Supplementary Material for Paper #7916: ABPN: Adaptive Blend Pyramid Network for Real-Time Local Retouching of Ultra High-Resolution Photo

Biwen Lei†, Xiefan Guo*, Hongyu Yang, Miaomiao Cui†, Xuansong Xie†, Di Huang
†DAMO Academy, Alibaba Group
biwen.lbw@alibaba-inc.com, {guoxiefan, hongyu.yang.cv}@gmail.com,
miaomiao.cmm@alibaba-inc.com, xingtong.xxs@taobao.com, dhuang.cv@outlook.com

In this document, we present some additional statistics and more examples of the CRHD-3K dataset in Sec. 1, implementation details in Sec. 2, additional experiments in Sec. 3, limitations of the proposed method in Sec. 4, and discussions about the potential negative impact of our work in Sec. 5.

1. The CRHD-3K Dataset

This section provides more information about the CRHD-3K dataset. In Fig. 1, we show the statistics of retouching mask ratios, *i.e.*, retouching region-to-image area ratios. Fig. 2, 3, and 4 provide more examples from the CRHD-3K dataset, which are classified by *gender*, *type of portrait photo* and *type of clothing*, respectively.

License. As mentioned in the main paper, the CRHD-3K dataset consists of 3,022 high-definition raw portrait photos from the Unsplash website[1], which grants us an irrevocable, nonexclusive, worldwide copyright license[2] to download, copy, modify, distribute, perform, and use photos from Unsplash for free, including for commercial purposes, without permission from or attributing the photographer or Unsplash.

2. Implementation Details

2.1. Details of Architecture

As mentioned in the main paper, the proposed network is composed of two components: a context-aware local retouching layer (LRL) and an adaptive blend pyramid layer (BPL). Since the architecture of BPL is relatively simple and has been described in detail in the main paper, we mainly provide the details of LRL below. The detailed parameters of an implemented version of LRL are shown in Table 1, in which the size of the original input is 1024×1024 and the number of layers l in BPL is 2 as default.

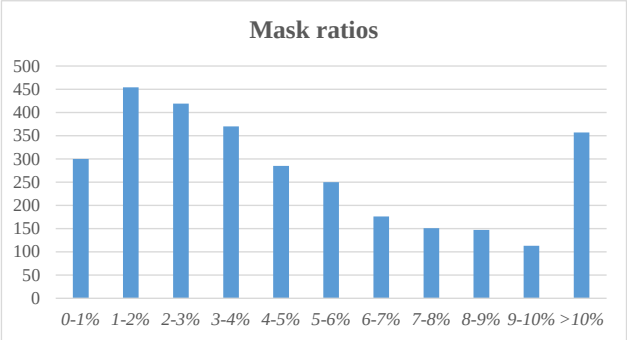


Figure 1. Histograms of retouching mask ratios of the CRHD-3K dataset. We categorize the photos with different retouching region-to-image area ratios. Each category contains at least 110 photos.

2.2. Initialization of Parameters

We initialize the parameters of all convolutions in the model following [1]. Besides, for the learnable parameters j_i and k_i in ABM and R-ABM, we randomly initialize them with the Gaussian distribution. Since the blend layer is sensitive to j_i and k_i, and an improper initialization of j_i and k_i may lead to gradient explosion, we clamp all values in the blend layers into the range [-10, 10] to improve the training stability.

3. Additional Experiments

3.1. Analysis of ABM Parameters

We obtain the values of parameters j_i and k_i of ABM from the models trained on CRHD-3K and FFHQR respectively, and the results are presented in Table 2. We substitute these values for j_i and k_i in Eq. (3) in the main paper, and replace the matrices $\boldsymbol{I}$, $\boldsymbol{B}$, $\boldsymbol{R}$ with the elements x, b, y re-

*This work was done while Xiefan Guo was an intern at the DAMO.
[1]https://unsplash.dogedoge.com
[2]https://unsplash.com/license

spectively, where $x \in [0, 1]$ is the value of an element in $\boldsymbol{I}$, b and y are those of the corresponding elements in $\boldsymbol{B}$ and $\boldsymbol{R}$ respectively. Then, Eq. (3) in the main paper for cloth retouching (CRHD-3K) and face retouching (FFHQR) can be written respectively as:

$$y = x + b(0.19 + 0.31x - 0.24x^2) \quad (1)$$

$$y = x + b(0.16 + 0.21x - 0.20x^2) \quad (2)$$

after which y is clamped into [0, 1] to maintain the same scale with x. As we can see, these two equations have similar patterns, but differ in coefficients. When $b = 0$, y is always equal to x, thereby keeping the corresponding area untouched. When $b > 0$, the target area will be lightened; when $b < 0$, the target area will be darkened. In addition, compared with face retouching, the transformation of the target region in cloth retouching tends to be more drastic, which is also consistent with the coefficients of the two equations above, proving the adaptability of ABM to the LPR task with different data distributions.

3.2. More Visualization Results

More comparison examples on the FFHQR and CRHD-3K datasets are shown in Fig. 5. It can be seen that our method performs favorably against the others.

3.3. Visualization Results of Ultra High-resolution Images

Owing to the excellent extensibility, our method achieves high-quality local retouching on ultra high-resolution images, facilitating the practical applications of deep learning methods in the field of professional photography. We present some 4K retouched results of our method in Fig 6. As we can see, even for the ultra high-resolution images, the proposed method exhibits remarkable performance in terms of global consistency and detail fidelity (see the magnified local regions).

4. Limitations

Considering the limited capacity of the refining module and the operating mechanism of the blend layer, our proposed method cannot well handle the spatial deformation problem. As shown in Fig 7, when the contours of clothing require to be retouched, the performance of our model is somewhat unsatisfactory, arising halo artifacts in the corresponding regions.

5. Negative Impact Concerns

As discussed in the main paper, different from general image editing tasks, local photo retouching aims to enhance the visual aesthetic quality of the objects in photos. Taking the face retouching task shown in this paper as an example, the method is designed to remove blemishes in the face while preserving personally identifiable information. To sum up, instead of tying to a particular application, we focus on the local photo retouching task and propose a general solution to it, which shows no potential negative impact to the society.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *CVPR*, 2015. 1

[2] Jie Liang, Hui Zeng, and Lei Zhang. High-resolution photo-realistic image translation in real-time: A laplacian pyramid translation network. In *CVPR*, 2021. 5

[3] Alireza Shafaei, James J Little, and Mark Schmidt. Autoretouch: Automatic professional face retouching. In *WACV*, 2021. 5

[4] Tamar Rott Shaham, Michaël Gharbi, Richard Zhang, Eli Shechtman, and Tomer Michaeli. Spatially-adaptive pixelwise networks for fast image translation. In *CVPR*, 2021. 5

[5] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 5

[6] Yi Wang, Ying-Cong Chen, Xin Tao, and Jiaya Jia. Vcnet: A robust approach to blind image inpainting. In *ECCV*, 2020. 5

Figure 2. Examples classified by *gender* from the CRHD-3K Dataset (zoom in for a better view). *Top*: female, *bottom*: male.

Figure 3. Examples classified by *type of portrait photo* from the CRHD-3K Dataset (zoom in for a better view). *Top*: full-length photo, *bottom*: half-length photo.

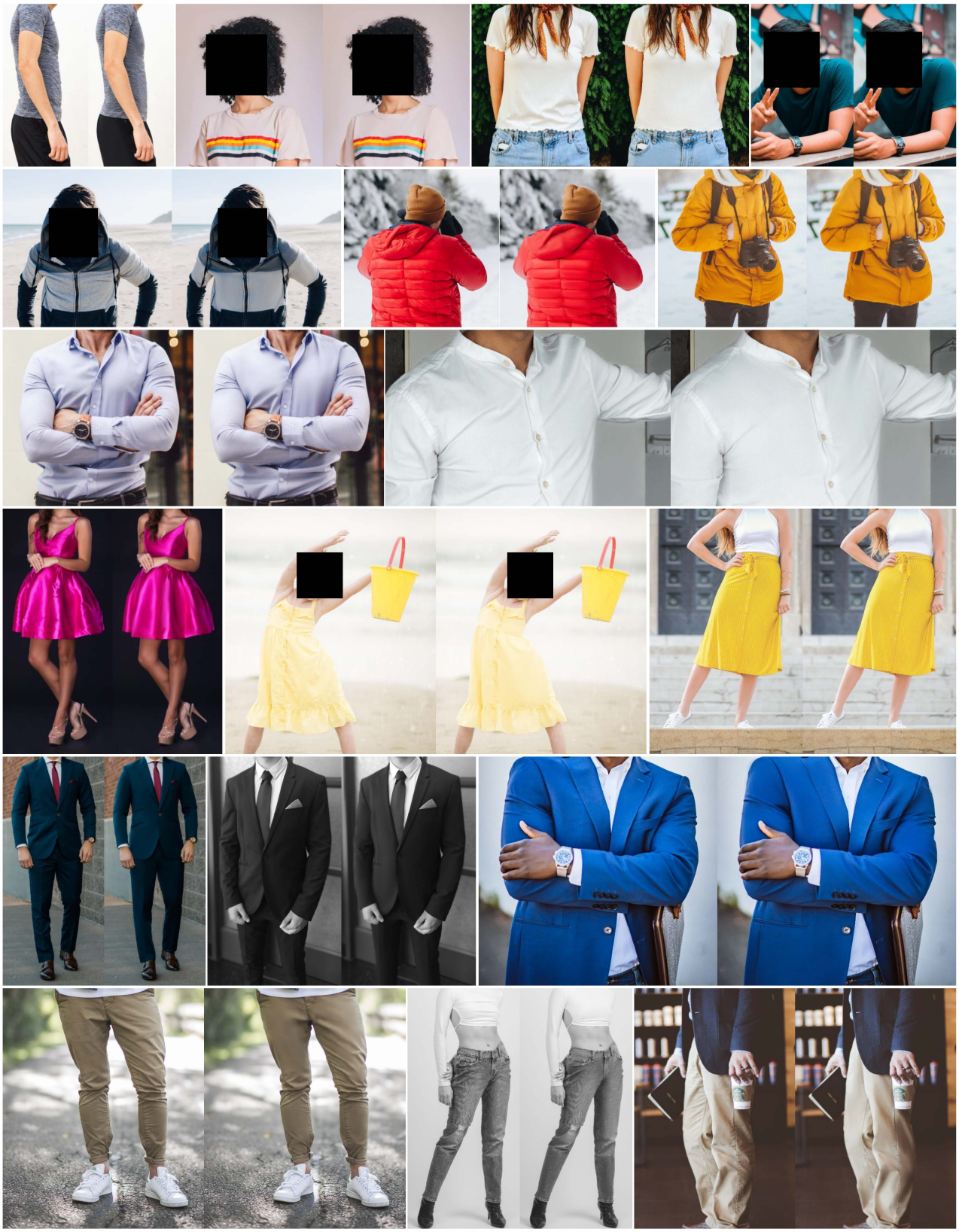
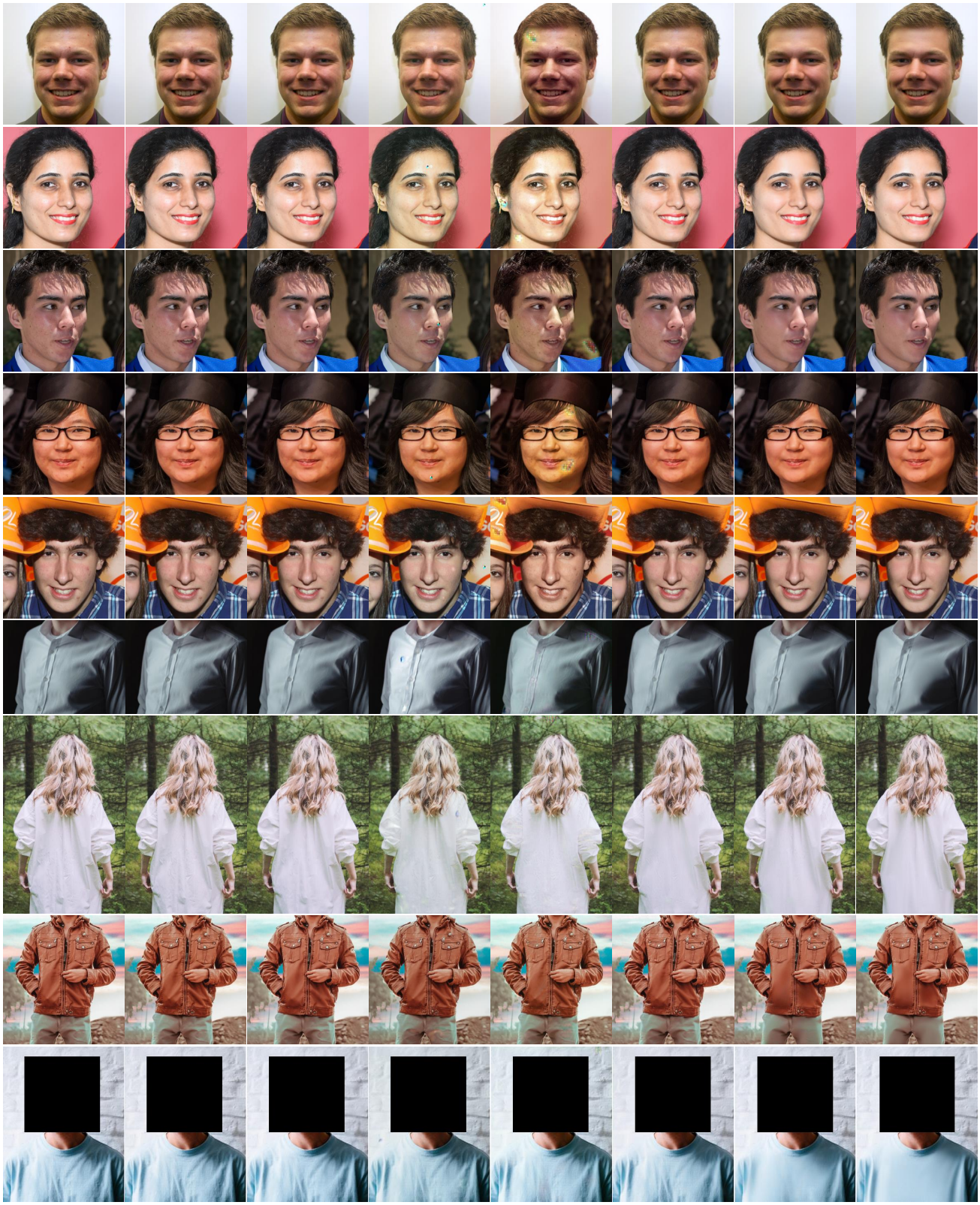


Figure 4. Examples classified by *type of clothing* from the CRHD-3K Dataset (zoom in for a better view) From top to bottom: T-shirt, coat, shirt, skirt, suit and trousers.



(a) Input (b) VCNet (c) AutoRetouch (d) pix2pixHD (e) ASAPNet (f) LPTN (g) Ours (h) Target

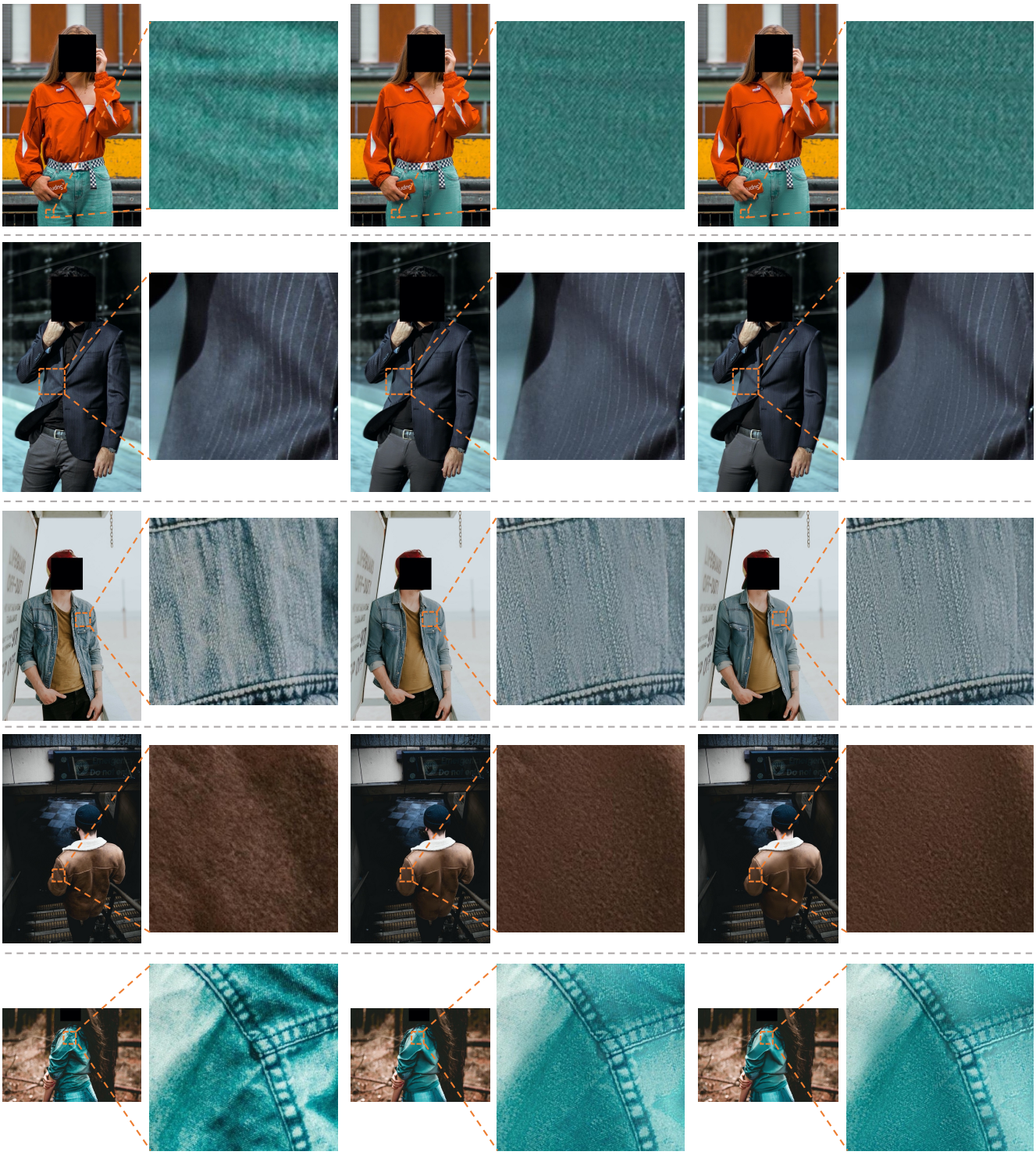
Figure 5. Qualitative comparison on FFHQR and CRHD-3K (zoom in for a better view): (a) original images, (b) VCNet [6], (c) AutoRetouch [3], (d) pix2pixHD [5], (e) ASAPNet [4], (f) LPTN [2], (g) Ours, and (h) ground-truth images.

Module	Block details	Input	Output
Mutual Encoder	Conv(k=3, s=2, p=1), BN, ReLu	(N, 3, 256, 256)	(N, 64, 128, 128)
	Conv(k=3, s=2, p=1), BN, ReLu	(N, 64, 128, 128)	(N, 128, 64, 64)
	Conv(k=3, s=2, p=1), BN, ReLu	(N, 128, 64, 64)	(N, 256, 32, 32)
	Conv(k=3, s=2, p=1), BN, ReLu	(N, 256, 32, 32)	(N, 512, 16, 16)
	Conv(k=3, s=2, p=1), BN, ReLu	(N, 512, 16, 16)	(N, 512, 8, 8)
	Conv(k=3, s=2, p=1), BN, ReLu	(N, 512, 8, 8)	(N, 512, 4, 4)
MPB	Conv(k=3, s=1, p=1), BN, LeakyReLu	(N, 512+512, 8, 8)	(N, 512, 8, 8)
	Conv(k=3, s=1, p=1), BN, LeakyReLu	(N, 512+512, 16, 16)	(N, 512, 16, 16)
	Conv(k=3, s=1, p=1), BN, LeakyReLu	(N, 512+256, 32, 32)	(N, 256, 32, 32)
	Conv(k=3, s=1, p=1), BN, LeakyReLu	(N, 256+128, 64, 64)	(N, 128, 64, 64)
	Conv(k=3, s=1, p=1)	(N, 128, 64, 64)	(N, 1, 64, 64)
LRB	LAM(k=3, s=1, p=1), BN, LeakyReLu	(N, 512+512+1, 8, 8)	(N, 512, 8, 8)
	LAM(k=3, s=1, p=1), BN, LeakyReLu	(N, 512+512+1, 16, 16)	(N, 512, 16, 16)
	LAM(k=3, s=1, p=1), BN, LeakyReLu	(N, 512+256+1, 32, 32)	(N, 256, 32, 32)
	LAM(k=3, s=1, p=1), BN, LeakyReLu	(N, 256+128+1, 64, 64)	(N, 128, 64, 64)
	LAM(k=3, s=1, p=1), BN, LeakyReLu	(N, 128+64+1, 128, 128)	(N, 64, 128, 128))
	LAM(k=3, s=1, p=1)	(N, 64+3+1, 256, 256)	(N, 3, 256, 256)

Table 1. The parameter details for the modules in LRL. Note that the "k", "s", and "p" denote the kernel size, stride, and padding size of the convolutions respectively, "LAM" is the local attentive module proposed in the main paper, the third and fourth columns present the tensor shape of the input and output of each block respectively, where "N" indicates the batch size.

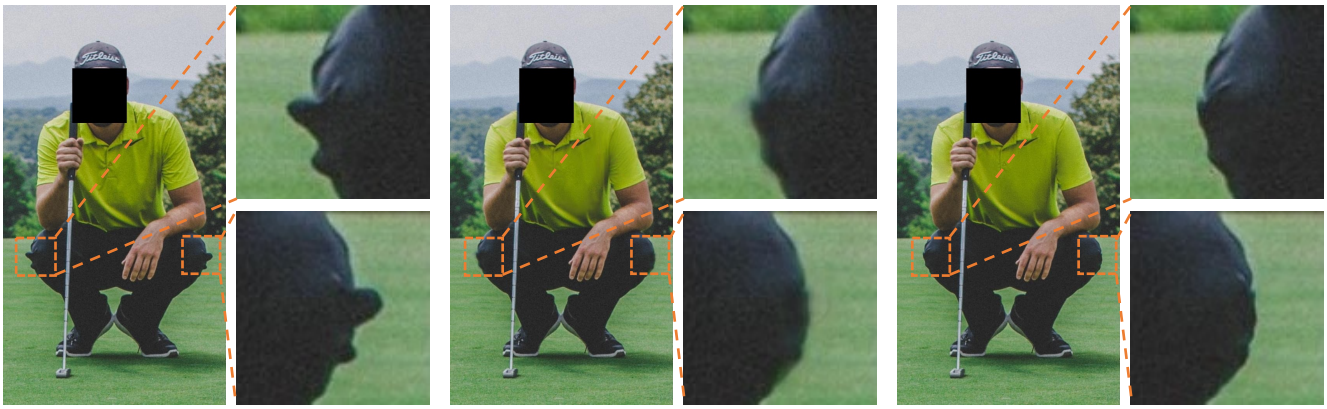
Dataset	j_0	k_0	j_1	k_1	j_2	k_2
CRHD-3K	0.19	$\to 0$	0.31	1.00	-0.24	$\to 0$
FFHQR	0.16	$\to 0$	0.21	1.00	-0.20	$\to 0$

Table 2. The values of parameters j_i and k_i of ABM trained on CRHD-3K and FFHQR.



(a) Input (b) Ours (c) Target

Figure 6. Visualization results of 4K images on CRHD-3K (zoom in for a better view).



(a) Input (b) Ours (c) Target

Figure 7. Visualization result in bad case on CRHD-3K.