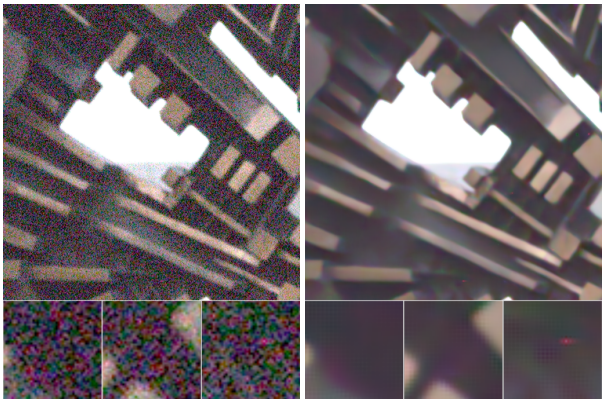
CycleISP: Real Image Restoration via Improved Data Synthesis

Syed Waqas Zamir[1] Aditya Arora[1] Salman Khan[1] Munawar Hayat[1]
Fahad Shahbaz Khan[1] Ming-Hsuan Yang[2,3] Ling Shao[1]
[1]Inception Institute of Artificial Intelligence, UAE
[2]University of California, Merced [3]Google Research

Abstract

The availability of large-scale datasets has helped unleash the true potential of deep convolutional neural networks (CNNs). However, for the single-image denoising problem, capturing a real dataset is an unacceptably expensive and cumbersome procedure. Consequently, image denoising algorithms are mostly developed and evaluated on synthetic data that is usually generated with a widespread assumption of additive white Gaussian noise (AWGN). While the CNNs achieve impressive results on these synthetic datasets, they do not perform well when applied on real camera images, as reported in recent benchmark datasets. This is mainly because the AWGN is not adequate for modeling the real camera noise which is signal-dependent and heavily transformed by the camera imaging pipeline. In this paper, we present a framework that models camera imaging pipeline in forward and reverse directions. It allows us to produce any number of realistic image pairs for denoising both in RAW and sRGB spaces. By training a new image denoising network on realistic synthetic data, we achieve the state-of-the-art performance on real camera benchmark datasets. The parameters in our model are ∼5 times lesser than the previous best method for RAW denoising. Furthermore, we demonstrate that the proposed framework generalizes beyond image denoising problem e.g., for color matching in stereoscopic cinema. The source code and pre-trained models are available at https://github.com/swz30/CycleISP.



(a) Noisy Input PSNR(RAW) / PSNR(sRGB) | (b) N3NET [45] 38.24 dB / 32.42 dB

(c) UPI [7] 37.37 dB / 35.49 dB | (d) Ours **40.44 dB / 36.16 dB**

Figure 1: Denoising a real camera image from DND dataset [44]. Our model is effective in removing real noise, especially the low-frequency chroma and defective pixel noise.

1. Introduction

High-level computer vision tasks, such as image classification, object detection and segmentation have witnessed significant progress due to deep CNNs [33]. The major driving force behind the success of CNNs is the availability of large-scale datasets [17, 38], containing hundreds of thousands of annotated images. However, for *low-level* vision problems (image denoising, super-resolution, deblurring, etc.), collecting even small datasets is extremely challenging and non-trivial. For instance, the typical procedure to acquire noisy paired data is to take multiple noisy images of the same scene and generate clean ground-truth image by pixel-wise averaging. In practice, spatial pixels misalignment, color and brightness mismatch is inevitable due to changes in lighting conditions and camera/object motion. Moreover, this expensive and cumbersome exercise of acquiring image pairs needs to be repeated with different camera sensors, as they exhibit different noise characteristics.

Consequently, single image denoising is mostly per-

formed in synthetic settings: take a large set of clean sRGB images and add synthetic noise to generate their noisy versions. On synthetic datasets, existing deep learning based denoising models yield impressive results, but they exhibit poor generalization to real camera data as compared to conventional methods [8, 15]. This trend is also demonstrated in recent benchmarks [1, 44]. Such behavior stems from the fact that deep CNNs are trained on synthetic data that is usually generated with the Additive White Gaussian Noise (AWGN) assumption. Real camera noise is fundamentally different from AWGN, thereby causing a major challenge for deep CNNs [6, 22, 24].

In this paper, we propose a synthetic data generation approach that can produce realistic noisy images both in RAW and sRGB spaces. The main idea is to inject noise in the RAW images obtained with our learned device-agnostic transformation rather than in the sRGB images directly. The key insight behind our framework is that the real noise present in sRGB images is convoluted by the series of steps performed in a regular image signal processing (ISP) pipeline [6, 46]. Therefore, modeling real camera noise in sRGB is an inherently difficult task as compared to RAW sensor data [35]. As an example, noise at the RAW sensor space is signal-dependent; after demosaicking, it becomes spatio-chromatically correlated; and after passing through the rest of the pipeline, its probability distribution not necessarily remains Gaussian [53]. This implies that the camera ISP heavily transforms the sensor noise, and therefore more sophisticated models that take into account the influence of imaging pipeline are needed to synthesize realistic noise than uniform AWGN model [1, 26, 44].

In order to exploit the abundance and diversity of sRGB photos available on the Internet, the main challenge with the proposed synthesis approach is how to transform them back to RAW measurements. Brooks *et al*. [7] present a technique that inverts the camera ISP, step-by-step, and thereby allows conversion from sRGB to RAW data. However, this approach requires prior information about the target camera device (e.g., color correction matrices and white balance gains), which makes it specific to a given device and therefore lacks in generalizability. Furthermore, several operations in a camera pipeline are proprietary and such black boxes are very difficult to reverse engineer. To address these challenges, in this paper we propose a CycleISP framework that converts sRGB images to RAW data, and then back to sRGB images, without requiring any knowledge of camera parameters. This property allows us to synthesize any number of clean and realistic noisy image pairs in both RAW and sRGB spaces. Our main contributions are:

- Learning a device-agnostic transformation, called CycleISP, that allows us to move back and forth between sRGB and RAW image spaces.
- Real image noise synthesizer for generating clean/noisy paired data in RAW and sRGB spaces.
- A deep CNN with dual attention mechanism that is effective in a variety of tasks: learning CycleISP, synthesizing realistic noise, and image denoising.
- Algorithms to remove noise from RAW and sRGB images, setting new state-of-the-art on real noise benchmarks of DND [44] and SIDD [1] (see Fig. 1). Moreover, our denoising network has much fewer parameters (2.6M) than the previous best model (11.8M) [7].
- CycleISP framework generalizes beyond denoising, we demonstrate this via an additional application *i.e*., color matching in stereoscopic cinema [41, 59, 49].

2. Related Work

The presence of noise in images is inevitable, irrespective of the acquisition method; now more than ever, when majority of images come from smartphone cameras having small sensor size but large resolution. Single-image denoising is a vastly researched problem in the computer vision and image processing community, with early works dating back to 1960's [6]. Classic methods on denoising are mainly based on the following two principles. (1) Modifying transform coefficients using the DCT [63], wavelets [19, 55], etc. (2) Averaging neighborhood values: in all directions using Gaussian kernel, in all directions only if pixels have similar values [56, 58] and along contours [42, 51].

While these aforementioned methods provide satisfactory results in terms of image fidelity metrics and visual quality, the Non-local Means (NLM) algorithm of Buades *et al*. [8] makes significant advances in denoising. The NLM method exploits the redundancy, or self-similarity [20] present in natural images. For many years the patch-based methods yielded comparable results, thus prompting studies [11, 12, 37] to investigate whether we reached the theoretical limits of denoising performance. Subsequently, Burger *et al*. [9] train a simple Multi-Layer Perceptron (MLP) on a large synthetic noise dataset. This method performs well against previous sophisticated algorithms. Several recent methods use deep CNNs [4, 7, 25, 28, 45, 66, 67, 2] and demonstrate promising denoising performance.

Image denoising can be applied to RAW or sRGB data. However, capturing diverse large-scale real noise data is a prohibitively expensive and tedious procedure, consequently leaving us to study denoising in synthetic settings. The most commonly used noise model for developing and evaluating image denoising is AWGN. As such, algorithms that are designed for AWGN cannot effectively remove noise from real images, as reported in recent benchmarks [1, 44]. A more accurate model for real RAW sensor noise contains both the signal-dependent noise component (the shot noise), and the signal-independent additive Gaussian component (the read noise) [22, 23, 24]. The camera ISP transforms RAW sensor noise into a complicated form

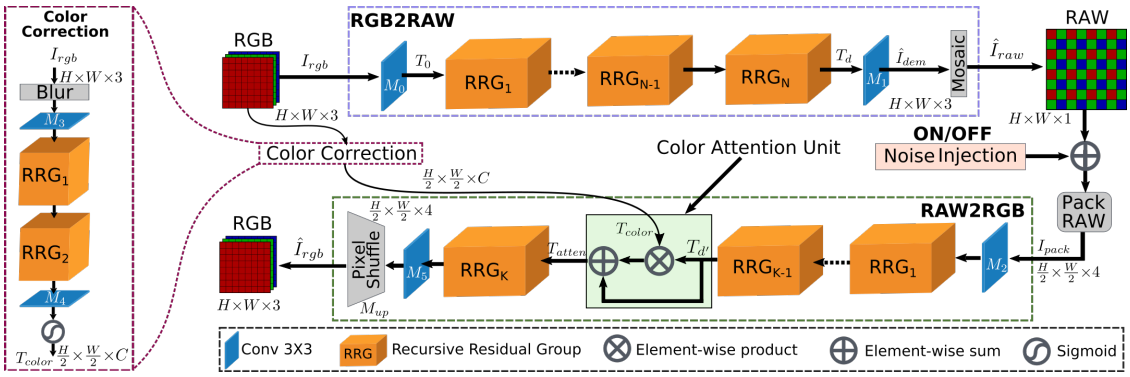


Figure 2: Our CycleISP models the camera imaging pipeline in both directions. It comprises two main branches: RGB2RAW and RAW2RGB. The RGB2RAW branch converts sRGB images to RAW measurements, whereas the RAW2RGB branch transforms RAW data to sRGB images. The auxiliary color correction branch provides explicit color attention to RAW2RGB network. The noise injection module is switched OFF while training the CycleISP (Section 3), and switched ON when synthesizing noise data (Section 4).

(spatio-chromatically correlated and not necessarily Gaussian). Therefore, estimating a noise model for denoising in sRGB space requires careful consideration of the influence of ISP. In this paper, we present a framework that is capable of synthesizing realistic noise data for training CNNs to effectively remove noise from RAW as well as sRGB images.

3. CycleISP

To synthesize realistic noise datasets, we use a two-stage scheme in this work. First, we develop a framework that models the camera ISP both in forward and reverse directions, hence the name CycleISP. Second, using CycleISP, we synthesize realistic noise datasets for the tasks of RAW denoising and sRGB image denoising. In this section, we only describe our CycleISP framework that models the camera ISP as a deep CNN system. Fig. 2 shows the modules of the CycleISP model: (a) RGB2RAW network branch, and (b) RAW2RGB network branch. In addition, we introduce an auxiliary color correction network branch that provides explicit color attention to the RAW2RGB network in order to correctly recover the original sRGB image.

The noise injection module in Fig. 2 is only required when synthesizing noisy data (Section 4), and thus we keep it in the 'OFF' state while learning CycleISP. The training process of CycleISP is divided in two steps: the RGB2RAW and RAW2RGB networks are first independently trained, and then joint fine-tuning is performed. Next, we present details of different branches of CycleISP. Note that we use RGB instead of sRGB to avoid notation clutter.

3.1. RGB2RAW Network Branch

Digital cameras apply a series of operations on RAW sensor data in order to generate the monitor-ready sRGB images [46]. Our RGB2RAW network branch aims to invert the effect of camera ISP. In contrast to the *unprocessing* technique of [7], the RGB2RAW branch does not require any camera parameters.

Given an input RGB image $\mathbf{I}_{rgb} \in \mathbb{R}^{H\times W\times 3}$, the RGB2RAW network first extracts low-level features $T_0 \in \mathbb{R}^{H\times W\times C}$ using a convolutional layer M_0 as: $T_0 = M_0(\mathbf{I}_{rgb})$. Next, we pass the low-level feature maps T_0 through N recursive residual groups (RRGs) to extract deep features $T_d \in \mathbb{R}^{H\times W\times C}$ as:

$$T_d = RRG_N\left(...(RRG_1(T_0))\right), \tag{1}$$

where each RRG contains multiple dual attention blocks, as we shall see in Section 3.3.

We then apply the final convolution operation M_1 to the features T_d and obtain the demosaicked image $\hat{I}_{dem} \in \mathbb{R}^{H\times W\times 3}$. We deliberately set the number of output channels of M_1 layer to three rather than one in order to preserve as much structural information of the original image as possible. Moreover, we empirically found that it helps the network to learn the mapping from sRGB to RAW faster and more accurately. At this point, the network is able to invert the effects of tone mapping, gamma correction, color correction, white balance, and other transformations, and provide us with the image $\hat{I}_{dem}$ whose values are linearly related to the scene radiance. Finally, in order to generate the mosaicked RAW output $\hat{\mathbf{I}}_{raw} \in \mathbb{R}^{H\times W\times 1}$, the Bayer sampling function f_{Bayer} is applied to $\hat{I}_{dem}$ that omits two color channels per pixel according to the Bayer pattern:

$$\hat{\mathbf{I}}_{raw} = f_{bayer}(M_1(T_d)). \tag{2}$$

The RGB2RAW network is optimized using the L_1 loss in linear and log domains as:

$$\begin{aligned}\mathcal{L}_{s\to r}(\hat{\mathbf{I}}_{raw}, \mathbf{I}_{raw}) &= \left\|\hat{\mathbf{I}}_{raw} - \mathbf{I}_{raw}\right\|_1 \\ &+ \left\|\log(\max(\hat{\mathbf{I}}_{raw}, \epsilon)) - \log(\max(\mathbf{I}_{raw}, \epsilon))\right\|_1,\end{aligned} \tag{3}$$

where ϵ is a small constant for numerical stability, and $\mathbf{I}_{raw}$ is the ground-truth RAW image. Similar to [21], the log loss term is added to enforce approximately equal treatment for all the image values; otherwise the network dedicates more attention to recovering the highlight regions.

3.2. RAW2RGB Network Branch

While the ultimate goal of RAW2RGB network is to generate synthetic realistic noise data for the sRGB image denoising problem, in this section we first describe how we can map *clean* RAW images to *clean* sRGB images (leaving the noise injection module 'OFF' in Fig. 2).

Let $\mathbf{I}_{raw}$ and $\hat{\mathbf{I}}_{rgb}$ be the input and output of the RAW2RGB network. First, in order to restore translation invariance and reduce computational cost, we pack the 2×2 blocks of $\mathbf{I}_{raw}$ into four channels (RGGB) and thus reduce the image resolution by half [7, 13, 25]. Since the input RAW data may come from different cameras having different Bayer patterns, we ensure the channel order of the packed image to be RGGB by applying the Bayer pattern unification technique [39]. Next, a convolutional layer M_2 followed by $K-1$ RRG modules encode the packed RAW image $I_{pack} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times4}$ into a deep feature tensor $T_{d'} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times C}$ as:

$$T_{d'} = RRG_{K-1}(...(RRG_1(M_2(\text{Pack}(\mathbf{I}_{raw}))))). \quad (4)$$

Note that $\mathbf{I}_{raw}$ is the original camera RAW image (not the output of RGB2RAW network) because our objective here is to first learn RAW to sRGB mapping, independently.

Color attention unit. To train the CycleISP, we use the MIT-Adobe FiveK dataset [10] that contains images from several different cameras having diverse and complex ISP systems. It is extremely difficult for a CNN to accurately learn a RAW to sRGB mapping function for all different types of cameras (as one RAW image can potentially map to many sRGB images). One solution is to train one network for each camera ISP [13, 52, 64]. However, such solutions are not scalable and the performance may not generalize to other cameras. To address this issue, we propose to include a color attention unit in the RAW2RGB network that provides explicit color attention via a *color correction branch*.

The color correction branch is a CNN that takes as input an sRGB image $\mathbf{I}_{rgb}$ and generates a color-encoded deep feature tensor $T_{color} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times C}$. In the color correction branch, we first apply Gaussian blur to $\mathbf{I}_{rgb}$, followed by a convolutional layer M_3, two RRGs and a gating mechanism with sigmoid activation σ:

$$T_{color} = \sigma(M_4(RRG_2(RRG_1(M_3(K * \mathbf{I}_{rgb}))))), \quad (5)$$

where $*$ denotes convolution, and K is the Gaussian kernel with standard deviation empirically set to 12. This strong blurring operation ensures that only the color information

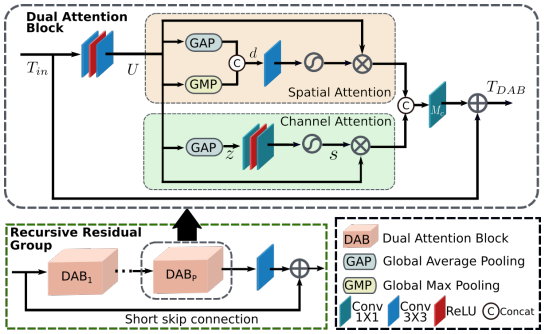


Figure 3: Recursive residual group (RRG) contains multiple dual attention blocks (DAB). Each DAB contains spatial attention and channel attention modules.

flows through this branch, whereas the structural content and fine texture comes from the main RAW2RGB network. Using weaker blurring will undermine the effectiveness of the feature tensor $T_{d'}$ of Eq. (4). The overall color attention unit process becomes:

$$T_{atten} = T_{d'} + (T_{d'} \otimes T_{color}), \quad (6)$$

where, $\otimes$ is Hadamard product. To obtain the final sRGB image $\hat{\mathbf{I}}_{rgb}$, the output features T_{atten} from the color attention unit are passed through a RRG module, a convolutional layer M_4 and an upscaling layer M_{up} [54], respectively:

$$\hat{\mathbf{I}}_{rgb} = M_{up}(M_5(RRG_K(T_{atten}))). \quad (7)$$

For optimizing RAW2RGB network, we use the L_1 loss:

$$\mathcal{L}_{r\to s}(\hat{\mathbf{I}}_{rgb}, \mathbf{I}_{rgb}) = \left\| \hat{\mathbf{I}}_{rgb} - \mathbf{I}_{rgb} \right\|_1. \quad (8)$$

3.3. RRG: Recursive Residual Group

Motivated by the advances of recent low-level vision methods [48, 65, 66, 68] based on the residual learning framework [29], we propose the RRG module, as shown in Fig. 3. The RRG contains P dual attention blocks (DAB). The goal of each DAB is to suppress the less useful features and only allow the propagation of more informative ones. The DAB performs this feature recalibration by using two attention mechanisms: (1) channel attention (CA) [30], and (2) spatial attention (SA) [60]. The overall process is:

$$T_{DAB} = T_{in} + M_c([\text{CA}(U), \text{SA}(U)]), \quad (9)$$

where $U \in \mathbb{R}^{H\times W\times C}$ denotes features maps that are obtained by applying two convolutions on input tensor $T_{in} \in \mathbb{R}^{H\times W\times C}$ at the beginning of the DAB, and M_c is the last convolutional layer with filter size 1×1.

Channel attention. This branch is designed to exploit the inter-channel dependencies of convolutional features. It first performs a *squeeze* operation in order to encode the

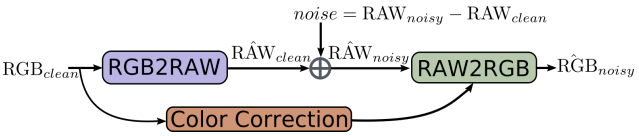


Figure 4: Fine-tuning CycleISP to synthesize realistic sRGB noise data.

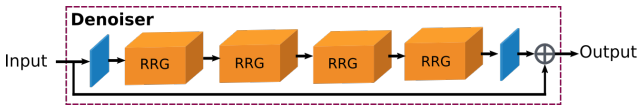


Figure 5: Proposed denoising network. It has the same network structure for denoising both RAW images and sRGB images, except in the handling of input and output.

spatially global context, which is then followed by an *excitation* operation to fully capture channel-wise relationships [30]. The squeeze operation is realized by applying global average pooling (GAP) on feature maps U, thus yielding a descriptor $z \in \mathbb{R}^{1\times1\times C}$. The excitation operator recalibrates the descriptor z using two convolutional layers followed by the sigmoid activation and results in activations $s \in \mathbb{R}^{1\times1\times C}$. Finally, the output of CA branch is obtained by rescaling U with the activations s.

Spatial attention. This branch exploits the inter-spatial relationships of features and computes a spatial attention map that is then used to rescale the incoming features U. To generate the spatial attention map, we first independently apply global average pooling and max pooling operations on features U along the channel dimensions and concatenate the output maps to form a spatial feature descriptor $d \in \mathbb{R}^{H\times W\times 2}$. This is followed by a convolution and sigmoid activation to obtain the spatial attention map.

3.4. Joint Fine-tuning of CycleISP

Since the RGB2RAW and RAW2RGB networks are initially trained independently, they may not provide the optimal-quality images due to the disconnection between them. Therefore, we perform joint fine-tuning in which the output of RGB2RAW becomes the input of RAW2RGB. The loss function for the joint optimization is:

$$\mathcal{L}_{joint} = \beta\mathcal{L}_{s\rightarrow r}(\hat{\mathbf{I}}_{raw}, \mathbf{I}_{raw}) + (1-\beta)\mathcal{L}_{r\rightarrow s}(\hat{\mathbf{I}}_{rgb}, \mathbf{I}_{rgb}),$$

where β is a positive constant. Note that the RAW2RGB network receives gradients from the RAW2RGB sub-loss (only the second term). Whereas, the RGB2RAW network receives gradients from both sub-losses, thereby effectively contributing to the reconstruction of the final sRGB image.

4. Synthetic Realistic Noise Data Generation

Capturing perfectly-aligned real noise data pairs is extremely difficult. Consequently, image denoising is mostly studied in artificial settings where Gaussian noise is added to the clean images. While the state-of-the-art image denoising methods [9, 66] have shown promising performance on these synthetic datasets, they do not perform well when applied on real camera images [1, 44]. This is because the synthetic noise data differs fundamentally from real camera data. In this section, we describe the process of synthesizing realistic noise image pairs for denoising both in RAW and sRGB space using the proposed CycleISP method.

Data for RAW denoising. The RGB2RAW network branch of the CycleISP method takes as input a clean sRGB image and converts it to a clean RAW image (top branch, Fig. 2). The noise injection module, which we kept off while training CycleISP, is now turned to the 'ON' state. The noise injection module adds shot and read noise of different levels to the output of RGB2RAW network. We use the same procedure for sampling shot/read noise factors as in [7]. As such, we can generate clean and its corresponding noisy image pairs $\{\text{RAW}_{clean}, \text{RAW}_{noisy}\}$ from any sRGB image.

Data for sRGB denoising. Given a synthetic RAW_{noisy} image as input, the RAW2RGB network maps it to a noisy sRGB image (bottom branch, Fig. 2); hence we are able to generate an image pair $\{\text{sRGB}_{clean}, \text{sRGB}_{noisy}\}$ for the sRGB denoising problem. While these synthetic image pairs are already adequate for training the denoising networks, we can further improve their quality with the following procedure. We fine-tune the CycleISP model (Section 3.4) using the SIDD dataset [1] that is captured with real cameras. For each static scene, SIDD contains clean and noisy image pairs in both RAW and sRGB spaces. The fine-tuning process is shown in Fig. 4. Notice that the noise injection module which adds random noise is replaced by (only for fine-tuning) per-pixel noise residue that is obtained by subtracting the real RAW_{clean} image from the real RAW_{noisy} image. Once the fine-tuning procedure is complete, we can synthesize realistic noisy images by feeding clean sRGB images to the CycleISP model.

5. Denoising Architecture

As illustrated in Fig. 5, we propose an image denoising network by employing multiple RRGs. Our aim is to apply the proposed network in two different settings: (1) denoising RAW images, and (2) denoising sRGB data. We use the same network structure under both settings, with the only difference being in the handling of input and output. For denoising in the sRGB space, the input and output of the network are the 3-channel sRGB images. For denoising the RAW images, our network takes as input a 4-channel noisy packed image concatenated with a 4-channel noise level map, and provides us with a 4-channel packed denoised output. The noise level map provides an estimate of the standard deviation of noise present in the input image, based on its shot and read noise parameters [7].

6. Experiments

6.1. Real Image Datasets

DND [44]. This dataset consists of 50 pairs of noisy and (nearly) noise-free images captured with four consumer cameras. Since the images are of very high-resolution, the providers extract 20 crops of size 512×512 from each image, thus yielding a total of 1000 patches. The complete dataset is used for testing because the ground-truth noise-free images are not publicly available. The data is provided for two evaluation tracks: RAW space and sRGB space. Quantitative evaluation in terms of PSNR and SSIM can only be performed through an online server [16].

SIDD [1]. Due to the small sensor size and high-resolution, smartphone images are much more noisy than those of DSLRs. This dataset is collected using five smartphone cameras. There are 320 image pairs available for training and 1280 image pairs for validation. This dataset also provides images both in RAW format and in sRGB space.

6.2. Implementation Details

All the models presented in this paper are trained with Adam optimizer ($\beta_1 = 0.9$, and $\beta_2 = 0.999$) and image crops of 128×128. Using the Bayer unification and augmentation technique [39], we randomly perform horizontal and vertical flips. We set a filter size of 3×3 for all convolutional layers of the DAB except the last for which we use 1×1.

Initial training of CycleISP. To train the CycleISP model, we use the MIT-Adobe FiveK dataset [10], which contains 5000 RAW images. We process these RAW images using the LibRaw library and generate sRGB images. From this dataset, 4850 images are used for training and 150 for validation. We use 3 RRGs and 5 DABs for both RGB2RAW and RAW2RGB networks, and 2 RRGs and 3 DABs for the color correction network. The RGB2RAW and RAW2RGB branches of CycleISP are independently trained for 1200 epochs with a batch size of 4. The initial learning rate is 10^{-4}, which is decreased to 10^{-5} after 800 epochs.

Fine-tuning CycleISP. This process is performed twice: first with the procedure presented in Section 3.4, and then with the method of Section 4. In the former case, the output of the CycleISP model is noise-free, and in the latter case, the output is noisy. For each fine-tuning stage, we use 600 epochs, batch size of 1 and learning rate of 10^{-5}.

Training denoising networks. We train four networks to perform denoising on: (1) DND RAW data, (2) DND sRGB images, (3) SIDD RAW data, and (4) SIDD sRGB images. For all four networks, we use 4 RRGs and 8 DABs, 65 epochs, batch size of 16, and initial learning rate of 10^{-4} which is decreased by a factor of 10 after every 25 epochs. We take 1 million images from the MIR flickr extended dataset [31] and split them into a ratio of 90:5:5 for training, validation and testing. All the images are preprocessed with the Gaussian kernel ($\sigma = 1$) to reduce the effect of noise, and other artifacts. Next, we synthesize clean/noisy paired training data (both for RAW and sRGB denoising) using the procedure described in Section 4.

Table 1: RAW denoising results on the DND benchmark dataset [44]. * denotes that these methods use variance stabilizing transform (VST) [40] to provide their best results.

	RAW		sRGB	
Method	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
TNRD* [14]	45.70	0.96	36.09	0.888
MLP* [9]	45.71	0.963	36.72	0.912
FoE [50]	45.78	0.967	35.99	0.904
EPLL* [69]	46.86	0.973	37.46	0.925
KSVD* [3]	46.87	0.972	37.63	0.929
WNNM* [27]	47.05	0.972	37.69	0.926
NCSR* [18]	47.07	0.969	37.79	0.923
BM3D* [15]	47.15	0.974	37.86	0.930
DnCNN [66]	47.37	0.976	38.08	0.936
N3Net [45]	47.56	0.977	38.32	0.938
UPI (Raw) [7]	48.89	0.982	40.17	0.962
Ours	**49.13**	**0.983**	**40.50**	**0.966**

Table 2: RAW denoising results on the SIDD dataset [1].

	RAW		sRGB	
Method	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
EPLL [69]	40.73	0.935	25.19	0.842
GLIDE [57]	41.87	0.949	25.98	0.816
TNRD [14]	42.77	0.945	26.99	0.744
FoE [50]	43.13	0.969	27.18	0.812
MLP [9]	43.17	0.965	27.52	0.788
KSVD [3]	43.26	0.969	27.41	0.832
DnCNN [66]	43.30	0.965	28.24	0.829
NLM [8]	44.06	0.971	29.39	0.846
WNNM [27]	44.85	0.975	29.54	0.888
BM3D [15]	45.52	0.980	30.95	0.863
Ours	**52.41**	**0.993**	**39.47**	**0.918**

6.3. Results for RAW Denoising

In this section, we evaluate the denoising results of the proposed CycleISP model with existing state-of-the-art methods on the RAW data from DND [44] and SIDD [1] benchmarks. Table 1 shows the quantitative results (PSNR/SSIM) of all competing methods on the DND dataset obtained from the website of the evaluation server [16]. Note that there are two super columns in the table listing the values of image quality metrics. The numbers in the sRGB super column are provided by the server after passing the denoised RAW images through the camera imaging pipeline [32] using image metadata. Our model consistently performs better against the learning-based as well as conventional denoising algorithms. Furthermore, the proposed model has $\sim 5\times$ lesser parameters than previous best method [7]. The trend is similar for the SIDD dataset, as shown in Table 2. Our algorithm achieves 6.89 dB improvement in PSNR over the BM3D algorithm[15].

A visual comparison of our result against the state-of-

Table 3: Denoising sRGB images of the DND benchmark dataset [44].

Method	EPLL [69]	TNRD [14]	NCSR [18]	MLP [9]	BM3D [15]	FoE [50]	WNNM [27]	KSVD [3]	MCWNNM [62]	FFDNet+ [67]	TWSC [61]	CBDNet [28]	RIDNet [4]	Ours
PSNR ↑	33.51	33.65	34.05	34.23	34.51	34.62	34.67	36.49	37.38	37.61	37.94	38.06	39.23	**39.56**
SSIM ↑	0.824	0.831	0.835	0.833	0.851	0.885	0.865	0.898	0.929	0.942	0.940	0.942	0.953	**0.956**

Table 4: Denoising sRGB images of the SIDD benchmark dataset [1].

Method	DnCNN [66]	MLP [9]	GLIDE [57]	TNRD [14]	FoE [50]	BM3D [15]	WNNM [27]	NLM [8]	KSVD [3]	EPLL [69]	CBDNet [28]	RIDNet [4]	Ours
PSNR ↑	23.66	24.71	24.71	24.73	25.58	25.65	25.78	26.76	26.88	27.11	30.78	38.71	**39.52**
SSIM ↑	0.583	0.641	0.774	0.643	0.792	0.685	0.809	0.699	0.842	0.870	0.754	0.914	**0.957**

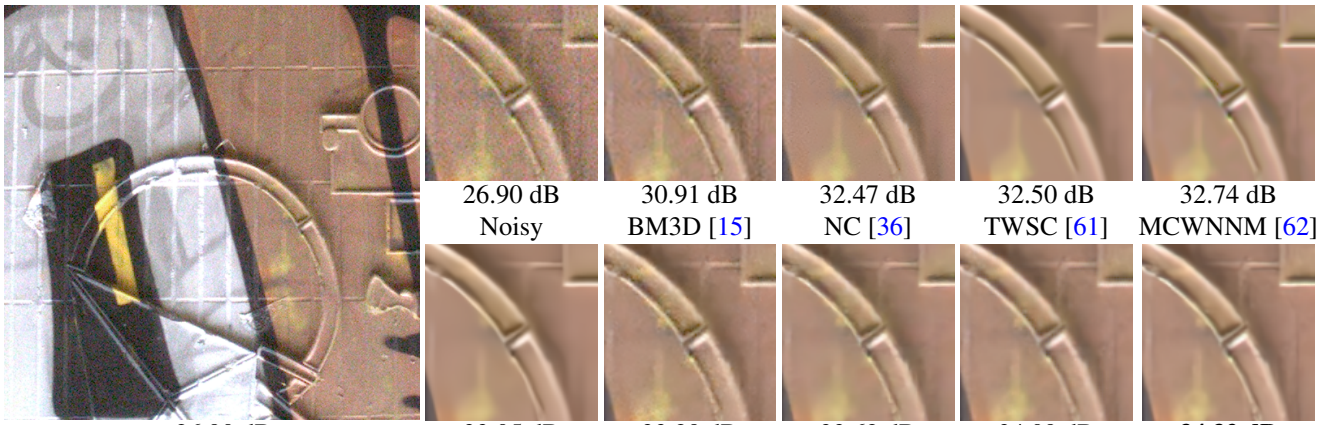


Figure 6: Denoising sRGB image from DND [44]. Our method preserves better structural content than other algorithms.

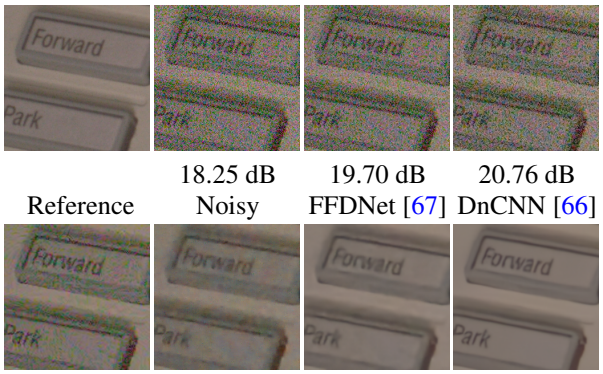


Figure 7: Denoising results of different methods on a challenging sRGB image from the SIDD dataset [1].

the-art algorithms is presented in Fig. 1. Our model is very effective in removing real noise, especially the low-frequency chroma noise and defective pixel noise.

6.4. Results for sRGB Denoising

While it is recommended to apply denoising on RAW data (where noise is uncorrelated and less complex) [26], denoising is commonly studied in the sRGB domain. We compare the denoising results of different methods on sRGB images from the DND and SIDD datasets. Table 3 and 4 show the scores of image quality metrics. Overall, the proposed model performs favorably against the state-of-the-art. Compared to the recent best algorithm RIDNet [4], our approach demonstrates the performance gain of 0.33 dB and 0.81 dB on DND and SIDD datasets, respectively.

Fig. 6 and 7 illustrate the sRGB denoising results on DND and SIDD, respectively. To remove noise, most of the evaluated algorithms either produce over-smooth images (and sacrifice image details) or generate images with splotchy texture and chroma artifacts. In contrast, our method generates clean and artifact-free results, while faithfully preserving image details.

6.5. Generalization Test

To compare the generalization capability of the denoising model trained on the synthetic data generated by our method and that of [7], we perform the following experiments. We take the (publicly available) denoising model of [7] trained for DND, and directly evaluate it on the RAW images from the SIDD dataset. We repeat the same procedure for our denoising model as well. For a fair comparison, we use the same network architecture (U-Net) and noise model as of [7]. The only difference is data conver-

Table 5: Generalization Test. U-Net model is trained only for DND [44] with our technique and with the UPI [7] method, and directly evaluated on the SIDD dataset [1].

Method	DND [44] PSNR	DND [44] SSIM	SIDD [1] PSNR	SIDD [1] SSIM
UPI [7]	48.89	0.9824	49.17	0.9741
Ours	**49.00**	**0.9827**	**50.14**	**0.9758**

Table 6: Ablation study: RAW2RGB branch.

Short skip connections		✓	✓	✓	✓	✓
Color correction branch	✓	✓		✓	✓	✓
Channel Attention (CA)	✓		✓	✓		✓
Spatial attention (SA)	✓		✓		✓	✓
PSNR (in dB)	23.22	42.96	33.58	44.67	45.08	**45.41**

Table 7: Layout of SA and CA in DAB.

Layout	CA + SA	SA + CA	CA & SA in parallel
PSNR (in dB)	45.17	45.16	**45.41**

sion from sRGB to RAW. The results in Table 5 show that the denoising network trained with our method not only performs well on the DND dataset but also shows promising generalization to the SIDD set (a gain of $\sim$ 1 dB over [7]).

6.6. Ablations

We study the impact of individual contributions by progressively integrating them to our model. To this end, we use the RAW2RGB network that maps *clean* RAW image to *clean* sRGB image. Table 6 shows that the skip connections cause the largest performance drop, followed by the color correction branch. Furthermore, it is evident that the presence of both CA and SA is important, as well as their configuration (see Table 7), for the overall performance.

6.7. Color Matching For Stereoscopic Cinema

In professional 3D cinema, stereo pairs for each frame are acquired using a stereo camera setup, with two cameras mounted on a rig either side-by-side or (more commonly) in a beam splitter formation [5]. During movie production, meticulous efforts are required to ensure that the twin cameras perform in exactly the same manner. However, oftentimes visible color discrepancies between the two views are inevitable because of the imperfect camera adjustments and impossibility of manufacturing identical lens systems. In movie post-production, color mismatch is corrected by a skilled technician, which is an expensive and highly involved procedure [41].

With the proposed CycleISP model, we can perform the color matching task, as shown in Fig. 8. Given a stereo pair, we first choose one view as the target and apply morphing to fully register it with the source view. Next, we pass the source RGB image through RGB2RAW model and obtain the source RAW image. Finally, we map back the source RAW image to the sRGB space using the RAW2RGB network, but with the color correction branch providing the color information from the '*target*' RGB image (rather than the source RGB). Fig. 9 compares our method with three other color matching techniques [34, 43, 47]. The proposed method generates results that are perceptually more faithful to the target views than other competing approaches.

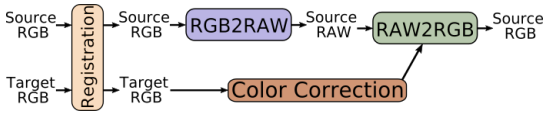
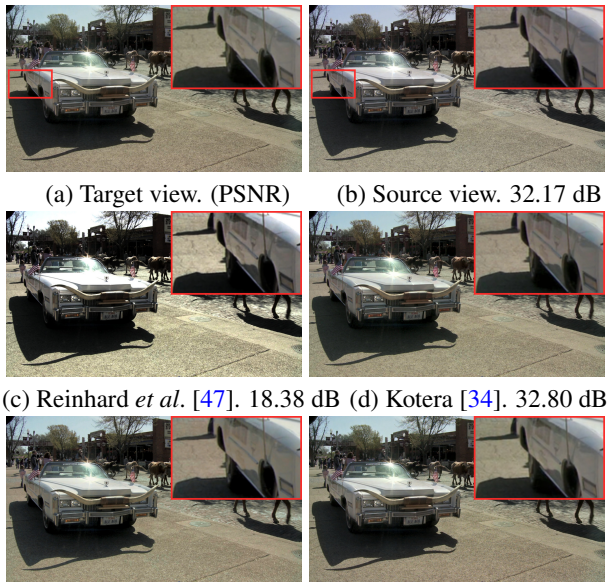


Figure 8: Scheme for color matching 3D pairs.



(a) Target view. (PSNR) (b) Source view. 32.17 dB

(c) Reinhard *et al*. [47]. 18.38 dB (d) Kotera [34]. 32.80 dB

(e) Pitié *et al*. [43]. 33.38 dB (f) Ours. **36.60 dB**

Figure 9: Example of color correction for 3D cinema. Compare the colors of the ground and side of the car in zoomed-in crops. Images are property of Mammoth HD Inc.

7. Conclusion

In this work, we propose a data-driven CycleISP framework that is capable of converting sRGB images to RAW data and back to sRGB images. The CycleISP model allows us to synthesize realistic clean/noisy paired training data both in RAW and sRGB spaces. By training a novel network for the tasks of denoising the RAW and sRGB images, we achieve state-of-the-art performance on real noise benchmark datasets (DND [44] and SIDD [1]). Furthermore, we demonstrate that the CycleISP model can be applied to the color matching problem in stereoscopic cinema. Our future work includes exploring and extending the CycleISP model for other low-level vision problems such as super-resolution and deblurring.

Acknowledgments. Ming-Hsuan Yang is supported by the NSF CAREER Grant 1149783.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 2, 5, 6, 7, 8

[2] Abdelrahman Abdelhamed, Radu Timofte, and Michael S Brown. Ntire 2019 challenge on real image denoising: Methods and results. In *CVPRW*, 2019. 2

[3] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *Trans. Sig. Proc.*, 2006. 6, 7

[4] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. *ICCV*, 2019. 2, 7

[5] Marcelo Bertalmío. *Image Processing for Cinema*. CRC Press, 2014. 8

[6] Marcelo Bertalmío. *Denoising of Photographic Images and Video*. Springer, 2018. 2

[7] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[8] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 2, 6, 7

[9] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *CVPR*, 2012. 2, 5, 6, 7

[10] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*, 2011. 4, 6

[11] Priyam Chatterjee and Peyman Milanfar. Is denoising dead? *TIP*, 2009. 2

[12] Priyam Chatterjee and Peyman Milanfar. Fundamental limits of image denoising: are we there yet? In *ICASSP*, 2010. 2

[13] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *CVPR*, 2018. 4

[14] Yunjin Chen, Wei Yu, and Thomas Pock. On learning optimized reaction diffusion processes for effective image restoration. In *CVPR*, 2015. 6, 7

[15] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *TIP*, 2007. 2, 6, 7

[16] https://noise.visinf.tu-darmstadt.de/benchmark/, 2017. [Online; accessed 15-Nov-2019]. 6

[17] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[18] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *TIP*, 2012. 6, 7

[19] David L Donoho. De-noising by soft-thresholding. *Trans. on information theory*, 1995. 2

[20] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999. 2

[21] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K Mantiuk, and Jonas Unger. HDR image reconstruction from a single exposure using deep cnns. *TOG*, 2017. 4

[22] Alessandro Foi. Clipped noisy images: Heteroskedastic modeling and practical denoising. *Signal Processing*, 2009. 2

[23] Alessandro Foi, Sakari Alenius, Vladimir Katkovnik, and Karen Egiazarian. Noise measurement for raw-data of digital imaging sensors by automatic segmentation of nonuniform targets. *Sensors*, 2007. 2

[24] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *TIP*, 2008. 2

[25] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *TOG*, 2016. 2, 4

[26] Gabriela Ghimpeţeanu, Thomas Batard, Tamara Seybold, and Marcelo Bertalmío. Local denoising applied to raw images may outperform non-local patch-based methods applied to the camera output. In *Electronic Imaging*, 2016. 2, 7

[27] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014. 6, 7

[28] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019. 2, 7

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[30] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 4, 5

[31] Mark J Huiskes, Bart Thomee, and Michael S Lew. New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative. In *ACM MIR*, 2010. 6

[32] Hakki Can Karaimer and Michael S Brown. A software platform for manipulating the camera imaging pipeline. In *ECCV*, 2016. 6

[33] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, and Mohammed Bennamoun. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, 8(1):1–207, 2018. 1

[34] Hiroaki Kotera. A scene-referred color transfer for pleasant imaging on display. In *ICIP*, 2005. 8

[35] Marc Lebrun, Miguel Colom, Antoni Buades, and Jean-Michel Morel. Secrets of image denoising cuisine. *Acta Numerica*, 2012. 2

[36] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. The noise clinic: a blind image denoising algorithm. *Image Processing On Line*, 2015. 7

[37] Anat Levin and Boaz Nadler. Natural image denoising: Optimality and inherent bounds. In *CVPR*, 2011. 2

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[39] Jiaming Liu, Chi-Hao Wu, Yuzhi Wang, Qin Xu, Yuqian Zhou, et al. Learning raw image denoising with bayer pattern unification and bayer preserving augmentation. In *CVPR Workshops*, 2019. 4, 6

[40] Markku Makitalo and Alessandro Foi. Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise. *TIP*, 2012. 6

[41] Bernard Mendiburu. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press, 2009. 2, 8

[42] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *TPAMI*, 1990. 2

[43] François Pitié, Anil C Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Trans. on CVIU*, 2007. 8

[44] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 1, 2, 5, 6, 7, 8

[45] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NeurIPS*, 2018. 1, 2, 6

[46] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew. Color image processing pipeline. *IEEE Signal Processing Magazine*, 2005. 2, 3

[47] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *Trans. on Computer graphics and applications*, 2001. 8

[48] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: a better and simpler baseline. In *CVPR*, 2019. 4

[49] Raquel Gil Rodríguez, Javier Vazquez-Corral, and Marcelo Bertalmío. Color matching images with unknown non-linear encodings. *TIP*, 2020. 2

[50] Stefan Roth and Michael J Black. Fields of experts. *IJCV*, 2009. 6, 7

[51] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 1992. 2

[52] Eli Schwartz, Raja Giryes, and Alex M Bronstein. DeepISP: Towards learning an end-to-end image processing pipeline. *TIP*, 2018. 4

[53] Tamara Seybold, Özlem Cakmak, Christian Keimel, and Walter Stechele. Noise characteristics of a single sensor camera in digital color image processing. In *CIC*, 2014. 2

[54] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 4

[55] Eero P Simoncelli and Edward H Adelson. Noise removal via bayesian wavelet coring. In *ICIP*, 1996. 2

[56] Stephen M Smith and J Michael Brady. SUSAN—a new approach to low level image processing. *IJCV*, 1997. 2

[57] Hossein Talebi and Peyman Milanfar. Global image denoising. *TIP*, 2013. 6, 7

[58] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *ICCV*, 1998. 2

[59] Javier Vazquez-Corral and Marcelo Bertalmío. Color stabilization along time and across shots of the same scene, for one or several cameras of unknown specifications. *TIP*, 2014. 2

[60] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *ECCV*, 2018. 4

[61] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *ECCV*, 2018. 7

[62] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *ICCV*, 2017. 7

[63] Leonid P Yaroslavsky. Local adaptive image restoration and enhancement with the use of DFT and DCT in a running window. In *Wavelet Applications in Signal and Image Processing IV*, 1996. 2

[64] Syed Waqas Zamir, Aditya Arora, Salman Khan, Fahad Shahbaz Khan, and Ling Shao. Learning digital camera pipeline for extreme low-light imaging. *arXiv preprint arXiv:1904.05939*, 2019. 4

[65] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Multi-scale single image dehazing using perceptual pyramid deep network. In *CVPR Workshops*, 2018. 4

[66] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 2, 4, 5, 6, 7

[67] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *TIP*, 2018. 2, 7

[68] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 4

[69] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011. 6, 7