Color Enhancement using Global Parameters and Local Features Learning

Enyu Liu, Songnan Li, and Shan Liu

Tencent Media Lab, Shenzhen, China
{enyuliu, sunnysnli, shanl}@tencent.com

Abstract. This paper proposes a neural network to learn global parameters and extract local features for color enhancement. Firstly, the global parameters extractor subnetwork with dilated convolution is used to estimate a global color transformation matrix. The introduction of the dilated convolution enhances the ability to aggregate spatial information. Secondly, the local features extractor subnetwork with a light dense block structure is designed to learn the matrix of local details. Finally, an enhancement map is obtained by multiplying these two matrices. A novel combination of loss functions is formulated to make the color of the generated image more consistent with that of the target. The enhanced image is formed by adding the original image with an enhancement map. Thus, we make it possible to adjust the enhancement intensity by multiplying the enhancement map with a weighting coefficient. We conduct experiments on the MIT-Adobe FiveK benchmark, and our algorithm generates superior performance compared with the state-of-the-art methods on images and videos, both qualitatively and quantitatively.

1 Introduction

Color enhancement boosts the picture quality of an image or video by adjusting the global contrast, intensifying the local details, and generating more vivid colors. Due to low light, poor shooting equipment, bad weather conditions and other factors, the colors of images and videos may fade and distort, which seriously affect the visual quality. Using image processing software such as photoshop, many people like to manually modify image color to subjectively adjust the contrast, brightness, saturation, hue, exposure and etc. Different image contents need specific operations for maximize its visual quality. If all images are processed individually, it is time-consuming and the color coherence between adjacent video frames would be disrupted. For these reasons, color enhancement for both images and videos is a challenging and popular task in computer vision.

On the other hand, in the literature many algorithms have been developed to enhance colors automatically. The traditional algorithms [1][2] enhanced images from the spatial domain or frequency domain by exploiting techniques, such as histogram equalization, retinex, and wavelet multi-scale transform. However, these methods cannot enhance image details effectively. Deep learning based methods are widely investigated for this task in recent years [3][4], which can achieve better performance compared with traditional methods. There are two mainstream methods: one is to learn the local features; the other is to estimate the global transformation parameters. For the method of learning local features, it is indeed effective to obtain good effects, but improper network design may cause flickering problems in videos, so postprocessing is needed to keep the brightness consistency between video frames. While for the method of estimating global parameters, compared with the method of learning local features, it can better maintain the consistency in video processing, but a global matrix may cause the inadequate enhancement of the local contrast, making the picture details in dark and bright regions less visible.

Therefore, we propose a novel deep model for tackling these problems, which obtains superior color enhancement effects by combining the global parameters and local features. The design of our network considers its practical applicability. The input can be of arbitrary resolution and the color enhancement intensity can be adjusted conveniently by tuning a single parameter. Estimating a global color transformation matrix as in [5] is appealing for its computational efficiency and the ability to coherently enhance the global contrast of adjacent video frames. And image details are also considered by learning local features.

1. In order to extract global parameters, a multi-branch network with multi-scale dilated convolution layers is proposed to enlarge the receptive field so as to get a more accurate global transformation matrix.

2. For extracting local features, a network with a light dense block is employed, which can strengthen the feature propagation and enhance the local contrast.

3. Considering both structure and color similarity, we propose a novel combination of loss functions, including a L2 loss in CIElab color space, a structure similarity (SSIM) loss, and an improved color loss.

Our algorithm can enhance the picture color and contrast both globally and locally. We conduct subjective and objective experiments to prove its superiority over the state-of-art methods. Furthermore, the proposed method can be applied on video, since it can effectively avoid flicker and maintain the color consistency of adjacent video frames.

2 Related Work

For color enhancement, many algorithms have been proposed to improve brightness, contrast and saturation. Converting to HSV color space is one of the effective approaches [6][7]. HSV represents hue, saturation and brightness respectively. The HSV space can subtly separate each feature, so it is more convenient to process each factor and obtain good enhancement results. However, a global enhancement may cause the loss of local details.

Example-based enhancement methods are also widely investigated. Wang et al. [8] proposed a method to discover mathematical relationships and to learn the nonlinear mappings of color and tone from a pair of corresponding images before and after adjustments. Lee et al. [9] presented an unsupervised algorithm by selecting several styles from subsets and transforming the color of the input into the style and tone of selected exemplars. However, selecting suitable example images and learning the mapping between inputs and example images are tricky and difficult problems.

The generative adversarial networks (GAN) based approach is able to achieve good results [4]. Hui et al. [10] designed lightweight local residual networks that can be applied on smartphone cameras. Specifically, transformation of teacher-student information is introduced to maintain the realism of the outputs. Chen et al. [11] proposed a global feature network, which used Wasserstein GAN (WGAN) to speed up the convergence. Two-way GANs exploited individual batch normalization layers for each generator which can adjust to their own distributions easily. Chai et al. [5] trained an unsupervised model which applied the cycle consistency of CycleGAN [12] and also employed individual batch normalization layers.

Deep reinforcement learning can be applied for this task as well. Park et al. [13] used reinforcement learning to give a guidance of several image enhancement operations, and a distort-recover scheme was proposed for unpaired images training. The advantage of these methods is that the unsupervised learning uses unpaired images for training which are more accessible than paired images. However, it is hard to converge, and is subject to color distortions.

Recently, investigation of using supervised CNN for color enhancement are thriving. Since training on matched pairs is more accurate and controllable, our algorithm also uses paired images for training. Chen et al. [14] proposed to use dilated convolution and individual batch normalization on a fully convolutional network to learn the mapping from the input to the output. Ignatov et al. [15] collected a dataset that consists of real photos captured from different cameras and trained an end-to-end network to generate the enhanced images for mobile devices. Huang et al. [3] proposed a range scaling layer based on UNet [16] to extract features at different resolutions so as to reduce the output artifacts. Isola et al. [4] also used UNet [16] to learn the mapping between the target and the input.

Rather than directly generating an enhanced image, there are approaches that use deep neural networks to predict intermediate color transformation matrices, which then are applied on the input image to produce the output [5][17]. Gharbi et al. [17] introduced the HDRNet which predicted the intermediate color transformation matrices in a low resolution bilateral grid, and then used bilateral slicing to up-sample the grid into the orginal resolution. Bianco et al. [18] proposed to predict the coefficients of local polynomial transformations that are applied to the input image to remove artistic photo filter. Based on previous research, Bianco et al. [19] later designed the novel CNN that estimates the coefficients to be used to the basis function (including polynomial, piecewise, cosine and radial) to get the color transformation. Afifi et al. [20] trained several pairs of incorrect white-balanced images and computed nonlinear color correction matrices that map the incorrect colors to the target, and finally their corresponding correctly white-balanced images are obtained by weighting multiple effects. Wang et al. [21] developed a network to predict a mapping between the input image and its illumination, and the input was combined with the illumination map for color enhancement. This method can successfully cover different photographic adjustments. Biano et al. [22] proposed to use two neural networks. One for a global tone enhancement, while the other for local adjustment to spatially align the input and the ground truth. Maron et al. [23] introduced a novel approach to automatically enhance images using learned parameters of three spatial filters. Chai et al. [5] presented a supervised model that used the CNN to predict color transformation parameters. And then the parameters are multiplied by the color basis vector for each pixel to get the enhanced RGB value.

3 Proposed Method

3.1 Overview

We develop our method based on [5] with modifications to enhance the output both globally and locally. Specifically, an enhancement map M is calculated by our neural network, and then the enhancement map is added on the input image to obtain the enhanced image. The process can be simply defined as:

$$O = M + I \tag{1}$$

where I is the input with RGB channels, O is the output, M is the enhancement map, $M \in \mathbb{R}^{w \times h \times 3}$, w and h are the width and height of the image. Furthermore, by using Eq. (2) which multiply M by a weighting factor, we can easily control the color enhancement intensity:

$$O = \alpha \cdot M + I \tag{2}$$

where α is the weighting coefficient. If $\alpha > 1$, the enhancement strength would be enlarged; if $0 < \alpha < 1$, the strength would be reduced. This coefficient can be adjusted according to different scenes or videos. In this paper, experiments are performed by setting $\alpha = 1$, which shows the original intensity learned from the training set. The calculation of the enhancement map consists of two parts. One is to estimate the matrix of the global parameters from the global parameters extractor subnetwork, which is defined as P. The input of P is I_{ds},which is a downsampled version of the input image I. The output of P is a global color transformation matrix $\theta \in \mathbb{R}^{10 \times 3}$, so the inference process of P can be expressed as:

$$\theta_P = P(I_{ds}) \tag{3}$$

The structure of the global parameters extractor subnetwork P is elaborated in Section 3.2. The other part is to estimate the local features from the local features extractor subnetwork. We use F as the notation for this subnetwork. The input of F is defined as $I_F \in \mathbb{R}^{w \times h \times 10}$, which contains the quadratic color basis vectors of I. $p_F(x, y)$ is the quadratic color basis vector at coordinate (x, y) in I_F, and $p(x, y)$ represents RGB values at coordinate (x, y) in I, i.e., $p(x, y) = [R(x, y), G(x, y), B(x, y)]$. Thus, $p_F(x, y)$ can be defined as:

$$
\begin{aligned}
p_F(x, y) = [&R(x, y), G(x, y), B(x, y), R(x, y)^2, G(x, y)^2, B(x, y)^2, \\
&R(x, y) \cdot G(x, y), G(x, y) \cdot B(x, y), B(x, y) \cdot R(x, y), 1]
\end{aligned} \tag{4}
$$

which is also used in [5][18], in order to make it easy to preserve the details in the input [18]. The output of the local feature extractor subnetwork F is $\theta_F \in \mathbb{R}^{w \times h \times 10}$, which has the same size as its input. The inference process of F can be expressed as:

$$\theta_F = F(I_F) \tag{5}$$

We elaborate on the structure of the local feature extractor subnetwork in Section 3.3. Finally, the enhancement map M is defined as:

$$M = \theta_P \theta_F = F(I_F)P(I_{ds}) \tag{6}$$

In order to make the structure and color of the output as close as possible to the ground truth, we propose a loss function which consists of three loss terms as to be described in Section 3.4. The overall architecture of our algorithm is shown in Fig. 1.

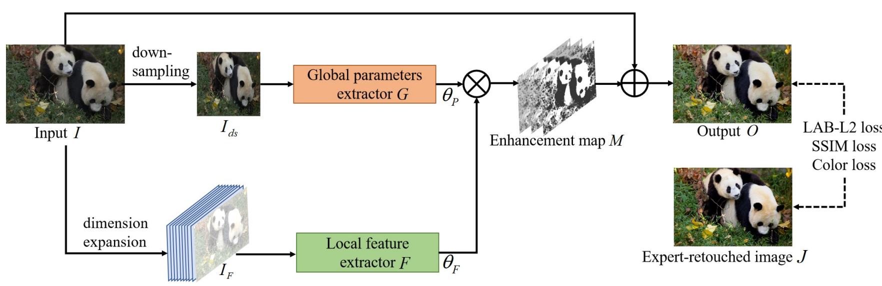


Fig. 1. The architecture of our color enhancement network.

3.2 Global Parameters Extractor Subnetwork

To extract the global parameters, we augment the multi-branch network of [5] with dilated convolution as shown in Fig. 2. The whole global parameters extractor subnetwork consists of five branches, and each branch has an identical structure.

Each branch consists of four parts, including feature extraction, context aggregation, feature extraction and parameter compression. Firstly, the input image is downsampled to $256 \times 256 \times 3$. The first layer consists of 5×5 filtering followed by Leaky Relu, and the second and third layers use 3×3 filter with stride

2. Every subsequent convolutional layer is followed by Batch Normalization and Leaky Relu.

Next, the $32 \times 32 \times 48$ feature map are expanded to $32 \times 32 \times 96$ after the context aggregation. Three dilated convolution layers are added in the middle of the architecture with a kernal size of 3×3, and the dilation rates of 1, 2 and 3 separately. The special combination of three dilation rates has been proved to be helpful and defined as the hybrid dilated convolution (HDC) framework [24]. It can amplify the receptive fields of the network, aggregate global information, and decrease the gridding issue produced by the normal dilated convolution operation.

After that, the $32 \times 32 \times 96$ feature maps are further convolved by 3×3 filters to generate a $8 \times 8 \times 192$ tensor. Finally, the dimensions of the parameters are compressed. An average pooling with a kernel size of 8 decreases the $8 \times 8 \times 192$ feature maps to a $1 \times 1 \times 192$ tensor, followed by a 1×1 filter, and two Leaky Relu layers. Subsequently the $1 \times 1 \times 192$ tensor is processed by two fully connected layers, a leaky relu and a softmax layer to generate a $1 \times 1 \times 30$ tensor. Since there are five identical branches, we obtain five $1 \times 1 \times 30$ tensors which are averaged and rearranged to generate the final 10×3 global color transformation matrix θ_P.

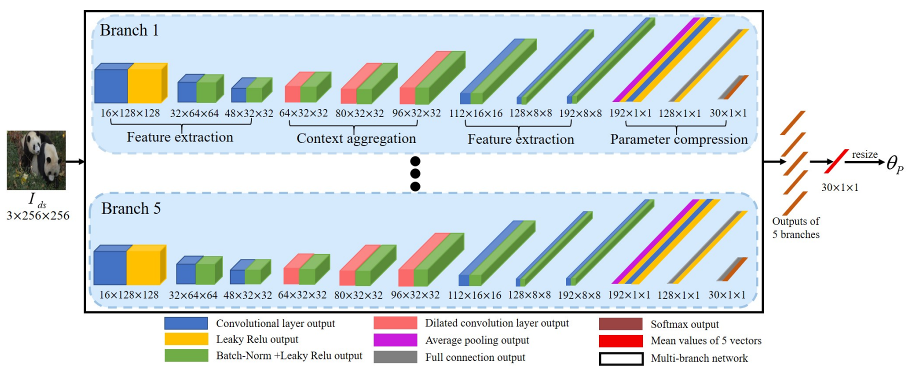


Fig. 2. The architecture of the global parameters extractor subnetwork.

3.3 Local Features Extractor Subnetwork

In order to extract features in different scales and overcome the vanishing gradient, deep neural networks with skip connections are proposed, such as UNet [4], ResNet [25] and DenseNet [26]. Empirically we adopt the DenseNet as the local feature extractor subnetwork. But unlike the original DenseNet which has three dense blocks, our neural network only uses one dense block. And the reason for adopting only one denseblock in our network is based on the following considerations: first is the complexity of the network, and the second is the small amount of datasets. An overly complex network may cause overfitting.

Here is a detailed description of the local features extractor subnetwork F, which is shown in Fig.3. First, I_F goes through a convolutional layer to expand the dimensions of I_F. And then the matrix enters the dense block. In the dense block, a bottleneck [26] layer with kernel size 1×1 is used before a convolution with kernel size 3×3, which is proposed to fuse features in DenseNet [26]. Such combination loops 3 times, which is far less than the times of each dense block in DenseNet. Therefore, a light dense block is presented in our network. In a denseblock, the output of each 3×3 convolution is the concatenation of the inputs of all convolutional layers before and the output of this convolutional layer. Finally, the feature matrix goes through the translation layer with kernel size 1×1 to reduce dimension and obtain θ_F. Every subsequent convolutional layer is followed by Batch Normalization and Relu layers. These layers are not shown in Fig. 3, in order to illustrate the convolutional layers of our light dense block more clearly and make comparisons with alternative architectures more easily. In Section 4.3, we will analyze the selection and adjustment of our local features extractor subnetwork.

We compare the results of whether to use local features extractor subnetwork, in Fig. 4. When subnetwork is not used, $\theta_F = I_F$, the algorithm is similar to PCE [5]. It can be clearly seen that without F subnetwork, the results have obvious enhancement effects, but some details at the bright area are lost, such as the silver decoration of the camera and the texture of the hand in the red box in Fig. 4(a), and the details of ceiling in Fig. 4(b). These features are very clear in the input, but after enhancement, local details are missing. It can be seen much clearer from the residual images that with the utility of the local features extractor subnetwork, the contents of bright areas are kept and the information of dark areas is restored.

3.4 Loss Function

We combine three loss terms to make the generated image as close as possible to the ground truth. Firstly, we use the $\mathcal{L}_2$ loss in the CIElab color space which correlate better with the human perception of color differences than other color spaces, such RGB, XYZ, YUV and so on. CIElab was derived from CIEXYZ. And the intention behind CIElab was to create a space that can be computed via simple formulas from CIEXYZ space but is more perceptually uniform then CIEXYZ. I is the input with RGB channels, O is the output, J is the target, and O^{lab} is the output transformed into the CIElab color space, J^{lab} is the target transformed into the CIElab color space. The $\mathcal{L}_2$ loss term is given by:

$$\mathcal{L}_2 = \frac{1}{N} \sum_i \|O_i^{lab} - J_i^{lab}\|^2 \tag{7}$$

N is the number of pixels, and i is the pixel index. We further add an color loss using the angular difference to measure the color similarity [21]. The RGB values

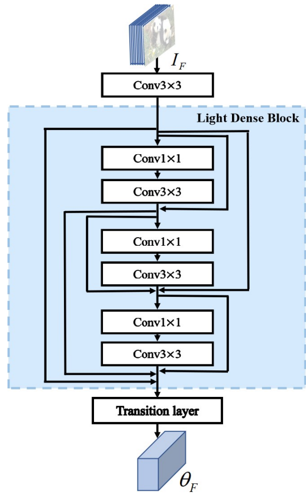


Fig. 3. The architecture of the local features extractor subnetwork.

of a pixel can be regarded as a 3D vector, and when two pixels are similar, the angle of the two RGB vectors approaches 0°, and its cosine value is close to 1. Therefore, the cosine of the angle should be as close as possible to 1. The color loss is defined as:

$$\mathcal{L}_{colorloss} = \frac{1}{N} \sum_i 1 - \cos \angle(O_i, J_i) \tag{8}$$

Where $\angle(,)$ means the angle of two vectors.

Besides considering the color similarity, we also pay attention to the structure of the image by using a SSIM loss [27] as defined in Eq. (9):

$$\mathcal{L}_{SSIM} = 1 - SSIM(O_i, J_i) \tag{9}$$

Finally, the total loss is given by:

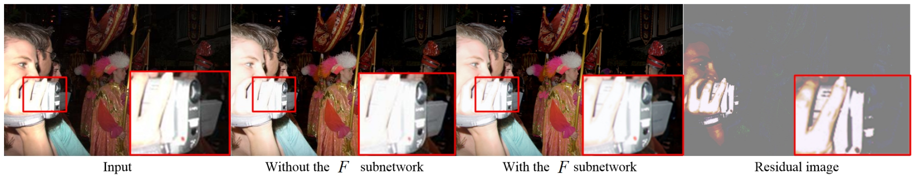
$$\mathcal{L} = \omega_1 \mathcal{L}_2 + \omega_2 \mathcal{L}_{colorloss} + \omega_3 \mathcal{L}_{SSIM} \tag{10}$$

where ω_1, ω_2, ω_3 are the weighting coefficients. The functionality of each loss function is analyzed in Section 4.2.

4 Experiments

4.1 Training Details

Training Datasets MIT-Adobe FiveK [28] is a high-quality dataset for color enhancement containing a collection of five sets of retouched image pairs by expert A/B/C/D/E. Like most previous methods [5][13][21], we select 5000 images retouched by Expert C as the ground truth. We randomly select 250 image pairs for validation and test, each including 125 pairs, and use the remaining 4750 images for training.



(a)

(b)

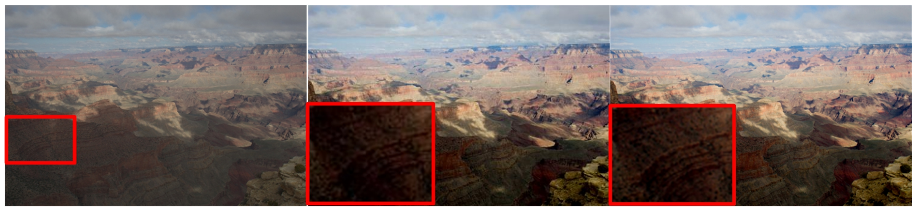
Fig. 4. The effects of the utilization of the local features extractor subnetwork. From left to right (1) Input (2) Without the F subnetwork (3) With the F subnetwork (4) Residual image.

Experiment Setting We train the proposed network using the Adam optimizer on a Nvidia Tesla P40 GPU with 24GB of memory. The batch size is 20 and the base learning rate is 9×10^{-4}. The learning rate linearly decays to 2×10^{-6}, and the training stops at the 500th epoch. ω_1, ω_2, ω_3 in the loss function (11) are set to 1, 200, and 10 respectively. PSNR and SSIM are used to evaluate the performance of our algorithm. Higher PSNR and SSIM values indicate better performance. For features extractor subnetwork, the input channels of the dense block is 24, and the growthrate is set as 12.

4.2 Ablation Study

Dilated Convolution Dilated convolution has the effect of expanding the receptive field. The specific role in the global parameters extractor subnetwork is further distinguished through the subjective effect. With dilated convolution, the details of the enhanced image are kept better. For example, in the red area of Fig. 5, without dilated convolution, the texture of the mountain is less visible.

Loss Function $\mathcal{L}_2$ loss, SSIM loss, and the improved color loss are used in this paper. Generally, we found that using SSIM can enhance the overall image quality, but in some cases, as illustrated by Fig. 6, the color diversity becomes more obvious. When using the improved color loss term, the color consistency can be improved, as shown in the fourth column of Fig. 6.



(a) Input (b) Without dilated convolution (c) Proposed

Fig. 5. The texture of the mountain is clearer when using the dilated convolution.

4.3 Alternative Architecture

For the local feature extractor network, we compare three different architectures. One of the structures is our proposed subnetwork F with the light dense block, which is illustrated in Section 3.3. Removing all of the skip connections in the light dense block, another structure is obtained, which is denoted as F'. For a fair comparison, the number of channels of each 3×3 convolutional layer matches that of the architecture. We also compare the setting that using the feature map $I_F = \theta_F$ directly without the local feature extractor subnetwork. The average MSEs of the validation set are shown in Table 1. It can be seen that the light dense block network has the lowest MSE and compared to F' it has a lower complexity, so we finally choose it as the local features extractor subnetwork in this paper.

Table 1. The average MSE values of different local feature extractor network.

Network structure	Without F	F'	Proposed
MSE	8.954	8.602	8.553

4.4 Comparison with the State-of-the-Art Methods

We compare our algorithm with eight state-of-the-art methods. There are six deep learning based methods, including HDRNet [17], Deep Photo Enhancer [11], Pix2pix [4], DPED [14], the supervised model in PCE [5] and RSGU [3]. For fair comparison, the same training set are used for these models. The other two methods are traditional algorithms, including NPEA [1] and SIRE [2].

Objective and subjective performance Table 2 shows the PSNR and SSIM of different methods. We only compare with deep learning methods, because these methods need ground truth. While for traditional algorithms, the purpose is to recover more details and colors in dark or bright scenes, sometimes

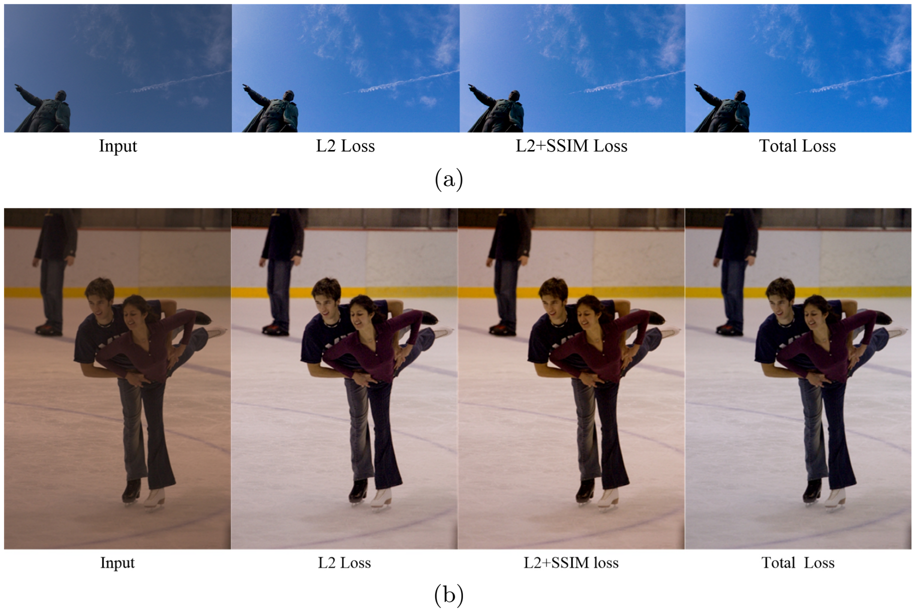


Fig. 6. Comparison with different loss functions. From left to right (1) Input (2) $\mathcal{L}_2$ loss only (3) $\mathcal{L}_2$+SSIM loss (4) Total loss.

these results show more content, but they are far from Expert-retouched images, resulting in extremely low values of PSNR and SSIM. While for DPED, the pre-training model is used. Moreover, because the training set of DPED is inconsistent with the training set of our algorithm and other comparison methods. So it is unfair to compare these two indicators with DPED [14], NPEA [1], SIRE [2], which cannot reflect the real effect of these algorithms. With the same dataset, our algorithm exhibits the highest PSNR and SSIM values. The objective indexes of HDRNet [17] are the worst, while PCE [5] and Deep Photo Enhancer[11] are relatively better.

Table 2. Objective performance of the deep learning methods in comparison.

Method	PSNR	SSIM
HDRNet[17]	19.138	0.860
Deep Photo Enhancer[11]	23.486	0.935
Pix2pix[4]	20.581	0.890
PCE[5]	24.127	0.905
RSGU[3]	20.818	0.905
Proposed	**24.684**	**0.948**

Subjective comparison is given in Fig. 7. It can be seen from the comparison that our algorithm can not only maintain and restore more details, but also have brighter colors and higher contrast. For example, the sky in Fig. 7(a), our method is bluer and more vivid, and the contrast of the mountain is higher; the enlarged part of the white flowers of our algorithm in Fig. 7(b) still maintains excellent details; the color and contrast of the squirrel and background in Fig. 7(c) are superior to the effect of other algorithms.

User Study We conducted a user study and adopted pairwise comparisons with seven state-of-the-art methods, which included 20 images and 58 participants. We compared the paired results of our algorithm and the state-of-art method, and each set of images was compared 7 times. While testing, the paired-images were displayed side-by-side, allowing users to choose which one they prefer, according to colors and details of the image. The 20 images were randomly selected from our test set. The results of user study are shown in Fig. 8. Our algorithm is more frequently picked as the better one compared with other state-of-the-art methods.

Application on Video Our algorithm can also be applied on videos. Lightroom is a popular image processing software which can also process videos. We use the Lightroom Auto-Tune feature to enhance the images. We downloaded 4K videos from Youtube, and compared the enhancing performance of our algorithm and the lightroom. Our algorithm shows superior vivid results, and the enhanced colors of adjacent frames stay consistent. Processed videos are attached as supplementary materials.

5 Conclusion

We introduced a novel approach for color enhancement that the enhancement map is learned using an end-to-end neural network, and added to the input to obtain an enhanced image. The enhancement intensity can be adjusted by tuning a coefficient, predefined or automatically determined according to the image or video content. To generate the enhancement map, the extraction of global parameters and local features are two important components. In the global parameter extractor network, a multi-branch network is used, and the operation of multi-scale dilated convolution layers is introduced to aggregate global information. We also design a network with a light dense block that can help to enhance local contrasts. Furthermore, we present a new loss function, combining the improved color loss with a $\mathcal{L}_2$ loss and a SSIM loss. We conducted experiments to compare our algorithm with the state-of-art methods. Our algorithm shows superior performance evaluated by both objective metrics and a subjective user study.

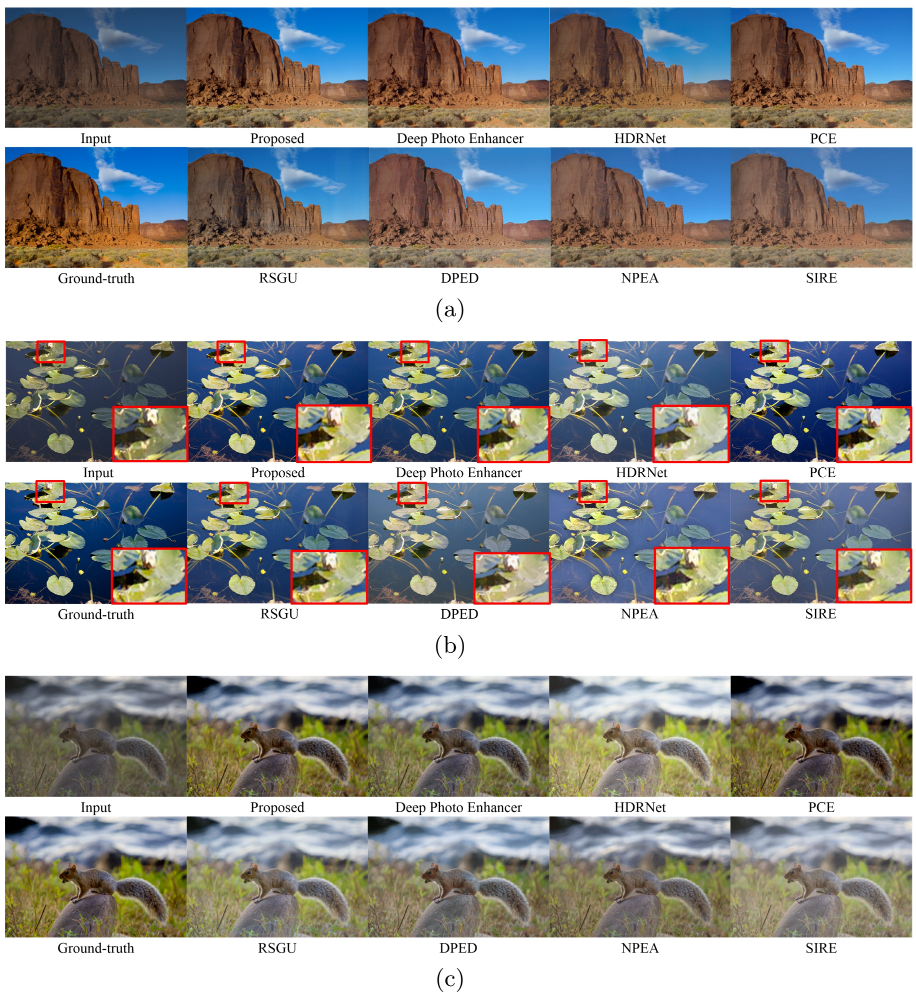


Fig. 7. Subjective quality comparisons with the state-of-the-art methods. From left to righ, from top to bottom (1) Input (2) Proposed (3) Deep Photo Enhancer (4) HDRNet (5) PCE (6) Expert (7) RSGU (8) DPED (9) NPEA (10) SIRE.

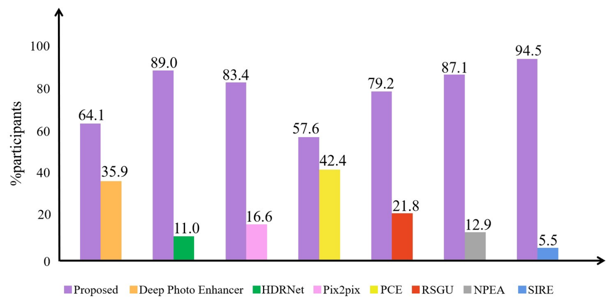


Fig. 8. User study results.

References

1. Wang, S., Zheng, J., Hu, H., Li, B.: Naturalness preserved enhancement algorithm for non-uniform illumination images. IEEE Trans. Image Process. **22** (2013) 3538–3548
2. Fu, X., Liao, Y., Zeng, D., Huang, Y., Zhang, X.S., Ding, X.: A probabilistic method for image enhancement with simultaneous illumination and reflectance estimation. IEEE Trans. Image Process. **24** (2015) 4965–4977
3. Huang, J., Zhu, P., Geng, M., Ran, J., Zhou, X., Xing, C., Wan, P., Ji, X.: Range scaling global u-net for perceptual image enhancement on mobile devices. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 230–242
4. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1125–1134
5. Chai, Y., Giryes, R., Wolf, L.: Supervised and unsupervised learning of parameterized color enhancement. In: The IEEE Winter Conference on Applications of Computer Vision. (2020) 992–1000
6. Veluchamy, M., Subramani, B.: Image contrast and color enhancement using adaptive gamma correction and histogram equalization. Optik **183** (2019) 329–337
7. Hassanpour, H., Samadiani, N.: A new image enhancement method considering both dynamic range and color constancy. scientiairanica **26** (2019) 1589–1600
8. Wang, B., Yu, Y., Xu, Y.Q.: Example-based image color and tone style enhancement. ACM Transactions on Graphics (TOG) **30** (2011) 1–12
9. Lee, J.Y., Sunkavalli, K., Lin, Z., Shen, X., So Kweon, I.: Automatic content-aware color and tone stylization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2470–2478
10. Hui, Z., Wang, X., Deng, L., Gao, X.: Perception-preserving convolutional networks for image enhancement on smartphones. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 197–213
11. Chen, Y.S., Wang, Y.C., Kao, M.H., Chuang, Y.Y.: Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 6306–6314
12. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. (2017) 2223–2232

13. 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. (2018) 5928–5936
14. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: Dslr-quality photos on mobile devices with deep convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 3277–3285
15. Chen, Q., Xu, J., Koltun, V.: Fast image processing with fully-convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 2497–2506
16. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. (2015) 234–241
17. Gharbi, M., Chen, J., Barron, J.T., Hasinoff, S.W., Durand, F.: Deep bilateral learning for real-time image enhancement. ACM Transactions on Graphics (TOG) **36** (2017) 1–12
18. Bianco, S., Cusano, C., Piccoli, F., Schettini, R.: Artistic photo filter removal using convolutional neural networks. Journal of Electronic Imaging **27** (2017) 011004
19. Bianco, S., Cusano, C., Piccoli, F., Schettini, R.: Learning parametric functions for color image enhancement. (2019) 209–220
20. Afifi, M., Price, B., Cohen, S., Brown, M.S.: When color constancy goes wrong: Correcting improperly white-balanced images. (2019) 1535–1544
21. Wang, R., Zhang, Q., Fu, C.W., Shen, X., Zheng, W.S., Jia, J.: Underexposed photo enhancement using deep illumination estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 6849–6857
22. Bianco, S., Cusano, C., Piccoli, F., Schettini, R.: Content-preserving tone adjustment for image enhancement. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2019) 1936–1943
23. Moran, S., Marza, P., McDonagh, S., Parisot, S., Slabaugh, G.: Deeplpf: Deep local parametric filters for image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 12826–12835
24. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., Cottrell, G.: Understanding convolution for semantic segmentation. In: 2018 IEEE winter conference on applications of computer vision (WACV). (2018) 1451–1460
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
26. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 4700–4708
27. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. Volume 2. (2003) 1398–1402
28. Bychkovsky, V., Paris, S., Chan, E., Durand, F.: Learning photographic global tonal adjustment with a database of input/output image pairs. In: CVPR 2011. (2011) 97–104