This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learned Lightweight Smartphone ISP with Unpaired Data

Andrei Arhire
Faculty of Computer Science
Alexandru Ioan Cuza University of Iasi
andrei.arhire@student.uaic.ro

Radu Timofte
Computer Vision Lab, CAIDAS & IFI
University of Würzburg
radu.timofte@uni-wuerzburg.de

Abstract

The Image Signal Processor (ISP) is a fundamental component in modern smartphone cameras responsible for conversion of RAW sensor image data to RGB images with a strong focus on perceptual quality. Recent work highlights the potential of deep learning approaches and their ability to capture details with a quality increasingly close to that of professional cameras. A difficult and costly step when developing a learned ISP is the acquisition of pixel-wise aligned paired data that maps the raw captured by a smartphone camera sensor to high-quality reference images.

In this work, we address this challenge by proposing a novel training method for a learnable ISP that eliminates the need for direct correspondences between raw images and ground-truth data with matching content. Our unpaired approach employs a multi-term loss function guided by adversarial training with multiple discriminators processing feature maps from pre-trained networks to maintain content structure while learning color and texture characteristics from the target RGB dataset. Using lightweight neural network architectures suitable for mobile devices as backbones, we evaluated our method on the Zurich RAW to RGB and Fujifilm UltraISP datasets. Compared to paired training methods, our unpaired learning strategy shows strong potential and achieves high fidelity across multiple evaluation metrics. The code and pre-trained models are available at https://github.com/AndreiiArhire/Learned-Lightweight-Smartphone-ISP-with-Unpaired-Data.

1. Introduction

The transformation steps required for the image signal to pass through, starting from the CMOS sensor readings, on its way to reaching the refined RGB image seen by the users, are entirely handled by the Image Signal Processor (ISP). Several of such processing stages include denoising, demosaicing, color consistency, gamma correction, and compression. In traditional ISPs, these steps are handcrafted and applied sequentially. Consequently, they propagate a small error through the processing chain, leading to a degradation of the results.

Multiple tasks performed by ISP have been addressed individually through deep learning with great outcomes. In recent years, the idea of creating a deep neural network capable of outperforming conventional ISP has received growing attention [17, 20]. Considering the trade-off between latency and performance, developing a learned ISP intended to run on edge devices has been the subject of various challenges and currently represents an active field of research.

A learnable ISP has the ability to partially overcome specific constraints, such as a small sensor and limited optical system, reducing the perceptual gap between smartphone cameras and professional DSLRs. To get the best results, the model is expected to be trained using paired pixel-wise aligned data. In practice, such data sets are difficult to obtain and must be collected individually for each new camera, since the characteristics of one camera directly impact the raw data. A recent solution to this challenge is Rawformer [33], which proposes a state-of-the-art unsupervised method to translate the raw training data set from a specific camera domain to the target camera domain. However, a learned ISP with unpaired data could potentially provide a more accurate representation of the ground truth as it works directly with the original data.

Inspired by the WESPE work of Ignatov *et al*. [14], we introduce an unpaired learning approach to train a learnable ISP. To ensure minimal latency on edge devices, our experiments primarily use the model architecture developed by the winner of the Mobile AI & AIM 2022 Learned Smartphone ISP Challenge [21]. In our proposed pipeline, the model is trained using a multi-term loss function with dedicated components for content, color, and texture. To capture various characteristics of the statistical distribution of the target dataset, three discriminators are used during training. Guided by relativistic adversarial losses, the model learns to enhance color fidelity and perceptual realism, while structural consistency is preserved through a self-supervised loss. Our approach is evaluated on two real-world RAW-to-

RGB datasets, Zurich RAW-to-RGB [17] and Fujifilm UltraISP [20]. The generated images achieve fidelity scores (PSNR, SSIM [39], MS-SSIM [38]) comparable to those obtained through paired training, while maintaining a favorable perceptual quality (LPIPS score [40]).

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the proposed methods. Section 4 describes the experimental setup and the achieved results, while the conclusions are drawn in Section 5.

2. Related Work

A learnable ISP is trained to translate the image from the RAW format to an RGB domain with superior visual quality, refined for human perception. Typically, this is achieved by training on paired and pixel-wise aligned image patches using RAW data from a particular camera sensor and images from a high-quality DSLR camera.

PyNET [17] is one of the first learnable ISPs that achieves the efficiency of the Huawei P20 commercial pipeline ISP and obtained superior results in the evaluation of mean opinion scores (MOS). It relies on an inverse pyramidal CNN architecture, which processes input on different scales. On each scale, specific features are learned with dedicated loss functions. In follow-up research, PyNET-CA [25] improves performance by incorporating a channel-attention mechanism. Mobile-suitable variants, including PyNET-v2 [19] and MicroISP [20] have been developed for real-time execution on devices constrained by their hardware resources. The advancement in the field has been encouraged by competitions, including the Learned Smartphone ISP Challenge, part of Mobile AI (MAI) workshops in conjunction with CVPR 2021 [18] and CVPR 2022 [21].

During the aforementioned challenges, teams were invited to submit their models and address two tracks. The solutions were designed to optimize the trade-off between runtime and fidelity (measured by PSNR) in the first track, while the second track has focus on perceptual quality and is evaluated with MOS. A valuable software used by participants, the AI Benchmark application [15] provides an environment to test Tensorflow Lite models on Android smartphones, taking advantage of the supported acceleration options.

In both editions, compact networks (Fig. 1) with three convolutional layers followed by a pixel-shuffle layer achieved the highest score according to the formula adopted for track one, as they provided the best balance between processing speed and output quality.

CSANet [11] obtained the best results in terms of both PSNR and SSIM in the MAI 2021 challenge. At the core of the network are two double attention modules with skip connections able to learn the spatial and channel dependencies from feature maps. The desired output size of the

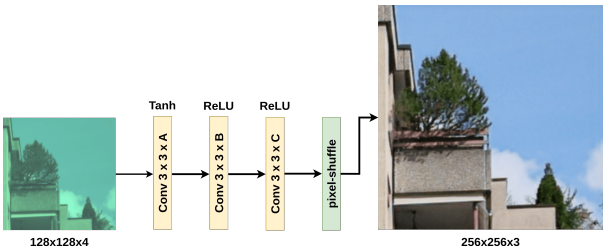


Figure 1. Overview of the winning architectures from the MAI 2021 and 2022 challenges. The dh_isp team [18] uses channel sizes [16, 16, 16], while MiAlgo [21] uses [12, 12, 12].

image is obtained by further use of transpose convolution and depth-to-space layer. LAN [34] is built on this work and increases performance by introducing several improvements to the original architecture. A strided convolutional layer is applied instead of space-to-depth operation at the beginning to improve sharpness and lower GPU latency on smartphones. High-frequency details are better preserved through the implementation of a high-level skip connection via concatenation. Other differences include model pretraining with classical demosaicing, a custom loss function composed of multiple components, and adjustments on the selection of activation functions. Attention mechanisms have also shown notable performance when integrated into U-Net-like architectures that incorporate discrete wavelet transforms (DWT) to emphasize the representation of fine-grained structures, as demonstrated by MW-ISPNet [16] and AWNet [5].

RMFA-Net [29] is a recently proposed network architecture that achieves state-of-the-art (sota) image quality on the Fujifilm UltraISP data set [20]. The architecture consists of an input module composed of two convolutional layers, followed by a stack of RMFA blocks, and an output module implemented as a single convolutional layer. Each RMFA block includes a texture module that processes both high-frequency and low-frequency textures, a tone mapping module based on Retinex Theory [28], a spatial attention module, and a channel attention module. Each block also incorporates a skip connection to prevent information loss. In the preprocessing phase, the authors propose a three-channel split and black level subtraction, which play a substantial role in model performance.

Generative Adversarial Networks (GANs) have demonstrated strong capabilities in transferring feature representations across different domains. Reference methods such as CycleGAN [41], UNIT [30], and U-GAT-IT [26] employ cycle consistency constraints, allowing unpaired image-to-image translation. Dedicated discriminators have been used successfully in prior works such as [13] and [14] to guide the learning of color and texture representations in the context of image enhancement. Various GAN objectives have

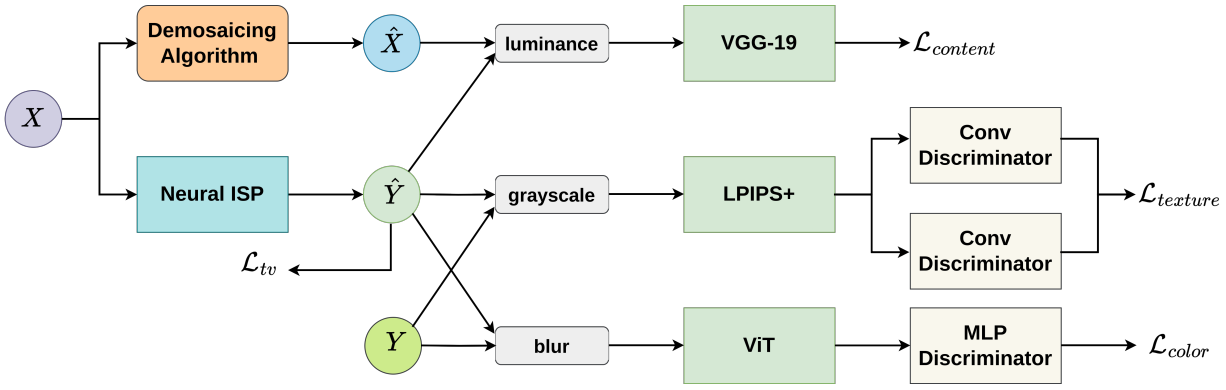


Figure 2. Overview of our proposed unpaired training method.

been explored to stabilize training and improve the quality of results, starting from the original adversarial loss [9], to the Wasserstein loss [1] with gradient penalty [10], the relativistic GAN loss [24], and more recently the regularized relativistic GAN loss (R3GAN) [12]. Researchers commonly design neural networks for ISP learning using multi-term loss functions, where each term targets a specific attribute. Among the frequently used loss functions for capturing pixel-level differences are L1 and L2 losses, along with more robust variants such as Huber and Charbonnier, which are aimed at reducing sensitivity to outliers. SSIM and MS-SSIM are widely used to assess structural similarity based on local image patterns. To improve color fidelity, a common strategy is to first apply Gaussian blur to reduce the influence of textures, followed by measuring color differences in a suitable color space. DISTS [6] combines deep features with structural information for a more perceptually aligned measure. Perceptual similarity is often measured using feature activations [23] from pretrained networks such as VGG-19 [35]. LPIPS [40] builds on this idea by computing distances in a deep feature space, fine-tuned to match human perceptual preferences. Our work builds upon several of these definitions, with a central focus on maximizing perceptual image quality.

3. Proposed Method

Although we prioritize fast inference, the training process is not limited by computational cost. Therefore, we incorporate additional networks for adversarial learning and feature extraction (Fig. 2). Alongside the unpaired training strategy, we define a method that uses paired data, serving as an upper bound for our unpaired approach.

The color, texture, and content attributes are captured using dedicated loss functions. For each, we utilize feature embeddings extracted from pre-trained networks during loss computation, rather than relying directly on pixel-wise differences of the generated images. Additionally, a total variation (TV) loss is applied to promote spatial smoothness and reduce visual artifacts. A detailed description of each loss function used in our experiments is provided in the following sections.

3.1. Content Loss

Following the approach in WESPE [14], structural consistency is enforced by computing the Mean Squared Error (MSE) between the feature maps of the generated image and the corresponding reference. These feature maps are extracted from the relu_5_4 layer of the VGG-19 network.

$$\mathcal{L}_{\text{content}} = \frac{1}{CHW} \sum_{i,j,k} (F_{ijk}^{\text{relu5_4}}(I_1) - F_{ijk}^{\text{relu5_4}}(I_2))^2 \quad (1)$$

In the paired setting, the generated image and its corresponding ground truth are directly passed into VGG-19. In the unpaired setting, the reference is obtained by applying a specialized demosaicing algorithm to the RAW input. The generated and reference RGB images are then converted to the LAB color space. Only the L (luminance) channel is retained and replicated across all three channels to match the input format required by VGG-19. We denote this loss as $\mathcal{L}_{\text{content (paired)}}$ or $\mathcal{L}_{\text{content (unpaired)}}$, depending on the data access type.

3.2. Paired Color Loss

As introduced in DPED [13], a Gaussian kernel is applied to the images prior to computing the MSE to better quantify the color discrepancies.

$$\mathcal{L}_{\text{color}} = \frac{1}{N} \sum_{i,j,k} (B(I_1)_{ijk} - B(I_2)_{ijk})^2 \quad (2)$$

$$B(I)_{ijk} = \sum_{m,n} G(m,n) \cdot I_{i,j+m,k+n} \quad (3)$$

$$G(m,n) = A \exp\left(-\frac{(m-\mu_x)^2}{2\sigma_x^2} - \frac{(n-\mu_y)^2}{2\sigma_y^2}\right) \quad (4)$$

This approach reduces the influence of fine textures and has been shown to improve contrast and brightness while preserving color fidelity. The resulting loss is referred to as $\mathcal{L}_{\text{color}}$ in the total loss calculation. Moreover, it is tolerant to small pixel misalignments and has been adopted by recently developed sota methods [29].

3.3. Paired Texture Loss

We integrate LPIPS+ [3] together with DISTS [6] as loss components ($\mathcal{L}_{\text{LPIPS+}}$, $\mathcal{L}_{\text{DISTS}}$) responsible for texture and perceptual learning using paired input and ground truth images.

LPIPS (Learned Perceptual Image Patch Similarity) measures perceptual similarity by comparing deep features of two images across multiple layers with fine-tuned weights calibrated to match human visual perception.

LPIPS+ extends LPIPS by using reference image features as semantic weights in a weighted average pooling scheme. This focuses the metric on important semantic regions, resulting in perceptual quality assessments that better align with human judgments.

DISTS assesses the perceptual quality of an image by combining structure similarity, calculated using normalized correlation between corresponding feature maps, and texture similarity, calculated using normalized similarity between their spatial means, across multiple layers of a modified pre-trained VGG [35] network.

3.4. Relativistic Adversarial Color Loss

$$R_1 = \frac{\gamma}{2}\mathbb{E}_{x_r \sim P}\left[\|\nabla_{x_r} D(x_r)\|^2\right] \quad (5)$$

$$R_2 = \frac{\gamma}{2}\mathbb{E}_{x_f \sim Q}\left[\|\nabla_{x_f} D(x_f)\|^2\right] \quad (6)$$

$$L_D = \mathbb{E}_{x_r, x_f}\left[\mathrm{f}(-(D(x_r) - D(x_f)))\right] + R_1 + R_2 \quad (7)$$

$$L_G = \mathbb{E}_{x_r, x_f}\left[\mathrm{f}(-(D(x_f) - D(x_r)))\right] \quad (8)$$

Unpaired coloring is learned in an adversarial manner. We adopt a relativistic loss [24] with zero-centered gradient penalties [12]. The real and generated images are initially fed into a pre-trained ViT-base-patch16-224 model [8]. The feature embeddings from the last hidden state of the transformer (excluding the CLS token) are then fed into the color discriminator to predict the realism of the given colors. This adversarial objective constitutes the $\mathcal{L}_{\text{adv (color)}}$ loss term. The color discriminator consists of a three-layer MLP with ReLU activations, followed by a mean pooling to aggregate the final prediction (Fig. 3).

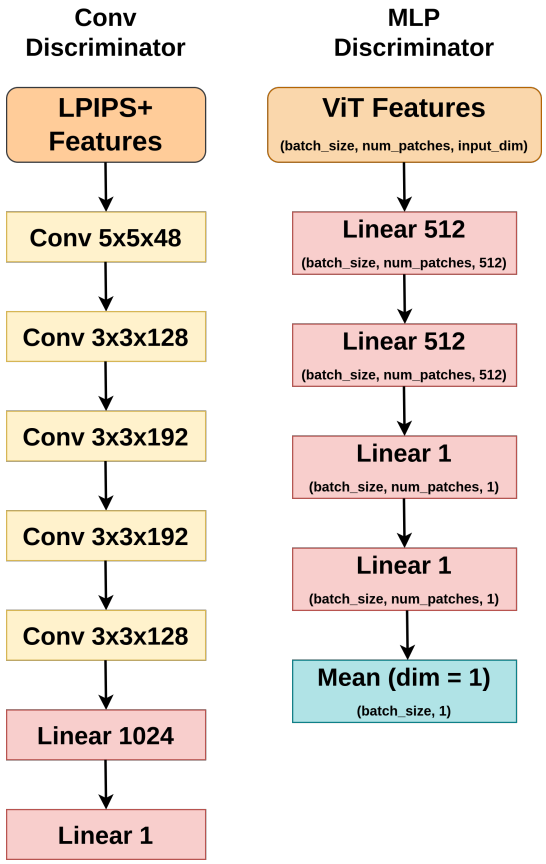


Figure 3. Architecture of the discriminators presented in Fig. 2.

3.5. Relativistic Adversarial Texture Loss

The loss formulas are the same as in the unpaired color loss Eq. (7), Eq. (8). Since the LPIPS (and variants) metric is responsible for evaluating perceptual quality, we choose to first convert generated and real images to grayscale and feed them to the LPIPS+ network.

Then, two discriminators process different levels of LPIPS+ features to evaluate realism from distinct perspectives. In the LPIPS+ architecture, lin0 and lin3 refer to linear layers that process features extracted from different depths of the backbone (AlexNet [27] in IQA-PyTorch [2] implementation) - lin0 processes features from the first convolutional block (64 channels) capturing low-level details such as edges and sharpness, while lin3 processes features from the fourth block (256 channels) representing more complex patterns, which emphasizes higher-level perceptual quality. The first discriminator receives features from the lin0 layer and contributes with the adversarial loss term $\mathcal{L}_{\text{adv (lin0)}}$, while the second receives lin3 features and corresponds to $\mathcal{L}_{\text{adv (lin3)}}$. Textural discriminators have a CNN architecture adapted from [13] with five convolutional layers followed by two fully connected layers, using Leaky ReLU activations and progressive downsampling of input (Fig. 3).

3.6. Total Variation Loss

This loss ($\mathcal{L}_{\text{TV}}$) penalizes differences between adjacent pixels, promoting spatial smoothness in the generated images. It plays an important complementary role to content loss, which effectively preserves high-level structures but often fails to capture fine details.

$$L_{\text{TV}} = \frac{1}{N}\left(\frac{\sum_{i,j}(I_{i+1,j} - I_{i,j})^2}{H(W-1)C} + \frac{\sum_{i,j}(I_{i,j+1} - I_{i,j})^2}{(H-1)WC}\right) \tag{9}$$

If the weight of this loss in the final objective function is too small, unwanted artifacts may occur. Conversely, if its contribution dominates other loss terms, the output images tend to become overly smooth or blurred. Texture-related losses help mitigate this kind of over-smoothing effect.

3.7. Total Loss Function

Each training method uses a weighted sum of specific loss terms. The total loss for each stage is defined as follows:

$$\mathcal{L}_{\text{paired}} = \sum_i \lambda_i \mathcal{L}_i \tag{10}$$

$$\mathcal{L}_{\text{unpaired}} = \sum_j \lambda_j \mathcal{L}_j \tag{11}$$

$$\mathcal{L}_{\text{pretrain}} = \sum_k \lambda_k \mathcal{L}_k \tag{12}$$

The loss terms used in each stage are as follows:

- **Paired:**

$$\mathcal{L}_i \in \left\{\begin{matrix}\mathcal{L}_{\text{content (paired)}}, \mathcal{L}_{\text{LPIPS+}}, \mathcal{L}_{\text{DISTS}}, \\ \mathcal{L}_{\text{TV}}, \mathcal{L}_{\text{color}}, \mathcal{L}_{\text{adv (lin0)}}, \mathcal{L}_{\text{adv (lin3)}}\end{matrix}\right\}$$

- **Unpaired:**

$$\mathcal{L}_j \in \left\{\begin{matrix}\mathcal{L}_{\text{content (unpaired)}}, \mathcal{L}_{\text{adv (color)}}, \\ \mathcal{L}_{\text{adv (lin0)}}, \mathcal{L}_{\text{adv (lin3)}}, \mathcal{L}_{\text{TV}}\end{matrix}\right\}$$

- **Pretraining:**

$$\mathcal{L}_k \in \{\mathcal{L}_{\text{content (paired)}}, \mathcal{L}_{\text{MSE}}, \mathcal{L}_{\text{TV}}\}$$

Each λ denotes the corresponding weight for its associated loss term. The scaling factors are dynamically computed at each training step to ensure that, upon reaching the generator, the gradient norm of each loss component is normalized to 1. This strategy, referred to as Dynamic Loss Adaptation, ensures balanced gradient contributions from all losses during optimization.

When training with adversarial losses, it is important that the model already demonstrates structural consistency and a reasonable level of color reconstruction, as these elements are essential for learning stability. To ensure this, the network is first pre-trained to perform demosaicing on the RAW input with the loss terms specified in Eq. (12).

In our experiments, we consider 3 training scenarios:

- Paired data are available, and the formulation from Eq. (10) is adopted, except for adversarial losses.
- Paired data are available, and the formulation described in Eq. (10) is fully adopted.
- The data is unpaired and the training follows the configuration described in Eq. (11).

4. Experiments

4.1. Dataset

Our method is evaluated on the ZRR [17] and Fujifilm UltraISP [20] datasets to demonstrate its generalization across differing data distributions. A key advantage of our method is that since it does not require paired data at the content level, it is robust by design to various sources of misalignment (Fig. 4).

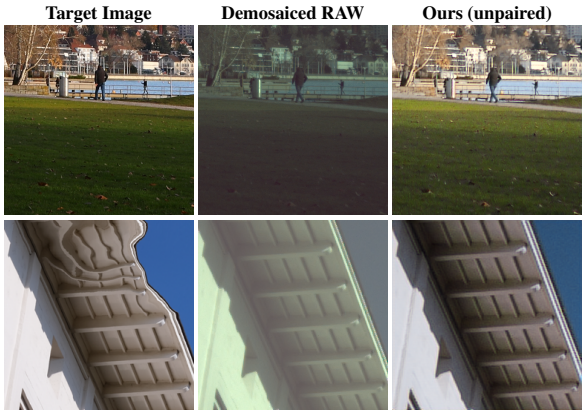


Figure 4. Dataset challenges. The first row shows images from the ZRR dataset (training subset) [17], which include dynamic elements and slight viewpoint misalignments. The second row shows a Fujifilm UltraISP [20] training sample with noticeable warping caused by the alignment algorithm.

Each RAW training image from the ZRR dataset [17] is captured using a 12.3 MP Sony Exmor IMX380 Bayer sensor and paired with a corresponding image generated by a high-end Canon 5D Mark IV camera. For global alignment, SIFT keypoints [31] and RANSAC [37] were used, followed by patch extraction (448×448) using a sliding window. To further refine the alignment, the patch positions were adjusted to maximize cross-correlation, resulting in 48K aligned RAW–RGB samples. From this set, 1.2K pairs were reserved for testing, the remainder being used for training and validation. This data set is entirely available to the public.

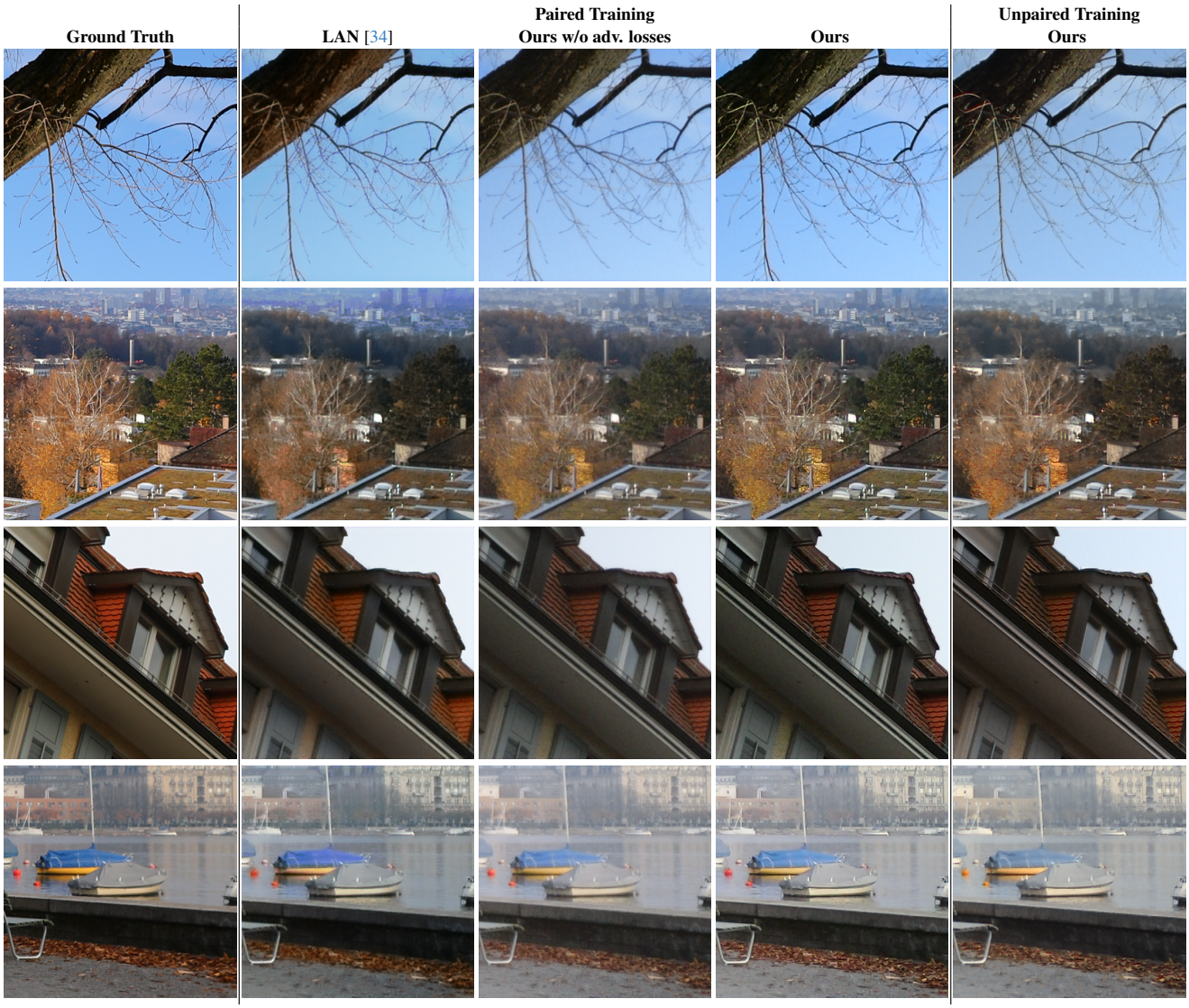


Figure 5. Visual comparisons of outputs and target images on ZRR dataset (test subset) [17]. Last three columns show visual results of Efficient ISP trained under different data access settings.

In Fujifilm UltraISP [20], the authors used a Sony IMX586 Quad Bayer camera sensor and a Fujifilm GFX100 DSLR to acquire visual data. To enhance alignment with the demosaiced input, target images were processed using PDC-Net [36], followed by the extraction of 256×256 pixel patches. Only training pairs and raw validation patches were publicly released. Participants could upload their output through the contest platform and receive PSNR and SSIM scores for the official validation set.

When developing locally, we first removed 17.6% of the training samples from the Fujifilm UltraISP dataset due to small misalignments and then split the remaining data so that 1,024 images were used for validation and another 1,024 for testing. For evaluation on the ZRR dataset, we randomly sampled 1,024 images from the training set to create a validation set.

4.2. Ablation Study

To effectively guide texture learning through a GAN-based loss, the discriminator should receive information that emphasizes features relevant to the target objective, while suppressing non-essential ones. Previous works have addressed this by converting the image to grayscale before passing it to the discriminator, removing color information, and allowing the model to focus more on structural and textural details. However, texture is strongly correlated with perceptual quality, which is often evaluated using the LPIPS [40] measure. Since LPIPS computes a weighted difference of deep features across multiple layers, feeding such feature

Data Access	Method	Backbone	Params ↓	Latency (ms) ↓	PSNR ↑	SSIM ↑	MS-SSIM ↑	LPIPS ↓
Paired	ours w/o adv.losses	Efficient ISP	3K	32.9	19.667	0.699	0.831	0.240
	ours	Efficient ISP	3K	32.9	19.692	0.699	0.837	**0.198**
	original [34]	LAN	46K	241	**20.403**	0.697	**0.843**	0.213
	original [7]	SRCNN	25K	1013	19.833	0.693	0.808	0.370
Unpaired	ours	Efficient ISP	3K	32.9	19.448	**0.700**	0.832	0.239

Table 1. Evaluation on ZRR test data [17]. All models were trained on ZRR. Inference time is measured on mobile GPU on a Full HD (1920 × 1088) image.

Data Access	Method	Backbone	Local Validation PSNR ↑	SSIM ↑	MS-SSIM ↑	LPIPS ↓	Local Test PSNR ↑	SSIM ↑	MS-SSIM ↑	LPIPS ↓	Competition Valid. PSNR ↑	SSIM ↑
Paired	ours w/o adv.losses	RMFA-Net tiny	23.6253	0.8256	0.9354	0.1597	23.7468	0.8321	0.9360	0.1545	23.52	0.82
		Efficient ISP	23.1083	0.8214	0.9382	0.1902	23.2692	0.8273	0.9374	0.1851	23.24	0.81
		Robust ISP	23.0679	0.8236	0.9414	0.1960	23.1823	0.8286	0.9402	0.1930	23.17	0.81
	ours	RMFA-Net tiny	24.4667	0.8653	0.9573	0.1147	24.5764	0.8698	0.9572	0.1121	24.27	0.85
		Efficient ISP	23.5346	0.8367	0.9502	0.1490	23.6477	0.8401	0.9498	0.1465	23.65	0.83
		Robust ISP	23.3905	0.8381	0.9489	0.1557	23.4914	0.8411	0.9482	0.1547	23.52	0.83
Unpaired	ours	RMFA-Net tiny	22.8727	0.8399	0.9455	0.1593	22.7739	0.8451	0.9452	0.1586	22.75	0.83
		Efficient ISP	23.0275	0.8259	0.9457	0.1635	23.0711	0.8300	0.9449	0.1614	23.10	0.81
		Robust ISP	22.7885	0.8259	0.9446	0.1785	22.9401	0.8309	0.9445	0.1751	22.97	0.81

Table 2. Quantitative results on the Fujifilm UltraISP dataset [20] and on the Mobile AI 2025 competition validation data [22].

Demosaicing	$Beta_1$	PSNR ↑	SSIM ↑	MS-SSIM ↑	LPIPS ↓
OpenCV (BG2RGB)	0.0	19.4509	0.6925	0.8294	0.2351
	0.5	**19.5277**	**0.7009**	**0.8322**	0.2396
Menon [32] 2007	0.0	19.4545	0.6916	0.8301	0.2336
	0.5	19.4809	0.6907	0.8290	**0.2328**

Table 3. Results reported on ZRR test set [17]. Efficient ISP was trained under our unpaired setting with different demosaicing algorithms and momentum values.

Layer	PSNR ↑	SSIM ↑	MS-SSIM ↑	LPIPS ↓
lin0	**19.1230**	**0.6661**	0.8139	0.2841
lin1	18.6935	0.6639	**0.8152**	0.2920
lin2	18.3013	0.6514	0.8065	0.2950
lin3	18.2038	0.6539	0.8090	**0.2838**
lin4	17.9374	0.6474	0.7962	0.3121

Table 4. Performance comparison of Efficient ISP trained with unpaired data from the ZRR [17] dataset. Results are reported on ZRR test set using a single discriminator conditioned on different LPIPS+ [3] feature map layers.

representations directly into the discriminator can promote better LPIPS performance. As a result, it leads to improved texture reconstruction and perceptual realism. Different layers capture progressively more abstract feature maps, ranging from low-level information to high-level representations as the network goes deeper. Among the tested configurations, the set-up that uses one discriminator to learn features from lin0 and another from lin3 provides the best results. This outcome is expected, as both discriminators individually outperformed the others in previous experiments on the perceptual score (Tab. 4). The lin0-based discriminator plays a key role in counteracting the potential over-smoothing introduced by the total variation loss by emphasizing high-frequency information, including fine details and noise. In contrast, the lin3-based discriminator reduces unwanted noise without compromising the structural fidelity enhanced by the lin0-based one.

Using blurred versions of the images, normalized and fed into the convolutional discriminator, has proven effective for learning color. Although with this approach, we achieved comparable performance, we observed that passing the blurred image through a pre-trained network and feeding the resulting feature maps to the discriminator led to significantly faster convergence, more stable training, and reduced variation caused by updates. We opted for a Vision Transformer architecture as it demonstrated better learning stability compared to other options such as VGG or AlexNet.

In general, the desired balance required to make the GAN training work requires a careful choice of hyperparameters and sometimes additional empirical adjustments. An important hyperparameter is momentum (Adam $Beta_1$), whose value was directly related to training success, as highlighted in [12]. To address stability and generalization in the unpaired approach, we performed experiments with different values for $Beta_1$, as well as different demosaicing algorithms, *i.e.* Menon 2007 [32] and the

OpenCV built-in demosaicing method (BG2RGB). The results in Tab. 3 show consistent performance across settings, indicating robustness in this regard.

Backbones We adopt the network architecture proposed by the winner of [21], hereafter referred to as Efficient ISP. It consists of only three convolutional layers with 3×3 kernels and 12 channels each, followed by a pixel-shuffle layer. The first activation function is Tanh, while ReLU is used in the subsequent layers. Despite its simplicity, the model demonstrated good performance and notable computational efficiency in the competition.

We evaluated the performance of Efficient ISP, trained with paired and unpaired data, in comparison to two locally trained sota models: LAN [34], using original source code provided by the authors and a custom implementation of SRCNN [7]. We also explored alternative channel configurations and proposed a second backbone with 16, 4, and 12 channels, called a Robust ISP. The latter has been chosen because it is faster, has fewer parameters, and provides a higher competition score [21] when measured locally. The tiny version of RMFA-Net [29] is our third backbone. It is designed to run on smartphones and the authors reported the sota performance on MAI 2022 data set of 24.549 dB in PSNR, which is consistent with our results in paired setting.

As shown in Tab. 1, the models trained with our method perform well in structural metrics, achieving particularly favorable LPIPS scores, the main objective in our case. In the challenge dataset, the unpaired method generally outperforms the paired variant without GAN-based losses, both structurally and perceptually. Besides the contribution of textural component, one factor implied is that the unpaired method's content loss remains unaffected by potential misalignments. The results are consistent across all three splits, indicating robust generalization.

It should be mentioned that the winning model of the 2022 edition of the MAI Learned ISP Challenge got 23.33 dB PSNR [21] in the final ranking. The same network (Efficient ISP), trained with our unpaired strategy, obtained 23.10 dB PSNR on the official validation set and above 23 dB on the other data partitions we used for evaluation (Tab. 2).

Furthermore, the texture component demonstrated a significant contribution to perceptual quality (LPIPS score) when multiple models were trained with access to paired data (Tab. 2). This effect is consistent in the ZRR dataset, as can be observed in quantitative (Tab. 1) and qualitative (Fig. 5) results.

4.3. Implementation Details

The models were trained using Adam optimizer with a batch size of 32. Since the generator deals with a more complex task with its lightweight architecture and receives feedback from multiple sources, the discriminators learning process needs to be slowed down through reduced learning rates and appropriate update ratios. These hyperparameters are required to be fine-tuned for each learnable ISP, depending on the complexity of the network.

Efficient and Robust ISPs used a learning rate of $5 \cdot 10^{-4}$ and their discriminators were trained with a learning rate of 10^{-5} at every 10th step. The tiny RMFA-Net used a learning rate of 10^{-4} and was trained in a ratio of 4:1 with discriminators using a learning rate of $5 \cdot 10^{-5}$.

We performed the training on multiple cloud virtual machines, each configured with an NVIDIA RTX 4090 GPU. The code was implemented in Pytorch framework and uses the support of IQA-PyTorch Toolbox [2].

5. Conclusion

In this work, we introduced a new method for training a learnable ISP capable of running on mobile devices without the restriction of paired data. With the same backbone architecture as the 2022 MAI challenge winner, our unpaired data training method achieves a PSNR score only 0.3 dB lower than the original approach that relied on paired images [21]. With the help of discriminators that receive perceptually relevant feature maps from pre-trained networks, the neural ISP is guided to focus on fine details and textures that enhance perceptual quality. When integrated in a paired setting, the adversarial component of texture leads to even greater visual fidelity.

For the paired approach, further improvements in color accuracy and tone mapping can be achieved by integrating NILUT [4] as a preprocessing step. To address the challenges in the unpaired training setting, future work will focus on improving training performance through adaptive hyperparameter selection and reducing the fidelity gap, particularly with respect to PSNR, between the results obtained from training with unpaired data and those obtained using paired data.

Acknowledgments

This work was partially supported by the Humboldt Foundation.

References

[1] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 214–223. PMLR, 2017. 3

[2] Chaofeng Chen and Jiadi Mo. IQA-PyTorch: Pytorch toolbox for image quality assessment. [Online]. Available: https://github.com/chaofengc/IQA-PyTorch, 2022. 4, 8

[3] Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Topiq: A

top-down approach from semantics to distortions for image quality assessment. *IEEE Transactions on Image Processing*, 2024. 4, 7

[4] Marcos V Conde, Javier Vazquez-Corral, Michael S Brown, and Radu Timofte. NILUT: Conditional neural implicit 3d lookup tables for image enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1371–1379, 2024. 8

[5] Linhui Dai, Xiaohong Liu, Chengqi Li, and Jun Chen. Awnet: Attentive wavelet network for image isp. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 185–201. Springer, 2020. 2

[6] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2020. 3, 4

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 7, 8

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021. 4

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017. 3

[11] Ming-Chun Hsyu, Chih-Wei Liu, Chao-Hung Chen, Chao-Wei Chen, and Wen-Chia Tsai. Csanet: High speed channel spatial attention network for mobile isp. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2486–2493, 2021. 2

[12] Nick Huang, Aaron Gokaslan, Volodymyr Kuleshov, and James Tompkin. The gan is dead; long live the gan! a modern gan baseline. *Advances in Neural Information Processing Systems*, 37:44177–44215, 2024. 3, 4, 7

[13] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3277–3285, 2017. 2, 3, 4

[14] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Wespe: weakly supervised photo enhancer for digital cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 691–700, 2018. 1, 2, 3

[15] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[16] Andrey Ignatov, Radu Timofte, Zhilu Zhang, Ming Liu, Haolin Wang, Wangmeng Zuo, Jiawei Zhang, Ruimao Zhang, Zhanglin Peng, Sijie Ren, et al. Aim 2020 challenge on learned image signal processing pipeline. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 152–170. Springer, 2020. 2

[17] Andrey Ignatov, Luc Van Gool, and Radu Timofte. 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, 2020. 1, 2, 5, 6, 7

[18] Andrey Ignatov, Cheng-Ming Chiang, Hsien-Kai Kuo, Anastasia Sycheva, Radu Timofte, et al. 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*, pages 2503–2514, 2021. 2

[19] Andrey Ignatov, Grigory Malivenko, Radu Timofte, Yu Tseng, Yu-Syuan Xu, Po-Hsiang Yu, Cheng-Ming Chiang, Hsien-Kai Kuo, Min-Hung Chen, Chia-Ming Cheng, et al. Pynet-v2 mobile: Efficient on-device photo processing with neural networks. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 677–684. IEEE, 2022. 2

[20] Andrey Ignatov, Anastasia Sycheva, Radu Timofte, Yu Tseng, Yu-Syuan Xu, Po-Hsiang Yu, Cheng-Ming Chiang, Hsien-Kai Kuo, Min-Hung Chen, Chia-Ming Cheng, et al. Microisp: processing 32mp photos on mobile devices with deep learning. In *European Conference on Computer Vision*, pages 729–746. Springer, 2022. 1, 2, 5, 6, 7

[21] Andrey Ignatov, Radu Timofte, Shuai Liu, Chaoyu Feng, Furui Bai, Xiaotao Wang, Lei Lei, Ziyao Yi, Yan Xiang, Zibin Liu, et al. Learned smartphone isp on mobile gpus with deep learning, mobile ai & aim 2022 challenge: report. In *European Conference on Computer Vision*, pages 44–70. Springer, 2022. 1, 2, 8

[22] Andrey Ignatov, Georgii Perevozchikov, Radu Timofte, et al. Learned smartphone isp on mobile gpus, mobile ai 2025 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 7

[23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 3

[24] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018. 3, 4

[25] Byung-Hoon Kim, Joonyoung Song, Jong Chul Ye, and Jae-Hyun Baek. Pynet-ca: enhanced pynet with channel attention for end-to-end mobile image signal processing. In *European Conference on Computer Vision*, pages 202–212. Springer, 2020. 2

[26] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. 2

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017. 4

[28] Edwin Herbert Land and John J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61 1:1–11, 1971. 2

[29] Fei Li, Wenbo Hou, and Peng Jia. Rmfa-net: A neural isp for real raw to rgb image reconstruction. *arXiv preprint arXiv:2406.11469*, 2024. 2, 4, 8

[30] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30, 2017. 2

[31] David G Low. Distinctive image features from scale-invariant keypoints. *Journal of Computer Vision*, 60(2):91–110, 2004. 5

[32] Daniele Menon, Stefano Andriani, and Giancarlo Calvagno. Demosaicing with directional filtering and a posteriori decision. *IEEE Transactions on Image Processing*, 16(1):132–141, 2007. 7

[33] Georgy Perevozchikov, Nancy Mehta, Mahmoud Afifi, and Radu Timofte. Rawformer: Unpaired raw-to-raw translation for learnable camera isps. In *European Conference on Computer Vision*, pages 231–248. Springer, 2024. 1

[34] Daniel Wirzberger Raimundo, Andrey Ignatov, and Radu Timofte. Lan: Lightweight attention-based network for raw-to-rgb smartphone image processing. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 807–815, 2022. 2, 6, 7, 8

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015. 3, 4

[36] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. 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, 2021. 6

[37] Andrea Vedaldi and Brian Fulkerson. Vlfeat: an open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM International Conference on Multimedia*, page 1469–1472, New York, NY, USA, 2010. Association for Computing Machinery. 5

[38] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, pages 1398–1402. Ieee, 2003. 2

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2

[40] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2, 3, 6

[41] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2