Supplemental Materials of Rawformer: Unpaired Raw-to-Raw Translation for Learnable Camera ISPs

Georgy Perevozchikov[1], Nancy Mehta[1]*, Mahmoud Afifi[2]**, and Radu Timofte[1]

[1] Computer Vision Lab, CAIDAS & IFI, University of Würzburg, John Skilton Str. 4a, 97074 Würzburg, Germany
{georgii.perevozchikov,nancy.mehta,radu.timofte}@uni-wuerzburg.de
[2] York University, 4700 Keele St, Toronto, Ontario, Canada, M3J 1P3
m.3afifi@gmail.com

In this supplementary material, we first discuss the holistic architecture of our proposed Rawformer along with the experimental details in Sec. 1. Then, we delve into the details of the style modulator along with the demonstration of its efficacy in the restoration of the overall structural fidelity of the translated images in Sec. 2. Next, we present some additional ablation studies and inference time comparison to prove the effectiveness of the proposed components in Sec. 3. Lastly, we provide additional results, including mappings between DSLR and mobile phone cameras, as well as additional visual results, in Sec. 5.

1 Architecture and Experimental Details

In the main paper, we introduced Rawformer, a fully unsupervised framework for raw-to-raw translation. The overview of the proposed framework and the training flow are illustrated in Fig. 1. As can be seen, we train two generator networks, G_{A-B} and G_{B-A}, alongside two discriminator networks, D_A and D_B, utilizing cycle consistency loss, identity loss, and discriminator loss. Our framework accurately maps images from domain A to domain B through a fully unsupervised training scheme.

1.1 Pre-training of Generator Networks

The pre-training phase of the generator network, essential for initializing the weights effectively, involves a self-supervised image inpainting task spanning 500 epochs. This task is designed to enhance the network's detail preservation capabilities in input images. Specifically, the network is trained on 32×32 pixel patches from the source dataset comprising of images of size 256×256, with 40% of these patches randomly masked. The masking is conducted by zeroing out the pixel values. The objective of the generator is to reconstruct the original image

* Corresponding Author
** Now at Google

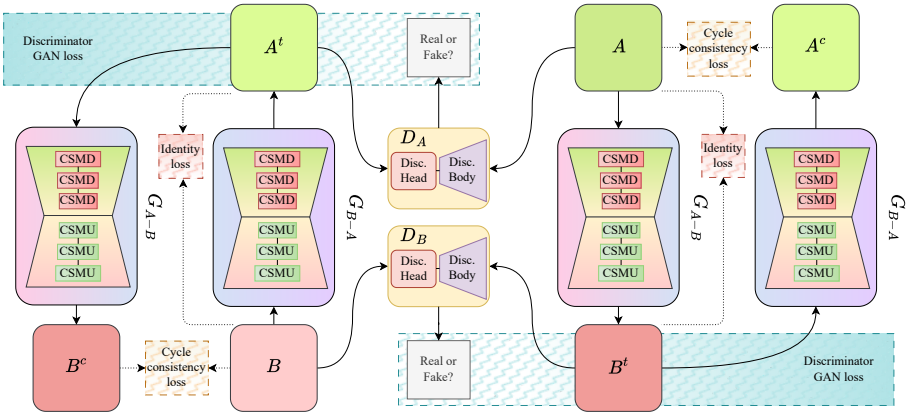

Fig. 1: Overview of the proposed architecture and training flow. A^t and B^t refer to translated images used by the discriminator loss, while A^c and B^c refer to the produced images used by the cycle consistency loss.

from its partially obscured version by minimizing a pixel-wise loss function, as discussed in the main paper. The pre-training is performed with batch size of 16 and we apply random horizontal and vertical flips on both the considered datasets. This pretraining phase employs an AdamW optimizer with betas (0.9, 0.99), a weight decay of 0.05, an initial learning rate of 0.005, and a learning schedule managed through the Cosine Annealing Warm Restarts strategy.

1.2 Raw Translation Training

After pre-training, the model enters the Generative Adversarial Network (GAN) training phase, which lasts for 500 epochs and focuses on unpaired image translation. This phase employs a pixel-wise loss function for both the generator and discriminator components, with the discriminator optimized using the Adam optimizer with betas (0.5, 0.99) and a learning rate of 0.0001. The generator uses the same optimizer but with a slightly lower learning rate of 0.00005. The batch size is kept to 1 and data augmentations, including random horizontal flips and vertical flips, are further applied on the dataset in this phase as well. The overall size of image caches is set to 3 and the overall batch head has 4 samples to compute the batch statistics. The overall training process is illustrated in Fig. 1.

2 Style Modulator

To augment the capability of the generator, we have expanded its functionality to deduce the correct target style for every input image, using a Vision Transformer (ViT) [10]. Thereafter, we modulate the decoding part of the generator with the

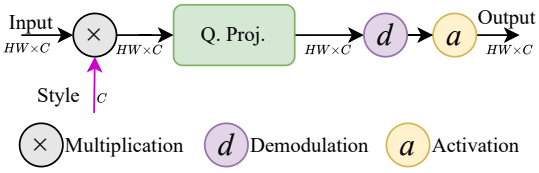

Fig. 2: Details of the proposed style modulation process.

Fig. 3: Enhancement in translation accuracy by the style modulator. The improvements are demonstrated on the Raw-to-Raw dataset [1] (from iPhone X to Samsung S9), showcasing the positive impact of the style modulator on translation accuracy.

obtained target style for markedly boosting its expressive power [7]. The main technique of style modulation is depicted in Figure 2.

In particular, at the generator's bottleneck, the image is encoded into a series of tokens for the ViT network. We enhance this series with an extra trainable style token S, which, after processing through the ViT, encapsulates the latent style of the image. For each layer in the Rawformer's decoding section, we derive a unique style vector s_i from S through trainable linear transformations.

The process of style modulation [7] effectively adjusts the weights $w_{i,j,h,w}$ of the Q vector with the designated style vector s_i, resulting in modulated weights:

$$w'_{i,j,h,w} = s_i \times w_{i,j,h,w}, \tag{1}$$

where i, j denote the input and output feature maps, respectively, and h, w represent spatial dimensions. To maintain the activation magnitudes, the modulated weights, $w'_{i,j,h,w}$, are subjected to demodulation, renormalizing the convolutional Q vector weights as:

$$w''_{i,j,h,w} = \frac{w'_{i,j,h,w}}{\sqrt{\sum_{h,w}(w'_{i,j,h,w})^2 + \epsilon}}, \tag{2}$$

with ϵ being a minimal value to avoid numerical issues. Figure 3 illustrates that the introduced style modulator significantly enhances translation precision.

Table 1: Quantitative results of optimized models. Shown results are for Samsung-to-iPhone mapping (using the Raw-to-Raw dataset [1]) on different hardware platforms, along with the inference time in milliseconds (ms). The results show that our proposed Rawformer holds promise for integration into the mobile devices.

Device	Dtype	Framework	Time (ms)	PSNR	SSIM	MAE	ΔE
CPU Intel i9-12900K	fp32	PyTorch	526	**40.98**	**0.97**	**0.01**	**2.09**
CPU Intel i9-12900K	int8	OpenVINO	179	37.21	0.95	0.03	5.14
GPU NVIDIA RTX 4090	fp32	PyTorch	26	**40.98**	**0.97**	**0.01**	**2.09**
GPU NVIDIA RTX 4090	fp16	TensorRT	**18**	40.92	**0.97**	**0.01**	2.11
Google Coral Edge TPU	int8	TF Lite	<u>68</u>	37.20	0.95	0.03	5.21

3 Inference time and Additional Ablation studies

3.1 Inference time

In the main paper, we presented the inference time of our model without optimization. Here, we present the results of various optimized versions of our model for raw translation. Specifically, we applied model post-training quantization to our trained Rawformer, which was trained to map Samsung S9's raw images to iPhone X's camera raw space. We report the results of float16, float32 and int8 quantization on different hardware platforms in Table 1. It is worth noting that we achieve nearly identical accuracy with float16 conversion, running at approximately 18 milliseconds on an NVIDIA RTX 4090 GPU.

3.2 Ablation Study

In the main paper, we presented several ablation studies conducted to validate the decisions made in the proposed design of Rawformer. Here, we present additional ablation experiments performed to further validate the operations of the major components in our Rawformer: condensed query attention (CQA), contextual-scale aware upsampler (CSAU) block and the contextual-scale aware downsampler (CSAD) blocks. Tables 2, and 3 demonstrates the impact of incorporating the spatial compression (where and how), on the attention vectors of CQA block. The results of the ablation studies shown in Tables 4, and 5 proves the efficacy of our designed composite upsamplers and downsamplers. Additionally, to prove that the different hierarchical levels in our proposed SPFN module offer a more intuitive approach for the impact of capturing local image structure across different scales, we provided an ablation study as shown in Table 6. All these ablation studies clearly reveal, that the proposed design with the inclusion of all the components achieves the best results across all quantitative metrics.

4 Comparison with SoTA approaches

Our method represents the first practical and precise solution to this challenge, relying solely on unpaired sets of raw images. While our network design draws

Table 2: Ablation results of the impact of the spatial conpression operation on different vectors of attention on the NUS dataset [2]. 'Q', 'K', and 'V' indicate the use of the Q, K, and V vectors in the CQA block. 'Ours' represents the proposed design discussed in the main paper. The shown results exhibits the proficiency of applying the spatial compression/condensation operation on the query vectors.

Spatial Compression operation	Canon-to-Nikon		Nikon-to-Canon	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
K	41.12	0.98	41.09	0.98
V	41.14	0.98	41.11	0.98
Q (Ours)	**41.89**	**0.98**	**41.37**	**0.98**

Table 3: Ablation results on the impact of different query projection operations for the condensed query attention block. Here 'Patch merging' involves splitting the incoming image feature into patches and then merging across the channel dimension, and 'Ours' represents the proposed design discussed in the main paper. The shown results clearly reveal that using average pooling and linear projection, helps in refining the overall results.

Query Projection Operation	Canon-to-Nikon		Nikon-to-Canon	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Depthwise Conv (stride=2)	40.13	0.97	40.12	0.97
Conv (stride=2)	39.52	0.97	39.48	0.97
Patch merging	38.83	0.96	38.73	0.96
MaxPool + Linear Projection	41.09	0.98	41.03	0.98
AvgPool + Linear projection	**41.89**	**0.98**	**41.37**	**0.98**

inspiration from existing techniques like channel dependencies from SENet [4] and depthwise convolutions from MobileNet [3], our goal is to develop an efficient network capable of achieving precise unpaired image translation with minimal resource requirements. Given the typical deployment of such tasks on resource-constrained devices, our modifications aim to not only reduce resource demands compared to alternatives mentioned above, but also to enhance accuracy. As demonstrated in the main paper (Tables 1-3), our design outperforms several alternative methods (e.g., UVCGANv2). Even when substituting certain blocks in our design with alternative ones, our proposed design achieves better results with reduced computations. For instance, our discriminator utilizes an attention mechanism with caching, yielding improved results compared to the UVCGANv2 discriminator: 41.89 dB vs. 40.51 dB on the Canon-to-Nikon set (CNS). Our efficient CQA block, which performs spatial compression/condensation operations on query vectors, achieves better results compared to traditional self-attention: 41.89 dB vs. 39.93 dB on CNS with a 50.7% reduction in FLOPs (also see Table 2 in supp. materials). Using our single resolution SPFN block leads to improved results compared to mutli-resolution HRNet [12] feedforward block : 41.89 dB vs. 41.77 dB on CNS with ∼87% reduction in FLOPs. We will further refine

Table 4: Ablation results on the impact of different generator model configurations on the NUS dataset [2]. 'CUp' indicates the upsampling block in the contextual-scale aware upsampler (CSAU) block, while 'Deconv' refers to the classical deconvolution block. 'Ours' represents the proposed design discussed in the main paper. The shown results demonstrate the merits of deploying hybrid upsampling in the overall design.

Configuration	Canon-to-Nikon		Nikon-to-Canon	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Deconv	40.93	0.97	40.84	0.97
CUp (Ours)	**41.89**	**0.98**	**41.37**	**0.98**

Table 5: Ablation results on the impact of different generator model configurations on the NUS dataset [2]. 'CDown' indicates downsampling in the contextual-scale aware downsampler (CSAD) block, while 'Conv' signifies the simple convolution block with stride 2 for downsampling. 'Ours' represents the proposed design discussed in the main paper.

Configuration	Canon-to-Nikon		Nikon-to-Canon	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Conv (stride 2)	40.14	0.97	40.07	0.97
CDown (Ours)	**41.89**	**0.98**	**41.37**	**0.98**

our main contributions and clarify our design motivation, as suggested by R1& R4, respectively thus eventually highlighting the ability of accurate raw-to-raw with unpaired data.

5 Additional Results

The experiments presented in the main paper focus on the raw translation of raw images captured by various mobile phone cameras, representing a real-world scenario and demonstrating one of the most promising applications of the proposed method—reducing costs associated with the development of mobile phone camera's neural-based ISP for new camera models. Here, we conducted experiments where we examined the raw mapping between DSLR and mobile phone cameras. Specifically, we trained our method, along with other unsupervised methods [8, 13], to map between the Canon EOS 600D DSLR camera (from the NUS dataset [2]) and the main camera of the Huawei P20 smartphone (from the Zurich raw-to-RGB dataset [6]). Additionally, we trained a LAN neural-based ISP [11] on the target camera. In Table 7, we present quantitative results obtained by rendering raw images mapped using our method and other methods, utilizing the pre-trained ISP. As demonstrated, our approach yields superior raw translation results compared to alternative methods, as evidenced by the quality of the rendered sRGB images in comparison to the ground-truth sRGB images.

Table 6: Ablation results of different modifications of SPFN.

Task	PSNR	SSIM	MAE ($\times 1e-2$)	ΔE
Multi-scale SPFN	41.89	0.98	1.26	2.04
Single-scale SPFN	41.11	0.97	1.70	2.57

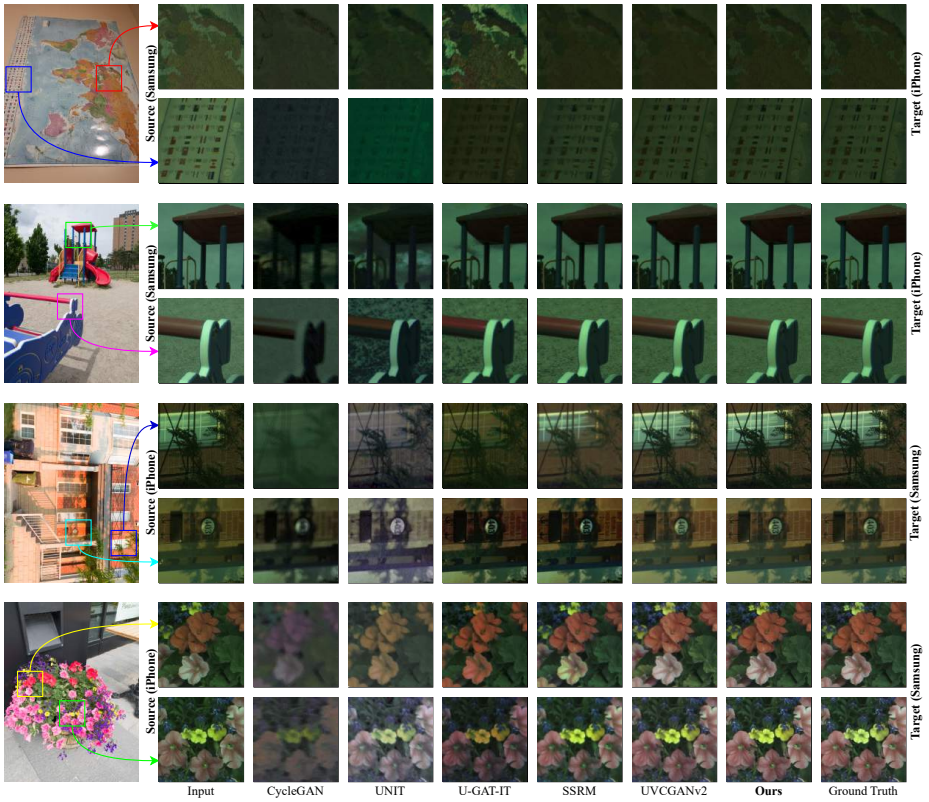

Fig. 4: Qualitative results of raw translation on the Raw-to-Raw dataset [1]. Shown are images captured by Samsung S9 and iPhone X in sRGB (left) and two cropped patches from each image in raw (right). On the right, we show the input raw patch from the corresponding camera and the corresponding ground-truth raw patch from the other camera, along with the results by other methods. Our proposed Rawformer is better at preserving the domain consistent features.

Additional qualitative raw translation results are shown in Figs. 4 and 5. In Fig. 5, we also show the results of mapped raw images (transformed from the Sony IMX586 camera to the main camera of the Huawei P20 smartphone) after rendering using a pre-trained LAN neural-based ISP [11], which was trained on the raw space of the target camera. Specifically, the ISP was trained to render images captured by the Huawei P20 smartphone's main camera. As can be seen in

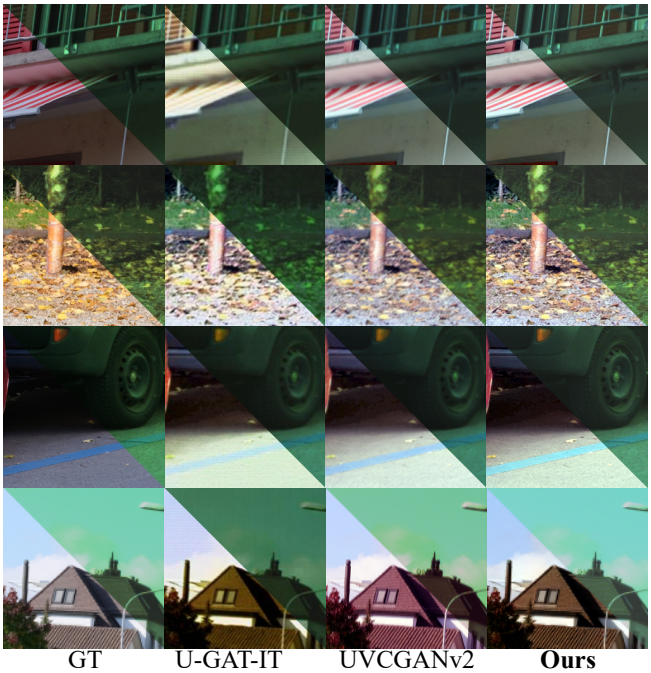

Fig. 5: Qualitative comparisons on raw translation and ISP rendering. We show the ground-truth (GT) raw/sRGB images from the the Mobile AIM21 dataset (Sony IMX586) [5], alongside the corresponding mapped raw images to the Zurich raw-to-RGB dataset (Huawei P20 smartphone's main camera) [6] generated using various methods, including ours. Additionally, we show the rendered sRGB images by processing each mapped raw image using a neural-based ISP [11] trained to render raw images from the Zurich dataset source camera (i.e., the Huawei P20 smartphone camera).

Table 7: Translation results for mapping between raw images from DSLR and mobile phone cameras. The results are on the NUS dataset [2] and the Zurich raw-to-RGB dataset (ZRR) [6]. Specifically, the mapping results of ZRR raw images (captured by the Huawei P20 smartphone camera) to the Canon EOS 600D DSLR camera, and vice versa, are shown. Both our method and other techniques are compared. *Best results are highlighted in bold.*

Methods	Canon-to-ZRR		ZRR-to-Canon	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
CycleGAN [13]	12.63	0.54	12.81	0.58
UNIT [8]	14.91	0.67	17.73	0.70
UVCGANv2 [10]	17.32	0.71	22.29	0.87
Ours	**18.71**	**0.72**	**24.35**	**0.89**

Figs. 4 and 5, our method achieves superior raw translation, resulting in visually

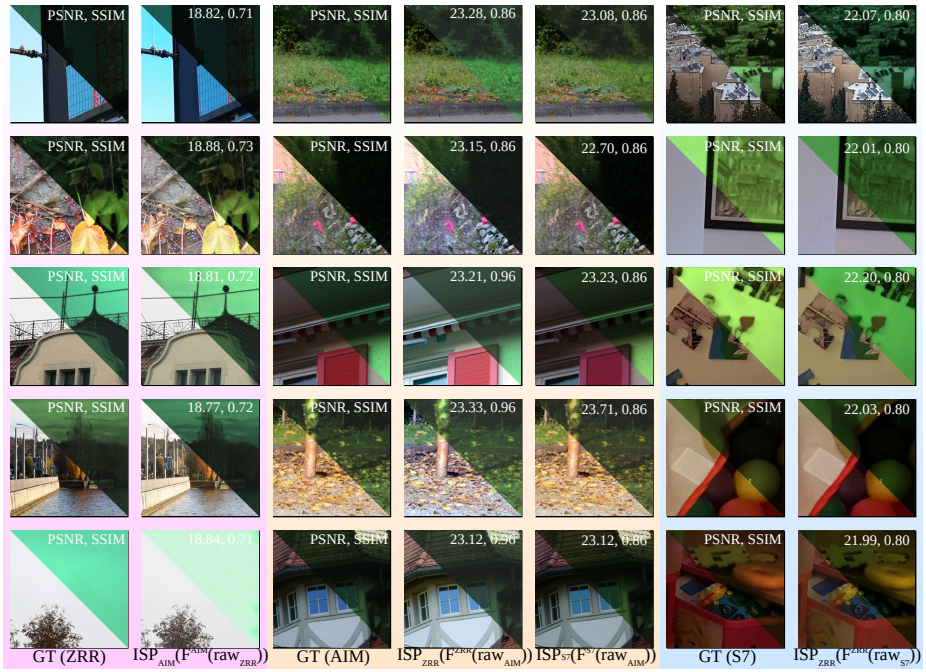

Fig. 6: ISP rendering results with our raw translation on various datasets. Each set includes ground-truth (GT) raw-sRGB paired images and LAN ISP [11] results on mapped raw images from different cameras. F^y represents our Rawformer trained to map raw images, $\texttt{raw}_x$, from a source camera, x, to target camera, y. $ISPy$ denotes LAN ISP [11] trained on raw images from camera y. S7, ZRR, and AIM stand for the Samsung S7 ISP dataset [9], Zurich raw-to-RGB dataset [6], and Mobile AIM21 dataset [5], respectively. The shown results are consistent with the ground-truth, demonstrating the proficiency of our model.

enhanced sRGB images when rendered using the pre-trained ISP compared to other alternative methods. Lastly, Fig. 6 shows additional qualitative results of our raw mapping and the rendered sRGB images using trained ISP [11] on the target camera raw space.

Additionally, we conducted experiments to test the impact of our proposed method on rendering images using Adobe Lightroom. For this experiment, we used the Raw-to-Raw dataset [1]. We compared raw images captured by a Samsung smartphone, rendered by Adobe Lightroom with metadata stored in raw DNG images captured by an iPhone, against Samsung raw images that were mapped to the iPhone raw space using our Rawformer. We performed a similar comparison on rendering iPhone images with Samsung DNG metadata, both with and without our mapping.

Table 8 presents the results, along with those of Adobe Lightroom rendering raw/metadata taken from the same phone, which represents the best case

Table 8: Results (PSNR/SSIM) via Lightroom (LR). Off-diagonal results: mapping source camera raw to target camera using Rawformer before rendering with LR using target camera's metadata.

	Target camera	
Source camera	Samsung	iPhone
Samsung	22.76/0.80	21.98/0.77
iPhone	21.42/0.76	22.08/0.79

where no mapping is needed. The results are compared against the ground-truth sRGB images of the target camera. As shown in Table 8, we achieve consistent results similar to those in the referenced paper. The empirical results demonstrate that besides achieving superior performance on learnable image signal processors (ISPs), Rawformer also exhibits a competitive edge when evaluated against commercial ISPs, like Adobe Lightroom. This indicates its robust adaptability and effectiveness across diverse ISP implementations, highlighting its potential for both research and commercial applications.

References

1. Afifi, M., Abuolaim, A.: Semi-supervised raw-to-raw mapping. In: BMVC (2021)
2. Cheng, D., Prasad, D.K., Brown, M.S.: Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. JOSA A **31**(5), 1049–1058 (2014)
3. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
4. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
5. Ignatov, A., Chiang, C.M., Kuo, H.K., Sycheva, A., Timofte, R.: Learned smartphone ISP on mobile NPUs with deep learning, mobile AI 2021 challenge: Report. In: CVPRW (2021)
6. Ignatov, A., Timofte, R., Ko, S.J., Kim, S.W., Uhm, K.H., Ji, S.W., Cho, S.J., Hong, J.P., Mei, K., Li, J., et al.: AIM 2019 challenge on raw to rgb mapping: Methods and results. In: ICCVW (2019)
7. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR (2020)
8. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: NeurIPS (2017)
9. Schwartz, E., Giryes, R., Bronstein, A.M.: DeepISP: Toward learning an end-to-end image processing pipeline. IEEE Transactions on Image Processing **28**(2), 912–923 (2018)
10. Torbunov, D., Huang, Y., Yu, H., Huang, J., Yoo, S., Lin, M., Viren, B., Ren, Y.: UVCGAN v2: An improved cycle-consistent GAN for unpaired image-to-image translation. arXiv preprint arXiv:2303.16280 (2023)
11. Wirzberger Raimundo, D., Ignatov, A., Timofte, R.: LAN: Lightweight attention-based network for raw-to-rgb smartphone image processing. In: CVPRW. pp. 807–815 (2022)

12. Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L., Sang, N., Wang, J.: Lite-hrnet: A lightweight high-resolution network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10440–10450 (2021)
13. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)