Supplementary Material for Transform your Smartphone into a DSLR Camera: Learning the ISP in the Wild

Ardhendu Shekhar Tripathi[1], Martin Danelljan[1], Samarth Shukla[1], Radu Timofte[1,2], and Luc Van Gool[1,3]

[1] ETH Zurich, Switzerland
[2] University of Wurzburg, Germany
[3] KU Leuven, Belgium

In the supplementary material, we present details such as the network architecture for each of the components in our architecture. We also provide additional full-resolution results for our approach. Further, we provide additional ablations and some more qualitative results. Concretely:

- We provide the closed-form solution for the minimization problem stated for our color mapping in the main paper (Sec. A).
- We provide details about the network architecture and some other important details for all the components in our framework (Sec. B).
- We provide some additional ablations and qualitative results for the ablations stated in the main paper (Sec. C).
- We provide some additional full resolution results for our approach (Sec. D).
- We provide some more qualitative results for state-of-the-art comparisons of our method with other approaches (Sec. E).
- We provide some example captures from our ISPW dataset (Sec. F).
- We provide quantitative comparisons of our approach with some more methods (Sec. G).
- We provide a detailed study on the computation time and model complexity of our approach (Sec. H).
- We visualize the intermediate results for our ISP in the wild pipeline (Sec. I).
- We provide some additional experiments for our approach (Sec. J).

A Color Mapping

Here, we present the closed form solution to the minimization problem for learning the affine transformation for each bin centroid in our color mapping scheme (Sec. 3.3 of the main paper) stated in equation (4) of the main paper. We define $V_b^j \in \mathbb{R}^{4\times 1}$ as the affine transform calculated for bin centroid b and channel j. V_b^j is a column vector of length 4 that contains $A_b^j \in \mathbb{R}^{3\times 1}$ as the first 3 elements and $B_b^j \in \mathbb{R}$ as the last element. Using pseudo-inverses:

$$V_b^j = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T c^j \tag{1}$$

Here, $\tilde{X} \in \mathbb{R}^{N \times 4}$, where N is the total number of pixels in $\tilde{x}$ which is the output of our pre-processing network $\mathcal{P}$ (sec. 3.3 of the main paper). The i^{th} row of $\tilde{X}$, $\tilde{X}_i = \sqrt{w_{ib}^j}[\tilde{x}_i^1 \ \tilde{x}_i^2 \ \tilde{x}_i^3 \ 1]$. And $c^j \in \mathbb{R}^{N \times 1}$ are the intensity values of the j^{th} channel in the target color image c. Note that the color image c is given by the downsampled target DSLR sRGB during training and during inference, $c = \mathcal{G}(x)$ is given by our color prediction network (Sec. 3.2 of the main paper). Further, $\tilde{x}_i^1$, $\tilde{x}_i^2$ and $\tilde{x}_i^3$ are the intensity values of the red, green and blue channels, respectively at the i^{th} location in the pre-processed source image $\tilde{x}$ (Sec. 3.3 of the main paper) . The weights w_{ib}^j are calculated as in section 3.3 of the main paper.

B Network Architecture and Other Details

In this section, we provide the network architectures for each of the components proposed in the main paper.

B.1 The Color Conditional ISP Network

Here, we discuss the architecture for our color conditional RAW-to-sRGB network. Our DSLR sRGB network $\mathcal{F}(x, \hat{c})$ is conditioned on the color $\hat{c}$. Hence, it takes a 7-channel input which we get by concatenating the 4-channel phone RAW x and the 3-channel color $\hat{c}$ in the channel dimension. Our restoration net $\mathcal{F}$ comprises of a convolutional layer followed by 8 Residual-in-Residual Dense Blocks (RRDB) [7]. The resulting feature map is 2x up-scaled using an upconv layer. Our upconv layer applies a convolution followed by a leakyReLu to the 2x up-scaled feature map from the RRDB layer via nearest-neighbour interpolation.

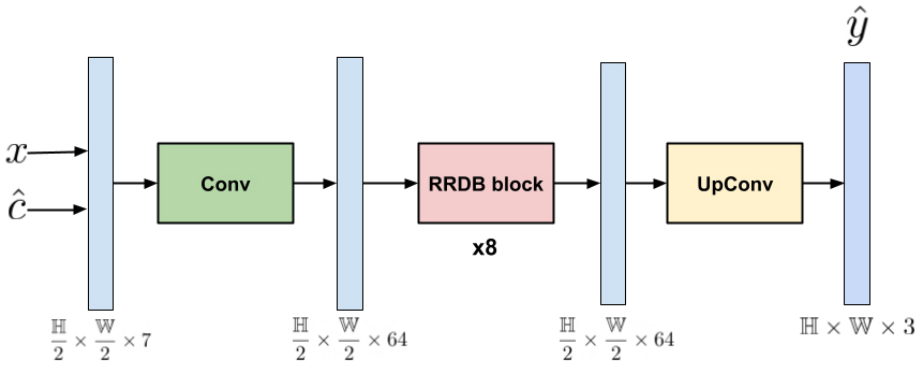


Fig. 1: Our color conditional DSLR sRGB restoration network $\mathcal{F}$.

B.2 The Pre-processing Network

Here, we state the architecture for our pre-processing net $\mathcal{P}$. The pre-processing net $\mathcal{P}$ comprises of a noise estimation module η. The architecture for our pre-processing network $\mathcal{P}$ is shown in Fig. 2. It is important to note that 2-channel 2D positional coordinates are concatenated in the channel dimension to the 3-channel processed RAW x' to mitigate the effects of vignetting that is a common phenomenon in RAW data.

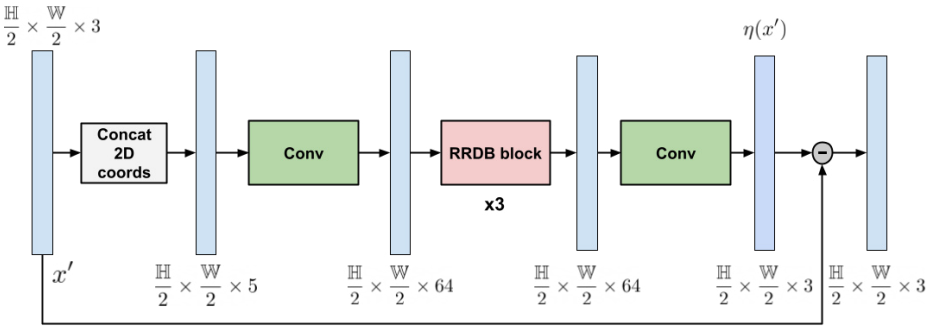


Fig. 2: Our RAW pre-processing network $\mathcal{P}$.

The processed phone RAW $x' = \Gamma(x)$ is a rough visualization of the RAW data x. We define the operation $\Gamma(x)$, henceforth. To get x', we first neglect one of the green channels in x and then normalize the resulting 3-channel image between $[0, 1]$. Further, we apply a constant approximate gamma correction to the final processed image x'. The scaling and gamma correction operations can be listed as:

$$x'^1 \coloneqq \left(x^1 / \max(x^1_{max}, 1/2.5)\right)^{\frac{1}{2.2}} \tag{2}$$

$$x'^2 \coloneqq \left(x^3 / \max(x^3_{max}, 1)\right)^{\frac{1}{2.2}} \tag{3}$$

$$x'^3 \coloneqq \left(x^4 / \max(x^4_{max}, 1/1.4)\right)^{\frac{1}{2.2}}. \tag{4}$$

The above operations encompass the functional $\Gamma(x)$. Here, x'^1, x'^2 and x'^3 are the red, green and blue channels, respectively of x'. And, x^1_{max}, x^3_{max} and x^4_{max} are the max values in the red, green (one of the green) and blue channels, respectively of the RAW x. The specific scaling factors in the above mentioned power law were arrived by quantitative evaluation of the data. Further, the gamma correction factor of $1/2.2$ is a commonly used value in imaging systems.

B.3 The Color Prediction Network

Encoder block: Figure 3 shows the architecture at each of the levels in the contracting path of our U-Net. Each of these modules comprises of 2 convolutional

layers comprising of successive convolution and leakyReLu activations. The convolutional layer is followed by an efficient Global . A skip connection between the input and output of the Global Context Transformer makes the learning more stable and efficient. The resulting feature map is then average pooled and passed on to the next contracting level. The number of input channels at level l is given by $\mathbb{D}_l = 64 \times 2^l$ where $l \in \{1, 2, 3\}$. For level $l = 0$, $\mathbb{D}_l = 6$ *i.e.* the phone RAW data is concatenated with the 2D positional coordinates to mitigate vignetting that is a common in RAW sensor data. For the Global Context Transformer, the learned latent vector $Z_l \in \mathbb{R}^{\frac{1024}{2^l} \times 2^{l+7}}$ at level l of the contracting path. Fixing the size of the latent vectors limits the computational complexity for attention to linear in the input instead of quadratic. The number of levels in both, the contracting and expanding path's is set to 4.

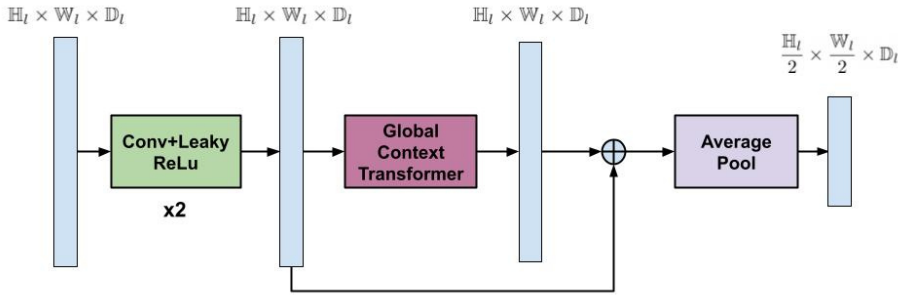


Fig. 3: The encoder blocks in the contracting path of our DSLR color predictor $\mathcal{G}$.

Decoder block: Figure 4 shows the architecture at each of the levels in the expanding paths (both our decoders). Each of these modules comprises of a transposed2D convolution with kernel size=2 and the stride=2. This is followed by concatenating the features from the corresponding level in the contracting path. The resulting feature map is finally passed through a couple of convolutional layers comprising of successive convolutions and leakyReLu activations.

As a final layer, our RAW reconstruction decoder applies an extra 3×3 convolution to the output of the respective U-Net decoder branch. And, the DSLR color predictor branch employs a RRDB block to the output of the respective decoding branch.

C Detailed Ablative Experiments

In this section, we provide additional ablations for our approach and provide qualitative results for the ablations discussed in the main paper.

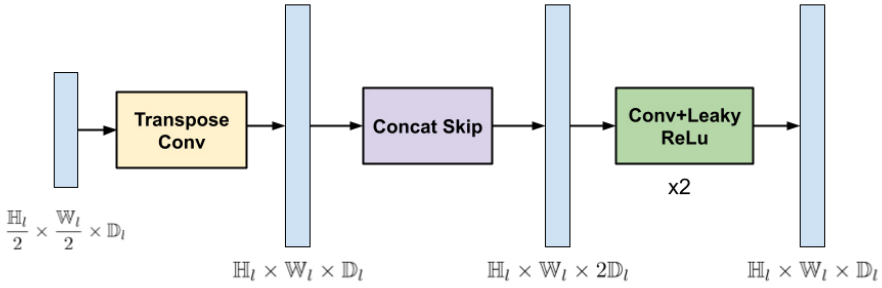


Fig. 4: The decoder blocks in the expanding path of our color predictor $\mathcal{G}$.

C.1 Additional Ablations

In addition to the ablations provided in the main paper, here we provide some more ablations on the test set of the ZRR dataset. The evaluation criteria remains the same as in the main paper.

Table 1: Impact of joint fine-tuning of our model components $\mathcal{F}$ and $\mathcal{G}$, starting from the independent training used in the paper. Results listed on the ZRR dataset.

	Independent Train	**Joint Fine-tuning**
PSNR↑	25.24	25.27
SSIM↑	0.879	0.883

Impact of joint fine-tuning of our model components $\mathcal{F}$ and $\mathcal{G}$, starting from the independent training: Here, we do a comparative study of the independent training of our ISP network $\mathcal{F}$ and Color Prediction $\mathcal{G}$ versus the joint fine-tuning of $\mathcal{F}$ and $\mathcal{G}$. Training $\mathcal{F}$ and $\mathcal{G}$ independently allows us to use larger batch sizes, hence faster convergence of the training. We investigate joint fine-tuning of both, our ISP net $\mathcal{F}$ and the Color Prediction net $\mathcal{G}$ by starting from the independently pretrained $\mathcal{F}$ and $\mathcal{G}$ models. The batch size is reduced to 8 (versus 16 when we train $\mathcal{F}$ and $\mathcal{G}$ independently). Table 1 shows the effect of this joint fine-tuning compared to independent training of our $\mathcal{F}$ and $\mathcal{G}$ on the ZRR dataset. It is evident from Tab.1 that the improvement is negligible when we jointly fine-tune our ISP net $\mathcal{F}$ and our color predictor $\mathcal{G}$. Thus, justifying our choice of independently training $\mathcal{F}$ and $\mathcal{G}$.

Impact of different alignment strategies for ISP Network loss computation: Next, we analyze the different alignment strategies in our ISP Network Loss (Eq. 6 of the manuscript). First, we report results for align the DSLR sRGB with the phone RAW (Align GT with RAW) for ISP Network Loss calculation.

Table 2: Impact of different alignment strategies for ISP Network Loss computation (Eq. 6) of the main paper. Results listed on the ZRR dataset.

	Align GT with RAW	**Align GT with Prediction**	**Align Prediction with GT**
PSNR↑	25.09	25.24	25.26
SSIM↑	0.874	0.879	0.881
Training time (hrs)↓	26.0	26.8	29.2

We observe a drop in performance compared to the case where we align the DSLR sRGB with the ISP Net prediction (Align GT with Prediction). This drop can be explained by the fact that aligning the DSLR sRGB with the RAW involves estimation of the optical flow in a low resolution (downsampled DSLR sRGB aligned with x') and then upscaled (via bilinear interpolation) by a factor of 2. This introduces some warping inaccuracies and hence, the drop in performance. On the other hand, aligning the ISP Net prediction with the DSLR sRGB (Align Prediction with GT) gives a very slight improvement in terms of the PSNR while increasing the training time of the ISP Net $\mathcal{F}$ by almost 10% because this alignment strategy involves differentiating through the warping process. Hence, we align the DSLR sRGB with the ISP Net prediction for the ISP Network Loss calculation.

We also time each of our training iterations (with a batch size of 16). Computation of the optical flow and warping in each training step is not the bottleneck: only 11% of the time in a training iteration (2.6s). The forward time was found to be $1.1s$, while the backward time was $0.9s$. The total loss calculation takes $0.6s$ (this also encompasses the optical flow). It is important to note that the timings are a bit inflated because of the time() function usage in python.

Table 3: Additional ablative study for our color mapping scheme - unlike the ablation provided in the main paper (Tab. 1 of the main paper), we feed in directly the color $\hat{c} = \mathcal{G}(x)$ into $\mathcal{F}$ without the color mapping $\mathcal{C}$ during inference. Results listed on the ZRR dataset.

	PSNR↑	**SSIM↑**
NoColorPred	21.27	0.844
ColorBlur	23.43	0.857
LinearMap	22.16	0.839
ConstValMap	22.96	0.860
AffineMapIndep	23.90	0.863
AffineMapDep	24.46	0.873
+Preprocess	25.19	0.878

Effect of color mapping during inference: We additionally ablate the use of our color mapping scheme $\mathcal{C}$ at inference for our approach. In table 1 of the

main paper, we provided the ablation for various color mapping schemes. Here, we provide an additional ablation (Tab. 3) where unlike in the main paper we feed in directly the color $\hat{c} = \mathcal{G}(x)$ into $\mathcal{F}$ without the color mapping $\mathcal{C}$ during inference. For each of the ablations the corresponding network is still trained with the respective color mapping scheme. From Tab. 3, it is evident that for the less powerful color mapping schemes, it is better to directly feed in the the color image $\hat{c} = \mathcal{G}(x)$ into our color conditional restoration network $\mathcal{F}$. On the other hand, we observe that using a powerful and a more flexible color mapping scheme like ours is beneficial during inference giving a boost of 0.05 in PSNR over the case where we do not employ the color mapping at inference (Tab. 3). Hence, in our final architecture we apply our color mapping from Pre-processed RAW $\tilde{x}$ to the predicted color c by our color prediction net $\mathcal{G}$ during inference. This provides an additional regularization for spurious local colors that may occur in c.

Table 4: Influence of using a processed RAW x' in place of a 3-channel version of x (by neglecting one of the green channels) for our color mapping and pre-processing network. Results listed on the ZRR dataset.

	PSNR↑	**SSIM↑**
Ours-RAW	24.97	0.875
Ours	25.24	0.879

Effect of using x' instead of a 3-channel version (by neglecting one of the green channels) of the RAW x in our framework: Here, we provide an ablation for the utility of using the processed RAW x' (Eq. (2)) instead of a 3-channel version of x (by neglecting a green channel) in our color mapping $\mathcal{C}$ and our pre-processing network $\mathcal{P}$. Table 4 shows that using a processed RAW x' (Ours) aids both, our color mapping $\mathcal{C}$ and our pre-processing net $\mathcal{P}$. Hence, achieving an improvement in PSNR by 0.27 dB in comparison to the version where we use the RAW x (Ours-RAW).

Table 5: Ablative study for exploiting the 2D positional coordinates of the RAW to counter vignetting. Results listed on the ZRR dataset.

	PSNR↑	**SSIM↑**
Ours-No2DCoords	25.07	0.877
Ours	25.24	0.879

Effect of concatenating the 2D positional coordinates to the input RAW for our pre-processing network and the color predictor: Table 5 shows that using the 2D positional coordinates in our pre-processing network

and the color predictor provides us an improvement of 0.17 dB in PSNR over Ours-No2DCoords where we do not concatenate the 2D positional information to the raw input in the pre-processing network $\mathcal{P}$ and our color predictor $\mathcal{G}$. It is important to note that we found concatenating the positional information only in $\mathcal{P}$ and $\mathcal{G}$ to be beneficial. We believe that this is due to the fact that our color conditional restoration net $\mathcal{F}$ is very efficient in exploiting the color information c provided by the color predictor $\mathcal{G}$.

C.2 Color Mapping

Figure 5 shows the qualitative results for our ablative study for our proposed flexible soft attention based color mapping scheme (Sec. 3.3 of the main paper). The qualitative results clearly demonstrate that having a more expressive and flexible color mapping scheme like ours is pivotal in capturing accurate colors of the target DSLR. The qualitative results reiterate the trends noticed in the quantitative results presented in the main paper. A simple feed forward network without a color prediction network (NoColorPred) produces less accurate colors since it does not inherently capture many other factors like camera parameters and external environmental conditions that effect the color in an image. Incorporating a color prediction network in our DSLR sRGB restoration network provides us with a boost as seen in Fig. 5. Among the various alternatives that were tried, the CycleISP [8] inspired ColorBlur version fails to capture the sudden changes of color in the image contour and produces blurry results. On the other hand LinearMap computes a global color correction matrix which produces inaccurately colored images specially in terms of contrast due to its non-local addressing of the problem by LinearMap.

Among the flexible parametric color mapping based versions of our color-mapping scheme $\mathcal{C}$ (Sec. 3.3 of the main paper), the ConstValMap version that learns a fixed numeric value for each bin centroid is not powerful enough in terms of expressivity and having just 15 bins does not suffice for a reasonable performance. The accuracy in colors predicted by AffineDepMap in comparison to AffineIndepMap clearly demonstrates the the utility of exploiting the dependence between the color channels in an image for our color-mapping. Further, pre-processing the RAW (as discussed in Sec. 3.2 of the main paper) aids our color mapping immensely by getting rid of the noise that is detrimental for color mapping. As seen in the results, our Color conditional RAW-to-sRGB pipeline aided by our color prediction module $\mathcal{G}$ achieves almost identical colors to the target DSLR sRGB

C.3 Loss

Here, we show qualitatively the effectiveness of using a masked aligned loss for learning accurate RAW-to-sRGB mapping in the wild. Figure 6 shows the visual results for the ablation study of our robust masked aligned loss (refer to Sec. 5.1.2 of the main paper). The qualitative results show that computing a non-aligned loss (NoAlign) produces a blurry result due to the misalignment between

the phone RAW and the corresponding DSLR sRGB during training. Further, aligning the RAW-sRGB pairs (+AlignedLoss) during training by explicit optical flow computations [6] improves the results but, the output during inference still remains blurry and is characterized by a noticeable color shift. This is due to the fact that we do not account for the inaccuracies in optical flow computations that may occur due to many reasons such as occlusions and inaccurate flows in homogeneous regions or regions with repeating patterns. To mitigate these inaccuracies in the optical flow computation, employing a forward-backward optical flow consistency mask (Sec. 3.4 of the main paper) to our aligned loss (+Mask) produces a more detailed output with colors consistent with the target DSLR sRGB. This shows that accurate supervision using our masked loss during training provides immense gains to our DSLR sRGB restoration network.

C.4 Color Prediction

In this section, we provide the the qualitative results for our color prediction network $\mathcal{G}$. Figure 7 shows the qualitative results for the ablative study on our color prediction network. From Fig. 7, it becomes evident that conditioning RAW-to-sRGB pipeline on the color information (+U-Net) is pivotal for RAW-to-sRGB mapping in the wild. Introducing a reconstruction loss (+Reconstruct) on the reconstructed phone RAW, further improves the visual quality. Specifically, we notice that +Reconstruct accurately determines the lighting conditions (and other parameters on which the color in an image depends) at the time of capture. Thus, pointing to the utility of the reconstruction branch that helps our encoder in the color predictor module to encapsulate all the information into the encoding that is necessary for accurate color prediction. Finally, integrating our Global Context Block (+GlobalContext) outputs more coherent and consistent colors with the target DSLR sRGB. For the first example in Fig. 7, exploiting global cues helps our ISP Net to predict a sRGB image more consistent (see top right corner of the image) with the DSLR sRGB. And, in the second example the Global-Context transformer aids in predicting accurate colors for the green leaves in the image. Our final version produces colors almost identical to that of the target DSLR sRGB.

D Results on Full Resolution Images

In this section, we present full resolution results for our approach. Fig. 8 shows the full resolution (2736x3648) predictions of our approach on the ISPW dataset. Our approach produces accurate globally coherent colors w.r.t. the DSLR sRGB. On the other hand, LiteISPNet [9] produces dull inaccurate colors. Thus, underlining the utility of leveraging global context by our color prediction network. Importantly, our efficient fixed size latent-array based global attention aids in applying our models on large images since the computational complexity of our Global Context Transformer layer scales linearly with the image size. Additionally, LiteISPNet results in loss of detail compared to the DSLR quality sRGB

images produced by our approach. This shows the effectiveness of employing a masked aligned loss during training.

E Qualitative State-of-the-Art Comparisons

In this section, we exhibit our results qualitatively in comparison to other existing methods on the test sets of ZRR dataset [4] and our ISPW dataset. Figures 9 and 10 show the state-of-the-art comparison of our approach with other existing approaches on the ZRR and the ISPW datasets, respectively. The visual results clearly show the supremacy of our method in comparison to previous methods. In particular MW-ISPNet [4] and AWNet [2] produce blurry results hence demonstrating their ineffectiveness in handling misalignment between the phone RAW and the DSLR sRGB pairs during training. The effect is more adverse in case of the ISPW dataset where the degree of the aforementioned pairwise misalignment is worse as compared to the ZRR dataset. Further, the LiteISPNet [9] uses an aligned loss for learning a mapping between the phone RAW and the DSLR sRGB. Though, this reduces the blur (does not completely get rid of it) in the results as in previously mentioned methods, it lacks detail and suffers a significant color shift. Our approach on the other hand leapfrogs LiteISPNet significantly by providing very crisp results capturing rich details and accurate colors. This is clearly evident from our visual results. Further, in Fig. 10 we also show the results from the phone ISP. We notice that in many cases our results are richer in detail as compared to the target DSLR sRGB and the resulting sRGB from the phone ISP. This underlines the effectiveness of our approach for RAW-to-sRGB mapping in the wild.

F ISP in the Wild (ISPW) dataset

Here, we demonstrate a few example images captured in our ISP in the Wild (ISPW) dataset. Fig. 11 demonstrates that our ISPW dataset is captured in varying lighting and weather conditions. Thus making ISPW a very challenging dataset for training and benchmarking ISP pipelines in the wild.

Further, we provide a few example crops from our ISPW dataset after data processing (Sec. 4 of the main paper). We capture the DSLR sRGB at 3 different exposures for the same phone RAW (Fig. 12). We consider the DSLR sRGB captured with an EV setting 0 as the target for our RAW-to-sRGB mapping in the wild. Apart from providing various additional metadata that can further aid RAW-to-sRGB mapping in the wild research, we also provide the DSLR sRGB at 2 additional exposure settings which can be further used by the community for research directions such as automatic exposure correction [1] and various other avenues. In Tab. 6, we show the distributions for the time of day and shutter speed for the phone camera, proving the variability in the dataset.

Time of the day						Shutter speed						
1000hrs	**1200hrs**	**1400hrs**	**1600hrs**	**1800hrs**	**2000hrs**	**1/8000s**	**1/4000s**	**1/2000s**	**1/1000s**	**1/500s**	**1/250s**	**1/125s**
10.14%	25.57%	19.07%	28.43%	13.21%	3.58%	11.27%	18.51%	12.36%	24.29%	26.14%	5.4%	2.03%

Table 6: Distribution of capture time and shutter speed in ISPW dataset.

	STAIR [5]	**Baidu [5]**	**skyb [5]**	**Airia_CG [5]**	**PyNet [3]**	**MW-ISPNet [5]**	**AWNet [2]**	**LiteISPNet [9]**	**Ours**
PSNR	21.57	21.91	21.93	22.26	22.73	23.13	23.52	23.81	**25.24**
SSIM	0.785	0.783	0.787	0.791	0.845	0.849	0.855	0.873	**0.879**

Table 7: More state-of-the-art comparison on the ZRR dataset.

	Ours	**OursV1.1**	**OursV1.2**	**OursFast**	**LiteISPNet [9]**
PSNR	**25.05**	25.02	24.76	24.57	23.51
SSIM	**0.821**	0.819	0.816	0.815	0.809
Time (320x320 crop)	55.7ms	41.3ms	20.3ms	**13.8ms**	17.2ms
Time (2736x3648 full res. image)	2.1s	1.3s	0.52s	**0.34**s	0.46s

Table 8: Time and performance comparison on ISPW dataset.

G Comparison with more methods

We report more works in Tab. 7. We outperform all previous approaches by a large margin. Since learning-based ISP is an emerging research direction, there are very few existing works in CVPR/ICCV/ECCV. We believe that our work and dataset will bring further interest to this important research direction.

H Computation Time and Model Complexity

We report computation time and performance of ours compared to LiteISPNet in Tab. 8. We evaluate 3 additional settings of our approach. In **OursV1.1** we omit the color mapping regularization $\mathcal{C}$ after the color prediction network. In **OursV1.2** and **OursFast** we further reduce the number of parameters by $\sim 2\times$ and $\sim 3\times$ resp., by reducing the depth of all three networks and dimensionality of the Global Context Block. Our fastest setting outperforms LiteISPNet by 1.06dB while being 26% faster, requiring only 0.34s for full resolution images. Further, Tab. 9 reports the component wise #params and runtime for our network. The color prediction net and ISP net have similar contribution to #params and runtime.

I Visual results for various components in our ISP pipeline

In this section, we show the visual results for different components in our RAW-to-sRGB mapping in the wild pipeline. Figure 13 shows the intermediate results

	Ours		OursV1.1		OursV1.2		OursFast	
	#params	**time**	**#params**	**time**	**#params**	**time**	**#params**	**time**
Preprocessing Net	5.1M	4.9ms	5.1M	4.9ms	3.7M	3.4ms	2.3M	2.1ms
Color Prediction Net	17.7M	21.1ms	17.7M	21.1ms	9.2M	10.1ms	6.5M	7.2ms
Color Mapping	0	14.4ms	-	-	-	-	-	-
ISP Net	12.4M	15.3ms	12.4M	15.3ms	5.9M	6.8ms	4.6M	4.5ms
Total	35.2M	55.7ms	35.2M	41.3ms	18.8M	20.3ms	13.4M	13.8ms

Table 9: Parameters and runtime for each network component.

for our ISP Network. We show that $x' = \Gamma(x)$ (Eq. 2) provides our pipeline with a rough visualization for the phone RAW x. This processed RAW x' aids in creating a mask for regions where alignment is difficult leading to a more accurate training supervision. We also see that, our Global-Context transformer based color predictor predicts a color image $c = \mathcal{G}(x)$ that is consistent with the colors in the target DSLR sRGB y. Our flexible parametric color mapping scheme is powerful enough to color-map the pre-processed RAW $\tilde{x}$ to the predicted color image $c = \mathcal{G}(x)$ very accurately with just 15 bins. Finally, our RAW-to-sRGB restoration network predicts the DSLR quality sRGB image $y = \mathcal{F}(x, \hat{c})$.

J Additional Experiments

Feature maps from our Color-Prediction Net: Figure 14 shows the feature maps from different encoder decoder levels in our U-Net color predictor network $\mathcal{G}$. The network captures detailed image information at different levels.
Cross-dataset experiment: Next, to check how our models perform on datasets they are not trained on. We do inference on the ISPW dataset using the model trained on the ZRR dataset and vice versa. Figures 15 and 16 show the visual results on example crops from both the datasets. It is evident from the qualitative results that our framework is able to produce feasable DSLR quality sRGB's even when it is run on a dataset it is not trained on.

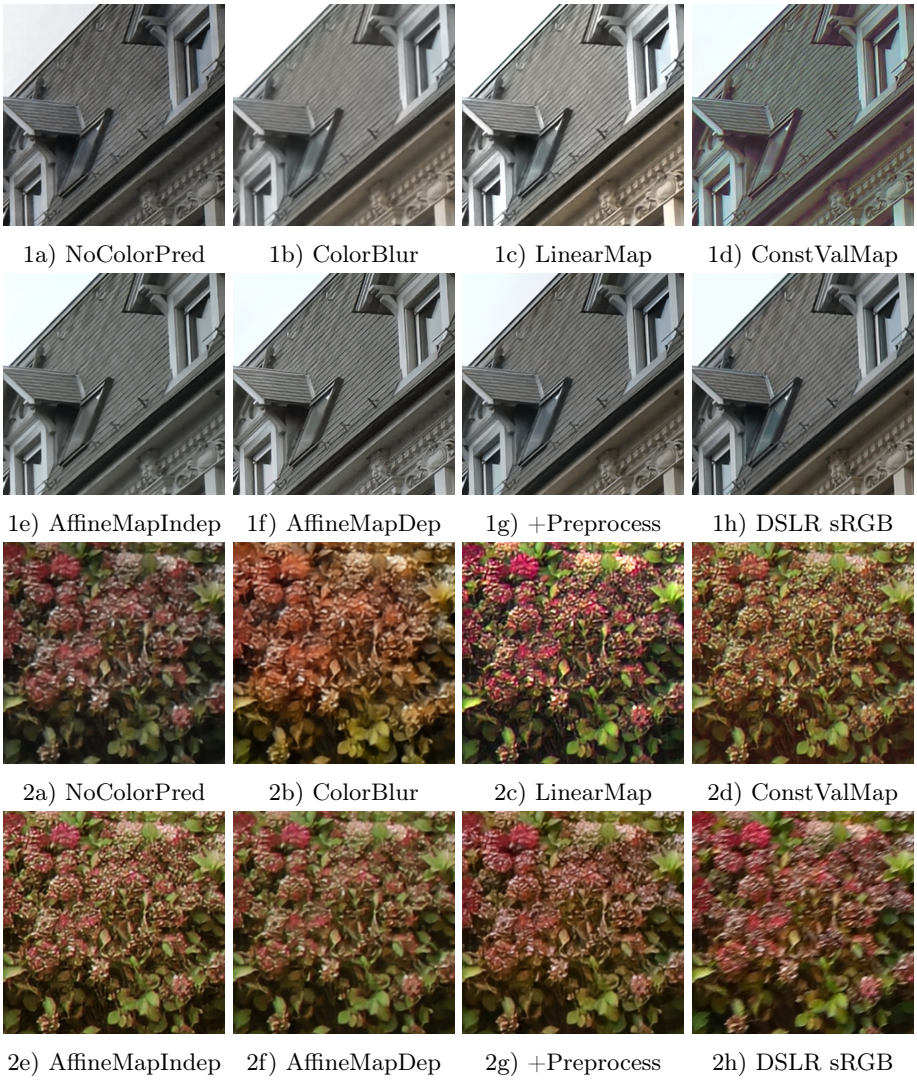
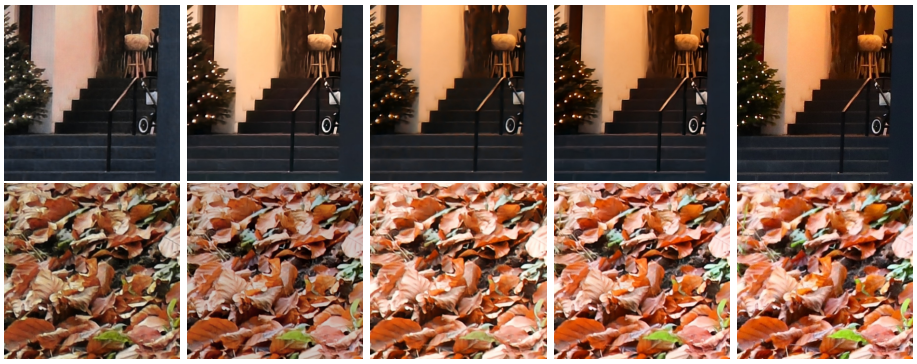


Fig. 5: Qualitative results for the ablation of our color mapping (Sec. 3.3 of the main paper). These results demonstrate qualitatively our ablation study in section 5.1.1 of the main paper. The crops are taken from the ZRR dataset. Best viewed with zoom.

(a) NoAlign (b) +AlignedLoss (c) +Mask (d) DSLR sRGB

Fig. 6: Qualitative results for the ablation of our robust masked loss (Sec. 3.4 of the main paper). These results demonstrate qualitatively our ablation study in section 5.1.2 of the main paper. The crops are taken from the ZRR dataset. Best viewed with zoom.



(a) NoColor-Pred (b) +U-Net (c) +Recon-struct (d) +Global-Context (e) DSLR sRGB

Fig. 7: Qualitative results for the ablation of our color prediction network (Sec. 3.2 of the main paper). These results demonstrate qualitatively our ablation study in section 5.1.3 of the main paper. The crops are taken from the ZRR dataset. Best viewed with zoom.

1a) RAW Visualized 1b) LiteISPNet

1c) Ours 1d) DSLR sRGB

2a) RAW Visualized 2b) LiteISPNet

2c) Ours 2d) DSLR sRGB

Fig. 8: Full resolution results on our ISPW dataset. We compare our method against the best performing competing method LiteISPNet [9]. Our approach captures more details and more accurate colors w.r.t. the DSLR sRGB. On the other hand, LiteISPNet produces dull colors and results in loss of detail. Best viewed with zoom.

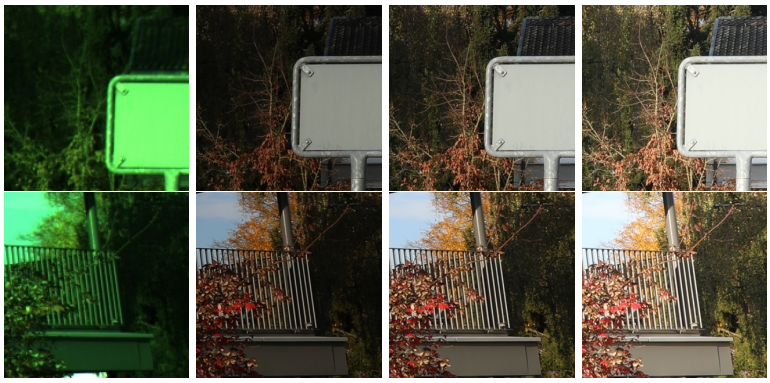
(a) MWISPNet (b) AWNet (c) LiteISPNet (d) Ours (e) DSLR sRGB

Fig. 9: Some more visual results for state-of-the-art comparison on the ZRR [4] dataset. Best viewed with zoom.

(a) MWISPNet (b) AWNet (c) LiteISPNet (d) Ours (e) DSLR sRGB

Fig. 10: Some more visual results for state-of-the-art comparison on our ISPW dataset. Best viewed with zoom.

Fig. 11: Example captures from our ISPW dataset. We show some example captures from the DSLR camera. As demonstrated, the ISPW dataset is collected in various lighting and weather conditions which makes it a very challenging dataset for learning and benchmarking the full ISP pipeline in the wild.

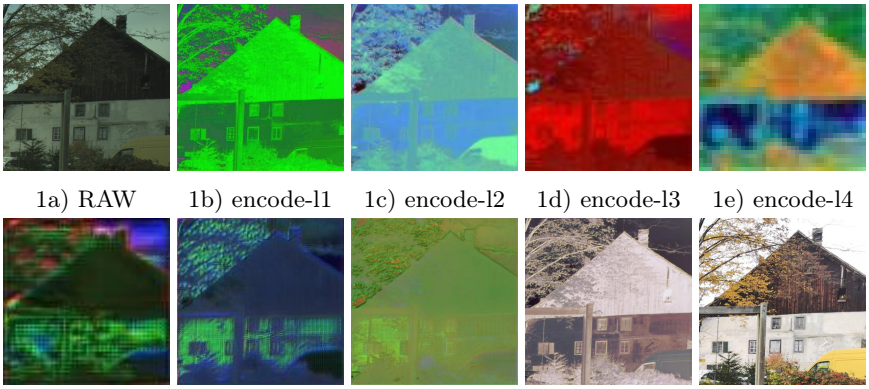


(a) Phone RAW (b) DSLR sRGB: EV = -1 (c) DSLR sRGB: EV = 0 (d) DSLR sRGB: EV = 1

Fig. 12: Example crops from our ISPW dataset. We collect DSLR sRGB's at three different exposure settings. Note that we use the DSLR sRGB at EV setting of 0 for training our Color conditional DSLR sRGB restoration network.

(a) x (b) x' (c) $\tilde{x}$ (d) c (e) $\hat{c}$ (f) $\hat{y}$ (g) y

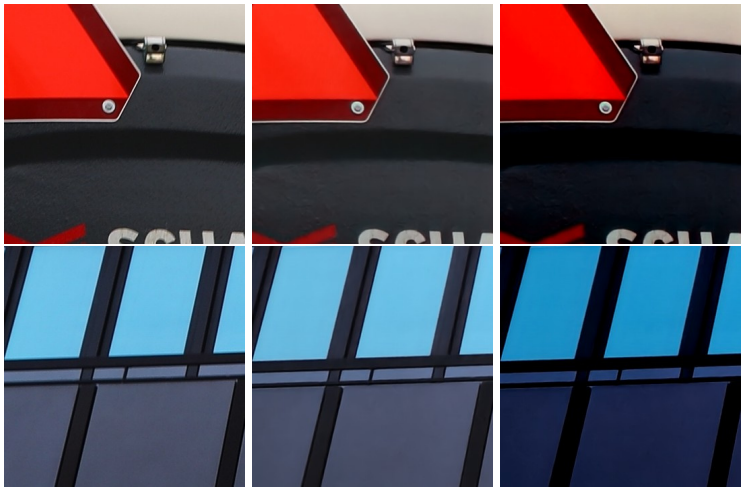
Fig. 13: We show the intermediate predictions in our framework for a few examples in the ZRR dataset. In the figure, x is the visualized RAW from the phone and $x' = \Gamma(x)$ (Eq. 2). The output of the Pre-processing network (Sec. 3.3 of the main paper) $\tilde{x}$ is shown in column 3. Further, $c = \mathcal{G}(x)$ is the predicted low-resolution color image by our color prediction network (Sec. 3.2 of the main paper) that integrates a global context transformer to integrate global cues for predicting accurate colors. The pre-processed RAW $\tilde{x}$ is then color mapped to c using our parametric color mapping formulation (Sec. 3.3 of the main paper). The color mapped image $\hat{c} = \mathcal{C}(\tilde{x}, c)$. During inference the parametric color mapping $\mathcal{C}$ aids in smoothing out the spurious color predictions that may occur in c. Finally, our ISP network predicts the final DSLR quality $\hat{y} = \mathcal{F}(x, \hat{c})$. The last column shows the DSLR sRGB (y) crop. Best viewed with zoom.



1a) RAW 1b) encode-l1 1c) encode-l2 1d) encode-l3 1e) encode-l4

1f) decode-l1 1g) decode-l2 1h) decode-l3 1i) decode-l4 1j) c

Fig. 14: We show the visualized (by taking the first 3 channels) resulting feature maps at each U-Net level (both encoder and the DSLR decoder) for an example crop from our ISPW dataset. Here, encode-ln signifies the feature map output from our encoder block at level n. Similarly, decode-ln is the feature map output from our encoder block at level n. Best viewed with zoom.



(a) DSLR sRGB (ZRR dataset) (b) Ours-ZRR (c) Ours-ISPW

Fig. 15: Testing our model trained on the ISPW dataset on two example crops from the ZRR dataset. Ours-ISPW shows the results for the model trained on our ISPW dataset. Ours-ZRR is the result of the model trained on the ZRR dataset. produces

(a) DSLR sRGB (ISPW dataset) (b) Ours-ISPW (c) Ours-ZRR

Fig. 16: Testing our model trained on the ZRR dataset on two example crops from the ISPW dataset. Ours-ISPW shows the results for the model trained on our ISPW dataset. Ours-ZRR is the result of the model trained on the ZRR dataset.

References

1. Afifi, M., Derpanis, K.G., Ommer, B., Brown, M.S.: Learning multi-scale photo exposure correction. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 9157–9167. Computer Vision Foundation / IEEE (2021), https://openaccess.thecvf.com/content/CVPR2021/html/Afifi_Learning_Multi-Scale_Photo_Exposure_Correction_CVPR_2021_paper.html
2. Dai, L., Liu, X., Li, C., Chen, J.: Awnet: Attentive wavelet network for image ISP. In: Bartoli, A., Fusiello, A. (eds.) Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12537, pp. 185–201. Springer (2020). https://doi.org/10.1007/978-3-030-67070-2_11, https://doi.org/10.1007/978-3-030-67070-2_11
3. Ignatov, A., Gool, L.V., Timofte, R.: Replacing mobile camera ISP with a single deep learning model. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020. pp. 2275–2285. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPRW50498.2020.00276
4. Ignatov, A., Timofte, R., Zhang, Z., Liu, M., Wang, H., Zuo, W., Zhang, J., Zhang, R., Peng, Z., Ren, S., Dai, L., Liu, X., Li, C., Chen, J., Ito, Y., Vasudeva, B., Deora, P., Pal, U., Guo, Z., Zhu, Y., Liang, T., Li, C., Leng, C., Pan, Z., Li, B., Kim, B., Song, J., Ye, J.C., Baek, J., Zhussip, M., Koishekenov, Y., Ye, H.C., Liu, X., Hu, X., Jiang, J., Gu, J., Li, K., Tang, P., Hou, B.: AIM 2020 challenge on learned image signal processing pipeline. In: Bartoli, A., Fusiello, A. (eds.) Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12537, pp. 152–170. Springer (2020). https://doi.org/10.1007/978-3-030-67070-2_9, https://doi.org/10.1007/978-3-030-67070-2_9
5. Ignatov, A., Timofte, R., Zhang, Z., Liu, M., Wang, H., Zuo, W., Zhang, J., Zhang, R., Peng, Z., Ren, S., Dai, L., Liu, X., Li, C., Chen, J., Ito, Y., Vasudeva, B., Deora, P., Pal, U., Guo, Z., Zhu, Y., Liang, T., Li, C., Leng, C., Pan, Z., Li, B., Kim, B., Song, J., Ye, J.C., Baek, J., Zhussip, M., Koishekenov, Y., Ye, H.C., Liu, X., Hu, X., Jiang, J., Gu, J., Li, K., Tang, P., Hou, B.: AIM 2020 challenge on learned image signal processing pipeline. In: Bartoli, A., Fusiello, A. (eds.) Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12537, pp. 152–170. Springer (2020). https://doi.org/10.1007/978-3-030-67070-2_9, https://doi.org/10.1007/978-3-030-67070-2_9
6. Sun, D., Yang, X., Liu, M., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. CoRR **abs/1709.02371** (2017), http://arxiv.org/abs/1709.02371
7. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European conference on computer vision (ECCV) workshops. pp. 0–0 (2018)
8. Zamir, S.W., Arora, A., Khan, S.H., Hayat, M., Khan, F.S., Yang, M., Shao, L.: Cycleisp: Real image restoration via improved data synthesis. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 2693–2702. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.00277

9. Zhang, Z., Wang, H., Liu, M., Wang, R., Zhang, J., Zuo, W.: Learning raw-to-srgb mappings with inaccurately aligned supervision. CoRR **abs/2108.08119** (2021), https://arxiv.org/abs/2108.08119