Supplementary material for "Dual Residual Networks Leveraging the Potential of Paired Operations for Image Restoration"

Xing Liu[†] Masanori Suganuma[†‡] Zhun Sun[‡] Takayuki Okatani[†‡]
[†]Graduate School of Information Sciences, Tohoku University [‡]RIKEN Center for AIP
{ryu,suganuma,zhun,okatani}@vision.is.tohoku.ac.jp

This document provides additional explanations about the experimental setting for each of the five image restoration tasks.

A. Implementation Details and Additional Results for the Five Tasks

A.1. Details of Training Method

We use the Adam optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 1.0 \times 10^{-8}$ for training all the proposed DuRNs. For loss functions, we use a weighted sum of SSIM and l_1 loss, specifically, $1.1 \times \text{SSIM} + 0.75 \times l_1$, for all the tasks. There are two exceptions. One is Gaussian noise removal on the BSD500-grayscale dataset [2], where we use l_2 loss. The other is raindrop removal, where we use the same weighted loss for the first 4,000 epochs, and then switch it to a single l_1 loss for additional 100 epochs. The initial learning rate is set to 0.0001 for all the tasks. All the experiments are conducted using PyTorch [3]. Our code and trained models will be made publicly available at *https://github.com/liu-vis/DualResidualNetworks*

A.2. Noise Removal

Specification of ct_1^l and ct_2^l We show the specification of ct_1^l and ct_2^l for each DuRB-P in Table 1, in which $l(= 1, \ldots, 6)$ denotes the block-id of a DuRB; the "recep." denotes the receptive field of convolution. It is observed that the paired convolution has a large- and small- receptive field for each DuRB-P (see each row in the table), and the size of the receptive fields of ct_1^l and ct_2^l increases with l with an exception at $l = 5$, which is to avoid too large a receptive field. By this design we intend to make each block look at the input image at an increasing scale with layers in the forward direction.

Table 1: The specification of ct_1^l and ct_2^l for DuRB-P's for noise removal. The "recep." denotes the receptive field of convolution, i.e., delation rate $\times$ (kernel size - 1) $+ 1$.

layer	kernel	dilation	recep.	layer	kernel	dilation	recep.
$ct_1^{l=1}$	5	1	5×5	$ct_2^{l=1}$	3	1	3×3
$ct_1^{l=2}$	7	1	7×7	$ct_2^{l=2}$	5	1	5×5
$ct_1^{l=3}$	7	2	13×13	$ct_2^{l=3}$	5	1	5×5
$ct_1^{l=4}$	11	2	21×21	$ct_2^{l=4}$	7	1	7×7
$ct_1^{l=5}$	11	1	11×11	$ct_2^{l=5}$	5	1	5×5
$ct_1^{l=6}$	11	3	31×31	$ct_2^{l=6}$	7	1	7×7

Experimental Setting for Gaussian Noise Removal In training, we set batch size $= 100$. Each input image in a batch is obtained by randomly cropping a 64×64 region from an original training noisy image. We exactly followed the procedure of [5] to generate noisy images for training our network.

Experimental Setting for Real-World Noise Removal In training, we randomly select 30 out of 40 pairs of a high resolution noisy image and a mean image (used as ground truth) for constructing the training dataset. We set input patch size $= 128 \times 128$, and use 30 patches (each of which is randomly cropped from a different training image) to create one batch. To test the CNNs including ours and the baselines, we use the remaining 10 image pairs; specifically, we randomly crop ten 512×512 patches from each of them, yielding 100 patches that are used for the test.

A.3. Motion Blur Removal

Table 2: The specification of ct_1^l for DuRB-U's for motion blur removal.

layer	kernel	dilation	recep.	layer	kernel	dilation	recep.
$ct_1^{l=1}$	3	3	7	$ct_1^{l=4}$	7	1	7
$ct_1^{l=2}$	7	1	7	$ct_1^{l=5}$	3	2	5
$ct_1^{l=3}$	3	3	7	$ct_1^{l=6}$	5	1	5

Specification of ct_1^l and ct_2^l The specification of ct_1^l is shown in Table 2. For ct_2^l, we use an identical configuration, kernel size $= 3 \times 3$, dilation rate $= 1$ and stride $= 2$, for all DuRB-U's. We intend to simply perform down-sampling with ct_2^l.

Experimental Setting on GoPro Dataset In training, we set batch size $= 10$. Each input image in a batch is obtained by randomly cropping a 256×256 patch from the re-sized

Table 3: The specification of ct_1^l for DuRB-US's for haze removal.

layer	kernel	dilation	recep.	layer	kernel	dilation	recep.
$ct_1^{l=1}$	5	1	5	$ct_1^{l=7}$	11	1	11
$ct_1^{l=2}$	5	1	5	$ct_1^{l=8}$	11	1	11
$ct_1^{l=3}$	7	1	7	$ct_1^{l=9}$	11	1	11
$ct_1^{l=4}$	7	1	7	$ct_1^{l=10}$	11	1	11
$ct_1^{l=5}$	11	1	11	$ct_1^{l=11}$	11	1	11
$ct_1^{l=6}$	11	1	11	$ct_1^{l=12}$	11	1	11

version (640×360) of an original training image of size 1280×720. In testing, we use the re-sized version (640×360) of the original test images of size 1280×720 as in training.

Experimental Setting on Car Dataset The Car dataset was used only for evaluation. We down-scale the blur images from their original size 720×720 to 360×360 and input them to the DuRN-U trained using GoPro-train dataset for de-blurring. The result is then up-scaled to 700×700 and fed into YOLOv3.

Additional Examples More examples of motion blur removal on GoPro-test dataset are shown in Fig. 1.

A.4. Haze Removal

Specification of ct_1^l and ct_2^l The specification of ct_1^l is shown in Table 3. For ct_2^l, we use an identical configuration, i.e., kernel size $= 3 \times 3$, dilation rate $= 1$ and stride $= 2$, for all the DuRB-US's. We intend to simply perform down-sampling with ct_2^l.

Experimental Setting on Dehaze Dataset In training, we set batch size $= 20$. Each input image in a batch is obtained by randomly cropping a 256×256 region from an original training image of size 512×512.

Experimental Setting on RESIDE In training, we set batch size $= 48$. Each input image in a batch is obtained by randomly cropping a 256×256 region from an original image of size 620×460.

Visualization of Internal Layer Activation Figure 5 shows activation maps of several chosen blocks (i.e., DuRB-US's) in the network for different input images. They are the sums in the channel dimension of activation maps of the input to the first DuRB-US ($l = 0$), and of the output from the third ($l = 3$), sixth ($l = 6$), and twelfth ($l = 12$) DuRB-US's. It is seen that the DuRN-US computes a map that looks similar to transmission map at around $l = 3$.

Additional Examples More examples of haze removal are shown on Figs. 2, 3 and 4. In Fig. 4, we show the results for images of hazy scenes that are captured using iPhone-6 plus by us.

Table 4: The specification of ct_1^l for DuRB-S's and DuRB-P's of the DuRN-S-P for raindrop removal.

DuRB-S			
layer	kernel	dilation	recep.
$ct_1^{l=1}$	3	12	25
$ct_1^{l=2}$	3	8	17
$ct_1^{l=3}$	3	6	13

DuRB-P			
layer	kernel	dilation	recep.
$ct_1^{l=1}$	3	2	5
$ct_1^{l=2}$	5	1	5
$ct_1^{l=3}$	3	3	7
$ct_1^{l=3}$	7	1	7
$ct_1^{l=3}$	3	4	9
$ct_1^{l=3}$	7	1	7

A.5. Raindrop Removal

Specification of ct_1^l and ct_2^l The specification of ct_1^l for the three DuRB-S's and the six DuRB-P's is shown in Table 4. For ct_2^l, we use an identical configuration, kernel size $= 3 \times 3$ and dilation rate $= 1$, for all the DuRB-S's, and use an identical configuration, kernel size $= 5 \times 5$ and dilation rate $= 1$, for all the DuRB-P's.

Experimental Setting on RainDrop Dataset In training, we set batch size $= 24$. Each input image in a batch is obtained by randomly cropping a 256×256 region from the original image of size 720×480. As mentioned before, we train the network $1.1 \times$ SSIM $+ 0.75 \times l_1$ using the loss for 4,000 epochs, and then switch the loss to l_1 alone, training the network for additional 100 epochs. We did this for faster converging.

Additional Examples More examples of raindrop removal are shown in Fig. 6.

A.6. Rain-streak Removal

Specification of ct_1^l and ct_2^l We use the same configuration as noise removal. See Table. 1. Note that we use DuRB-S for this task.

Experimental Setting on DDN Data To train the DuRN-S, we set batch size $= 40$. Each input image in a batch is obtained by randomly cropping a 64×64 region from an original training image.

Experimental Setting on DID-MDN Data In training, we set batch size $= 80$. Each input image in a batch is obtained by randomly cropping a 64×64 region from an original training image.

Additional Examples More examples of rain-streak removal on synthetic rainy images and on real-world rainy images are shown in Fig. 7 and Fig. 8, respectively.

A.7. Performance of DuRBs on Non-target Tasks

We have presented the four versions of DuRB, each of which is designed for a single task. To verify the effectiveness of the design choices, we examine the performance of each DuRB on its non-target tasks. Specifically, we evaluate

Table 5: Performance (PSNR/SSIM) of the four versions of DuRBs (i.e., -P, -U, -US, and -S) on different task.

	Real-noise	Motion blur	Haze	Raindrop	Rain-streak
DuRB-P	**36.83** / **0.9635**	29.40 / 0.8996	29.33 / 0.9761	24.69 / 0.8067	32.88 / 0.9214
DuRB-U	36.63 / 0.9600	29.90 / 0.9100	30.79 / 0.9800	24.30 / 0.8067	33.00 / **0.9265**
DuRB-US	36.61 / 0.9591	**29.96** / **0.9101**	**32.60** / **0.9827**	22.72 / 0.7254	32.84 / 0.9238
DuRB-S	36.82 / 0.9629	29.55 / 0.9023	31.81 / 0.9792	**25.13** / **0.8134**	**33.21** / 0.9251

the performance of every combination of the four versions of DuRB and the five tasks. For noise, motion blur, haze, raindrop and rain-streak removal, we train and test networks consisting of each version of DuRB on Real-World Noisy Image Dataset, GoPro Dataset, Dehaze Dataset, RainDrop Dataset and DID-MDN Data. The results are shown in Table 5. It is seen that in general, each DuRB yields the best performance for the task to which it was designed. For motion blur removal, DuRB-US performs comparably well or even slightly better than DuRB-U, which is our primary design for the task. We think this is reasonable, as DuRB-US contains the same paired operation as DuRB-U (i.e., up- and down-sampling), contributing to the good performance. Their performance gap is almost negligible and thus DuRB-U is a better choice, considering its efficiency.

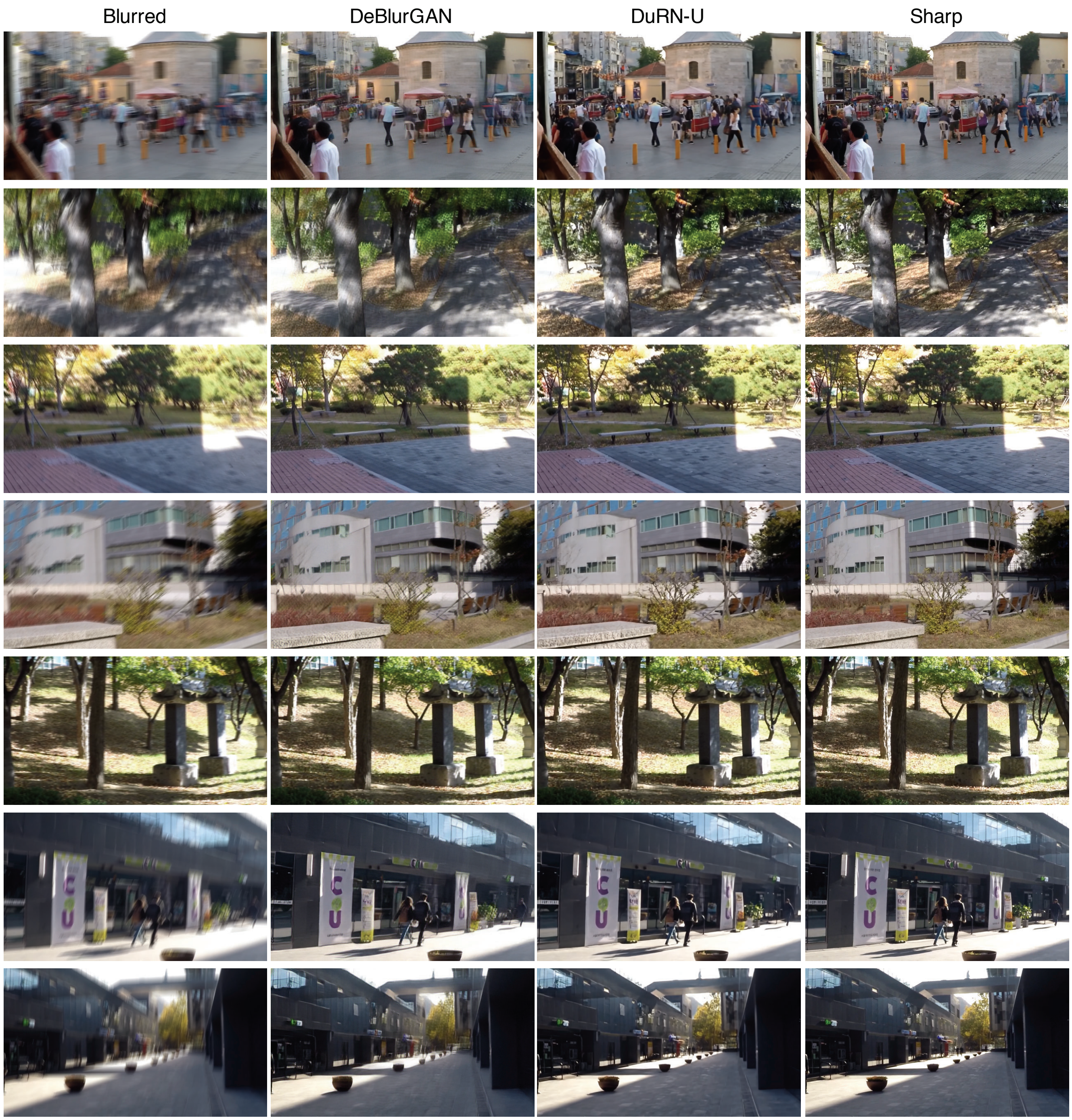

Figure 1: Examples for motion blur removal on GoPro-test dataset.

Figure 2: Examples of haze removal on synthetic hazy images.

Figure 3: Examples of haze removal on the hazy images used in previous works such as [1, 4, 6]

Figure 4: Examples of haze removal on real-world hazy images.

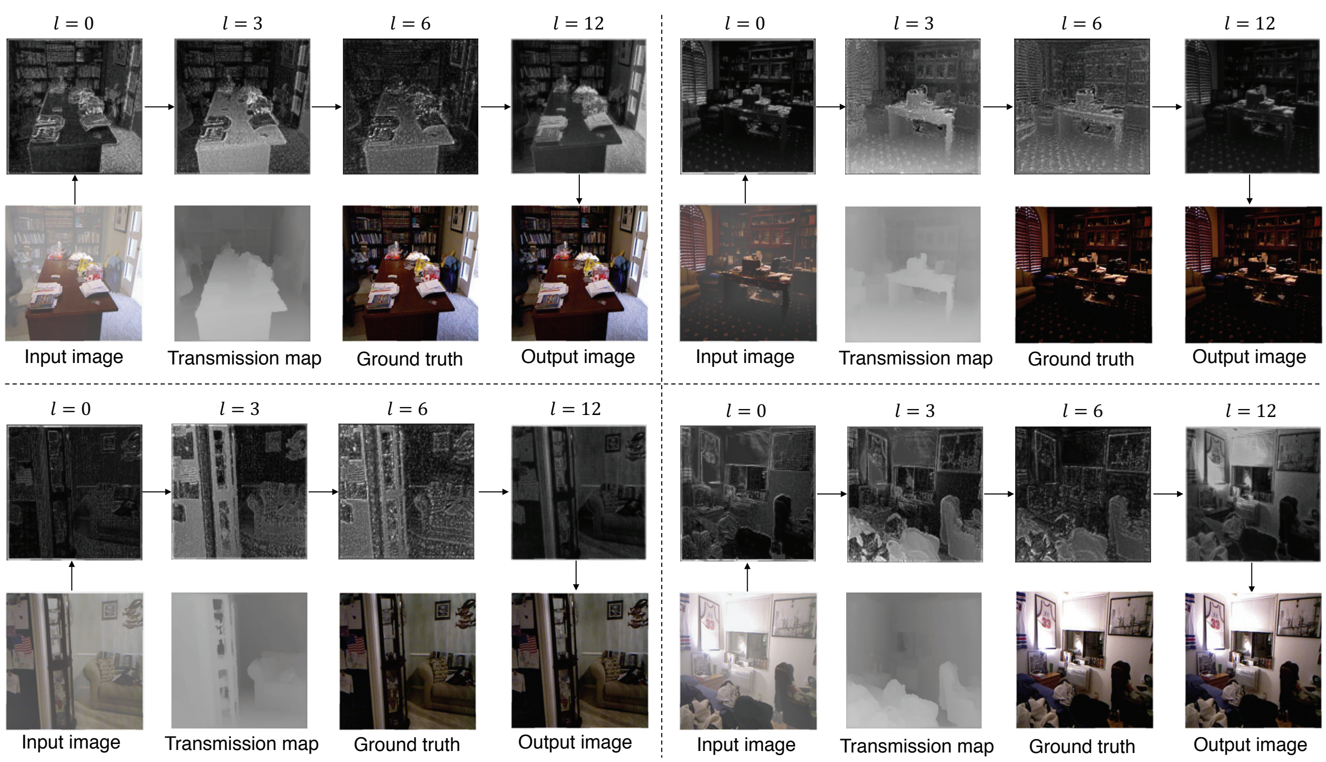

Figure 5: Visualization of internal activation maps of the DuRN-US.

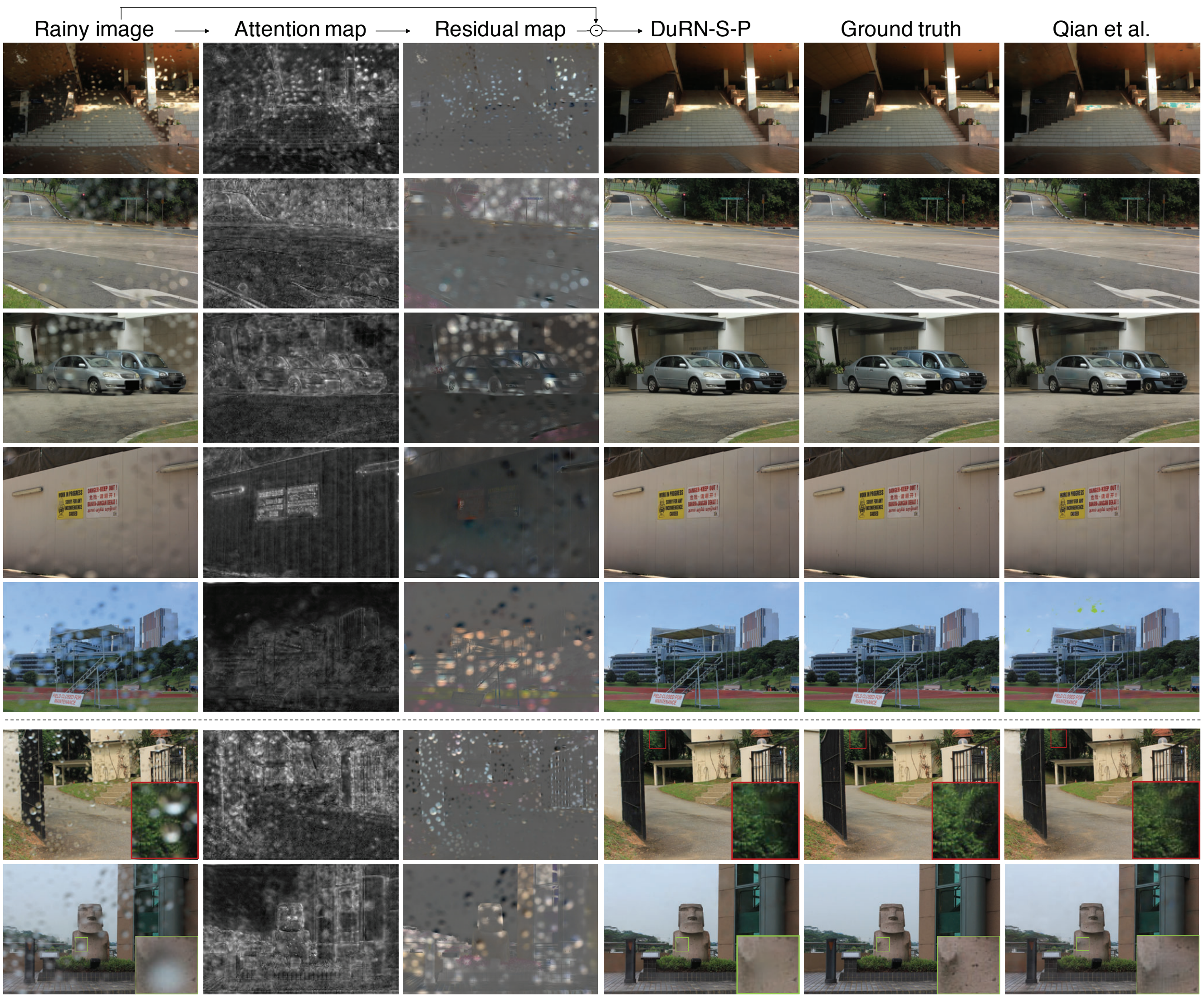

Figure 6: Examples of raindrop removal along with interal activation maps of DuRN-S-P. The “Attention map” and “Residual map” are the outputs of the Attentive-Net and the last $Tanh$ layer shwon in Fig. 13 in the main-text; they are normalized for better visibility.

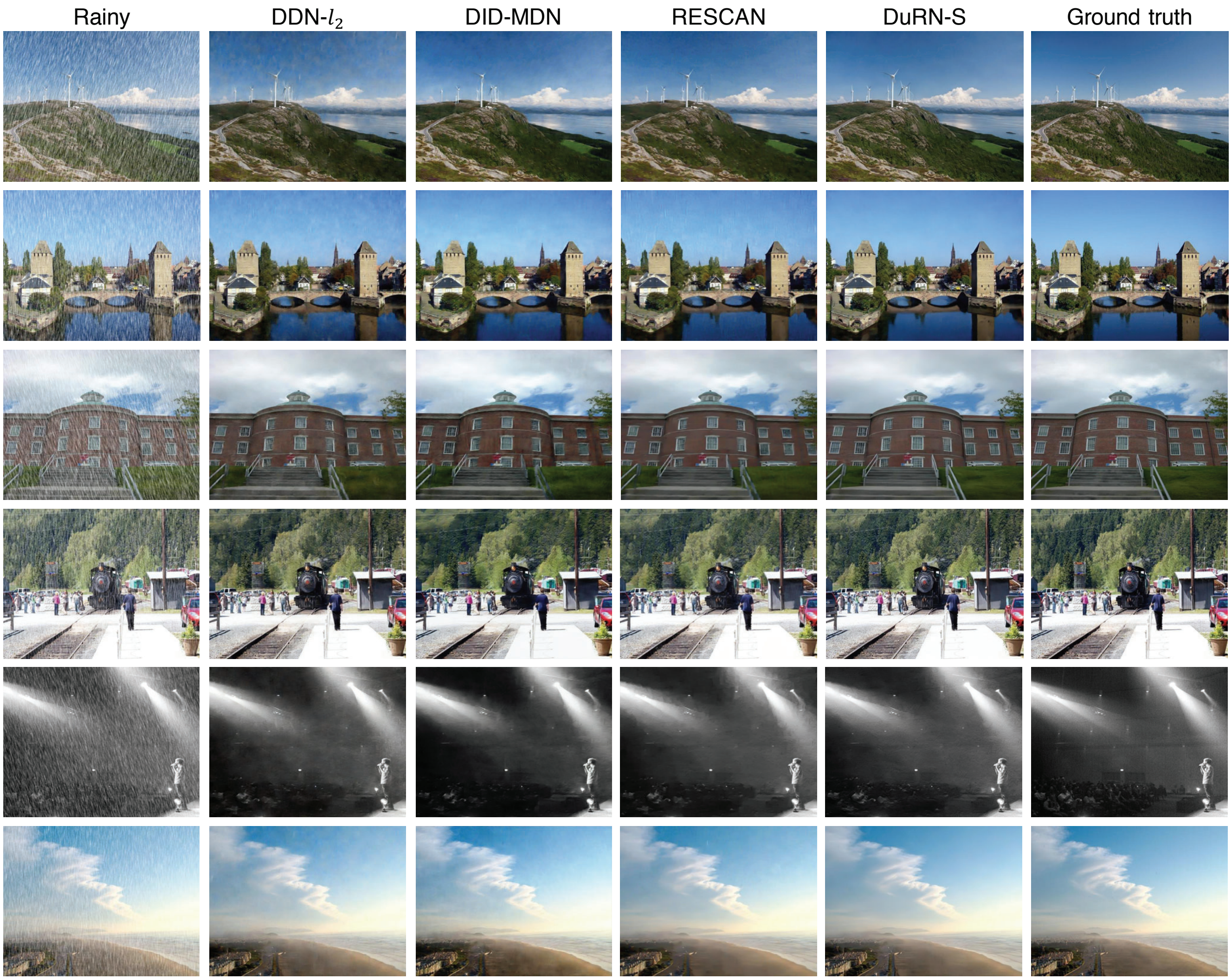

Figure 7: Examples of deraining on synthetic rainy images.

Figure 8: Examples of deraining on real-world rainy images.

References

[1] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[2] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. International Conference on Computer Vision*, 2001.

[3] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Proc. International Conference on Neural Information Processing Systems Workshop: The Future of Gradient-based Machine Learning Software and Techniques*, 2017.

[4] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *Proc. Conference on Computer Vision and Pattern Recognition*, 2018.

[5] Masanori Suganuma, Mete Ozay, and Takayuki Okatani. Exploiting the potential of standard convolutional autoencoders for image restoration by evolutionary search. In *Proc. International Conference on Machine Learning*, 2018.

[6] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *Proc. Conference on Computer Vision and Pattern Recognition*, 2018.