

Improving Image Restoration by Revisiting Global Information Aggregation (Supplementary Material)

Xiaojie Chu, Liangyu Chen, Chengpeng Chen, and Xin Lu

MEGVII Technology

In this document, we provide details of comparison between inference with (overlapping) patches and our TLC (Section A) and additional visualized results (Section B) of our approach and existing methods.

A Inference with (overlapping) patches vs our TLC

In this section, we compare our TLC with patch inference. First, Table A1 experiments on GoPro dataset found that MPRNet model with our test-time method (TLC) achieves higher PSNR (33.31 dB *vs.* 33.15 dB) with less inference time (1.69s *vs.* 6.50s) than inference on overlapping patches cropped from image. Second, boundary artifacts are also found in the predictions based on overlapping patches (Figure A1) while our TLC generates natural result without visible artifacts (Figure 2d). We will describe the details next.

A.1 Discussion of inference on overlapping patches.

Inference with overlapping patches reduces the train-test statistic inconsistency so that it also improves the performance of models. However, it has three main drawbacks:

First, it introduces additional computational costs, as the overlapping regions are restored twice or more by the entire model. While models with our TLC directly restore whole images and TLC has low extra computing costs (Table 7).

Second, it can *not* alleviate boundary artifacts on deblurring tasks. We speculate that this is because the global statistics of the two overlapping patches may differ significantly. Restoration of images affected by severe blur requires

Table A1: PSNR of MPRNet on GoPro when test with different methods. Inference time per image is measured on RTX 2080Ti.

Input	#Overlap	TLC	PSNR (dB)	Time (s)
Whole Image	-	✗	32.66	1.60
	-	✓	33.31	2.03
	-	✓ [†]	33.31	1.69
Overlapping Patches	16	-	33.09	2.60
	128	-	33.15	6.50

[†] denotes optimised implementation of TLC

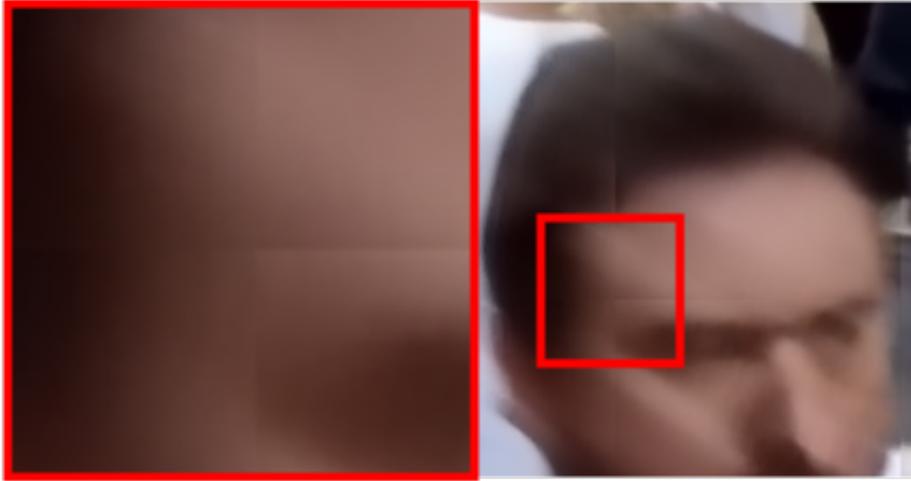


Fig. A1: MPRNet’s deblurring result when inference with overlapping 256×256 patches, which also presents vertical and horizontal artifact split lines. Overlapping of different patches is 128.

large receptive field information and the inference of many models (e.g., MPRNet) highly depends on global information. As a result, predictions of different patches have different estimate of motion blur so that their fusion also show unnatural dividing lines (Figure A1).

Third, limited size of patch limits the receptive field of information, which harms the model performance. Pixels in an overlapping region have more range of other pixels for interactions, so that the results based on more overlapping regions are better than fewer ones (Table A1). While our TLC utilises full image information and achieves the best results.

A.2 Inference Speed

Table A1 shows the inference time of MPRNet with different test methods on RTX2080Ti GPU for a 720×1280 image. With naive implementation using the cumulative sum function provided by Pytorch, TLC introduces 27% extra times (2.03s *vs.* 1.6s). Note that MPRNet with our TLC still faster and better than inference with overlapping patches. This shows the efficiency of our method.

The cumulative sum operator is unfriendly for GPU because it is a sequential algorithm. As a result, the extra test time caused by TLC will be greater than the theoretical value. A simple way to speed up is to reduce the number of cumulative sum calculations by sampling. Specifically, we can reduce the size of the matrix to r^2 times its original size by grid sampling with stride r and use the mean of the sampled matrix to approximate the mean of the original matrix. As a result, the number of calculations needed to do the (one-dimensional) cumulative sum is reduce by r times so that TLC will has $4.78\times$ faster speed. With this careful design, faster TLC only introduces 5.6% extra times (1.69s *vs.* 1.6s).

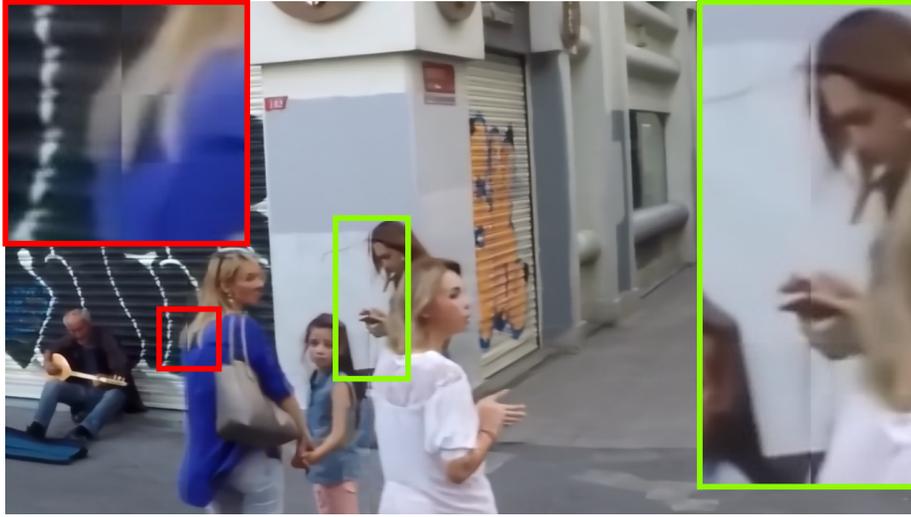


Fig. A2: Deblurring results of MPRNet [10] on GoPro [5] when inference with patches. They introduce visible artifacts at patch boundaries, which look like there are vertical stripes cutting through the picture.

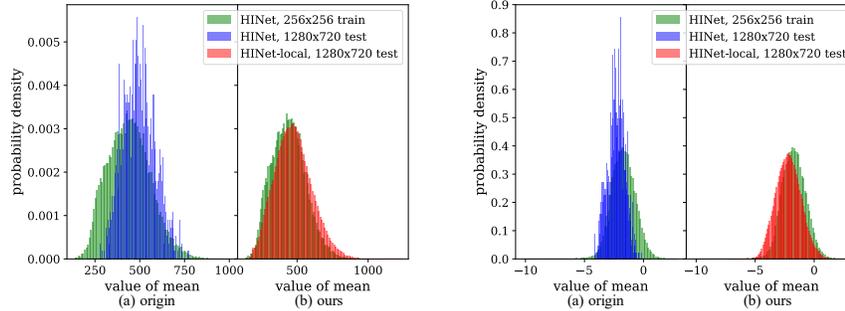
A.3 Boundary artifacts.

Inference with patches Cropping the image into patches and predict the result independently induces unsmoothness of the boundary (i.e. “boundary artifacts” as demonstrated in [3]). We give some example images of block boundary artifacts in Figure A2, which are generated by MPRNet [10] on GoPro [5] dataset. There are obvious boundary artifacts in Figure A2 which seriously degrades the quality of the image.

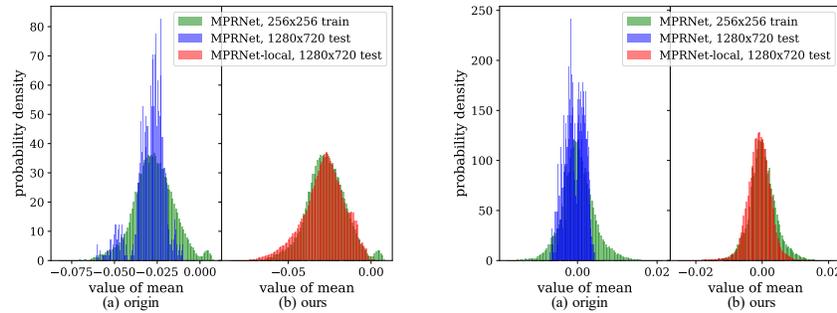
Inference with overlapping patches. We follow the implementation details of SwinIR to do inference with overlapping 256×256 patches cropped from image and the results of overlapping region are averaged. Boundary artifacts are also found in the deblurred images restored by MPRNet on GoPro dataset. An example is shown in Figure A1 where the blurry image is the same as in Figure 2a (i.e., filename: “GOPR0384_11_05-004042.png”). Compared with our TLC (Figure 2d), inference with overlapping patches (Figure A1) still introduces visible boundary artifacts. This phenomenon is different from SwinIR’s findings. We attribute this to the nature of severe motion blur and SwinIR did not test on deblurring tasks.

B Visualized Results

In this section, we provide additional visual results of statistics distribution (Section B.1) and qualitative comparisons between our approach and existing methods (Section B.2).



(i) Mean distribution of Instance Normalizations [7] in HINet [1]



(ii) Mean distribution of SE [2] layers in MPRNet [11]

Fig. A3: Visualization of the statistics (mean) distribution of features. Green: the distribution when training with patches; Blue: the distribution when inference with images; Red: the distribution when inference with images and TLC is adopted (denoted with -local suffix). For each sub-figure: (a) The original test scheme results in distribution shifts. (b) Distribution shifts could be reduced by our proposed TLC.

B.1 Global Information Distribution

We provide more results of global information (i.e., mean statistics) distribution between training and inference on GoPro [5] dataset. Statistics aggregated by HINet [1] are shown in Figure A3i. Statistics aggregated by MPRNet [11] are shown in Figure A3ii. For both HINet [1] and MPRNet [11], the statistics distribution shifts from training (green) to inference (blue). The statistics distribution shifts is reduced by TLC as shown in Figure A3i-b and Figure A3ii-b compares to Figure A3i-a and Figure A3ii-a: the statistics distribution obtained by our HINet-local/MPRNet-local (red) is close to the original HINet/MPRNet in the training phase (green).

B.2 Qualitative Comparisons

In this section, provide additional qualitative results on various image restoration tasks (e.g. deburring, deraining and dehazing) for qualitative comparisons.

Deburring. We give the comparison of the visual effect in Figure A4 for qualitative comparisons. Compared to the original MPRNet [10] which test with patches (Figure A4b), our approach (Figure A4d) restores high quality images without boundary artifacts. Compared to the original MPRNet [10] which test with images (Figure A4c), our approach restores clearer and sharper images.

Deraining. We give the comparison of the visual effect in Figure A5 for qualitative comparisons. Compared to the original SPDNet [10], our approach restores clearer images.

Dehazing. We give the comparison of the visual effect in Figure A6 for qualitative comparisons. Compared to the original FFANet [10], our approach restores clearer images.

References

1. Chen, L., Lu, X., Zhang, J., Chu, X., Chen, C.: Hinet: Half instance normalization network for image restoration. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2021)
2. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
3. Lee, N.Y.: Block-iterative richardson-lucy methods for image deblurring. EURASIP Journal on Image and Video Processing **2015**(1), 1–17 (2015)
4. Li, B., Ren, W., Fu, D., Tao, D., Feng, D., Zeng, W., Wang, Z.: Benchmarking single-image dehazing and beyond. IEEE Transactions on Image Processing pp. 492–505 (2018)
5. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3883–3891 (2017)
6. Qin, X., Wang, Z., Bai, Y., Xie, X., Jia, H.: Ffa-net: Feature fusion attention network for single image dehazing. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 11908–11915 (2020)
7. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
8. Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., Lau, R.W.: Spatial attentive single-image deraining with a high quality real rain dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12270–12279 (2019)
9. Yi, Q., Li, J., Dai, Q., Fang, F., Zhang, G., Zeng, T.: Structure-preserving deraining with residue channel prior guidance. In: IEEE International Conference on Computer Vision (2021)
10. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14821–14831 (2021)
11. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: CVPR (2021)



Fig. A4: Deblurring results of MPRNet [10] on GoPro [5] generated by different inference schemes. **Left:** full-images. **Right:** crops from left image. (b) Test with patches; (c) Test with images; (d) Test with images and TLC is adopted (ours). It illustrates that (d) provides sharper results than (c) while avoids the boundary artifacts in (b).

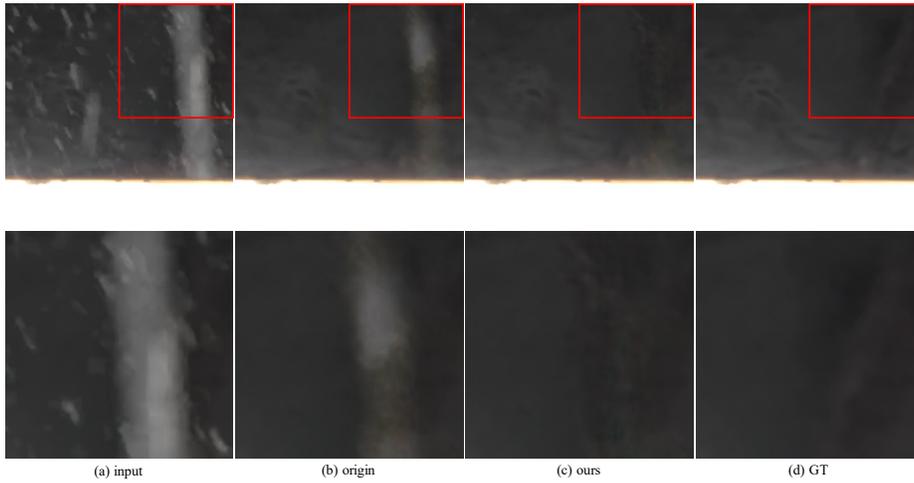


Fig. A5: Deraining results of SPDNet [9] on SPA-Data [8] dataset generated by different inference methods. (a) Rainy images as inputs. (b) Results based on full-image produced by the original SPDNet. Some of the rainwater in images is not removed cleanly. (c) Results based on full-image produced by SPDNet with the proposed TLC, which are clearer. (d) Ground truth for reference.

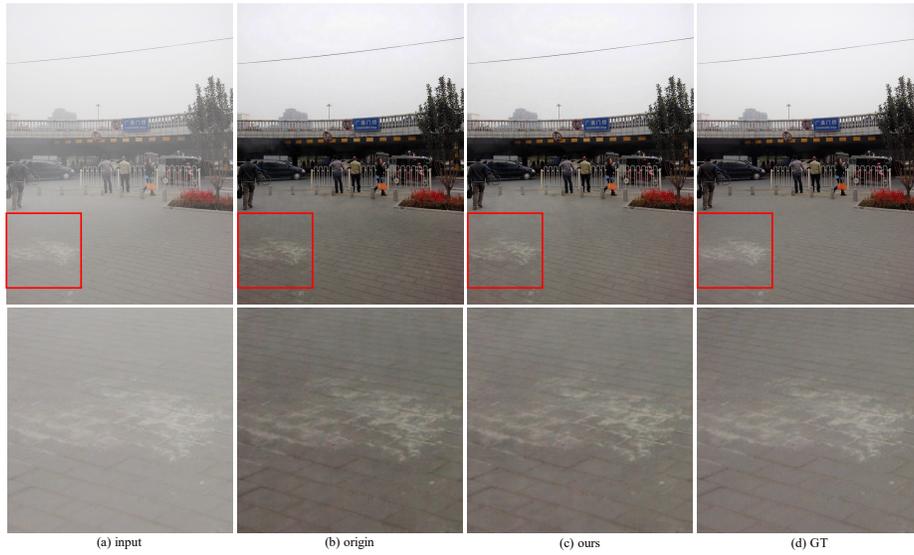


Fig. A6: Dehazing results of FFANet [6] on Synthetic Objective Testing Set (SOTS) from RESIDE [4] dataset generated by different inference methods. (a) Hazy images as inputs. (b) Result produced by original FFANet, which is gray with obvious noise. (c) Result produced by FFANet with our TLC, which is brighter with fewer noises. (d) Ground truth for reference.