



Learning End-to-End Deep Learning Based Image Signal Processing Pipeline Using a Few-Shot Domain Adaptation

Georgy Perevozchikov^{1,2} ^a and Egor Ershov²  ^b

¹Moscow Institute of Physics and Technology (National Research University), Dolgoprudny, Russia

²Institute for Information Transmission Problems, Moscow, Russia

Keywords: Computational Photography, Image Signal Processing Pipeline, Domain Adaptation, Image Processing.

Abstract: Nowadays the quality of mobile phone cameras plays one of the most important roles in modern smartphones, as a result, more attention is being paid to the camera Image Signal Processing (ISP) pipeline. The current goal of the scientific community is to develop a neural-based end-to-end pipeline to remove the expensive and exhausting process of classical ISP tuning for each next device. The main drawback of the neural-based approach is the necessity of preparing large-scale datasets each time a new smartphone is designed. In this paper, we address this problem and propose a new method for few-shot domain adaptation of the existing neural ISP to a new domain. We show that it is sufficient to have 10 labeled images of the target domain to achieve state-of-the-art performance on the real camera benchmark datasets. We also provide a comparative analysis of our proposed approach with other existing ISP domain adaptation methods and show that our approach allows us to achieve better results. Our proposed method exhibits notably comparable performance, with only a marginal 2% drop in performance compared to the learned from scratch in the whole dataset baseline. We believe that this solution will significantly reduce the cost of neural-based ISP production for each new device.

1 INTRODUCTION

Deep CNNs have made tremendous progress in high-level computer vision applications including object identification, segmentation, and picture classification (Medioni and Dickinson, 2016). The availability of large-scale datasets with thousands of tagged pictures is a significant contributor to the generalizing ability and performance of CNNs. But on the reverse side of the coin, gathering large-scale datasets for each new device sensor is a time-consuming and expensive process.

For instance, the data acquisition procedure for the image processing pipeline is fraught with great difficulties, since it is necessary not only to collect large-scale dataset but also reconstruct its spectral sensitivities (Karaimer and Brown, 2016), which also may differ for two separate pixels of the same sensor. In practice, obtaining such data pairs often requires additional complex equipment such as a color checker, integration sphere, stable light sources, and specially equipped rooms. Moreover, the collection of such

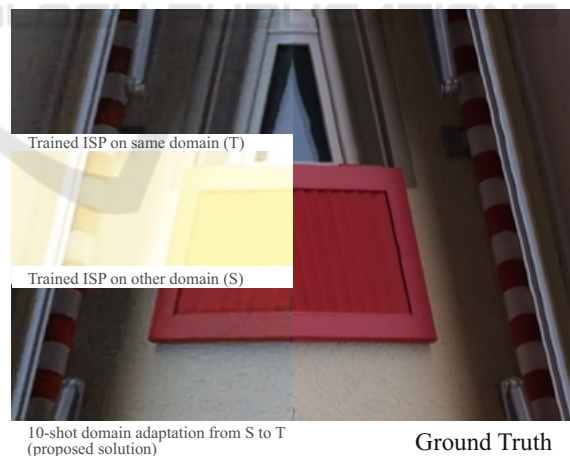




Figure 1: Visualization of the various ISP training approaches. Our few-shot domain adaptation approach is effective in RAW image enhancement tasks. Here are S is Zurich RAW-to-RGB dataset and T is Mobile AIM21.

samples should contain a wide variety of frames with different scene parameters, illumination levels, etc.

Another part of the problem is the strong sensitivity of convolutional neural networks to the dataset distribution (Deng et al., 2009). Differences in color

^a  <https://orcid.org/0009-0009-7176-6242>

^b  <https://orcid.org/0000-0001-6797-6284>

spaces, noise characteristics, and camera spectral sensitivities for different camera manufacturers create a huge domain gap between different sensors. Even more, the difference in color reproduction exists among the same sensor's instances, which is caused by the production inaccuracy. Consequently, a model trained on the raw data of one camera performs sub-optimally on the raw data of another camera (fig. 1). The quality of the currently existing photography pipeline drops significantly when the pipeline is applied directly to a new camera sensor. In addition, existing non-machine learning-based algorithms do not have an exact solution and depend on many free variables (Karaimer and Brown, 2016), which also complicates their usage.

Inspired by several unsupervised and few-shot domain adaptation approaches (Ganin and Lempitsky, 2015), (Ganin et al., 2016), (Prabhakar et al., 2023) in learning domain-invariant features we propose a paradigm shift for the raw image enhancement task using a few-shot domain adaptation to address the above-mentioned challenges. The primary distinguishing factors of our proposed method are its use of inverse gradient utilization, the employment of an AW-Net architecture as a base pipeline, and the utilization of marginally distinct loss functions to achieve state-of-the-art performance. In this work, we consider each camera as a separate domain. The main idea is to apply a domain adaptation method based on the reverse gradient to the U-Net-like deep learning-based image signal processing pipeline (Ronneberger et al., 2015). We improve the performance of our approach by using a large existing data collection from the source domain and transferring the task onto a new target domain with only 10 labeled samples. We anticipate that this domain adaptation method can make the development of the image processing pipelines easier (both for a new device and for different sensor instances of the same model), which can bring benefits to the related digital camera industries.

In summary, our contributions are as follows:

- We propose a new domain adaptation method for learning the image signal processing pipeline;
- We show that with a few labeled samples (10 images) from the target domain our approach can reach the comparable quality of the model trained from scratch with a complete target domain dataset;
- We present the results of the experiments illustrating the blazing performance of our method and compare it with other methods such as existing domain adaptation techniques, transfer learning, and projective transformation.

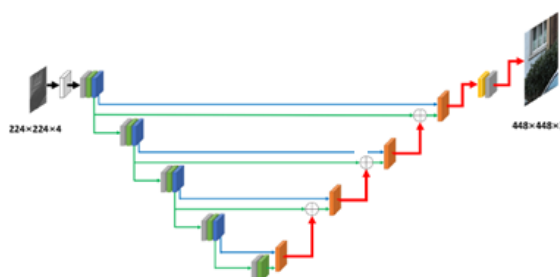


Figure 2: The main network architecture of the AWNet.

2 RELATED WORK

Domain Adaptation: Domain adaptation is a branch of transfer learning where the goal is to adapt a model learned on a source domain to perform with a high quality on another (target) domain. A popular practice for domain adaptation is to use the reverse gradient to obtain domain-invariant data representations. The pioneer of this practice was V. Lempitsky et al., authors propose to use unsupervised domain adaptation by backpropagation (DANN) (Ganin and Lempitsky, 2015) for a handwritten digit classifier. After that, a number of breakthrough papers have been published. The MADA technique (Pei et al., 2018), for instance, uses captured multi-mode structures to align various data distributions more effectively. There is also a few-shot (Motiian et al., 2017) domain adaptation work that uses a few labeled samples with many unlabeled samples in the target domain for image classification. Currently, mixed approaches are becoming increasingly common, including both elements of unsupervised and a few-shot domain adaptation (Shang et al., 2022), (Yue et al., 2021). The domain adaptation technique, which is gaining popularity, has found its application in RAW-to-RGB image signal processing. M. Afifi et al., used domain adaptation to learn RAW-to-RGB transformations between different cameras (Afifi and Abuolaim, 2021). An article of particular significance was also presented, pioneering the application of domain adaptation techniques to RAW-to-RGB image processing pipelines using only a few images (Prabhakar et al., 2023). The central premise of this research involves the employment of a common ISP pipeline to extract domain-invariant features.

Image Signal Processing Pipeline: The classic pipeline was described by M. Brown et al. (Ramanath et al., 2005). Such ISP often consists of many stages such as black level offset, normalization, bad pixel mask, demosaicing, white balance, noise reduction, color transform and etc. Each of these steps depends on a large number of parameters and may have a large

computational complexity, which complicates its usage.

Contrariwise, the recent increase in the processing capacity of smartphones and embedded devices has also contributed to the rise in popularity of deep learning for RAW-to-RGB image mapping during the past few years. This led to the appearance of various open RAW-to-RGB datasets, competitions, and scientific researches around deep learning-based ISP.

First RAW-to-RGB datasets were Samsung S7 ISP (Schwartz et al., 2018) and Zurich RAW-to-RGB (Ignatov et al., 2019). In 2021, the eponymous dataset was presented at the Mobile AIM challenge, one of the major image signal processing competition (Ignatov et al., 2021) which is running for 4 years now. In addition, Cube++ dataset made by E. Ershov et al. (Ershov et al., 2020) can also be used for learning ISP models.

There are also a number of publications devoted to neural-based ISP. In fact, nearly all of the proposed solutions are based on U-Net (Ronneberger et al., 2015): it is true as for the first approaches, namely proposed in 2018 DeepISP (Schwartz et al., 2018) and in 2019 PyNET (Ignatov et al., 2020b), as for the recently presented MW-ISPNet (Ignatov et al., 2020a) and AW-Net (Dai et al., 2020). A new state-of-the-art neural network called *dh.isp* was presented at Mobile AIM 2021 challenge (Ignatov et al., 2021).

Nowadays the two primary research goals are to improve image quality by discovering an effective network architecture and training method and to modify the network to perform within the computational limits of smartphones.

3 PROPOSED METHOD

We present an image processing pipeline based on AW-Net and determine the way of applying domain adaptation to it. In our approach, we combine the ideas of using a common ISP pipeline with separated pre-encoders (Prabhakar et al., 2023) and an inverse gradient (Ganin and Lempitsky, 2015) to obtain a domain-invariant representation. Since the output is an RGB image but the input is a RAW image, the network has to learn camera hardware-specific enhancement in addition to its entire ISP pipeline. The domain gap results from the fact that a model developed using data from a single camera (the source domain) does not perform similarly when applied to data from another camera (the target domain). Specifically, not only spectral sensitivities of the sensors usually are different but also the ISP itself. The situation is even more challenging because of the fact that ISP is usu-

ally a proprietary software and it is almost impossible to perform its reverse engineering. To cope with these limitations, we propose a domain adaptation method to move the ISP task from a large-scale labeled source dataset to a small set of labeled target data in order to produce output in the target domain.

Here and after we will denote a source dataset as a large enough for training neural-based ISP with a good quality, while the target domain will be considered as a small one (about 10 images). Our goal is to adapt a neural-based ISP pipeline train on a source domain to a target domain without noticeable quality loss. To achieve this goal we train our model to generate RGB images with both source and target domains as input. Our method is illustrated in Fig. 3 with the source and target training pipelines. It is an end-to-end trainable deep network that takes the RAW image as input and performs image reconstruction utilizing the source data for domain adaptation to the target domain using a few target-labeled samples.

3.1 Model Architecture

The neural network is a small AW-Net with additional blocks such as a domain classifier with a reverse gradient layer and pre-encoders for the possibility of applying domain adaptation to it. The description of each of the blocks is described below.

Pre-Encoders: The pre-encoder is a small convolutional network made up of three *Convolution2D* layers with 3x3 kernels and a number of filters - 8, 16, 32. Pre-encoders are needed in order to reduce the significant domain gap between different cameras by extracting individual and independent features from each of them. It takes a 4-channel GRGB RAW (before debayering) image as an input and produces a 32-channel output.

Common AW-Net: A lightweight U-Net-like autoencoder with 3 downsampling and 4 upsampling blocks. It takes a 32-channel image from each pre-encoder as input and produces two outputs: a 3-channel RGB image and a 256-dimensional vector from the bottleneck.

Domain Classifier: To reduce the gap between domains and increase the performance we add a binary domain classifier (Ganin et al., 2016) with an inverse gradient (Ganin and Lempitsky, 2015) using a convolutional neural network with *GlobalAveragePooling2D* and two *Dense* layers at the end. It takes a 256-dimensional vector as input from AW-Net.

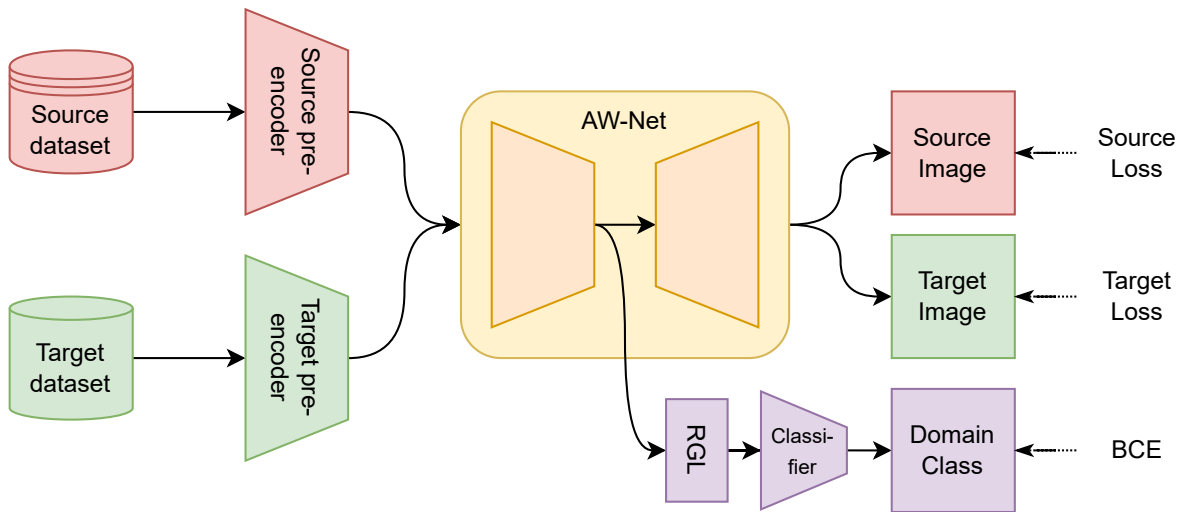


Figure 3: Illustration of the proposed few-shot domain adaptation approach. We use separate pre-encoders from each domain to extract camera-specific features. We use AW-Net as a common ISP pipeline and domain classifier with reverse gradient layer (RGL) to midimase domain gap and learn domain-invariant features.

3.2 Training Process

To achieve a good performance of domain adaptation, the network is trained in two stages:

1. Perform pipeline pre-training on the source domain. At this stage, we use only the source domain pre-encoder and AW-Net’s RGB output; the outputs of the domain classifier are not considered and the target domain pre-encoder is not used.
2. Initialize the initial weights of the target pre-encoder by the weights from the source pre-encoder to preserve structural integrity while transmitting the signal through the pre-encoder specific to the target domain.
3. Start the domain adaptation stage training the whole network and using the entire source domain and a small part of the target domain. During this stage at each training step, we sequentially feed image crops from the target and source domains to the corresponding pre-encoder and calculate the corresponding loss functions. In addition, we take into account the predictions of the domain classifier and its inverse gradient.

3.3 Loss Functions

Within this section, we expound upon the loss functions pertinent to both the pre-training and domain adaptation phases. It is imperative to underscore that all the enumerated loss functions are composed as amalgamations encompassing perceptual loss, L_1 loss, MS-SSIM loss, as well as additional components, namely, color loss and exposure fusion loss,

the latter of which will be elucidated in subsequent discussions. The nomenclature employed herein designates the predicted image as \hat{I} and the ground truth RGB image as I .

Perceptual Loss. To mitigate pixel misalignment discrepancies, we employ the perceptual loss derived from the output of the pre-trained VGG19 network (Simonyan and Zisserman, 2014). The loss function is defined as follows:

$$L^{vgg} = L_2(VGG(I) - VGG(\hat{I})). \quad (1)$$

Here, VGG represents the output from the final convolutional layer of the pre-trained VGG-19 network, and L_2 denotes the mean squared error, facilitating the minimization of discrepancies between the reconstructed image (\hat{I}) and the ground truth image (I). This approach effectively accounts for perceptual qualities and pixel-level fidelity.

L_1 Loss. We use L_1 loss as strong supervision to optimize pixel values during the training of the network. We also do not use this loss during training the target domain on a few data samples to avoid overfitting the neural network.

MS-SSIM Loss. The multi-scale structural similarity loss L_{MSSSIM} is used to enhance the reconstructed RGB images by the structural similarity index. The loss function can be defined as:

$$L^{MSSSIM} = 1 - MSSSIM(I, \hat{I}), \quad (2)$$

where $MSSSIM$ is a multi-scale structural similarity (Wang et al., 2003). This approach facilitates the preservation of structural characteristics and perceptual quality in the reconstructed images.

Color Loss. This loss is measured as the cosine similarity between the RGB vectors to minimize the color difference between the predicted image and the ground truth. We denote it as L^{rgb} .

Exposure Fusion Loss. Exposure fusion technique (Mertens et al., 2009) is used for fusing a bracketed exposure sequence into a high-quality image, without converting to HDR first. Exposure fusion computes a target image by identifying the best parts of multiple exposures. We use a set of quality measures to guide the process, which we consolidate into a scalar-valued weight map. The exposure fusion loss function minimizes the difference of these maps between the predicted and the ground truth images, which helps to build a more accurate exposure, avoiding overexposure and darkening. The loss function can be defined as:

$$L^{exp} = L_1(Exp(I), Exp(\hat{I})), \quad (3)$$

where Exp is an exposure fusion technique.

Pre-Training Stage. In the beginning, we pre-train the pipeline using only source domain data. At this stage, we use the following loss function:

$$L^{pretrain} = L_1 + L^{vgg} + L^{MSSSIM} + L^{rgb} + L^{exp}. \quad (4)$$

Domain Adaptation Stage. In the second stage, we make domain adaptation using source and target domain data together. In this case, we minimize three losses:

- Loss for source domain: $L^{source} = L_1$;
- Loss for target domain: $L^{target} = L^{vgg} + L^{MSSSIM} + L^{rgb} + L^{exp}$;
- Loss for domain classifier: $L^{classifier} = BCE$ (binary cross-entropy).

4 EXPERIMENTS

We used TensorFlow 2.9.0 and Python 3.9 to implement the proposed neural network and then trained the model with the following server environment: Ubuntu 21.10, AMD Ryzen 7 5800X, 64G RAM, and NVIDIA GeForce RTX 2080 Ti GPU x1. It should be noted that each experiment was run 5 times and quality measurements were averaged. We also used *Adam* optimiser with default parameters: $lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.9$, $\epsilon = 1 \cdot 10^{-7}$.

4.1 Datasets

In our experimentation involving domain adaptation, we employed three open datasets: the Zurich RAW-to-RGB dataset, the Samsung S7 ISP dataset, and the Mobile AIM21 dataset. It is essential to underscore

that each training experiment for domain adaptation incorporated 10 pairs of RAW-RGB images, each at full resolution. A comprehensive description of each of these datasets can be found in Table 1. Furthermore, we conducted color checker photographs using two distinct devices to establish a projective transformation between the color spaces of their respective cameras.

Table 1: The datasets comparison.

Dataset	Crops per image	Train size (Crops)	Domain adaptation size (Crops)	Validation size (Crops)
Zurich RAW-to-RGB	60	32000	600	8043
Samsung S7 ISP	46	4000	460	1060
Mobile AIM21	700	19200	7000	4961

Zurich RAW-to-RGB Dataset. This dataset was submitted for the Mobile AIM19 challenge in 2021 and is the largest dataset to date. The dataset contains more than 20K pairs of outdoor images taken synchronously by a Canon 5D Mark IV DSLR camera and a Huawei P20 smartphone with a Sony IMX380 mobile sensor (12M pixel) capturing images in the RAW format. The images were taken during the daytime in a wide variety of places and in various illumination and weather conditions. Since training deep learning models on high-resolution images is infeasible, the patches of size 224×224 pixels were extracted from the P20-RAW / Canon image pairs preliminary aligned. As a result, 48043 crops were selected, where about 60 patches come to one full-size image.

Samsung S7 ISP Dataset. This dataset consists of the RAW and JPEG image pairs captured using the Samsung S7 smartphone. For each scene, both normally lit and low light are captured (low light is simulated by shorter exposure). A total of 110 scenes are captured in full resolution (12M pixels). For our experiments, images with standard lighting were selected that were cut into crops of 512×512 without overlap.

Mobile AIM21 Dataset. The dataset was generated using a Sony IMX586 quad Bayer mobile sensor (48M pixel), and a Fujifilm GFX100 DSLR. Since the captured RAW-RGB image pairs are not perfectly aligned, they were matched using an advanced dense correspondence algorithm (Truong et al., 2021), and

then smaller patches of size 256×256 pixels were extracted. We obtained 24K training RAW-RGB image pairs 256×256 , where about 700 patches come to one full-size image.

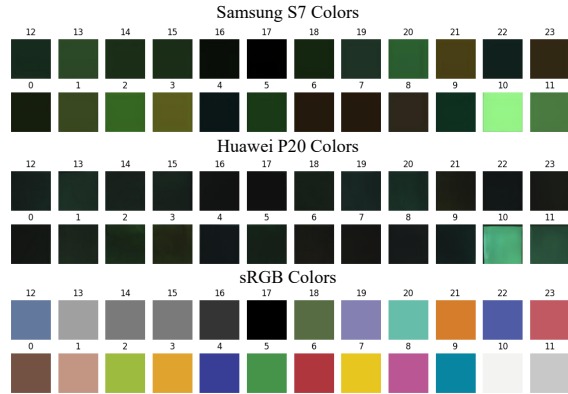


Figure 4: Visualization of color pairs from ColorChecker and sRGB representations. A set of 24 color pairs was obtained by capturing images with Samsung S7 and Huawei P20 devices under standardized D50 lighting conditions.

ColorChecker Dataset. Data acquisition involved the deployment of two devices, namely the Samsung S7 and Huawei P20, to capture a scene featuring a ColorChecker under D50 illumination conditions. These specific devices align with Samsung S7 ISP and Zurich RAW-to-RGB datasets. Subsequently, the target color information was extracted from each RAW image for every color patch on the ColorChecker. This process yielded a total of 24 pairs of colors, represented in the RGGb format as 4-dimensional vectors.

4.2 Training Description and Results

In the investigation of domain adaptation, we harnessed a triad of datasets, such as Zurich RAW-to-RGB, Samsung S7 ISP, and Mobile AIM21, in diverse source and target domain combinations. Furthermore, for the sake of comparative analysis with established methodologies, such as the color space transform (CST), transfer learning, and Prabhakar’s domain adaptation (PDA) approach (Prabhakar et al., 2023), we selected the Samsung S7 ISP dataset as the source domain and the Zurich RAW-to-RGB dataset as the target domain (Table 4, Figure 7). It is imperative to highlight that, within the framework of our experiments, we restricted the utilization of merely ten images for the target domain.

Pre-Training. Our initial phase entails pipeline pre-training on the source domain data. During this stage, exclusively the source domain pre-encoder and AW-Net RGB output are utilized. Training of the neural network commences from a pristine state, encompass-

ing the entire dataset sourced from Zurich RAW-to-RGB, Mobile AIM21, and Samsung S7 ISP, spanning four training epochs. This pre-training endeavor culminated in our model’s attainment of noteworthy outcomes, as elaborated in Table 2, and visually represented in Figure 6.

Domain Adaptation. Subsequently, we embarked on domain adaptation by exploring all feasible dataset combinations, resulting in six distinct instances. In this phase, a modest subset of ten images, each at full resolution, was exclusively employed. The neural network underwent a two-epoch training process, resulting in the outcomes detailed in Table 2 and visually depicted in Figure 5. Notably, our approach to domain adaptation demonstrated a commendable performance, achieving a mere 2% reduction in efficacy compared to learning from scratch.

Furthermore, we engaged in a detailed comparative study to evaluate the impact of different combinations of loss function components. The loss function under scrutiny was defined as $L^{\text{target}} = L^{\text{vgg}} + L^{\text{MSSSIM}} + L^{\text{rgb}} + L^{\text{exp}}$. This analysis aimed to elucidate the individual contribution of each term to the overall performance. Additionally, our methodology was juxtaposed with existing domain adaptation techniques, including Color Space Transform (CST), Transfer Learning, and Prabhakar Domain Adaptation (PDA) as delineated in (Prabhakar et al., 2023). The comparative results of these techniques are systematically presented in Table 4 and Figure 7. The empirical evidence substantiates that our proposed approach outperforms other domain adaptation methodologies in terms of effectiveness.

Table 2: The validation scores (PSNR and SSIM), were computed across various datasets. The diagonal of the matrix represents scores when learning from scratch. Our proposed 10-shot domain adaptation method exhibits notably comparable performance, with only a marginal 2% drop in performance compared to the learned from scratch in the whole dataset baseline.

Domain Adaptation		Target Domain		
		Zurich RAW-to-RGB	Mobile AIM21	Samsung S7 ISP
Source Domain	Zurich RAW-to-RGB	19.46, 0.73	23.15, 0.86	22.07, 0.79
	Mobile AIM21	18.92, 0.71	23.48, 0.87	22.03, 0.79
	Samsung S7 ISP	18.85, 0.71	23.08, 0.85	22.16, 0.81

Furthermore, we conducted experiments to evaluate domain adaptation from Zurich RAW-to-RGB to Mobile AIM21 using varying numbers of images: 1,



Figure 5: Predictions visualization for Domain adaptation from source domains to Zurich RAW-to-RGB and Mobile AIM21. Training from scratch – an AW-Net ISP pipeline trained from scratch on the corresponding dataset (Mobile AIM21 on top, Zurich RAW-to-RGB on bottom). Zurich to AIM21, AIM21 to Zurich, etc. – a demonstration of the work of our domain adaptation approach on 10 images for the corresponding datasets.

5, 10, 20, 40, and 80 (Table 3, Figure 6). The findings indicate that our current approach yields near state-of-the-art quality, specifically, with five images, the quality decrement is 8%, surpassing the efficacy of conventional transfer learning, and for a single image, it stands at 19%. Performance close to trained from scratch ISP is achieved with 10 images. A further increase in the number of images does not provide a significant increase in performance.

Table 3: Validation scores (using the validation set) for domain adaptation from Zurich RAW-to-RGB to Mobile AIM21 using $k = 1, 5, 10, 20, 40, 80$ images.

k images	PSNR \uparrow	SSIM \uparrow
1	19.05	0.77
5	21.70	0.79
10	23.15	0.86
20	23.31	0.86
40	23.39	0.87
80	23.43	0.87

Color Space Transform (CST). In addressing the fundamental challenge of domain transfer, we initially explored a rudimentary yet pragmatic solution, involving color space transformations from the Zurich RAW-to-RGB domain to the Samsung S7 ISP domain. To implement this approach, we conducted the linear regression training with polynomial features of

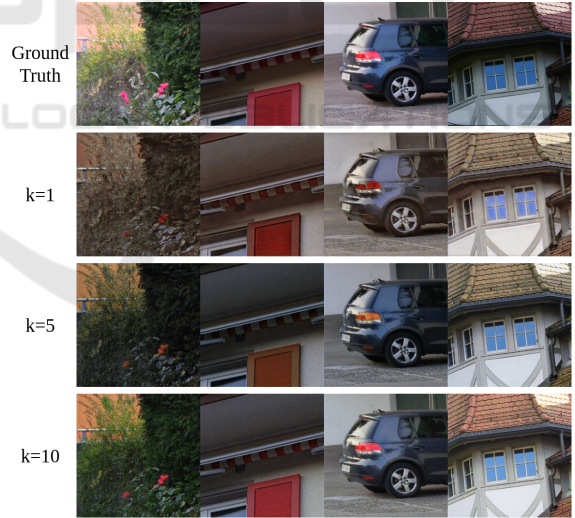


Figure 6: Visualisation of predictions for domain adaptation from Zurich RAW-to-RGB to Mobile AIM21 using $k = 1, 5, 10$ images.

the third degree, extracted from our ColorChecker dataset corresponding to each camera:

$$P_{3,3} = \{r, g, b\} \cup \{r^2, g^2, b^2, rg, rb, gr\} \cup \{r^3, g^3, b^3, r^2g, r^2b, g^2r, g^2b, b^2r, b^2g, rgb\} \quad (5)$$

The selection of the polynomial degree was determined through a systematic grid-search proce-

ture. Subsequently, we applied the trained regression model to transform the RAW images within the Zurich RAW-to-RGB dataset. Following this transformation, we employed the image processing pipeline, which had been trained from scratch on the Samsung S7 ISP dataset, to process the transformed data.

Transfer Learning. To facilitate comparative analysis, we pursued the strategy of transferring learning acquired by a pre-trained model from the Samsung S7 ISP domain to the Zurich RAW-to-RGB domain. The training process for this transfer involved two epochs and encompassed the complete dataset. The same loss function employed during training from scratch was used for transfer learning.

Prabhakar Domain Adaptation (PDA). In our evaluation, we conducted a comparative analysis between our proposed approach and what, to the best of our knowledge, stands as the sole third-party method outlined in (Prabhakar et al., 2023). The primary distinguishing factors of our proposed method are its use of inverse gradient utilization, the employment of an AW-Net architecture instead of U-Net, and the utilization of marginally distinct loss functions.

To rigorously assess this adaptation approach, referred to as PDA, we conducted experiments involving 10 images sourced from the Samsung S7 ISP domain, adapted to the Zurich RAW-to-RGB domain. The comparative evaluation reveals that, in comparison to our approach, the PDA method exhibits a slightly inferior performance, as delineated in Table 4 and Figure 7.

Table 4: Validation scores (using the validation set) for domain adaptation approaches from Samsung S7 ISP to Zurich RAW-to-RGB. Our domain adaptation approach with only ten images showcases superior performance compared to existing methods and achieves performance comparable to training from scratch on the entire dataset.

Method	PSNR \uparrow	SSIM \uparrow
Learning from scratch	19.46	0.73
Domain adaptation (ours)	18.85	0.71
Domain adaptation (ours) $L^{VGG} + L^{MSSIM} + L^{rgb}$	18.02	0.70
Domain adaptation (ours) $L^{VGG} + L^{MSSIM}$	17.21	0.67
CST	15.16	0.73
Transfer learning	16.74	0.67
PDA	17.12	0.69

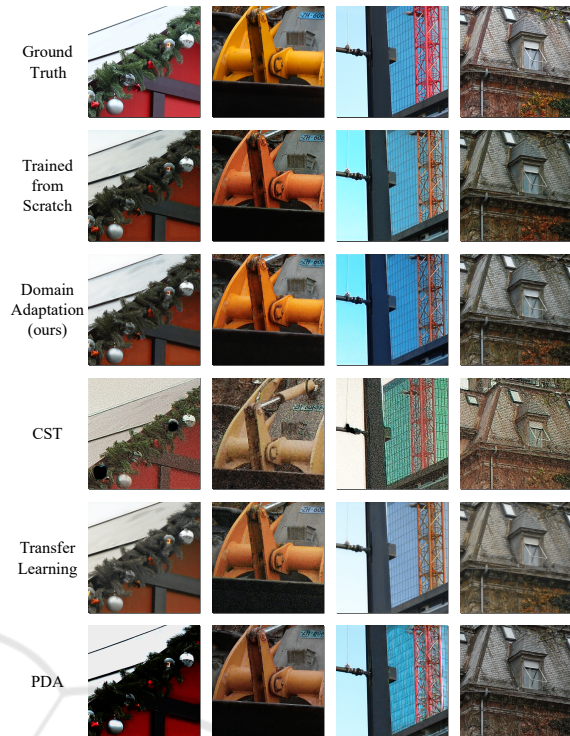


Figure 7: Visualisation of predictions for domain adaptation of different approaches from Samsung S7 ISP to Zurich RAW-to-RGB. Our approach has the best performance that is closest to the result of an ISP trained from scratch.

5 CONCLUSION

Using only a small number of labeled samples from the target domain and a large number of samples from the source domain, we demonstrated the SoTA (state-of-the-art) domain adaptation approach for the RAW to RGB image signal processing pipeline that can be very useful for manufacturers of digital cameras and smartphones, as it can significantly reduce financial and time production costs. We first obtain camera-specific information using pre-encoders, followed by domain invariant characteristics that are extracted using the AW-Net network. We apply domain adaptation using the back-propagation approach to decrease the domain gap. Our findings demonstrate that, compared to training with huge target domain data, using our approach with even very few (about a dozen) labeled samples from the target domain is enough to provide a comparable performance level. We believe that our approach will stimulate more explorations in these fields and will be applied in the production of digital cameras.

REFERENCES

- Affi, M. and Abuolaim, A. (2021). Semi-supervised raw-to-raw mapping. *arXiv preprint arXiv:2106.13883*.
- Dai, L., Liu, X., Li, C., and Chen, J. (2020). Awnet: Attentive wavelet network for image isp. In *European Conference on Computer Vision*, pages 185–201. Springer.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Ershov, E., Savchik, A., Semenov, I., Banić, N., Belokopytov, A., Senshina, D., Koščević, K., Subašić, M., and Lončarić, S. (2020). The cube++ illumination estimation dataset. *IEEE Access*, 8:227511–227527.
- Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.
- Ignatov, A., Chiang, C.-M., Kuo, H.-K., Sycheva, A., and Timofte, R. (2021). Learned smartphone isp on mobile npus with deep learning, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2503–2514.
- 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. (2019). Aim 2019 challenge on raw to rgb mapping: Methods and results. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3584–3590. IEEE.
- Ignatov, A., Timofte, R., Zhang, Z., Liu, M., Wang, H., Zuo, W., Zhang, J., Zhang, R., Peng, Z., Ren, S., et al. (2020a). Aim 2020 challenge on learned image signal processing pipeline. In *European Conference on Computer Vision*, pages 152–170. Springer.
- Ignatov, A., Van Gool, L., and Timofte, R. (2020b). Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 536–537.
- Karaimer, H. C. and Brown, M. S. (2016). A software platform for manipulating the camera imaging pipeline. In *European Conference on Computer Vision*, pages 429–444. Springer.
- Medioni, G. and Dickinson, S. (2016). Synthesis lectures on computer vision.
- Mertens, T., Kautz, J., and Van Reeth, F. (2009). Exposure fusion: A simple and practical alternative to high dynamic range photography. In *Computer graphics forum*, volume 28, pages 161–171. Wiley Online Library.
- Motiian, S., Jones, Q., Iranmanesh, S., and Doretto, G. (2017). Few-shot adversarial domain adaptation. *Advances in neural information processing systems*, 30.
- Pei, Z., Cao, Z., Long, M., and Wang, J. (2018). Multi-adversarial domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*.
- Prabhakar, K. R., Vinod, V., Sahoo, N. R., and Babu, R. V. (2023). Few-shot domain adaptation for low light raw image enhancement. *arXiv preprint arXiv:2303.15528*.
- Ramanath, R., Snyder, W. E., Yoo, Y., and Drew, M. S. (2005). Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- Schwartz, E., Giryes, R., and Bronstein, A. M. (2018). Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923.
- Shang, J., Niu, C., Huang, J., Zhou, Z., Yang, J., Xu, S., and Yang, L. (2022). Few-shot domain adaptation through compensation-guided progressive alignment and bias reduction. *Applied Intelligence*, pages 1–17.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Truong, P., Danelljan, M., Van Gool, L., and Timofte, R. (2021). Learning accurate dense correspondences and when to trust them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5714–5724.
- Wang, Z., Simoncelli, E. P., and Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee.
- Yue, X., Zheng, Z., Zhang, S., Gao, Y., Darrell, T., Keutzer, K., and Vincentelli, A. S. (2021). Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13834–13844.