Deep Photo Scan: Semi-Supervised Learning for dealing with the real-world degradation in Smartphone Photo Scanning

Man M. Ho
Hosei University
Tokyo, Japan
man.hominh.6m@stu.hosei.ac.jp

Jinjia Zhou
Hosei University
Tokyo, Japan
jinjia.zhou.35@hosei.ac.jp

Contents

1. Technical details

1.1. Network Architectures

1.1.1 G1

$G1$ includes two main components: encoder and decoder. In that, each consists of 4 RECA U-Blocks. The reconstructed image $\hat{Y}$ is inferred from the scanned photo X through $G1$ as (according to the inference flow): Convolutional Layer (Conv2D+EvoNorm [11]) $\longrightarrow$ RECA Block $\longrightarrow$ Encoder $\longrightarrow$ Residual Blocks (ResBlocks) $\longrightarrow$ Decoder $\longrightarrow$ Convolutional Layer $\longrightarrow$ Conv2D + Tanh. Each encoder and decoder contains 3 residual layers (ResBlock), and each ResBlock consists of 2 convolutional layers. The first Conv2D uses the kernel size of 5, padding of 3, and a stride of 1; meanwhile, all other Conv2D modules use the kernel size of 3, padding of 1, and a stride of 1. The channel sizes in inference order are set as: $[3, 32, 32]$ for the first two layers, $[64, 128, 256, 512]$ for 4 RECA U-Blocks of the encoder, $[512, 512, 512]$ for 3 ResBlocks, $[256, 128, 64, 32]$ for RECA U-Blocks in the decoder, and $[32, 3]$ for two last layers, where the final channel size of 3 represents of the restored image Y, which values are in a range of $[-1, 1]$.

1.1.2 G2

The network architecture and inference flow for G2 are similar to $G1$. However, it does not have the RECA Block. Besides, the RECA U-Blocks are replaced by pure convolutional layers. The channel sizes and skip-connections transferring feature maps from the encoder part to the decoder part are described in Figures 1 and 2.

1.2. Implementation

We train our models using Adam optimizer [9] with a learning rate of 0.0001 for generators, 0.0004 for the discriminator, $\beta_1 = 0.9$, $\beta_2 = 0.999$, the batch size of 8 for pre-training and 4 for fine-tuning. Our models are pre-trained and fine-tuned in $200,000$ iterations.

2. Discussion

2.1. Failed to detect a complete contour

The scanned photo's contour needs to be identified before perspective warp transforming a smartphone-scanned photo to have a top-down view. The traditional edge detection techniques [1, 8] achieve real-time performance in

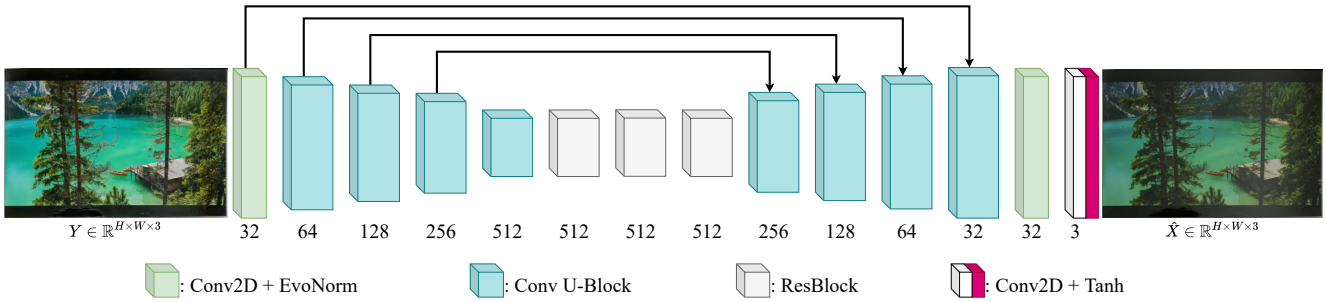

Figure 1. The baseline network architecture for the generators of DPScan as well as for $G2$ inferring the smartphone-scanned $\hat{X}$ from the high-quality Y.

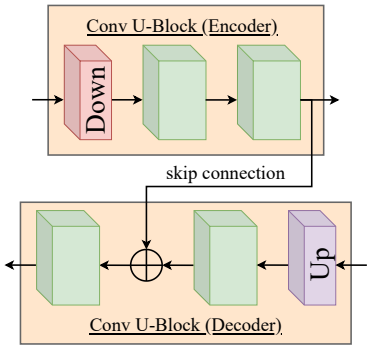

Figure 2. Illustration of Conv U-Block in encoder and decoder parts. Green block denotes Conv2D + EvoNorm [11]. Regarding "*DOWN*"-sampling and "*UP*"-sampling, we use the anti-aliasing max pooling and bi-linear interpolation from [19]. $\bigoplus$ represents a summation.

Method	Valset	Testset
Flow-Warping Block (FWB)	21.89	22.03
Residual Feature-based Attention (RFA)	21.79	22.03
Residual Self-Attention (RSA)	21.86	22.12
Residual Channel Attention Block (RCAB)	21.83	22.19
Residual Efficient Channel Attention (RECA)	**22.14**	**22.46**

Table 1. Ablation study on deep modules, such as Flow-Warping Block (FWB) [5], Residual Feature-based Attention (RFA), Residual Self-Attention (RSA) [18], Residual Channel Attention Block (RCAB) [21], and Residual Efficient Channel Attention (RECA) [17]. We one-by-one add the ablation modules after the first layer of the baseline architecture for DPScan (similar to the architecture of G2 shown in Figure 1) and train all ablation models in the same condition. The customized RECA shows the best restoration performance in PSNR (*higher is better*) as **bold** values.

identifying the contour of interests; however, they still suffer from the homologous colors between the boundary leading to an incomplete contour, as shown in Figure 4. DNN-based edge detection [13] can achieve much better performance. However, it requires a longer processing time. Taking advantage of the methods as mentioned earlier, we combine [1] (with opening morphing noise reduction) and DNN-based [13] techniques for our contour detection, which can significantly reduce the processing time with high performance. We manually correct failed cases by drawing additional lines or providing a new contour when both methods are failed in detecting a complete contour, as shown in Figure 4.

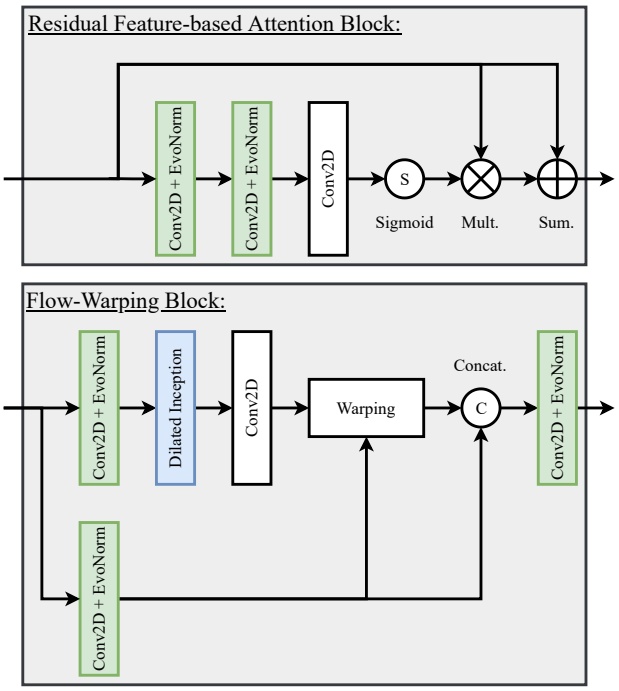

Figure 3. Illustrations of ablation modules such as Flow-Warping Block (FWB) and Residual Feature-based Attention (RFA). FWB is designed to solve the structural mismatch at the beginning.

2.2. Unstable performance of GAN-based framework in training

Generative Adversarial Network (GAN) and its variants have achieved high performance in image synthesis; how-

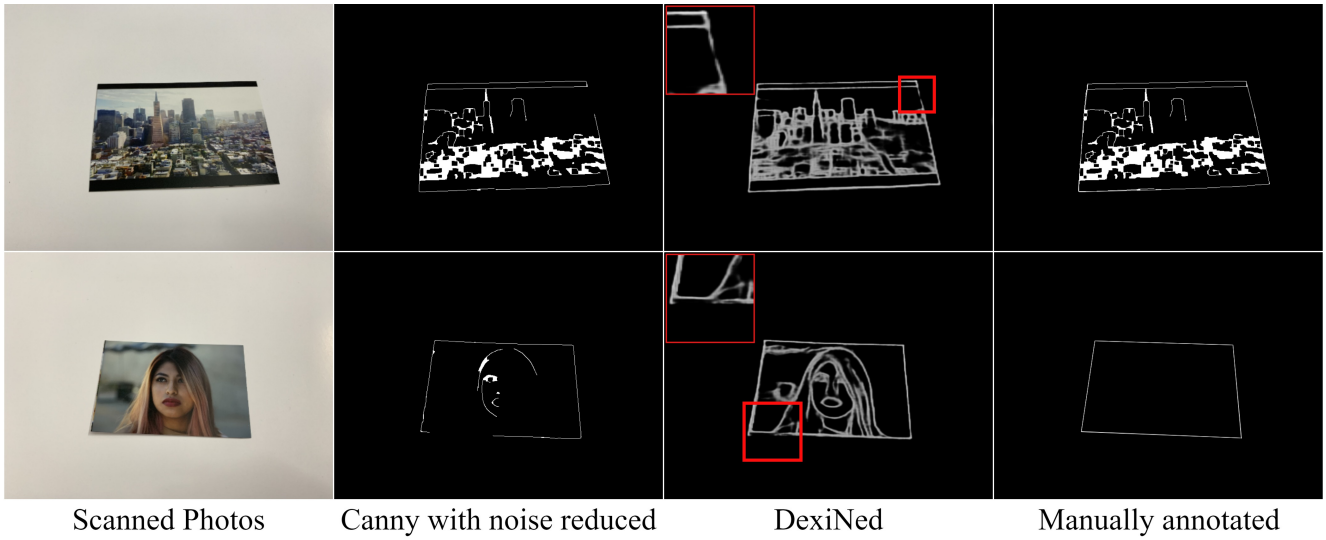

Figure 4. Detecting contours of interests using Canny edge detection [1] with opening morphing noise reduction (*second column*) and DexiNed [13] (*third column*). We manually annotate the contours when both methods are failed (*last column*). Highlighted rectangles reveal the failure in detecting a complete contour.

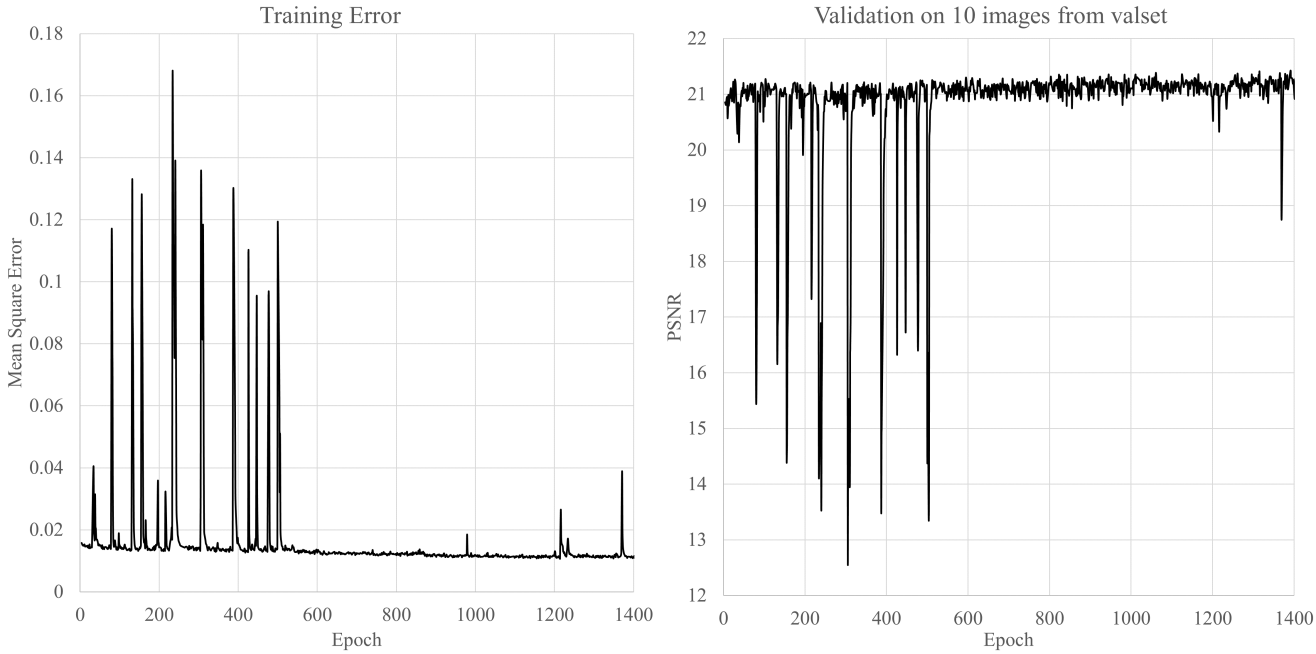

Figure 5. Mean Square Errors of G1 in training and the restoration performance (PSNR) in validation when $G1$, $G2$, and $D2$ are trained together. The training error sometimes becomes dramatically worse. Eventually, the restoration performance of $G1$ is still improved.

ever, they often generate unreal textures such as in Pix2Pix [6]'s restored images as well as our pseudo inputs, as shown in Figure 6. Consequently, the performance of our semi-supervised DPScan is unstable while training on them, as shown in Figure 5. We also visualize in Figure 6 the restored images being dramatically worse during the epoch 499 to 506, which is described in Figure 5. Even so, the restoration performance of our DPScan is still significantly improved using the proposed semi-supervised approach, providing fewer artifacts with fine edges than its supervised pre-trained model.

a) Our G2 generates abnormal textures while training with G1

b) Pix2Pix generates abnormal textures

c) Generated samples of G1 during epochs from 499 to 506

Figure 6. GAN-based network occasionally generates abnormal textures, as described in (a) the pseudo inputs of our G2 while being trained with G1 and (b) restored images of the re-trained Pix2Pix [6]. Addressing this issue can lead to further improvement. Also, we show the restored images of G1 during epochs 499 to 506, visualizing the unstable performance of the GAN-based framework while training.

3. Experiments

3.1. Deep modules for network architecture

We adopt U-Net [14], residual modules [4, 7], anti-aliasing down-/up-samplers [19], EvoNorm [11] to design a base architecture for DPScan, as shown in Figure 1 and 2. While considering improving network architecture, we conduct an ablation study on customized deep learning techniques such as Flow-Warping Block (FWB) [5], Residual Feature-based Attention (RFA), Residual Self-Attention (RSA) [18], Residual Channel Attention Block (RCAB) [21], and Residual Efficient Channel Attention (RECA) [17]. In that, FWB is designed to correct structural misalignment at the beginning of the network, while RFA extracts the attention based on the whole feature map using a Sigmoid function, as shown in Figure 3. Besides, RSA has down-/up-sampling pooling placed between Self-Attention [18] to reduce the computational cost. We one-by-one add the ablation modules after the first layer of the baseline architecture and train them in the same condition. As a result, the customized RECA outperforms other ablation techniques with the best PSNR as **22.14** and **22.46** on *valset* and *testset*, respectively, with the first center crop ratio $R_1 = 75\%$ (no further processing), as shown in Table 1.

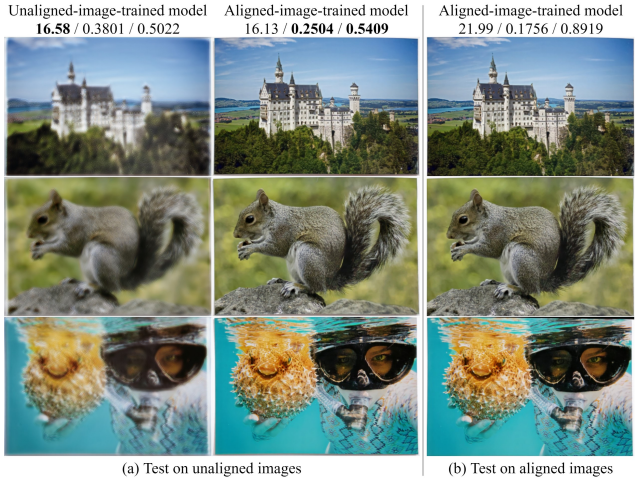

Figure 7. Ablation study on training data including aligned and unaligned DIV2K-SCAN. We compare two models (baseline DP-Scan), which are trained on unaligned/aligned images in 500 epochs, on both (a) unaligned and (b) aligned testset qualitatively and quantitatively in PSNR ↑ /LPIPS ↓ /MS-SSIM ↑. **Bold** values show the best performance corresponding to the metric for (a) on DIV2K-SCAN *testset*. Although the model trained on unaligned images has a better performance in correcting distorted shapes as higher PSNR, the details are not restored well compared with the model trained on aligned images. Therefore, the pose in input and ground-truth images must be the same.

3.2. How important image alignment is for learning capability and evaluation

As mentioned about data preparation in the main paper, we avoid learning structural correction due to misalignment remaining in data using traditional image alignment [12, 2, 15], so that our DPScan can be trained effectively to solve the real-world degradation. In this ablation study, we train a baseline architecture of DPScan on aligned and unaligned DIV2K-SCAN in the same condition. As a result, although the model trained on unaligned images quantitatively outperforms the model trained on aligned images on the unaligned testset (a), it generates highly distorted results. In contrast, the model trained on aligned images can provide much better restoration performance with finer edges. It says, due to the misalignment in data, 1) similarity metrics are falsified, and 2) learning capability is harmed as the model trained on unaligned images also tries to correct the pose, leading to significantly distorted images, as evaluated in Figure 7. Even though the misalignment is significantly reduced after applying image alignment with SIFT and RANSAC, the local misalignment still occurs. We thus present Local Alignment (LA) to reduce the remaining misalignment. A quantitative result of locally-aligned scanned photos (inputs) in Figure 10 shows that the image quality of the inputs is gradually reduced in the ascending order of image size. That is to say, the larger the image size, the more serious the misalignment, the lower the restoration performance, the less reliable the evaluation using similarity metrics. We thus train our network on locally-aligned images cropped to 256×256 and evaluate all methods on three image sizes 176×176, 256×256, and 384×384.

3.3. An illustration of pseudo-scanned photos

We visualize our pseudo-scanned photos in iPhone XR and generalized domains in Figure 8. Generating pseudo-scanned photos for unscanned images helps diversify our training image content, leading to better restoration performance, as proved in the main paper.

3.4. A comparison of performance on mimicking the smartphone-scanned photo degradation

The purpose of mimicking the degradation of smartphone-scanned photos is to provide pseudo inputs for an unlimited amount of high-quality images so that our network can be trained on them, representing a semi-supervised learning approach. In this work, we adopt the concept of GANs [3, 18, 6] to train our degradation network $G2$ to degrade the unscanned photos as if a smartphone scanned them. As a result, our work provides the pseudo-scanned photos closer to the real-scanned photos than CycleGAN [22] trained in the same condition. Quantitatively, our $G2$ obtains a higher PSNR of **24.8 dB**, higher MS-SSIM of **0.9364**, approximate LPIPS of **0.1542** on *testset*, as shown in Figure 9.

3.5. Ablation study on the number of simulated domains for fine-tuning the pre-trained 1-domain DPScan

This section conducts an ablation study on how the number of simulated domains (K) affects our generalization performance in fine-tuning 1D-DPScan pre-trained on iPhone XR. As a result, the Generalized DPScan (G-DPScan) gains much better generalization in the first $100,000$ iterations when $K = 75$. Since DPScan may need more iterations to generalize so many photos from 100 domains, we continue fine-tuning the model with $K = 75$ and $K = 100$ more $100,000$ iterations. As a result, G-DPScan with $K = 100$ can gain the highest generalization performance; meanwhile, the performance of G-DPScan with $K = 75$ is saturated, as shown in Figure 2.

3.6. A quantitative comparison with previous works on 1-domain DIV2K-SCAN

We show a comprehensive version of the quantitative comparison between ablation models and the previous work Pix2Pix [6] and CycleGAN [22] presented in our main paper using PSNR, LPIPS [20], and MS-SSIM on the image sizes of 176×176, 256×256, 384×384, 576×576, and 1072×720 in Figure 10. The previous works and all ablation models are trained and evaluated on 1-domain DIV2K-SCAN (iPhone XR). As a result, the final version of our DPScan trained on only iPhone XR (1D-DPScan) outperforms its baseline architecture, typical works [6, 22] trained in the same condition.

3.7. A quantitative comparison of generalization performance with previous works and industrial products on multiple-domain DIV2K-SCAN

We show a comprehensive version of the quantitative comparison of generalization performance presented in our main paper using PSNR, LPIPS [20], and MS-SSIM on the image sizes of 176×176, 256×256, 384×384, 576×576, and 1072×720 in Figure 11. Even though our performance is reduced after generalizing our 1D-DPScan (DPScan trained on iPhone XR only) on iPhone XR, Generalized DPScan (G-DPScan) significantly outperforms its 1-domain version on other unseen domains, previous research work OPR [16] and industrial products Google Photo Scan and Genius Scan entirely.

3.8. A qualitative comparison between 1-domain and multiple-domain (generalized) works

We show a full version of the qualitative comparison on iPhone XR in Figure 12, and the additional comparisons

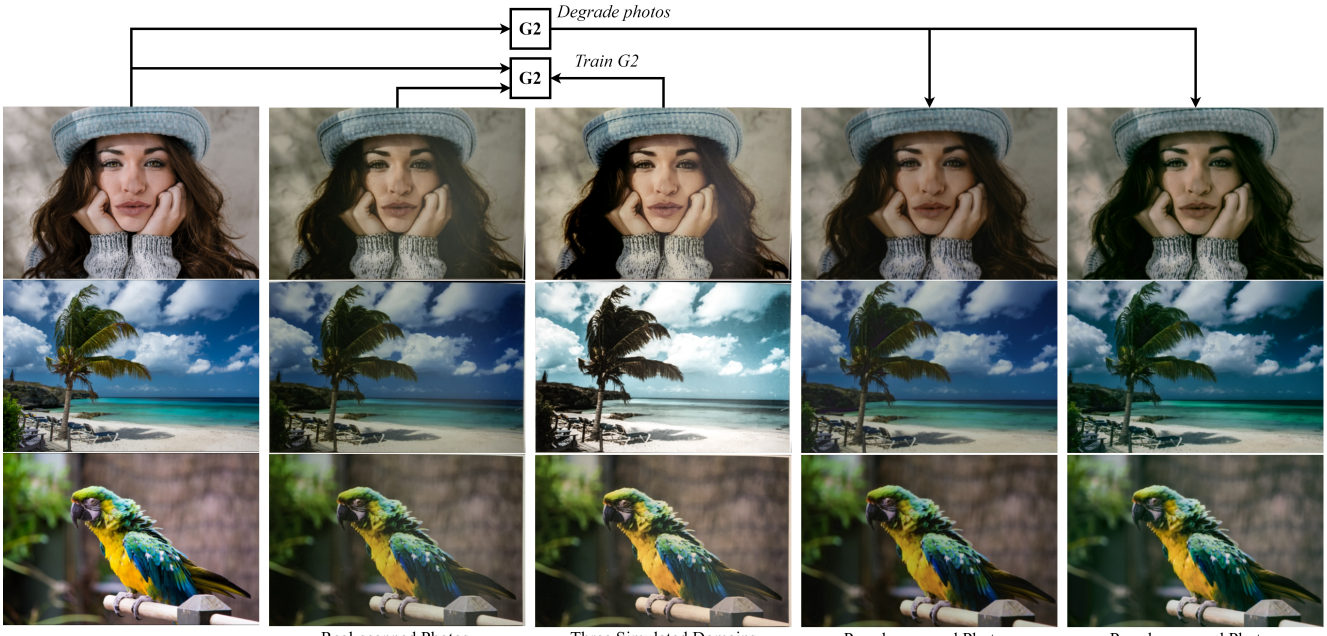

Figure 8. Our pseudo-scanned photos in iPhone XR and generalized domains from 1D-DPScan and G-DPScan, respectively. Generating pseudo-scanned photos for unscanned photos helps diversify our training image content, leading to better restoration performance.

Method	#Styles	iPhone XR (seen)			iPhone XR + SCB (unseen)			Xperia XZ1 (unseen)			Average		
		PSNR↑	LPIPS↓	MS-SSIM↑	PSNR↑	LPIPS↓	MS-SSIM↑	PSNR↑	LPIPS↓	MS-SSIM↑	PSNR↑	LPIPS↓	MS-SSIM↑
1D-DPScan	0	**25.26**	**0.1242**	**0.9446**	20.80	0.1883	0.9172	21.65	0.2357	0.8972	22.57	0.1827	0.9197
G-DPScan	25	24.55	0.1389	0.9387	22.22	0.1709	0.9268	22.27	0.2178	0.9042	23.02	0.1759	**0.9232**
	50	24.35	0.1399	0.9376	22.63	0.1684	0.9273	22.09	0.2265	0.8993	23.02	0.1783	0.9214
	75	24.40	0.1417	0.9386	22.71	0.1670	0.9252	22.25	0.2252	0.9030	23.12	0.1780	0.9223
	100	24.30	0.1408	0.9378	22.68	0.1703	0.9258	21.95	0.2234	0.9011	22.98	0.1782	0.9216
	75*	24.21	0.1439	0.9374	**22.97**	**0.1602**	**0.9276**	22.11	0.2158	0.9001	23.10	0.1733	0.9217
	100*	24.10	0.1413	0.9363	22.93	0.1610	**0.9276**	**22.48**	**0.2134**	**0.9045**	**23.17**	**0.1719**	0.9228

Table 2. Ablation study on the number of simulated domains (K) for fine-tuning 1D-DPSCan pre-trained on iPhone XR, where $K \in \{25, 50, 75, 100\}$, in $100,000$ iterations. * means we continue fine-tuning the models more $100,000$ iterations. By fine-tuning more iterations, 1D-DPScan with $K = 100$ can gain the better generalization performance as **bold** (best)/ underline (second best) values; meanwhile, the performance of 1D-DPScan with $K = 75$ is saturated. ↑/↓: higher/lower is better.

with the industrial products Google Photo Scan, Genius Scan, the previous work Old Photo Restoration [16], two re-trained Pix2Pix [6], CycleGAN [22] in Figures 15 and 16.

Besides, we provide a full version of qualitative comparison of generalization performance on many different domains including iPhone XR, Simplest-Color-Balanced [10] DIV2K-SCAN (iPhone XR+SCB) and photos taken by Xperia XZ1, as shown in Figures 12, 13, and 14 respectively.

References

[1] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 1986.

[2] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Man M Ho, Jinjia Zhou, Gang He, Muchen Li, and Lei Li. Sr-cl-dmc: P-frame coding with super-resolution, color learning, and deep motion compensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 124–125, 2020.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adver-

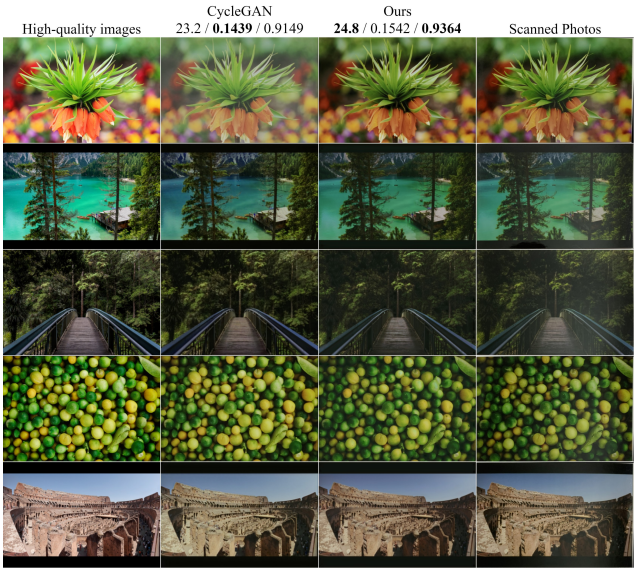

Figure 9. Comparison on mimicking degradation to CycleGAN [22] qualitatively and quantitatively in PSNR ↑ / LPIPS ↓ / MS-SSIM ↑. **Bold** values denotes the best performance corresponding to the metric on DIV2K-SCAN *testset*.

sarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[8] Claudio Rosito Jung and Rodrigo Schramm. Rectangle detection based on a windowed hough transform. In *Proceedings. 17th Brazilian Symposium on Computer Graphics and Image Processing*, pages 113–120. IEEE, 2004.

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

[10] Nicolas Limare, Jose-Luis Lisani, Jean-Michel Morel, Ana Belén Petro, and Catalina Sbert. Simplest color balance. *Image Processing On Line*, 1:297–315, 2011.

[11] Hanxiao Liu, Andrew Brock, Karen Simonyan, and Quoc V Le. Evolving normalization-activation layers. *arXiv preprint arXiv:2004.02967*, 2020.

[12] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[13] Xavier Soria Poma, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1923–1932, 2020.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[15] Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[16] Ziyu Wan, Bo Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Jing Liao, and Fang Wen. Old photo restoration via deep latent space translation. *arXiv preprint arXiv:2009.07047*, 2020.

[17] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11534–11542, 2020.

[18] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019.

[19] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.

[20] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[21] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.

[22] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

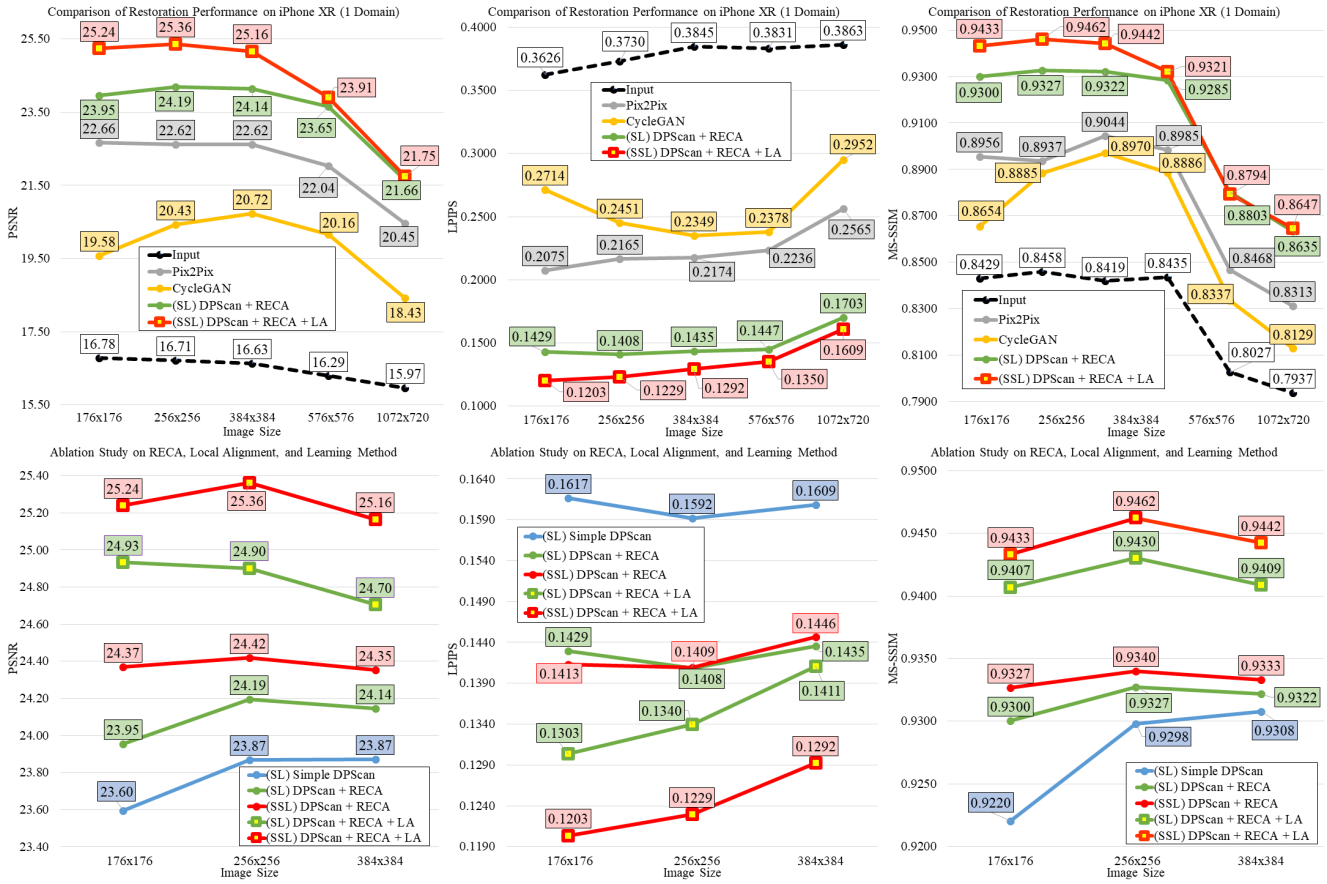

Figure 10. A comprehensive version of the quantitative comparison presented in our main paper using PSNR, LPIPS [20], and MS-SSIM on the image sizes of 176×176, 256×256, 384×384, 576×576, and 1072×720. The reduction of image quality in ascending order of image size reveals the local misalignment remaining in data. The final version of our DPScan outperforms its baseline architecture, previous research works. All models are trained and evaluated on 1-domain DIV2K (iPhone XR).

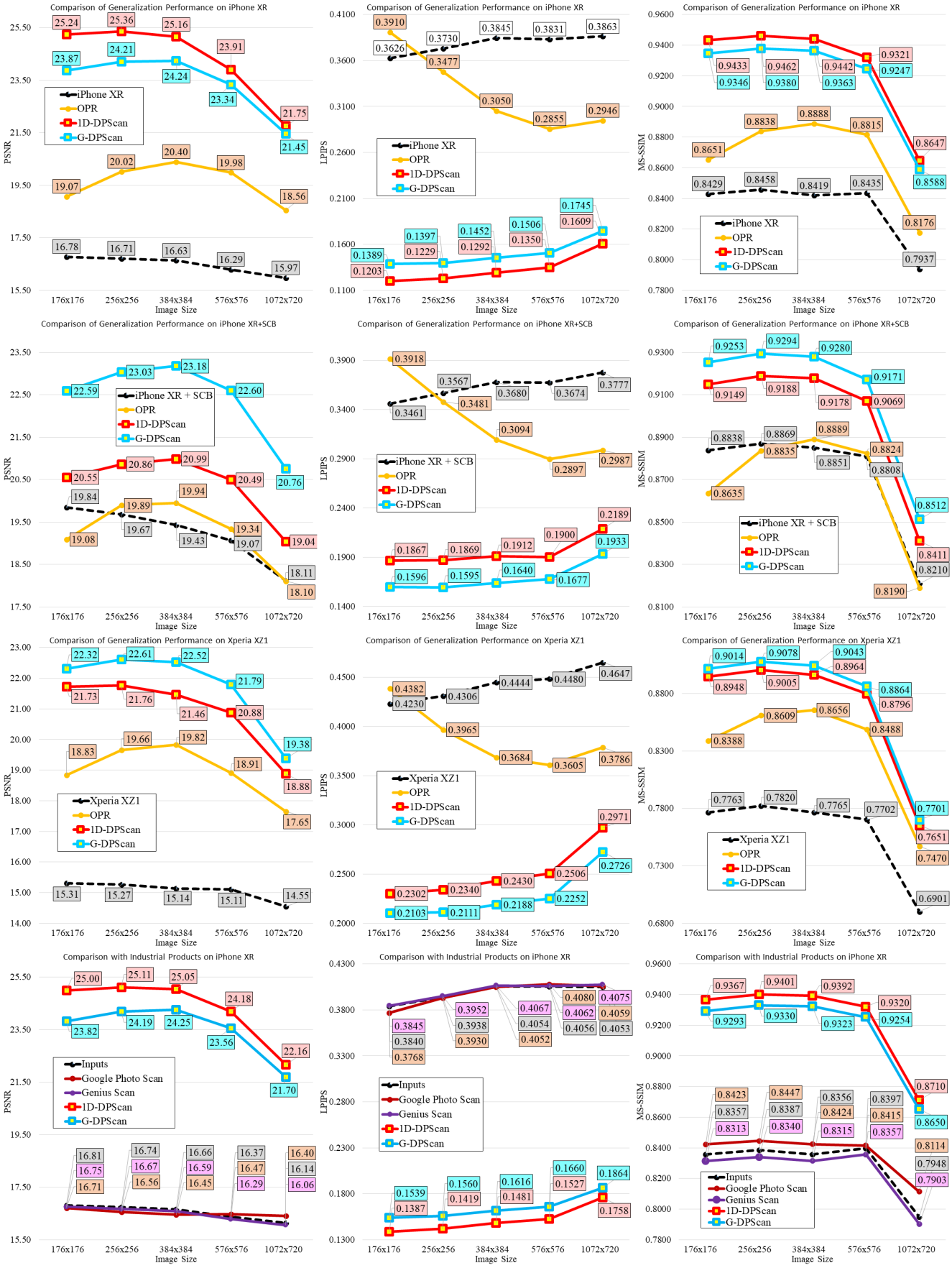

Figure 11. An extended version of the quantitative comparison presented in our main paper using PSNR, LPIPS [20], and MS-SSIM on the image size from 176×176 to 1072×720. Even though the performance is reduced after generalizing 1D-DPScan (DPScan trained on iPhone XR only) on iPhone XR, Generalized DPScan (G-DPScan) significantly outperforms its 1-domain version on other unseen domains, previous research work OPR [16] and industrial products Google Photo Scan and Genius Scan in total.

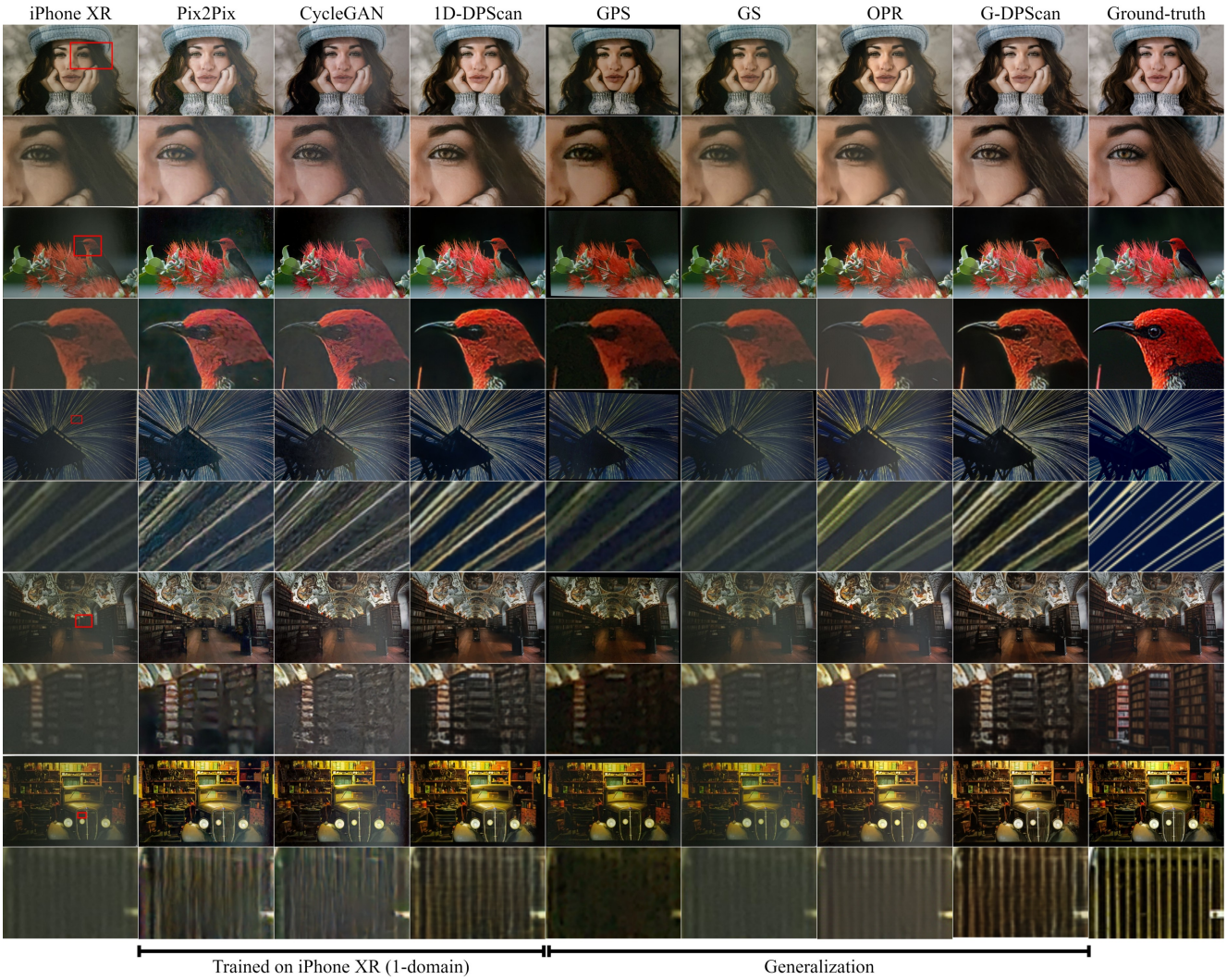

Figure 12. A full version of qualitative comparison shown in the main paper between two typical works Pix2Pix [6], CycleGAN [22] trained on 1-domain DIV2K-SCAN (iPhone XR), two industrial products Google Photo Scan (GPS), Genius Scan (GS), the previous work Old Photo Restoration (OPR) [16], and our 1-domain DPScan, Generalized DPScan (G-DPScan). Ours produces the most detailed photos without glare and color fading. **Better when zoomed in**.

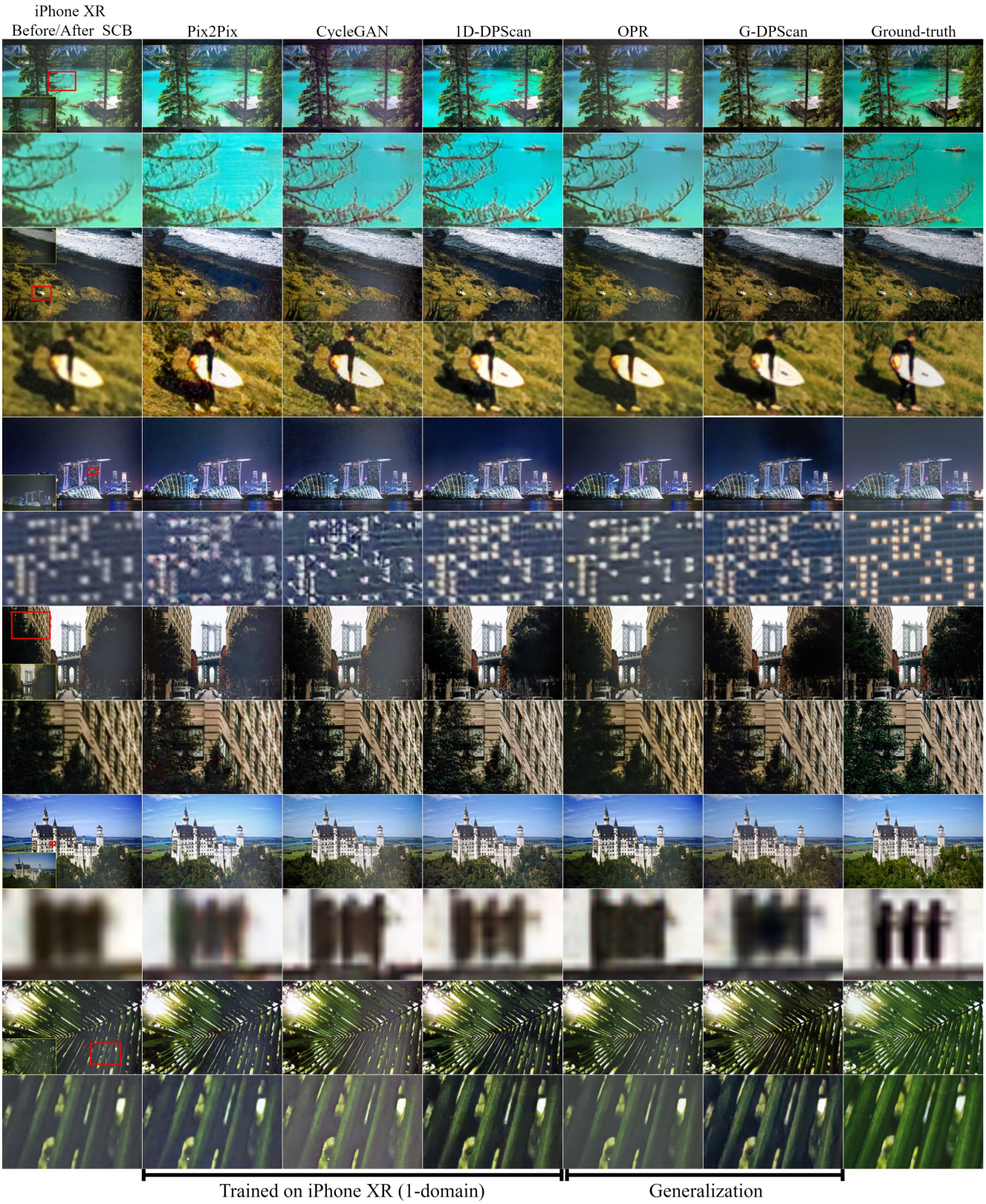

Figure 13. Qualitative comparison in case of out-of-distribution between our work, Old Photo Restoration (OPR) [16], Pix2Pix [6], CycleGAN [22] on Simplest-Color-Balanced (SCB) [10] DIV2K-SCAN.

Figure 14. Qualitative comparison in case of out-of-distribution between our work, Old Photo Restoration (OPR) [16], Pix2Pix [6], CycleGAN [22] on testset taken by Xperia XZ1.

Figure 15. Additional comparison 1.

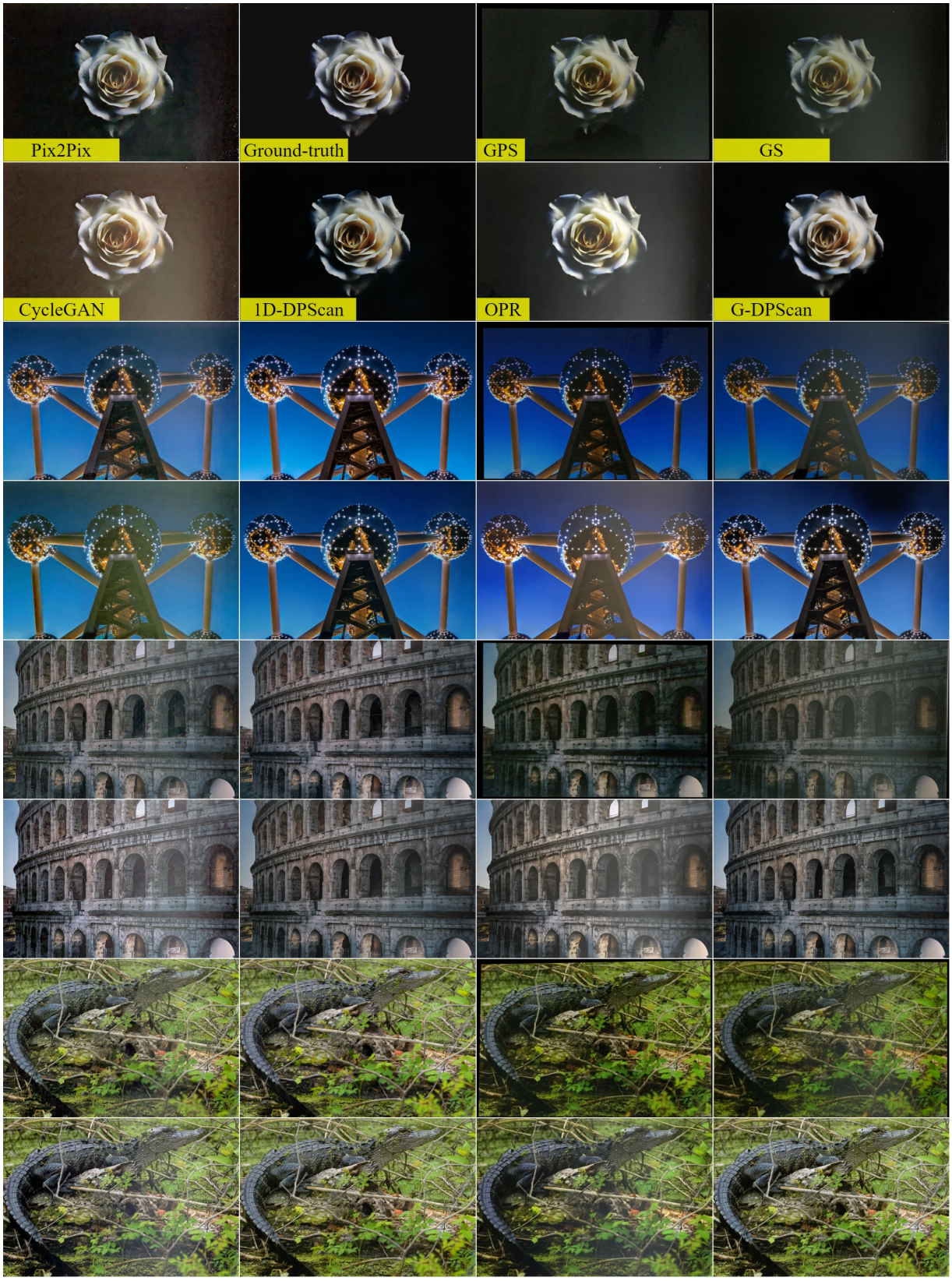

Figure 16. Additional comparison 2.