Guidance Network with Staged Learning for Image enhancement

Luming Liang Ilya Zharkov Faezeh Amjadi Hamid Reza Vaezi Joze Vivek Pradeep
Microsoft
One Microsoft Way, Redmond WA, USA
{lulian,zharkov,faamja,hava,vpradeep}@microsoft.com

Abstract

Many important yet not fully resolved problems in computational photography and image enhancement, e.g. generating well-lit images from their low-light counterparts or producing RGB images from their RAW camera inputs share a common nature: discovering a color mapping between input pixels to output pixels based on both global information and local details. We propose a novel deep neural network architecture to learn the RAW to RGB mapping based on this common nature. This architecture consists of both global and local sub-networks, where the first sub-network focuses on determining illumination and color mapping, the second sub-network deals with recovering image details. The result of the global network serves as a guidance to the local network to form the final RGB images. Our method outperforms state-of-the-art with a significantly smaller size of network features on various image enhancement tasks.

1. Introduction

Modern cameras sequentially perform many signal processing steps to reconstruct an RGB image from a RAW sensor input, which has only 1 color per pixel, either red, green or blue. This sequential processing pipeline, namely Image Signal Processing (ISP) is complicated, which consists of various stages, for example, defect pixel removal, denoising [2], demosaicing [17], gamma correction, white balancing [13] and so on. Different ISP pipelines have to be tuned by different groups of well-trained camera experts for a relatively long time before it can be used in the commercial cameras. Domain knowledge such as optics, mechanics of the cameras, electronics and human perception of colors and contrast are necessary in this tuning process. Replacing this highly skilled and tedious tuning process with deep neural network is a recent research direction in computational photography [26, 24, 12]. In addition to replacing the ISP with machine learning models, generating well-lit images from their low-light version is another interesting direction in this field [20, 21, 5, 4, 33, 39, 28]. The problem suffers from low signal-to-noise ratio and lack of semantic content. Since only an extremely low number of photons can be captured by the camera with low exposure time in already very dark scenarios, substantial noise contamination is inevitable. More than 1 type of noise exists in this low light RAW images, including dark pattern noise, Poisson noise and scan line noise. In the meantime, color mapping schema from the RAW image to final RGB channels is hard to learn because of lack of content.

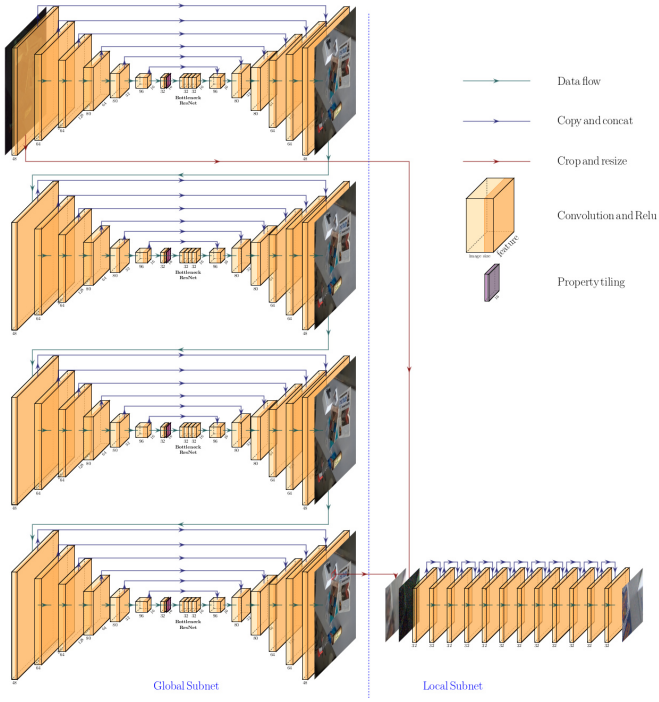


Figure 1. Network architecture

We propose to solve these two problems with a common architecture. The network first learns color and illumination mapping in a resized domain, then learns image detail recovery with the patches cropped from full resolu-

tions. We name the first part as the global sub-network, which is a stacked U-net, while the second part as the local sub-network, which consists of several residual blocks. This design enables our network to be used with arbitrary resolution.

Our main contributions include

- Learning method of replacing the ISP and en-lighting extreme low light images with a common deep neural network architecture (Figure 1), which achieves the state-of-the-art with significantly lower number (about 20%) of features;
- Use stacked U-nets to learn global mappings as the guidance network, which is then used to guide the local color mapping in the detail recovery stage, see Section 3.2;
- A new training method for the local sub-network with introducing intermediate ground truth, see Section 3.3. This novel training method yields a 0.3db peak signal to noise ratio increase according to the ablation study with an exactly same network architecture.

2. Related Work

Computational photography is an active topic across computer vision, graphics, image processing and machine learning. Research in this domain is always closer to real world applications and can be integrated to consumer electronics more naturally. For example, different to traditional image processing, i.e. RGB image denoising [34, 16, 35, 31, 15], denoising in the RAW domain [22, 10, 1, 9] has a wider usage in the current industrial pipeline. We focus on research works that take RAW as input, generating RGB as output, and briefly review them in this section.

Difficulties in computational photography research can be summarized into 2 aspects:

1. ISP tuning requires a lot of domain knowledge, and different ISPs may produce totally different results. It is not trivial to reproduce the full ISP pipeline with a single end-to-end method without understanding the details inside. Subjective quality metrics and heuristic rules used during ISP design complicates the algorithmic replication even further[26].

2. RAW images always contain different kinds of capture noise, and which pattern is unknown. Solutions based presumably specific kind of noise, like Additive White Gaussian Noise (AGWN), for example methods like [34] and [35] will fail on real image denoising. This inability is widely discussed in [10, 1].

Generating well-lit images from their low light counterpart is a well-studied but yet unresolved topic, many traditional efforts aim at finding bidirectional correspondences between them, for example, dual illumination [36] and perceptual similarity [37]. These methods often have many heuristics and are not applicable to extreme, less than 0.1 second exposure in the dark scene, low light. Learning based methods [20, 21, 29, 5, 39, 28, 11, 19, 4] are of the mainstream in low light image recovery. Within these networks, only [5, 39, 28, 33, 19, 4] take RAW images captured with extreme low light scenarios, we focus on them. Chen et al. [5] release See In the Dark (SID) dataset and propose a huge u-net to learn the recovering process, later they extend this work to video [4]. Liba et al. [19] present an extreme low-light recovery method based on merging a set of burst images. Zhu et al. [39] propose an edge-aware model to better learn edge details in the recovery. Zamie et al. [33] use the exactly same architecture as Chen et al.'s [5], but adding a perceptual loss based on VGG19 to better recover the context of the images. Wang et al. [28] adopts a model similar to Lv et al.'s model [21], which consists of a set of multi resolution feature extractions and fusions. In a more recent work, [32] tries to recover image information according to different frequency components, where the low frequency information is created via a Gaussian filter. All these methods report their performance on SID data set [5], we compare our method with them also in SID data set, details are shown in Section Evaluation.

Replacing the expert-tuned ISP with a fully automatic method relies more and more on deep learning, all of the recent methods approach it by training an end-to-end deep neural network [26, 24, 12]. Schwartz et al. [26] release a data set, named Samsung S7 data set, contains RAW and RGB image pairs with both short and medium exposures. They design a network that first process the image locally then globally. Ratnasingam [24] replicates the steps of a full ISP with a group of sub networks, achieves the-state-of-the-art by training and testing on a set of synthetic images.

Recently, training a network with 2 stages becomes common in RAW image to RGB transformation. DeepISP[26], CameraNet [18] and Decomposition-and-Enhancement network[32] all adopt this strategy: 1 sub-network for detail recovery (usually just denoising) and 1 sub-network for color and luminance recovery. Our network also follows this line of ideas, however, we have very unique design here:

1. Opposite to CameraNet and DeepISP, we perform color recovery before denoising, which is similar to Decomposition-and-Enhancement network, learns to recovery low frequency component before higher frequency component;
2. Our method recovers RGB information in a fixed resized domain, which is distinct to all other state-of-the-art, and proved to be efficient in Evaluation.
3. Different to Decomposition-and-Enhancement network, we recover the image high frequency details by

taking both the original raw image and the recovered low frequency image as inputs to the detail recovery sub-network. This choice is based on a natural observation: although contaminated by the noise, the orignal input contains many useful local details. Furthermore, by adopting this scheme, we can further take advantage of progressive training shown later in Section Methodology.

3. Methodology

In some commercial ISP, complex algorithms related to overall color and brightness histograms, type of scene and light source are performed on *full* but *resized* to smaller-resolution images. On the contrary, some other relatively simple algorithms, e.g. defect pixel detecting and replacing, denoising, demosacing and deblurring, are performed on *individual* pixels. We named the first ones global operations and the second ones local operations. These local operations involve only a limited number of neighboring pixels, i.e. filters with compact support.

We follow this principle: use a more complex low resolution guidance network to reconstruct a low resolution target image; use a high resolution correction network to obtain a final RGB images from high resolution input RAW images and up-sampled outputs of the guidance network. This idea is common to 2 different computational photography tasks: low-light image recovery and learning the ISP.

There exist several architectures applying similar approaches [25, 6, 8, 29]. Chen et al. [6] uses target low resolution image as a guidance to modify input high resolution image via bilateral filtering, which speedups slow algorithms by applying them at a smaller scale. This work is further extended with deep neural nets by Gharbi et al. in [8], namely HDRnet, where they use a CNN to generate a lookup table for different pixels positions and brightness levels. Wang and Zhang et al. [29] further extends in this path, by replacing the lookup table learning with illumination estimation [7] to recover the image from under exposure.

3.1. Network Architecture

We define our network architecture in Figure 1. The global (left) part takes resized raw images (usually 4 channels for Bayer pattern) as the input (for low light application, we also multiply the input raw with the exposure time ratio as described by [5]), output also a resized image with RGB 3 channels. In practice, we set the resize size as 512×512. With this choice, we can use deeper u-nets than [5].

Stacked u-nets with the exact same structure are adopted. Each u-net outputs a resized RGB image, and this intermediate output will be used as the input of the next u-net. The output of the last u-net will be cropped and resized and then stacked with the local input at exactly the same location. This pair of stacked images are the inputs of the local sub-net, as shown in Figure 1. Each level of encoder reduces both the height and the width by a factor of 2 and each level of decoder increases them in an opposite way. In the bottleneck of each u-net, we stack the features with the tiled properties (exposure rate for low light image recovery; lens position, digital gain, analogue gain and exposure time for ISP learning). Then, we feed the stacked features into a set of residual blocks. This design is based on observations: properties affect the global image quality, e.g. shorter exposure time or lower digital gain often leads to larger amount of noise; lens position affects the vignette effect.

The local sub-net is simply a group of residual blocks. This sub network takes the guidance–color corrected image patches, learnt from the global part to guide the process of local input patches. As shown in Figure 2, the learnt global patch, cropped and resized from the smaller resolution full image, has better color, but looks blurry, since the local information is missing, while in the meantime, the local input patch has more details around edges, but the color is incorrect and also noisy.

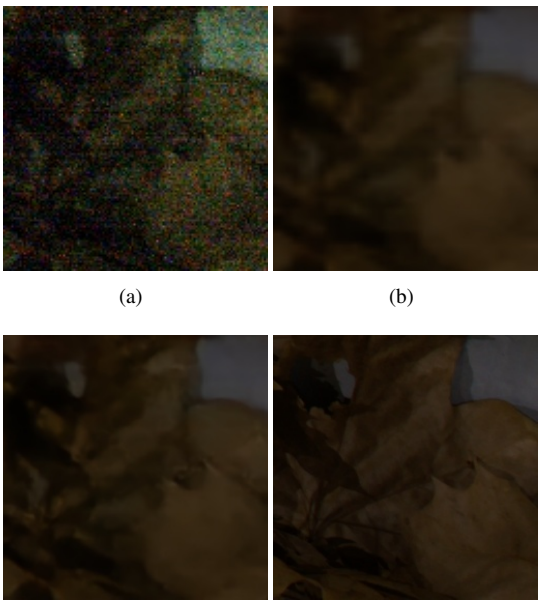


Figure 2. Local learning: by stacking (a) input patch with (b) cropped patch from the result image of the global guidance network, we learn a (c) more detailed output patch; (d) is the ground truth patch.

3.2. Training global network with staged learning

Our purpose of adopting a stacked u-net in the global learning is to let the network gradually learn the color mapping. Each u-net in the global network produces an inter-

mediate result of the resized RGB image. Here, we adopt the summation of l_1 distance and multi-scale structure similarity (MS-SSIM) [30] as the loss function of each u-net, suggested by [38]. We simply aggregate the loss of each stage to form the global loss L_g as:

$$L_i(\hat{I}_i, I) = ||\hat{I}_i - I|| + 0.5 \times (1 - MSSSIM(\hat{I}_i, I)),$$
$$L_g = \sum_i L_i, \tag{1}$$

where $\hat{I}_i$ is the estimated resized RGB image from u-net i, I is the resized ground truth image.

The apparent visual improvement through the stacked u-nets is shown in Figure 3. This learning schema reduces the harder problem into several simpler problems, which is originated from [23, 3]. With this global training schema, we do not need to put all pixels from the original image into the network in the training time, or crop a relatively large patch (1024×1024 in [26]) to let the network be able to learn the global transformation in ISP.

3.3. Progressive train local network with resized gt

The local sub-network takes 2 inputs, RAW input and RGB from the global guidance, outputs the final recovered image. We hope the final result gets as close as possible to the ground truth RGB image with the training process. In addition, this sub-network has a fully-convolutional style, therefore, our network can be applied on images with arbitrary sizes. To achieve these goals, we adopt 2 novel training methods:

1. As shown in Figure 1, we crop and resize the result from the output of the guidance network to the original size. Then we stack this crop and resized result with the RAW input. The benefits of performing this training is that we save a lot of GPU memory during the training. This idea is originated from Gharbi et al. [8]. However, instead of directly fusing global features and local features in [8] by a slicing layer, we use crop and resize to perform the global guidance.

2. To better utilize global guidance, we train this local sub-network by alternating the guidance input with cropped resized ground truth. We use the same weight of the local sub-network, while working in tandem on resized truth and the current guidance image. As the guidance image gets closer and closer to the ground truth, the training of the local network will be converge. This idea is similar to progressive growing of GANs [14]. The effectiveness of this training is shown in the ablation study.

We define the loss function of the local training as

$$L_l = ||\hat{I}_p - I_p|| + 0.5 \times (1 - MSSSIM(\hat{I}_p, I_p)) + ||\hat{I}_p - G_p|| + 0.5 \times (1 - MSSSIM(\hat{I}_p, G_p)), \tag{2}$$

where $\hat{I}_p$ is the estimated image patch, I_p is the ground truth image patch and G_p is the image patch from the global result, which is the last u-net output.

3.4. Final loss function

We train the full network shown in Figure 1 together with a combined loss function:

$$L = L_g + L_l,$$

where L_g and L_l are defined in Equation (1) and (2), respectively.

4. Evaluation

4.1. Data set and training setup

We evaluate our method on 3 different data sets:

- **See-In-the-Dark (SID)**: proposed by [5], is a Raw-RGB dataset captured in extreme low-light conditions, where each short-exposure raw image is paired with its long-exposure RGB counterpart for training and testing [33]. Images in this dataset were captured using two cameras: Sony $\alpha 7SII$ (raw: GRBG 4-channel Bayer) and Fujifilm X-T2 (raw: 9-channel XTrans), each subset contains about 2500 images, with about 20% of them are test images. Besides RAW and RGB data, their exposure times are provided alongside.
- **Samsung S7**: captured by [26], contains 110 different RAW-RGB pairs, with train/test/validation split as 90/10/10. Different to SID, this one does not provide related camera properties.
- **ImagePairs**: captured by [27]. For each image, the meta data such as gain, exposure, lens position and scene categories were stored. The data set is divided into train and test sets including 8591 (75%) and 2830 (25%) image, respectively. The ISP used to process the RAW input is named ARC, which contains more than 20 steps, including both global steps like lens shading correction and auto white balancing and local steps like denoising and defect pixel repairing. In this experiment instead of using LR/HR pair we use LR and its raw information in order to train an ISP.

For all these 3 data set, we train our model with 2 pairs of inputs and outputs: resized RAW/RGB and cropped patches of RAW/RGB, using above-mentioned loss function. We

input u-net 1 u-net 2 u-net 3 u-net 4 gt

Figure 3. Global learning: stacked u-nets gradually refine the output resized image to make it approach the ground truth (gt).

choose Adam as the optimizer with an exponetially decay learning rate, defined as:

$$lr = 0.001 \times 0.5^{\frac{iter}{10000}},$$

where *iter* is the number of iterations. Batch size is usually set to 6 or 8 (almost no difference on the final result after the convergence in practice), depends on the memory limit. The training is conducted on a workstation with GPU Titan RTX 24G. For the sake of comparison, we also implemented Schwartz et al.'s DeepISP method [26] and Chen et al.'s SID method [5] with smallest necessary changes.

4.2. Comparisons

Peak Signal to Noise Ratio (PSNR) and Structural SIMilarity (SSIM) index are used as quantitative performance criteria. In addition, we also compare the model size, i.e. the amount of features.

4.2.1 SID data set

Detailed performance comparisons on SID [5] data set can be found in Table 1. Although each sub data set (Sony and Fuji) contains about 1800 training images, the data size is still very limited. Since many training images are captured under burst mode, there are only about 180 different scenes. As noted by Chen et al. [5], it is very easy to run into overfitting. To alleviate this problem, we also adopted data augmentations used by Chen et al. [5]: flip horizontally and flip vertically. However, we did not use rotation by 90 degrees since there exist apparent horizontal noise patterns in the RAW inputs.

Our method generates vivid and also closer color (Figure 4(i)) to the ground truth then [26] and [5]. In addition, results produced by our model do not have unpleasant color artifact, e.g. 4(c) and (h).

Implementation details of other methods:

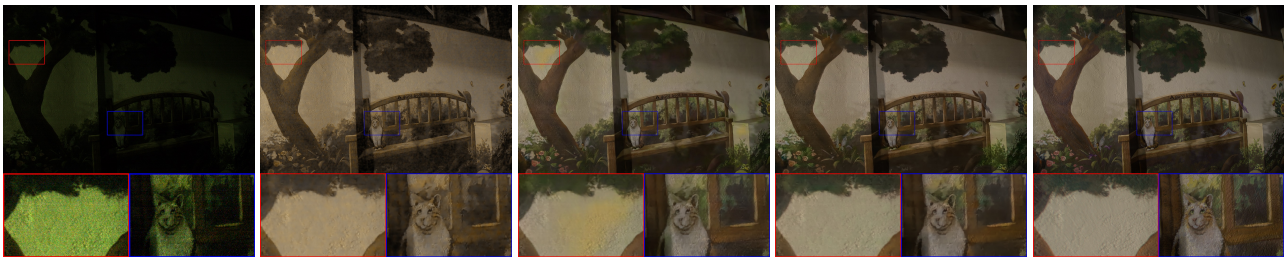
1. We implemented and trained DeepISP[26] with 2 small changes: 1) multiply the exposure ratio rate upon the 4-channel RAW input; 2) use this 4-channel RAW images as inputs to the network, add a deconvolution layer before their local branch, since their local branch only takes 3-channel input. We also randomly crop 1024×1024 patches to train, as same as their paper suggested [26]. The number of residual blocks is set to 16 with the feature size of each convolution layer set to 64.

2. Results from [5, 33, 39] are reported in their papers and related github pages. Results from DeepUPE and EEMEFN and reported in [32].

3. Wang et al. [28] achieves similar performance (29.79 on Sony subset) in PSNR by first training their model on a large set of synthetic images (more than 10k pairs) produced by [21] and then fine tune on SID data set, which avoids the over-fitting problem. In addition, we failed to find the implementation details of their method in [28], therefore, we do not list this method in Table 1.

4.2.2 Samsung S7 data set

We train both our network and the DeepISP network [26] on the training set (first 90 image pairs) of Samsung S7 data set. We train DeepISP network strictly following the settings in [26] with randomly cropping 1024×1024 patches from the images. The number of residual blocks is set to 16 with the feature size of each convolution layer set to 64. The input is a short exposure RAW image, the output is a well-lit medium exposure RGB image. The exposure time ratio is 4. We use this ratio as the property input to our network. We perform 3 different kinds data augmentations: flip horizontally, flip vertically and rotation by 90 degrees, since no apparent noise shown in this data set, see the left



(a) RAW (b) DeepISP (c) SIDnet (d) ours (e) GT

(f) RAW (g) DeepISP (h) SIDnet (i) ours (j) GT

Figure 4. Qualitative comparisons on SID: red and blue boxes are magnified in the bottom of their corresponding images.

Table 1. Comparisons on SID data set. Best in red and the second best in blue.

Model	# parameters	Size%	Sony		Fuji	
			PSNR (db)	SSIM	PSNR (db)	SSIM
SIDnet[5]	7724748	100%	29.18	0.79	27.34	0.68
PerNet[33]	7724748	100%	29.43	N/A	27.63	N/A
DeepISP[26]	668560	8.65%	26.20	0.85	22.29	0.78
DeepUPE[29]	N/A	N/A	29.13	0.79	N/A	N/A
EEMEFN[39]	N/A	N/A	29.60	0.80	27.38	0.72
Decomposition-and-Enhancement[32]	N/A	N/A	29.56	0.80	N/A	N/A
Our method	3329333	43.09%	29.73	0.89	28.11	0.85

most column in Figure 5. As shown in Table 2, we achieve a 1.64db gain on PSNR and 0.01 gain on SSIM over DeepISP network on this data set.

Table 2. Comparisons on Samsung S7 data set.

Model	# parameters	PSNR (db)	SSIM
DeepISP	668560	26.38	0.91
Our method	3329333	28.02	0.92

Figure 5 illustrates the visual comparisons between our results and results of DeepISP network [26], here both images are chosen from the test set. The results of our method show closer colors to the ground truth.

4.2.3 ImagePairs data set

We trained DeepISP net [26], SID net [5] and our baseline network on ImagePairs data set, which contains about 11000 image pairs. All networks read RAW images and associated 4 camera properties: analogue gain, digital gain, exposure time and lens position. Here, the exposure time is in micosecond; the lens position is the distance between the camera and the scene, in centimeters. When modifying DeepISP net [26] for this task, we tile and concatenate these 4 features with the output of their local sub-network and feed it to the global sub-network to estimate the quadratic transformation coefficients. For SID net [5], we simply tile and concatenate 4 features with the input image, since SID net is just a huge U-net.

Table 3. Comparisons on ImagePairs data set. Best in red, the second best in blue.

Model	# parameters	PSNR (db)	SSIM
DeepISP	668560	20.30	0.89
SIDnet	7724748	23.08	0.90
Our method	3329333	29.22	0.96

Table 3 illustrates the metrics of three models. Our baseline model outperforms at least 6db on the average of more than 2500 real images.

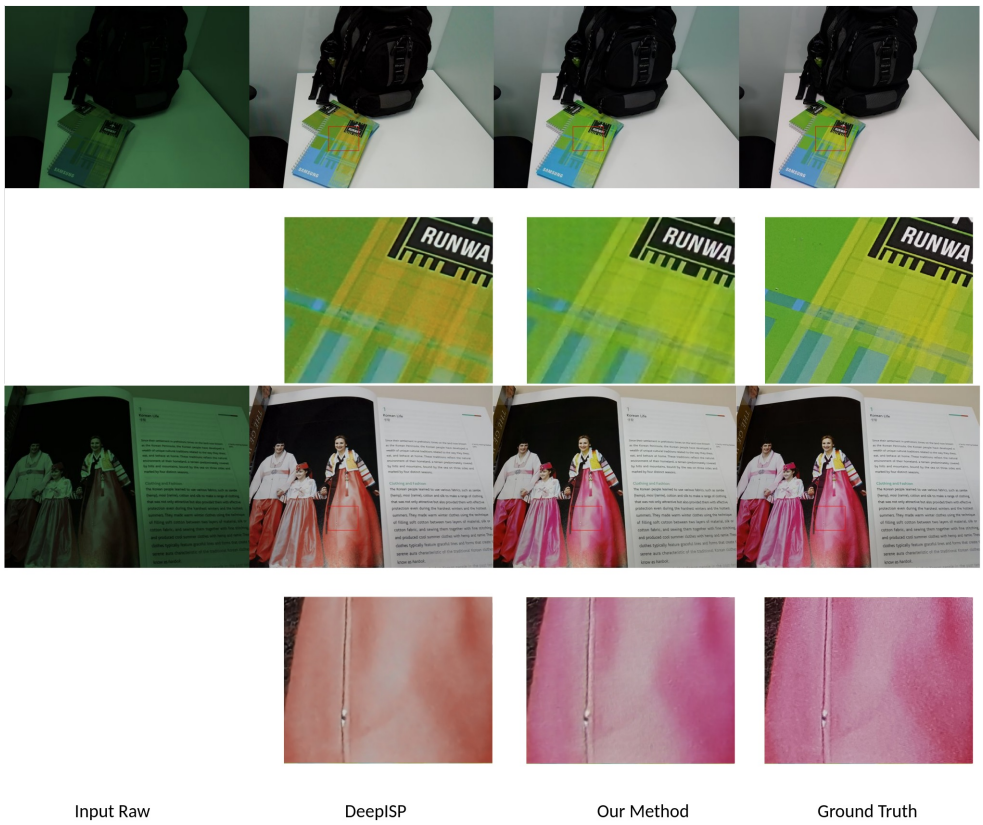


Figure 5. Visual comparisons on Samsung S7 data set between DeepISP network [26] and our method. Sub images in the red box are zoomed in for a more detailed comparison.

Figure 6 shows the visual comparisons of these methods on various image categories. DeepISP net [26] has a hard time to learn the color mapping, since they confine this mapping into a quadratic function. Results of SID net [5] contains block-wise visual artifacts that are similar to results in SID data set shown in Figure 4.

4.3. Ablation study

We conduct an ablation study on our baseline model by removing different parts of it, including using less stacked u-nets in the global sub-network and removing the progressive local training.

Table 4. Ablation study on SID data set. Best in red and the second best in blue, PSNR (db)/SSIM are shown.

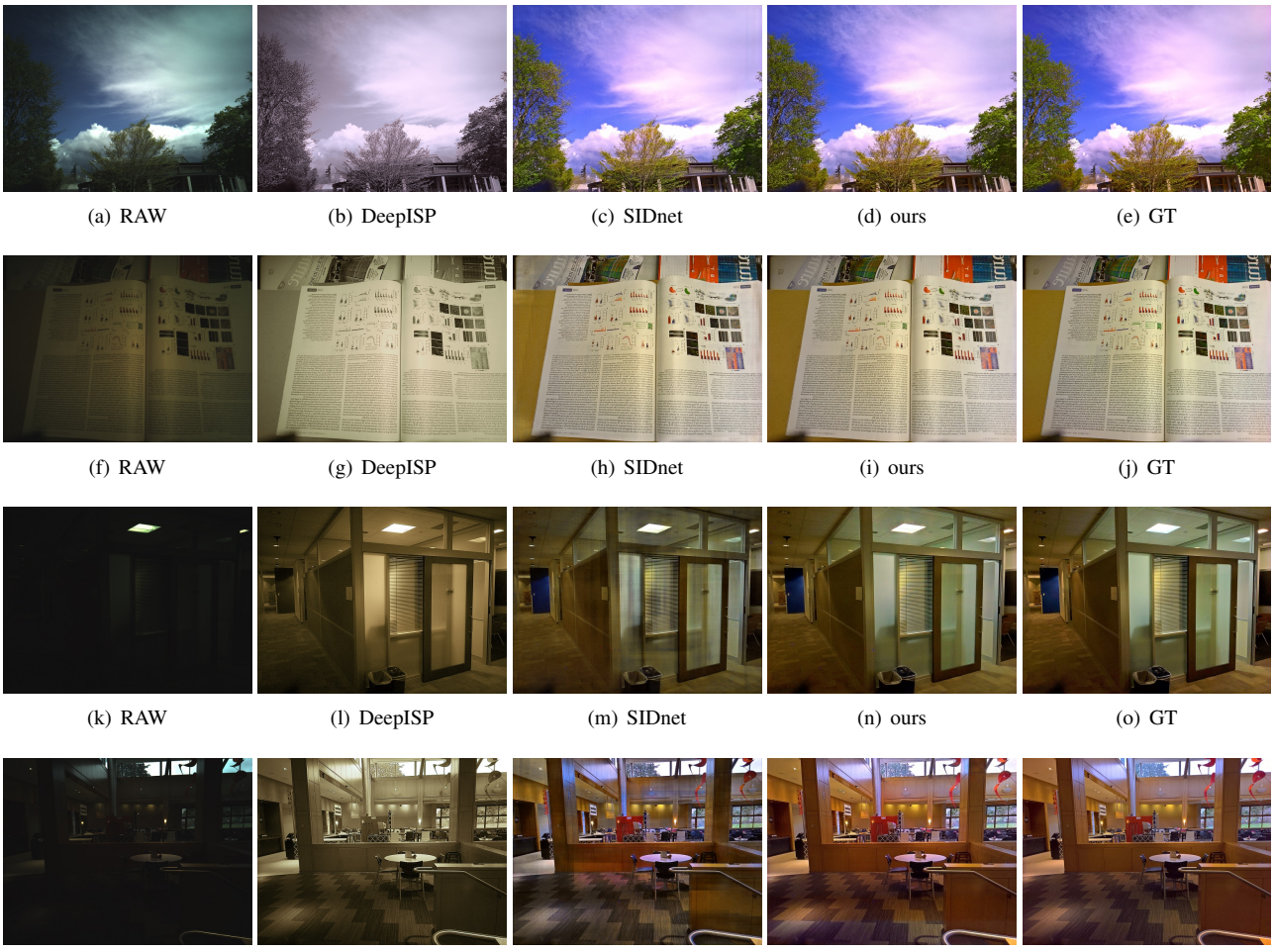
Model	# param	Sony	Fuji
4 u-nets	3.3M (43.09%)	29.72/0.89	28.11 /0.85
no local gt	3.3M (43.09%)	29.43/0.89	28.06/0.85
3 u-nets	2.5M (32.76%)	29.56/0.89	28.00/0.85
2 u-nets	1.7M (22.43%)	29.43/0.89	27.90/0.85
1 u-net	0.9M (12.09%)	29.07/0.88	27.78/0.85

Here, all comparisons are performed on both subsets of SID data set [5] with exactly the same training settings described before. 100% of Column Size% is defined in Table 1. According to Table 4, our baseline achieves the best performance on both subsets. For Sony subset, removing 1 u-net will reduce PSNR by 0.2db in average; while for Fuji subset, this amount is about 0.1db. On the local part, the progressive training schema gives the model a boost of 0.3db on Sony subset and 0.05db boost on Fuji set.

If we compare the results in Table 4 and those in Table 1, our smallest model (1 u-net) has a relative same size of the DeepISP model [26], but achieves way better performances on both subset, i.e. 3db increase on Sony, 5db increase on Fuji.

To further investigate the effectiveness of different components and the choice of the loss functions, we conduct another set of ablation studies on Sony subset in SID data set, as shown in Table 5. Note that our baseline model uses 4 unets in the global guidance sub network, 8 residual blocks in the local refinement sub network; the basic loss function is the addition of L1 loss and ssim loss, as shown in Eq. 1 and Eq. 2. Terms "resized from global" in Table 5 are the metrics of results generated by the guidance networks (usually blurry). Each row in Table 5 denotes a model that changes at most 1 component to the baseline model.

We conduct 2 ablation studies on the guidance network, 2 studies on the local refinement network and other 2 studies



(a) RAW (b) DeepISP (c) SIDnet (d) ours (e) GT

(f) RAW (g) DeepISP (h) SIDnet (i) ours (j) GT

(k) RAW (l) DeepISP (m) SIDnet (n) ours (o) GT

(p) RAW (q) DeepISP (r) SIDnet (s) ours (t) GT

Figure 6. Qualitative comparisons on 4 images with different categories: Tree, Document, Office and Cafeteria, chosen from the test set of ImagePairs data set.

Table 5. Further ablation study on SID data Sony subset.

Model	PSNR/SSIM	PSNR/SSIM resized global
Baseline	29.73/0.89	28.60/0.87
4 unets $\rightarrow$ 5 u-nets	29.76/0.89	28.64/0.87
remove property layer	28.92/0.88	27.93/0.86
loss $\rightarrow$ L1	29.33/0.88	28.29/0.87
loss $\rightarrow$ L2	29.08/0.88	28.13/0.87
8 $\rightarrow$ 4 resblocks	29.44/0.89	28.39/0.87
8 $\rightarrow$ 2 resblocks	29.48/0.89	28.50/0.87

on the loss functions. On the guidance part, we observe that using 1 more unet in the guidance sub network will slightly increase the performance comparing to the baseline; by removing the property tiling layer from the bottlenecks of the unets, we see a large drop on performances. On the local part, By switching the current choice of the loss function to some other forms (L1 solely, L2 solely) will decrease the performance. This observation tells us the current form of the loss function is effective.

5. Conclusion

The common architecture proposed for transforming RAW to RGB images takes advantage of the ideas and experiences from the traditional camera ISP pipeline: gradually learning mappings in the global domain and then fix the local features with fine tuning. This architecture consists of both global and local sub-networks, where the first sub-network focuses on determining illumination and color mapping, the second sub-network deals with recovering image details. The result of the global network serves as a guidance to the local network to form the final RGB images. Evaluations prove the effectiveness of this idea alongside with our training methods. Our method outperforms state-of-the-art with a significantly smaller size of network features on various image enhancement tasks.

References

[1] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing images for learned raw denoising. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[2] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005. 1

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2018. 4

[4] Chen Chen, Qifeng Chen, Minh Do, , and Vladlen Koltun. Seeing motion in the dark. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[5] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 3, 4, 5, 6, 7

[6] Jiawen Chen, Andrew Adams, Neal Wadhwa, and Samuel W. Hasinoff. Bilateral guided upsampling. *ACM Transactions on Graphics*, 35(6):1–8, Nov. 2016. 3

[7] Mark S Drew, Hamid Reza Vaezi Joze, and Graham D Finlayson. The zeta-image, illuminant estimation, and specularity manipulation. *Computer Vision and Image Understanding*, 127:1–13, 2014. 3

[8] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4), 2017. 3, 4

[9] Shuhang Gu, Yawei Li, Luc Van Gool, and Radu Timofte. Self-guided network for fast image denoising. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[10] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[11] X. Guo, Y. Li, and H. Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, Feb 2017. 2

[12] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. *arXiv preprint arXiv:2002.05509*, 2020. 1, 2

[13] Hamid Reza Vaezi Joze and Mark S Drew. Exemplar-based color constancy and multiple illumination. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):860–873, 2013. 1

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, April 2018. 4

[15] Yuma Kinoshita and Hitoshi Kiya. Convolutional neural networks considering local and global features for image enhancement. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2110–2114, 2019. 2

[16] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *International Conference on Machine Learning (ICML) 2018*, pages 2971–2980, 2018. 2

[17] Xin Li, Bahadir Gunturk, and Lei Zhang. Image demosaicing: A systematic survey. In *Visual Communications and Image Processing 2008*, volume 6822, page 68221J. International Society for Optics and Photonics, 2008. 1

[18] Zhetong Liang, Jianrui Cai, Zisheng Cao, and Lei Zhang. Cameranet: A two-stage framework for effective camera isp learning, 2019. 2

[19] Orly Liba, Kiran Murthy, Yun-Ta Tsai, Tim Brooks, Tianfan Xue, Nikhil Karnad, Qiurui He, Jonathan T. Barron, Dillon Sharlet, Ryan Geiss, Samuel W. Hasinoff, Yael Pritch, and Marc Levoy. Handheld mobile photography in very low light. *ACM Trans. Graph.*, 38(6):164:1–164:16, 2019. 2

[20] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognition*, 61:650–662, 2017. 1, 2

[21] Feifan Lv, Feng Lu, Jianhua Wu, and Chongsoon Lim. Mbllen: Low-light image/video enhancement using cnns. In *British Machine Vision Conference (BMVC)*, 2018. 1, 2, 5

[22] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[23] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 483–499, September 2016. 4

[24] Sivalogeswaran Ratnasingam. Deep camera: A fully convolutional neural network for image signal processing. In *International Conference on Computer Vision Workshops (ICCVW)*, August 2019. 1, 2

[25] Yaniv Romano, John Isidoro, and Peyman Milanfar. Raisr: Rapid and accurate image super resolution, 2016. 3

[26] Eli Schwartz, Raja Giryes, and Alexander M. Bronstein. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28:912–923, 2019. 1, 2, 4, 5, 6, 7

[27] H. R. Vaezi Joze, I. Zharkov, K. Powell, C. Ringler, L. Liang, A. Roulston, M. Lutz, and V. Pradeep. Imagepairs: Realistic super resolution dataset via beam splitter camera rig. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2190–2200, 2020. 4

[28] Lei Wang, Guangtao Fu, Zhuqing Jiang, Guodong Ju, and Aidong Men. Low-light image enhancement with attention and multi-level feature fusion. In *IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 276–281, July 2019. 1, 2, 5

[29] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 6

[30] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 4

[31] Chao Dong Xintao Wang, Ke Yu and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[32] Ke Xu, Xin Yang, Baocai Yin, and Rynson W.H. Lau. Learning to restore low-light images via decomposition-and-enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 5, 6

[33] Syed Waqas Zamir, Aditya Arora, Salman Khan, Fahad Shahbaz Khan, and Ling Shao. Learning digital camera pipeline for extreme low-light imaging. Technical report, ArXiV, 2019. 1, 2, 4, 5, 6

[34] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 2

[35] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for CNN based image denoising. *IEEE Transactions on Image Processing*, 2018. 2

[36] Qing Zhang, Yongwei Nie, , and Wei-Shi Zheng. Dual illumination estimation for robust exposure correction. *Computer Graphics Forum (Proceedings of Pacific Graphics 2019)*, 2019. 2

[37] Qing Zhang, Yongwei Nie, Lei Zhu, Chunxia Xiao, and Wei-Shi Zheng. Enhancing underexposed photos using perceptually bidirectional similarity. *Tech report, arXiv*, October 2019. 2

[38] Hang Zhao and Orazio Gallo ans Iuri Frosio ans Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3:47–57, 2017. 4

[39] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Eemefn: Low-light image enhancement via edge-enhanced multi-exposure fusion network. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2019. 1, 2, 5, 6