining performance reaches its best when the patch size is 3 in both PSNR and SSIM compared to the cases of patch size being 5 and 7. Further, we can observe that the PSNR decreases as l increases. This demonstrates that a bigger patch size cannot help improve the deraining perfor-mance. As such, we set the default graph patch size as 3.

Effect of the Number of Nearest Neighbors. We analyze the effect of the number of nearest neighbors, i.e., k, and the results are presented in Table 5. It can be observed that the results are the best when $k = 5$ in terms of PSNR and SSIM.

l	3	5	7
PSNR	29.627	29.606	29.584
SSIM	0.9178	0.9175	0.9178

Table 4: Effect of the size of graph patch.

k	3	5	7
PSNR	29.501	29.627	29.565
SSIM	0.9167	0.9178	0.9172

Table 5: Analysis on the number of nearest neighbors.
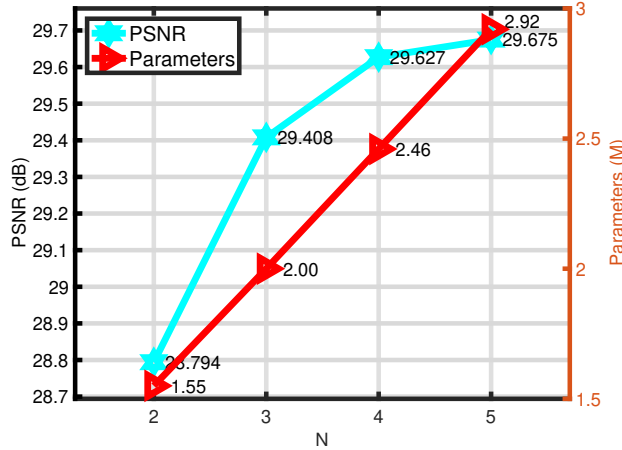
Figure 7: Results on the number of sub-networks.

(a) Input   (b) w/o 1   (c) w/o $\frac{1}{2}$&$\frac{1}{4}$

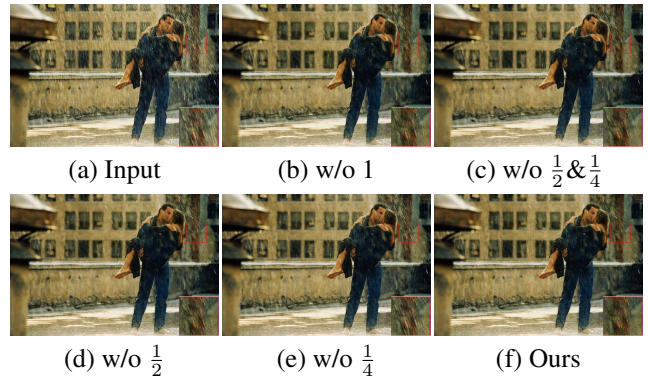(d) w/o $\frac{1}{2}$ (e) w/o $\frac{1}{4}$ (f) Ours

Figure 8: The effectiveness of exploiting multi-scale images on a real-world image. The proposed internal multi-scale image strategy is able to help better deraining.

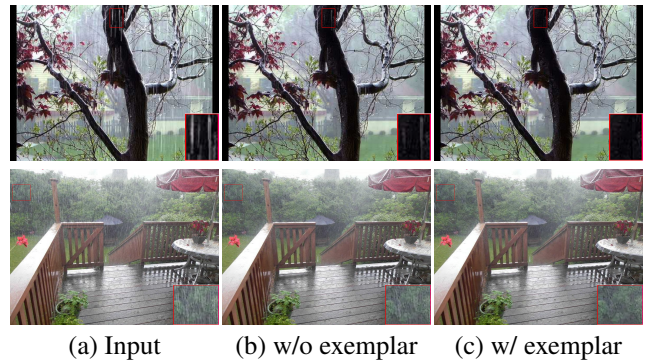(a) Input   (b) w/o exemplar   (c) w/ exemplar

Figure 9: Effectiveness of the exemplar on the real-world dataset. The proposed external images as exemplars are able to improve image deraining quality.

And, both $k = 3$ and $k = 7$ perform worse than $k = 5$. Hence, we select $k = 5$ as the default setting of the network. **Analysis of the Number of Sub-networks.** We analyze the effect of the number of sub-networks N, and the results are presented in Figure 7. It can be seen that $N = 4$ achieves good results and the deraining performance will not be significantly improved as N increases in terms of PSNR. Therefore, we set $N = 4$ as our default setting due to the trade-off between the deraining performance and the number of parameters. Note that the model also achieves state-of-the-art performance compared with other methods on the most challenging Rain200H [Yang *et al.*, 2017] when $N = 2$, while the number of parameters only has 1.55M.

Analysis on the Use of Multi-Scale Images. Since our graph neural network is based on the multi-scale aggregation module, we provide an analysis on its effectiveness. We show the results on the real-world dataset in Figure 8. We can observe that the default model with all scales obtains clearer and cleaner results, while other models always hand down some rain streaks. Hence, the use of multi-scale input images in our framework is helpful for the deraining task.

Effect of the Exemplar. Here, we demonstrate the effectiveness of the exemplar on deraining, and the results are shown in Figure 9. We can observe that the model with an exem-

plar is able to improve the deraining performance, while the model without an exemplar hands down some rain streaks. It illustrates that our proposed model with exemplars can effectively utilize more information on rainy images, which makes the model more robust under different rainy conditions.

5 Conclusion

In this paper, we have proposed a deep multi-scale graph neural network with exemplars, called MSGNN, for image deraining. By exploiting internal non-local similarity in multi-scale input images and external non-local similarity in an exemplar image with a graph model and attention mechanism, our model achieves favorable results on both synthetic and real-world datasets compared with state-of-the-art methods, demonstrating the effectiveness of our deraining model. In future work, we plan to explore different strategies for selecting the exemplar image for better deraining.

References

[Buades *et al.*, 2005] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *CVPR*, pages 60–65, 2005.

[Chang *et al.*, 2004] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *CVPR*, pages 275–282, 2004.

[Chen and Hsu, 2013] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *ICCV*, pages 1968–1975, 2013.

[Cui *et al.*, 2022] Xin Cui, Cong Wang, Dongwei Ren, Yunjin Chen, and Pengfei Zhu. Semi-supervised image deraining using knowledge distillation. *IEEE TCSVT*, 32(12):8327–8341, 2022.

[Dabov *et al.*, 2007] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen O. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE TIP*, 16(8):2080–2095, 2007.

[Fu *et al.*, 2017] 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 1715–1723, 2017.

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

[Jiang *et al.*, 2020] 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 8343–8352, 2020.

[Jin *et al.*, 2022a] Yeying Jin, Wending Yan, Wenhan Yang, and Robby T Tan. Structure representation network and uncertainty feedback learning for dense non-uniform fog removal. In *ACCV*, pages 2041–2058, 2022.

[Jin *et al.*, 2022b] Yeying Jin, Wenhan Yang, and Robby T Tan. Unsupervised night image enhancement: When layer decomposition meets light-effects suppression. In *ECCV*, pages 404–421, 2022.

[Jin *et al.*, 2023] Yeying Jin, Beibei Lin, Wending Yan, Yuan Yuan, Wei Ye, and Robby T Tan. Enhancing visibility in nighttime haze images using guided apsf and gradient adaptive convolution. In *ACM MM*, pages 2446–2457, 2023.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Li *et al.*, 2016] 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.

[Li *et al.*, 2018] 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 262–277, 2018.

[Li *et al.*, 2019] 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.

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

[Plötz and Roth, 2018] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NeurIPS*, pages 1095–1106, 2018.

[Ren *et al.*, 2019] 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.

[Wang *et al.*, 2004] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[Wang *et al.*, 2019a] Guoqing Wang, Changming Sun, and Arcot Sowmya. Erl-net: Entangled representation learning for single image de-raining. In *ICCV*, 2019.

[Wang *et al.*, 2019b] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson W. H. Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, pages 12270–12279, 2019.

[Wang *et al.*, 2020a] Cong Wang, Yutong Wu, Zhixun Su, and Junyang Chen. Joint self-attention and scale-aggregation for self-calibrated deraining network. In *ACM MM*, pages 2517–2525, 2020.

[Wang *et al.*, 2020b] Cong Wang, Xiaoying Xing, Yutong Wu, Zhixun Su, and Junyang Chen. DCSFN: deep cross-scale fusion network for single image rain removal. In *ACM MM*, pages 1643–1651, 2020.

[Wang *et al.*, 2020c] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, pages 3100–3109, 2020.

[Wang *et al.*, 2020d] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Ecanet: Efficient channel attention for deep convolutional neural networks. In *CVPR*, pages 11531–11539, 2020.

[Wang *et al.*, 2021] Cong Wang, Xiaoying Xing, Guangle Yao, and Zhixun Su. Single image deraining via deep shared pyramid network. *The Visual Computer*, 37:1851–1865, 2021.

[Wang *et al.*, 2022] Cong Wang, Jinshan Pan, and Xiao-Ming Wu. Online-updated high-order collaborative networks for single image deraining. In *AAAI*, volume 36, pages 2406–2413, 2022.

[Wang *et al.*, 2024a] Cong Wang, Jinshan Pan, Yeying Jin, Liyan Wang, Wei Wang, Gang Fu, Wenqi Ren, and Xiaochun Cao. How powerful potential of attention on image restoration? *arXiv preprint arXiv:2403.10336*, 2024.

[Wang *et al.*, 2024b] Cong Wang, Jinshan Pan, Wanyu Lin, Jiangxin Dong, Wei Wang, and Xiao-Ming Wu. Self-promer: Self-prompt dehazing transformers with depth-consistency. In *AAAI*, volume 38, pages 5327–5335, 2024.

[Wang *et al.*, 2024c] Cong Wang, Jinshan Pan, Wei Wang, Jiangxin Dong, Mengzhu Wang, Yakun Ju, and Junyang Chen. Promptrestorer: A prompting image restoration method with degradation perception. *NeurIPS*, 36, 2024.

[Wang *et al.*, 2024d] Cong Wang, Jinshan Pan, Wei Wang, Gang Fu, Siyuan Liang, Mengzhu Wang, Xiao-Ming Wu, and Jun Liu. Correlation matching transformation transformers for uhd image restoration. In *AAAI*, volume 38, pages 5336–5344, 2024.

[Wei *et al.*, 2019] 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.

[Xu and He, 2022] Kepeng Xu and Gang He. Dnas: A decoupled global neural architecture search method. In *CVPR workshops*, pages 1979–1985, 2022.

[Xu *et al.*, 2019] Kepeng Xu, Yunye Zhang, Wenxin Yu, Zhiqiang Zhang, Jingwei Lu, Yibo Fan, Gang He, and Zhuo Yang. Segmentation of building footprints with xception and iouloss. In *ICME workshops*, pages 420–425, 2019.

[Xu *et al.*, 2023] Kepeng Xu, Gang He, Li Xu, Xingchao Yang, Ming Sun, Yuzhi Wang, Zijia Ma, Haoqiang Fan, and Xing Wen. Towards robust sdrtv-to-hdrtv via dual inverse degradation network. *arXiv preprint arXiv:2307.03394*, 2023.

[Yang *et al.*, 2017] 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 1685–1694, 2017.

[Yang *et al.*, 2020] Zongxin Yang, Linchao Zhu, Yu Wu, and Yi Yang. Gated channel transformation for visual recognition. In *CVPR*, pages 11791–11800, 2020.

[Yasarla *et al.*, 2020] Rajeev Yasarla, Vishwanath A. Sindagi, and Vishal M. Patel. Syn2real transfer learning for image deraining using gaussian processes. In *CVPR*, pages 2723–2733, 2020.

[Zamir *et al.*, 2021] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021.

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

[Zhou *et al.*, 2020] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. In *NeurIPS*, 2020.

[Zhou *et al.*, 2021] Man Zhou, Jie Xiao, Yifan Chang, Xueyang Fu, Aiping Liu, Jinshan Pan, and Zheng-Jun Zha. Image de-raining via continual learning. In *CVPR*, pages 4907–4916, 2021.

[Zhu *et al.*, 2020] Honghe Zhu, Cong Wang, Yajie Zhang, Zhixun Su, and Guohui Zhao. Physical model guided deep image deraining. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020.