

ParamISP: Learned Forward and Inverse ISPs using Camera Parameters

Woohyeok Kim^{1*} Geonu Kim^{1*} Junyong Lee^{2†}
 Seungyong Lee¹ Seung-Hwan Baek¹ Sunghyun Cho¹
¹POSTECH ²Samsung AI Center Toronto

Abstract

RAW images are rarely shared mainly due to its excessive data size compared to their sRGB counterparts obtained by camera ISPs. Learning the forward and inverse processes of camera ISPs has been recently demonstrated, enabling physically-meaningful RAW-level image processing on input sRGB images. However, existing learning-based ISP methods fail to handle the large variations in the ISP processes with respect to camera parameters such as ISO and exposure time, and have limitations when used for various applications. In this paper, we propose ParamISP, a learning-based method for forward and inverse conversion between sRGB and RAW images, that adopts a novel neural-network module to utilize camera parameters, which is dubbed as ParamNet. Given the camera parameters provided in the EXIF data, ParamNet converts them into a feature vector to control the ISP networks. Extensive experiments demonstrate that ParamISP achieve superior RAW and sRGB reconstruction results compared to previous methods and it can be effectively used for a variety of applications such as deblurring dataset synthesis, raw deblurring, HDR reconstruction, and camera-to-camera transfer.

1. Introduction

A camera ISP converts a RAW image into a visually pleasing sRGB image, which is typically saved as a JPEG image. A camera ISP performs a series of operations including defective pixel correction, denoising, lens shading correction, white balance, color filter array interpolation, color space conversion, tone reproduction, and non-linear contrast enhancement. Detailed operations of camera ISPs are typically sealed to the public and vary from camera to camera.

Unlike sRGB images, RAW images provide physically-meaningful and interpretable information such as noise distributions as they have the linear relationship between image intensity and radiant energy incident on a camera

sensor. Such properties of RAW images have been exploited for denoising [1, 17, 23], HDR [5, 14], and super-resolution [24, 32], leading to superior quality than using only sRGB images. However, a RAW image demands large memory to store due to the use of high-precision bits without any compression. Therefore, only sRGB images are often shared without their RAW counterparts, making it difficult to utilize the useful properties of RAW images.

Recently, several approaches have been proposed to reconstruct RAW images from sRGB images and vice versa by modeling forward and inverse ISPs [3, 4, 7, 29, 30]. However, despite notable improvements, they still suffer from limitations. First, real-world ISPs adjust their operations based on the camera parameters, e.g., exposure time and sensor sensitivity, as shown in Fig. 1. However, previous methods overlook this adaptive nature of real-world ISPs, and learn average ISP operations, which leads to low reconstruction performance. Second, previous methods adopt rather simple network architectures disregarding the complexity of the ISP operations, which leads to low RAW reconstruction quality. They construct a single network by stacking invertible or residual blocks [29, 30], or organize modules by simply arranging convolution layers [3].

In this paper, we present a novel forward and inverse ISP framework, *ParamISP*. ParamISP is designed to faithfully reflect the real-world ISP operations that change based on camera parameters. To this end, ParamISP leverages camera parameters provided in the EXIF metadata of a JPEG image. To effectively incorporate the camera parameters, ParamISP consists of a pair of forward and inverse ISP networks that include a novel neural network module *ParamNet*. ParamNet extracts a feature vector from the camera parameters including exposure time, sensitivity, aperture size, and focal length, and feeds it to the forward and inverse ISP subnetworks to control their behaviors.

To learn ISP operations for varying camera parameters, we need to address the following problems. First, the camera parameters have significantly different scales, such as an exposure time of 0.01 sec. and a sensitivity of 800. Moreover, some parameters such as exposure time and sensor sensitivity have non-linearly increasing parameter val-

* Equal contribution.

† Work done prior to joining Samsung.

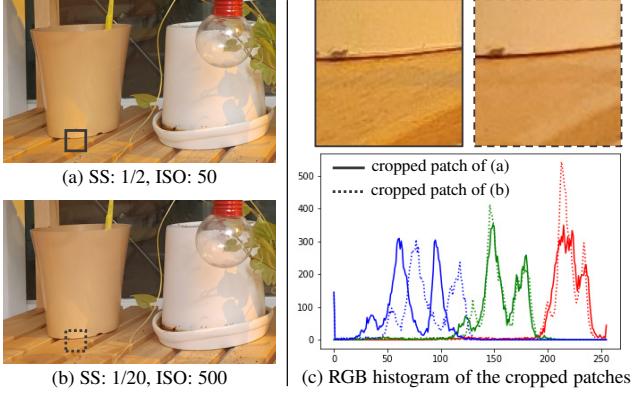


Figure 1. Impact of camera parameters on a camera ISP. Images (a) and (b) are taken by a Samsung Galaxy S22 with different camera parameters. SS and ISO indicate the shutter speed and sensor sensitivity, respectively. In (c), we visualize RGB histograms of each cropped patch of (a) and (b). Despite the same target scene, the captured images exhibit distinct histograms, implying complex ISP operations dependent on the camera parameters.

ues, e.g., the pre-set exposure time of commodity cameras roughly increases exponentially (1/250 sec., 1/125 sec., 1/60 sec., ...). Second, existing datasets lack of diversity in camera parameters [8, 21, 23], while a significant effort is required to collect a sufficient number of images for possible combinations of different camera parameters. Such scale difference and insufficient amount of training data make it difficult to reliably learn ISP operations for different camera parameters. Therefore, we propose a non-linear equalization scheme that adjusts the scales of the camera parameters and a random-dropout-based learning strategy to effectively learn the effects of all the camera parameters.

Finally, we present novel network architectures that reflect the real-world ISP operations. Specifically, our ISP networks consist of four subnetworks: *CanoNet*, *LocalNet* and *GlobalNet*, along with *ParamNet*. *CanoNet* performs common ISP operations such as demosaicing, white balance, and color space conversion using fixed operations without learnable weights. *GlobalNet* performs global tone manipulation with respect to the camera parameters. *LocalNet* performs other residual operations that are not captured by *CanoNet* and *GlobalNet*.

By incorporating camera parameters, novel network architectures, and training schemes, ParamISP achieves accurate reconstruction performance with a smaller network size. Our extensive evaluation shows that ParamISP surpasses previous learning-based ISP models by an average of 1.93 dB and 1.84 dB in RAW and sRGB reconstruction, respectively (Sec. 4). Furthermore, ParamISP is robustly applicable to various applications (e.g., deblurring dataset synthesis, RAW deblurring, HDR reconstruction, camera-to-camera transfer) (Sec. 5).

Our contributions are summarized as follows:

- We propose *ParamISP*, a novel learning-based forward and inverse ISP framework that leverages camera parameters (e.g., exposure time, sensitivity, aperture size, and focal length) for high-quality sRGB/RAW reconstruction.
- To effectively incorporate camera parameters, we develop *ParamNet*, a novel neural network module that controls the forward and inverse ISP networks according to camera parameters. We also introduce a non-linear equalization scheme and a random-dropout-based learning strategy for stable and effective learning of ISP operations with respect to camera parameters.
- We present novel network architectures for the forward and inverse ISP networks that better reflect real-world ISPs. (*CanoNet*, *LocalNet*, *GlobalNet*, *ParamNet*)
- We demonstrate the performance of *ParamISP* and its internal modules on many cameras, and show its applicability to various applications.

2. Related Work

Parametric ISPs To approximate the mapping between RAW and sRGB images, ISP approaches based on non-learnable parametric operations [4, 7] have been proposed. They employ simple invertible ISPs that are composed of non-learnable operations using either camera parameters [4] (e.g., white balance and color correction matrices) or parameters learned from DNNs [7]. However, they do not consider complex nonlinear ISP operations (e.g., local tone mapping and denoising), resulting in inaccurate reconstruction of sRGB and RAW images.

Learnable ISPs For more accurate approximation to forward and inverse ISP operations, DNN-based approaches [3, 29, 30] have been proposed, for which symmetrical forward and inverse ISP networks [3, 30] and an invertible ISP network [29] are employed to learn mapping between sRGB and RAW images. However, they are designed without considering camera parameters (e.g., exposure time, sensor sensitivity, etc), which limit their RAW and sRGB reconstruction quality.

Moreover, some of the previous methods primarily focus on cyclic reconstruction (sRGB-to-RAW-to-sRGB) and try to minimize the difference between the input sRGB image and the sRGB image restored back by the inverse and forward ISPs. Specifically, CycleISP [30] uses the input sRGB image for restoring the tone when reconstructing an sRGB image back from a RAW image, which makes it unsuitable for applications that manipulate the tone in the RAW space. InvISP [29] employs a single normalizing flow-based invertible network for both inverse and forward processes. While this design choice leads to near perfect reconstruction quality for cyclic reconstruction, its quality significantly degrades when the intermediate RAW images are altered, making it unsuitable for applications that manipulate im-

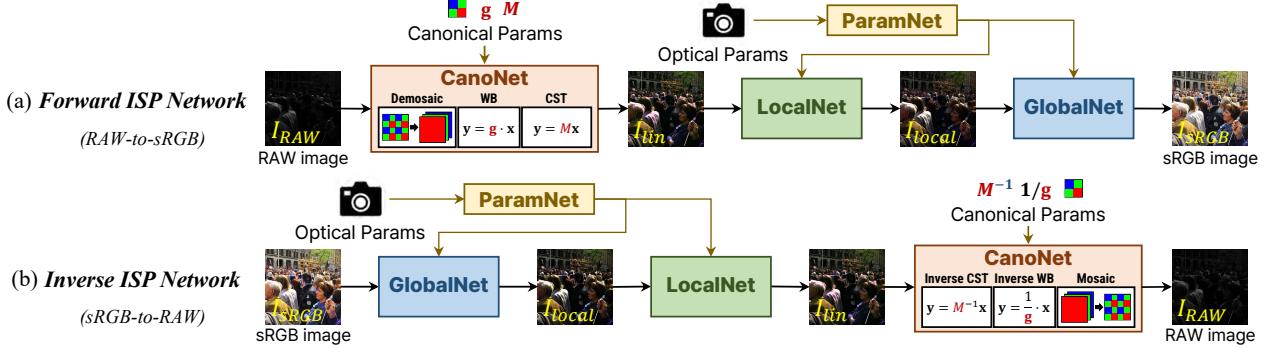


Figure 2. Overview of the proposed ParamISP framework. The full pipeline is constructed by combining learnable networks (ParamNet, LocalNet, GlobalNet) with invertible canonical camera operations (CanoNet). CanoNet consists of differentiable operations without learnable weights, where WB and CST denote white balance and color space transform, respectively.

ages in the RAW space as demonstrated in Sec. 5. In contrast, we design our forward and inverse ISP networks to operate independently during inference time to cater to a broader range of applications.

RAW Reconstruction using Additional Information To achieve precise RAW reconstruction from an sRGB image, another line of research that utilizes additional information has been introduced [11, 18, 20, 26]. This approach encodes necessary metadata such as a small portion of a RAW image within an sRGB image at capture time to reconstruct the original RAW image with high accuracy. However, it necessitates a modification to the existing ISP process to store the metadata. In contrast, our approach utilizes the EXIF metadata that commodity cameras already provide.

3. ParamISP

Given a target camera, our goal is to learn its forward and inverse ISP processes that change with respect to camera parameters. To accomplish this, ParamISP is designed to have a pair of forward (RAW-to-sRGB) and inverse (sRGB-to-RAW) ISP networks (Fig. 2). Both networks are equipped with ParamNet so that they adaptively operate based on camera parameters.

In ParamISP, we classify camera parameters into two distinct categories: optical parameters (including exposure time, sensitivity, aperture size, and focal length) and canonical parameters (Bayer pattern, white balance coefficients, and a color correction matrix). The canonical parameters directly influence fundamental ISP operations like demosaicing, white balance adjustments, and color space conversion. These operations are relatively straightforward, and their relationship with the canonical parameters is well-defined.

While previous approaches have leveraged the canonical parameters [4, 7], the optical parameters have remained untapped. In contrast, ParamISP exploits both sets of parameters to achieve highly accurate sRGB and RAW reconstruction. To harness the canonical parameters, our ISP networks

incorporate CanoNet, a subnetwork that performs canonical ISP operations without learnable weights. For the optical parameters, we introduce ParamNet, which is the key component to dynamically control the behavior of the ISP networks based on the optical parameters.

In the following, we describe ParamNet and our training strategy for stable and effective training. We then explain the forward and inverse ISP subnetworks in detail.

3.1. ParamNet

As described in Fig. 3, ParamNet takes optical parameters as input and converts them into an optical parameter feature vector z , which is then fed to both LocalNet and GlobalNet. ParamNet consists of a non-linear equalization layer, and fully-connected layers. The non-linear equalization layer computes a normalized feature vector for each optical parameter to compensate for the scale difference between the optical parameters. The equalized feature vectors are then arithmetically summed together, and fed to the fully-connected layers to obtain an optical parameter feature vector z .

Non-linear Equalization As discussed in Sec. 1, the optical parameters exhibit significantly different scales, and their values are non-linearly distributed. While some optical parameters may substantially alter the behavior of the camera ISP, others may have minimal effect. Moreover, the influence of optical parameters may vary across different camera models. As a result, naively incorporating the optical parameters leads to unstable training of ParamNet.

To resolve this issue, the non-linear equalization layer of ParamNet applies various non-linear mapping to each optical parameter and learns the best composition of them that enables stable training. Specifically, we first apply non-linear mapping functions such as $x, 1/x, \sqrt{x}, x^{-1/2}, x^{1/4}, x^{-1/4}, \log(x), \sin(\log(x)), \cos(\log(x)), \sin(c \cdot x)$, and $\cos(c \cdot x)$ to each optical parameter, where x is the value of an optical parameter, and c controls the frequency of the sinusoidal functions, for which we use three different val-

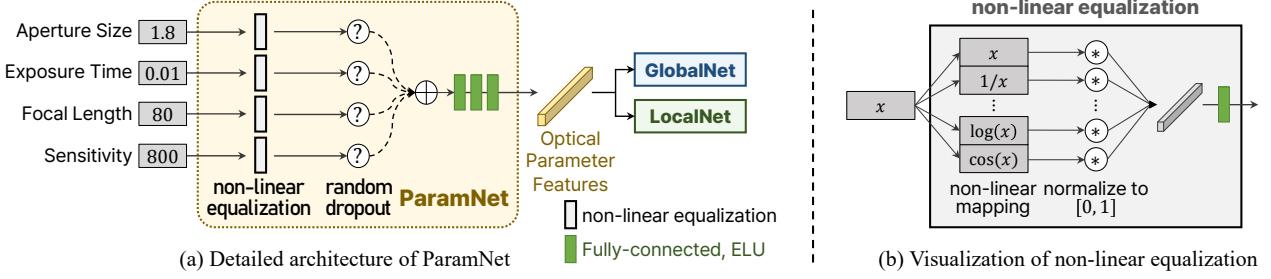


Figure 3. Architecture of ParamNet. (a) Given camera optical parameters, ParamNet estimates optical parameter features used for modulating the LocalNet and GlobalNet. (b) In order to deal with different scales and non-linearly distributed values of optical parameters, we propose to use non-linear equalization that exploits multiple non-linear mapping functions.

ues empirically chosen. We then normalize each of the non-linear mapping results into $[0, 1]$. As a result, we obtain a 15-dim. vector for each optical parameter. Then, each vector is processed through a subsequent fully-connected layer to extract useful information from each parameter vector.

Our non-linear equalization layer incorporates a range of diverse non-linear mapping functions, rather than relying on a predefined set of carefully-chosen functions, in order to cover the wide spectrum of potential relationships between each optical parameter and the camera ISP. Thus, some mapping functions may look redundant or unnecessary. Nevertheless, the subsequent fully-connected layer can successfully learn to extract only useful information from them by combining them with different weights.

Random Dropout Even with the non-linear equalization, ParamNet can still be trained to overfit to a subset of the optical parameters due to the scale difference and the insufficient amount of training data. To mitigate the problem, during training, we randomly drop out the equalized feature vector of each optical parameter. Specifically, each optical parameter is randomly dropped out by the probability of 20% at each training iteration.

3.2. Forward ISP Network

The forward ISP network consists of four subnetworks: CanoNet, LocalNet, GlobalNet, and ParamNet (Fig. 2(a)). CanoNet performs canonical ISP operations: demosaicing [9], white balance, and color space conversion, exploiting the canonical parameters. LocalNet performs local operations of an ISP such as denoising, sharpening and local tone mapping in addition to compensating for the residual errors of the canonical ISP operations of CanoNet. GlobalNet performs global tone manipulation with respect to the camera parameters. ParamNet is connected to LocalNet and GlobalNet and controls their behavior. In the following, we explain each of CanoNet, LocalNet and GlobalNet in detail.

CanoNet CanoNet takes a RAW image $I_{RAW} \in \mathbb{R}^{1 \times H \times W}$ and first performs demosaicing [9] to produce $I_{Dem} \in \mathbb{R}^{3 \times H \times W}$ according to the Bayer pattern of I_{RAW} . The demosaicing algorithm of CanoNet may differ from that

of the target camera ISP. Nonetheless, such discrepancy is compensated by the subsequent LocalNet. It then performs white balance using the coefficients $gWB \in \mathbb{R}^3$. Lastly, it transforms the color space from the RAW space to the linear sRGB space using the color correction matrix $M_{Cam} \in \mathbb{R}^{3 \times 3}$. We denote the resulting image as $I_{lin} \in \mathbb{R}^{3 \times H \times W}$.

LocalNet Fig. 4 shows an overview of LocalNet. LocalNet takes I_{lin} and an optical parameter feature vector z from ParamNet as input, and performs local ISP operations. As well as I_{lin} , LocalNet also takes additional handcrafted features: a gradient map, a soft histogram map, and an over-exposure mask computed from I_{lin} , as they help improve reconstruction quality, as shown in [12–14]. For more details, refer to the supplementary material. Finally, LocalNet predicts a residual image, which is then added to I_{lin} to obtain the output image I_{local} .

As the goal of LocalNet is to learn local ISP operations, LocalNet adopts a network architecture based on the U-Net [22], which has been proven to be highly effective for various image restoration and enhancement tasks [2, 6, 10, 27, 31]. Specifically, the input image and the additional input features computed from the input image are concatenated, and fed to LocalNet. Besides, the optical parameter feature vector z is duplicated horizontally and vertically to build a feature map of the same spatial size as the input image. The feature map is then added to an intermediate feature map of LocalNet. Then, the resulting feature map is processed through multi-scale residual blocks (ResBlocks) with convolutional block attention modules (CBAMs) [28], and converted to a residual image.

GlobalNet GlobalNet globally adjusts tone and color according to the content of its input image and the optical parameter feature vector z . To this end, GlobalNet adopts parametric global operations and predicts their parameters. Specifically, GlobalNet adopts two types of parametric global operations: quadratic transformation f_q and gamma correction f_g . Let us denote the input image to global adjustment operation as I , and its pixel as $p = (p_r, p_g, p_b)$. Then, the quadratic transformation f_q is defined as:

$$f_q(p) = Wp' \quad (1)$$

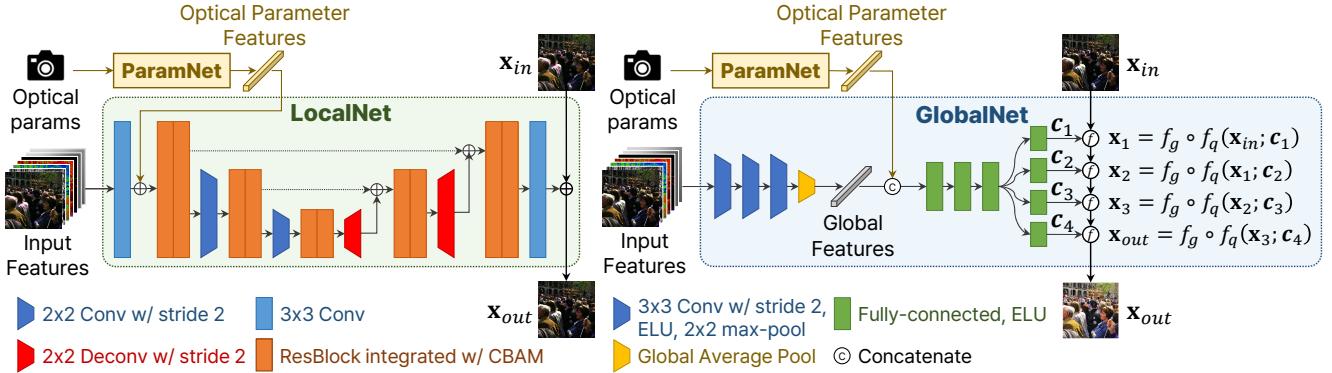


Figure 4. Detailed architecture of LocalNet and GlobalNet. In GlobalNet, f_g and f_q represent the gamma correction and quadratic transformation, respectively, while c_n represents the n -th coefficients G^n and W^n , as explained in Sec. 3.2.

where p' is a 10-dim. vector defined as $p' = [p_r^2, p_g^2, p_b^2, p_r p_g, p_g p_b, p_b p_r, p_r, p_g, p_b, 1]^T$. W is a 3×10 matrix with the coefficients of the quadratic transformation, which is uniformly applied to all the pixels in I . The gamma correction operator f_g is defined as:

$$f_{g,c}(p_c) = \frac{(\alpha_c p_c + \beta_c)^{\gamma_c} - \beta_c^{\gamma_c}}{(\alpha_c + \beta_c)^{\gamma_c} - \beta_c^{\gamma_c}} \quad (2)$$

where the subscript c is an index to each color channel, i.e., $c \in \{r, g, b\}$. γ_c is a gamma parameter, while α_c and β_c are a scale and an offset, respectively. The set of the gamma correction coefficients is denoted by G , i.e., $G = \{\alpha_r, \beta_r, \gamma_r, \alpha_g, \beta_g, \gamma_g, \alpha_b, \beta_b, \gamma_b\}$. Similar to W , G is also uniformly applied to all the pixels in I . For the sake of notational simplicity, we represent the quadratic transformation and gamma correction of an entire image I as $f_q(I)$ and $f_g(I)$, respectively, in the rest of the paper.

To support a wide range of potential non-linear tone adjustments of commodity camera ISPs, GlobalNet models the global tone adjustment as a series of gamma correction and quadratic transformation. Specifically, given an input image I , GlobalNet performs global tone adjustment as:

$$\hat{I} = f_{gq}^N \circ \dots \circ f_{gq}^1(I) \quad (3)$$

where $f_{gq}^n(I)$ is defined as $f_{gq}^n = f_g^n(f_q^n(I))$. f_g^n and f_q^n are the gamma correction and quadratic transformation with the n -th coefficients G^n and W^n , respectively. In our experiments, we use $N = 4$ as it leads to the best quality (Refer to Tab. S3 in the supplementary material).

The quadratic transformation is also adopted by previous modular learnable ISPs [3, 23]. However, a single quadratic transformation cannot accurately model diverse non-linear tone adjustments performed by commodity camera ISPs including gamma correction. Therefore, in our approach, we extend the global tone adjustment by adopting the gamma correction and iteratively applying both gamma correction and quadratic transformation.

Fig. 4 shows an overview of GlobalNet. To predict the coefficients $\{\dots, (G^n, W^n), \dots\}$, GlobalNet takes a concatenation of I_{local} and the handcrafted features computed from I_{local} as done in LocalNet as input, and extracts a global feature vector through a series of convolution layers with max pooling, and a global average pooling layer. The global feature vector is then added to the optical parameter feature vector, and processed through fully-connected layers to obtain the coefficients.

3.3. Inverse ISP Network

The inverse ISP network also consists of CanoNet, LocalNet, GlobalNet and ParamNet with an inverse order of the forward ISP network (Fig. 2(b)). Specifically, GlobalNet takes an sRGB image as input and performs inverse tone manipulation. Next, LocalNet executes inverse local operations, and finally, CanoNet handles inverse color space conversion, inverse white balance, and mosaicing. GlobalNet and LocalNet use the same network architectures as those in the forward ISP network. On the other hand, CanoNet consists of the inverse operations of those of the forward ISP network, but in reverse order.

4. Experiments

Implementation We used PyTorch [19] to implement our models. We train our model in two stages: pre-training and fine-tuning. In the pre-training stage, we train our models with multiple datasets captured from multiple cameras. In the fine-tuning stage, we train our models on a specific target camera. Although our models aim at learning the ISP of a single target camera model, we found that pre-training with multiple cameras substantially improves the reconstruction quality. In the pre-training stage, we use the RAISE dataset [8] from Nikon D7000, D90, and D40, the RealBlur dataset [21] from Sony A7R3, and the S7 ISP dataset [23] from Samsung Galaxy S7. More details such as the effect of the pre-training, the statistics of the datasets, and how we split the datasets to training, validation, and test

| Baseline (Ours w/o ParamNet) | | | | |
|------------------------------|----------------------------|-------------------|--------------|--------|
| ParamNet (w/ Opt. Params) | Non-linear Equalization | Random Dropout | PSNR↑ | Param↓ |
| ✓ | | | 34.77 | 0.68M |
| ✓ | ✓ | | - | - |
| ✓ | ✓ | ✓ | 35.64 | 0.71M |
| | | | 36.21 | 0.71M |

Table 1. Ablation study on the components of ParamNet.

| Optical Params | w/o A | w/o B | w/o C | w/o D | Full |
|----------------|--------|--------|---------------|--------|---------------|
| PSNR↑ | 35.13 | 35.25 | 35.87 | 35.46 | 36.21 |
| SSIM↑ | 0.9678 | 0.9718 | 0.9724 | 0.9705 | 0.9724 |

Table 2. Ablation study on the impact of each optical parameter. A, B, C , and D represent sensor sensitivity, exposure time, aperture size, and focal length, respectively.

sets can be found in the supplementary material.

We train our network in the pre-training stage for 520 epochs with an initial learning rate of 2.0×10^{-4} , and in the fine-tuning stage for 2,140 epochs with an initial learning rate of 2.0×10^{-5} . For both pre-training and fine-tuning, we employ AdamW [15] and the learning rate is step-decayed with a rate of 0.8 every 10 epochs. In each epoch in the pre-training stage, we randomly sample 1024 cropped patches of size 448×448 from the dataset of each camera, resulting in a total of 5120 patches. In the fine-tuning stage, we sample 1024 patches from the dataset of a target camera in each epoch. We evaluated our models and other models on a PC equipped with an NVIDIA RTX A6000. For a fair comparison, we also apply both pre-training and fine-tuning to other models in our experiments.

We train our forward and inverse ISP networks in an end-to-end fashion using losses \mathcal{L}_{for} and \mathcal{L}_{inv} :

$$\mathcal{L}_{for} = \|\hat{I}_{sRGB} - I_{sRGB}\|_1 \quad (4)$$

$$\mathcal{L}_{inv} = \|\hat{I}_{RAW} - I_{RAW}\|_1 \quad (5)$$

respectively, where \hat{I}_{sRGB} , I_{sRGB} , \hat{I}_{RAW} , and I_{RAW} are a reconstructed sRGB image, its ground-truth sRGB image, a reconstructed RAW image, and its ground-truth RAW image, respectively.

4.1. Ablation Study

For all the ablation studies in this section, we train inverse (sRGB-to-RAW) ISP network using the D7000 training images of the RAISE dataset [8]. Additional ablation studies using the forward ISP network and different camera datasets, and on the effect of the input features and training approach can be found in the supplementary material.

ParamNet We first evaluate the effect of the non-linear equalization and random optical parameter dropout of ParamNet. To this end, we compare variants of our model in Tab. 1. The first row in the table corresponds to a baseline model without ParamNet. Introducing ParamNet without non-linear equalization (2nd row) results in the failure of

| Baseline (CIE XYZ Net [3]) | | PSNR↑ | SSIM↑ | Param↓ |
|----------------------------|-----------|--------------|---------------|--------|
| LocalNet | GlobalNet | | | |
| ✓ | | 30.04 | 0.9461 | 1.3M |
| ✓ | ✓ | 33.12 | 0.9644 | 1.7M |
| | | 33.66 | 0.9646 | 0.6M |

Table 3. Ablation study on the effects of LocalNet and GlobalNet.

training with diverging losses. On the other hand, the non-linear equalization (3rd row) not only stabilizes the training but also improves the reconstruction quality compared to the baseline model in the first row, which proves that the non-linear equalization is essential for training, and that exploiting the optical parameters is necessary to improve the reconstruction quality. Finally, random dropout of the optical parameters (4th row) further enhances the reconstruction quality as it prevents overfitting and enhances the generalization ability of ParamNet.

Optical Parameters We now analyze the impact of the optical parameters. To this end, we compare four variants of ParamNet, for each of which, we exclude each one of the four optical parameters. Tab. 2 shows the quantitative ablation results. Compared to the model that uses all the optical parameters, all the other variants exhibit performance degradation, indicating that all the optical parameters need to be considered for high-quality reconstruction. We can also observe that excluding the sensitivity and exposure time shows the largest performance degradation. While how commodity camera ISPs utilize the optical parameters is unknown, these optical parameters may affect the image enhancement operation of camera ISPs, such as denoising, as they are closely related to noise. The impact of the aperture size and focal length is not as significant as the others, but the reconstruction quality still decreases without them. This is because aperture size and focal length are related to defocus blur and lens aberrations, so the camera ISP operations may alter based on them.

ISP Network Architecture We verify the effectiveness of the network architecture of our ISP subnetworks. As our modular network structure is inspired by CIE XYZ Net [3] that consists of a pair of local and global mapping modules, we consider its local and global mapping networks as our baseline for LocalNet and GlobalNet. Specifically, we build a baseline model for the inverse ISP network by replacing the LocalNet and GlobalNet with the local and global mapping networks of CIE XYZ Net, then compare the performance of the baseline and its variants to ours. The local mapping network of CIE XYZ Net simply stacks convolution layers, while the global mapping network models global tone manipulation as a single quadratic function and estimate its parameters using fully-connected layers. Tab. 3 shows the comparison result. The table shows that LocalNet significantly improves the PSNR by more than 3 dB over the local mapping of CIE XYZ Net. Moreover, GlobalNet further improves the PSNR by around 0.5 dB and leads to

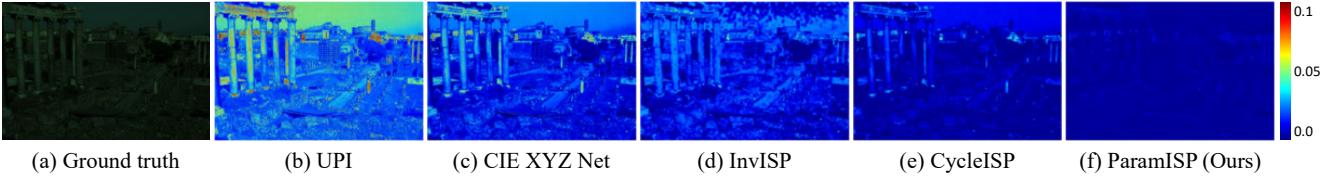


Figure 5. sRGB-to-RAW reconstruction. We show error maps between reconstructed and GT RAW images.

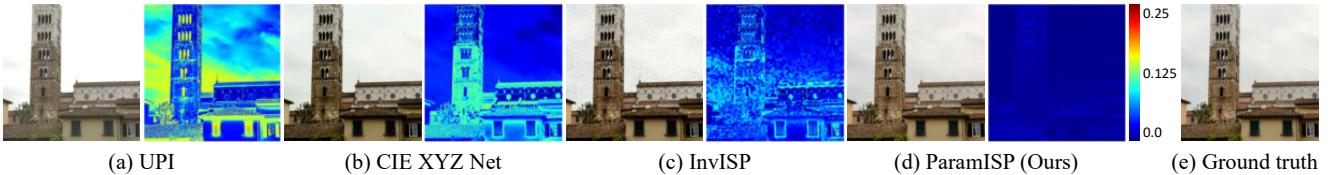


Figure 6. RAW-to-sRGB reconstruction. We show error maps between reconstructed and GT sRGB images.

| Method | sRGB → RAW measured in PSNR↑ | | | | | Param↓ |
|-----------------|------------------------------|--------------|--------------|--------------|--------------|--------|
| | D7000 | D90 | D40 | S7 | A7R3 | |
| UPI [4] | 20.67 | 26.57 | 22.05 | 29.98 | 30.48 | - |
| CIE XYZ Net [3] | 30.04 | 32.62 | 38.57 | 33.24 | 36.42 | 1.3M |
| CycleISP [30] | 35.52 | 35.85 | 42.83 | 34.55 | 45.35 | 3.1M |
| InvISP [29] | 33.48 | 35.39 | 45.08 | 34.29 | 47.14 | 1.4M |
| ParamISP (Ours) | 38.49 | 37.06 | 45.97 | 35.20 | 48.33 | 0.7M |

Table 4. Quantitative comparison on RAW reconstruction.

a significantly reduced model size thanks to our global tone adjustment model and efficient network architecture.

4.2. RAW & sRGB Reconstruction

In this section, we compare the sRGB-to-RAW and RAW-to-sRGB reconstruction ability of ParamISP with existing state-of-the-art methods: UPI [4], CIE XYZ Net [3], CycleISP [30], and InvISP [29]. UPI is a non-learnable parametric method that approximates the general structure of the camera ISP. CIE XYZ Net, CycleISP, and InvISP are learning-based approaches that employ symmetrical forward and inverse ISP networks. For CIE XYZ Net whose output is an image in the CIE XYZ color space, we convert its output to a RAW image using canonical ISP operations as in CanoNet for a fair comparison. For InvISP whose output is white-balanced RAW image, we apply inverse white balancing and mosaicing operations to obtain a RAW image. These learning-based methods train the forward and inverse networks together, but we found that separately training them for each yields better performance. Therefore, we train them separately in the same manner as ours.

Tab. 4 shows quantitative results of the inverse sRGB-to-RAW reconstruction in terms of PSNR evaluated on our test set. UPI [4] results in high reconstruction errors due to its parametric model-based pipeline. CIE XYZ Net [3], CycleISP [30], and InvISP [29] show better reconstruction quality than UPI, but their quality is still limited compared to our method. ParamISP achieves the best reconstruction quality despite its much smaller model size, thanks to leveraging the optical parameters and our carefully-designed net-

| Method | RAW → sRGB measured in PSNR↑ | | | | | Param↓ |
|-----------------|------------------------------|--------------|--------------|--------------|--------------|--------|
| | D7000 | D90 | D40 | S7 | A7R3 | |
| UPI [4] | 18.81 | 20.30 | 16.01 | 20.05 | 19.37 | - |
| CIE XYZ Net [3] | 26.76 | 27.61 | 34.84 | 27.63 | 37.19 | 1.3M |
| InvISP [29] | 30.20 | 28.89 | 37.86 | 28.96 | 43.93 | 1.4M |
| ParamISP (Ours) | 34.14 | 30.83 | 39.54 | 29.02 | 45.51 | 0.7M |

Table 5. Quantitative comparison on sRGB reconstruction.

work architecture. Fig. 5 shows the qualitative results of error maps between reconstructed RAW images and ground-truth RAW images. As the figure shows, our method produces much less error than the others.

Tab. 5 reports quantitative results of the forward reconstruction (RAW-to-sRGB). CycleISP is not included in this comparison because it needs an input sRGB image for sRGB reconstruction in the sRGB-to-RAW-to-sRGB cyclic reconstruction. Similar to the inverse sRGB-to-RAW reconstruction, ParamISP clearly outperforms the other methods, showing the effectiveness of our approach.

5. Applications

Forward and inverse ISP models can benefit various applications as shown in [3, 29, 30]. In this section, we demonstrate a couple of applications of ParamISP: RAW deblurring and HDR reconstruction. Other additional applications including deblurring dataset synthesis and camera-to-camera transfer are also provided in the supplementary material. Before applying ParamISP to applications, we further perform joint fine-tuning on separately trained forward and inverse ISP networks. For details on the joint fine-tuning, refer to the supplementary material.

Deblurring in RAW Space While the irradiance of a blurred image has a linear relationship with its latent sharp image, such a linear relationship no longer holds in sRGB images after non-linear operations have been applied by camera ISPs. Such non-linearity varies across different cameras. Consequently, learning sRGB image deblurring requires camera-specific training datasets, which leads to

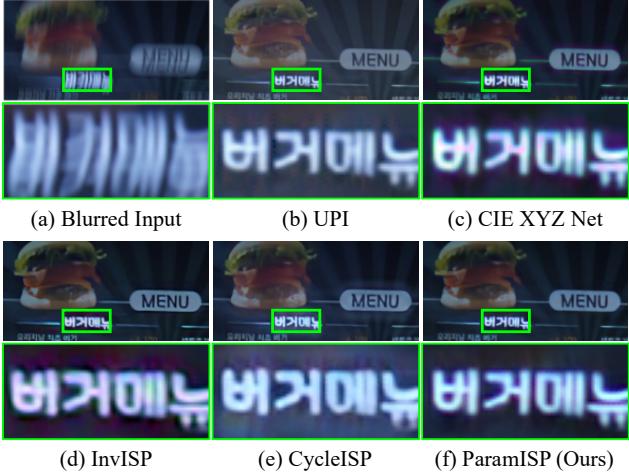


Figure 7. Qualitative results of RAW deblurring. In (c) and (d), red artifacts are visible in white text, while in (b) and (e), the results are blurrier than ours.

| Method | UPI [4] | CIE XYZ Net [3] | InvISP [29] | CycleISP [30] | ParamISP (Ours) |
|--------|---------|-----------------|-------------|---------------|-----------------|
| PSNR↑ | 24.63 | 28.30 | 28.82 | 29.94 | 30.70 |
| SSIM↑ | 0.8011 | 0.8444 | 0.8721 | 0.8855 | 0.8984 |
| Param↓ | - | 2.7M | 1.4M | 7.4M | 1.4M |

Table 6. Quantitative results of RAW deblurring.

an excessive amount of training time and memory space to support diverse camera models. Instead, we can effectively reconstruct a RAW image from an sRGB image using ParamISP, apply a camera-independent deblurring model to the RAW image, and obtain a deblurred sRGB image. While ParamISP also requires camera-wise training, it requires a much smaller number of parameters ($2 \times 0.7\text{M}$) compared to deblurring models (Stripformer: 20M [25] and Uformer-B: 51M [27]), and supports a wide range of applications.

Tab. 6 and Fig. 7 show quantitative and qualitative comparisons of deblurring results in the RAW space using different ISP networks. For deblurring in the RAW space, we use Stripformer [25] trained on the RealBlur-R dataset, which is a real-world RAW blurry image dataset [21]. The evaluation is done on the RealBlur-J test set [21]. As the comparisons show, ParamISP clearly outperforms all the other ISP networks in deblurring performance thanks to its high-quality sRGB and RAW reconstruction.

HDR Reconstruction As shown in CIE XYZ Net [3], RAW image reconstruction can also be used for high-dynamic-range (HDR) image reconstruction from a single low-dynamic-range (LDR) image as RAW images provide a wider tonal value range. Specifically, we first convert an LDR sRGB image to an LDR RAW image, and multiply the reconstructed RAW image by synthetic digital gains (0.1, 1.4, 2.7, 4.0) to create four RAW images. Each of these is then passed through the forward ISP to reconstruct sRGB images. Finally, we apply an off-the-shelf exposure-fusion

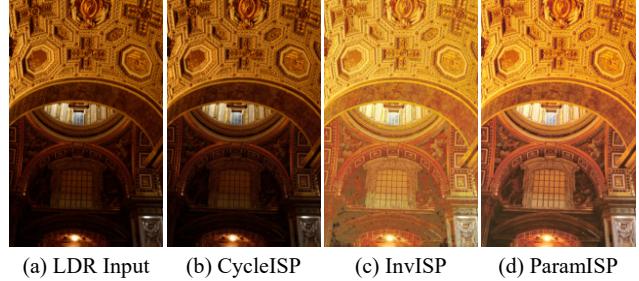


Figure 8. Qualitative results of HDR reconstruction.

algorithm [16] to obtain a single HDR image.

Fig. 8 shows qualitative results using different models. Previous methods primarily focus on cyclic reconstruction. Specifically, CycleISP uses an input LDR sRGB image for the forward ISP to adjust the tone, while InvISP, composed of a single flow-based network, is not robust when intermediate RAW images are altered. On the other hand, ParamISP successfully produces an HDR reconstruction result compared to other methods. For more qualitative results, refer to the supplementary material.

6. Conclusion

In this paper, we present ParamISP, a novel learning-based forward and inverse ISP framework that leverages camera parameters. To incorporate camera parameters effectively, we introduce ParamNet to control the forward and inverse ISP networks, proposing a stable and effective training strategy with respect to camera parameters. We also present novel network architectures for ISP networks that better reflect real-world ISP operations. Through our extensive experiments, ParamISP shows the state-of-the-art performance in RAW and sRGB reconstruction and the robust applicability to various applications.

Limitations and Future Work While we have quantitatively confirmed that incorporating the aperture size and focal length improves the reconstruction quality in Tab. 2, it is still unclear how such parameters affect the behaviors of the ISP operations. While we use datasets captured by various cameras in the pre-training stage, different ISPs may behave differently with respect to different optical parameters, and such inconsistency may potentially have negative impact on a pre-trained ISP model. A more sophisticated training strategy may resolve such issue.

Acknowledgments This work was supported by the NRF grants (RS-2023-00211658, RS-2023-00280400, 2022R1A6A1A03052954, 2023R1A2C200494611) and IITP grant (2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH)) funded by the Korea government (MSIT). This work was also supported by Samsung Research Funding Center (SRFCIT1801-52) and Samsung Electronics Co.

References

- [1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1
- [2] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *Proceedings of the European conference on computer vision (ECCV)*, 2020. 4
- [3] Mahmoud Afifi, Abdelrahman Abdelhamed, Abdullah Abuolaim, Abhijith Punnappurath, and Michael S Brown. CIE XYZ Net: Unprocessing images for low-level computer vision tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(9):4688–4700, 2021. 1, 2, 5, 6, 7, 8
- [4] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 7, 8
- [5] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1
- [6] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. Deep photo enhancer: Unpaired learning for image enhancement from photographs with GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4
- [7] Marcos V. Conde, Steven McDonagh, Matteo Maggioni, Ales Leonardis, and Eduardo Pérez-Pellitero. Model-based image signal processors via learnable dictionaries. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 36(1):481–489, 2022. 1, 2, 3
- [8] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. RAISE: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM multimedia systems conference (MMSys)*, 2015. 2, 5, 6
- [9] Pascal Getreuer. Malvar-he-cutler linear image demosaicking. *Image Processing on Line*, 1:83–89, 2011. 4
- [10] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8183–8192, 2018. 4
- [11] Leyi Li, Huijie Qiao, Qi Ye, and Qinmin Yang. Metadata-based raw reconstruction via implicit neural functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [12] Steve Lin and L. Zhang. Determining the radiometric response function from a single grayscale image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 4
- [13] Steve Lin, Jinwei Gu, Shuntaro Yamazaki, and Harry Shum. Radiometric calibration from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [14] Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Single-image hdr reconstruction by learning to reverse the camera pipeline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 4
- [15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 6
- [16] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *15th Pacific Conference on Computer Graphics and Applications (PG)*, 2007. 8
- [17] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1
- [18] Seonghyeon Nam, Abhijith Punnappurath, Marcus A Brubaker, and Michael S Brown. Learning srgb-to-raw-rgb de-rendering with content-aware metadata. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3
- [19] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *Proceedings of the Neural Information Processing Systems Workshops (NeurIPSW)*, 2017. 5
- [20] Abhijith Punnappurath and Michael S Brown. Spatially aware metadata for raw reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. 3
- [21] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking de-blurring algorithms. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 5, 8
- [22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention (MICCAI)*, 2015. 4
- [23] Eli Schwartz, Raja Giryes, and Alex M Bronstein. DeepISP: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing (TIP)*, 28(2):912–923, 2018. 1, 2, 5
- [24] Chengzhou Tang, Yuqiang Yang, Bing Zeng, Ping Tan, and Shuaicheng Liu. Learning to zoom inside camera imaging pipeline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1
- [25] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, and Chia-Wen Lin. Stripformer: Strip transformer for fast image deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 8
- [26] Yufei Wang, Yi Yu, Wenhuan Yang, Lanqing Guo, Lap-Pui Chau, Alex C Kot, and Bihan Wen. Raw image reconstruction with learned compact metadata. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

- [27] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [4](#), [8](#)
- [28] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, 2018. [4](#)
- [29] Yazhou Xing, Zian Qian, and Qifeng Chen. Invertible image signal processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [1](#), [2](#), [7](#), [8](#)
- [30] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [1](#), [2](#), [7](#), [8](#)
- [31] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [4](#)
- [32] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [1](#)

ParamISP: Learned Forward and Inverse ISPs using Camera Parameters

— Supplementary Material —

Woohyeok Kim^{1*} Geonu Kim^{1*} Junyong Lee^{2†}
 Seungyong Lee¹ Seung-Hwan Baek¹ Sunghyun Cho¹
¹POSTECH ²Samsung AI Center Toronto

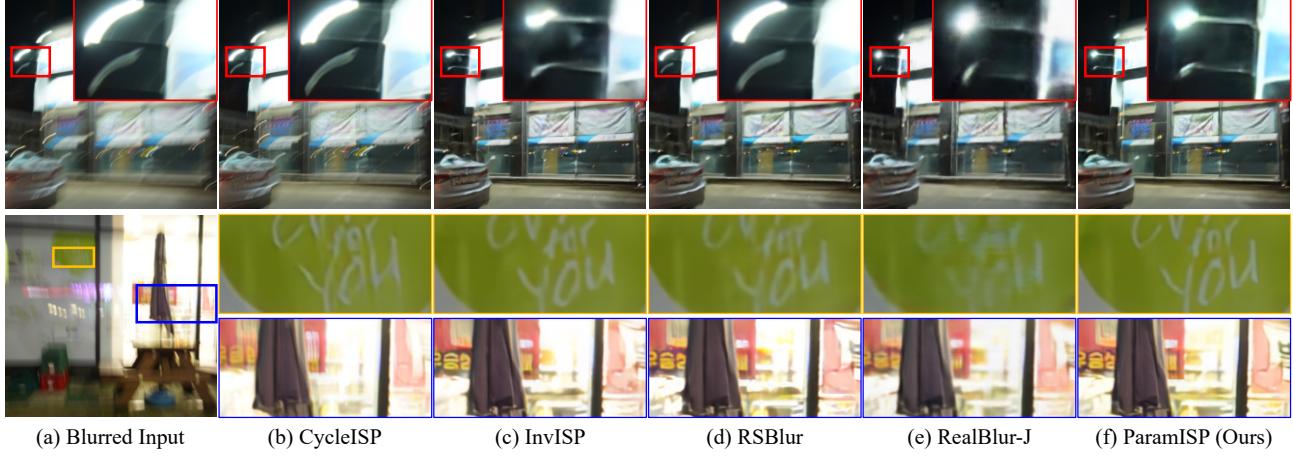


Figure S1. Qualitative results of deblurring dataset synthesis. Compared to other methods, the deblurring model [3] trained on the dataset synthesized by our ISP models more effectively restores sharp details.

S1. Overview

In this supplementary material, we describe two additional applications: deblurring dataset synthesis and camera-to-camera transfer (Sec. S2). We then provide details and an ablation study on the input features used in LocalNet and GlobalNet (Sec. S3), as well as additional experiments on the global adjustment operation in GlobalNet (Sec. S4) and additional discussions on the training strategy (Sec. S5). We also provide experimental results on the relationship between optical parameters and ParamNet (Sec. S6). Finally, we present additional quantitative and qualitative results (Sec. S7), along with the detailed architecture of ParamISP (Sec. S8).

S2. Additional Applications

ParamISP can be applied to various applications (*e.g.*, deblurring dataset synthesis, RAW deblurring, HDR reconstruction, and camera-to-camera transfer) unlike previous methods. In this section, we describe two additional applications (*i.e.*, deblurring dataset synthesis, camera-to-camera transfer) that were not covered in the main paper. We find that joint fine-tuning on separately trained forward and inverse ISP networks brings additional performance improvements in applications. Therefore, before applying ParamISP to applications excluding camera-to-camera transfer, we conduct additional joint fine-tuning.

In the joint fine-tuning stage, we train our pretrained forward and inverse ISP networks for 450 epochs with an initial learning rate of 1.0×10^{-4} . We jointly fine-tune separately trained forward and inverse ISP networks in an end-to-end manner using loss \mathcal{L}_{joint} :

$$\begin{aligned} \mathcal{L}_{joint} = & \|f_{for}(f_{inv}(I_{sRGB})) - I_{sRGB}\|_1 \\ & + \|f_{inv}(I_{sRGB}) - I_{RAW}\|_1, \end{aligned} \quad (\text{S1})$$

* Equal contribution.

† Work done prior to joining Samsung.

| Method | Synthetic | | | | Real |
|--------|------------------|----------------|----------------|--------------------|-------------------|
| | CycleISP [14] | InvISP [13] | RSBlur [10] | ParamISP (Ours) | RealBlur-J [9] |
| PSNR↑ | 29.98 | 31.06 | 31.16 | 31.31 | 31.82 |
| SSIM↑ | 0.8806 | 0.9134 | 0.9142 | 0.9148 | 0.9203 |

Table S1. Quantitative results of deblurring dataset synthesis. Our dataset synthesis outperforms all the other synthesis methods. While ours still achieves lower performance than the RealBlur-J training set, this is partly due to the photometric misalignment present in the RealBlur-J dataset.

where f_{for} and f_{inv} are the forward and inverse ISP networks, respectively, and I_{sRGB} is an sRGB image. Other training conditions are the same as those explained in Sec. 4 of the main paper.

S2.1. Deblurring Dataset Synthesis

It is essential to reflect the camera ISP in order to synthesize a realistic image restoration dataset such as deblurring datasets [10]. ParamISP can enhance the accuracy of synthetic deblurring datasets as it can more accurately model real-world camera ISPs. Tab. S1 shows a comparison among different dataset synthesis approaches. On the table, RSBlur [10] is a baseline model that uses a simple parametric ISP model, while InvISP [13], CycleISP [14], and ParamISP mean variants of the RSBlur pipeline whose ISP model is replaced by the corresponding ISP models. For fair comparisons, we train the other ISP models according to their respective learning approaches. We synthesize deblurring datasets using each of the approaches on the table, train a deblurring model [3] using each dataset, and evaluate their performance on the RealBlur-J test set [9], which is a real-world blur dataset. The last column of the table represents the results obtained by directly training the deblurring model on the RealBlur-J train set, and we consider this as the upper bound.

As Tab. S1 shows, ParamISP achieves the closest PSNR to the upper bound, outperforming all the other synthesis approaches. Interestingly, InvISP [13] and CycleISP [14] achieve worse performance than RSBlur [10] although they are learnable approaches. This is because they primarily focus on cyclic reconstruction (sRGB-to-RAW-to-sRGB) and are less suitable for applications that manipulate RAW images as described in the main paper. Specifically, CycleISP uses the input sRGB image for restoring the tone when reconstructing an sRGB image back from a RAW image. Here, for the smooth tone restoration of CycleISP, we apply the same blur kernel used to create the blurry RAW image to the input sharp sRGB image, rather than using the input sharp sRGB image directly. InvISP uses a single normalizing flow-based invertible network, resulting in near-perfect reconstruction quality for cyclic reconstruction. However, its quality significantly degrades when the intermediate RAW images are altered.

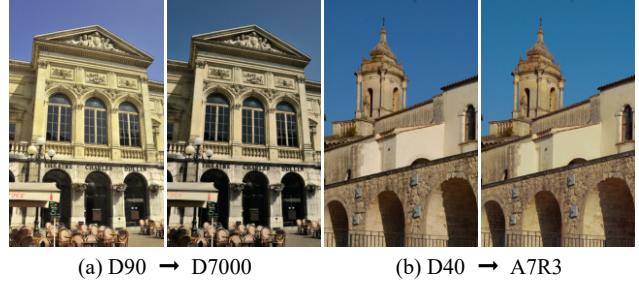


Figure S2. Qualitative results of camera-to-camera transfer.

While our method outperforms all the other synthetic dataset generation processes both in PSNR and SSIM, it still achieves lower performance than the upper bound model trained using the RealBlur-J training set. We emphasize that the performance gap between the upper bound and ours is partly due to the existence of remaining photometric misalignment in the blurry and sharp image pairs in the RealBlur-J dataset. The sharp and blurry images of the RealBlur-J dataset were captured by different cameras, so they have slightly different tones. While the postprocessing process of the RealBlur-J dataset applies photometric alignment to mitigate this issue, the images in the dataset still have remaining tone difference, which could only be learned from the RealBlur-J training set. Dataset synthesis methods that synthesize blurry images using only sharp images are unable to depict such tone difference between different cameras and, in fact, there is no need to depict such tone differences for the purpose of deblurring. Fig. S1 shows a qualitative comparison. The deblurring model trained with ParamISP visually outperforms each deblurring model trained with other methods, including the upper bound RealBlur-J.

S2.2. Camera-to-Camera Transfer

Given an sRGB image, ParamISP can manipulate the image as if it was taken by a different camera using the inverse and forward ISP networks trained on the different camera. Fig. S2 qualitatively shows the camera-to-camera transfer results of ParamISP. The resulting camera-transferred images (2nd and 4th images) show color tone changes from the input images (1st and 3rd images) without any noticeable artifacts.

S3. Input Features for LocalNet and GlobalNet

It is reported that leveraging the gradient map and color histogram maps improves the ISP network performance by several previous works [6–8]. In our preliminary experiments, we also find a similar finding. Based on this, we design our LocalNet and GlobalNet to be fed additional input features as well as an input image. As additional input features, we include a gradient map, a soft histogram map, and an over-exposure mask to improve the ISP network perfor-

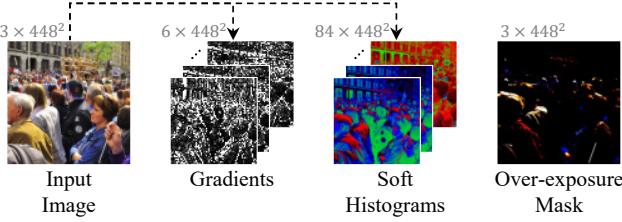


Figure S3. Input features of LocalNet and GlobalNet.

| Input Features | w/o A | w/o B | w/o C | Full |
|----------------|--------|--------|--------|---------------|
| PSNR↑ | 34.32 | 34.40 | 34.29 | 34.77 |
| SSIM↑ | 0.9693 | 0.9665 | 0.9677 | 0.9712 |

Table S2. Ablation study on the impact of each input feature. A , B , and C represent gradient map, soft histogram, and over-exposure mask, respectively.

mance (Fig. S3). For the gradient map, we apply the Sobel filter [5] on an input image and compute per-channel gradient maps in vertical and horizontal directions, resulting in a 6-channel gradient map. For the soft histogram map, we compute the soft histogram [8], for which we measure the relative distance between each channel value of a pixel and the center of histogram bins. In practice, we use 28 histogram bins, resulting in an 84-channel soft histogram map. We also include an over-exposure mask as a hint for restoring pixels of range-clipped values, where we compute the mask by setting its values as $10 \max(x - \tau, 0)$ where x is a pixel value of an input image, and τ is a threshold. We use 0.9 as the threshold in our implementation.

Ablation Study To validate the effects of the input features, we conduct an ablation study on RAW reconstruction using the D7000 images of the RAISE dataset [4]. Tab. S2 shows a quantitative ablation study on the input features of LocalNet and GlobalNet. To analyze only the effects of the input features, we use our model without ParamNet as the baseline (4th column). We prepare our baseline with its three model variants, where LocalNet and GlobalNet in each model variant do not use each one of the three input features.

It may be unnecessary to explicitly provide hand-crafted features such as image gradients and soft histograms, as a network can learn to extract such features from an input image. However, without these features, the network may not fully exploit its capacity to learn features more useful for local and global non-linear operations, resulting in low reconstruction quality (1st and 2nd columns). Furthermore, discarding an over-exposure mask may waste the network capacity in estimating and restoring pixels of range-clipped values, resulting in decreased reconstruction performance (3rd column). Additionally, in the first and second rows of Tab. S6, we describe the results of experiments conducted with various cameras and forward ISP networks to assess the impact of input features.

| Operation | A | B | $C \times 1$ | $C \times 2$ | $C \times 3$ | $C \times 4$ | $C \times 5$ |
|-----------|-------|-------|--------------|--------------|--------------|--------------|--------------|
| PSNR↑ | 32.70 | 32.77 | 32.87 | 32.96 | 33.27 | 33.66 | 33.50 |
| SSIM↑ | 0.962 | 0.961 | 0.961 | 0.962 | 0.963 | 0.965 | 0.967 |

Table S3. Ablation study on the effects of the global adjustment operation of GlobalNet. A and B represent the 3×6 polynomial mapping function of CIE XYZ Net [1] and the 3×10 quadratic transformation of DeepISP [11], respectively, while $C \times N$ denotes N pairs of gamma correction and quadratic transformation (Ours).

| Camera model | A7R3[9] | D7000[4] | D90[4] | D40[4] | S7[11] |
|--------------|---------|----------|--------|--------|--------|
| Training # | 7766 | 4600 | 1700 | 26 | 50 |
| Validation # | 200 | 200 | 100 | - | 20 |
| Testing # | 1000 | 1000 | 400 | 50 | 150 |

Table S4. Statistics of our dataset used for ParamISP.

S4. Global Adjustment Operation of GlobalNet

In this section, we verify the effect of the global adjustment operations of GlobalNet. To this end, we prepare variants of ParamISP with different global adjustment operations, and train them using the D7000 training images of the RAISE dataset [4]. Then, we evaluate their performances using the D7000 test set (Tab. S3). In this evaluation, we include the global adjustment operations of previous methods [1, 11] as well as different numbers of gamma correction and quadratic transformation operations. Tab. S3 shows that our global tone adjustment operation substantially improves the performance in PSNR and SSIM.

S5. Training Strategy

Datasets we use three datasets consisting of RAW and sRGB image pairs captured from multiple cameras: the RAISE dataset [4] from Nikon D7000, D90, and D40, the RealBlur dataset [9] from Sony A7R3; and the S7 ISP dataset [11] from Samsung Galaxy S7. The statistics of each dataset are shown in Tab. S4. We extract camera parameters from the EXIF metadata [12] included in JPEG images.

Pre-training with Images from Diverse Cameras To achieve high-quality reconstruction, we pre-train our models using datasets of multiple cameras as if they were captured by a single camera model. We then fine-tune the models for our target camera. Despite the differences across different camera models, we find that this two-stage training substantially improves the ISP performance as the ISP models can learn common knowledge on the ISP operations.

To validate our two-stage training approach, we present a quantitative ablation study result in Tab. S7. In the table, ‘Generic’, ‘Individual’, and ‘Generic+individual’ mean models trained using multiple camera datasets, models trained using only target camera datasets, and models trained using our two-stage training scheme, respectively. All the ‘Generic’, ‘Individual’ and ‘Generic+individual’

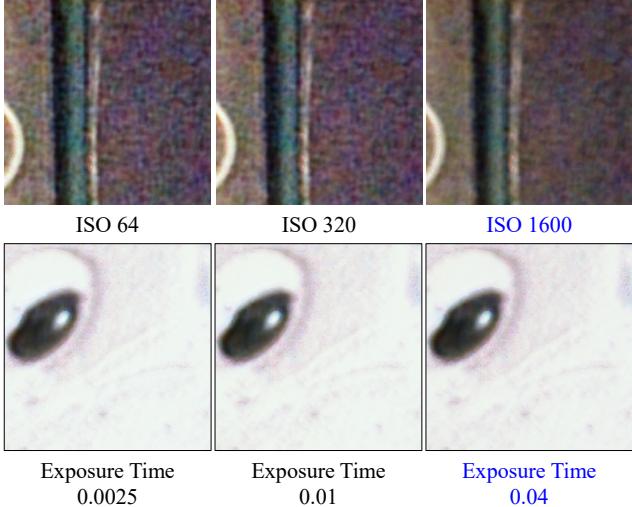


Figure S4. Example of sRGB reconstruction with modified optical parameters (sensitivity and exposure time) using the S7 ISP dataset [11]. The blue text represents the ground-truth value.

models are without ParamNet. We also include our final model ‘ParamISP’ trained using our two-stage training scheme in the table.

In the table, the ‘Individual’ models show higher RAW and sRGB reconstruction performance than the ‘Generic’ models on average as the ‘Generic’ models cannot properly learn the behaviors of specific camera models. The table also shows that the ‘Generic+individual’ models achieve higher performance than both ‘Generic’ and ‘individual’ models as they can exploit common knowledge on the camera ISP operations across various camera models, and at the same time, they can accurately learn the behaviors of specific camera models. Finally, our full models (ParamISP) outperform all the other models thanks to ParamNet.

Optimal Performance of Each Module While ParamISP features a modularized network architecture where each module has specific objectives, all the modules are jointly trained in an end-to-end fashion utilizing a reconstruction loss. Instead, we may explicitly train each module to serve its intended purpose. Here, we compare these two training approaches.

Given that ParamNet takes only optical parameters as input, and that GlobalNet performs only global adjustment operations, it is clear that ParamNet and GlobalNet are trained to serve their respective purposes. On the other hand, LocalNet may be trained to perform global operations as well as local ones. To explicitly train LocalNet and GlobalNet for their respective purposes, we may 1) train GlobalNet without LocalNet, and 2) fix GlobalNet and train LocalNet. We found that this sequential training results in a RAW reconstruction performance of 33.62dB, which is almost the same as that of the joint training (33.66dB). For clarity, ParamNet was excluded from this experiment. The

| Otical Params | A | B | C | D | GT |
|---------------|--------|--------|--------|---------------|---------------|
| PSNR↑ | 34.25 | 35.53 | 35.59 | 35.99 | 36.21 |
| SSIM↑ | 0.9636 | 0.9682 | 0.9703 | 0.9709 | 0.9724 |

Table S5. Reconstruction quality of ParamISP with respect to varying optical parameters. A , B , C , and D represent incorrect sensitivity, exposure time, aperture size, and focal length, respectively. GT represents ground-truth. ParamISP achieves the best reconstruction quality for ground-truth optical parameters, while the quality degrades for incorrect parameters, indicating that ParamISP can correctly reflect the adaptive behavior of real-world ISPs.

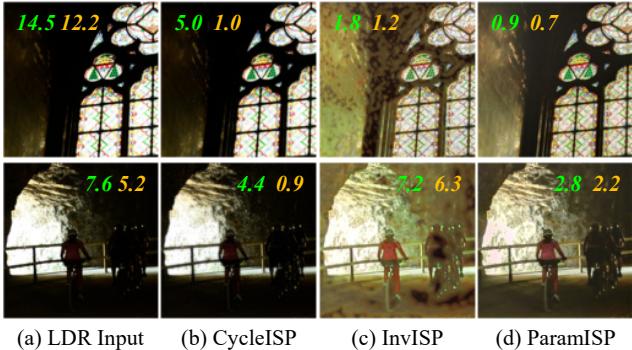
results suggest that although each module could be explicitly trained for its specific function, our joint training approach suffices to effectively train ParamISP.

S6. ParamNet & Optical Parameters

In Sec. 4.1 of the main paper, we analyze the impact of each optical parameter on performance in Tab. 2. To supplement the experimental results in the main paper, we present additional qualitative and quantitative analyses here.

We first verify whether ParamNet operates to reflect the characteristics of optical parameters. As shown in Fig. S4, we manipulate the values of optical parameters and observe changes in visual results. Note that increasing the exposure-related optical parameters of ParamNet, such as aperture size, is not equivalent to increasing the actual exposure. For instance, a manipulated high ISO value for the ISP does not mean that the final processed image should be brightened. Instead, it tells the ISP that its input RAW data is captured with a high ISO value, which results in more noise. The ISP can then adapt its operations, such as increasing the denoising strength. ParamISP operates in the same way as well. As the ISO value increases, the strength of denoising increases (1st row), and as the exposure time increases, saturated areas are restored more effectively (2nd row). This experimental result shows that ParamISP can precisely mimic real-world ISP operations that change according to the optical parameters.

We also present a quantitative analysis. In this analysis, we measure the reconstruction performance of ParamISP against various optical parameters. If ParamISP can accurately replicate the adaptive behavior of a real-world ISP, it is expected to attain optimal reconstruction quality with the ground-truth optical parameters, while the quality should diminish with incorrect parameters. To this end, we use the D7000 test images of the RAISE dataset [4] and their optical parameters. Specifically, for each optical parameter of an image, we randomly change its value and measure its reconstruction quality. Tab. S5 shows a result. In the table, significant performance drops are observed when incorrect values are used for optical parameters. This result indicates that ParamISP can correctly mimic the adaptive behavior of real-world ISPs.



(a) LDR Input (b) CycleISP (c) InvISP (d) ParamISP

Figure S5. HDR reconstruction results of over-exposed regions. Green: proportion of the pixels with values ≥ 250 . Yellow: proportion of the pixels with values $= 255$. The results show that our method produces a higher-quality HDR image with more details and also demonstrates a significant reduction in overly bright areas numerically.



(a) LDR Input (b) CycleISP (c) InvISP (d) ParamISP

Figure S6. Qualitative examples of HDR reconstruction.

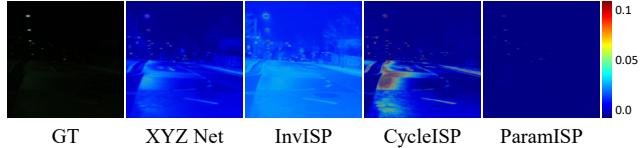


Figure S7. Qualitative results obtained by performing sRGB-to-Raw reconstruction using official pretrained models. We show error maps between the reconstructed and ground-truth (GT) RAW images. The GT RAW image in this figure is demosaicked for visualization.

S7. Additional Results

We provide additional detailed quantitative results to supplement the experimental results in the main paper: an ablation study on the effects of input features and the proposed ParamNet (Tab. S6), the training strategy (Tab. S7), and comparison on RAW & sRGB reconstruction (Tab. S8). The first two results supplement the experimental results in Sec. 4.1 of the main paper, while the third one supplements Sec. 4.2. Furthermore, we show additional qualitative results on HDR reconstruction (Fig. S5 & Fig. S6), sRGB-to-RAW reconstruction (Fig. S8), and RAW-to-sRGB reconstruction (Fig. S9). In all qualitative results, ParamISP shows visually better results compared to other methods, confirming its superior performance.

We also report a quantitative comparison (sRGB-to-Raw) using the official pretrained models provided by the authors on the Sony A7R3 dataset [9] (Ours: 48.33 (dB), CIE XYZ Net: 27.86, InvISP: 26.43, CycleISP: 25.20). It is worth noting that the official pretrained models of the previous methods were trained on different cameras than the target camera, which explains their lower performance. Fig. S7 shows qualitative results. Our model, trained and evaluated on the Sony A7R3 dataset, demonstrates minimal reconstruction errors. This indicates the importance of training neural ISP models on the target camera, as ISPs are dependent on the specific characteristics of cameras.

S8. Network Architecture

We visualize detailed network architectures for LocalNet (Fig. S10), GlobalNet (Fig. S11), and ParamNet (Fig. S12).

References

- [1] M. Afifi, A. Abdelhamed, A. Abuolaim, A. Punnappurath, and M. S. Brown. Cie xyz net: Unprocessing images for low-level computer vision tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(9):4688–4700, 2021. 3, 7
- [2] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7
- [3] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *Proceedings of the European conference on computer vision (ECCV)*, 2022. 1, 2
- [4] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM multimedia systems conference (MMSys)*, 2015. 3, 4, 7

- [5] N. Kanopoulos, N. Vasanthavada, and Robert L. Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of Solid-State Circuits (JSSC)*, 23(2):358–367, 1988. 3
- [6] S. Lin and L. Zhang. Determining the radiometric response function from a single grayscale image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 2
- [7] S. Lin, J. Gu, S. Yamazaki, and H. Shum. Radiometric calibration from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [8] Y.-L. Liu, W.-S. Lai, Y.-S. Chen, Y.-L. Kao, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang. Single-image hdr reconstruction by learning to reverse the camera pipeline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3
- [9] J. Rim, H. Lee, J. Won, and S. Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 5, 7
- [10] J. Rim, G. Kim, J. Kim, J. Lee, S. Lee, and S. Cho. Realistic blur synthesis for learning image deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2
- [11] E. Schwartz, R. Giryes, and A. M. Bronstein. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing (TIP)*, 28(2):912–923, 2018. 3, 4, 7
- [12] Tsurozoh Tachibanaya. Description of exif file format. <http://www.fifi.org/doc/jhead/exif-e.html>, 2001. 3
- [13] Y. Xing, Z. Qian, and Q. Chen. Invertible image signal processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 7
- [14] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 7

| Components | D7000 [4] | | D90 [4] | | D40 [4] | | S7 [11] | | A7R3 [9] | | Average | | |
|---------------|--------------------------------------|-------|---------|-------|---------|-------|---------|-------|----------|-------|---------|-------|--------|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | |
| sRGB → RAW | Ours (w/o Input Features & ParamNet) | 33.66 | 0.9646 | 34.50 | 0.9551 | 44.87 | 0.9879 | 34.51 | 0.9074 | 44.66 | 0.9892 | 38.44 | 0.9608 |
| | +Input Features | 34.77 | 0.9712 | 34.98 | 0.9672 | 44.95 | 0.9847 | 34.45 | 0.9063 | 46.72 | 0.9912 | 39.17 | 0.9641 |
| | +ParamNet w/o dropout | 35.64 | 0.9702 | 36.26 | 0.9724 | 45.41 | 0.9858 | 34.81 | 0.9007 | 47.47 | 0.9922 | 39.92 | 0.9643 |
| | +w/ dropout | 36.21 | 0.9724 | 36.31 | 0.9731 | 45.73 | 0.9883 | 35.14 | 0.9115 | 47.80 | 0.9922 | 40.24 | 0.9675 |
| RAW → sRGB | Ours (w/o Input Features & ParamNet) | 29.12 | 0.9399 | 29.46 | 0.9544 | 38.81 | 0.9843 | 28.48 | 0.7593 | 44.46 | 0.9806 | 34.07 | 0.9237 |
| | +Input Features | 29.21 | 0.9381 | 29.49 | 0.9558 | 39.08 | 0.9840 | 28.42 | 0.7635 | 44.39 | 0.9805 | 34.12 | 0.9244 |
| | +ParamNet w/o dropout | 29.87 | 0.9421 | 30.39 | 0.9639 | 39.11 | 0.9836 | 28.43 | 0.7668 | 44.63 | 0.9808 | 34.49 | 0.9274 |
| | +w/ dropout | 29.89 | 0.9422 | 30.50 | 0.9664 | 39.63 | 0.9849 | 28.60 | 0.7690 | 44.78 | 0.9815 | 34.68 | 0.9288 |

Table S6. Ablation study on the effects of input features and the proposed ParamNet. Each result was trained and evaluated using a specific camera only.

| Strategy | D7000 [4] | | D90 [4] | | D40 [4] | | S7 [11] | | A7R3 [9] | | Average | | |
|---------------|----------------------|-------|---------|-------|---------|-------|---------|-------|----------|-------|---------|-------|--------|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | |
| sRGB → RAW | Generic | 34.62 | 0.9689 | 35.61 | 0.9738 | 41.37 | 0.9796 | 34.65 | 0.9082 | 44.04 | 0.9871 | 38.06 | 0.9635 |
| | Individual | 34.77 | 0.9712 | 34.98 | 0.9672 | 44.95 | 0.9847 | 34.45 | 0.9063 | 46.72 | 0.9912 | 39.17 | 0.9641 |
| | Generic+individual | 36.47 | 0.9758 | 36.59 | 0.9796 | 45.75 | 0.9879 | 34.99 | 0.9118 | 47.18 | 0.9920 | 40.20 | 0.9694 |
| | ParamISP (Gen + Ind) | 38.49 | 0.9809 | 37.06 | 0.9810 | 45.97 | 0.9877 | 35.20 | 0.9125 | 48.33 | 0.9930 | 41.01 | 0.9710 |
| RAW → sRGB | Generic | 28.71 | 0.9262 | 28.44 | 0.9447 | 34.90 | 0.9685 | 28.06 | 0.7580 | 39.51 | 0.9603 | 31.92 | 0.9115 |
| | Individual | 29.21 | 0.9381 | 29.49 | 0.9558 | 39.08 | 0.9840 | 28.42 | 0.7635 | 44.39 | 0.9805 | 34.12 | 0.9244 |
| | Generic+individual | 31.51 | 0.9491 | 29.50 | 0.9535 | 38.97 | 0.9831 | 28.42 | 0.7716 | 44.95 | 0.9823 | 34.67 | 0.9279 |
| | ParamISP (Gen + Ind) | 34.14 | 0.9628 | 30.83 | 0.9670 | 39.54 | 0.9844 | 29.02 | 0.7868 | 45.51 | 0.9841 | 35.81 | 0.9370 |

Table S7. Ablation study on the effects of the training strategy. ‘Generic’, ‘Individual’, and ‘Generic+individual’ mean models trained using multiple camera datasets, models trained using only target camera datasets, and models trained using our two-stage training scheme, respectively. All the ‘Generic’, ‘Individual’ and ‘Generic+individual’ models are without ParamNet. We also include our final model ‘ParamISP’ trained using our two-stage training scheme in the table.

| Method | D7000 [4] | | D90 [4] | | D40 [4] | | S7 [11] | | A7R3 [9] | | Average | | |
|-----------------|-----------------|--------|---------|--------|---------|--------|---------|--------|----------|--------|---------|--------|--------|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | |
| sRGB → RAW | UPI [2] | 20.67 | 0.7854 | 26.57 | 0.8623 | 22.05 | 0.7679 | 29.98 | 0.8482 | 30.48 | 0.9368 | 25.95 | 0.8401 |
| | CIE XYZ Net [1] | 30.04 | 0.9461 | 32.62 | 0.9521 | 38.57 | 0.9809 | 33.24 | 0.8918 | 36.42 | 0.9779 | 34.18 | 0.9498 |
| | CycleISP [14] | 35.52 | 0.9740 | 35.85 | 0.9786 | 42.83 | 0.9831 | 34.55 | 0.9056 | 45.35 | 0.9916 | 38.82 | 0.9666 |
| | InvISP [13] | 33.48 | 0.9685 | 35.39 | 0.9747 | 45.08 | 0.9866 | 34.29 | 0.9095 | 47.14 | 0.9924 | 39.08 | 0.9663 |
| RAW → sRGB | ParamISP (Ours) | 38.49 | 0.9809 | 37.06 | 0.9810 | 45.97 | 0.9877 | 35.20 | 0.9125 | 48.33 | 0.9930 | 41.01 | 0.9710 |
| | UPI [2] | 18.81 | 0.6326 | 20.30 | 0.8010 | 16.01 | 0.7649 | 20.05 | 0.4205 | 19.37 | 0.5324 | 18.91 | 0.6303 |
| | CIE XYZ Net [1] | 26.76 | 0.8703 | 27.61 | 0.9183 | 34.84 | 0.9635 | 27.63 | 0.6978 | 37.19 | 0.9396 | 30.81 | 0.8779 |
| | InvISP [13] | 30.20 | 0.9393 | 28.89 | 0.9448 | 37.86 | 0.9816 | 28.96 | 0.7862 | 43.93 | 0.9786 | 33.97 | 0.9261 |
| ParamISP (Ours) | 34.14 | 0.9628 | 30.83 | 0.9670 | 39.54 | 0.9844 | 29.02 | 0.7868 | 45.51 | 0.9841 | 35.81 | 0.9370 | |

Table S8. Quantitative comparison on RAW & sRGB reconstruction. Note that CycleISP is not included in this comparison because it needs an input sRGB image for sRGB reconstruction.

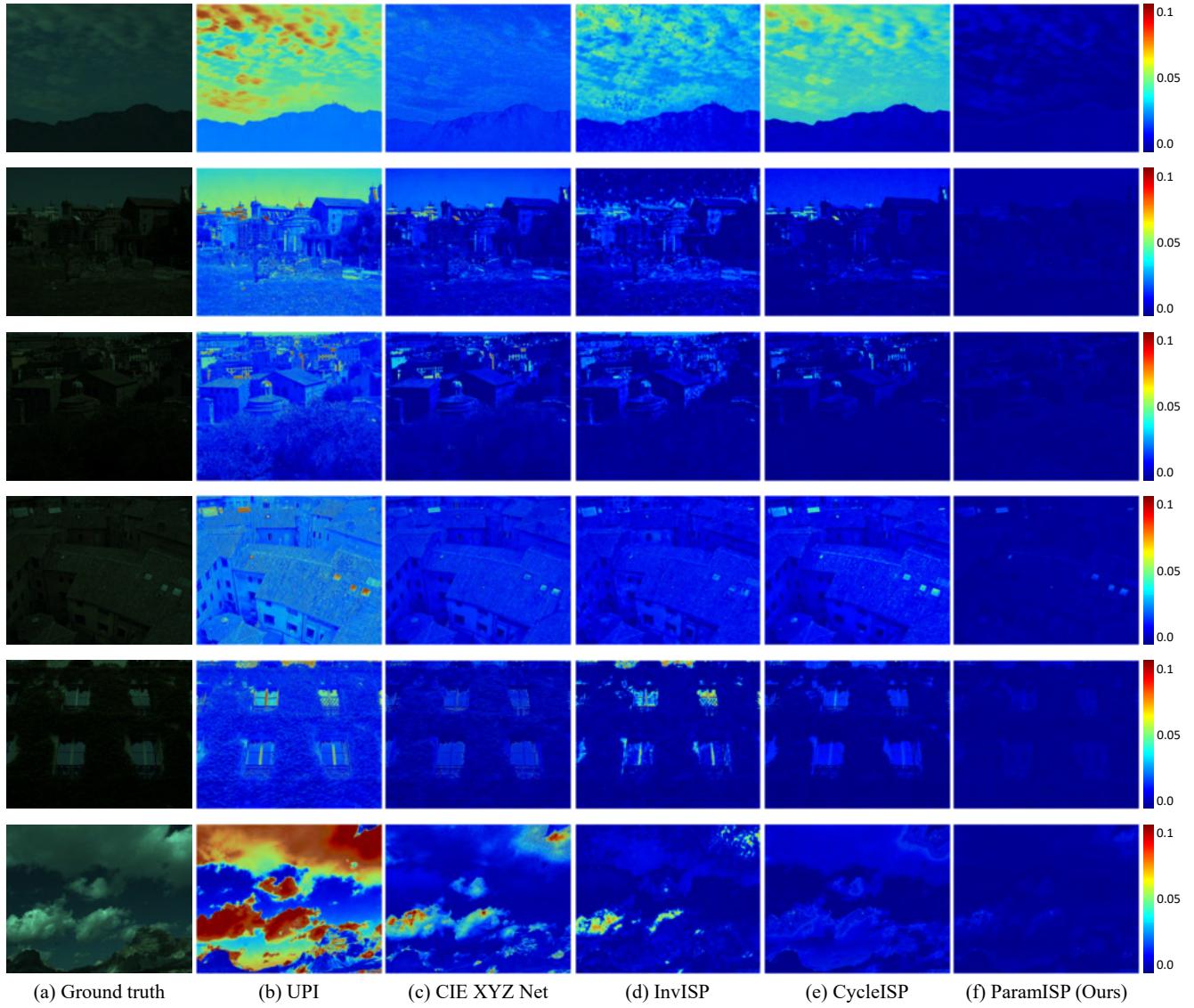


Figure S8. sRGB-to-RAW reconstruction. We show error maps between reconstructed and GT RAW images.

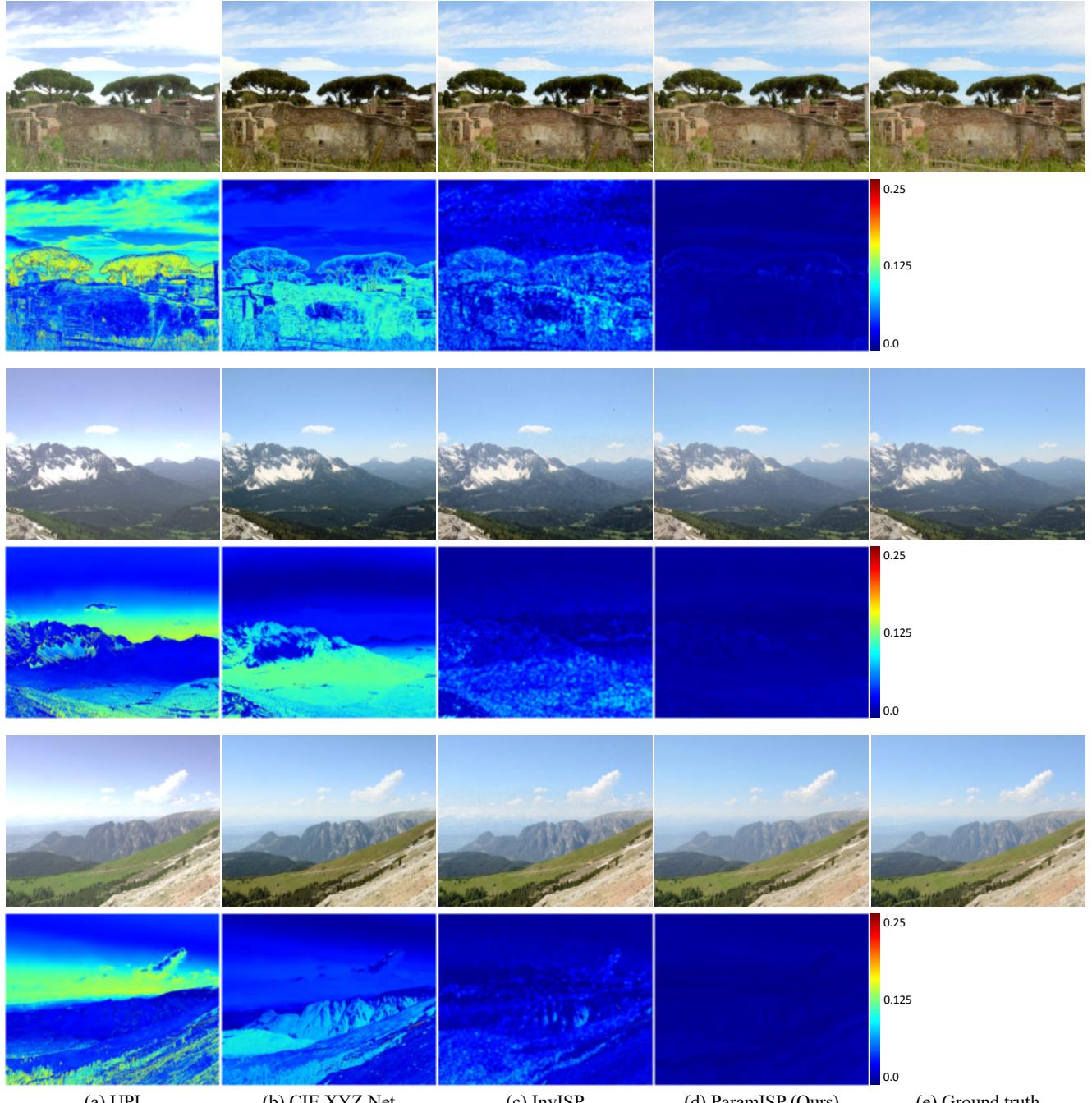


Figure S9. RAW-to-sRGB reconstruction. We show error maps between reconstructed and GT sRGB images.

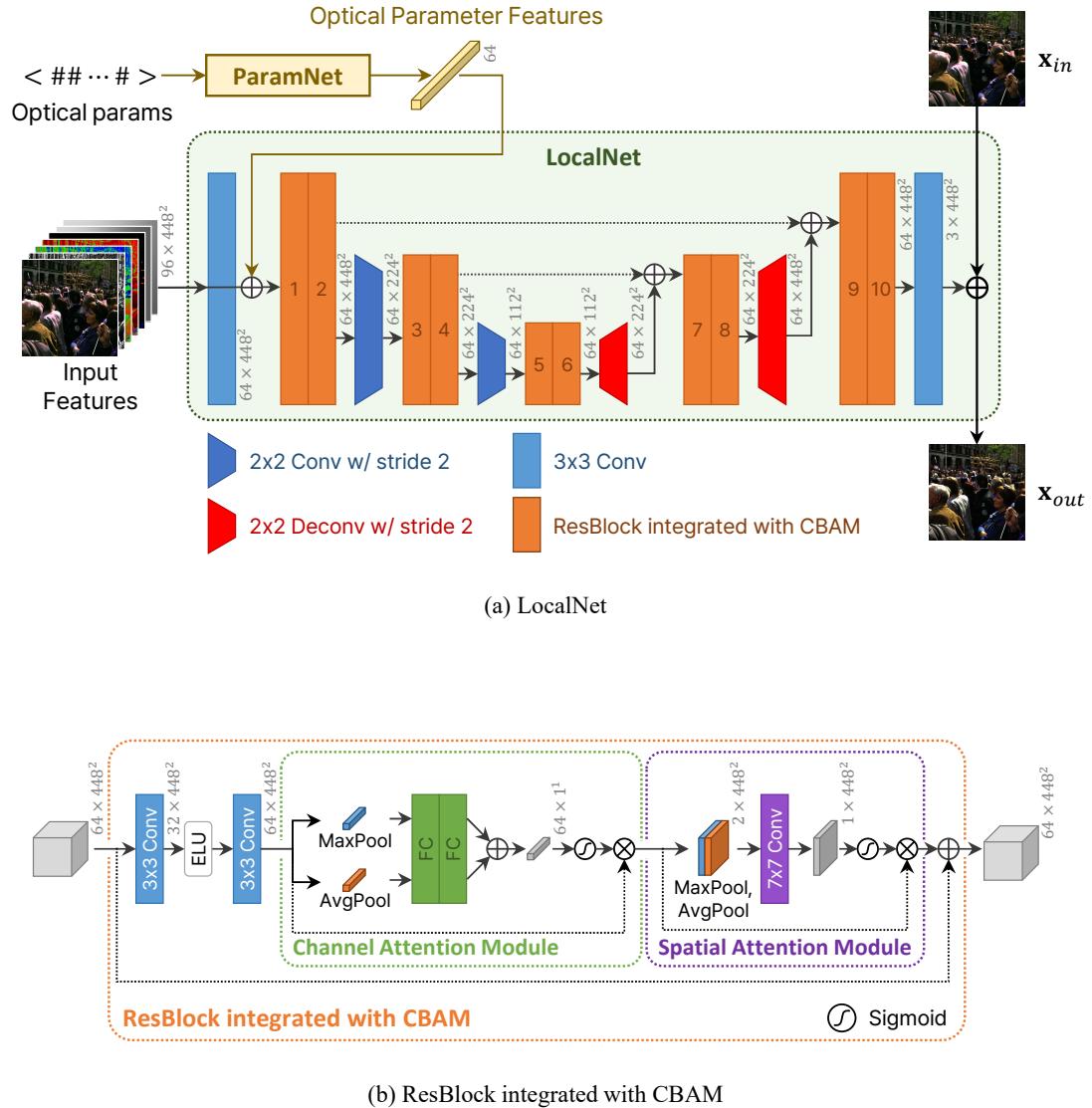


Figure S10. Detailed architecture of LocalNet.

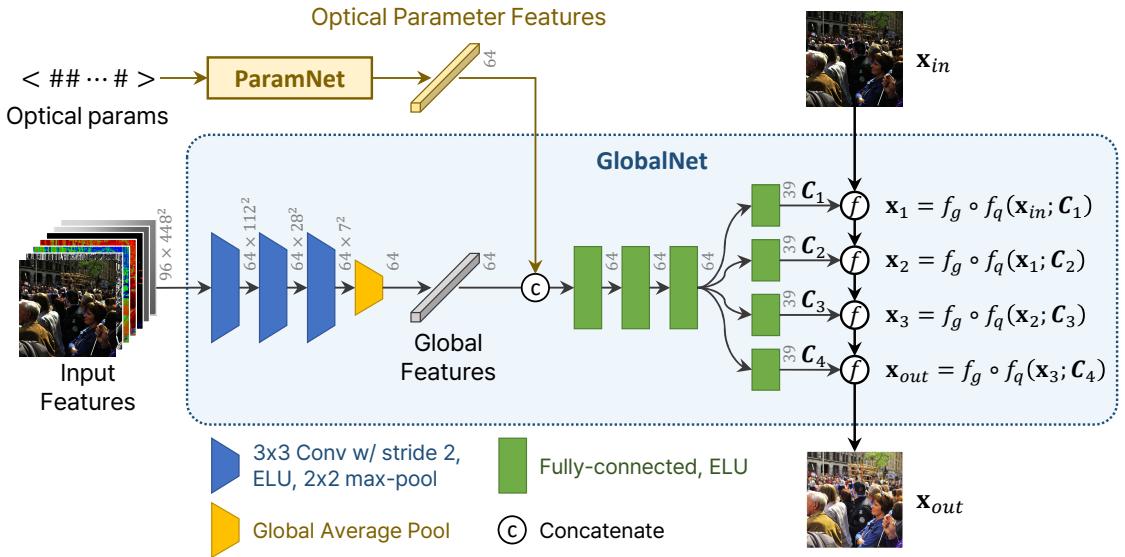


Figure S11. Detailed architecture of GlobalNet.

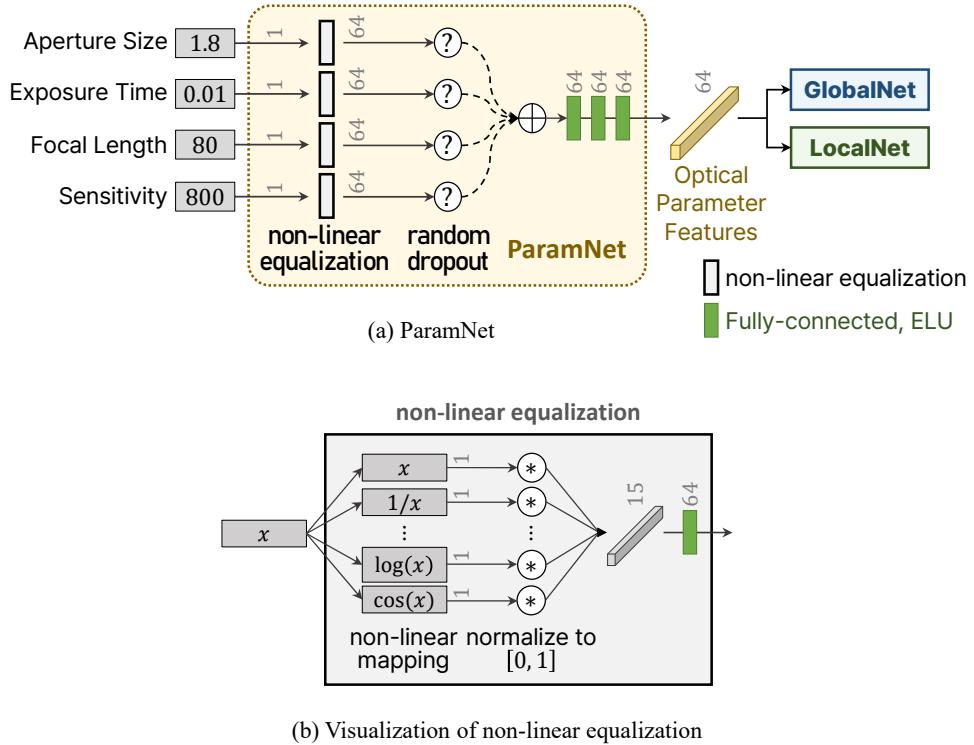


Figure S12. Detailed architecture of ParamNet.