Hybrid Image Enhancement With Progressive Laplacian Enhancing Unit

Jie Huang, Zhiwei Xiong, Xueyang Fu, Dong Liu, Zheng-Jun Zha
University of Science and Technology of China
zwxiong@ustc.edu.cn

ABSTRACT

In this paper, we propose a novel hybrid network with Laplacian enhancing unit for image enhancement. We combine the merits of two representative enhancement methods, i.e., the scaling scheme and the generative scheme, by forming a hybrid enhancing module. Meanwhile, we model image enhancement in a progressive manner with a deep cascading CNN architecture, in which the previous feature maps are used to enhance subsequent features to get an improved performance. Specifically, we propose a Laplacian enhancing unit, which can adjustably enhance the detail information by adding the residual of previous feature maps. This unit is embedded across layers for progressively enhancing the features. We build our network on the U-Net architecture and name it Hybrid Progressive Enhancing U-Net. Experiments show that our method achieves superior image enhancement results compared with the state-of-the-arts, while retaining competitive implementation efficiency.

CCS CONCEPTS

• **Computing methodologies → Computational photography**.

KEYWORDS

Hybrid enhancement; Laplacian enhancing unit; Feature enhancing.

ACM Reference Format:
Jie Huang, Zhiwei Xiong, Xueyang Fu, Dong Liu, Zheng-Jun Zha. 2019. Hybrid Image Enhancement With Progressive Laplacian Enhancing Unit. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350855

1 INTRODUCTION

With the increasing usage of mobile phones and digital cameras, obtaining images becomes cheaper and easier. Still, images taken by the nonspecialist or under poor lighting conditions often suffer from low quality. To improve the visual experience of these captured images, softwares such as Photoshop are widely used for image enhancement. However, using the specially designed softwares is time-consuming and requires professional skills.

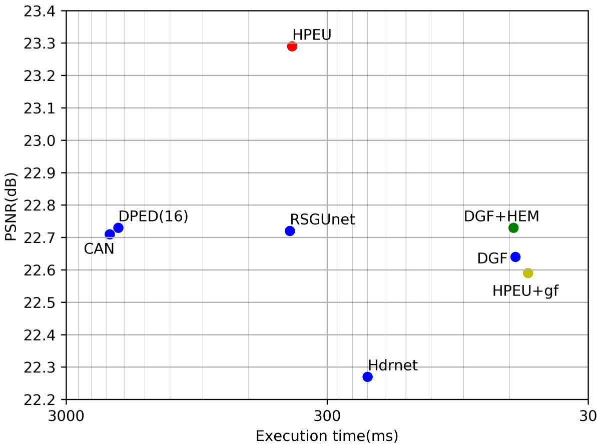


Figure 1: PSNR and execution time comparisons on MIT-Adobe FiveK dataset of different methods. Our HPEU achieves the best performance. Note that, after adding the guided filter, our method achieves the highest efficiency. And after adding our HEM, the performance of DGF has improved without decelerating the speed.

Recently, a number of deep learning methods have been proposed for automatic image enhancement, which can be roughly summarized into two categories: scaling methods [10, 12, 22] and generative methods [4, 13]. The former usually scale the input pixels or features by learning a mapping function for the stretching relationship between input and target images, while the latter generate new components from extracted features to reconstruct output images. Although these two kinds of methods have achieved promising results, the task of image enhancement still remains challenging due to the complexity of factors that degrade the image quality, and there is still a large room for improvement.

In this paper, we propose a hybrid enhancing module (HEM) by jointly exploiting the merits of scaling methods and generative methods. We implement this module by splitting feature maps in the last layer of a deep network into two branches. One is used to obtain the confidence map for pixel stretching, which then produces the scaling component from the input image. The other is used to obtain the generative component for information compensation. The final enhanced result is obtained by fusing the scaling component and the generative component. To the best of our knowledge, this is the first time to combine the scaling scheme and the generative scheme for image enhancement.

MM '19, October 21–25, 2019, Nice, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350855

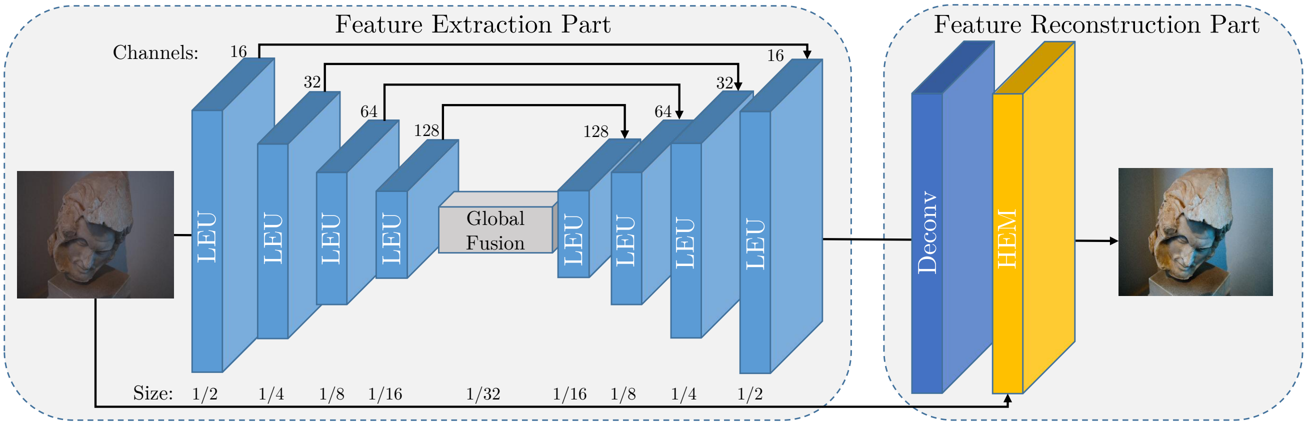


Figure 2: The overall structure of HPEU. Our network is based on a U-Net architecture. The LEU is embedded in the feature extraction part and the HEM mainly forms the feature reconstruction part.

On the other hand, traditional image enhancement is a progressive procedure, where an image is processed by a number of steps [23, 26, 30] and the intermediate result can benefit the subsequent step. However, existing deep learning methods directly propagate the extracted features to the next layer, and these features are not further processed. Since features at different layers exhibit different image characteristics [31], enhancing the extracted features can highlight specific information. This helps the subsequent layer to make better use of the information contained in the previous layer. Inspired by the Laplacian enhancement method [15], we develop a Laplacian enhancing unit (LEU) which can enhance the feature maps according to their different layers. We implement this unit by calculating the residual between adjacent layers, and then add the residual to the subsequent layer followed by a multiplier. This procedure can be regarded as a network embodiment of the Laplacian enhancement method where the multiplier is learned from training data. Since LEU can be embedded across all network layers, it can progressively enhance the features.

To guarantee the implementation efficiency, we integrate the proposed HEM and LEU under a lightweight U-Net architecture [12] to form our Hybrid Progressive Enhancing U-Net (HPEU). The contributions of this work are summarized as follows:

- We propose a novel HEM by jointly exploiting the merits of generative methods and scaling methods, which achieves an improved performance.
- We introduce a new LEU by adding the residual of adjacent layers to the subsequent layer, which realizes progressive enhancement and boosts features with adjustable enhancing degrees.
- Our lightweight HPEU integrating HEM and LEU achieves state-of-the-art image enhancement performance on several widely used datasets while retaining competitive efficiency, as shown in Fig. 1.

2 RELATED WORK

In the past decades, various methods have been proposed for image enhancement. We divide these methods into two categories: scaling methods and generative methods.

2.1 Scaling Methods

Traditional image enhancement methods mainly belong to the scaling category. Most of them process the image in the spatial domain [6]. For example, Tian *et al.* [22] propose a local and global histogram based method for image enhancement. In [15], the Laplacian operator is utilized to scale the high-frequency components of the image. Other methods enhance the image by manipulating components in some transform domains, such as the wavelet transform [1]. Hybrid domain methods combine spatial domain methods and frequency domain methods. For example, Fan *et al.* [29] convolve the input image with an optimal Gaussian filter, divide the original histogram into different areas by the valley values, and separately process each area.

Recently, deep learning methods have been introduced for image enhancement, many of which adopt the scaling scheme. Deng *et al.* [5] propose a deep neural network to globally enhance the input image by learning the coefficient of some adjustment operators, such as gamma adjustment and contrast adjustment. Gharbi *et al.* [8] propose a novel network architecture that can process 1080p resolution video in real-time on smartphones by learning the affine transformation between input and target images. Huang *et al.* [12] develop a lightweight U-Net based architecture that uses the global feature to learn a scaling layer operating on the input features to get the final output. Park *et al.* [20] propose a reinforcement learning based framework that adjusts the pixels step-by-step. All of the above methods attempt to obtain the stretching relationship between input and target images. Furthermore, the scaling scheme has also been adopted in other low-level vision tasks such as color constancy [11] and image dehazing [16].

2.2 Generative Methods

The generative methods mainly use CNN architectures, which have made great progress in many low-level vision tasks, including super-resolution [7], dehazing [3], denoising [32], and hyper-spectral reconstruction [27]. This kind of approaches extract the features from the input image and reconstruct them to form the target image. To name a few, Ignatov *et al.* [13] propose an architecture which comprises several residual blocks and exploits perceptual losses to train the network. Yan *et al.* [28] propose an architecture which fuses the information of the color histograms from the input image. EnhanceNet [21] generates images with more realistic texture by using a perceptual loss. Chen *et al.* [4] use a sequence of dilated convolution layers to achieve both large receptive fields and fast speed. Based on [4], Wu *et al.* [25] propose a framework which processes the image on the low resolution and up-samples it by applying a guided filter layer to achieve a fast speed.

Despite the encouraging progress, existing methods use either the scaling scheme or the generative scheme, which cannot take full advantage of these two ways. Besides, while traditional image enhancement generally processes the image progressively, existing deep learning methods only extract and propagate the features without further processing. Intuitively, if the features can be enhanced at each step, they can provide more effective information for the following operations. The above issues motivate us to design a new architecture to jointly exploit the merits of scaling methods and generative methods, which also inherits the advantage of traditional image enhancement by enhancing the features progressively.

3 PROPOSED METHOD

In this section, we first briefly introduce the network structure of HPEU and then elaborate HEM and LEU, which are the two cores of our method.

3.1 Network Structure of HPEU

Our proposed method is based on the U-Net architecture [12] as shown in Fig. 2, which contains an encoder and a decoder. The encoder down-samples the features to the low-resolution space, while the decoder up-samples the features to the original size. To acquire global information, we adopt a global fusion layer as in [12], which uses average pooling to process the features and then concatenates them with former features after a fully-connected layer. By deploying this operation, this architecture can process any size of images and avoid artifacts.

Different from [12], we remove the first layer and replace the convolution layer of stride 2 with the max-pooling layer to down-sample the features, which accelerates the implementation speed. We divide our HPEU into two parts: feature extraction and feature reconstruction. In the feature extraction part, we embed LEU in the network in place of two successive convolution layers, which can progressively enhance the feature at the different layers. In the feature reconstruction part, we adopt HEM as a substitute for the last layer of the network in [12]. The HEM generates a confidence map and a generative component at the same time. The confidence map is used to scale the input image in a pixel-wise manner to produce the scaling component and the generative component is used to complement more information. The final enhanced image is the summation of the scaling component and the generative component.

3.2 Hybrid Enhancing Module (HEM)

We design HEM by exploiting the scaling scheme and the generative scheme in a complementary way. Since both methods have promising performance, combining them together can fully utilize their advantages.

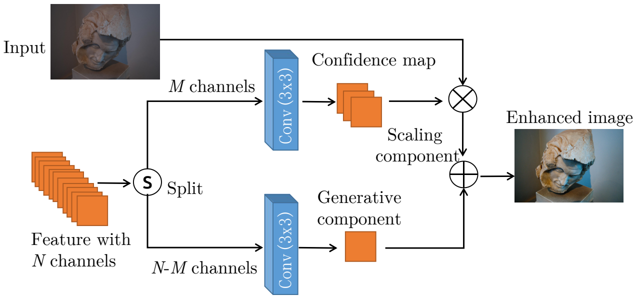
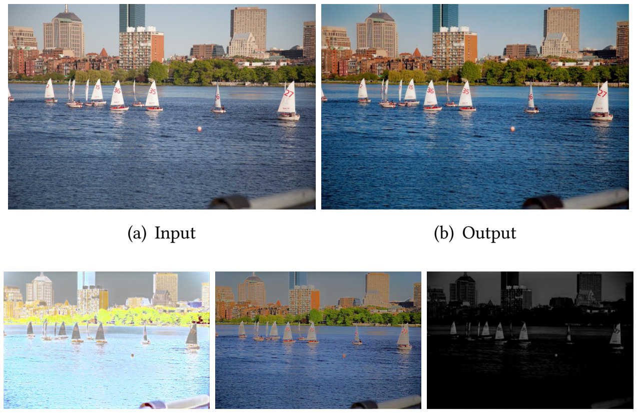


Figure 3: The structure of hybrid enhancing module (HEM).



(a) Input (b) Output

(c) Confidence map (d) Scaling component (e) Generative component

Figure 4: Illustration of different components in HEM. The confidence map scales the input image while the generative component compensates the scaling component.

We now describe the detail of HEM. As shown in Fig. 3, the feature in the last layer of HPEU contains N channels. These channels are separated into two branches: one is convoluted into a confidence map with three channels, and the other is convoluted into the generative component with one channel. The kernel size of these convolution layers is 3. The activation functions of the scaling branch and the generative branch are Leaky Relu [18] and Tanh [14] respectively. This procedure can be expressed as follows

$$\begin{cases} c = A_1(\Phi_1(S(T, M))), \\ g = A_2(\Phi_2(S(T, N-M))) \cdot \alpha + \beta, \end{cases} \tag{1}$$

where T denotes the feature map with N channels, M is the number of channels assigned for producing the confidence map, S is the separate operation for channels, Φ_1 and Φ_2 are two convolution layers, A_1 and A_2 are leaky Relu and Tanh activation functions, respectively, c is the produced confidence map, and g is the generative component. We multiply the Tanh activation result with a factor of α and add a bias of β to make sure that the value of the generative component is similar to a normal image.

We obtain the scaling component by multiplying the input image with the confidence map. The final output is obtained by

$$y[i,j] = x[i,j] \cdot c[i,j] + g[i,j], \tag{2}$$

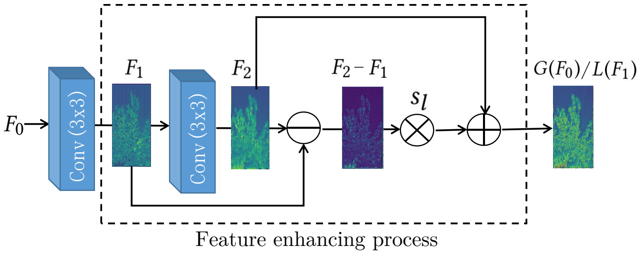


Figure 5: The structure of Laplacian enhancing unit (LEU). The feature enhancing process is the core of LEU.

where x is the input image, y is the final output, i and j denote the pixel position in the spatial dimensions. Note that the three channels of the confidence map correspond to the R, G, B channels of the input image, and the generative component is only for the luminance compensation.

Eq. (2) demonstrates that the confidence map can scale the input image in a pixel-wise manner. Since different objects in an image have different scene information, the deep network can force them to be enhanced in adjustable degrees. As shown in Fig. 4(c), the confidence map scales different regions of the input image. The generative component as shown in Fig. 4(d) is reconstructed from the features. Since the scaling component looks unnatural as shown in Fig. 4(e), the generative component plays a role as a compensation of the scaling one. By adding the generative component and the scaling component together, the visual quality is further improved. Alternative channel division methods can be used to produce the confidence map and the generative component by changing the value of M in Eq. (1).

3.3 Laplacian Enhancing Unit (LEU)

Inspired by the Laplacian enhancement method [15], we design LEU to simulate the progressive procedure for feature enhancement. The Laplacian enhancement method enhances images by adding the scaled high-frequency part of images. It can be expressed as

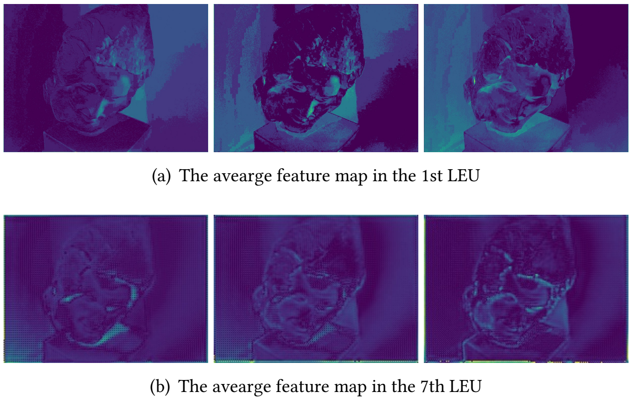
$$E = I + s_c \cdot (I - h(I)), \tag{3}$$

where h is the blur kernel function, I is the original image, s_c is the constant scaling coefficient and E denotes the enhanced image.

Regarding the kernel function in Laplacian enhancement as a specific convolution kernel that is not learnable, the features in the deep network can be enhanced by using the same principle. Specifically, we use the residual between adjacent layers as the substitute of the high-frequency part, and replace the scaling coefficient s_c with a learnable parameter. That is

$$L(F_1) = F_1 + s_l \cdot (F_1 - \Phi(F_1)), \tag{4}$$

where $L(\cdot)$ denotes the feature enhancing process, F_1 is the feature in the first layer, Φ is the convolution kernel and s_l is the learnable coefficient.


(a) The avearge feature map in the 1st LEU

(b) The avearge feature map in the 7th LEU

Figure 6: Visualization of the absolute average feature maps after different LEUs. From left to right: F_1, $F_2 - F_1$, and $L(F_1)$, respectively.

However, compared with F_1, the features in the subsequent layer contain information with a larger scale due to the increase of receptive fields. Thus, in order to enhance the features behind F_1, we form LEU as shown in Fig. 5 and the feature enhancing process in LEU can be expressed as

$$L(F_1) = \Phi(F_1) + s_l \cdot (\Phi(F_1) - F_1), \tag{5}$$

If we denote $\Phi(F_1)$ as F_2, Eq. (5) can be expressed as the same format of Eq. (3) and Eq. (4) as

$$L(F_1) = F_2 + s_l \cdot (F_2 - \Phi_{-1}(F_2)), \tag{6}$$

where Φ_{-1} is the inverse procedure of convolution.

The only difference between Eq. (6) and Eq. (3) is that the blur kernel function in Eq. (3) is replaced by the inverse procedure of convolution. With such a Laplacian enhancing unit, we can enhance the features with increasing receptive fields. The degree of the enhancement can be learned in the form of the scaling coefficient s_l. If the feature needs to be enhanced more, s_l would be larger. Moreover, since the convolution layer is a learnable kernel, the residuals now represent any desired information to emphasize rather than just high-frequency parts.

In this way, LEU can enhance different levels of features with adjustable degrees. For example, in shallow layers, it mainly enhances the local regions with edges. On the contrary, due to the increased receptive fields, it enhances the global structures in deep layers. As shown in Fig. 6(a), in shallow layers, LEU enhances the local regions with rich details, such as the textures on the face. While in deep layers as shown in Fig. 6(b), the global structures of head and hair are enhanced.

Since LEU also contains the convolution layer for feature extraction, we express the whole procedure through LEU as

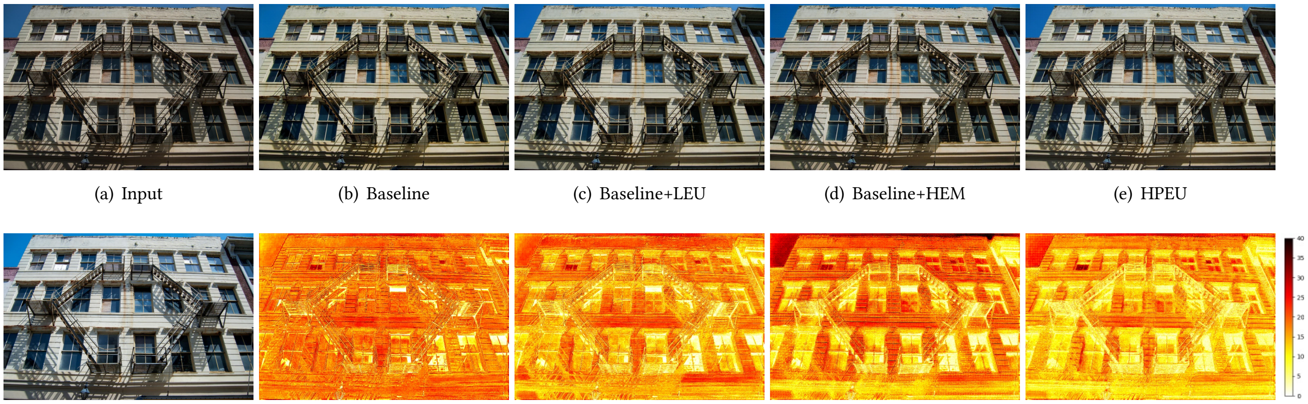
$$G(F_0) = L(\Phi_{ex}(F_0)) = L(F_1), \tag{7}$$

where F_0 is the input feature of LEU, Φ_{ex} is the convolution layer for feature extraction. By embedding LEU across all network layers, HPEU can progressively enhance the features and the subsequent layer of LEU is able to utilize the enhanced features better. This progressive procedure can be formulated as

$$F_0^n = G^n(G^{n-1}(...(G^2(G^1(F_0^0))))), \tag{8}$$

where F_0^n denotes the features processed after n-th LEUs.

We repeatedly deploy the LEU in the feature extraction part. Since the final confidence map and the generative component are



(a) Input (b) Baseline (c) Baseline+LEU (d) Baseline+HEM (e) HPEU

(f) Ground truth (g) Error map of (b) (h) Error map of (c) (i) Error map of (d) (j) Error map of (e)

Figure 7: Visual results from the MIT-Adobe FiveK dataset for ablation study. Error maps show the residual between the output image and the ground truth, measured by Manhattan distance in 0-255 RGB space. Darker pixels indicate larger errors.

transformed from the features, better enhanced features effectively promote the enhancement performance, as demonstrated in the experiments below.

4 EXPERIMENTS

4.1 Datasets and Settings

The MIT-Adobe FiveK dataset [2] is widely used for image processing tasks, such as enhancement, image smoothing, and nonlocal dehazing. We adopt it as the main dataset to evaluate the performance of our method. This dataset contains 5,000 high-resolution photographs covering a broad range of scenes, subjects and lighting conditions. Each picture has five well-aligned categories with high-quality edition provided by five experts. These five categories are named Expert A, B, C, D, and E, respectively. Since the category Expert C is mostly used for the image enhancement task, it is selected as the high-quality image sets in our experiments. Since the content of these images is not relevant to the index, we use the first 4.5K pairs from it for training and the other 497 pairs for testing (the 3 pairs left are not well aligned). To train our network, we resize the training images to the resolution of 480s (i.e., the short side of an image is resized to 480 without changing the aspect ratio) for the pre-training stage in the first 150 epochs. For the fine-tuning stage in the latter 30 epochs, we randomly resize the training images to 320s, 480s, 720s, 1080s and 1440s, which is also adopted to resize the testing images. This is in accordance with previous works [4, 25].

To further evaluate the performance of our method, we also adopt a subset of the DPED dataset [13] tailored for enhancement, from which image pairs are separated into many small patches and are not precisely aligned. The DPED dataset contains photographs taken by four devices, including three mobile phones which are of low-quality and one DSLR camera which captures high-quality images. The images in this dataset are cut into patches of size 100×100 and then coarsely aligned. We mainly use the images captured by iPhone 3GS and the corresponding DLSR version in our experiments, which contains 160,471 patches for training and 4,354 patches for testing, named DPED-iPhone. Moreover, it contains 29 full-size iPhone-3GS images without corresponding DSLR images, which we use as additional testing images. The other two sub-datasets are called DPED-Sony and DPED-Blackberry, which are used to evaluate the generalization capability of the networks.

We implement our network by using Tensorflow. For the MIT-Adobe FiveK dataset, we train the images with the batch size of one by using the learning rate of 0.0001 on one single NVIDIA-GTX1080Ti GPU for 180 epochs. While for the DPED dataset, we train the network by using the learning rate of 0.0002 for 150000 iterations with the batch size of 32 on the same device. We evaluate the running time of different networks on a low-end NVIDIA-GTX730 GPU, which can better distinguish the difference of their speed. For HPEU, we assign the values of N and M to 4 and 3, and α and β to 0.58 and 0.5 as the same in [13]. Other compared methods are conducted under the same conditions and training data. PSNR and Multi-Scale SSIM (MS-SSIM) [24] are adopted to evaluate the performance of enhancement.

4.2 Ablation Study

Investigation of HEM and LEU. We conduct several ablation experiments to demonstrate the effectiveness of the proposed HEM and LEU. For a fair comparison, we use the mean square error (MSE) loss uniformly during the training process for all compared methods, unless noted otherwise. We adopt the HPEU without HEM and LEU as the baseline in our ablation study. The results in Table 1 show that, while using either HEM or LEU alone can improve the PSNR, combining them together to form HPEU can achieve a notable performance gain of 0.62dB compared with the baseline. Note that, compared with the baseline, HPEU retaining the implementation efficiency without introducing extra parameters. Furthermore, after adding HEM and LEU, the visual quality is closer to the ground truth as shown in Fig. 7.

Investigation of parameters. We investigate the contribution of parameters by adjusting the number of filters in each layer. We set the number of filters in each layer to 0.125, 0.25, 0.5, and 2 times of that in the original HPEU respectively, and we also conduct experiments for their corresponding baselines. As shown in Fig. 8, HPEU can obviously improve the performance of the baseline if the

Table 1: Ablation study of HEM and LEU on MIT-Adobe FiveK dataset.

Methods	PSNR	MS-SSIM	Time (ms)	Parameters
Baseline	22.67	0.9355	**371**	3.8M
+LEU	22.94	0.9352	383	3.8M
+HEM	22.79	0.9405	390	3.8M
HPEU	**23.29**	**0.9431**	408	3.8M

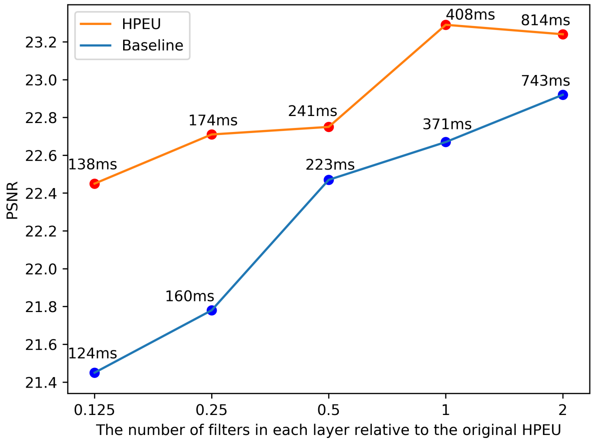


Figure 8: Investigation of parameters.

number of parameters is relatively small. Note that when we use 25 percent of filters in each layer, HPEU can still keep competitive performance compared to other methods shown in Table 2, while achieving a fast speed compared to other networks shown in Table 4, which demonstrates an elegant balance between performance and speed of HPEU.

4.3 Comparison with State-of-the-Arts

Comparison with representative enhancement networks. We compare our HPEU with several state-of-the-art supervised enhancement networks including DPED [13], RSGUnet [12] and CAN [4]. Among these methods, RSGUNet belongs to the scaling category, while CAN and DPED are generative methods. Since DPED consumes massive computational resources and is unable to train on existing GPUs on the full-size MIT-Adobe FiveK dataset, we replace 64 channels with 16 channels in DPED to enable its training on this dataset. For a fair comparison, we only use the MSE loss to train all compared methods.

Table 2: Quantitative results of representative networks on MIT-Adobe FiveK and DPED-iPhone datasets with MSE loss.

	MIT-Adobe FiveK		DPED-iPhone	
Methods	PSNR	MS-SSIM	PSNR	MS-SSIM
DPED [13]	22.73	0.9282	**22.90**	**0.9182**
CAN [4]	22.71	0.9340	19.21	0.7770
RSGUnet [12]	22.72	0.9359	22.63	0.9186
HPEU	**23.29**	**0.9431**	22.82	0.9173

Table 3: Quantitative results of fast image processing methods on MIT-Adobe FiveK dataset.

Methods	PSNR	MS-SSIM
DGF [25]	**22.64**	0.9353
Hdrnet [8]	22.27	0.9391
HPEU+guided-filter-layer	22.59	**0.9383**

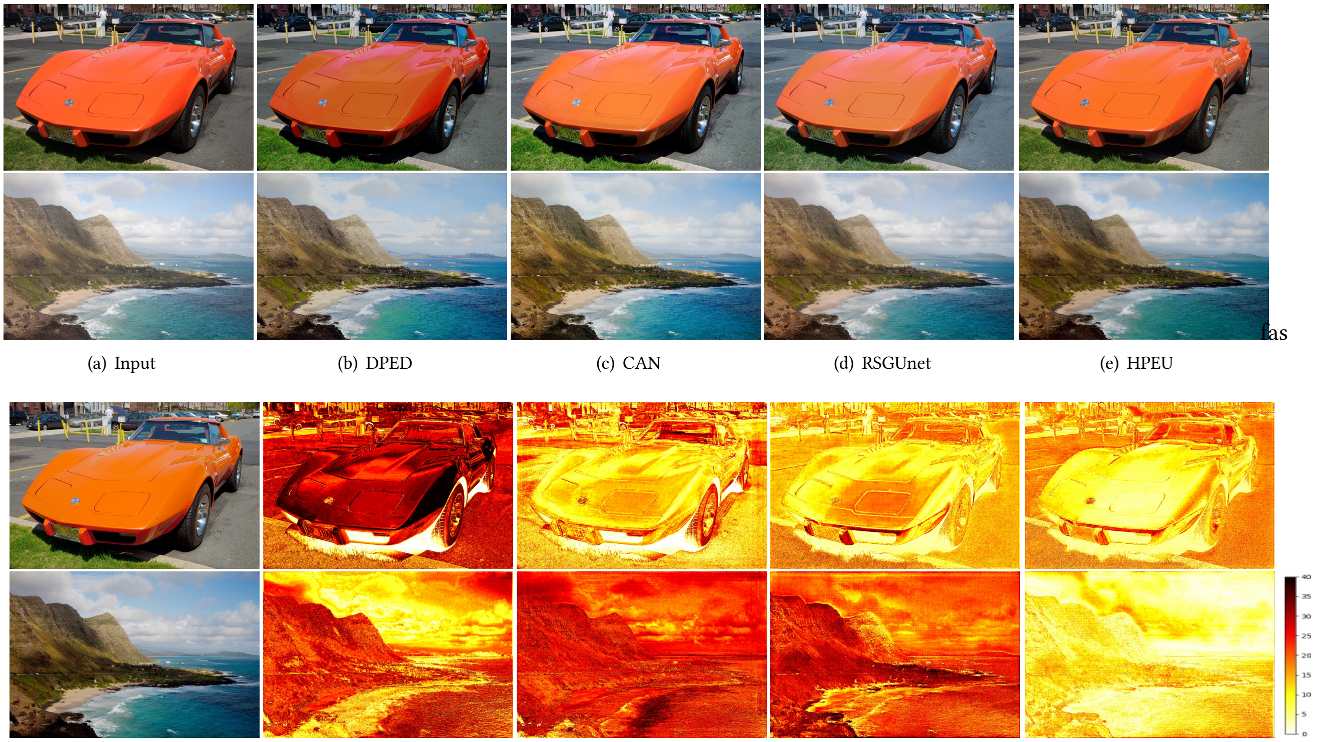
As shown in Table 2, HPEU outperforms other methods in terms of both PSNR and MS-SSIM on the MIT-Adobe FiveK dataset. The qualitative results are shown in Fig. 9, where our result is closer to the ground truth. We also find that HPEU achieves a better performance than other methods except for DPED on DPED-iPhone. The reason is probably that the dataset is patch-wise and not precisely aligned, and HPEU tends to introduce more relative deviation in the down-sampled features. Still, the performance of our HPEU is very close to DPED and achieves a much faster speed as shown in Table 4. In summary, HPEU can achieve a promising performance with a fast speed on both datasets, which demonstrates the effectiveness and efficiency of our method.

Comparison with fast image processing methods. Recently, fast image processing methods have drawn more attention. Some representative methods, such as Hdrnet [8] and DGF [25] have been proposed. They can achieve a fast speed by processing images on low resolution and then up-sample the outputs through the guidance of the original images. For a fair comparison, we add the guided filter in DGF to HPEU and the input image of the network is resized to 100×100 resolution. We conduct the experiments on full-size MIT-Adobe FiveK dataset. Table 3 and Table 4 demonstrate that after adding the guided-filter, HPEU can achieve an even faster speed as well as a comparable performance with DGF and Hdrnet.

Comparison of speed. We use the 1080×1080 resolution patches to evaluate the speed of different methods, as this resolution is close to the size of images daily captured. The results are shown in Table 4, from which we can see that the speed of the original HPEU is faster than other enhancement networks except for DGF and Hdrnet. After adding the guided filter, HPEU achieves the highest speed among all compared methods.

Comparison of perceptual quality. It is important to evaluate the perceptual quality in the image enhancement task. Since RSGUnet and DPED have achieved promising perceptual quality on DPED-iPhone, we use this dataset for perceptual quality comparison. We add the perceptual loss of RSGUnet to HPEU and compare HPEU with other networks. As shown in Table 5, our method gives the highest PSNR and MS-SSIM after adding the perceptual loss. Moreover, to evaluate the perceptual quality of enhancement, we use 29 full-size images with no references from DPED-iPhone to test different methods. The perceptual index proposed from PIRM@ECCV 2018 challenge is adopted, which combines the Ma [17] and NIQE [19] indexes, and is expressed as 0.5*((10-Ma)+NIQE) (a lower score indicates better visual quality). The results shown in Table 5 demonstrate that our method also achieves the best perceptual index.

We further illustrate the visual results in Fig. 10. DPED can keep the details of the image in the red bounding box while introducing noise in the green bounding box, and RSGUnet has the opposite



(a) Input (b) DPED (c) CAN (d) RSGUnet (e) HPEU

(f) Ground truth (g) Error map of (b) (h) Error map of (c) (i) Error map of (d) (j) Error map of (e)

Figure 9: Visual results from the MIT-Adobe FiveK dataset for different methods. Error maps show the residual between the output image and the ground truth, measured by Manhattan distance in 0-255 RGB space. Darker pixel indicates larger errors.

Table 4: The average runtime of different methods for processing a 1080×1080 resolution patch.

Methods	DPED (64) [13]	DPED (16)	CAN [4]	RSGUnet [12]	HPEU	Hdrnet [8]	DGF [25]	HPEU+guided-filter
Runtime(ms)	Out of memory	1895	2043	417	408	210	57	51

Table 5: Quantitative results of different methods with perceptual loss on DPED-iPhone dataset.

Methods	PSNR	MS-SSIM	Perceptual index
Input	17.08	0.8372	4.10
DPED [13]	22.57	0.9186	3.67
RSGUnet [12]	22.83	0.9254	3.77
HPEU	**23.02**	**0.9255**	**3.50**

effect that oversmooths the details in the red bounding box. As these two methods belong to generative methods and scaling methods respectively, our HPEU combines the merits of the two catogiries, and achieves an elegant balance between detail enhancement and noise suppression.

Comparison of generalization capbility. In practice, the enhancement network trained on a specific dataset will be finally applied to real-world images. To demonstrate the generalization performance of the above networks, we apply their models trained on the MIT-Adobe FiveK dataset to DPED-iPhone, DPED-Sony, and DPED-Blackberry. As shown in Table 6, HPEU outperforms other networks by a large margin, which demonstrates that our method has the best generalization capability.

Table 6: Generalization capbility of different networks on three kinds of DPED datasets in terms of PSNR.

Methods	DPED-iPhone	DPED-Sony	DPED-Blackberry
DPED [13]	17.22	16.58	16.32
CAN [4]	16.33	15.02	15.72
RSGUnet [12]	16.17	14.97	15.17
HPEU	**18.32**	**16.91**	**17.39**

4.4 Further Study

Extension of HEM to other architectures. To further demonstrate the effectiveness of HEM, we replace the last layer of CAN, DPED, and DGF with this module. We separate half of their channels to produce the generative component and the other half to produce the confidence map. The results in Table 7 show that HEM can help improve the performance of different networks, which is especially

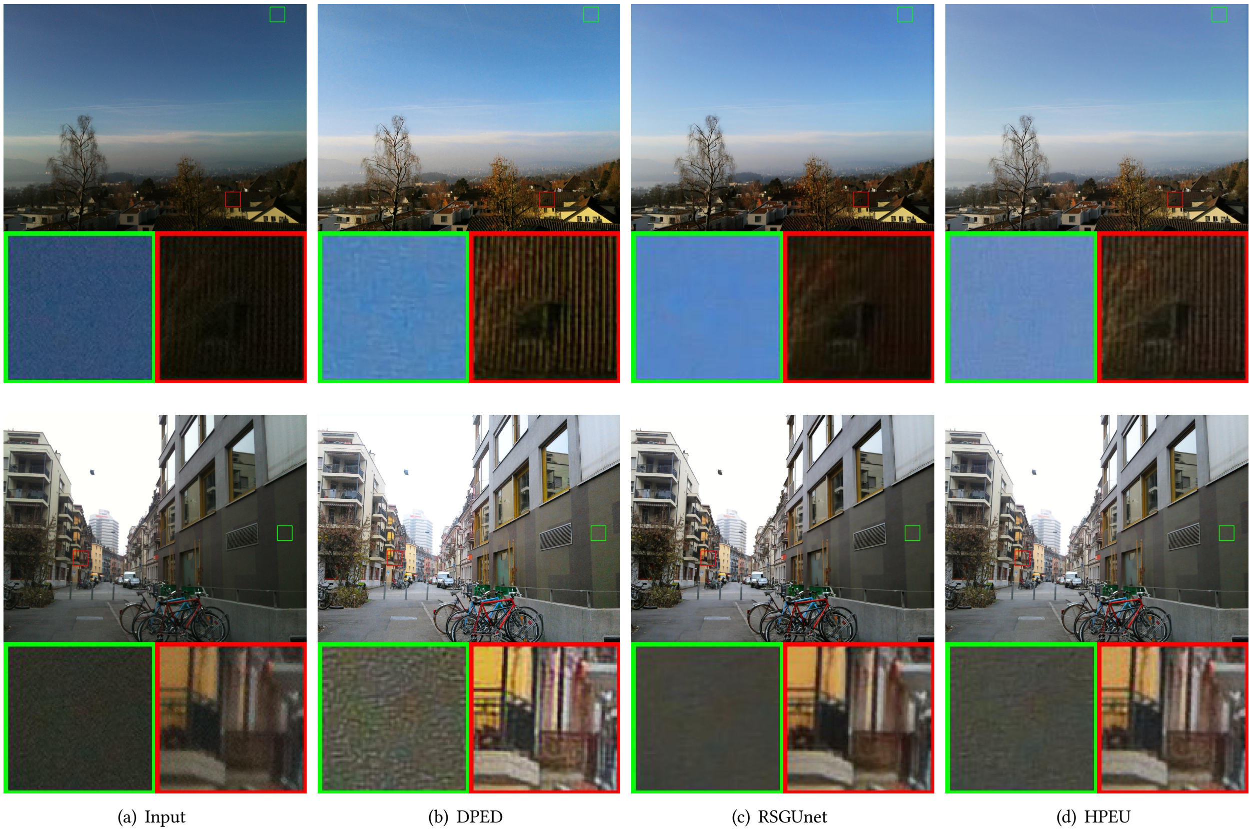


Figure 10: Visual results from the full-size DPED-iphone test images by using perceptual loss.

Table 7: Quantitative results of different methods before and after adding HEM on MIT-Adobe FiveK dataset.

	Original		After adding HEM	
Methods	PSNR	MS-SSIM	PSNR	MS-SSIM
DPED [13]	22.73	0.9282	22.84	0.9296
CAN [4]	22.71	0.9340	23.09	0.9407
DGF [25]	22.64	0.9353	22.73	0.9359

Table 8: Comparisons of LEU and ResBlock on MIT-Adobe FiveK dataset.

Methods	LEU-1	ResBlock-1	LEU-3	ResBlock-3
PSNR	20.87	20.75	20.9	20.71
Parameters	17K	17K	44K	44K

important to DGF without decelerating its speed as shown in Fig. 1.

Comparison of LEU with ResBlock. We further compare LEU with ResBlock [9] to demonstrate that LEU is more effective than the commonly used ResBlock. Specifically, we compare LEU and ResBlock in a simple cascaded network with 16 channels. As can be seen from the results in Table 8, LEU outperforms ResBlock with both 1 and 3 blocks in terms of PSNR, without introducing extra parameters.

5 CONCLUSION

We propose a novel hybrid network with the Laplacian Enhancing Unit (LEU) and the Hybrid Enhancing Module (HEM) for image enhancement. Different from previous CNN based methods, the proposed HEM jointly exploits the merits of scaling methods and generative methods to achieve an improved performance. The proposed LEU can progressively enhance the features on different levels and help the convolution layer better utilize the information. We integrate HEM and LEU under a U-Net architecture to form our lightweight HPEU, which has demonstrated excellent performance as well as high inference speed. In future work, we plan to extend our method to other image processing tasks.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China under Grants 2017YFA0700800 and 2017YFB1300201, the National Natural Science Foundation of China (NSFC) under Grants 61671419, 61622211, 61620106009 and 61525206 as well as the Fundamental Research Funds for the Central Universities under Grant WK2100100030.

REFERENCES

[1] Sarabjeet Singh Bedi and Ritvik Sanjay Khandelwal. 2013. Various image enhancement techniques-a critical review. *International Journal of Advanced Research in Computer and Communication Engineering* 2, 3 (2013), 1605–1609.

[2] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Fredo Durand. 2011. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*. 97–104.

[3] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. 2016. DehazeNet: An End-to-End System for Single Image Haze Removal. *IEEE Transactions on Image Processing* 25 (2016), 5187–5198.

[4] Qifeng Chen, Jia Xu, and Vladlen Koltun. 2017. Fast Image Processing With Fully-Convolutional Networks. In *ICCV*. 2516–2525.

[5] Yubin Deng, Chen Change Loy, and Xiaoou Tang. 2018. Aesthetic-Driven Image Enhancement by Adversarial Learning. In *ACM MM*. 870–878.

[6] KA Divya and KI Roshna. 2015. A Survey on Various Image Enhancement Algorithms for Naturalness Preservation. *International Journal of Computer Science and Information Technologies* 6, 3 (2015), 2043–2045.

[7] C. Dong, C. C. Loy, K. He, and X. Tang. 2016. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 2 (2016), 295–307.

[8] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. 2017. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 118.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.

[10] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. 2018. Exposure: A White-Box Photo Post-Processing Framework. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 26:1–26:17.

[11] Yuanming Hu, Baoyuan Wang, and Stephen Lin. 2017. FC4: Fully Convolutional Color Constancy With Confidence-Weighted Pooling. In *CVPR*. 4085–4094.

[12] Jie Huang, Pengfei Zhu, Mingrui Geng, Jiewen Ran, Xingguang Zhou, Chen Xing, Pengfei Wan, and Xiangyang Ji. 2019. Range Scaling Global U-Net for Perceptual Image Enhancement on Mobile Devices. In *ECCV Workshops*. 230–242.

[13] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. 2017. DSLR-quality photos on mobile devices with deep convolutional networks. In *ICCV*. 3277–3285.

[14] B. L. Kalman and S. C. Kwasny. 1992. Why tanh: choosing a sigmoidal function. In *IJCNN*. 578–581.

[15] Y. H. Lee and S. Y. Park. 1990. A study of convex/concave edges and edge-enhancing operators based on the Laplacian. *IEEE Transactions on Circuits and Systems* 37, 7 (1990), 940–946.

[16] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. 2017. AOD-Net: All-in-One Dehazing Network. In *ICCV*. 4780–4788.

[17] Chao Ma, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang. 2016. Learning a No-Reference Quality Metric for Single-Image Super-Resolution. *Computer Vision and Image Understanding* 158 (2016), 1–16.

[18] Andrew L. Maas. 2013. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *ICML*, Vol. 30.

[19] A. Mittal, R. Soundararajan, and A. C. Bovik. 2013. Making a âĂIJCompletely BlindâĂİ Image Quality Analyzer. *IEEE Signal Processing Letters* 20, 3 (2013), 209–212.

[20] Jongchan Park, Joon-Young Lee, Donggeun Yoo, and In So Kweon. 2018. Distort-and-Recover: Color Enhancement Using Deep Reinforcement Learning. In *CVPR*. 5928–5936.

[21] Mehdi S. M. Sajjadi, Bernhard Scholkopf, and Michael Hirsch. 2017. EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis. In *ICCV*. 4491–4500.

[22] Qi-Chong Tian and Laurent D. Cohen. 2017. Global and Local Contrast Adaptive Enhancement for Non-Uniform Illumination Color Images. In *ICCV Workshops*. 3023–3030.

[23] Xuewen Wang and Lixia Chen. 2018. Contrast enhancement using feature-preserving bi-histogram equalization. *Signal, Image and Video Processing* 12, 4 (2018), 685–692.

[24] Z. Wang, E. P. Simoncelli, and A. C. Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers*. 1398–1402.

[25] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. 2018. Fast End-to-End Trainable Guided Filter. (2018), 1838–1847.

[26] S. Wu, Q. Zhu, Y. Yang, and Y. Xie. 2013. Feature and contrast enhancement of mammographic image based on multiscale analysis and morphology. In *ICIA*. 521–526.

[27] Zhiwei Xiong, Zhan Shi, Huiqun Li, Lizhi Wang, Dong Liu, and Feng Wu. 2017. HSCNN: CNN-Based Hyperspectral Image Recovery From Spectrally Undersampled Projections. In *ICCV Workshops*. 518–525.

[28] Zhicheng Yan, Hao Zhang, Baoyuan Wang, Sylvain Paris, and Yizhou Yu. 2016. Automatic Photo Adjustment Using Deep Neural Networks. *ACM Transactions on Graphics (TOG)* 35 (2016), 11:1–11:15.

[29] Wu J. Yang, F. 2010. An improved image contrast enhancement in multiple-peak images based on histogram equalization. In *International Conference On Computer Design and Applications*. 346–349.

[30] S. Yun, J. H. Kim, and S. Kim. 2010. Image enhancement using a fusion framework of histogram equalization and laplacian pyramid. *IEEE Transactions on Consumer Electronics* 56, 4 (2010), 2763–2771.

[31] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *ECCV*. 818–833.

[32] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. 2017. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing* 26, 7 (2017), 3142–3155.