CameraNet: A Two-Stage Framework for Effective Camera ISP Learning

Zhetong Liang, Jianrui Cai, Zisheng Cao, and Lei Zhang, *Fellow, IEEE*

***Abstract*— Traditional image signal processing (ISP) pipeline consists of a set of cascaded image processing modules onboard a camera to reconstruct a high-quality sRGB image from the sensor raw data. Recently, some methods have been proposed to learn a convolutional neural network (CNN) to improve the performance of traditional ISP. However, in these works usually a CNN is directly trained to accomplish the ISP tasks without considering much the correlation among the different components in an ISP. As a result, the quality of reconstructed images is barely satisfactory in challenging scenarios such as low-light imaging. In this paper, we firstly analyze the correlation among the different tasks in an ISP, and categorize them into two weakly correlated groups: restoration and enhancement. Then we design a two-stage network, called CameraNet, to progressively learn the two groups of ISP tasks. In each stage, a ground truth is specified to supervise the subnetwork learning, and the two subnetworks are jointly fine-tuned to produce the final output. Experiments on three benchmark datasets show that the proposed CameraNet achieves consistently compelling reconstruction quality and outperforms the recently proposed ISP learning methods.**

***Index Terms*— Image signal processing, image restoration, image enhancement, convolutional neural networks.**

I. Introduction

THE raw image data captured by camera sensors are typically red, green and blue channel-mosaiced irradiance signals containing noise, less vivid colors and improper tones [1], [2]. To reconstruct a displayable high-quality sRGB image, an in-camera image signal processing (ISP) pipeline is generally required, which consists of a set of cascaded components, including color demosaicking, denoising, white balance, color space conversion, tone mapping and color enhancement, etc. The performance of an ISP plays the key role to improve the quality of sRGB images output from a camera.

Manuscript received June 8, 2020; revised October 2, 2020; accepted January 6, 2021. Date of publication January 20, 2021; date of current version January 27, 2021. This work was supported by the Hong Kong Research Grants Council (RGC) Research Impact Fund (RIF) under Grant R5001-18. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Damon M. Chandler. *(Corresponding author: Lei Zhang.)*

Zhetong Liang, Jianrui Cai, and Lei Zhang are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csztliang@comp.polyu.edu.hk; csjcai@comp.polyu.edu.hk; cslzhang@comp.polyu.edu.hk).

Zisheng Cao is with DJI Company Ltd., Shenzhen 518057, China (e-mail: zisheng.cao@dji.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2021.3051486, provided by the authors.

Digital Object Identifier 10.1109/TIP.2021.3051486

The traditional ISP is usually designed as a set of hand-crafted modules, each of which addresses a specific task [1]. For instance, a 3D lookup table is typically employed for the color enhancement task [2]. In most traditional ISP models, the modules are designed in a divide-and-conquer manner (i.e., splitting the ISP into a set of modules and developing them independently), while little attention has been paid to design them as a whole [3]. Moreover, it is time-consuming to tune each module for high image quality since the best output of one module may not result in the desired quality of the final output. Besides the standard ISP pipeline, there are also some ISP methods designed for burst imaging in the literature [4], [5]. However, these methods are subject to the effectiveness of image alignment techniques [6], which may generate ghost artifacts caused by object motion.

Recently, it has been shown that the performance of some image processing tasks, such as denoising [7], [8], white balance [9], [10], color demosaicking [11], [12], color enhancement [13]–[15], etc, can be significantly improved by deep learning techniques. In these methods, a convolutional neural network (CNN) is trained with a task-specific dataset that contains image pairs for supervised learning. Inspired by these methods, an intuitive idea is that we can train a subnetwork for each subtask of the ISP pipeline, and then chain them together as a whole ISP network. However, this is still a divide-and-conquer strategy as used in the traditional ISP design, which is cumbersome and ineffective. First, it is difficult and expensive to construct a dataset which has a ground truth for each subtask in the ISP. If we use different task-specific datasets to train different subnetworks separately, errors will be accumulated as in traditional ISP. Second, training a subnetwork for each subtask will make the whole network very heavy and complex. Third, some subtasks in an ISP are not independent but correlated. It has been verified that for correlated tasks, it is more effective to treat them jointly and train a shared network for them [12], [16], [17].

Instead of learning a subnetwork for each subtask, some works have been reported to directly train a CNN model for all ISP subtasks as a whole [18]–[20]. Like the many CNN methods for image denoising and super-resolution [8], [21], [22], in these works a single-stage network is straightforwardly trained as an ISP in an end-to-end manner. However, an ISP is a composition of multiple image processing tasks, some of which may not be correlated too much with each other. Directly training them as a whole may make

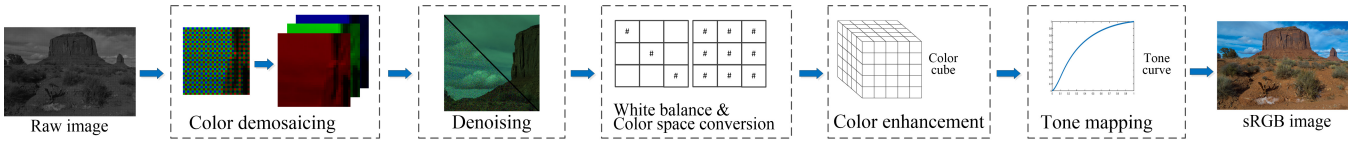


Fig. 1. Major components in a traditional camera image signal processing pipeline.

the network difficult to optimize, and lead to unsatisfactory learning performance.

In this paper, we propose a new framework for deep-learning-based ISP pipeline design, which includes a two-stage CNN and the associated training scheme. We firstly analyze the relationships of individual subtasks of an ISP and group them into two weakly correlated clusters, namely, the restoration group and the enhancement group. Then a CNN model called CameraNet is proposed with two subnetworks to address the two groups of subtasks, respectively. Accordingly, a restoration and an enhancement ground truths are specified and used to train CameraNet in a progressive manner. With this arrangement, the two-stage CameraNet allows collaborative processing of correlated ISP subtasks while avoiding mixed treatment of weakly correlated subtasks, leading to high quality sRGB image reconstruction in various imaging scenarios. In our experiments, CameraNet outperforms the state-of-the-art ISP learning methods and obtains consistently compelling results on three publically available benchmark datasets, including HDR+ [4], SID [20] and FiveK datasets [23].

The rest of the paper is organized as follows. Section II reviews the related work. Section III presents the framework of CameraNet, including the CNN architecture and training scheme. Section IV presents the experimental results. Section V concludes the paper.

II. Related Work

Our work is related to the research of camera ISP pipeline design as well as deep learning for low level vision, which are reviewed briefly as follows.

A. Image Signal Processing Pipeline

There are a number of image processing components in the ISP pipeline of a camera [24]–[29]. The major ones include demosaicking, noise reduction, white balancing, color space conversion, tone mapping and color enhancement, as shown in Fig. 1. The demosaicking operation interpolates the raw color filter array (CFA) image with repetitive mosaic pattern (e.g., Bayer pattern) into a full color image [26], followed by a denoising step to enhance the signal-to-noise ratio [27]. White balance corrects the color that is shifted by illumination according to human perception [28]. Color space conversion first transforms the image in camera color space to an intermediate color space (e.g., CIE XYZ) for processing, and then transforms the image to sRGB space for display. Tone mapping transforms the image in the high dynamic range irradiance space to a standard dynamic range image for display with image structures preserved. To achieve this goal, many tone mapping methods decompose the image into a base layer and a detail layer, and perform dynamic range reduction on base layer and detail enhancement on detail layer [30], [31]. Color enhancement operation manipulates the color style of an image [29]. One commonly used technique is the 3D lookup table search. A detailed survey of the ISP components can be found in [1], [2].

In traditional ISP design, usually an algorithm is developed for a specific ISP subtask. Such a divide-and-conquer strategy decomposes the complex ISP design problem into many simpler sub-problems, however, it may cause error accumulation along the algorithm flow in the pipeline [3]. In addition, the traditional ISP is difficult to produce high-quality images under challenging scenarios, such as low-light imaging.

Recently, a few learning-based approaches have been proposed for ISP pipeline design [18]–[20], [32]. One pioneering work is Jiang *et al.*'s local regression framework [32], where the raw image patches are clustered based on some simple features and then per-class filters are learned to transform the raw patches into the sRGB patches. This approach however has limited regression performance due to the use of simple parametric models. Motivated by the recent advances in deep learning [33], [34], Schwartz *et al.* proposed to learn a CNN model to replace the ISP pipeline in smartphone cameras [18]. Ratnasingam proposed a multiscale network featured with parallel connections for ISP learning [19]. In these deep-learning-based ISP methods, a CNN is trained for all the ISP components without considering much the relationships among them. This makes the network learning less effective in sRGB image reconstruction.

There exist a few datasets that can be used for ISP pipeline learning in different imaging scenarios [4], [20], [23]. These datasets contain raw images and the corresponding ground truth sRGB images that are manually processed and retouched in a controlled setting. Specifically, the FiveK dataset is featured with images retouched by five photographers to have different color styles [23], the HDR+ dataset is featured with burst denoising and sophisticated style retouching [4], and the SID dataset is featured with strong noise in nighttime imaging [20].

B. Deep Learning for Low-Level Vision

Inspired by the great success of deep learning in high-level vision problems [33], [34], some pioneer works have been developed for low-level vision applications. For example, Dong *et al.* proposed a three-layer CNN for image super-resolution [22], Zhang *et al.* proposed a deep CNN featured with batch normalization for image denoising [8], and Gharbi *et al.* addressed the photographic style enhancement

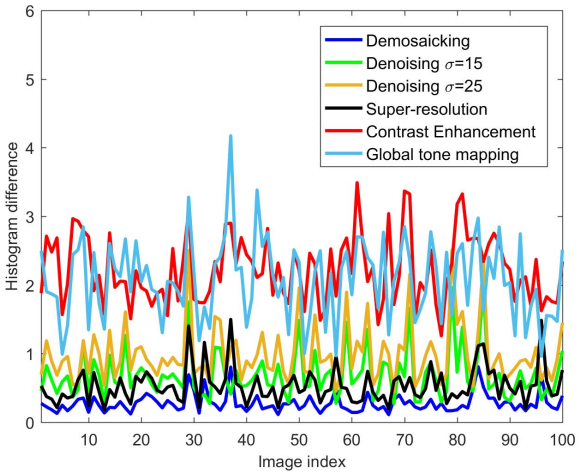


Fig. 2. The image histogram changes caused by different image processing operations, including demosaicking, denoising (with $\sigma = 15$ and 25), $4\times$ super-resolution, local contrast enhancement [38] and global tone mapping [39], on images in the BSD100 dataset [40]. The vertical axis denotes the ℓ_1 norm of histogram differences, while the horizontal axis denotes the image index in BSD100.

task by learning to estimate the per-pixel affine mapping in bilateral grid structure [14]. Many following works have been reported in the past several years [7], [9], [13], [15], [35], [36], and those deep-learning-based image restoration and enhancement methods have demonstrated significantly better performance than their traditional counterparts [26], [27], [29].

In addition to training a CNN to address a specific low-level vision task, one can also train a CNN to jointly address several related tasks [12], [17], [37]. Gharbi *et al*. trained a feedforward CNN for joint denoising and demosaicking [12]. Zhou *et al*. developed a residual network for joint demosaicking and super-resolution [37]. Recently, Qian *et al*. proposed a joint solution for denoising, super-resolution and demosaicking with raw images as input [17]. Despite the successes of CNN in joint restoration tasks, it needs more investigation on how to train a stable and effective CNN to address the complex mixture of image restoration and enhancement tasks, such as camera ISP learning. In this paper, we propose a two-stage CNN model to achieve this goal.

III. Framework

A. Problem Formulation

Suppose there are N essential subtasks in an ISP pipeline, including but not restricted to demosaicking, white balance, denoising, tone mapping and color enhancement. The traditional ISP pipeline employs N cascaded hand-crafted modules to address these subtasks. Let I_{cfa} be the raw CFA image and I_o be the output sRGB image. The traditional ISP can be represented as $I_o = f_N(f_{N-1}(\ldots(f_1(I_{cfa})\ldots))$, where $f_i, 1 \leq i \leq N$, denotes the ith algorithm component. The main drawback of such traditional ISP design is that each algorithm component is hand-crafted and it is difficult to optimize the pipeline as a whole, which limits the quality of output sRGB images.

In contrast to the traditional ISP design, we adopt the data-driven approach and model an ISP as a deep CNN system to address the N subtasks as a whole:

$$I_o = F_{isp}(I_{cfa}, \omega; \theta), \tag{1}$$

where $F_{isp}(.;\theta)$ refers to the CNN model with parameters θ to be optimized, and ω denotes the optional camera metadata (e.g., noise level, shutter speed) that can be used to help the network training and inference. We leverage a dataset S to train F_{isp} in a supervised manner. The dataset contains a set of input raw images I_{cfa}, and for each I_{cfa} there are K associated ground truth images $G_k, 1 \leq k \leq K$. In the case that $K = 1$, there is only one final ground truth output. In the case that $K > 1$, there are several intermediate ground truths G_k, $k < K$, leveraged to train the network, while G_K is the final ground truth for sRGB image reconstruction.

In the design of $F_{isp}(.;\theta)$, it is desirable that the CNN model can explicitly address the different ISP subtasks while keeping the network as compact and simple as possible. To this end, we propose an effective two-stage CNN architecture and the associated learning scheme, which are described in the following sections.

B. Two-Stage Grouping

As discussed in the previous subsection, F_{isp} is expected to address the ISP subtasks explicitly. One possible approach is to deploy a CNN subnetwork for each ISP subtask and chain them in sequence [41], [42]. As we discussed in the introduction section, however, such a divide-and-conquer strategy is cumbersome and ineffective, and it will make the whole network too heavy and complex. On the other hand, it has been demonstrated that some ISP subtasks, e.g., demosaicking and denoising, are correlated and they can be jointly addressed [12], [37]. Therefore, we propose to group the ISP subtasks into several weakly correlated clusters, while each cluster consists of several correlated subtasks. A CNN module is deployed for each cluster to allow joint learning of correlated subtasks, and then all the CNN modules are jointly fine-tuned to reduce the possible accumulated errors.

Based on the existing works in low-level vision, we group the ISP subtasks into two clusters: image restoration and enhancement. The goal of image restoration is to faithfully reconstruct the linear scene irradiance which contains genuine image structures and colors from raw image data. Typical restoration operations include color demosaicking, white balance, noise removal, deblurring, super-resolution, etc. They usually maintain the image distribution without largely changing the contrast and color style of an image. In contrast, the enhancement operations often nonlinearly change the image contrast and color distribution to make the image visually more appealing to human observers. Image enhancement operations are mainly located at the rear part of an ISP, such as tone mapping, color transform and contrast enhancement.

Let's perform a test to evaluate the influences of several typical restoration and enhancement operators on image

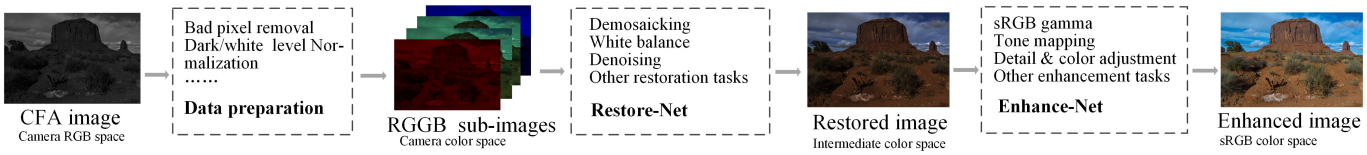


Fig. 3. The proposed CameraNet system for ISP learning.

distribution. The restoration operators, including demosaicking, denoising ($\sigma = 15, 25$) and super-resolution, and the enhancement operators, including local enhancement [38] and global tone mapping [39],[1] are employed in the test. White balance is excluded because it can be simply accounted for by per-channel global scaling. We denote the image before and after an operation $f(\cdot)$ as I and $f(I)$, respectively. Then, the ℓ_1 difference between the histogram vectors (with 256 bins) of I and $f(I)$ are computed to measure the amount of change on image intensity distribution. The BSD100 images are employed in the test [40]. For the restoration operations, we use the original images as $f(I)$ and degrade them to obtain I. Fig. 2 shows the ℓ_1 norms of histogram differences of the BSD100 images. We can see that the enhancement operators produce much higher changes on the image histogram than the restoration operators. This phenomenon validates that the enhancement and restoration operators have substantially different algorithm behaviors, which motivates us to employ a two-stage network design for ISP learning.

C. Two-Stage Network Design

According to the discussions in the above subsection, we categorize the ISP subtasks into two groups (restoration and enhancement), and propose a two-stage CNN system, namely CameraNet, which is illustrated in Fig. 3. It is composed of a data preparation module, a restoration module called Restore-Net, and an enhancement module called Enhance-Net.

The role of the data preparation module is to separate some simple operations from the training since they can be well performed beforehand. The pre-processing operations applied on the CFA image I_{cfa} include bad pixel repairing, dark and white level normalization and pixel rearrangement. Bad pixel repairing interpolates the pixels where there are no response due to manufacturing imperfection. We use the python package Rawpy for this operation, which replaces the bad pixels by their neighboring pixels. Dark and white level normalization normalizes the dynamic range to [0, 1]. Pixel rearrangement repacks the channel interlaced CFA image I_{cfa} to several single channel sub-images. Without loss of generality, we suppose that Bayer pattern is adopted in this paper. Then the CFA image I_{cfa} is rearranged as four sub-images (R, G, G, B) of the same size, and we denote by I_{rggb} the four sub-images for simplicity of expression.

[1]We firstly apply a reverse gamma conversion with parameter 2.4 to synthesize a linear raw image before applying the tone mapper.

Then Restore-Net, denoted by F_r, applies restoration-related operations, such as demosaicking, white balance and denoising, on the output of data preparation module, i.e., I_{rggb}. The output of Restore-Net is:

$$I_r = F_r(I_{rggb}, \omega_n; \theta_r) \tag{2}$$

where θ_r denotes the parameters of Restore-Net, and $\omega_n = \lambda_{shot} + \lambda_{read}^2$ is the input noise level to facilitate the denoising subtask. λ_{shot} and λ_{read} are the shot and readout noise parameters that can be obtained from the camera metadata. The restored image I_r is in an intermediate color space. The CIE XYZ space is considered here because it is designed to match human vision [43].

The Enhance-Net, denoted by F_e, takes the restored image I_r as input for processing. It first clips the intensity values below 0 and above 1, and applies an sRGB gamma function to the clipped image to account for the fixed nonlinear transformation from CIE XYZ space to sRGB space. Then the Enhance-Net learns to perform enhancement operations, such as tone mapping, detail enhancement and color manipulation, on I_r to produce the final output image I_o in sRGB color space:

$$I_o = F_e(I_r; \theta_e) \tag{3}$$

where θ_e denotes the parameters of Enhance-Net.

CNN architecture. There could be many possible designs for the Restore-Net and Enhance-Net modules. We consider a simple yet effective one, where two 5-level UNet like subnetworks are employed for Restore-Net and Enhance-Net, respectively. The architecture of the two subnetworks is shown in Fig. 4. A UNet has a contracting path to progressively reduce the resolution of feature maps, followed by an expanding path to progressively expand the resolution back [44]. Image structures are preserved by the skip connections from the contracting path to the expanding path at the same level. We adopt UNet for three reasons. First, the multi-scale processing nature of UNet can result in good image quality by learning adaptive operations for each scale. The finer scales focus on reproducing image local details and textures, while the coarser scales focus on enhancing image global colors and tones. Second, with UNet the main computations are deployed on the coarse image scales (lower resolution), resulting in relatively lower computational complexity. In addition, UNet can well solve multiple restoration subtasks by extracting common multiscale features to all subtasks and adopting a similar set of operations. Each subtask is flexibly accounted for in the network rather than rigidly treated.

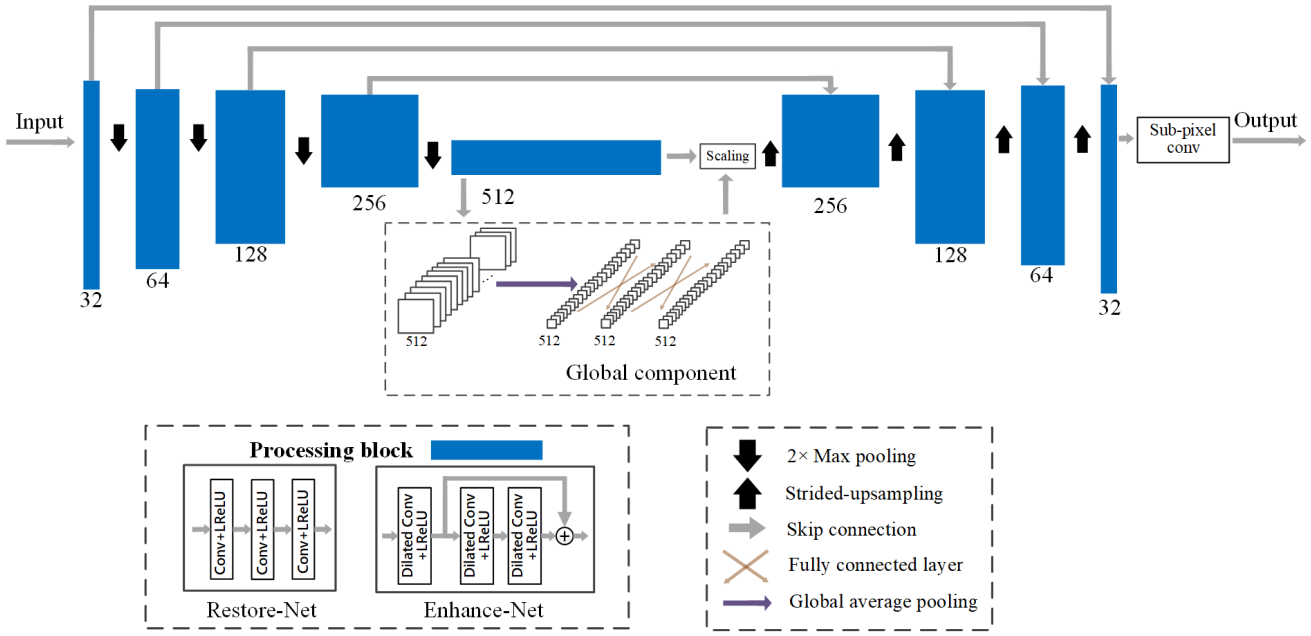


Fig. 4. The structure of UNet-like Restore-Net and Enhance-Net modules in the proposed CameraNet system.

Since the full color images I_r and I_o have twice the spatial resolution of the input sub-images I_{rggb} in each channel, a sub-pixel convolutional layer [45] is deployed at the end of Restore-Net and Enhance-Net to expand the resolution. In addition, to account for the global transformations in both modules (white balance in Restore-Net and global enhancement in Enhance-Net), we deploy an extra global transform block in the UNet modules, as shown in Fig. 4. This block first applies global averaging pooling to the input feature maps on the 5th level (lowest resolution), followed by 2 fully connected layers to obtain the globally scaled features as a 1D vector. Finally, the global features are multiplied to the output feature maps on the 5th level in a per-channel manner. This process can be described as:

$$H_{5,out} = U_5(H_{5,in}) \otimes L_{fc}(L_{fc}(L_p(H_{5,in}))), \tag{4}$$

where $H_{5,out}$ and $H_{5,in}$ denote the output and input feature maps on the 5th level, respectively. $U_5(.)$, $L_{fc}(.)$ and $L_p(.)$ denote the 5th level operation block of UNet, the fully connected layer and the global pooling layer, respectively. Symbol "$\otimes$" denotes per-channel multiplication.

To further promote the learning performance of Enhance-Net, we deploy two extra settings that are found helpful for enhancement tasks. First, the convolution dilation rates of Enhance-Net are set to 1,2,2,4,8 from the 1st level to the 5th level to enlarge the receptive field. By this setting, the network can refer to a larger context to enhance an image, which avoids halo artifacts around the edges. Second, Enhance-Net deploys a residual connection within a convolutional block, as shown in the specification of Fig. 4. The residual connection predicts and adds features upon the previous feature maps in the network, which is helpful for detail boosting.

D. Ground Truth Generation

Most existing datasets [4], [20], [23] contain only the final ground truth G_o of the network output. For example, the HDR+ [4] and FiveK [23] datasets provide the sRGB ground truths that are created by HDR+ algorithm and human retouching, respectively. However, for our proposed two-stage CameraNet system, it is expected that we could have a restoration ground truth G_r and an enhancement ground truth G_o, which are corresponding to the intermediately restored image I_r and the finally enhanced image I_o, for network training.

The ground truths G_r and G_o can be generated by using photo editing software, e.g., Adobe software. An example procedure is shown in Fig. 5. In the first step, the restoration ground truth G_r is created by performing restoration-related operations on the raw image I_{cfa}, including demosaicking, denoising and white balance. In our experiments on the FiveK dataset, the G_r is generated in this way. On some datasets (e.g., HDR+ dataset) where a raw image sequence is available for each scene, one can adopt additional operations to boost the quality of the restoration ground truth. For example, the sequence of raw images can be fused into one raw image to suppress noise, followed by other restoration operations. We use this method to generate the restoration ground truths on the HDR+ dataset. In the second step, the enhancement ground truth G_o is created by applying enhancement-related operations on the restoration ground truth G_r, including contrast adjustment, tone mapping, color manipulation and color conversion. This can be easily done by using photo retouching software, e.g., Adobe Lightroom.

Since the goal of restoration tasks is to objectively reconstruct genuine image structures and colors, the styles of the generated ground truths G_r from raw images are generally similar. In contrast, the enhancement tasks are subjective to human observers, which may result in various styles of enhancement ground truths G_o. Fig. 6 shows the image triplets from the HDR+ dataset and the FiveK dataset, including the raw image (demosaicked for better visualization), the restoration and the enhancement ground truths. We also show the reconstructed images in the two stages of our CameraNet for

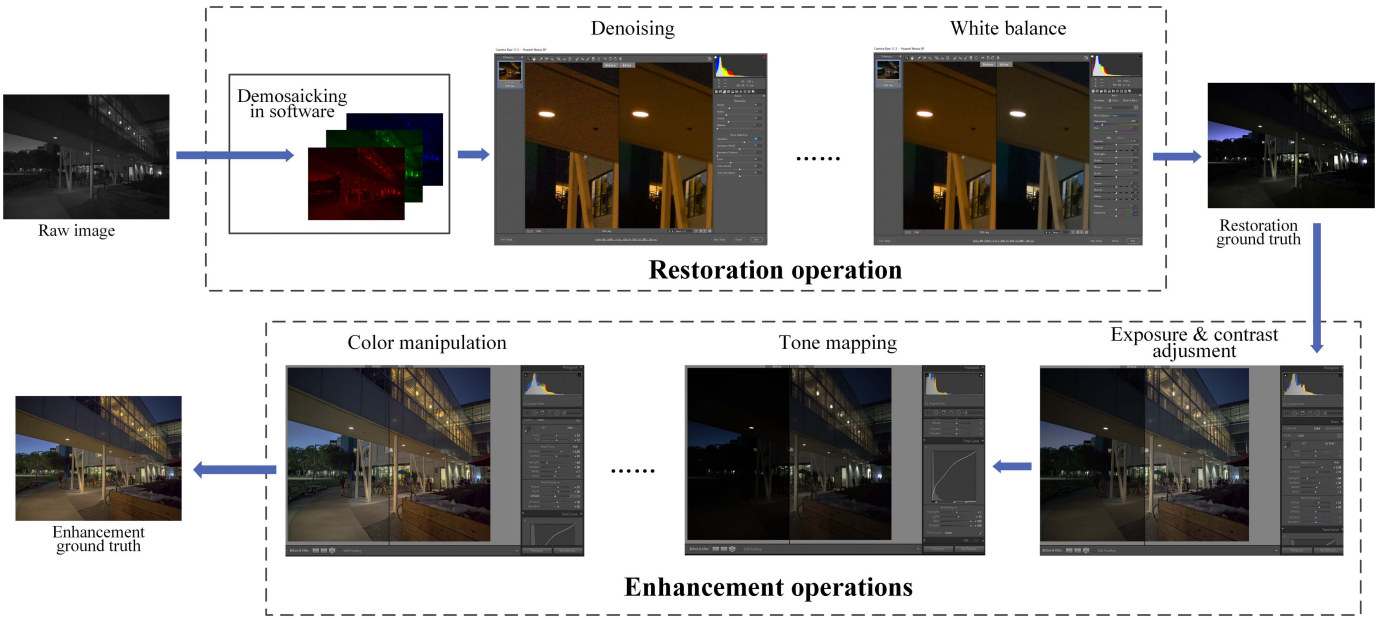


Fig. 5. The workflow of creating restoration and enhancement ground truths using Adobe software. The restoration ground truth is created in Adobe Camera Raw, while the enhancement ground truth is created in Lightroom. The dots in restoration operations refer to other possible restoration tasks such as aberration correction and deblurring, while the dots in enhancement operations refer to other possible adjustment of image features.

(a) Raw image (b) Restored by Restore-Net (c) Restoration ground truth (d) Enhanced by Enhance-Net (e) Enhancement ground truth

Fig. 6. Illustration of the two-stage network outputs and ground truths. The image in the first row is from the HDR+ dataset [4], while the image in the second row is from the FiveK dataset [23]. A gamma transform with parameter 2.2 is applied to the raw images and restoration ground truths for display.

reference. One can see that the two restoration ground truths exhibit similar visual attributes, whereas the two enhancement ground truths are of very different styles. The enhancement ground truth in HDR+ dataset emphasizes on detail enhancements while that in FiveK dataset focuses on color style manipulation.

E. Two-Step Training Scheme

Based on the two-stage structure of CameraNet, we propose a two-step training scheme of it. In the first step, the Restore-Net and Enhance-Net are independently trained in parallel, while in the second step, the two subnetworks are jointly fine-tuned. We adopt a set of ℓ_1 losses in the training of CameraNet because the ℓ_1 loss is simple to calculate and tends to converge to a visually good local minimum [46].

In the first step, the Restore-Net is trained with a restoration loss calculated between the restored image I_r and the ground truth G_r in linear and logarithmic space:

$$\mathcal{L}_r(I_r, G_r) = \|I_r - G_r\|_1 + \|log(max(I_r, \epsilon)) - log(max(G_r, \epsilon))\|_1, \quad (5)$$

where ϵ is a small value to avoid infinity. The use of the log sub-loss is based on the fact that the restored image I_r is in a linear space where the image intensity is proportional to scene radiance but not human visual response. Thus, to penalize the error in terms of human perception, we introduce this nonlinear term in the loss computation.

Meanwhile, the Enhance-Net is trained in parallel to Restore-Net. The restoration ground truth G_r is input to the Enhance-Net, and the output is denoted as $I_{o,r} = F_e(G_r; \theta_e)$. The enhancement loss is calculated as the ℓ_1 difference between $I_{o,r}$ and the ground truth G_o:

$$\mathcal{L}_o(I_{o,r}, G_o) = \|I_{o,r} - G_o\|_1, \quad (6)$$

It can be seen that the training of Enhance-Net does not rely on the output of Restore-Net. In addition, there is not a nonlinear term in the loss because the enhanced image is already in a nonlinear color space, i.e., sRGB space.

Once the parallel training of Restore-Net and Enhance-Net is finished in the first step, in the second step the two subnetworks are jointly fine-tuned with the following joint loss:

$$\mathcal{L}_{joint} = \lambda \cdot \mathcal{L}_r(I_r, G_r) + (1-\lambda) \cdot \mathcal{L}_o(I_o, G_o) \tag{7}$$

Note that in this step, the enhancement sub-loss takes I_o rather than $I_{o,r}$ in Eq. (6) as input for loss calculation. The joint fine-tuning has two roles. First, the Enhance-Net receives the gradients from the enhancement sub-loss, while the Restore-Net receives the gradients from both restoration and enhancement sub-losses, weighted by λ and $1-\lambda$, respectively. Thus, this step allows the Restore-Net to contribute to the final sRGB image reconstruction. Second, since the two subnetworks are trained independently in the first step, cumulative errors may occur due to the gap in the intermediate results. Joint fine-tuning can reduce such cumulative errors by facilitating the interaction between the two modules. The setting of parameter λ is scenario-specific. If the restoration subtasks dominate the ISP pipeline, e.g., in the low-light imaging scenario, λ should be set larger to emphasize the restoration functionality of Restore-Net, and vice versa.

While the adopted ℓ_1-based loss functions yield good results, our training scheme is open to other advanced loss design, e.g., adversarial loss [47] and perceptual loss [48]. Actually, we find that employing the perceptual loss in the fine-tuning step can slightly improve the visual appearance of the reconstructed images, which will be discussed in the experiment section.

IV. Experiments

In this section we perform extensive experiments to verify the learning capability and image reconstruction performance of our CameraNet system both quantitatively and qualitatively. Three objective indices, including PSNR, SSIM [49] and S-CIELAB [50], are employed in the quantitative evaluation. PSNR calculates the ratio of the peak signal power to the power of reconstruction errors, while SSIM measures the structural similarity between reconstructed and ground truth images. S-CIELAB measures the perceptual errors of two colors in the Lab space (the smaller the measure, the better the color fidelity). We use the code from [32] for calculating S-CIELAB. Without loss of generality, Bayer CFA pattern is used in all our experiments. However, it is not difficult to adapt CameraNet to a new CFA design. One can simply retrain the Restore-Net with the input data of the new CFA pattern, e.g., RGBW or RGBG.

A. Dataset Setting

Three publically available datasets that can be used for ISP learning are employed in our experiments, including the HDR+ dataset [4], the SID dataset [20] and the FiveK dataset [23]. These datasets have different features and they can be used to validate the performance of an ISP learning method from different aspects.

The HDR+ dataset [4] focuses on burst denoising and detail enhancement. For each scene, a burst of underexposed raw images are captured. Those images are firstly aligned and fused into one raw image to suppress noise, and then the HDR+ algorithm is applied to the fused raw image to produce the sRGB image. For each scene, the fused raw image and the corresponding sRGB image are provided in the dataset. We use DCraw to perform demosaicking, white balance and color conversion on the fused raw image to obtain the restoration ground truth, and treat the provided sRGB images as the enhancement ground truth. The Nexus 6P subset, which includes 665 scenes as training data and 240 scenes as testing data,[2] is used in the experiment. We take a single raw image (the reference frame in alignment) as the input of CameraNet. The data in the testing set are sampled according to the distribution of ISO values, which are rough indicators of noise level.

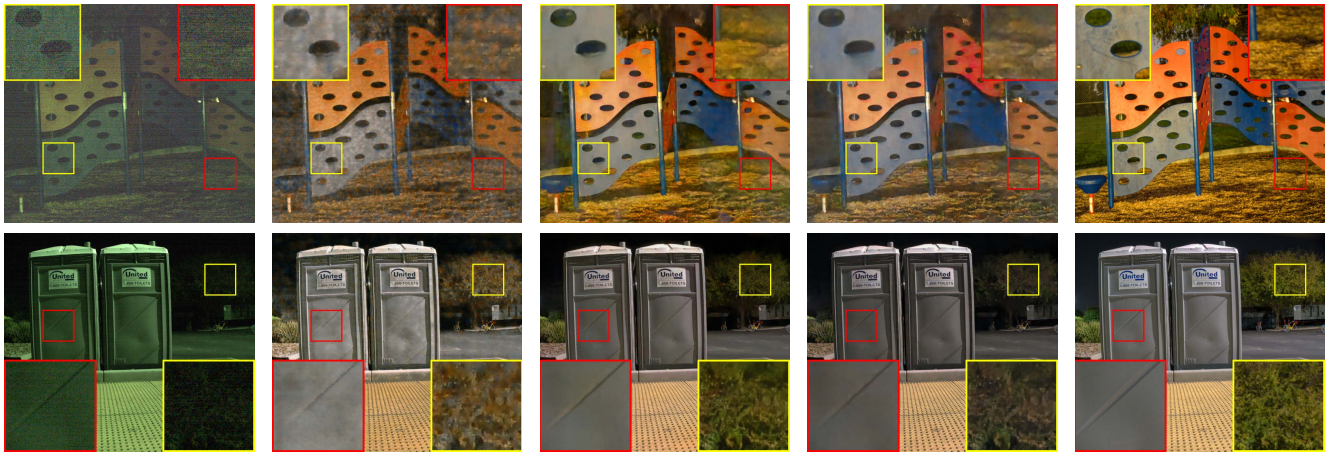
The SID dataset [20] focuses on denoising in low-light environment. For each scene, it provides a noisy raw image with short exposure and a relatively clean raw image with long exposure. To obtain the restoration ground truths, we use the DCRaw to perform restoration operations on the long-exposed raw images. Since the SID dataset does not involve any enhancement operation, we further process the restoration ground truth by the auto-enhancement tool in Photoshop to obtain the enhancement ground truth. We use the Sony A7S2 subset for experiments, which includes 181 and 50 scenes for training and testing, respectively. The data in the testing set are sampled according to the distribution of ISO values.

The FiveK dataset [23] is featured with strong manual retouching on image tone and color style. For each raw image of the 5,000 scenes, five photographers are employed to adjust various visual attributes of the image by using the Lightroom software and generate five images with different photographic styles. As in many previous works [14], [51], we take the set of images retouched by expert-C as the enhancement ground truths. Since the FiveK dataset does not contain restoration ground truth, we process the input raw image using DCRaw to obtain the restoration ground truth. The Nikon D700 subset with 500 training images and 150 uniformly sampled testing images is used in the experiments.

B. Experimental Setting

We use the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.99$) to train CameraNet and all the competing CNN models. In the first training step, Restore-Net is trained for 2000, 4000 and 1000 epochs on the FiveK, HDR+, and SID datasets, respectively, depending on the task complexity on the three datasets. The Enhance-Net is trained with 500 epochs on all the three datasets since the enhancement tasks on these datasets have comparable complexity. In the second fine-tuning step, 200 epochs are used for all the datasets.

[2]The other subsets are not used because there are some misalignments between the input images and the ground truths.



(a) Raw image (b) One-stage setting (c) Two-stage setting (d) Three-stage setting (e) Ground truth

Fig. 7. Results by one-stage, two-stage and three-stage CNN models. The two sets of images are from the SID dataset [20]. A gamma transform with parameter 2.2 is applied to the raw images and restoration ground truths for display.

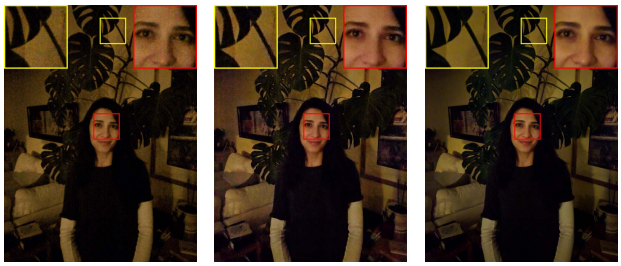
The initial learning rate for the first training step is set to 10^{-4}, and exponentially decays by 0.1 at 3/4 epochs. The learning rate for the fine-tuning step is fixed to 10^{-5}. Considering the importance of the restoration subtask on each dataset, the parameter λ in Eq. (7) is set to 0.1, 0.5 and 0.9 on the FiveK, HDR+, and SID datasets, respectively. In both training steps, the batch size is set to 1 and the patch size is set to 1024×1024. Random rotations, vertical and horizontal flippings are applied for data augmentation.

C. Ablation Study

We use the HDR+ and SID datasets for ablation study on the proposed two-stage network design, the training scheme, and the network architecture. All the evaluated models in this subsection are trained until convergence with the best testing performance.

The effectiveness of two-stage network design: To verify the effectiveness of the proposed two-stage design of CameraNet, we compare it with a one-stage and a three-stage counterparts. In the one-stage setting, a UNet with the same number of parameters as the two-stage CameraNet (i.e., double the number of processing blocks at each resolution level) is employed, and it is trained with the final enhancement ground truth. In the three-stage setting, three UNets are employed to progressively learn the ISP pipeline in three stages, i.e., demosaicking, denoising/white balance and enhancement. The number of parameters are maintained the same by reducing 1/3 the number of processing blocks at each resolution level.

The PSNR/SSIM/S-CIELAB results of the three competing networks on the HDR+ and SID datasets are shown in Table I. One can see that the default two-stage network works significantly better than the one-stage network, and much better than the three-stage network. Some visual comparison results are shown in Fig. 7. It can be seen that the one-stage network produces various visual artifacts, the three-stage network performs much better, while the two-stage network delivers the best visual quality. This experiment validates that it is difficult to use a single network to handle all ISP tasks together, while it is less effective to process correlated subtasks (e.g., demosaicking and denoising) using different networks. By grouping the ISP subtasks into two groups of correlated subtasks and deploying one network for each group, our two-stage CameraNet demonstrates highly effective ISP learning performance.



(a) Without step 1 (b) Defualt setting (c) Ground truth

Fig. 8. Comparison between the default training setting and the setting without step 1. The image is from the HDR+ dataset [4].

The two-step training scheme: We then evaluate the effectiveness of the proposed two-step training scheme. Firstly, we compare it with two variants. The first variant skips the first training step and directly goes to the second joint training step, i.e., we directly train the whole CameraNet with the loss in Eq. (7). The second variant keeps the first step but removes the second joint fine-tuning step. The results are shown in Table I. We can see that without the first step in training, the PSNR/SSIM/S-CIELAB scores become significantly worse. One visual example is presented in Fig. 8. We can see that some noises remain in the reconstructed image. This indicates the importance of progressive training of restoration and enhancement modules. On the other hand, from Table I we can see that without the joint fine-tuning step, the results are not that bad but still far behind our default two-step training scheme. One visual example is shown in Fig. 9. We can see that without the joint fine-tuning, the sky area has a sudden color change and has unnatural appearance.

Perceptual loss: The perceptual loss [48] has been widely used in many image restoration and enhancement networks to

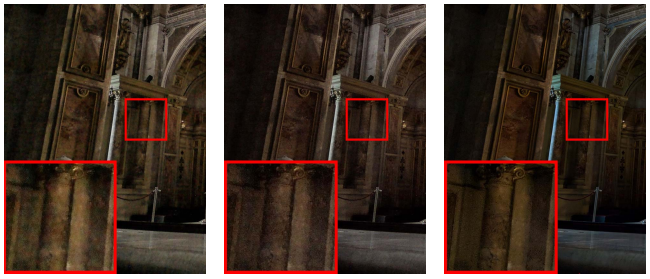
TABLE I

ABLATION STUDY ON THE HDR+ AND SID DATASETS. THE BEST AND SECOND BEST SCORES ARE HIGHLIGHTED IN RED AND BLUE FOR EACH COLUMN

	HDR+ dataset			SID dataset		
	PSNR	SSIM	S-CIELAB	PSNR	SSIM	S-CIELAB
Default two-stage setting	25.01	0.854	5.32	22.44	0.742	7.58
One-stage setting	21.53	0.816	6.48	19.04	0.692	9.48
Three-stage setting	23.57	0.834	5.99	21.56	0.735	8.40
Training without step 1	22.02	0.824	5.48	21.89	0.719	7.76
Training without step 2	23.98	0.843	5.34	22.21	0.738	7.63
Fine-tuning with perceptual loss	24.05	0.839	5.64	21.06	0.713	8.13
One-stage SRGAN+CAN24	21.72	0.801	7.18	19.85	0.682	9.50
Two-stage SRGAN+CAN24	22.31	0.815	6.93	20.96	0.714	8.85

(a) Without step 2 (b) Defualt setting (c) Ground truth

Fig. 9. Comparison between the default training setting and the setting without step 2. The image is from the HDR+ dataset [4].



(a) Without p.l. (b) With p.l. (c) ground truth

Fig. 10. Comparison between the results with and without perceptual loss (p.l.). We add the p.l. on the enhanced image in the fine-tuning step with weight 0.01. The image is from the HDR+ dataset [4].

improve the image visual quality. It is interesting to evaluate whether the perceptual loss can bring additional benefit to our CameraNet. We apply the perceptual loss (weighted by 0.01) on the enhanced images in the fine-tuning step.[3] The quantitative results are presented in Table I, and one visual example is shown in Fig. 10. We can see that the perceptual loss slightly improves the visual quality by reducing some subtle artifacts, while it leads to a moderate drop in the quantitative metrics since it penalizes the error in feature domain rather than the image domain.

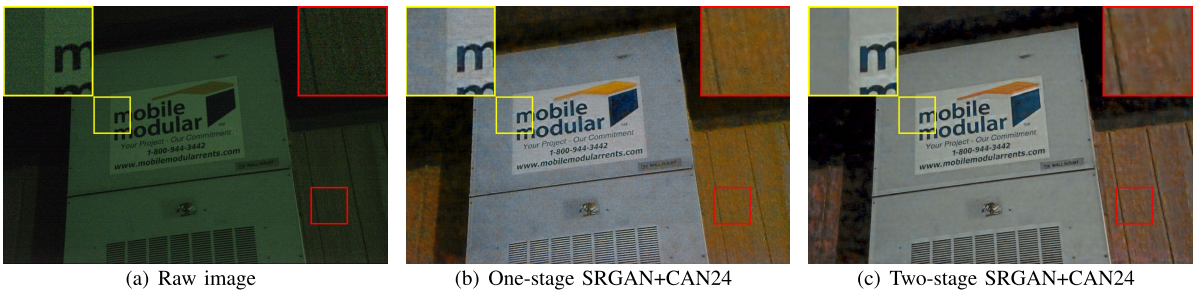
Other CNN architectures: To verify whether the proposed two-stage framework can be generalized to other CNN architectures, we further compare the one-stage and two-stage settings by using a different CNN architecture. We use SRGAN [47] with 10 layers as the restoration subnetwork and CAN24 [36] as the enhancement subnetwork. The PSNR/SSIM/S-CIELAB scores are shown in Table I, and one visual example is shown in Fig. 11. We can see that the two-stage setting of SRGAN+CAN24 outperforms its one-stage counterpart. Meanwhile, the two-stage SRGAN+CAN24 is not as effective as our CameraNet in noise removal. We think this is mainly because SRGAN+CAN24 lacks multiscale processing that facilitates the denoising task.

[3]We use the "relu2_2" and "relu5_4" layers in the VGG-19 network to calculate the loss.

D. Comparison With Recent Learning-Based ISP

In this section, we compare our CameraNet with those recently developed learning-based ISP methods, including L3 algorithm [32], DeepISP-Net [18] and DeepCamera [19]. The L3 algorithm firstly groups the patches of the input raw images according to the intensity level and then learns a per-class filter to obtain the sRGB image. DeepISP-Net and DeepCamera are single-stage CNN models trained in an end-to-end manner. In particular, DeepISP-Net takes a pre-demosaicked image as input and process the image with a single scale. DeepCamera takes the mosaic CFA image as input and adopts a multi-scale architecture. We train all the compared methods on the HDR+, FiveK and SID datasets until convergence with their best testing results. The source codes of L3 is provided by the authors. Because the source codes of DeepISP-Net and DeepCamera are unavailable, we implement them based on the settings described in the original papers and train them using the original loss functions. The PSNR/SSIM/S-CIELAB results of the compared methods are shown in Table II, while Figs. 12–14 present the visual results. More visual comparisons can be found in the supplementary file.

Results on the HDR+ dataset: The HDR+ dataset is featured with moderate denoising and strong detail enhancement. As can be seen from Table II, the proposed CameraNet achieves significantly better objective scores than the other methods. This is because the two-stage nature of CameraNet can effectively account for the restoration and enhancement tasks involved in the HDR+ dataset. In contrast, the L3 method, DeepISP-Net and DeepCamera mix the restoration and enhancement tasks to train the filters or networks, making the learning process more difficult. Fig. 12 shows a visual example for comparison. The proposed CameraNet obtains visually pleasing results, while the L3 method, DeepISP-Net and DeepCamera produce visual artifacts.



(a) Raw image (b) One-stage SRGAN+CAN24 (c) Two-stage SRGAN+CAN24

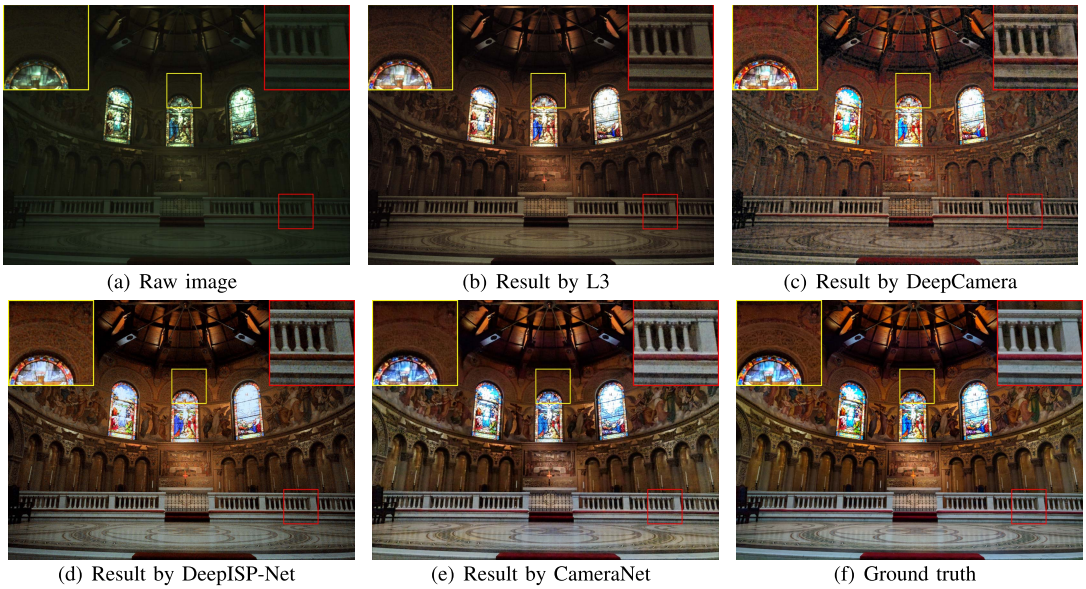
(d) CameraNet (e) Ground truth

Fig. 11. Comparison between SRGAN+CAN24 and CameraNet. A gamma transform with parameter 2.2 is applied to the raw images and restoration ground truths for display.

TABLE II

OBJECTIVE COMPARISON DIFFERENT LEARNING-BASED ISP METHODS

	HDR+ dataset			SID dataset			FiveK dataset		
	PSNR	SSIM	S-CIELAB	PSNR	SSIM	S-CIELAB	PSNR	SSIM	S-CIELAB
CameraNet	**25.01**	**0.854**	**5.32**	**22.44**	**0.742**	**7.58**	**23.57**	**0.849**	**6.74**
L3 algorithm [32]	19.23	0.682	9.84	16.47	0.462	12.64	20.00	0.797	10.70
DeepISP-Net [18]	22.88	0.818	6.78	18.26	0.649	10.18	22.59	0.845	7.38
DeepCamera [19]	20.65	0.738	8.81	18.19	0.587	10.97	20.67	0.776	8.86

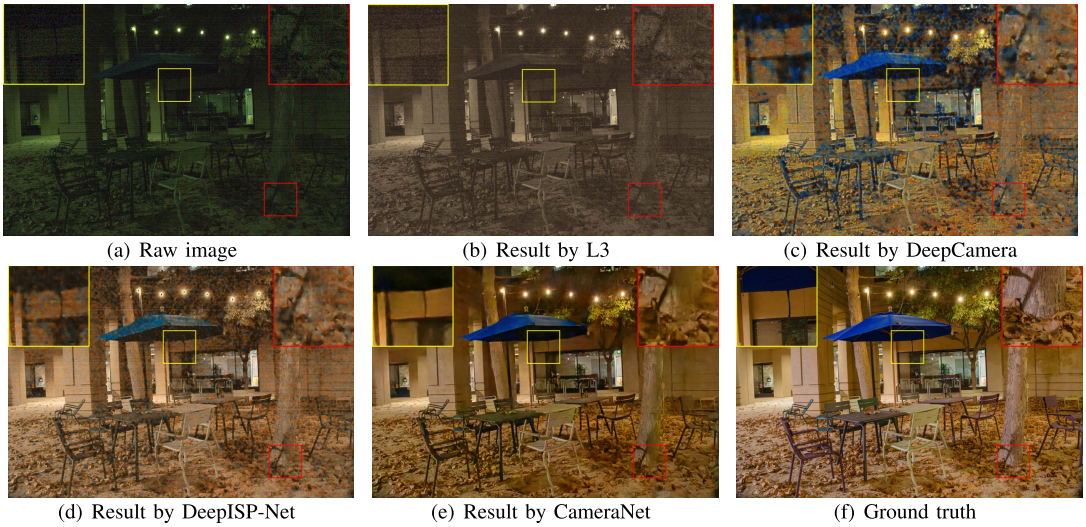


(a) Raw image (b) Result by L3 (c) Result by DeepCamera

(d) Result by DeepISP-Net (e) Result by CameraNet (f) Ground truth

Fig. 12. Results on a church image from the HDR+ dataset [4] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization.

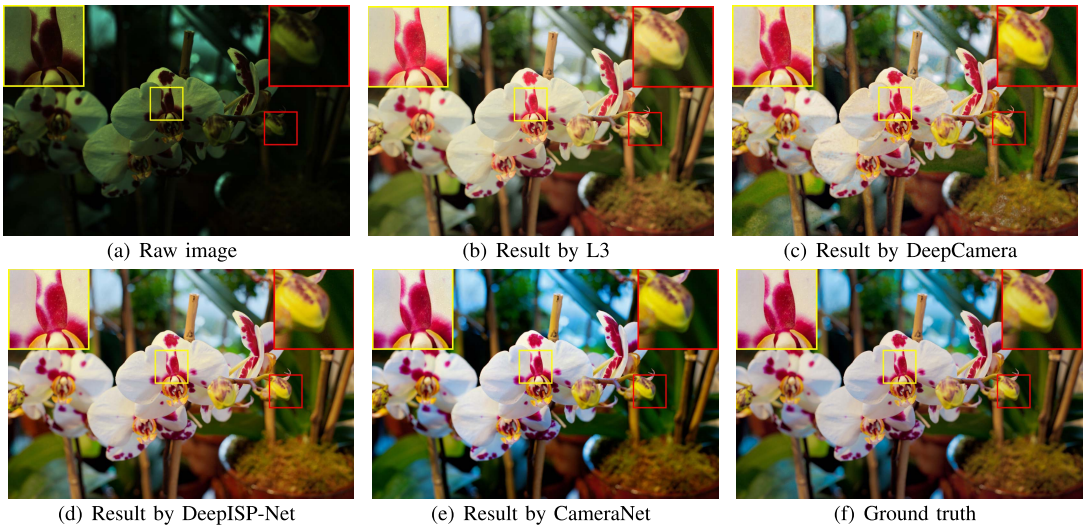
In particular, the L3 method barely performs denoising and produces false colors because the filter learning approach is too simple for the complex ISP tasks. DeepISP-Net and DeepCamera show better results, but they retain some noise-like artifacts. We suspect this is because DeepISP-Net and DeepCamera mix the denoising and color manipulation subtasks. As a result, the noise in the raw image is not effectively removed but amplified. In contrast, the proposed CameraNet



(a) Raw image (b) Result by L3 (c) Result by DeepCamera

(d) Result by DeepISP-Net (e) Result by CameraNet (f) Ground truth

Fig. 13. Results on a pavilion image from the SID dataset [20] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization.



(a) Raw image (b) Result by L3 (c) Result by DeepCamera

(d) Result by DeepISP-Net (e) Result by CameraNet (f) Ground truth

Fig. 14. Results on a flower image from the FiveK dataset [23] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization.

produces visually appealing results with much less artifacts, which can be attributed to the two-stage treatment of different ISP subtasks.

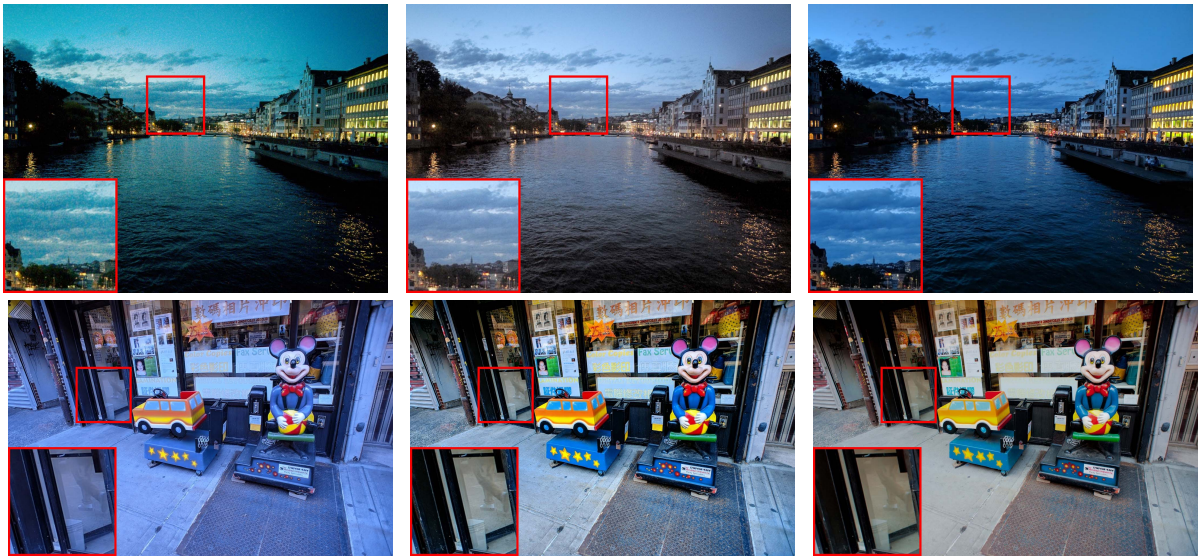
Results on the SID dataset: On the SID dataset, from Table II we can see that CameraNet outperforms the other methods by a large margin. This is because the noise level in the SID dataset is much higher than the HDR+ dataset, which requires the CNN model to have strong denoising capability. Our CameraNet meets this requirement by explicitly considering the denoising subtask in the restoration stage, whereas DeepISP-Net and DeepCamera mix all the ISP subtasks together in learning, leading to inferior performance. Fig. 13 shows an example of visual comparison. We can see that the visual quality of the proposed CameraNet is significantly higher than DeepISP-Net and DeepCamera. Specifically, DeepISP-Net produces inaccurate colors, while DeepCamera remains serious noise in the reconstructed images. In comparison, CameraNet effectively reduces the noise and enhances the image structures. Moreover, the results by the L3 method largely deviate from the ground truth. This is because the filter-learning-based L3 model is not expressive enough to perform the ISP tasks in challenging conditions such as low-light imaging.

Results on the FiveK dataset: Compared with the HDR+ and SID datasets, the FiveK dataset is less challenging because it does not involve the denoising subtask. In fact, the major task on the FiveK dataset is the enhancement of colors and tones on images captured by high-end cameras with little noise. From Table II, we can see that the advantage of CameraNet over DeepISP-Net is not as significant as that on the HDR+ and SID datasets because the dominant enhancement tasks can be well learned by DeepISP-Net. The results by DeepCamera and L3 model are much worse than CameraNet and DeepISP-Net. The inferior performance of DeepCamera

TABLE III

OVER-FITTING EVALUATION. THIS TABLE COMPARES THE TRAINING AND TESTING LOSSES OF THE LAST EPOCH FOR EACH TRAINING STEP ON THE THREE DATASETS. THE "GAP" MEANS THE DIFFERENCE BETWEEN THE TESTING LOSS AND THE TRAINING LOSS

	First step (Restore-Net)			First step (Enhance-Net)			Second step		
	Training	Testing	Gap	Training	Testing	Gap	Training	Testing	Gap
HDR+ dataset	0.0043	0.0058	+0.0015	0.0348	0.0421	+0.0073	0.0370	0.0482	+0.0112
SID dataset	0.0078	0.0117	+0.0039	0.0387	0.0676	+0.0289	0.0400	0.0769	+0.0369
FiveK dataset	0.0034	0.0067	+0.0033	0.0326	0.0659	+0.0333	0.0345	0.0670	+0.0325



(a) Full transfer (b) Enhance-Net transfer (c) Ground truth

Fig. 15. Cross-dataset testing. First row: the results of transferring the CameraNet trained on FiveK dataset to the testing image from HDR+ dataset. Second row: the results of transferring the CameraNet trained on HDR+ dataset to the testing image from FiveK dataset. "Full transfer" means transferring the whole CameraNet, while "Enhance-Net transfer" means transferring only the Enhance-Net.

may be caused by its use of mosaic CFA image as input to the network. In such case, the convolutional kernels at the early layers have extra burden to separate the color channels of CFA image, leading to less accurate results. Fig. 14 compares the results of different methods on a flower image. We can see that CameraNet and DeepISP-Net achieve satisfactory results, whereas the L3 method and DeepCamera generate some artifacts.

Over-fitting evaluation: Table III compares the training and testing losses of the last epoch of each training step on the three datasets. One can see that there is over-fitting on all datasets, especially on the SID and FiveK datasets since they have fewer training data than the HDR+ dataset. The over-fitting problem is mostly caused by the lack of training data on the three datasets. We believe it can be diluted if more data can be collected for training.

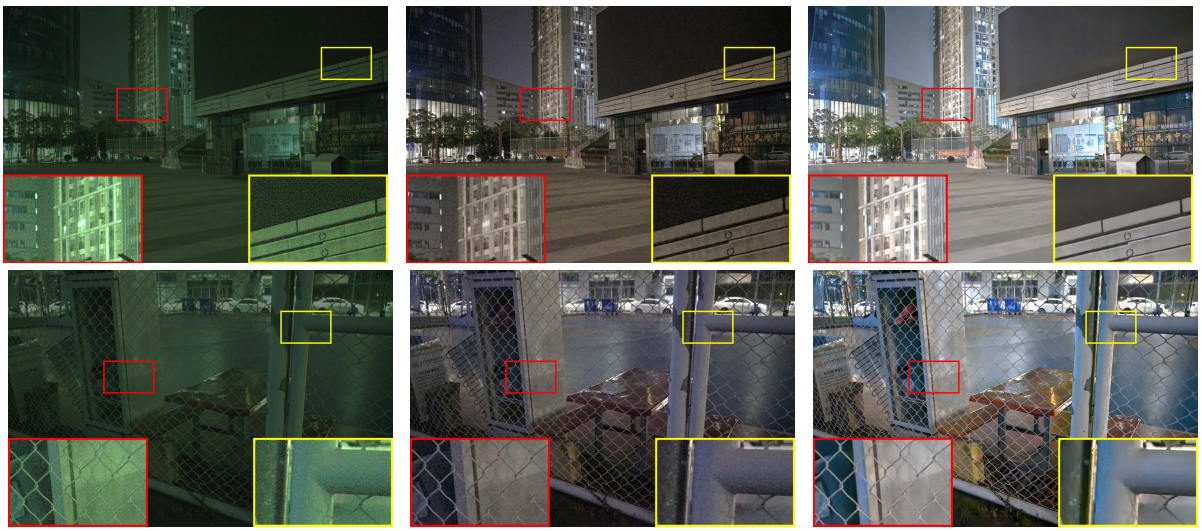
E. Cross-Dataset Testing

We use the HDR+ and FiveK datasets to test the cross-dataset performance of CameraNet. Specifically, we apply the network trained on one dataset to the testing set of another dataset. We only perform subjective evaluation because the two datasets have different types of ground truths, which makes the objective comparison less meaningful. Fig. 15 presents two cross-dataset testing examples, from which we can have two observations. On one hand, if we transfer the whole CameraNet trained on one dataset to the raw images of another dataset with a different sensor, the results have erroneous colors and details (see the left column of Fig. 15). For example, the result of "FiveK to HDR+" (top left image in Fig. 15) exhibits greenish color and noisy details. This is because the Restore-Net depends heavily on the camera sensor, and the mismatched sensor statistics will cause the inaccurate reconstruction of the sRGB image. On the other hand, if we only apply the Enhance-Net to the restored images by Restore-Net in another dataset, the results are perceptually acceptable but with a different image style (see the middle column of Fig. 15). This is because the restored images are in the similar color space so that the Enhance-Net depend less on the camera sensor.

The above observations imply that when we develop an ISP for a new sensor (possibly with a new CFA pattern), we may not need to completely retrain the CameraNet. We could only retrain the Restore-Net and then refine the Enhance-Net a little. In addition, different Enhance-Nets can be trained for a sensor to obtain different enhancement styles, such as nighttime, portrait, landscape, objects, etc.

F. Comparison With Traditional ISP

Since there is not a traditional ISP publically available to use, we compare CameraNet with the ISP onboard a Sony A7S2 camera (the same model as the one used in



(a) Raw image (b) Result by Sony ISP (c) Result by CameraNet

Fig. 16. Comparison with Sony A7S2 ISP in low-light scenarios. Both of the raw images are captured with aperture f3.5, exposure time 1/100s and ISO 12800. A gamma transform with parameter 2.2 is applied to the raw image for better visualization.

TABLE IV

COMPUTATIONAL COMPLEXITY OF THE COMPARED CNN MODELS. THE GFLOPS AND RUNNING TIME ARE EVALUATED ON AN IMAGE OF RESOLUTION 4032 × 3024

	GFLOPS	Running time (sec.)	Number of parameters (mill.)
CameraNet	**3306.69**	**0.892**	26.53
DeepISP-Net [18]	12869.79	2.12	0.629
DeepCamera [19]	4460.35	1.62	**0.467**

the SID dataset [20]) to demonstrate the advantage of our method over traditional ISP pipeline. Specifically, we use the Sony A7S2 camera to collect several noisy raw images in low-light environment with similar settings to those used in the construction of the SID dataset. The JPEG images output by the camera are collected as the results by Sony A7S2 ISP. The results by our approach are obtained by first applying CameraNet trained on the SID dataset to the collected noisy raw images, and then compressing the output sRGB images by JPEG. Fig. 16 shows the visual comparison between CameraNet and Sony A7S2 ISP on two raw images. One can see that in such low-light imaging scenario, the Sony A7S2 ISP produces results with residual noise and faded color, while the results by CameraNet exhibit clean structure, high local contrast and vivid color. This demonstrates the powerful image reconstruction capability of learning-based ISP methods in challenging scenarios. More comparison results can be found in the supplementary file.

G. Computational Complexity

In Table IV, we compare the computational complexity, running time and number of parameters of the competing CNN-based methods on Nvidia Quadro GV100. We can see that CameraNet has the lowest complexity and fastest speed. To produce an sRGB image of size 4032 × 3024, it consumes 3306.69 GFLOPS in 0.892s. DeepISP-Net consumes much more GFLOPS than CameraNet and DeepCamera and it runs the slowest. The lower computational complexity of CameraNet is mainly attributed to its multi-scale operations. However, CameraNet has 26.53 million parameters, which consumes much more memory than the other two CNN models. This is because the number of convolution channels grows exponentially in the contracting path of a UNet module, yielding roughly 33% parameters at the lowest resolution level. Since UNet deploys most of the computations on the lowest resolution level, the proposed CameraNet still has a low GFLOPS consumption.

H. Limitations

The proposed CameraNet has two main limitations. First, the number of parameters (26.53M) and computational cost (3306.69 GLOPS) are relatively high for application to mobile devices. It is expected that the network can be trimmed and compressed to attain better compactness and efficiency. Second, our CameraNet is designed for single-frame photography. In recent years, burst imaging is becoming more and more popular in mobile cameras, where multiple raw images are captured and fused into one sRGB image. For burst imaging, some additional components should be added to our current CNN architecture, such as frame alignment and fusion. How to compress our network for mobile devices and how to extend it to burst photography will be our future work.

V. CONCLUSION

We proposed an effective two-stage CNN system, namely CameraNet, for data-driven ISP pipeline learning. We exploited the intrinsic correlations among the ISP subtasks and categorized them into two sets of weakly correlated operations, i.e., restoration and enhancement. Accordingly, a two-stage architecture was adopted in the proposed CameraNet to account for the two sets of operations, facilitating the

learning capability while maintaining the model compactness. Two ground truths were specified to train the two-stage model, and a two-step training scheme was employed to train the whole model. Experiments showed that the proposed two-stage CNN framework significantly outperforms the commonly used one-stage framework in deep ISP learning. The proposed CameraNet outperforms state-of-the-art learning-based ISP models on three benchmark ISP datasets in terms of both quantitative measures and visual perception quality.

REFERENCES

[1] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 34–43, Jan. 2005.

[2] H. C. Karaimer and M. S. Brown, "A software platform for manipulating the camera imaging pipeline," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 429–444.

[3] F. Heide *et al.*, "FlexISP: A flexible camera image processing framework," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 231:1–231:13, 2014.

[4] S. W. Hasinoff *et al.*, "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Nov. 2016.

[5] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun, "Fast burst images denoising," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 1–9, Nov. 2014.

[6] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, pp. 1090–1097.

[7] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[8] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[9] Y. Hu, B. Wang, and S. Lin, "FC4: Fully convolutional color constancy with confidence-weighted pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4085–4094.

[10] S. Bianco, C. Cusano, and R. Schettini, "Color constancy using CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 81–89.

[11] R. Tan, K. Zhang, W. Zuo, and L. Zhang, "Color image demosaicking via deep residual learning," in *IEEE Int. Conf. Multimedia Expo (ICME)*, 2017, pp. 793–798.

[12] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 191:1–191:12, 2016.

[13] Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, "Deep photo enhancer: Unpaired learning for image enhancement from photographs with GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6306–6314.

[14] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Trans. Graph.*, vol. 36, no. 4, p. 118, 2017.

[15] J. Cai, S. Gu, and L. Zhang, "Learning a deep single image contrast enhancer from multi-exposure images," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 2049–2062, Apr. 2018.

[16] P. Vandewalle, K. Krichane, D. Alleysson, and S. Süsstrunk, "Joint demosaicing and super-resolution imaging from a set of unregistered aliased images," in *Proc. Digit. Photography III*, San Jose, CA, USA, Jan. 2007, Art. no. 65020A.

[17] G. Qian, J. Gu, J. S. Ren, C. Dong, F. Zhao, and J. Lin, "Trinity of pixel enhancement: A joint solution for demosaicking, denoising and super-resolution," *CoRR*, vol. abs/1905.02538, pp. 1–10, May 2019.

[18] E. Schwartz, R. Giryes, and A. M. Bronstein, "DeepISP: Toward learning an End-to-End image processing pipeline," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 912–923, Feb. 2019.

[19] S. Ratnasingam, "Deep camera: A fully convolutional neural network for image signal processing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3868–3878.

[20] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3291–3300.

[21] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.

[22] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 8692, 2014, pp. 184–199.

[23] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," in *Proc. CVPR*, Jun. 2011, pp. 97–104.

[24] K. Egiazarian, A. Foi, and V. Katkovnik, "Compressed sensing image reconstruction via recursive spatially adaptive filtering," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2007, pp. 549–552.

[25] Y. Kim, M. S. Nadar, and A. Bilgin, "Wavelet-based compressed sensing using a Gaussian scale mixture model," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3102–3108, Jun. 2012.

[26] X. Wu, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *J. Electron. Imag.*, vol. 20, no. 2, Apr. 2011, Art. no. 023016.

[27] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[28] D. Cheng, B. Price, S. Cohen, and M. S. Brown, "Beyond white: Ground truth colors for color constancy correction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 298–306.

[29] L. Yuan and J. Sun, "Automatic exposure correction of consumer photographs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 771–785.

[30] Z. Liang, J. Xu, D. Zhang, Z. Cao, and L. Zhang, "A hybrid ℓ_1-ℓ_0 layer decomposition model for tone mapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4758–4766.

[31] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257–266, Jul. 2002.

[32] H. Jiang, Q. Tian, J. Farrell, and B. A. Wandell, "Learning the image processing pipeline," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 5032–5042, Oct. 2017.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[35] Y. Romano, J. Isidoro, and P. Milanfar, "RAISR: Rapid and accurate image super resolution," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 110–125, Mar. 2017.

[36] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2497–2506.

[37] R. Zhou, R. Achanta, and S. Süsstrunk, "Deep residual network for joint demosaicing and super-resolution," *CoRR*, vol. abs/1802.06573, pp. 1–9, Feb. 2018.

[38] S. Wang, J. Zheng, H.-M. Hu, and B. Li, "Naturalness preserved enhancement algorithm for non-uniform illumination images," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3538–3548, Sep. 2013.

[39] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 267–276, Jul. 2002.

[40] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2001, pp. 416–425.

[41] H. Lin, S. Joo Kim, S. Susstrunk, and M. S. Brown, "Revisiting radiometric calibration for color computer vision," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 129–136.

[42] A. Chakrabarti, D. Scharstein, and T. Zickler, "An empirical camera model for Internet color vision," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, no. 2, 2009, p. 4.

[43] R. M. H. Nguyen and M. S. Brown, "Why you should forget luminance conversion and do something better," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5920–5928.

[44] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.

[45] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.

[46] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.
[47] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.
[48] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 9906, 2016, pp. 694–711.
[49] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[50] X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color-image reproduction," *J. Soc. Inf. Display*, vol. 5, no. 1, pp. 61–63, 1997.
[51] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu, "Automatic photo adjustment using deep neural networks," *ACM Trans. Graph.*, vol. 35, no. 2, pp. 11:1–11:15, 2016.

Zhetong Liang received the B.Sc. degree from the Guangdong University of Technology in 2013, and the M.Sc. degree from the South China University of Technology in 2016. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University. His research interests include HDR imaging, image enhancement, image denoising, and image processing pipeline.

Jianrui Cai received the B.Sc. and M.Sc. degrees from the College of Computer Science and Electronic Engineering, Hunan University, and the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University. His research interests include image processing, computational photography, and computer vision.

Zisheng Cao received the B.S. and M.S. degrees from Tsinghua University in 2005 and 2007, respectively, and the Ph.D. degree from The University of Hong Kong, in 2014. He is currently with the Imaging Group, DJI. Before joining DJI, he was a Research Scientist at Philips. His research interests include image signal processing and machine learning.

Lei Zhang (Fellow, IEEE) received the B.Sc. degree from the Shenyang Institute of Aeronautical Engineering, Shenyang, China, in 1995, and the M.Sc. and Ph.D. degrees in control theory and engineering from Northwestern Polytechnical University, Xi'an, China, in 1998 and 2001, respectively. From 2001 to 2002, he was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University. From January 2003 to January 2006, he worked as a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, McMaster University, Canada. In 2006, he joined the Department of Computing, The Hong Kong Polytechnic University, as an Assistant Professor, where he has been a Chair Professor since July 2017. His research interests include computer vision, image and video analysis, pattern recognition, and biometrics. He has published more than 200 articles in those areas. As of 2020, his publications have been cited more than 57 000 times in literature. He is a Senior Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, and is/was an Associate Editor of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *SIAM Journal on Imaging Sciences*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and *Image and Vision Computing*. He is a "Clarivate Analytics Highly Cited Researcher" from 2015 to 2020.