Model-Based Image Signal Processors via Learnable Dictionaries

Marcos V. Conde, Steven McDonagh, Matteo Maggioni, Aleš Leonardis, Eduardo Pérez-Pellitero

Huawei Noah's Ark Lab

Abstract

Digital cameras transform sensor RAW readings into RGB images by means of their Image Signal Processor (ISP). Computational photography tasks such as image denoising and colour constancy are commonly performed in the RAW domain, in part due to the inherent hardware design, but also due to the appealing simplicity of noise statistics that result from the direct sensor readings. Despite this, the availability of RAW images is limited in comparison with the abundance and diversity of available RGB data. Recent approaches have attempted to bridge this gap by estimating the RGB to RAW mapping: handcrafted model-based methods that are interpretable and controllable usually require manual parameter fine-tuning, while end-to-end learnable neural networks require large amounts of training data, at times with complex training procedures, and generally lack interpretability and parametric control. Towards addressing these existing limitations, we present a novel hybrid model-based and data-driven ISP that builds on canonical ISP operations and is both learnable and interpretable. Our proposed invertible model, capable of bidirectional mapping between RAW and RGB domains, employs end-to-end learning of rich parameter representations, *i.e*. dictionaries, that are free from direct parametric supervision and additionally enable simple and plausible data augmentation. We evidence the value of our data generation process by extensive experiments under both RAW image reconstruction and RAW image denoising tasks, obtaining state-of-the-art performance in both. Additionally, we show that our ISP can learn meaningful mappings from few data samples, and that denoising models trained with our dictionary-based data augmentation are competitive despite having only few or zero ground-truth labels.

1 Introduction

Advances in Convolutional Neural Networks have made great strides in many computer vision applications in the last decade, in part thanks to the proliferation of camera devices and the resulting availability of large-scale image datasets. The majority of these datasets contain sRGB image data, which is obtained via an in-camera Image Signal Processor (ISP) that converts the camera sensor's RAW readings into perceptually pleasant RGB images, suitable for the human visual system. However, the characteristics of RAW imagery (*e.g*. linear relationship with scene irradiance, raw and untampered signal and noise samples) are often better suited for the ill-posed, inverse problems that commonly arise in low-level vision tasks such as denoising, demosaicing, HDR, super-resolution (Qian et al. 2019; Abdelhamed, Lin, and Brown 2018; Wronski et al. 2019; Gharbi et al. 2016; Liu et al. 2020). For tasks within the ISP, this does not come as a choice but rather a must, as the input domain is necessarily in the RAW domain due to the camera hardware design (Buckler, Jayasuriya, and Sampson 2017; Ignatov et al. 2021).

Unfortunately, RAW image datasets are not nearly as abundant and diverse as their RGB counterparts, and thus some of the performance potential of CNN-based approaches cannot be fully utilized. To bridge this gap, recent methods aim to estimate the mapping from sRGB. The recent work of Brooks et al. introduces a generic camera ISP model composed of five canonical steps, each of them approximated by an invertible, differentiable function. Their proposed ISP can be conveniently plugged-in to any RGB training pipeline to enable RAW image processing. As each of the functions is constrained to perform a single task within the ISP, intermediate representations are fully interpretable, allowing for complete flexibility and interpretability in the ISP layout. Despite successful application to image denoising, this approach requires manually setting the true internal camera parameters, and these cannot be learnt from data. Although some DSLR cameras do provide access to such parameters, ISP layouts and their related inner settings are generally protected and inaccessible to the end user.

Alternative recent learning-based approaches (Punnappurath and Brown 2020; Zamir et al. 2020; Xing, Qian, and Chen 2021) attempt to learn the ISP in an end-to-end manner, in order to circumvent the noted problems. Focused on the RAW data synthesis from sRGB images, CycleISP (Zamir et al. 2020) adopts *cycle consistency* to learn both the forward (RAW-to-RGB) and reverse (RGB-to-RAW) directions of the ISP using 2 different networks and is trainable end-to-end. The authors show that RGB data can then be leveraged successfully to aid a RAW denoising task. The ISP is thus learned as a black-box, is not modular and therefore lacks both interpretability for intermediate representations and control of the ISP layout. However, these traits can be considered important when training for specific intermediate tasks within the ISP (*e.g*. colour constancy). Additionally, as

there is no model regularization, training CycleISP remains a complex procedure, requiring large amounts of RAW data.

Contemporary to our presented work, the InvISP (Xing, Qian, and Chen 2021) proposes the camera ISP model as an invertible ISP using a single invertible neural network (Kingma and Dhariwal 2018; Ho et al. 2019) to perform both the RAW-to-RGB and RGB-to-RAW mapping. This normalizing-flow-based approach has the advantages of being invertible and learnable, however, as CycleISP lacks of interpretability and control, requires large amounts of training data and its constrained by the invertible blocks (*i.e*. input and output size must be identical).

In this paper we introduce a novel hybrid approach that tackles the aforementioned limitations of ISP modelling and retains the best of both model-based and end-to-end learnable approaches (Shlezinger et al. 2021). We propose a modular, parametric, model-driven approach with a novel parameter dictionary learning strategy that builds on Brooks et al. We further improve this flexible, interpretable and constrained ISP architecture with additional lens shading modelling and a more flexible parametric tone mapping. To address the lack of in-camera parameters, discussed previously, we design an end-to-end learnable dictionary representation of inner camera parameters. This provides a set of parameter basis for optimal end-to-end reconstruction, and enables unlimited data augmentation in the RAW image manifold. Our proposed method is modular, interpretable and is governed by well-understood camera parameters. It provides a framework to learn an end-to-end ISP and related parameters from data. Moreover, it can be learnt successfully from very few samples, even when corrupted by noise. Note that we focus on the RAW reconstruction task and its downstream applications (*e.g*. denoising, HDR imaging). The forward pass or RAW-to-RGB processing, despite related, is a different research problem (Ignatov, Van Gool, and Timofte 2020; Schwartz, Giryes, and Bronstein 2019; Liang et al. 2021).

Our main contributions can be summarized as follows: (1) a modular and differentiable ISP model, composed of canonical camera operations and governed by interpretable parameters, (2) a training mechanism that, in conjunction with our model contribution, is capable of end-to-end learning of rich parameter representations, *i.e*. dictionaries or basis and related linear decomposition decoders, that result in compact ISP models, free from direct parameter supervision, (3) extensive experimental investigation; our learned RGB-to-RAW mappings are used to enable data augmentation towards down-stream task performance improvement, in multiple data regimes of varying size and noise.

2 Image Signal Processor

The group of operations necessary to convert the camera sensor readings into natural-looking RGB images are generally referred to as the Image Signal Processor (ISP). There is great variability in ISP designs with varying levels of complexity and functionalities, however a majority of them contain at least a number of operations that are generally considered to be a canonical representation of a basic ISP, namely white balance, color correction, gamma expansion and tone mapping (Brown 2016; Heide et al. 2014; Delbracio et al. 2021). Brooks et al. introduces a modular, differentiable ISP model where each module is an invertible function that approximates the aforementioned canonical operations. In this section we review that work, and introduce notation and parameter details about each operation as well as the complete function composition.

Let us initially define two images spaces: the RAW image domain $\mathcal{Y}$ and the sRGB image domain $\mathcal{X}$. The transformation done by the camera ISP can thus be defined as $f : \mathcal{Y} \rightarrow \mathcal{X}$. Intuitively, we can define a modular ISP function f as a composite function as follows:

$$x = (f_n \circ \cdots \circ f_2 \circ f_1)(y, p_n, \ldots, p_2, p_1), \tag{1}$$

where f_i is a function with related parameters p_i for a composition of arbitrary length n. In order to recover a RAW image y from the respective sRGB observation x (*i.e*. a mapping from $\mathcal{X} \rightarrow \mathcal{Y}$) we can choose f_i to be invertible and tractable bijective functions:

$$y = (f_1^{-1} \circ f_2^{-1} \circ \cdots \circ f_n^{-1})(x, p_1, p_2, \ldots, p_n). \tag{2}$$

2.1 Colour Filter Array Mosaic

Camera sensors use a Colour Filter Array (CFA) in order to capture wavelength-specific colour information. The sensor is covered with a pattern of colour filters arranged in a given spatial pattern, *e.g*. the well known Bayer pattern, which is a 2×2 distribution of R - G - G - B colours, that effectively produces a single colour measurement per spatial position.

In order to obtain the missing colour samples for each spatial position, the so called demosaicing methods aim to recover the missing pixels, commonly an ill-posed problem which, for the sake of simplicity, we will address as a simple bilinear interpolation: $f_6(y) = \text{bic}(y)$. The inverse of this function, is however, a straightforward mosaicing operation. It can be defined as:

$$f_6^{-1}(x_5, k_m) = x_5 * k_m, \tag{3}$$

where $*$ denotes a convolution with kernel k_m containing the mosaicing pattern, generally strictly formed by $\{0, 1\}$.

2.2 Lens Shading Effect

Lens Shading Effect (LSE) is the phenomenon of the reduced amount of light captured by the photoreceptor when moving from the center of the optical axis towards the borders of the sensor, mostly caused by obstruction of elements involved in the lens assembly. We can define this function as:

$$f_5(x_5, M) = x_5 \odot M, \tag{4}$$

where M is a mask containing per-pixel lens-shading gains. This can be inverted by inverting each per-pixel gain.

2.3 White Balance and Digital Gain

The White Balance (WB) stage aims to neutralize the scene light source colour such that after correction its appearance matches that of an achromatic light source. In practice, this is achieved by a global per-channel gain for two of the colours, *i.e*. red and blue gains, namely g_r and g_b respectively, which we arrange in a three colour vector $g_{\text{wb}} = [\ g_r \quad 1 \quad g_b\ ]$.

This scene illuminant is generally estimated heuristically, although more sophisticated approaches have also been explored (Gijsenij, Gevers, and Lucassen 2009; Barron and Tsai 2017; Hernandez-Juarez et al. 2020).

WB is normally applied in conjunction with a scalar digital gain g_d, which is applied globally to all three channels, and scales the image intensities as desired. This process can be conveniently described as a convolution:

$$f_4(x_4, g_{wb}, g) = x_4 * (g_d\ g_{\text{wb}}). \tag{5}$$

To obtain the inverse function f_4^{-1}, we just invert each of the gains individually, but instead of using the naive division $1/g$, we follow the highlight-preserving cubic transformation of Brooks et al.

2.4 Color Correction

The ISP converts the sensor color space into the output color space. This step is often necessary as the CFA colour spectral sensitivity does not necessarily match the output color standard, *e.g*. sRGB (Brown 2016; Afifi et al. 2021). A global change in the color space can be achieved with a 3×3 Color Correction Matrix (CCM):

$$f_3(x_3, \mathbf{C_m}) = \mathbf{X}_3\mathbf{C}_m, \tag{6}$$

where $\mathbf{C}_m$ denotes a CCM parameter and $\mathbf{X}_3$ denotes x_3 reshaped for convenience as a matrix, *i.e*. $\mathbf{X}_3 \in \mathbb{R}^{hw\times 3}$. Similarly to f_3, we can obtain f_3^{-1} by using $\mathbf{C}_m$ pseudo-inverse.

2.5 Gamma Correction

The camera sensor readings are linearly proportional to the light received, however the human visual system does not naturally perceive light linearly, but rather is more sensitive to darker regions. Thus it is common practice to adapt the linear sensor readings with a gamma logarithmic function:

$$f_2(x_2, \gamma) = \max(x_2, \epsilon)^{1/\gamma}, \tag{7}$$

where γ is a parameter regulating the amount of compression/expansion, generally with values around $\gamma = 2.2$. The inverse function can be defined as follows:

$$f_2^{-1}(x_1, \gamma) = \max(x_1, \epsilon)^{\gamma}. \tag{8}$$

2.6 Tone Mapping

Tone Mapping Operators (TMOs) have been generally used to adapt images to their final display device, the most common case being the TMO applied to High Dynamic Range Images (HDRI) on typical Low Dynamic Range display devices. As opposed to using an S-shaped polynomial function as proposed by (Reinhard et al. 2002; Brooks et al. 2019), we can use instead a parametric piece-wise linear function that we model as a shallow convolutional neural network (Punnappurath and Brown 2020) composed only by 1×1 kernels and ReLU activations:

$$f_1(x_1, \theta_f) = \phi_t(x_1, \theta_f), \tag{9}$$

where ϕ is a shallow CNN with learnable parameters θ_f for the forward pass. A different set of weights θ_r can be optimized for the reverse pass.

3 Learning Parameter Representations

In the previous section we introduced a modular and parametric ISP model, however that model alone does not allow for end-to-end training. In this section we introduce a strategy to enable end-to-end training for the presented ISP model. To the best of our knowledge, our method is the first data-driven, model-based approach to tackle the reverse ISP problem.

In Figure 1 we show an overview of our proposed approach, formed by a 6-stage ISP model (in blue colour) and separate networks that learn parameter dictionaries and feed parameters to the model (in green colour).

3.1 Parameter Dictionaries

Color Correction Modern smartphone cameras typically use different CCMs depending on specific light conditions and capture modes, so any method that assumes a single CCM mode might struggle to cope with colour variability. Additionally, an ISP model might be trained to reconstruct RAW images captured with different cameras and thus also different ISP and CCMs. As previously discussed, these matrices are generally not accessible to the end user. In order to learn the color space transformation done by the ISP, we create a dictionary $D_{ccm} \in \mathbb{R}^{N\times 3\times 3}$ of size N, where each atom is a CCM. To preserve the significance and physical meaning of these matrices, and avoid learning non-realistic parameters, we constrain the learnt atoms in the dictionary by column-normalizing each matrix following the ℓ_1 norm, as this is one of the most representative properties of realistic CCMs (Brooks et al. 2019; Koskinen, Yang, and Kämäräinen 2019). We perform the color correction as a convolution operation, where the convolutional kernels are the atoms of D_{ccm} and the input is the intermediate representation from the previous function in the ISP model. As the result of this operation we obtain $I_{ccm} \in \mathbb{R}^{N\times H\times W\times 3}$, which represents N RGB images, each one the result of applying each atom to the input image. This representation I_{ccm} passes through a CNN encoder E_{ccm} that produces a vector of weights $\mathbf{w}_{ccm} \in \mathbb{R}^N$. The resultant color transformed sRGB image is obtained as a linear combination of I_{ccm} and $\mathbf{w}_{ccm}$, which is equivalent to linearly combining the atoms in the dictionary, and applying the resultant CCM to the image. As illustrated in Figure 2, the model simultaneously learns D_{ccm} and E_{ccm}. This novel dictionary representation of the camera parameters can allow learning the CCMs of various cameras at once. Note that the encoder E^r_{ccm} used during the reverse pass is different from the E^f_{ccm} used in the forward pass as we show in Figure 1, however, both encoders have the same functionality.

Digital Gain and White Balance Similarly to the CCM dictionaries, we define $D_{wb} \in \mathbb{R}^{N\times 3}$ as a dictionary of N white balance and digital gains, thus, each atom is a triplet of scalars $(g_d\ g_r\ g_b)$. We apply each atom g from the dictionary as described by Brooks et al. and obtain $I_{wb} \in \mathbb{R}^{N\times H\times W\times 3}$, which represents a linear decomposition of the results from applying each g_i to the input image. An encoder E_{wb} produces a set of weights $\mathbf{w}_{wb} \in \mathbb{R}^N$ from such representation. Note that this encoder is different from the E_{ccm} used in the color correction step. The encoder and dictionary are learned jointly in the optimization. The linear combination

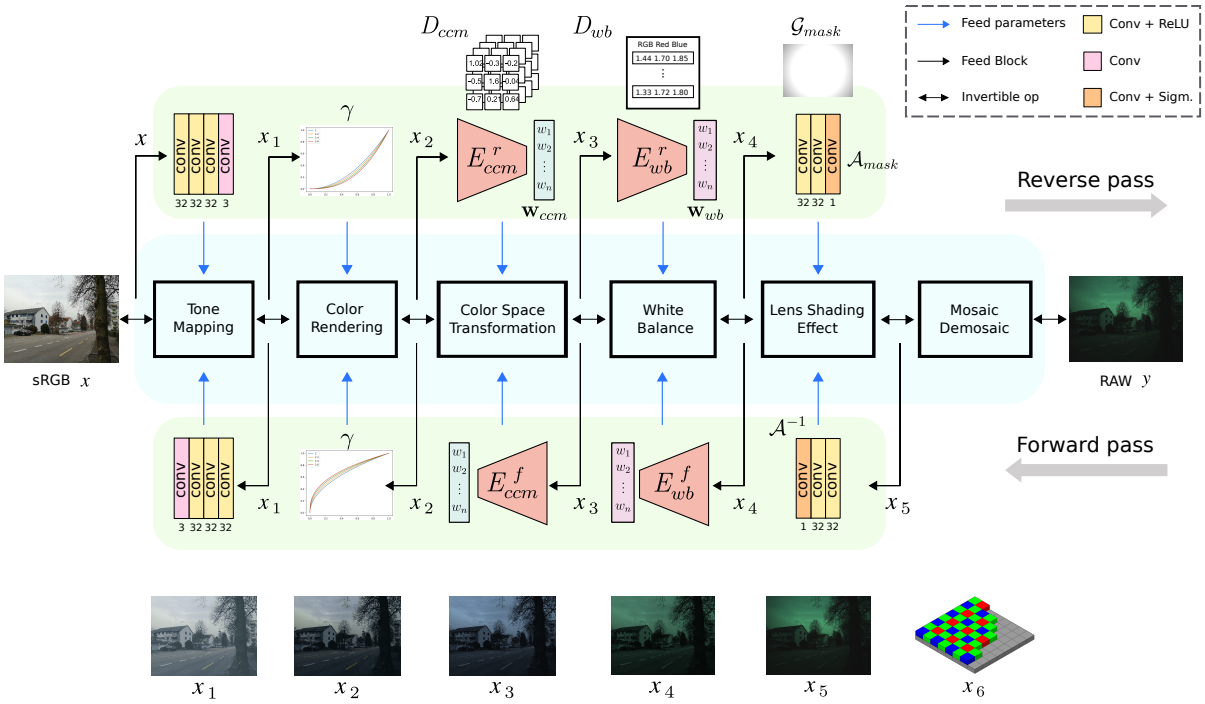

Figure 1: A visualization of our proposed model using as backbone (blue) the classical ISP operations described in Section 2, and additional learning component (green) described in Section 3. For visualization purposes, RAW images are visualized through bilinear demosaicing. This figure is best viewed in the electronic version.

of I_{wb} and $\mathbf{w}_{wb}$ produce our white balanced image. To ensure we keep the physical meaning and interpretability of the learnt WB gains, we found sufficient to initialize the atoms in D_{wb} using a uniform distribution $\mathcal{U}(1, 2)$ that encourages appropriate behaviour on the learnt gain vectors: non-negative and a reasonable range for pixel gains (i.e. approximately $[1, 2]$) (Brooks et al. 2019). As we show in Figure 1, the reverse pass encoder E^r_{wb} is different from the forward pass encoder E^f_{wb}, however, both work in the same way.

Dictionary Augmentations Two learnt dictionaries, *i.e.* D_{ccm} and D_{wb}, can be interpreted as a set of basis describing the parameter space. For a given RGB input, encoders find the combination of atoms in the dictionary that optimize the RAW image reconstruction, represented as a vector of weights $\mathbf{w}$. We can further exploit this representation to generate unlimited RAW images by adding small perturbations to the optimal weights $\mathbf{w}_{ccm}$ and $\mathbf{w}_{wb}$, by *e.g.* adding Gaussian noise. These dictionaries represent a convex hull of plausible parameters, and thus any variation within that space is likely to result in useful data for downstream tasks. Figure 2 shows this process. Once the dictionaries are learnt, it is possible to remove the related encoders (0.5M parameters) and sample convex combinations of the elements in the dictionary, reducing considerable the complexity of our model, and allowing to process high resolution images with low-latency.

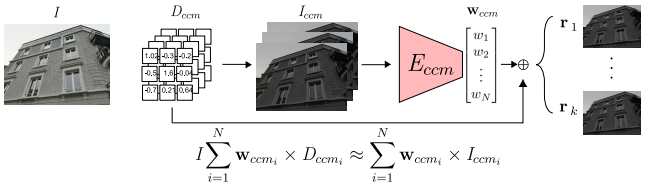

Figure 2: Dictionary-based Color Space Transformation. During training, D_{ccm} and E_{ccm} are learned together and generate a single output. At inference time, dictionary augmentations are used to generate k samples from a single RGB input. The $\mathbf{r}_k$ represent the random perturbations added to $\mathbf{w}_{ccm}$.

3.2 Piecewise Linear Tone Mapping

Tone mapping (Mantiuk et al. 2009) is a technique used by the camera to map one set of colors to adjust the image's aesthetic appeal by compressing the high-intensity and low-intensity values more than mid-intensity values. A tone map is usually designed as a 1D lookup table (LUT) that is applied per color channel to adjust the tonal values of the image, or as a "smoothstep" S-curve. To reconstruct RAW data tones from sRGB is challenging, especially at the over-exposed regions and high-dynamic range images require more sophisticated tone mapping. We propose a piecewise linear CNN as a learnable tone map (Punnappurath and Brown 2020). In the forward pass, tone mapping is performed using f_1. At the reverse pass, we perform the inverse tone mapping using f^{-1}. Both functions are shallow CNNs implemented using pixel-wise convolutional blocks to constraint the possible transformations and easily control the network, and representing by its definition piecewise linear models.

3.3 Lens Shading Modelling

Due to sensor optics, the amount of light hitting the sensor falls off radially towards the edges and produces a vignetting effect; known as lens shading. A typically early ISP stage constitutes Lens Shading Correction (LSC) (Young 2000) and is used to correct the effects of uneven light hitting the sensor, towards providing a uniform light response. This is done by applying a mask, typically pre-calibrated by the manufacturer, to correct for non-uniform light fallout effects (Delbracio et al. 2021). Modelling of the ISP therefore requires a method to add or correct the Lens Shading Effect (LSE) by modelling such a mask. We propose to model this mask as a pixel-wise gain map:

1. Gaussian mask $\mathcal{G}_{mask}(x, y) \sim \mathcal{N}_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ fitted from filtered sensor readings, assigns more or less intensity depending on the pixel position (x, y). Its two parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are further optimized together with the end-to-end ISP model.
2. Attention-guided mask $\mathcal{A}_{mask}$ using a CNN attention block, as was illustrated in Figure 1. These shallow blocks have constrained capacity to ensure the Lens Shading block only corrects per-pixel gains, and thus, we maintain the interpretability of the entire pipeline.

Both masks are in the space $\mathbb{R}^{H \times W}$. During the reverse pass, we apply both masks to the image using an element-wise multiplication (per-pixel gain), recreating the sensor's lens shading effect. To reverse this transformation or correct the LSE, we apply the LSC mask: (i) the inverse of $\mathcal{G}_{mask}$ (element-wise divide) and (ii) $\mathcal{A}^{-1}_{mask}$ estimated by the attention block in the forward pass.

3.4 Training

The complete pipeline is end-to-end trainable and we can use a simple ℓ_2 distance between the training RAW image y and the estimated RAW image $\hat{y}$. To ensure the complete pipeline is invertible, we add ℓ_2 loss terms for each intermediate image and also a consistency loss in the decomposition vectors $\mathbf{w}$ of the forward and reverse encoders. For more details we refer the reader to the supplementary material, where we also provide other relevant information about the training process *e.g*. GPU devices, batch sizes, network architectures.

4 Experimental Results

Throughout this section, we provide evidence that our method can effectively learn the RGB to RAW mapping of real unknown camera ISPs, obtaining state-of-the-art RAW reconstruction performance, and also validating the robustness of our model to operate under noisy data and data frugality (*i.e*. few-shot learning set-ups). Additionally, we conduct experiments on a downstream task, *i.e*. RAW image denoising, in order to validate ISP modelling beyond RAW image reconstruction, and the effectiveness of our proposed data augmentations. In all our experiments, we use the reverse pass of our model (Figure 1). During the denoising experiments, we use our ISP model as an on-line domain adaptation from RGB to RAW, guided by the proposed dictionary augmentations (see Section 3.1). We use PSNR as a metric for quantitative evaluation defining 2 variants: PSNR$_r$ for RAW reconstruction and PSNR$_d$ for denoising.

Table 1: Quantitative RAW reconstruction results on SIDD. The reconstruction PSNR$_r$ (dB) and top/worst 25% are shown for each baseline method. We also show quantitative RAW denoising results in terms of PSNR$_d$ to measure the impact of the synthetic data. Additionally we include the number of parameters of each ISP model (in millions).

Method	PSNR$_r$	Worst 25%	Best 25%	PSNR$_d$	# Params (M)
UPI	36.84	14.87	<u>57.10</u>	49.30	0.00
CycleISP	37.62	15.90	51.65	<u>49.77</u>	3.14
UNet	<u>39.84</u>	<u>20.27</u>	49.61	49.69	11.77
Ours	**45.21**	**21.58**	**66.33**	**50.02**	0.59

4.1 Datasets

SIDD (Abdelhamed, Lin, and Brown 2018; Abdelhamed, Timofte, and Brown 2019). Due to the small aperture and sensor, high-resolution smartphone images have notably more noise than those from DSLRs. This dataset provides real noisy images with their ground-truth, in both raw sensor data (raw-RGB) and sRGB color spaces. The images are captured using five different smartphone cameras under different lighting conditions, poses and ISO levels. There are 320 ultra-high-resolution image pairs available for training (*e.g*. 5328×3000). Validation set consist of 1280 image pairs.
MIT-Adobe FiveK dataset (Bychkovsky et al. 2011). We use the train-test sets proposed by InvISP (Xing, Qian, and Chen 2021) for the Canon EOS 5D and the Nikon D700, and the same processing using the LibRaw library to render ground-truth sRGB images from the RAW images.

4.2 RAW Image Reconstruction

We compare our RAW image reconstruction against other state-of-the-art methods, namely: **UPI** (Brooks et al. 2019) a modular, invertible and differentiable ISP model. Requires parameter tuning to fit the distribution of the SIDD dataset. **CycleISP** (Zamir et al. 2020) a data-driven approach for modelling camera ISP pipelines in forward and reverse directions. For generating synthetic RAW images, we use their publicly available pre-trained model, which has been fine-tuned using the SIDD dataset. **U-Net** (Ronneberger, Fischer, and Brox 2015) a popular architecture that has been previously utilized to learn ISP models (Ignatov, Van Gool, and Timofte 2020) as a *naive* baseline trained end-to-end without any other model assumptions or regularization.

In Table 1 we show reconstruction results in terms of PSNR$_r$ on the SIDD validation. Our model performs better than CycleISP despite being $\sim 5\times$ smaller, achieving $+7.6$dB improvement, and better than U-Net despite being $\sim 20\times$ smaller. We also perform better than hand-crafted methods as UPI by $+8.37$, which proves our capacity for learning camera parameters. In Figure 4 we show a qualitative comparison of RAW reconstruction methods. Additionally, we aim to prove that our pipeline is invertible, by doing the *cycle mapping* (sRGB to RAW and back to sRGB) our model

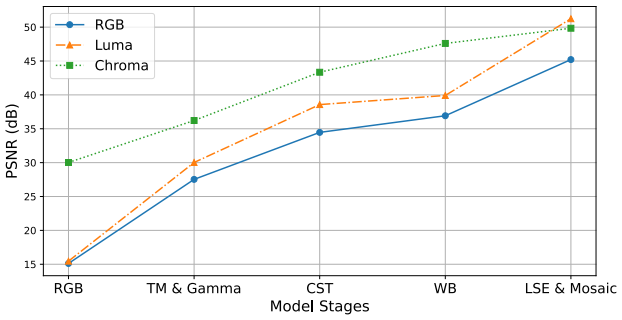

Figure 3: Ablation study for the proposed techniques. We compare, in terms of PSNR, the intermediate steps x_i with the original RAW image at both RGB and YUV (luminance Y, chrominance UV) colour domains. Results show a monotonic PSNR evolution at RGB domain, meaning that after each transformation the RGB moves "closer" to the RAW image.

achieves 37.82dB PSNR. More details about this experiment and qualitative results are accessible in the supplement.

Moreover, we measure the impact of the synthetic data by performing a simple denoising experiment. For a fair comparison, we use the same denoiser (U-Net) and training method as in Brooks et al. Under this setup, the only difference is the conversion from sRGB to RAW from the compared methods. We use the MIR Flickr dataset (Huiskes, Thomee, and Lew 2010) as a source of sRGBs, each model transforms them into synthetic RAWs that are used for training the denoiser. These are evaluated on the SIDD Dataset. Table 1 shows the effect of our synthetic data on the denoising task, the network trained with our data achieves an improvement of $+0.7$dB PSNR_d with respect to the baseline method.

Figure 3 shows the ablation of the intermediate performance of our method using the SIDD. The monotonic PSNR evolution at the RGB domain indicates that each component in our model is contributing to improve the final reconstruction. This ablation study, together with the Table 1 quantitative results provide strong empirical evidence supporting that our pipeline and learnt parameters are realistic. The color correction and white balance (colour constancy) perform the most significant transformation in the color space, as shown by the ℓ_2 distance reduction on the Chrominance space (the PSNR of UV components increases from 36.21dB to 43.34dB after applying our learnt CCMs, and to 47.59dB after applying our learnt WB gains). The LSE improves the Luminance space reconstruction from 39.91dB to 51.24dB.

We also test our approach using two DSLR cameras, the Canon EOS 5D and NikonD700. Using the train-test sets, loss, and patch-wise inference proposed by InvISP (Xing, Qian, and Chen 2021), our method also achieves SOTA results at RAW reconstruction as we show in Table 2. Note that InvISP is evaluated on the RAW with post-processed white balance provided by camera metadata, however, we do not use any information about the camera. As we have shown using the SIDD Dataset, our model is device-agnostic.

More details about the learnt parameters' distribution and qualitative comparisons are accessible in the supplement.

Table 2: Quantitative RAW Reconstruction evaluation among our model and baselines proposed by Invertible-ISP (Xing, Qian, and Chen 2021) using two DSLR cameras.

Method	Nikon PSNR_r	Canon PSNR_r
UPI (Brooks et al. 2019)	29.30	-
CycleISP (Zamir et al. 2020)	29.40	31.71
InvGrayscale (Xia, Liu, and Wong 2018)	33.34	34.21
UNet	38.24	41.52
Invertible-ISP (w/o JPEG)	43.29	45.72
Invertible-ISP (with JPEG Fourier)	**44.42**	<u>46.78</u>
Ours	<u>43.62</u>	**50.08**

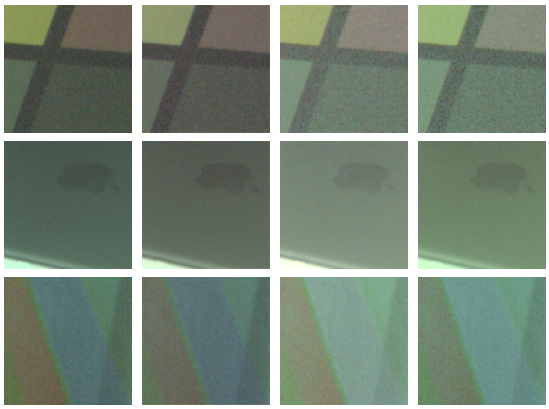

Figure 4: Qualitative RAW Reconstruction comparison using SIDD. Our model reconstructs better colours, tones and brightness of RAW images from different cameras.

4.3 Few-shot and Unsupervised Denoising

We aim to prove further the benefits of our approach and applications on downstream low-level vision tasks. In the next experiments, we use DHDN (Park, Yu, and Jeong 2019) as our denoiser. We sample shot/read noise factors as Zamir et al., as such, we can inject realistic noise into the images. The mean PSNR of the noisy-clean pairs we use for training is 41.18 dB. Our ISP model is always validated on the SIDD validation data using PSNR_r, our denoising experiments are validated in the same way and are reported using PSNR_d. We run two different experiments:

Few-shot experiments: In these experiments, we start with all available 320 sRGB-RAW *clean-noisy* pairs for training our ISP model as explained in Section 3. In Table 3 we denote the baseline method without augmentations as "DHDN", and the method with our ISP as on-line augmentations as "Ours". We can appreciate the benefit of using our approach to generate extra synthetic data, +0.46 dB improvement, and overall SOTA results. We explore how decreasing the amount of clean data available for training affects RAW reconstruction performance, and thus, RAW denoising. We experiment on three few data regimes: 184, 56, and only 5 pairs available for training. Table 3 shows that our denoiser trained with few-data (Ours-f) but using our augmentations, can achieve similar performance to the baseline trained with all the data.

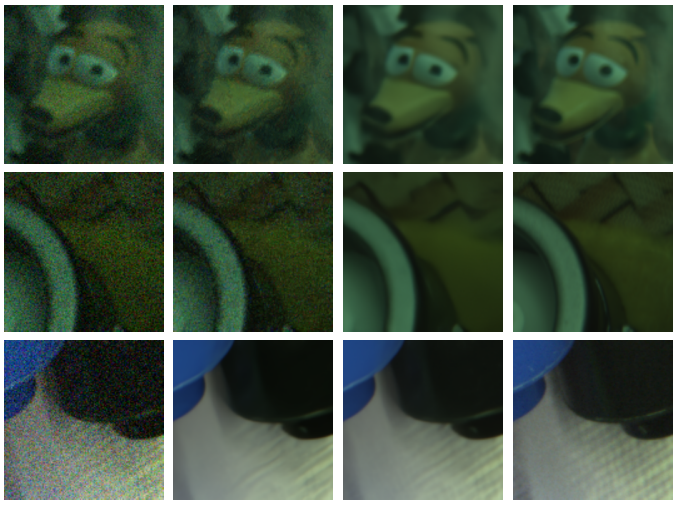

Figure 5: Qualitative RAW Denoising samples. Our model removes noise while keeping textures and details. More comparisons can be found in the supplementary material.

Unsupervised experiments: For the last two few-shot regimes (56 and 5 pairs), we do an additional experiment where clean ground-truth denoising labels are not available. In these cases, we only use sRGB-RAW noisy pairs for training the ISP and the denoiser networks. We convert sRGBs into noisy RAWs using our augmentation strategy, and we add extra noise to our already noisy signals in order to have pseudo *ground-truth* pairs (Imamura, Itasaka, and Okuda 2019). Our model learns how to reconstruct RAW data even if trained with few noisy data. This ablation study and denoising results for the few-shot and unsupervised scenarios are available in the supplement. As we show in Table 3, our denoiser (Ours-u) never saw ground-truth clean images, yet when trained using our method achieved better results than models trained on real data such as DnCNN (+6.6 dB).

SIDD Dataset: We report our denoising results in Table 3, and compare with existing state-of-the-art methods on the RAW data. We follow a standard self-ensemble technique: four flipped and rotated versions of the same image are averaged. We use CycleISP Denoiser (Zamir et al. 2020) publicly available weights trained on 1 million images. As shown in Table 3, our models trained on our synthetic data perform better than previous state-of-the-art despite being trained with only 320 images, and are competitive even under *Few-shot* and unsupervised conditions. We conjecture that this improvement owes primarily to the various and realistic synthetic data that our method is able to leverage.

We also test our denoiser on the SIDD+ Dataset (Abdelhamed et al. 2020) to show its generalization capability. Our model generalizes to new scenes and conditions, *i.e.* we improve 1.31 dB over CycleISP. We provide these quantitative results in the supplement.

Although we have made an exhaustive study focused on Denoising, other low-level tasks *e.g.* RAW data compression, Image retouching, HDR (Xing, Qian, and Chen 2021) can benefit from our approach. We show a potential HDR application in the supplementary material.

Table 3: RAW denoising results on the SIDD Dataset. Few-shot and unsupervised variants of our method are denoted as "Ours-f" and "Ours-u" respectively.

Method	PSNR ↑	SSIM ↑
Noisy	37.18	0.850
GLIDE (Talebi and Milanfar 2014)	41.87	0.949
TNRD (Chen and Pock 2017)	42.77	0.945
KSVD (Aharon, Elad, and Bruckstein 2006)	43.26	0.969
DnCNN (Zhang et al. 2017)	43.30	0.965
NLM (Buades, Coll, and Morel 2005)	44.06	0.971
WNNM (Gu et al. 2014)	44.85	0.975
BM3D (Dabov et al. 2007)	45.52	0.980
Ours-u	49.90	0.982
DHDN (Park, Yu, and Jeong 2019)	52.02	<u>0.988</u>
Ours-f	52.05	0.986
CycleISP (Zamir et al. 2020)	<u>52.38</u>	**0.990**
Ours	**52.48**	**0.990**

4.4 Limitations

Learning an approach to approximate inverse functions of real-world ISPs is not a trivial task for the following reasons: (i) The **quantization** of the 14-bit RAW image to the 8-bit RGB image lead to inevitable information lost. We estimate this error to be 0.0022 RMSE for the SIDD. As previous work (Brooks et al. 2019; Zamir et al. 2020) we considered the uniformly distributed quantization error to be negligible when compared to other aspects on the RAW reconstruction problem (e.g. colour shift, brightness shift). (ii) For modern camera ISP, the operations and their parameters are unknown. Some operations as the value **clipping** can not be accurately inverted, and we have observed that the method can potentially degrade when large portions of the RGB image are close to overexposure. This is however not a common phenomena, most of the images in the datasets are properly exposed, and thus, the impact on performance is quite limited. (iii) We endeavour to model real camera ISPs, currently via six modules, this naturally limits performance. Learning to model and invert additional modules (*e.g.* color enhancement, deblurring), will increase modelling power.

5 Conclusion

In this paper we have proposed a novel modular, interpretable and learnable hybrid ISP model, which combines the best of both model-based and end-to-end learnable approaches. Our model performs a reversible sRGB to RAW domain transformation while learning internal ISP parameters, even under extreme low data regimes or noise. The approach recovers high quality RAW data and improves previous synthetic RAW reconstruction methods. By learning how to reverse a camera sensor and generate realistic synthetic RAW images, we can improve in downstream low-level tasks, achieving state-of-the-art performance on real camera denoising benchmarks, even with an extremely small amount of training data.

References

Abdelhamed, A.; Afifi, M.; Timofte, R.; and Brown, M. S. 2020. NTIRE 2020 Challenge on Real Image Denoising: Dataset, Methods and Results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Abdelhamed, A.; Lin, S.; and Brown, M. S. 2018. A High-Quality Denoising Dataset for Smartphone Cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Abdelhamed, A.; Timofte, R.; and Brown, M. S. 2019. NTIRE 2019 Challenge on Real Image Denoising: Methods and Results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Afifi, M.; Abdelhamed, A.; Abuolaim, A.; Punnappurath, A.; and Brown, M. S. 2021. CIE XYZ Net: Unprocessing Images for Low-Level Computer Vision Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Aharon, M.; Elad, M.; and Bruckstein, A. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11): 4311–4322.

Barron, J. T.; and Tsai, Y. 2017. Fast Fourier Color Constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Brooks, T.; Mildenhall, B.; Xue, T.; Chen, J.; Sharlet, D.; and Barron, J. T. 2019. Unprocessing Images for Learned Raw Denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Brown, M. S. 2016. Understanding the In-Camera Image Processing Pipeline for Computer Vision. In *IEEE Computer Vision and Pattern Recognition - Tutorial*.

Buades, A.; Coll, B.; and Morel, J. . 2005. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Buckler, M.; Jayasuriya, S.; and Sampson, A. 2017. Reconfiguring the Imaging Pipeline for Computer Vision. In *Proceedings of the IEEE International Conference on Computer Vision*.

Bychkovsky, V.; Paris, S.; Chan, E.; and Durand, F. 2011. Learning photographic global tonal adjustment with a database of input / output image pairs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Chen, Y.; and Pock, T. 2017. Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6): 1256–1272.

Dabov, K.; Foi, A.; Katkovnik, V.; and Egiazarian, K. 2007. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8): 2080–2095.

Delbracio, M.; Kelly, D.; Brown, M. S.; and Milanfar, P. 2021. Mobile Computational Photography: A Tour. arXiv:2102.09000.

Gharbi, M.; Chaurasia, G.; Paris, S.; and Durand, F. 2016. Deep Joint Demosaicking and Denoising. *ACM Transactions on Graphics*, 35(6).

Gijsenij, A.; Gevers, T.; and Lucassen, M. P. 2009. Perceptual analysis of distance measures for color constancy algorithms. *Journal of the Optical Society of America A*, 26(10).

Gu, S.; Zhang, L.; Zuo, W.; and Feng, X. 2014. Weighted Nuclear Norm Minimization with Application to Image Denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Heide, F.; Steinberger, M.; Tsai, Y.-T.; Rouf, M.; Pajak, D.; Reddy, D.; Gallo, O.; Liu, J.; Heidrich, W.; Egiazarian, K.; Kautz, J.; and Pulli, K. 2014. FlexISP: A Flexible Camera Image Processing Framework. *ACM Transactions on Graphics*, 33(6).

Hernandez-Juarez, D.; Parisot, S.; Busam, B.; Leonardis, A.; Slabaugh, G.; and McDonagh, S. 2020. A multi-hypothesis approach to color constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2270–2280.

Ho, J.; Chen, X.; Srinivas, A.; Duan, Y.; and Abbeel, P. 2019. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. In *Proceedings of the International Conference on Machine Learning*.

Huiskes, M. J.; Thomee, B.; and Lew, M. S. 2010. New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative. In *Proceedings of the International Conference on Multimedia Information Retrieval*.

Ignatov, A.; Malivenko, G.; Plowman, D.; Shukla, S.; and Timofte, R. 2021. Fast and Accurate Single-Image Depth Estimation on Mobile Devices, Mobile AI 2021 Challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Ignatov, A.; Van Gool, L.; and Timofte, R. 2020. Replacing Mobile Camera ISP With a Single Deep Learning Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Imamura, R.; Itasaka, T.; and Okuda, M. 2019. Zero-Shot Hyperspectral Image Denoising With Separable Image Prior. In *Proceedings fo the IEEE/CVF International Conference on Computer Vision (ICCV) Workshop*.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31.

Koskinen, S.; Yang, D.; and Kämäräinen, J. 2019. Reverse Imaging Pipeline for Raw RGB Image Augmentation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*.

Liang, Z.; Cai, J.; Cao, Z.; and Zhang, L. 2021. CameraNet: A Two-Stage Framework for Effective Camera ISP Learning. *IEEE Transactions on Image Processing*, 30: 2248–2262.

Liu, Y.-L.; Lai, W.-S.; Chen, Y.-S.; Kao, Y.-L.; Yang, M.-H.; Chuang, Y.-Y.; and Huang, J.-B. 2020. Single-Image HDR Reconstruction by Learning to Reverse the Camera Pipeline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mantiuk, R.; Mantiuk, R.; Tomaszewska, A.; and Heidrich, W. 2009. Color correction for tone mapping. In *Computer Graphics Forum*, volume 28.

Park, B.; Yu, S.; and Jeong, J. 2019. Densely Connected Hierarchical Network for Image Denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Punnappurath, A.; and Brown, M. S. 2020. Learning Raw Image Reconstruction-Aware Deep Image Compressors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4): 1013–1019.

Qian, G.; Gu, J.; Ren, J. S.; Dong, C.; Zhao, F.; and Lin, J. 2019. Trinity of Pixel Enhancement: a Joint Solution for Demosaicking, Denoising and Super-Resolution. *arXiv preprint arXiv:1905.02538*.

Reinhard, E.; Stark, M.; Shirley, P.; and Ferwerda, J. 2002. Photographic Tone Reproduction for Digital Images. *ACM Transactions on Graphics*, 21(3): 267–276.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

Schwartz, E.; Giryes, R.; and Bronstein, A. M. 2019. DeepISP: Toward Learning an End-to-End Image Processing Pipeline. *IEEE Transactions on Image Processing*, 28(2): 912–923.

Shlezinger, N.; Whang, J.; Eldar, Y. C.; and Dimakis, A. G. 2021. Model-Based Deep Learning: Key Approaches and Design Guidelines. In *Proceedings of the IEEE Data Science and Learning Workshop (DSLW)*.

Talebi, H.; and Milanfar, P. 2014. Global Image Denoising. *IEEE Transactions on Image Processing*, 23(2): 755–768.

Wronski, B.; Garcia-Dorado, I.; Ernst, M.; Kelly, D.; Krainin, M.; Liang, C.-K.; Levoy, M.; and Milanfar, P. 2019. Handheld Multi-Frame Super-Resolution. *ACM Transactions on Graphics*, 38(4).

Xia, M.; Liu, X.; and Wong, T.-T. 2018. Invertible Grayscale. *ACM Transactions on Graphics*, 37(6).

Xing, Y.; Qian, Z.; and Chen, Q. 2021. Invertible Image Signal Processing. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Young, I. T. 2000. Shading correction: compensation for illumination and sensor inhomogeneities. *Current Protocols in Cytometry*, 14(1).

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; Yang, M.-H.; and Shao, L. 2020. CycleISP: Real Image Restoration via Improved Data Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7): 3142–3155.