# Conditional Sequential Modulation for Efficient Global Image Retouching Supplementary File

Jingwen He$^{\star 1,2}$, Yihao Liu$^{\star 1,2,3}$, Yu Qiao$^{1,2}$, and Chao Dong$^{\dagger 1,2}$

[1] ShenZhen Key Lab of Computer Vision and Pattern Recognition, SIAT-SenseTime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2] SIAT Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society
[3] University of Chinese Academy of Sciences
{jw.he, yh.liu4, yu.qiao, chao.dong}@siat.ac.cn

**Abstract.** In this supplementary file, we first present our analysis on retouching operations (White-balancing, Saturation controlling, and Tone mapping). Then, we conduct some experiments to support our analysis in Section 2. Besides, we provide more visual results of our proposed CSRNet and other state-of-the-art methods. For ablation study, we explore the condition network with different hyper-parameters. Finally, we show more results on smooth transition between multiple styles and strength control for image retouching.

## 1 Analysis on Retouching Operations

In the main paper, we investigate two retouching operations (global brightness change and contrast adjustment) and formulate them into the representaion of MLPs. In this section, we analyze more operations, including white-balancing, saturation controlling and tone-mapping.

**White-balancing.** In [1,3], the operation for white-balancing is described as follows:

$$\begin{aligned} I_R^{'} &= \alpha_R * I_R \\ I_G^{'} &= \alpha_G * I_G \\ I_B^{'} &= \alpha_B * I_B \end{aligned} \tag{1}$$

where $\alpha_R$, $\alpha_G$, $\alpha_B$ are the adjustment scalars for each color channel. The above operation can be represented as an MLP used on singal pixel. Note that the following derivation applies to three channels for each pixel location, therefore, there are totally $3MN$ input units.

$$Y = f(W^T X + b) \tag{2}$$

$^\star$ The first two authors are co-first authors. † Corresponding author

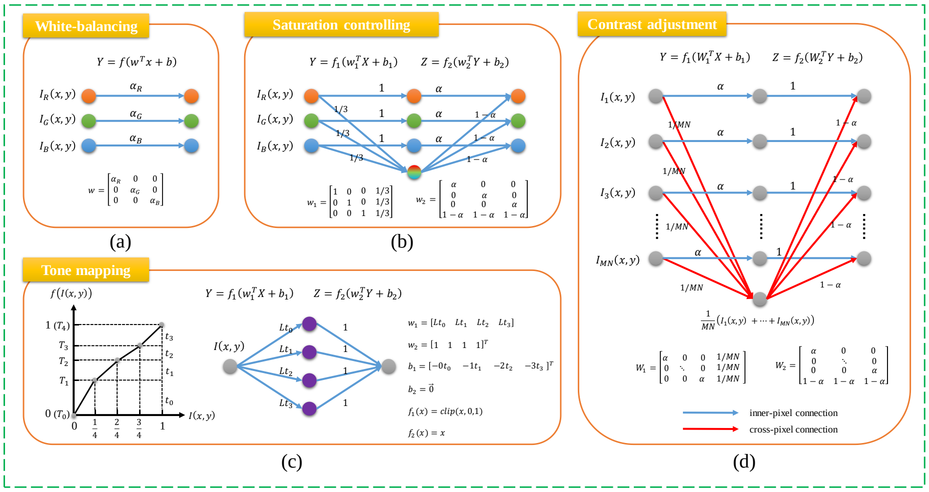

White-balancing
$Y = f(w^T x + b)$
$I_R(x,y)$
$I_G(x,y)$
$I_B(x,y)$
$\alpha_R$
$\alpha_G$
$\alpha_B$
$w = \begin{bmatrix} \alpha_R & 0 & 0 \\ 0 & \alpha_G & 0 \\ 0 & 0 & \alpha_B \end{bmatrix}$
(a)
Saturation controlling
$Y = f_1(w_1^T X + b_1)$
$Z = f_2(w_2^T Y + b_2)$
$I_R(x,y)$
$I_G(x,y)$
$I_B(x,y)$
1
1/3
$\alpha$
$1-\alpha$
$w_1 = \begin{bmatrix} 1 & 0 & 0 & 1/3 \\ 0 & 1 & 0 & 1/3 \\ 0 & 0 & 1 & 1/3 \end{bmatrix}$
$w_2 = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \\ 1-\alpha & 1-\alpha & 1-\alpha \end{bmatrix}$
(b)
Tone mapping
$f(I(x,y))$
$1\ (T_4)$
$T_3$
$T_2$
$T_1$
$0\ (T_0)$
$t_3$
$t_2$
$t_1$
$t_0$
0
$\frac{1}{4}$
$\frac{2}{4}$
$\frac{3}{4}$
1
$I(x,y)$
$Y = f_1(w_1^T X + b_1)$
$Z = f_2(w_2^T Y + b_2)$
$I(x,y)$
$Lt_0$
$Lt_1$
$Lt_2$
$Lt_3$
1
$w_1 = [Lt_0 \quad Lt_1 \quad Lt_2 \quad Lt_3]$
$w_2 = [1 \quad 1 \quad 1 \quad 1]^T$
$b_1 = [-0t_0 \quad -1t_1 \quad -2t_2 \quad -3t_3\ ]^T$
$b_2 = \vec{0}$
$f_1(x) = clip(x, 0, 1)$
$f_2(x) = x$
(c)
Contrast adjustment
$Y = f_1(W_1^T X + b_1)$
$Z = f_2(W_2^T Y + b_2)$
$I_1(x,y)$
$I_2(x,y)$
$I_3(x,y)$
$I_{MN}(x,y)$
$\alpha$
1
$1/MN$
$1-\alpha$
$\frac{1}{MN}(I_1(x,y) + \cdots + I_{MN}(x,y))$
$W_1 = \begin{bmatrix} \alpha & 0 & 0 & 1/MN \\ 0 & \ddots & 0 & 1/MN \\ 0 & 0 & \alpha & 1/MN \end{bmatrix}$
$W_2 = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \alpha \\ 1-\alpha & 1-\alpha & 1-\alpha \end{bmatrix}$
inner-pixel connection
cross-pixel connection
(d)

Fig. 1: Illustration of the equivalent MLPs for the corresponding retouching operations. Commonly-used retouching operations can be regarded as classic MLPs used on input image. Moreover, operations like white-balancing (a), saturation controlling (b) and tone-mapping (c), can further regarded as MLPs used on individual pixels, since these operations are pixel-independent and only contain inner-pixel connections. However, operations like contrast adjustment require global information and contain cross-pixel connections (d). The condition network can collaborate with the base network, providing global features and facilitating cross-pixel connections. For simplicity, the illustrations for tone-mapping and contrast adjustment only show the case of single channel.

where $X \in \mathbb{R}^{3MN}$ is the vector flattened from the input image, $W \in \mathbb{R}^{3MN \times 3MN}$ and $b \in \mathbb{R}^{3MN}$ are weights and biases, and $f(.)$ is the activation function. Let $w = diag\{\alpha_R, \alpha_G, \alpha_B\} \in \mathbb{R}^{3\times 3}$.

When $W = diag\{w, w, \dots, w\} = \begin{bmatrix} w & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_w \end{bmatrix} \in \mathbb{R}^{3MN \times 3MN}$ , $b = \mathbf{0}$ and $f(x) = x$, the above MLP (2) is equivalent to the white-balancing operation (1), as shown in Figure 1(a).

**Saturation controlling.** Saturation describes the purity of the color. In [3], the operation for controlling saturation is as follows:

$$I^{'}(x,y) = \alpha I(x,y) + (1-\alpha)\overline{I}_{RGB}(x,y) \tag{3}$$

where $\overline{I}_{RGB}(x,y) = \frac{1}{3}[I_R(x,y) + I_G(x,y) + I_B(x,y)]$ is the cross-channel average of the pixel on location $(x,y)$, and $I_R$, $I_G$, $I_B$ represent the RGB channels, respectively. The saturation adjustments operation above can be modeled in an MLP with $3MN$, $4MN$ and $3MN$ units in each layer. There are three channels

(RGB) for each pixel location, thus, there are totally $3MN$ input units.

$$\begin{aligned} Y &= f_1(W_1^T X + b_1) \\ Z &= f_2(W_2^T Y + b_2) \end{aligned} \tag{4}$$

where $X \in \mathbb{R}^{3MN}$ is the input vector, $W_1 \in \mathbb{R}^{3MN \times 4MN}$, $W_2 \in \mathbb{R}^{4MN \times 3MN}$ are the weight matrices, $b_1 \in \mathbb{R}^{4MN}$, $b_2 \in \mathbb{R}^{3MN}$ are the bias vectors, and $f_1(.)$, $f_2(.)$ are the activation functions. Let $w_1 = \begin{bmatrix} 1\,0\,0\,1/3 \\ 0\,1\,0\,1/3 \\ 0\,0\,1\,1/3 \end{bmatrix}$, $w_2 = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \\ 1-\alpha & 1-\alpha & 1-\alpha \end{bmatrix}$.

When $W_1 = diag\{w_1, w_1, \ldots, w_1\} = \begin{bmatrix} w_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_1 \end{bmatrix} \in \mathbb{R}^{3MN \times 4MN}$,

$W_2 = diag\{w_2, w_2, \ldots, w_2\} = \begin{bmatrix} w_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_2 \end{bmatrix} \in \mathbb{R}^{4MN \times 3MN}$, $b_1 = b_2 = \mathbf{0}$ and $f_1(x) = f_2(x) = x$, the MLP (4) is equivalent to formula (3). (see Figure 1(b))

**Color curve adjustment/Tone mapping.** Color curve is a channel-independent monotonic and piecewise-linear mapping function (Figure 1(c) Left). Suppose the curve is uniformly divided into $L$ pieces and the curve can be represented by the end points on the it: $\{(\frac{i}{L}, T_i) \mid i = 0, 1, \ldots, L\}$, as shown in Figure xxx. The height of each piece is $t_i = T_{i+1} - T_i, i = 0, 1, \ldots, L-1, T_0 = 0$. Then, an input pixel $I(x, y) \in [0, 1]$ is mapped to

$$f(I(x,y)) = \sum_{i=0}^{L-1} clip(L * I(x,y) - i, 0, 1) t_i \tag{5}$$

Above mapping is channel-independent and the operation is equivalent to an MLP. *For simplicity we only consider the case of a single channel.* We construct a three-layer MLP, in which the first, second and third layer contains $M \times N$, $L \times M \times N$ and $M \times N$ units, respectively.

$$\begin{aligned} Y &= f_1(W_1^T X + b_1) \\ Z &= f_2(W_2^T Y + b_2) \end{aligned} \tag{6}$$

where $X \in \mathbb{R}^{MN}$ is the input vector, $W_1 \in \mathbb{R}^{MN \times LMN}$, $W_2 \in \mathbb{R}^{LMN \times MN}$ are weight matrices, $b_1 \in \mathbb{R}^{LMN}$, $b_2 \in \mathbb{R}^{MN}$ are bias vectors, and $f_1(.)$, $f_2(.)$ are the activation functions. Let $E_{ij}$ be the basic matrix with only a one on the position $(i, j)$ and zeros elsewhere. For example, if $E \in \mathbb{R}^{2\times 3}$ then $E_{12} = \begin{bmatrix} 0\,1\,0 \\ 0\,0\,0 \end{bmatrix}$. Let $w_{1,i}, E \in \mathbb{R}^{MN \times L}$ and $w_{1,i} = Lt_i(E_{i1} + E_{i2} + \cdots + E_{iL})$, for $i = 1, 2, \ldots, MN$. Let $w_{2,i}, E \in \mathbb{R}^{L \times MN}$ and $w_{2,i} = E_{1i} + E_{2i} + \cdots + E_{Li}$, for $i = 1, 2, \ldots, MN$. Let $B = [0, -t_1, \ldots, -it_i, \ldots, -(L-1)t_{L-1}]^T$.

When $W_1 = \left[w_{1,1}, w_{1,2}, \cdots, w_{1,MN}\right]$, $b_1 = \left[B, B, \cdots, B\right]^T$ which contains $L$ stacked vector $B$, $W_2 = \left[w_{2,1}, w_{2,2}, \cdots, w_{2,MN}\right]^T$, $b_2 = \mathbf{0}$, $f_1(x) = clip(x, 0, 1)$ and $f_2(x) = x$, the above MLP (6) is equivalent to the tone mapping formula (5). Given an example, for $M = N = 2$ and $L = 2$, there should be $W_1 =$

$$\begin{bmatrix} Lt_0 & Lt_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Lt_0 & Lt_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Lt_0 & Lt_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Lt_0 & Lt_1 \end{bmatrix}, W_2 = \begin{bmatrix} 1\,0\,0\,0 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,0\,1 \end{bmatrix}, b_1 = \begin{bmatrix} 0 \\ -t_1 \\ 0 \\ -t_1 \\ 0 \\ -t_1 \\ 0 \\ -t_1 \end{bmatrix}, b_2 = \mathbf{0}.$$

**Discussions.** So far, we have shown that most commonly-used retouching operations can be formulated as classic MLPs used on the input image. These operations are pixel-independent or location-independent; that is to say, the manipulation on one pixel is uncorrelated with neighboring pixels or pixels on specific positions. Further, operations like brightness change, white-balancing, saturation controlling, tone-mapping, can be also viewed as MLPs used ***on a single pixel***, which is similar with the MLPconv proposed in [2]. Enlightened by this discovery, the base network in the proposed method is designed as a fully convolutional network with all the filter size of 1 × 1, which acts like an MLP worked on individual pixels and slides over the input image. Some operations, like contrast adjustment, may require global information that relates to all pixels in the image (e.g., image mean value). Such global information can be provided by the condition network in our method.

## 2 Demonstration Experiments on The Proposed Method

To support the analysis above, we use the proposed network to simulate the procedures of several retouching operations, including global brightness change, tone-mapping and contrast adjustment. Specifically, we adopt images retouched by expert C as inputs and apply retouching operations with specified adjustment coefficients on the inputs as supervision labels. Then we utilize the base network and the proposed CSRNet to learn such mappings.

Theoretically, the base network can perfectly handle operations like global brightness change and tone-mapping, because these pixel-independent operations are equivalent to MLPs used on individual pixels. For contrast adjustment, only the base network should not be enough, since it cannot extract global information like image mean value.

The results are shown in Table 1. As expected, the base network can successfully deal with the pixel-independent operations [4]. Nevertheless, we observe that a sole base network is unable to handle contrast adjustment, which requires

[4] Images are basically the same when PSNR > 50dB.

Table 1: Demonstration experiment on simulating retouching operations. Our method can successfully handle commonly-used retouching opereations, which is consistent with the theoretical analysis.

| Operations | original (Input-GT) | base netwok | condition netwok | PSNR |
|---|---|---|---|---|
| brightness | ✓ | × | × | 14.7413 |
| ($\alpha = 1.5$) | × | ✓ | × | **69.7061** |
| brightness | ✓ | × | × | 12.8460 |
| ($\alpha = 0.5$) | × | ✓ | × | **69.0525** |
| tone-mapping* | ✓ | × | × | 21.7580 |
| ($L = 4$) | × | ✓ | × | **56.1175** |
| contrast ($\alpha = 1.5$) | ✓ | × | × | 21.3584 |
| | × | ✓ | × | 28.6734 |
| | × | ✓ | ✓ | **60.5206** |

* The parameters for tone-mapping are set to $t_i = [3/8, 2/8, 1/8, 2/8]$.

global information. We can solve this problem by introducing the condition network. As we can see, the PSNR rises from 28dB to 60dB, demonstrating the effectiveness of the proposed method.

## 3 Visual comparison

In visual comparison, we observe that the results obtained by Pix2Pix are quite noisy. However, it might not be obvious when the images are downsized. Here, we specially provide some Pix2Pix examples for better visualization in Figure 2.

Besides, we provide more visual results of our proposed methods and other state-of-the-art methods in Figrue 3, 4 and some failure cases in Figure 5.

## 4 Ablation study

**Condition network.** The condition network aims to estimate a condition vector that represents global information of the input image. Here, we explore the condition network with different hyper-parameters. Specifically, we change the number of layers or increase the channel size in each convolutional layer. First, we fix the number of layers to 3 and change the channel size to 64 and 128. From Table 2, we can observe that larger channel size leads to higher PSNR but require much more parameters. Then, we change the number of layers by adding convolutional layers with kernel size $3 \times 3$ and stride 1 or remove the existing layers. However, there is no improvement with more layers in the condition network. Therefore, the extraction of global features are already well achieved by a shallow network.

Table 2: Results of ablation study for the condition network.

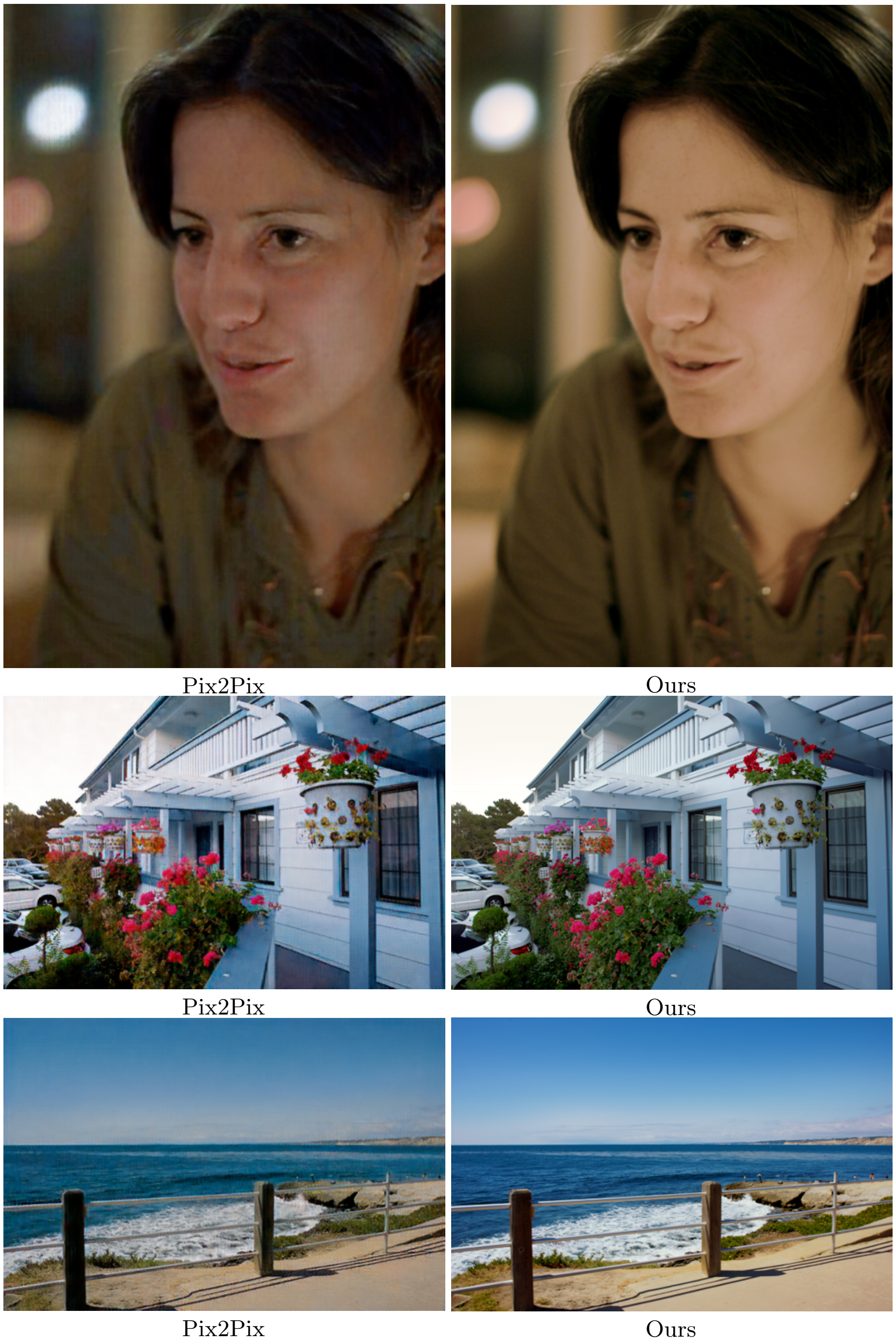
| layers | channel | PSNR | params |
|---|---|---|---|
| 3 | 32 | 23.69 | 36,489 (ours) |
| 3 | 64 | 23.73 | 104,969 |
| 3 | 128 | 23.81 | 352,521 |
| 2 | 32 | 23.48 | 27,241 |
| 3 | 32 | 23.69 | 36,489 (ours) |
| 5 | 32 | 23.65 | 54,985 |
| 7 | 32 | 23.62 | 73,481 |
| 5 | 64 | 23.67 | 178,825 |

## 5 Multiple Styles and Strength Control

In this section, we present more results about smooth transition between multiple styles (see Figure 6) and strength control (see Figure 7) on image retouching achieved by image interpolation.

## References

1. Hu, Y., He, H., Xu, C., Wang, B., Lin, S.: Exposure: A white-box photo post-processing framework. ACM Transactions on Graphics (TOG) **37**(2), 1–17 (2018)
2. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
3. Park, J., Lee, J.Y., Yoo, D., So Kweon, I.: Distort-and-recover: Color enhancement using deep reinforcement learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5928–5936 (2018)

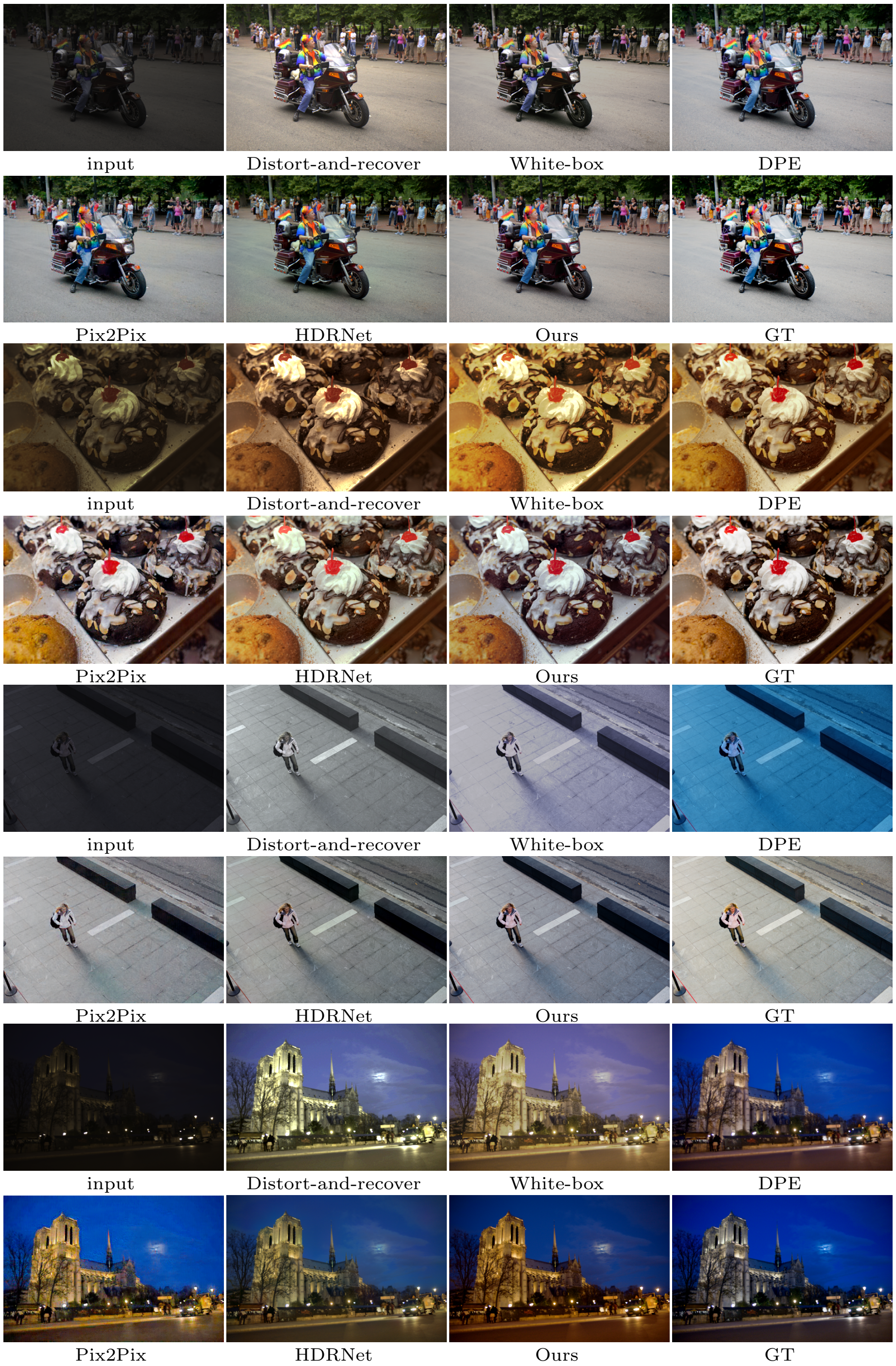


Pix2Pix Ours

Pix2Pix Ours

Pix2Pix Ours

Fig. 2: Artifacts in Pix2Pix.



input Distort-and-recover White-box DPE

Pix2Pix HDRNet Ours GT

input Distort-and-recover White-box DPE

Pix2Pix HDRNet Ours GT

input Distort-and-recover White-box DPE

Pix2Pix HDRNet Ours GT

input Distort-and-recover White-box DPE

Pix2Pix HDRNet Ours GT

Fig. 3: Visual comparision with state-of-the-arts (a).

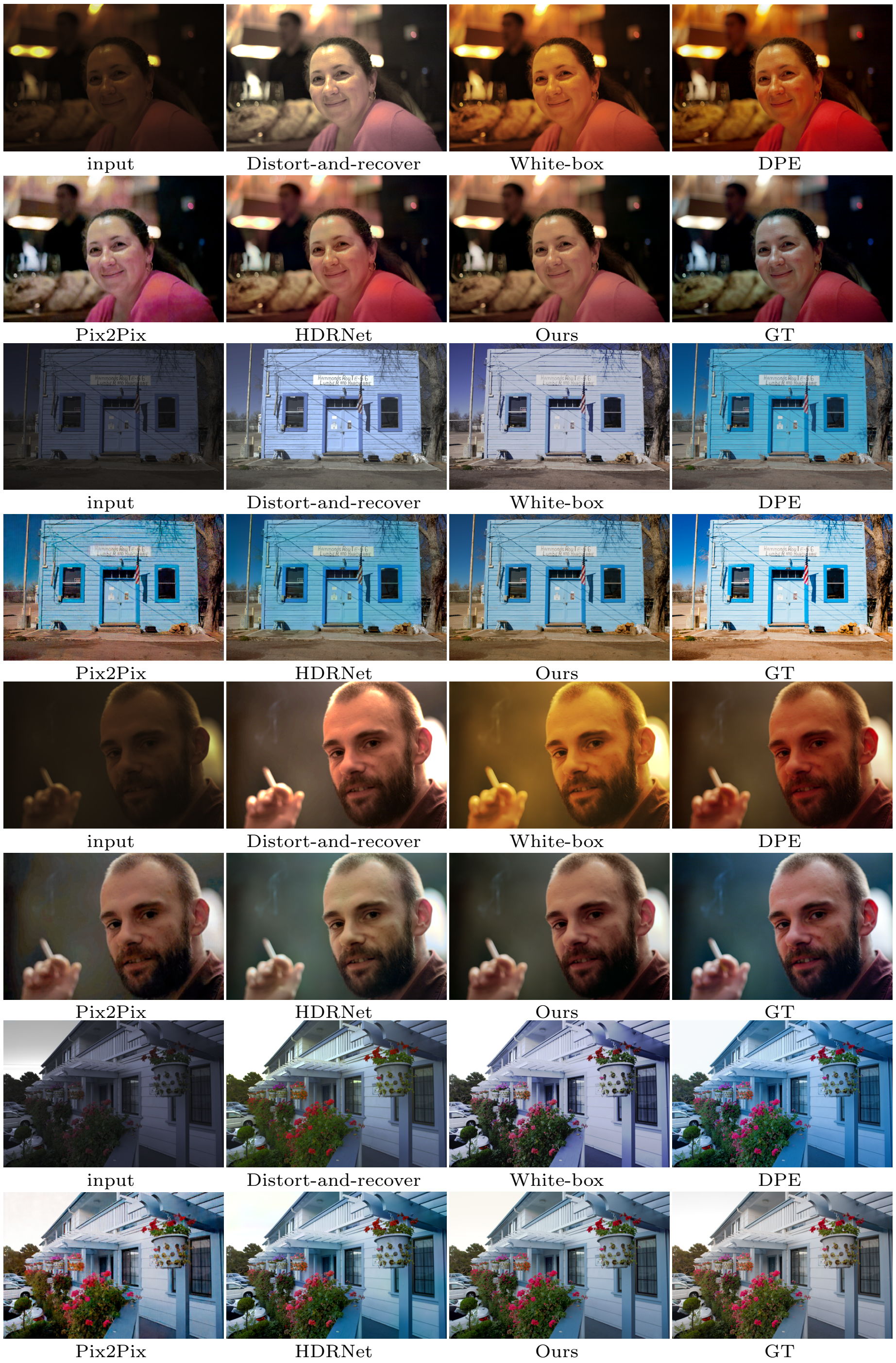


Fig. 4: Visual comparision with state-of-the-arts (b).

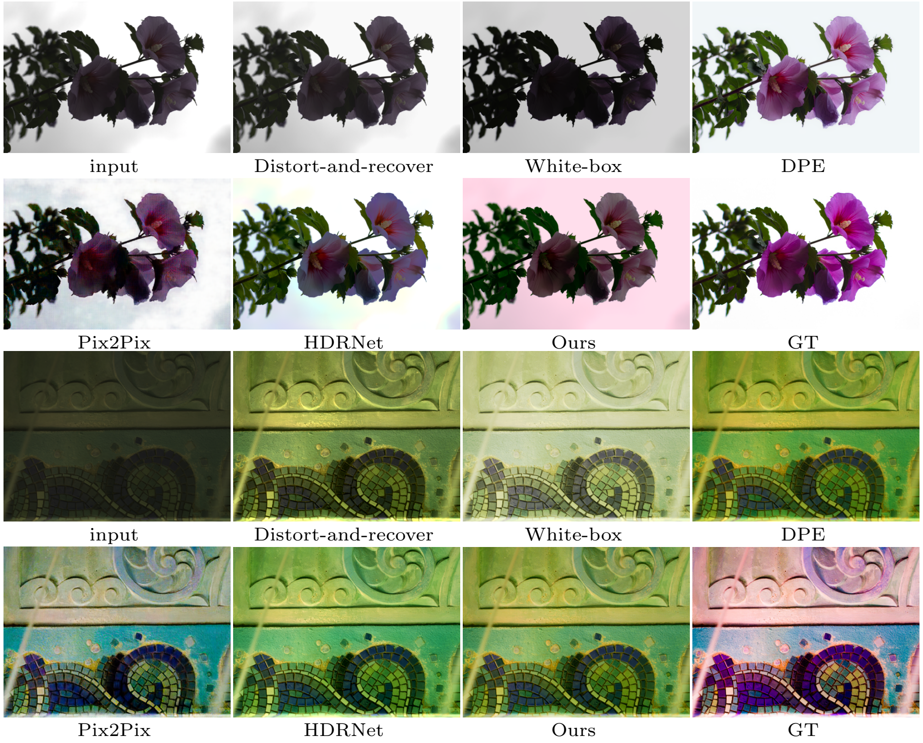
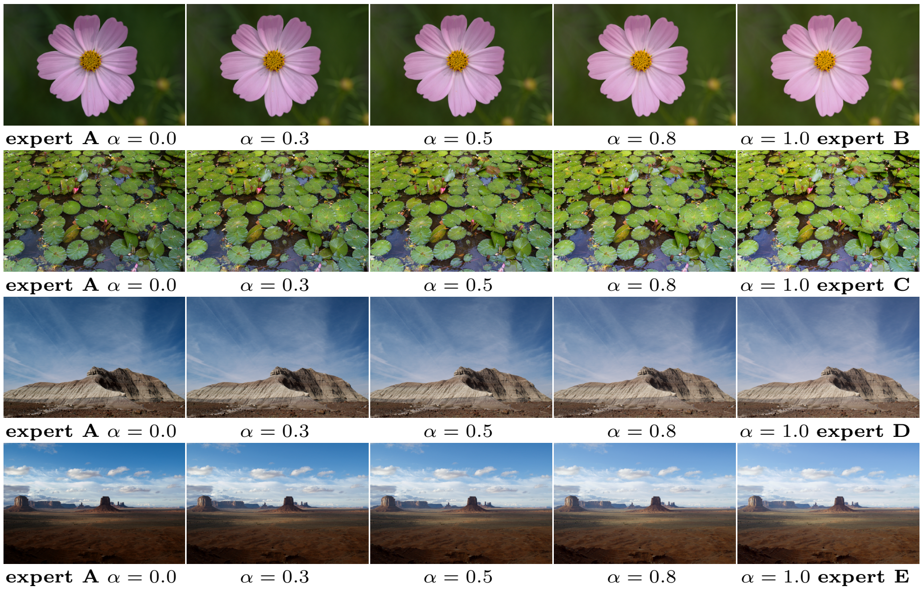


Fig. 5: Failure cases. For the first input, our method outputs a pink sky, which is supposed to be white. For the second input, our method is unable to change the original green tone.



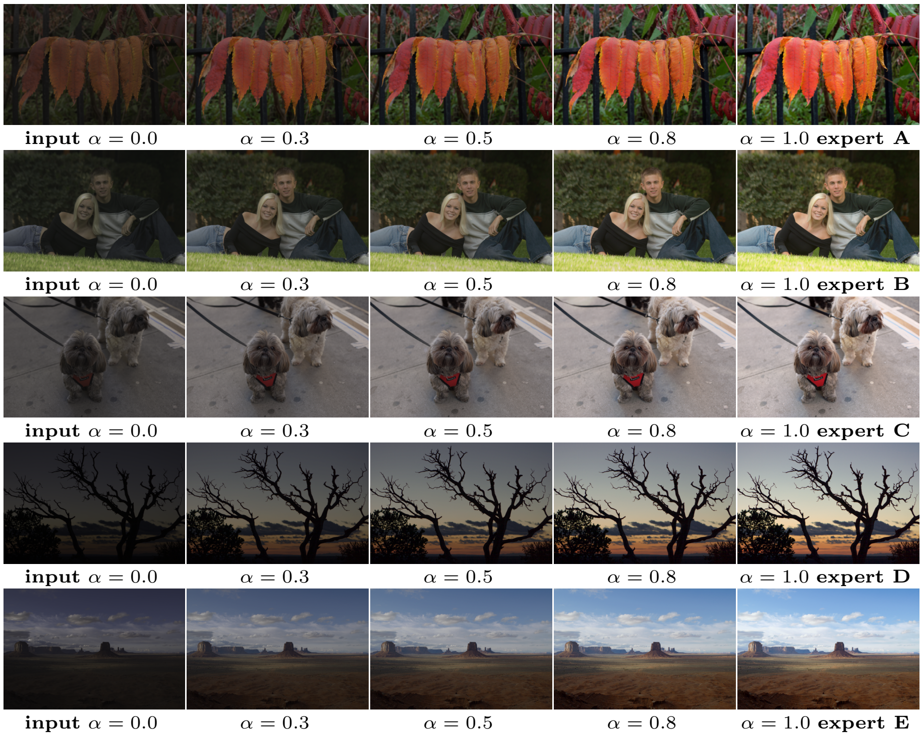
**expert A** $\alpha = 0.0$ | $\alpha = 0.3$ | $\alpha = 0.5$ | $\alpha = 0.8$ | $\alpha = 1.0$ **expert B**

**expert A** $\alpha = 0.0$ | $\alpha = 0.3$ | $\alpha = 0.5$ | $\alpha = 0.8$ | $\alpha = 1.0$ **expert C**

**expert A** $\alpha = 0.0$ | $\alpha = 0.3$ | $\alpha = 0.5$ | $\alpha = 0.8$ | $\alpha = 1.0$ **expert D**

**expert A** $\alpha = 0.0$ | $\alpha = 0.3$ | $\alpha = 0.5$ | $\alpha = 0.8$ | $\alpha = 1.0$ **expert E**

Fig. 6: Image interpolation between different styles.



**input** $\alpha = 0.0$ | $\alpha = 0.3$ | $\alpha = 0.5$ | $\alpha = 0.8$ | $\alpha = 1.0$ **expert A**

**input** $\alpha = 0.0$ | $\alpha = 0.3$ | $\alpha = 0.5$ | $\alpha = 0.8$ | $\alpha = 1.0$ **expert B**

**input** $\alpha = 0.0$ | $\alpha = 0.3$ | $\alpha = 0.5$ | $\alpha = 0.8$ | $\alpha = 1.0$ **expert C**

**input** $\alpha = 0.0$ | $\alpha = 0.3$ | $\alpha = 0.5$ | $\alpha = 0.8$ | $\alpha = 1.0$ **expert D**

**input** $\alpha = 0.0$ | $\alpha = 0.3$ | $\alpha = 0.5$ | $\alpha = 0.8$ | $\alpha = 1.0$ **expert E**

Fig. 7: Image interpolation for strength control.