Multi-Stage Progressive Image Restoration

Syed Waqas Zamir[*1] Aditya Arora[*1] Salman Khan[2] Munawar Hayat[3]
Fahad Shahbaz Khan[2] Ming-Hsuan Yang[4,5,6] Ling Shao[1,2]
[1]Inception Institute of AI [2]Mohamed bin Zayed University of AI [3]Monash University
[4]University of California, Merced [5]Yonsei University [6]Google Research

Abstract

Image restoration tasks demand a complex balance between spatial details and high-level contextualized information while recovering images. In this paper, we propose a novel synergistic design that can optimally balance these competing goals. Our main proposal is a multi-stage architecture, that progressively learns restoration functions for the degraded inputs, thereby breaking down the overall recovery process into more manageable steps. Specifically, our model first learns the contextualized features using encoder-decoder architectures and later combines them with a high-resolution branch that retains local information. At each stage, we introduce a novel per-pixel adaptive design that leverages in-situ supervised attention to reweight the local features. A key ingredient in such a multi-stage architecture is the information exchange between different stages. To this end, we propose a two-faceted approach where the information is not only exchanged sequentially from early to late stages, but lateral connections between feature processing blocks also exist to avoid any loss of information. The resulting tightly interlinked multi-stage architecture, named as MPRNet, delivers strong performance gains on ten datasets across a range of tasks including image deraining, deblurring, and denoising. The source code and pre-trained models are available at https://github.com/swz30/MPRNet.

1. Introduction

Image restoration is the task of recovering a clean image from its degraded version. Typical examples of degradation include noise, blur, rain, haze, etc. It is a highly ill-posed problem as there exist infinite feasible solutions. In order to restrict the solution space to valid/natural images, existing restoration techniques [19, 29, 39, 59, 66, 67, 100] explicitly use image priors that are handcrafted with empirical observations. However, designing such priors is a challenging task and often not generalizable. To ameliorate this issue, recent state-of-the-art approaches [17, 44, 57, 86, 87, 93, 94, 97] employ convolutional neural networks (CNNs) that implicitly learn more general priors by capturing natural image statistics from large-scale data.

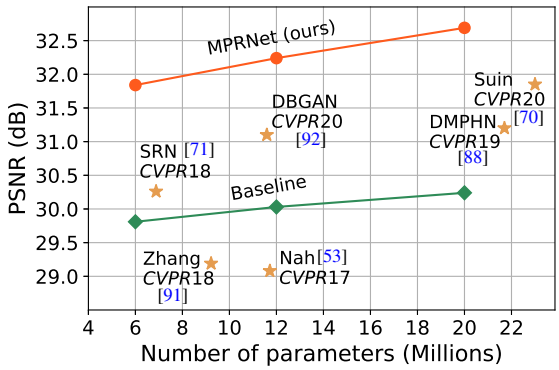


Figure 1: Image deblurring on the GoPro dataset [53]. Under different parameter capacities (x-axis), our multi-stage approach performs better than the single-stage baseline [65] (with channel attention [95]), as well as the state-of-the-art (PSNR on y-axis).

The performance gain of CNN-based methods over the others is primarily attributed to its model design. Numerous network modules and functional units for image restoration have been developed including recursive residual learning [4, 95], dilated convolutions [4, 81], attention mechanisms [17, 86, 96], dense connections [73, 75, 97], encoder-decoders [7, 13, 43, 65], and generative models [44, 62, 90, 92]. Nevertheless, nearly all of these models for low-level vision problems are based on *single-stage* design. In contrast, *multi-stage* networks are shown to be more effective than their single-stage counterparts in high-level vision problems such as pose-estimation [14, 46, 54], scene parsing [15] and action segmentation [20, 26, 45].

Recently, few efforts have been made to bring the multi-stage design to image deblurring [70, 71, 88], and image deraining [47, 63]. We analyze these approaches to identify the architectural bottlenecks that hamper their performance. First, existing multi-stage techniques either employ the *encoder-decoder* architecture [71, 88] which is effective in encoding broad contextual information but unreliable in preserving spatial image details, or use a *single-scale pipeline* [63] that provides spatially accurate but semanti-

[*]Equal contribution

cally less reliable outputs. However, we show that the combination of both design choices in a multi-stage architecture is needed for effective image restoration. Second, we show that naively passing the output of one stage to the next stage yields suboptimal results [53]. Third, unlike in [88], it is important to provide ground-truth supervision at each stage for progressive restoration. Finally, during multi-stage processing, a mechanism to propagate intermediate features from earlier to later stages is required to preserve contextualized features from the encoder-decoder branches.

We propose a multi-stage progressive image restoration architecture, called MPRNet, with several key components. 1). The earlier stages employ an encoder-decoder for learning multi-scale contextual information, while the last stage operates on the original image resolution to preserve fine spatial details. 2). A supervised attention module (SAM) is plugged between every two stages to enable progressive learning. With the guidance of ground-truth image, this module exploits the previous stage prediction to compute attention maps that are in turn used to refine the previous stage features before being passed to the next stage. 3). A mechanism of cross-stage feature fusion (CSFF) is added that helps propagating multi-scale contextualized features from the earlier to later stages. Furthermore, this method eases the information flow among stages, which is effective in stabilizing the multi-stage network optimization.

The main contributions of this work are:

- A novel multi-stage approach capable of generating contextually-enriched and spatially accurate outputs. Due to its multi-stage nature, our framework breaks down the challenging image restoration task into sub-tasks to progressively restore a degraded image.
- An effective supervised attention module that takes full advantage of the restored image at every stage in refining incoming features before propagating them further.
- A strategy to aggregate multi-scale features across stages.
- We demonstrate the effectiveness of our MPRNet by setting new state-of-the-art on **ten** synthetic and real-world datasets for various restoration tasks including image deraining, deblurring, and denoising while maintaining a low complexity (see Fig. 1). Further, we provide detailed ablations, qualitative results, and generalization tests.

2. Related Work

Recent years have witnessed a paradigm shift from high-end DSLR cameras to smartphone cameras. However, capturing high-quality images with smartphone cameras is challenging. Image degradations are often present in images either due to the limitations of cameras and/or adverse ambient conditions. Early restoration approaches are based on total variation [10, 67], sparse coding [3, 51, 52], self-similarity [8, 16], gradient prior [68, 80], *etc*. Recently, CNN-based restoration methods have achieved state-of-the-art results [57, 70, 86, 93, 97]. In terms of architectural design, these methods can be broadly categorized as single-stage and multi-stage.

Single-Stage Approaches. Currently, the majority of image restoration methods are based on a single-stage design, and the architectural components are usually based on those developed for high-level vision tasks. For example, residual learning [30] has been used to perform image denoising [2, 72, 93], image deblurring [42, 43] and image deraining [37]. Similarly, to extract multi-scale information, the encoder-decoder [65] and dilated convolution [83] models are often used [4, 28, 43]. Other single-stage approaches [5, 89, 97] incorporate dense connections [34].

Multi-Stage Approaches. These methods [24, 47, 53, 63, 70, 71, 88, 99] aim to recover clean image in a progressive manner by employing a light-weight subnetwork at each stage. Such a design is effective since it decomposes the challenging image restoration task into smaller easier subtasks. However, a common practice is to use the identical subnetwork for each stage which may yield suboptimal results, as shown in our experiments (Section 4).

Attention. Driven by its success in high-level tasks such as image classification [31, 32, 79], segmentation [21, 35] and detection [74, 79], attention modules have been used in low-level vision tasks [38]. Examples abound, including methods for image deraining [37, 47], deblurring [61, 70], super-resolution [17, 95], and denoising [4, 86]. The main idea is to capture long-range inter-dependencies along spatial dimensions [98], channel dimensions [32], or both [79].

3. Multi-Stage Progressive Restoration

The proposed framework for image restoration, shown in Fig. 2, consists of three stages to progressively restore images. The first two stages are based on encoder-decoder subnetworks that learn the broad contextual information due to large receptive fields. Since image restoration is a position-sensitive task (which requires pixel-to-pixel correspondence from the input to output), the last stage employs a subnetwork that operates on the original input image resolution (without any downsampling operation), thereby preserving the desired fine texture in the final output image.

Instead of simply cascading multiple stages, we incorporate a supervised attention module between every two stages. With the supervision of ground-truth images, our module rescales the feature maps of the previous stage before passing them to the next stage. Furthermore, we introduce a cross-stage feature fusion mechanism where the intermediate multi-scale contextualized features of the earlier subnetwork help consolidating the intermediate features of the latter subnetwork.

Although MPRNet stacks multiple stages, each stage has

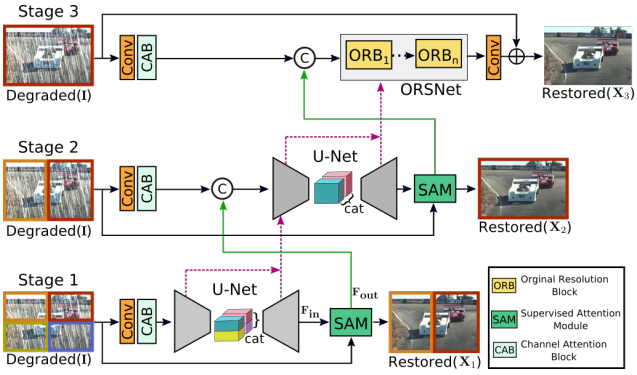


Figure 2: Proposed multi-stage architecture for progressive image restoration. Earlier stages employ encoder-decoders to extract multi-scale contextualized features, while the last stage operates at the original image resolution to generate spatially accurate outputs. A supervised attention module is added between every two stages that learns to refine features of one stage before passing them to the next stage. Dotted pink arrows represent the cross-stage feature fusion mechanism.

an access to the input image. Similar to the recent restoration methods [70, 88], we adapt the multi-patch hierarchy on the input image and split the image into non-overlapping patches: four for stage-1, two for stage-2, and the original image for the last stage, as shown in Fig. 2.

At any given stage S, instead of directly predicting a restored image $\mathbf{X}_S$, the proposed model predicts a residual image $\mathbf{R}_S$ to which the degraded input image $\mathbf{I}$ is added to obtain: $\mathbf{X}_S = \mathbf{I} + \mathbf{R}_S$. We optimize our MPRNet end-to-end with the following loss function:

$$\mathcal{L} = \sum_{S=1}^{3} \left[\mathcal{L}_{char}(\mathbf{X}_S, \mathbf{Y}) + \lambda \mathcal{L}_{edge}(\mathbf{X}_S, \mathbf{Y}) \right], \quad (1)$$

where $\mathbf{Y}$ represents the ground-truth image, and $\mathcal{L}_{char}$ is the Charbonnier loss [12]:

$$\mathcal{L}_{char} = \sqrt{\|\mathbf{X}_S - \mathbf{Y}\|^2 + \varepsilon^2}, \quad (2)$$

with constant ε empirically set to 10^{-3} for all the experiments. In addition, $\mathcal{L}_{edge}$ is the edge loss, defined as:

$$\mathcal{L}_{edge} = \sqrt{\|\Delta(\mathbf{X}_S) - \Delta(\mathbf{Y})\|^2 + \varepsilon^2}, \quad (3)$$

where Δ denotes the Laplacian operator. The parameter λ in Eq. (1) controls the relative importance of the two loss terms, which is set to 0.05 as in [37]. Next, we describe each key element of our method.

3.1. Complementary Feature Processing

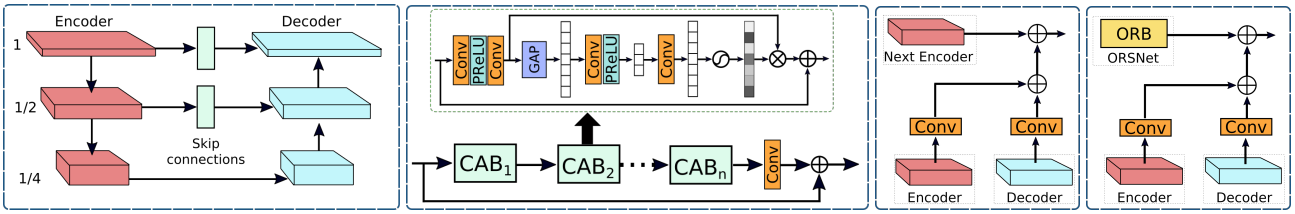
Existing single-stage CNNs for image restoration typically use one of the following architecture designs: 1). An encoder-decoder, or 2). A single-scale feature pipeline. The encoder-decoder networks [7, 13, 43, 65] first gradually map the input to low-resolution representations, and then progressively apply reverse mapping to recover the original resolution. While these models effectively encode multi-scale information, they are prone to sacrificing spatial details due to the repeated use of downsampling operation. In contrast, the approaches that operate on single-scale feature pipeline are reliable in generating images with fine spatial details [6, 18, 93, 97]. However, their outputs are semantically less robust due to the limited receptive field. This indicates the inherent limitations of the aforementioned architecture design choices that are capable of generating either spatially accurate or contextually reliable outputs, but not both. To exploit the merits of both designs, we propose a multi-stage framework where earlier stages incorporate the encoder-decoder networks, and the final stage employs a network that operates on the original input resolution.

Encoder-Decoder Subnetwork. Figure 3a shows our encoder-decoder subnetwork, which is based on the standard U-Net [65], with the following components. First, we add channel attention blocks (CABs) [95] to extract features at each scale (See Fig. 3b for CABs). Second, the feature maps at U-Net skip connections are also processed with the CAB. Finally, instead of using Transposed convolution for increasing spatial resolution of features in the decoder, we use bilinear upsampling followed by a convolution layer. This helps reduce checkerboard artifacts in the output image that often arise due to the Transposed convolution [55].

Original Resolution Subnetwork. In order to preserve fine details from the input image to the output image, we introduce the original-resolution subnetwork (ORSNet) in the last stage (see Fig. 2). ORSNet does not employ any downsampling operation and generates spatially-enriched high-resolution features. It consists of multiple original-resolution blocks (ORBs), each of which further contains CABs. The schematic of ORB is illustrated in Fig. 3b.

3.2. Cross-stage Feature Fusion

In our framework, we introduce the CSFF module between two encoder-decoders (see Fig. 3c), and between encoder-decoder and ORSNet (see Fig. 3d). Note that the features from one stage are first refined with 1×1 convolutions before propagating them to the next stage for aggregation. The proposed CSFF has several merits. First, it makes the network less vulnerable by the information loss due to repeated use of up- and down-sampling operations in the encoder-decoder. Second, the multi-scale features of one stage help enriching the features of the next stage. Third, the network optimization procedure becomes more stable as it eases the flow of information, thereby allowing us to add several stages in the overall architecture.



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

Figure 3: (a) Encoder-decoder subnetwork. (b) Illustration of the original resolution block (ORB) in our ORSNet subnetwork. Each ORB contains multiple channel attention blocks. GAP represents global average pooling [49]. (c) Cross-stage feature fusion between stage 1 and stage 2. (d) CSFF between stage 2 and the last stage.

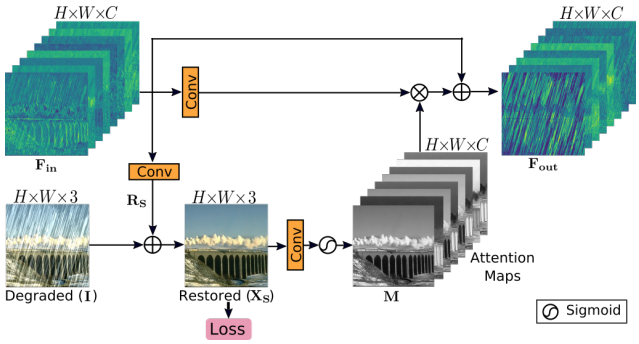


Figure 4: Supervised attention module.

3.3. Supervised Attention Module

Recent multi-stage networks for image restoration [70, 88] directly predict an image at each stage, which is then passed to the next consecutive stage. Instead, we introduce a supervised attention module between every two stages, which facilitates achieving significant performance gain. The schematic diagram of SAM is shown in Fig. 4, and its contributions are two-fold. First, it provides ground-truth supervisory signals useful for the progressive image restoration at each stage. Second, with the help of locally supervised predictions, we generate attention maps to suppress the less informative features at the current stage and only allow the useful ones to propagate to the next stage.

As illustrated in Fig. 4, SAM takes the incoming features $\mathbf{F_{in}} \in \mathbb{R}^{H\times W\times C}$ of the earlier stage and first generates a residual image $\mathbf{R}_S \in \mathbb{R}^{H\times W\times 3}$ with a simple 1×1 convolution, where $H\times W$ denotes the spatial dimension and C is the number of channels. The residual image is added to the degraded input image $\mathbf{I}$ to obtain the restored image $\mathbf{X}_S \in \mathbb{R}^{H\times W\times 3}$. To this predicted image $\mathbf{X}_S$, we provide explicit supervision with the ground-truth image. Next, per-pixel attention masks $\mathbf{M} \in \mathbb{R}^{H\times W\times C}$ are generated from the image $\mathbf{X}_S$ using a 1×1 convolution followed by the sigmoid activation. These masks are then used to re-calibrate the transformed local features $\mathbf{F_{in}}$ (obtained after 1×1 convolution), resulting in attention-guided features which are added to the identity mapping path. Finally, the attention-augmented feature representation $\mathbf{F_{out}}$, produced by SAM, is passed to the next stage for further processing.

4. Experiments and Analysis

We evaluate our method for several image restoration tasks, including **(a)** image deraining, **(b)** image deblurring, and **(c)** image denoising on 10 different datasets.

4.1. Datasets and Evaluation Protocol

Quantitative comparisons are performed using the PSNR and SSIM [76] metrics. As in [7], we report (in parenthesis) the reduction in error for each method relative to the best performing method by translating PSNR to RMSE (RMSE $\propto \sqrt{10^{-\mathrm{PSNR}/10}}$) and SSIM to DSSIM (DSSIM $= (1-\mathrm{SSIM})/2$). The datasets used for training and testing are summarized in Table 1 and described next.

Image Deraining. Using the same experimental setups of the recent best method on image deraining [37], we train our model on 13,712 clean-rain image pairs gathered from multiple datasets [23, 48, 81, 89, 90], as shown in Table 1. With this single trained model, we perform evaluation on various test sets, including Rain100H [81], Rain100L [81], Test100 [90], Test2800 [23], and Test1200 [89].

Image Deblurring. As in [70, 88, 43, 71], we use the GoPro [53] dataset that contains 2,103 image pairs for training and 1,111 pairs for evaluation. Furthermore, to demonstrate generalizability, we take our GoPro trained model and *directly apply* it on the test images of the HIDE [69] and RealBlur [64] datasets. The HIDE dataset is specifically collected for human-aware motion deblurring and its test set contains 2,025 images. While the GoPro and HIDE datasets are synthetically generated, the image pairs of RealBlur dataset are captured in real-world conditions. The RealBlur dataset has two subsets: (1) RealBlur-J is formed with the camera JPEG outputs, and (2) RealBlur-R is generated offline by applying white balance, demosaicking, and denoising operations to the RAW images.

Image Denoising. To train our model for image denoising task, we use 320 high-resolution images of the SIDD dataset [1]. Evaluation is conducted on 1,280 validation patches from the SIDD dataset [1] and 1,000 patches from the DND benchmark dataset [60]. These test patches are extracted from the full resolution images by the original au-

Table 1: Dataset description for various image restoration tasks.

Tasks	Deraining							Deblurring			Denoising	
Datasets	Rain14000 [23]	Rain1800 [81]	Rain800 [90]	Rain100H [81]	Rain100L [81]	Rain1200 [89]	Rain12 [48]	GoPro [53]	HIDE [69]	RealBlur [64]	SIDD [1]	DND [60]
Train Samples	11200	1800	700	0	0	0	12	2103	0	0	320	0
Test Samples	2800	0	100	100	100	1200	0	1111	2025	1960	40	50
Testset Rename	Test2800	-	Test100	Rain100H	Rain100L	Test1200	-	-	-	-	-	-

Table 2: Image deraining results. Best and second best scores are **highlighted** and underlined. For each method, reduction in error relative to the best-performing algorithm is reported in parenthesis (see Section 4.1 for error calculation technique). Our MPRNet achieves ∼20% relative improvement in PSNR over the previous best method MSPFN [37].

Methods	Test100 [90]		Rain100H [81]		Rain100L [81]		Test2800 [23]		Test1200 [89]		Average	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
DerainNet [22]	22.77	0.810	14.92	0.592	27.03	0.884	24.31	0.861	23.38	0.835	22.48 (69.3%)	0.796 (61.3%)
SEMI [77]	22.35	0.788	16.56	0.486	25.03	0.842	24.43	0.782	26.05	0.822	22.88 (67.8%)	0.744 (69.1%)
DIDMDN [89]	22.56	0.818	17.35	0.524	25.23	0.741	28.13	0.867	29.65	0.901	24.58 (60.9%)	0.770 (65.7%)
UMRL [82]	24.41	0.829	26.01	0.832	29.18	0.923	29.97	0.905	30.55	0.910	28.02 (41.9%)	0.880 (34.2%)
RESCAN [47]	25.00	0.835	26.36	0.786	29.80	0.881	31.29	0.904	30.51	0.882	28.59 (37.9%)	0.857 (44.8%)
PreNet [63]	24.81	0.851	26.77	0.858	32.44	0.950	31.75	0.916	31.36	0.911	29.42 (31.7%)	0.897 (23.3%)
MSPFN [37]	27.50	0.876	28.66	0.860	32.40	0.933	32.82	0.930	32.39	0.916	30.75 (20.4%)	0.903 (18.6%)
MPRNet (Ours)	**30.27**	**0.897**	**30.41**	**0.890**	**36.40**	**0.965**	**33.64**	**0.938**	**32.91**	**0.916**	**32.73** (0.0%)	**0.921** (0.0%)

thors. Both SIDD and DND datasets consist of real images.

4.2. Implementation Details

Our MPRNet is end-to-end trainable and requires no pre-training. We train separate models for three different tasks. We employ 2 CABs at each scale of the encoder-decoder, and for downsampling we use 2×2 max-pooling with stride 2. In the last stage, we employ ORSNet that contains 3 ORBs, each of which further uses 8 CABs. Depending on the task complexity, we scale the network width by setting the number of channels to 40 for deraining, 80 for denoising, and 96 for deblurring. The networks are trained on 256×256 patches with a batch size of 16 for 4×10^5 iterations. For data augmentation, horizontal and vertical flips are randomly applied. We use Adam optimizer [41] with the initial learning rate of 2×10^{-4}, which is steadily decreased to 1×10^{-6} using the cosine annealing strategy [50].

4.3. Image Deraining Results

For the image deraining task, consistent with prior work [37], we compute image quality scores using the Y channel (in YCbCr color space). Table 2 shows that our method significantly advances state-of-the-art by consistently achieving better PSNR/SSIM scores on all five datasets. Compared to the recent best algorithm MSPFN [37], we obtain a performance gain of 1.98 dB (average across all datasets), indicating 20% error reduction. The improvements on some datasets are as large as 4 dB, e.g., Rain100L [81]. Further, our model has 3.7× fewer parameters than MSPFN [37], while being 2.4× faster.

Figure 5 shows visual comparisons on challenging images. Our MPRNet is effective in removing rain streaks of different orientations and magnitudes, and generates images that are visually pleasant and faithful to the ground-truth. In contrast, other approaches compromise structural content (first row), introduce artifacts (second row), and do not completely remove rain streaks (third row).

4.4. Image Deblurring Results

We report the performance of evaluated image deblurring approaches on the synthetic GoPro [53] and HIDE [69] datasets in Table 3. Overall, our model performs favorably against other algorithms. Compared to the previous best performing technique [70], our method achieves 9% improvement in PSNR and 21% in SSIM on the GoPro [53] dataset, and a 11% and 13% reduction in error on the HIDE dataset [69]. It is worth noticing that our network is trained only on the GoPro dataset, but achieves the state-of-the-art results (+0.98 dB) on the HIDE dataset, thereby demonstrating its strong generalization capability.

We evaluate our MPRNet on the real-world images of a recent RealBlur [64] dataset under two experimental settings: 1). apply the GoPro trained model directly on RealBlur (to test generalization to real images), and 2). train and test on RealBlur data. Table 4 shows the experimental results. For setting 1, our MPRNet obtains performance gains of 0.29 dB on the RealBlur-R subset and 0.28 dB on the RealBlur-J subset over the DMPHN algorithm [88]. A similar trend is observed for setting 2, where our gains over SRN [71] are 0.66 dB and 0.38 dB on RealBlur-R and RealBlur-J, respectively.

Figure 6 shows some deblurred images by the evaluated approaches. Overall, the images restored by our model are sharper and closer to the ground-truth than those by others.

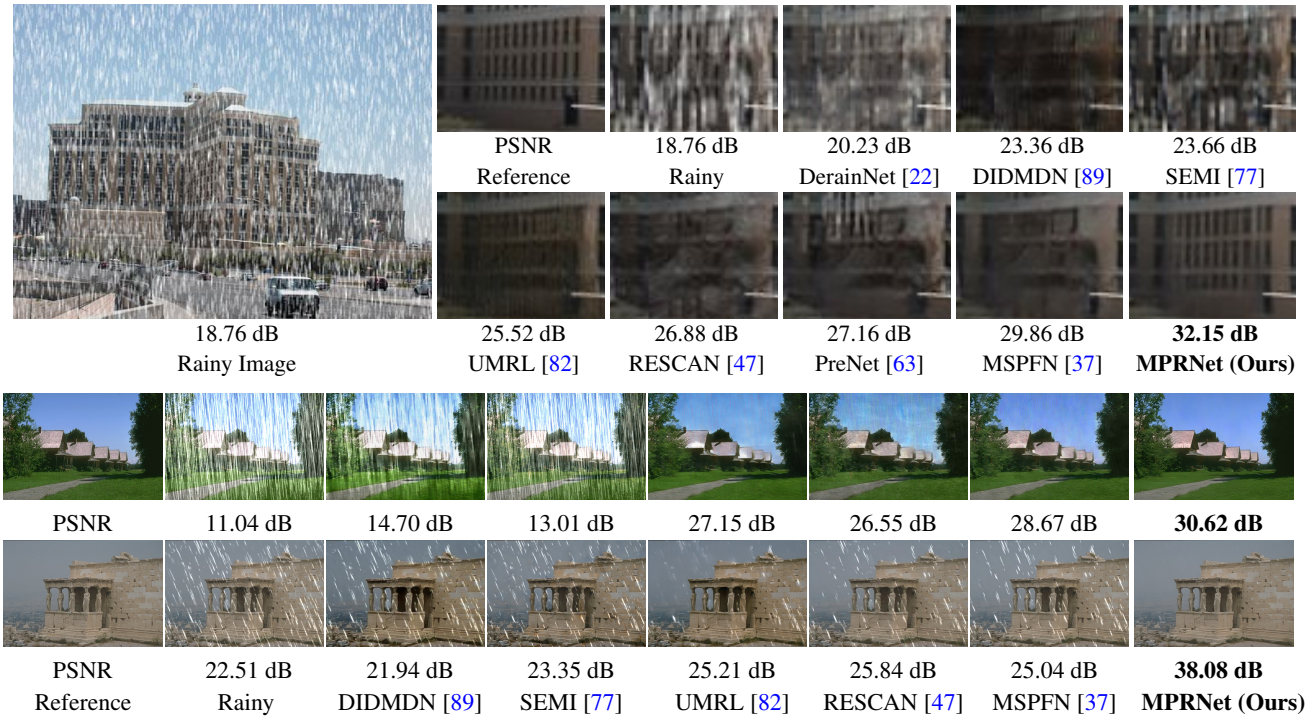


Figure 5: Image deraining results. Our MPRNet effectively removes rain and generates images that are natural, artifact-free and visually closer to the ground-truth.

Table 3: Deblurring results. Our method is trained only on the GoPro dataset [53] and directly applied to the HIDE dataset [69].

Method	GoPro [53] PSNR ↑		GoPro [53] SSIM ↑		HIDE [69] PSNR ↑		HIDE [69] SSIM ↑	
Xu *et al.* [80]	21.00	(73.9%)	0.741	(84.2%)	-		-	
Hyun *et al.* [36]	23.64	(64.6%)	0.824	(76.7%)	-		-	
Whyte *et al.* [78]	24.60	(60.5%)	0.846	(73.4%)	-		-	
Gong *et al.* [27]	26.40	(51.4%)	0.863	(70.1%)	-		-	
DeblurGAN [42]	28.70	(36.6%)	0.858	(71.1%)	24.51	(52.4%)	0.871	(52.7%)
Nah *et al.* [53]	29.08	(33.8%)	0.914	(52.3%)	25.73	(45.2%)	0.874	(51.6%)
Zhang *et al.* [91]	29.19	(32.9%)	0.931	(40.6%)	-		-	
DeblurGAN-v2 [43]	29.55	(30.1%)	0.934	(37.9%)	26.61	(39.4%)	0.875	(51.2%)
SRN [71]	30.26	(24.1%)	0.934	(37.9%)	28.36	(25.9%)	0.915	(28.2%)
Shen *et al.* [69]	-		-		28.89	(21.2%)	0.930	(12.9%)
Gao *et al.* [25]	30.90	(18.3%)	0.935	(36.9%)	29.11	(19.2%)	0.913	(29.9%)
DBGAN [92]	31.10	(16.4%)	0.942	(29.3%)	28.94	(20.8%)	0.915	(28.2%)
MT-RNN [58]	31.15	(16.0%)	0.945	(25.5%)	29.15	(18.8%)	0.918	(25.6%)
DMPHN [88]	31.20	(15.5%)	0.940	(31.7%)	29.09	(19.4%)	0.924	(19.7%)
Suin *et al.* [70]	31.85	(8.9%)	0.948	(21.2%)	29.98	(10.7%)	0.930	(12.9%)
MPRNet (Ours)	**32.66**	(0.0%)	**0.959**	(0.0%)	**30.96**	(0.0%)	**0.939**	(0.0%)

Table 4: Deblurring comparisons on the RealBlur dataset [64] under two different settings: 1). applying our GoPro trained model directly on the RealBlur set (to evaluate generalization to real images), 2). Training and testing on RealBlur data where methods are denoted with symbol ‡. The PSNR/SSIM scores for other evaluated approaches are taken from the RealBlur benchmark [64].

Method	RealBlur-R PSNR ↑		RealBlur-R SSIM ↑		RealBlur-J PSNR ↑		RealBlur-J SSIM ↑	
Hu *et al.* [33]	33.67	(23.4%)	0.916	(42.9%)	26.41	(23.2%)	0.803	(35.5%)
Nah *et al.* [53]	32.51	(33.0%)	0.841	(69.8%)	27.87	(9.1%)	0.827	(26.6%)
DeblurGAN [42]	33.79	(22.4%)	0.903	(50.5%)	27.97	(8.1%)	0.834	(23.5%)
Pan *et al.* [56]	34.01	(20.4%)	0.916	(42.9%)	27.22	(15.7%)	0.790	(39.5%)
Xu *et al.* [80]	34.46	(16.2%)	0.937	(23.8%)	27.14	(16.4%)	0.830	(25.3%)
DeblurGAN-v2 [43]	35.26	(8.1%)	0.944	(14.3%)	28.70	(0.0%)	0.866	(5.2%)
Zhang *et al.* [91]	35.48	(5.7%)	0.947	(9.4%)	27.80	(9.8%)	0.847	(17.0%)
SRN [71]	35.66	(3.7%)	0.947	(9.4%)	28.56	(1.6%)	0.867	(4.5%)
DMPHN [88]	35.70	(3.3%)	0.948	(7.7%)	28.42	(3.2%)	0.860	(9.3%)
MPRNet (Ours)	**35.99**	(0.0%)	**0.952**	(0.0%)	**28.70**	(0.0%)	**0.873**	(0.0%)
‡DeblurGAN-v2 [43]	36.44	(28.1%)	0.935	(56.9%)	29.69	(21.2%)	0.870	(40.0%)
‡SRN [71]	38.65	(7.3%)	0.965	(20.0%)	31.38	(4.3%)	0.909	(14.3%)
‡**MPRNet (Ours)**	**39.31**	(0.0%)	**0.972**	(0.0%)	**31.76**	(0.0%)	**0.922**	(0.0%)

4.5. Image Denoising Results

In Table 5, we report PSNR/SSIM scores of several image denoising methods on the SIDD [1] and DND [60] datasets. Our method obtains considerable gains over the state-of-the-art approaches, *i.e.*, 0.19 dB over CycleISP [86] on SIDD and 0.21 dB over SADNet [11] on DND. Note that the DND dataset does not contain any training images, *i.e.*, the complete publicly released dataset is just a test set. Experimental results on the DND benchmark with our SIDD trained model demonstrates our model generalizes well to different image domains.

Fig. 7 illustrates visual results. Our method is able to remove real noise, while preserving the structural and textural image details. In contrast, the images restored by other methods contain either overly smooth contents, or artifacts with splotchy textures.

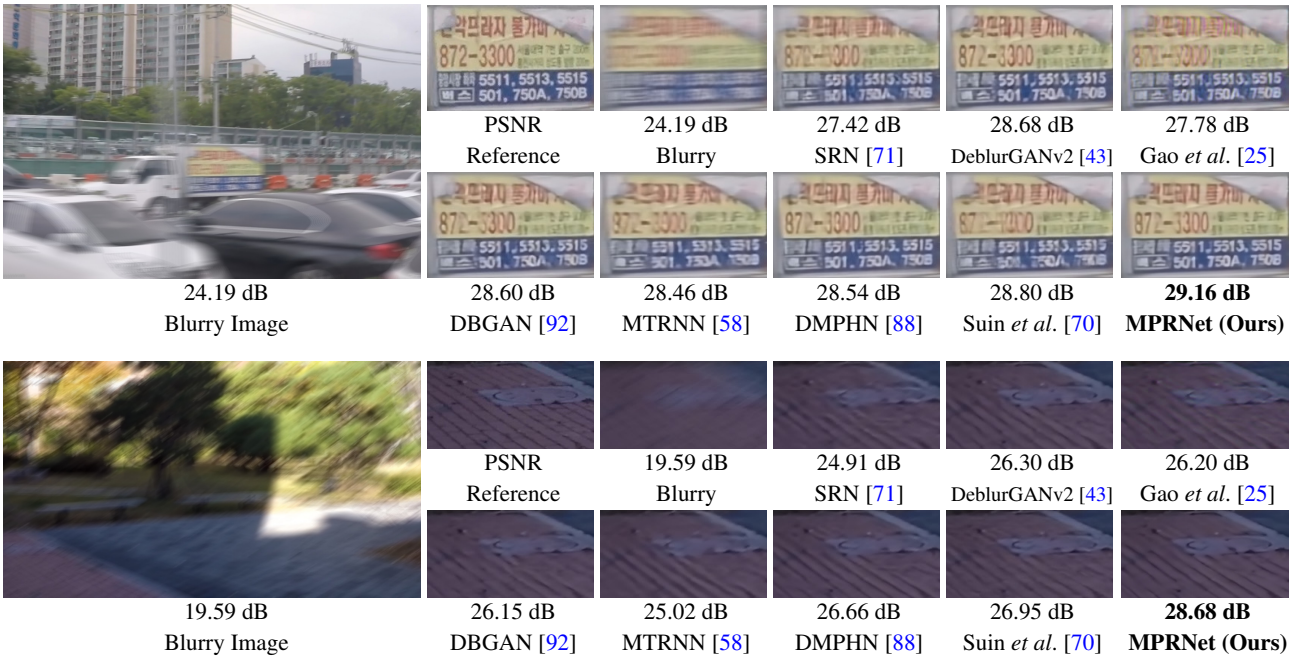


Figure 6: Visual comparisons for image deblurring on the GoPro datatset [53]. Compared to the state-of-the-art methods, our MPRNet restores more sharper and perceptually-faithful images.

Table 5: Denoising comparisons on SIDD [1] and DND [60] datasets. $*$ denotes the methods that use additional training data. Whereas our MPRNet is only trained on the SIDD images and directly tested on DND.

Method	SIDD [1] PSNR ↑		SIDD [1] SSIM ↑		DND [60] PSNR ↑		DND [60] SSIM ↑	
DnCNN [93]	23.66	(84.2%)	0.583	(89.9%)	32.43	(57.2%)	0.790	(79.1%)
MLP [9]	24.71	(82.2%)	0.641	(88.3%)	34.23	(47.3%)	0.833	(73.7%)
BM3D [16]	25.65	(80.2%)	0.685	(86.7%)	34.51	(45.6%)	0.851	(70.5%)
CBDNet* [28]	30.78	(64.2%)	0.801	(78.9%)	38.06	(18.2%)	0.942	(24.1%)
RIDNet* [4]	38.71	(10.9%)	0.951	(14.3%)	39.26	(6.0%)	0.953	(6.4%)
AINDNet* [40]	38.95	(8.4%)	0.952	(12.5%)	39.37	(4.8%)	0.951	(10.2%)
VDN [84]	39.28	(4.8%)	0.956	(4.6%)	39.38	(4.7%)	0.952	(8.3%)
SADNet* [11]	39.46	(2.8%)	0.957	(2.3%)	39.59	(2.4%)	0.952	(8.3%)
DANet+* [85]	39.47	(2.7%)	0.957	(2.3%)	39.58	(2.5%)	0.955	(2.2%)
CycleISP* [86]	39.52	(2.2%)	0.957	(2.3%)	39.56	(2.7%)	**0.956**	(0.0%)
MPRNet (Ours)	**39.71**	(0.0%)	**0.958**	(0.0%)	**39.80**	(0.0%)	0.954	(4.4%)

Table 6: Ablation study on individual components of the proposed MPRNet.

#Stages	Stage Combination	SAM	CSFF	PSNR
1	U-Net (baseline)	-	-	28.94
1	ORSNet (baseline)	-	-	28.91
2	U-Net + U-Net	✗	✗	29.40
2	ORSNet + ORSNet	✗	✗	29.53
2	U-Net + ORSNet	✗	✗	29.70
3	U-Nets + ORSNet	✗	✗	29.86
3	U-Nets + ORSNet	✗	✓	30.07
3	U-Nets + ORSNet	✓	✗	30.31
3	U-Nets + ORSNet	✓	✓	**30.49**

4.6. Ablation Studies

Here we present ablation experiments to analyze the contribution of each component of our model. Evaluation is performed on the GoPro dataset [53] with the deblurring models trained on image patches of size 128×128 for 10^5 iterations, and the results are shown in Table 6.

Number of stages. Our model yields better performance as the number of stages increases, which validates the effectiveness of our multi-stage design.

Choices of subnetworks. Since each stage of our model could employ different subnetwork design, we test different options. We show that using the encoder-decoder in the earlier stage(s) and the ORSNet in the last stage leads to improved performance (29.7 dB) as compared to employing the same design for all the stages (29.4 dB with U-Net+U-Net, and 29.53 dB with ORSNet+ORSNet).

SAM and CSFF. We demonstrate the effectiveness of the proposed supervised attention module and cross-stage feature fusion mechanism by removing them from our final model. Table 6 shows a substantial drop in PSNR from 30.49 dB to 30.07 dB when SAM is removed, and from 30.49 dB to 30.31 dB when we take out CSFF. Removing both of these components degrades the performance by a large margin from 30.49 dB to 29.86 dB.

5. Resource Efficient Image Restoration

CNN models generally exhibit a trade-off between accuracy and computational efficiency. In the pursuit of achieving higher accuracy, deeper and complex models are often developed. Although large models tend to perform better than their smaller counterparts, the computational cost can

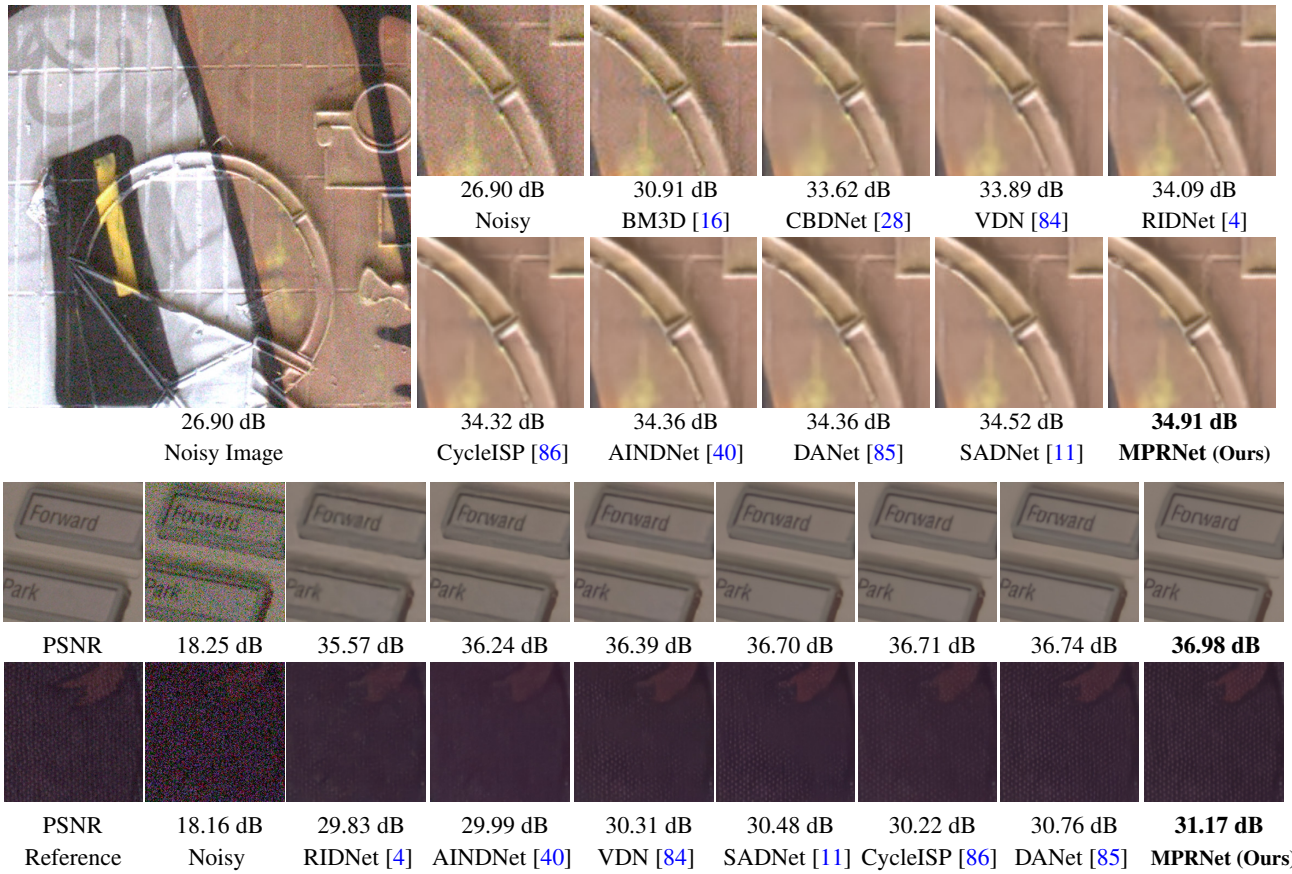


Figure 7: Image denoising comparisons. First example is from DND [60] and the others from SIDD [1]. The proposed MPRNet better preserves fine texture and structural patterns in the denoised images.

Table 7: Stage-wise deblurring performance of MPRNet on GoPro [53]. Runtimes are computed with the Nvidia Titan Xp GPU.

Method	DeblurGAN-v2 [43]	SRN [71]	DMPHN [88]	Suin *et al.* [70]	**MPRNet (ours)** 1-stage	2-stages	3-stages
PSNR	29.55	30.10	31.20	31.85	30.43	31.81	32.66
#Params (M)	60.9	6.8	21.7	23.0	5.6	11.3	20.1
Time (s)	0.21	0.57	1.07	0.34	0.04	0.08	0.18

be prohibitively high. As such, it is of great interest to develop resource-efficient image restoration models. One solution is to train the same network by adjusting its capacity every time the target system is changed. However, it is tedious and oftentimes infeasible. A more desirable approach is to have a single network that can make **(a)** early predictions for compute efficient systems and **(b)** latter predictions to obtain high accuracy. A *multi-stage* restoration model naturally offers such functionalities.

Table 7 reports the stage-wise results of our multi-stage approach. Our MPRNet demonstrates competitive restoration performance at each stage. Notably, our stage-1 model is light, fast, and yields better results than other sophisticated algorithms such as SRN [71] and DeblurGAN-v2 [43]. Similarly, when compared to a recent method DMPHN [88], our stage-2 model shows the PSNR gain of 0.51 dB while being more resource-efficient ($\sim$2$\times$ fewer parameters and 13$\times$ faster).

6. Conclusion

In this work, we propose a multi-stage architecture for image restoration that progressively improves degraded inputs by injecting supervision at each stage. We develop guiding principles for our design that demand complementary feature processing in multiple stages and a flexible information exchange between them. To this end, we propose contextually-enriched and spatially accurate stages that encode a diverse set of features in unison. To ensure synergy between reciprocal stages, we propose feature fusion across stages and an attention guided output exchange from earlier stages to the later ones. Our model achieves significant performance gains on numerous benchmark datasets. In addition, our model is light-weighted in terms of model size and efficient in terms of runtime, which are of great interest for devices with limited resources.

Acknowledgments. M.-H. Yang is supported in part by the NSF CAREER Grant 1149783. Special thanks to Kui Jiang for providing image deraining results.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 4, 5, 6, 7, 8

[2] Abdelrahman Abdelhamed, Radu Timofte, and Michael S Brown. NTIRE 2019 challenge on real image denoising: Methods and results. In *CVPRW*, 2019. 2

[3] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *Trans. Sig. Proc.*, 2006. 2

[4] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. *ICCV*, 2019. 1, 2, 7, 8

[5] Saeed Anwar and Nick Barnes. Densely residual laplacian super-resolution. *TPAMI*, 2020. 2

[6] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *ACM Computing Surveys*, 2019. 3

[7] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *CVPR*, 2019. 1, 3, 4

[8] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 2

[9] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *CVPR*, 2012. 7

[10] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *TIP*, 1998. 2

[11] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising. In *ECCV*, 2020. 6, 7, 8

[12] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994. 3

[13] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *CVPR*, 2018. 1, 3

[14] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018. 1

[15] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas S Huang, Wen-Mei Hwu, and Honghui Shi. SPGNet: Semantic prediction guidance for scene parsing. In *ICCV*, 2019. 1

[16] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *TIP*, 2007. 2, 7, 8

[17] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019. 1, 2

[18] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2015. 3

[19] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *TIP*, 2011. 1

[20] Yazan Abu Farha and Jurgen Gall. MS-TCN: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, 2019. 1

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

[22] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *TIP*, 2017. 5, 6

[23] 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*, 2017. 4, 5

[24] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. Lightweight pyramid networks for image deraining. *TNNLS*, 2019. 2

[25] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *CVPR*, 2019. 6, 7

[26] Pallabi Ghosh, Yi Yao, Larry Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. In *WACV*, 2020. 1

[27] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton Van Den Hengel, and Qinfeng Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *CVPR*, 2017. 6

[28] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019. 2, 7, 8

[29] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *TPAMI*, 2010. 1

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[31] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *NeurIPS*, 2018. 2

[32] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE TPAMI*, 2019. 2

[33] Zhe Hu, Sunghyun Cho, Jue Wang, and Ming-Hsuan Yang. Deblurring low-light images with light streaks. In *CVPR*, 2014. 6

[34] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[35] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. CCNet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019. 2

[36] Tae Hyun Kim, Byeongjoo Ahn, and Kyoung Mu Lee. Dynamic scene deblurring. In *ICCV*, 2013. 6

[37] Kui Jiang, Zhongyuan Wang, Peng Yi, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *CVPR*, 2020. 2, 3, 4, 5, 6

[38] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak

Shah. Transformers in vision: A survey. *arXiv:2101.01169*, 2021. 2

[39] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *TPAMI*, 2010. 1

[40] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *CVPR*, 2020. 7, 8

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 5

[42] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. DeblurGAN: Blind motion deblurring using conditional adversarial networks. In *CVPR*, 2018. 2, 6

[43] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, 2019. 1, 2, 3, 4, 6, 7, 8

[44] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 1

[45] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. MS-TCN++: Multi-stage temporal convolutional network for action segmentation. *TPAMI*, 2020. 1

[46] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv:1901.00148*, 2019. 1

[47] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, 2018. 1, 2, 5, 6

[48] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *CVPR*, 2016. 4, 5

[49] Wei Liu, Andrew Rabinovich, and Alexander C Berg. ParseNet: Looking wider to see better. *arXiv:1506.04579*, 2015. 4

[50] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 5

[51] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, 2015. 2

[52] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *TIP*, 2007. 2

[53] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 1, 2, 4, 5, 6, 7, 8

[54] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 1

[55] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 3

[56] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *CVPR*, 2016. 6

[57] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *ECCV*, 2020. 1, 2

[58] Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *ECCV*, 2020. 6, 7

[59] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *TPAMI*, 1990. 1

[60] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 4, 5, 6, 7, 8

[61] Kuldeep Purohit and AN Rajagopalan. Region-adaptive dense network for efficient motion deblurring. In *AAAI*, 2020. 2

[62] 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*, 2018. 1

[63] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, 2019. 1, 2, 5, 6

[64] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, 2020. 4, 5, 6

[65] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1, 2, 3

[66] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005. 1

[67] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 1992. 1, 2

[68] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *ToG*, 2008. 2

[69] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, 2019. 4, 5, 6

[70] Maitreya Suin, Kuldeep Purohit, and A. N. Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[71] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018. 1, 2, 4, 5, 6, 7, 8

[72] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 2020. 2

[73] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *ICCV*, 2017. 1

[74] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 2

[75] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN:

enhanced super-resolution generative adversarial networks. In *ECCVW*, 2018. 1

[76] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 4

[77] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *CVPR*, 2019. 5, 6

[78] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce. Non-uniform deblurring for shaken images. *IJCV*, 2012. 6

[79] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 2

[80] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, 2013. 2, 6

[81] 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*, 2017. 1, 4, 5

[82] Rajeev Yasarla and Vishal M Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *CVPR*, 2019. 5, 6

[83] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[84] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. In *NeurIPS*, 2019. 7, 8

[85] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, 2020. 7, 8

[86] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. CycleISP: Real image restoration via improved data synthesis. In *CVPR*, 2020. 1, 2, 6, 7, 8

[87] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *ECCV*, 2020. 1

[88] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[89] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, 2018. 2, 4, 5, 6

[90] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *TCSVT*, 2019. 1, 4, 5

[91] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson WH Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *CVPR*, 2018. 1, 6

[92] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *CVPR*, 2020. 1, 6, 7

[93] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 1, 2, 3, 7

[94] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *CVPR*, 2017. 1

[95] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 1, 2, 3

[96] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *ICLR*, 2019. 1

[97] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *TPAMI*, 2020. 1, 2, 3

[98] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018. 2

[99] Yupei Zheng, Xin Yu, Miaomiao Liu, and Shunli Zhang. Residual multiscale based single image deraining. In *BMVC*, 2019. 2

[100] Song Chun Zhu and David Mumford. Prior learning and gibbs reaction-diffusion. *TPAMI*, 1997. 1