Robust Image Denoising of No-Flash Images Guided by Consistent Flash Images

Geunwoo Oh[1], **Jonghee Back**[1], **Jae-Pil Heo**[2], **Bochang Moon**[1*]

[1]Gwangju Institute of Science and Technology, South Korea
[2]Sungkyunkwan University, South Korea
{gnuo8325, jongheeback}@gm.gist.ac.kr, jaepilheo@skku.edu, bmoon@gist.ac.kr

Abstract

Images taken in low light conditions typically contain distracting noise, and eliminating such noise is a crucial computer vision problem. Additional photos captured with a camera flash can guide an image denoiser to preserve edges since the flash images often contain fine details with reduced noise. Nonetheless, a denoiser can be misled by inconsistent flash images, which have image structures (e.g., edges) that do not exist in no-flash images. Unfortunately, this disparity frequently occurs as the flash/no-flash pairs are taken in different light conditions. We propose a learning-based technique that robustly fuses the image pairs while considering their inconsistency. Our framework infers consistent flash image patches locally, which have similar image structures with the ground truth, and denoises no-flash images using the inferred ones via a combination model. We demonstrate that our technique can produce more robust results than state-of-the-art methods, given various flash/no-flash pairs with inconsistent image structures. The source code is available at https://github.com/CGLab-GIST/RIDFnF.

1 Introduction

Image denoising is a long-standing problem, and its ultimate goal is to recover ground truth images (clean images) from noisy input images. In practice, an image frequently contains visually distracting noise when taken in a low light condition. Image denoisers, e.g., (Tomasi and Manduchi 1998; Buades, Coll, and Morel 2005), often reduce such noise by blending adjacent pixel colors while preserving fine details.

Additional photos taken with a camera flash can effectively guide image denoisers since those typically contain high-frequency details with much-reduced noise (Petschnigg et al. 2004; Eisemann and Durand 2004). Technically, the flash images can serve as edge-stopping functions that prevent image denoisers from blending neighboring pixel colors across edges.

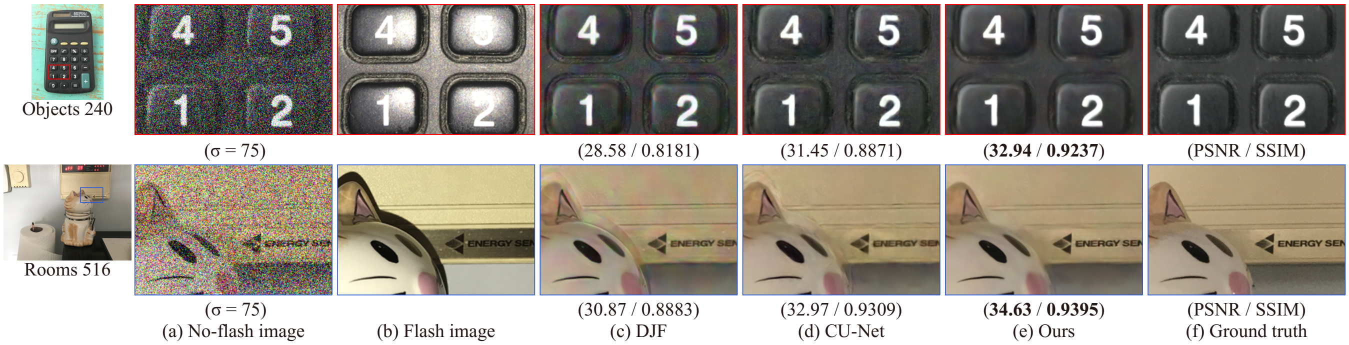
Nevertheless, it is unrealistic to assume that high-frequency information in flash images is perfectly matched to those in ground truth images, i.e., no-flash images without noise, since they are taken in different light conditions (i.e., with and without a camera flash). For example, the flashlight typically introduces additional image structures (e.g., hard shadows or highlights on specular objects) that do not exist in the ground truth image. It makes flash images less helpful in guiding a denoiser to preserve fine details in no-flash images. As a result, suppressing noise in a no-flash image by robustly exploiting an inconsistent flash image, which has image structures different from the ground truth, is a vital technical challenge for image denoisers that take a flash/no-flash image pair as input.

Training a deep neural network that denoises a no-flash image while exploiting an additional flash image has recently received attention. As a recent example, Deng and Dragotti (2021) demonstrated that a sophistically designed neural network, which fuses a flash/no-flash image pair, can drastically outperform existing methods (Kostadin Dabov and Egiazarian 2007; Zhang, Zuo, and Zhang 2018) that take only a no-flash image as input. Nonetheless, when structural information (e.g., image edges) in flash images is inconsistent with that in the corresponding ground truth (e.g., image areas greatly affected by the flashlight), the state-of-the-art techniques (Li et al. 2016; Deng and Dragotti 2021) often fail to generate a high-quality denoising output, as shown in Fig. 1.

We propose a learning-based framework that takes a flash/no-flash pair as input and generates a denoised no-flash image while robustly handling the inconsistency between the input image pair. Our framework is built upon deep combiner (DC) (Back et al. 2020), initially designed for fusing a synthetic image pair (i.e., two rendered images) via a localized combination model. A straightforward adaptation to the original DC is substituting their virtual input pair with flash/no-flash images. However, this direct application can produce less effective results since their combination model assumes the image pair is generated using an identical light condition. We relax their assumption (i.e., a consistent image pair) and reformulate a new combination model that merges a flash/no-flash image pair while taking inconsistency between the input pair into account. Our main contributions are as follows.

- We infer a consistent image patch, which is structurally similar to the ground truth, by applying per-pixel convolutional kernels to an input flash image locally. We train a deep neural network to produce the convolutional kernels so that estimated image patches can adequately

[*]Corresponding author.



Objects 240

(σ = 75) | | (28.58 / 0.8181) | (31.45 / 0.8871) | (**32.94** / **0.9237**) | (PSNR / SSIM)

Rooms 516

(σ = 75) | | (30.87 / 0.8883) | (32.97 / 0.9309) | (**34.63** / **0.9395**) | (PSNR / SSIM)

(a) No-flash image (b) Flash image (c) DJF (d) CU-Net (e) Ours (f) Ground truth

Figure 1: Comparisons with the state-of-the-art techniques (DJF (Li et al. 2016) and CU-Net (Deng and Dragotti 2021)) that take a pair of flash/no-flash images as input. The no-flash images (a) are corrupted by Gaussian noise with $\sigma = 75$. The image regions in the flash images (b) contain specular highlights (in the top row) and hard shadows (in the bottom) that do not exist in the no-flash images ((a) and (f)). These inconsistent flash images make the existing methods ((c) and (d)) generate noticeable artifacts (residual noise and ghosting). On the other hand, our method produces much-reduced artifacts with higher numerical accuracy by robustly exploiting such inconsistent flash images.

guide high-frequency information in a no-flash image.

- We combine a noisy no-flash image and inferred consistent image locally via a new combination model and output a denoised no-flash image.

We demonstrate that our denoising framework, which takes the inconsistency between the flash/no-flash pairs into account, significantly outperforms the state-of-the-art denoisers (Li et al. 2016; Deng and Dragotti 2021) visually and numerically given flash/no-flash image pairs where we add Gaussian noise into no-flash images (Fig. 1). We also show that our technique can suppress real noise in no-flash images more effectively than the previous methods, thanks to our robust handling of imperfect flash images.

2 Related Work

Image denoising without a flash image. The goal of image denoising is to restore a clean image from a noisy image. To this end, it is technically required to suppress random artifacts while retaining high-frequency information (i.e., edges) in the noisy input. Traditionally, hand-crafted filters have been designed, and well-known examples include wavelet shrinkage (Donoho 1995), bilateral filter (Tomasi and Manduchi 1998), anisotropic diffusion filter (Weickert 1998), regression-based reconstruction (Takeda, Farsiu, and Milanfar 2007), and non-local means filtering (Buades, Coll, and Morel 2005). A notable method is block-matching and 3D filtering (BM3D) (Kostadin Dabov and Egiazarian 2007) that collects similar image patches in a local neighborhood and denoises the collected patches together.

Training a deep neural network whose task is to restore a ground truth image has recently received strong attention. As an early attempt, Burger, Schuler, and Harmeling (2012) exploited a multi-layer perceptron for image denoising. Zhang et al. (2017) proposed a denoising convolutional neural network (DnCNN) that exploits residual learning (He et al. 2016) and batch normalization (Ioffe and Szegedy 2015). Later, it was extended to a flexible denoising convolution neural network (FFDNet) so that denoising could perform robustly against various noise levels (Zhang, Zuo, and Zhang 2018). Recently, transformer-based denoising neural networks were proposed to effectively exploit long-range dependencies in images (Wang et al. 2022; Zamir et al. 2022). Additionally, sophisticated neural networks were designed for handling real-noisy images (Anwar and Barnes 2019; Guo et al. 2019) and burst images (Mildenhall et al. 2018; Marinč et al. 2019). Also, it has been demonstrated that a denoising neural network can be trained without ground truth images in unsupervised or self-supervised fashions (Lehtinen et al. 2018; Krull, Buchholz, and Jug 2019; Huang et al. 2021).

We also aim at building a denoising neural network that estimates ground truth images from noisy images, like the learning-based methods. However, our neural network learns an optimal fusion of a pair of flash/no-flash images.

Denoising with a flash image. It has been known that noise in no-flash images can be reduced effectively when additional flash images can guide high-frequency information in clean no-flash photos. Petschnigg et al. (2004) and Eisemann and Durand (2004) are seminal works that reduce noise in no-flash images while minimizing excessive overblurring artifacts using the additional information (i.e., flash images). Inspired by the pioneering studies, more denoising approaches have been designed, such as guided filtering (He, Sun, and Tang 2013), denoising with a dark flash (Krishnan and Fergus 2009), and robust filtering with a scale map that encodes structural inconsistency between a pair of flash and no-flash images (Yan et al. 2013).

Recently a deep neural network has been actively utilized to fuse a flash/no flash image pair. Li et al. (2016) built a convolutional neural network that restores a clean no-flash image using a flash image, and Xia et al. (2021) proposed a kernel-predicting network that can combine flash/no-flash pixel colors robustly while addressing misalignments between the two input images. Besides, Deng and Dragotti (2021) presented the common and unique information splitting network (CU-Net) for fusing different

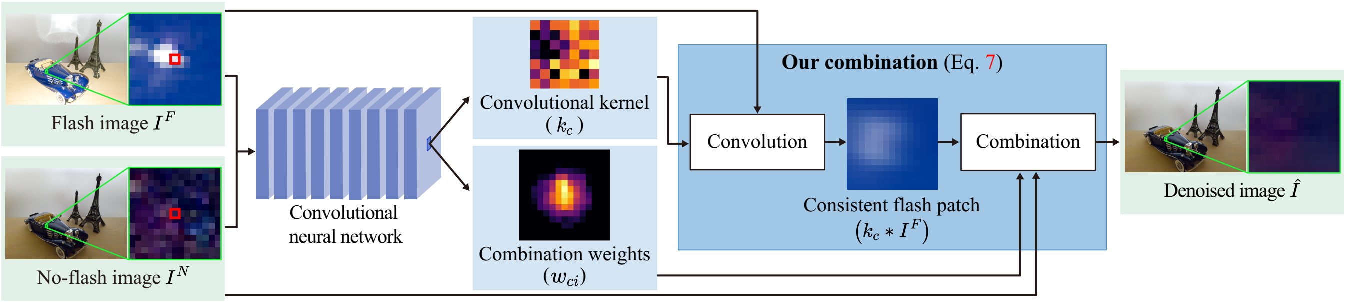

Flash image I^F
No-flash image I^N
Convolutional neural network
Convolutional kernel (k_c)
Combination weights (w_{ci})
Our combination (Eq. 7)
Convolution
Combination
Consistent flash patch ($k_c * I^F$)
Denoised image $\hat{I}$

Figure 2: Our denoising framework takes a flash/no-flash image pair (I^F and I^N) as input and outputs a denoised no-flash image $\hat{I}$ while considering the lighting discrepancy between the input pair. We exploit a convolutional neural network that infers two types of parameters of our combination model, a convolution kernel k_c and combination weights w_{ci} per pixel c (marked as a small red box in I^F and I^N). We infer a consistent flash image with similar image structures to the ground truth via a convolution between the input flash image and the estimated kernel k_c. We then combine the no-flash image and the inferred one using combination weights w_{ci} for producing the output pixel estimates $\hat{I}$.

source images and demonstrated outstanding denoising results using flash/no-flash image pairs.

In this paper, we also present a learning-based framework that combines a flash/no-flash image pair taken in different light conditions (i.e., with and without a camera flash). However, the key distinction from the existing learning-based techniques is that our combination framework explicitly handles the discrepancy between the input image pair by inferring consistent image patches locally using per-pixel convolutional kernels.

3 Combination of Flash/No-Flash Images

Our goal is to restore the (unknown) clean image I from a noisy no-flash image I^N using an additional flash image I^F. Note that the two input images (i.e., a flash/no-flash pair) are captured in different light conditions, i.e., with and without a camera flash. We statistically model the i-th pixel color I_i^N in the no-flash image I^N as

$$I_i^N = I_i + \epsilon_i, \tag{1}$$

where the expectation of the error ϵ_i is assumed to be zero, i.e., $E[\epsilon_i] = 0$.

To produce an estimate $\hat{I}$ of the ground truth I from the input pair (I^F and I^N), we propose a new denoising framework (Fig. 2) built upon deep combiner (DC) (Back et al. 2020). In Sec. 3.1, we introduce an adapted deep combiner that takes a flash/no-flash pair in the place of their original inputs (i.e., a rendered image pair) and motivate our technique that takes structural inconsistency between the image pair into account. We then propose a reformulated combination model that allows us to robustly denoise a no-flash image while utilizing an inconsistent flash image in Sec. 3.2.

3.1 Deep Combiner for Rendered Image Pairs

Deep combiner (DC) (Back et al. 2020) is a deep neural network for combining a rendered image pair with statistically different properties. Specifically, the first input to the DC is a noisy image corrupted by a random error, like the assumption for a no-flash image (Eq. 1). On the other hand, the second is a correlated image that can guide image structure (e.g., edges) in the ground truth. For example, an image can be ideal when it has all high-frequency information in the ground truth and does not include noise.

A straightforward adaptation to the DC is to take a pair of flash/no-flash images as its input instead of their original input (i.e., a rendered image pair). Technically, it corresponds to modeling that the pixel colors in a flash image I^F are linearly correlated to those in the ground truth I:

$$I_c^F - I_i^F = I_c - I_i + \epsilon_{ci}, \tag{2}$$

where I_c^F and I_i^F are the c-th and i-th pixel colors in the flash image I^F. Also, ϵ_{ci} is an error term that varies locally depending on the difference between $I_c^F - I_i^F$ and $I_c - I_i$. Unlike the assumption for a no-flash image I^N (Eq. 1), the color difference $I_c^F - I_i^F$ is exploited to approximate the unknown difference $I_c - I_i$.

A camera flash typically increases the overall brightness of the image, and thus the pixel colors in a flash image can be brighter than the ground truth colors. However, this global discrepancy does not mainly affect the error term ϵ_{ci} as it models the color difference instead of the color itself. On the other hand, the error ϵ_{ci} increases when the flash image has image structures that do not exist in the ground truth image, e.g., hard shadows or specular highlights introduced by a camera flash.

Given the statistical models for the two input images (Eqs. 1 and 2), a localized objective function J_c for estimating the c-th pixel color I_c in the ground truth image I is defined in a least-squares sense:

$$J_c = \frac{1}{2} w_{cc} \left(I_c^N - \hat{I}_c \right)^2 + \sum_{i \in \Omega_c, i \neq c} w_{ci} \left(I_i^N - \hat{I}_i \right)^2 + \sum_{i \in \Omega_c, i \neq c} w_{ci} \left\{ \left(I_c^F - I_i^F \right) - \left(\hat{I}_c - \hat{I}_i \right) \right\}^2, \tag{3}$$

where Ω_c is a set of neighboring pixels within an image window centered at pixel c (e.g., 15×15). w_{cc} and w_{ci} are posi-

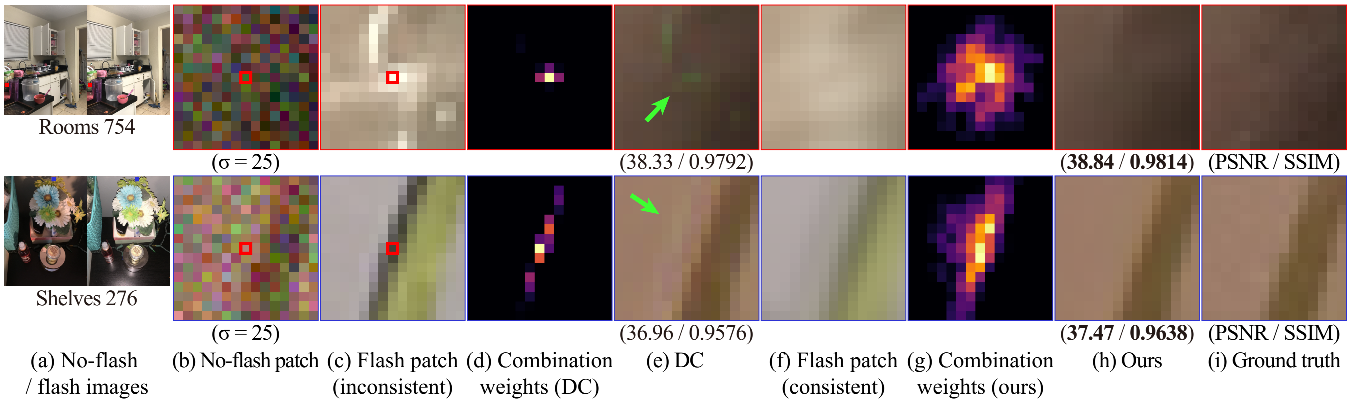


Figure 3: Denoising results of DC (e) and our method (h), together with their visualized combination weights, (d) and (g) for a center pixel c (marked as a small red box in (b) and (c)). The input flash patches (c) have image structures inconsistent with their ground truth (i) due to the specular highlights (in the top row) and hard shadows (in the bottom). DC assigns high combination weights only for small numbers of neighboring pixels (in (d)) to avoid an excessive denoising bias, and it results in residual noise in their results (e). On the other hand, our method infers consistent flash patches (f) that have similar image structures to the ground truth, and it allows us to generate improved denoising results (h) by exploiting more neighboring pixels (g).

tive weights assigned to the center pixel c and i-th neighboring pixel. By setting the gradient of J_c with respect to $\hat{I}_c$ and $\hat{I}_i$ to zero, a combination model that minimizes the objective function can be derived into

$$\hat{I}_c = \frac{1}{\sum\limits_{i \in \Omega_c} w_{ci}} \sum_{i \in \Omega_c} w_{ci} \left\{ I_i^N + \left(I_c^F - I_i^F \right) \right\}, \tag{4}$$

which produces an output estimate $\hat{I}_c$ at pixel c.

Intuitively, the combined estimate (Eq. 4) is a weighted average of the neighboring colors, I_i^N and $(I_c^F - I_i^F)$, and the combination weights w_{ci} control its estimation accuracy. For example, relatively large weights should be allocated into neighboring pixels i with small approximation errors ϵ_{ci} (in Eq. 2), i.e., the pixels i which have similar $I_c^F - I_i^F$ to the unknown difference $I_c - I_i$. We refer to the original paper (Back et al. 2020) that analyzes the formulation theoretically.

The estimation process (Eq. 4) is conducted per pixel independently, and thus the combination weights w_{ci} should be determined properly for each pixel c so that the output estimate becomes close to the ground truth. DC employs a convolutional neural network to determine the weights.

Our motivation. The original DC (Back et al. 2020) devised the combination model (Eq. 4) for fusing a pair of synthetic images rendered using the same lighting. Thus their statistical models for the inputs (Eqs. 1 and 2) could work reasonably well. For example, it can reduce the noise in I^N effectively when it is possible to exploit enough numbers of neighboring pixels i with minor approximation errors ϵ_{ci} by allocating high combination weights to such pixels. However, in our case, a flash image can be highly inconsistent with the ground truth no-flash image, e.g., regions where specular highlights or hard shadows exist only in the flash image in Fig. 3.

For the inconsistent areas, the DC should exploit only small numbers of neighboring pixels whose modeling errors (ϵ_{ci}) are minor since allocating high combination weights to the pixels with large errors can lead to an excessive bias in their combined estimates. It can lead to ineffective denoising results (e.g., under-blurred results), as shown in Fig. 3. It motivates us to propose a new combination model that robustly combines a potentially inconsistent pair of flash/no-flash images by letting a neural network infer a consistent flash image whose structures are similar to those in the ground truth image (Sec. 3.2).

3.2 Our Combination Model Using Locally Consistent Flash Images

Our key idea is to infer consistent flash image patches, which have image structures matched with corresponding ground truth patches, for denoising no-flash pixel colors robustly instead of directly relying on potentially inconsistent flash images. To this end, we define a new statistical model for the flash pixel colors as

$$\left(k_c * I^F\right)_c - \left(k_c * I^F\right)_i = I_c - I_i + \epsilon_{ci}, \tag{5}$$

where $k_c * I^F$ is the convolutional result between a convolutional kernel k_c of size $K \times K$ at pixel c and I^F, and $(\cdot)_c$ and $(\cdot)_i$ are the c-th and i-th pixel colors in the $k_c * I^F$, respectively.

Our main distinction from the previous model (Eq. 2) is that we have a technical gadget (i.e., the kernel k_c) to mitigate the inconsistency, ϵ_{ci}. Note that the error term ϵ_{ci} in the previous one (Eq. 2) is a constant (and thus noncontrollable). However, our model (Eq. 5) allows us to adjust the kernel k_c at each pixel c in order to lessen the modeling error. Specifically, we set the kernel k_c to a normalized kernel whose elements are non-negative.

Given the statistical model for the no-flash image (Eq. 1) and newly formulated one for the flash image (Eq. 5), we

Noise Level	Method	BM3D	FFDNet	Uformer-B	Restormer	DJF	CU-Net	DC	Ours
$\sigma = 25$	PSNR ↑	35.72	36.19	36.58	36.32	34.51	36.47	36.75	**37.09**
	SSIM ↑	0.9621	0.9654	0.9681	0.9665	0.9524	0.9673	0.9684	**0.9710**
$\sigma = 50$	PSNR ↑	32.65	33.50	34.04	33.88	32.11	33.87	34.71	**35.02**
	SSIM ↑	0.9348	0.9461	0.9515	0.9500	0.9295	0.9506	0.9558	**0.9595**
$\sigma = 75$	PSNR ↑	30.89	31.82	32.31	32.60	30.36	32.29	33.31	**33.62**
	SSIM ↑	0.9120	0.9298	0.9361	0.9400	0.9082	0.9369	0.9442	**0.9491**

Table 1: PSNR/SSIM averages of the tested methods for 256 test images corrupted by Gaussian noise.

define an objective function as

$$J_c = \frac{1}{2} w_{cc} \left(I_c^N - \hat{I}_c\right)^2 + \sum_{i \in \Omega_c, i \neq c} w_{ci} \left(I_i^N - \hat{I}_i\right)^2 + \sum_{i \in \Omega_c, i \neq c} w_{ci} \left[\left\{\left(k_c * I^F\right)_c - \left(k_c * I^F\right)_i\right\} - \left(\hat{I}_c - \hat{I}_i\right)\right]^2. \tag{6}$$

Analogously in the original DC (Sec. 3.1), this cost function can be minimized in the least-squares sense by setting the gradients of J_c with respect to $\hat{I}_c$ and $\hat{I}_i$ to zero. It leads to a new combination model (see the supplementary report for the derivation):

$$\hat{I}_c = \frac{1}{\sum\limits_{i \in \Omega_c} w_{ci}} \sum_{i \in \Omega_c} w_{ci} \left\{I_i^N + \left(k_c * I^F\right)_c - \left(k_c * I^F\right)_i\right\}, \tag{7}$$

which locally averages the pixel colors in a no-flash and inferred consistent flash image (i.e., $k_c * I^F$) using the combination weights w_{ci}. Note that it is not necessary to perform the convolution for all pixels in I^F since we only access the neighboring pixels ($i \in \Omega_c$) in the $k_c * I^F$. As a result, the required number of the convolutions is only $|\Omega_c|$ per pixel c.

To estimate the $\hat{I}_c$ using our combination model (Eq. 7), we should determine both combination weights w_{ci} and a convolutional kernel k_c per pixel c. Our framework uses a plain convolutional neural network that produces those combination parameters (see Fig. 2). We train the network with learnable parameters θ toward the optimal $\hat{\theta}$ by minimizing the following supervised l_2 loss:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{3n} \sum_{c=1}^{n} ||\hat{I}_c - I_c||^2, \tag{8}$$

where n is the total number of pixels in I.

Our modification to the DC (Sec. 3.1) can be considered conceptually simple, but our combination model is capable of handling the inconsistency (i.e., ϵ_{ci} in Eq. 5) in a flash image through the convolutional kernel k_c. This new technical gadget allows us to produce a robust denoising output, even when a flash image region contains image structures dissimilar to the ground truth, as shown in Fig. 3.

Network details. Our denoising framework exploits a plain convolutional neural network (Fig. 2) that takes a flash/no-flash image pair as input and produces the combination parameters (k_c and w_{ci}) per pixel. Precisely, the network consists of nine convolutional layers, and each of which uses 80 filters of size 5×5, except for the last one that uses $K \times K + |\Omega_c|$ filters to generate the k_c and w_{ci}, respectively. For the per-pixel kernel k_c, we normalize the kernel so that the sum of its elements becomes one. We use the rectified linear activation function (ReLU) for each convolutional layer. The sizes of the neighboring pixels Ω_c and convolutional kernel k_c are set to 15×15 and 7×7, respectively. We include an analysis of the size of the convolutional kernel k_c in Sec. 4. Given this configuration, the total number of trainable parameters of the network is approximately 1.84M.

4 Results and Discussion

We compare our denoising technique with state-of-the-art image denoisers, block-matching and 3D filtering (BM3D) (Kostadin Dabov and Egiazarian 2007), flexible denoising convolution neural network (FFDNet) (Zhang, Zuo, and Zhang 2018), Uformer-B (Wang et al. 2022), and Restormer (Zamir et al. 2022), which take only a no-flash image as input. We also test recent image denoisers, deep joint image filtering (DJF) (Li et al. 2016), common and unique information splitting network (CU-Net) (Deng and Dragotti 2021), and deep combiner (DC) (in Sec. 3.1), which use a pair of flash/no-flash images. All the tested methods, including ours, assume the same noise model, i.e., an additive noise whose expectation is zero. We report the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) (Wang et al. 2004) as numerical measures.

Training and test dataset. We have trained the learning-based methods (FFDNet, Uformer-B, Restormer, DJF, CU-Net, DC, and ours) using the public dataset (Aksoy et al. 2018), which includes 2775 flash/no-flash image pairs categorized into six classes: people, shelves, plants, toys, rooms, and objects. Specifically, we have randomly divided the data set into three sets: 2263 images for the training, 256 images for the validation, and the other 256 images for the test.

Training details. Given the dataset, we have added three levels of Gaussian noise ($\sigma = 25, 50,$ and 75) into no-flash images. Then, we have trained our neural network using Adam optimizer (Kingma and Ba 2015) for 50 epochs, and it has taken 72 hours given two NVIDIA TITAN RTX graphics cards. The initial learning rate has been set to 0.0005 and reduced to 0.0001 after 15 epochs. We have used 64×64 image patches and set the batch size to 64.

For the other learning-based methods, we have used the public implementations released by the respective authors

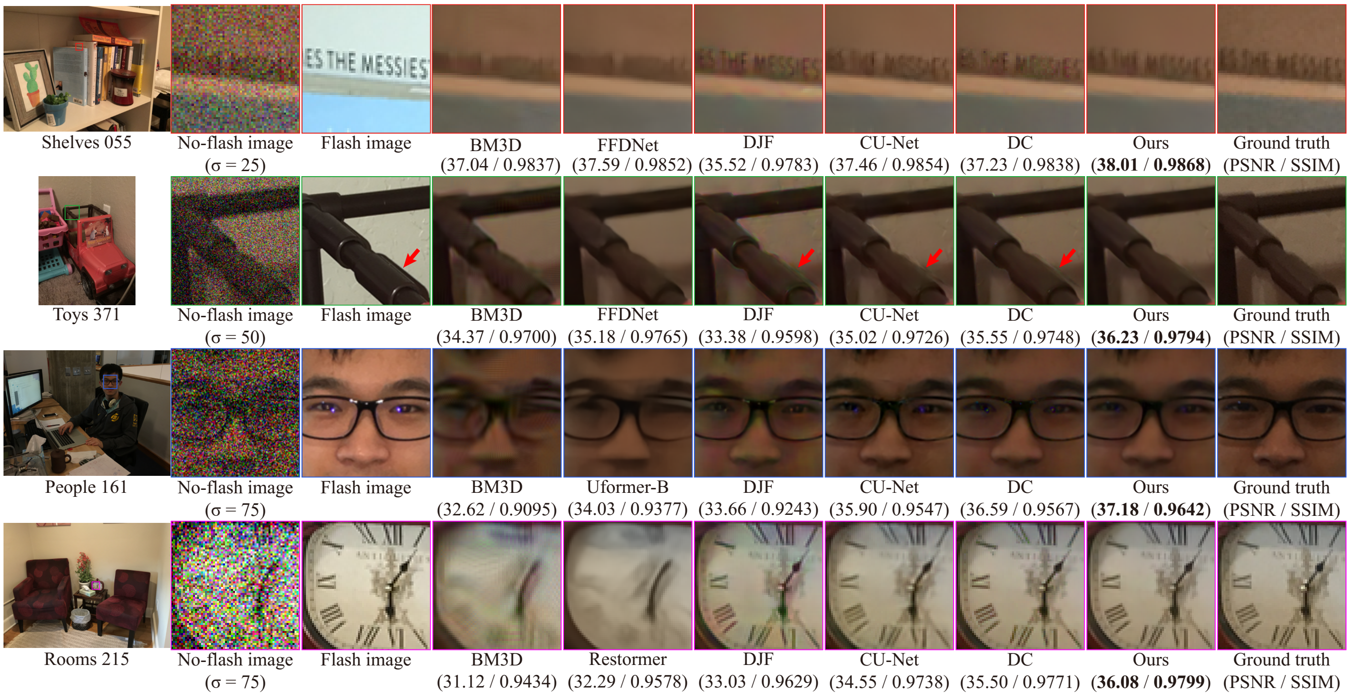


Figure 4: Visual comparisons of denoising techniques for noisy no-flash images corrupted by Gaussian noise. The single image denoisers (BM3D, FFDNet, Uformer-B, and Restormer), which use only no-flash images, tend to produce over-blurred artifacts. On the other hand, the other methods, which exploit flash images additionally, generate relatively sharp denoising results. Nevertheless, the existing methods (DJF, CU-Net, and DC) leave denoising artifacts (e.g., ghosting or residual noise) when flash pixel colors are inconsistent with the ground truth (e.g., hard shadows or specular highlights). Our technique addresses the inconsistency robustly and produces more visually pleasing results with the best PSNR and SSIM values.

and allocated enough time until their networks converge. Additionally, for DJF and CU-Net, we observed that training a separate network for each noise level could produce substantially higher denoising outputs than training a single network. Thus, for a fairer comparison, we have trained three different neural networks for DJF and CU-Net, respectively. On the other hand, we have used only a single network for DC and our technique, respectively.

For a fairer comparison between DC and our network, we have enlarged the network size of the DC by allocating more convolutional filters (e.g., 84 filters except for the last layer) so that its network size (1.89M learnable parameters) becomes similar to that of our method (1.84M).

Comparisons using Gaussian noise. We add Gaussian noise ($\sigma = 25, 50,$ and 75) to the no-flash images in the test set and feed the noisy images (with flash images) to all tested denoising methods. We show the PSNR/SSIM averages of their denoising results in Table 1 and compare the results visually in Fig. 4. BM3D, FFDNet, Uformer-B, and Restormer tend to produce over-blurred results (see Fig. 4) since restoring fine details using only the noisy no-flash images is technically challenging. DJF, CU-Net, and DC produce relatively sharp denoising results but introduce image artifacts, especially when flash images have image structures inconsistent with the ground truth images, e.g., hard shadows for the Toys 371 and specular highlights for the People 161 in Fig. 4. On the other hand, our technique handles such inconsistent flash images robustly and produces fewer visual artifacts while preserving high-frequency details well. This robustness allows our method to produce more accurate results (both PSNR/SSIM metrics) than the existing techniques, as shown in Table 1.

Comparisons using real noise. While all the tested methods assume a simple noise model (e.g., an additive noise such as Gaussian noise), evaluating the techniques for real noise scenarios is interesting. In particular, we have trained the learning-based methods (DJF, CU-Net, DC, and ours) using the synthetic noise, and thus this test can verify whether those can be generalized to real noise scenarios. For this real-noise test, we have taken flash/no-flash image pairs in low light conditions using a mobile camera of Galaxy S21+. Precisely, we have used the built-in camera app of the mobile phone for taking the images while fixing the ISO to 1600 and using the exposure times automatically adjusted by the app. Note that we have trained separate neural networks for DJF and CU-Net, and thus we have chosen the ones trained using Gaussian noise with $\sigma = 75$, which produce the best visual quality for the methods. We use the single networks for DC and ours.

As shown in Fig. 5, the single image denoisers (BM3D and FFDNet) tend to over-smooth high-frequency information, and the state-of-the-art denoisers (DJF and CU-Net)

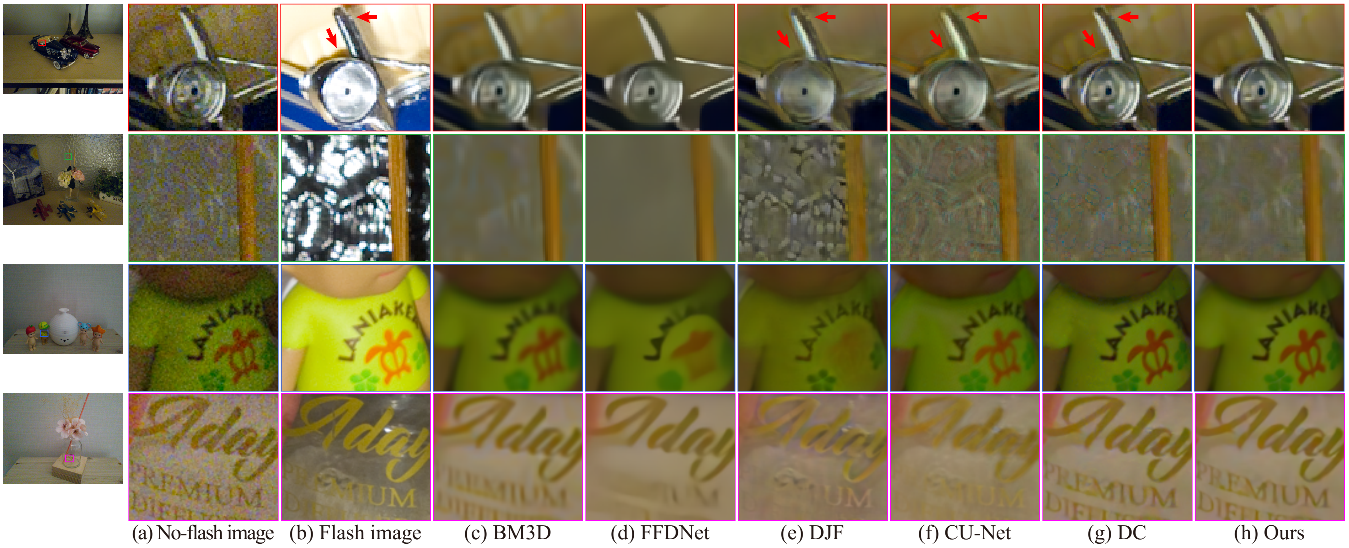


Figure 5: Qualitative comparisons of denoising methods for no-flash images with real noise. BM3D (c) and FFDNet (d) suppress the noise in the no-flash input images (a) but tend to generate over-blurred results. DJF, CU-Net, and DC ((e) to (g)) produce relatively sharp results but suffer from image artifacts caused by the structural inconsistency between the input pair ((a) and (b)), e.g., hard shadows (in the first row) and specular highlights (in the second row) only in the flash input (b). On the other hand, our technique (h) reduces the artifacts effectively thanks to our combination model that considers the inconsistency.

and DC leave some noticeable artifacts (e.g., ghosting or residual noise). Our technique, however, produces more visually pleasing results while robustly handling the lighting discrepancy between the input image pairs. In the supplemental report, we include more test images corrupted by real noise and an additional comparison with (Xia et al. 2021), which uses a different noise model.

Ablation studies with different consistent flash generations. As a design choice, we have exploited a localized kernel (i.e., the convolutional kernel k_c) to fulfill our key idea, i.e., inferring a consistent image that a network can optimize. However, one may consider alternatives that can estimate such a consistent image. We implemented the alternatives for generating consistent flash images by adjusting the last convolutional layer, and Table 2 compares our choice with the two alternatives (Gaussian and Direct).

For the Gaussian, we let the last layer produce the bandwidth of a Gaussian filter per pixel and apply the filter to the flash image to construct a consistent flash image. Specifically, the convolutional kernel k_c in Eq. 7 is replaced with the Gaussian at pixel c. Also, for the Direct, the last convolutional layer produces consistent flash images without forming an intermediate kernel, and the $(k_c * I^F)_c$ and $(k_c * I^F)_i$ in Eq. 7 are replaced with the colors of pixel c and i in the image produced by the last layer.

Table 2 shows the PSNR averages of the original DC and our approach with different design choices (Gaussian, Direct, and Convolutional kernel k_c). As shown in the table, our approach that transforms the input flash image into a learnable one produces more accurate results than the DC since it can allow a neural network to handle discrepancies between the input flash/no-flash image pair. Also, our current choice with the convolutional kernel k_c produces more accurate denoising results than the alternatives.

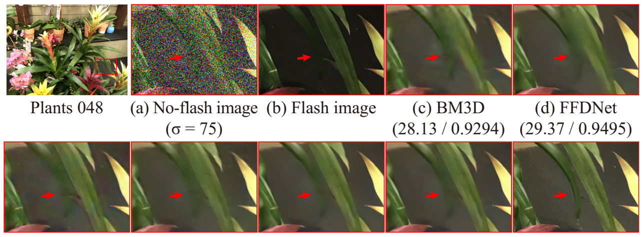
Methods	DC PSNR ↑	Gaussian PSNR ↑	Direct PSNR ↑	Convolutional k_c PSNR ↑
$\sigma = 25$	36.75	36.98	37.02	**37.09**
$\sigma = 50$	34.71	34.92	34.89	**35.02**
$\sigma = 75$	33.31	33.54	33.45	**33.62**

Table 2: PSNR comparisons between DC and our method with different design choices (Gaussian, Direct, and Convolutional kernel k_c), measured using the 256 image pairs.

Analysis of convolutional kernel sizes. Our combination model exploits a convolutional kernel k_c of size $K \times K$ so that a consistent image patch per pixel c can be generated by the convolution between the kernel and the flash image. In Table 3, we measure the average PSNR values of our denoising results using the 256 image pairs in the test set while varying the kernel size $K \times K$ from 1×1 to 9×9. Note that our combination model (Eq. 7) with the smallest kernel size (i.e., 1×1) is equivalent to the existing model of DC (Back et al. 2020), which does not exploit the convolutional kernel, since, in this case, the element of the kernel is always one due to the normalization. As shown in Table 3, there is a noticeable quality improvement when increasing the kernel size from 1×1 to 3×3. It indicates that our localized convolution using a per-pixel kernel (with a size bigger than 1×1) plays a vital role in addressing the inconsistency between the input pair. It can also be seen that the increase of the kernel size from 7×7 to 9×9 does not introduce a noticeable improvement. On the other hand, the inference time

Kernel size	1×1	3×3	5×5	7×7	9×9
	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑
$\sigma = 25$	36.75	37.02	37.07	37.09	**37.10**
$\sigma = 50$	34.69	34.97	35.00	**35.02**	**35.02**
$\sigma = 75$	33.29	33.58	33.61	**33.62**	33.61
Inference time	0.76 s	0.78 s	1.12 s	1.70 s	2.66 s

Table 3: Numerical accuracy of our technique with different convolutional kernel sizes.


Plants 048 (a) No-flash image ($\sigma = 75$) (b) Flash image (c) BM3D (28.13 / 0.9294) (d) FFDNet (29.37 / 0.9495)

(e) DJF (26.85 / 0.9166) (f) CU-Net (28.37 / 0.9382) (g) DC (30.01 / 0.9533) (h) Ours (**30.21 / 0.9561**) (i) Ground truth (PSNR / SSIM)

Figure 6: A failure case of the denoising methods. The flash image (b) completely misses the high-frequency information in the ground truth (see the regions marked by a red arrow). For this extreme case, the flash image does not guide a denoiser to preserve the missing edges, and all the techniques (DJF, CU-Net, DC, and ours), which rely on the problematic flash image, produce over-smoothed results, like the single-denoisers (BM3D and FFDNet).

for a test image pair with 1440×1080 resolution increases as the kernel size becomes large. As a result, we choose the kernel size to 7×7 for all tests.

Limitations and future work. A technical limitation of our technique is that our denoising quality mainly relies on the input flash image like the other denoising methods that take the pair of flash/no-flash images. When a flash image does not capture high-frequency details in the ground truth no-flash image, the benefit from exploiting the additional flash image can disappear. Fig. 6 shows this worst-case scenario where our method does not preserve the edges, and all the denoisers using flash images (DJF, CU-Net, DC, and ours) show over-blurred results like the single-image denoisers (BM3D and FFDNet). It would be interesting to investigate an effective means of inferring an improved consistent image with similar image structures to the ground truth, even for this extreme scenario.

Additionally, we do not explicitly model a misalignment between the input image pair. As analyzed in the supplementary report, our kernel-based approach can handle moderate misalignments, e.g., smaller than the convolutional kernel size (7×7). However, significant shifts between the input pair can introduce ghosting artifacts like the other tested methods (DJF, CU-Net, and DC). Incorporating deformable convolution (Dai et al. 2017) into our framework could be a means for handling such severe misalignments, and we leave it as future research.

5 Conclusion

This paper has proposed a denoising framework that effectively reduces random artifacts in no-flash images while exploiting inconsistent flash images robustly. As our central idea, we have designed a new combination model that fuses a flash/no-flash image pair while mitigating structural inconsistency between the input image pair using per-pixel convolutional kernels. We have demonstrated that our framework, which infers consistent image patches structurally similar to the ground truth, allows producing high-quality denoising outputs while reducing unpleasant denoising artifacts compared to state-of-the-art methods for various test images corrupted by Gaussian and real noise.

Acknowledgments

We appreciate the anonymous reviewers for their constructive comments and suggestions. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (no. 2022-0-00566, the development of object media technology based on multiple video sources).

References

Aksoy, Y.; Kim, C.; Kellnhofer, P.; Paris, S.; Elgharib, M.; Pollefeys, M.; and Matusik, W. 2018. A Dataset of Flash and Ambient Illumination Pairs from the Crowd. In *Eur. Conf. Comput. Vis.*, 644–660.

Anwar, S.; and Barnes, N. 2019. Real Image Denoising With Feature Attention. In *Int. Conf. Comput. Vis.*, 3155–3164.

Back, J.; Hua, B.-S.; Hachisuka, T.; and Moon, B. 2020. Deep Combiner for Independent and Correlated Pixel Estimates. *ACM Trans. Graph.*, 39(6): 1–12.

Buades, A.; Coll, B.; and Morel, J.-M. 2005. A non-local algorithm for image denoising. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 60–65.

Burger, H. C.; Schuler, C. J.; and Harmeling, S. 2012. Image denoising: Can plain neural networks compete with BM3D? In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2392–2399.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable Convolutional Networks. In *Int. Conf. Comput. Vis.*, 764–773.

Deng, X.; and Dragotti, P. L. 2021. Deep Convolutional Neural Network for Multi-Modal Image Restoration and Fusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10): 3333–3348.

Donoho, D. 1995. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3): 613–627.

Eisemann, E.; and Durand, F. 2004. Flash Photography Enhancement via Intrinsic Relighting. *ACM Trans. Graph.*, 23(3): 673–678.

Guo, S.; Yan, Z.; Zhang, K.; Zuo, W.; and Zhang, L. 2019. Toward Convolutional Blind Denoising of Real Photographs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1712–1722.

He, K.; Sun, J.; and Tang, X. 2013. Guided Image Filtering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(6): 1397–1409.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 770–778.

Huang, T.; Li, S.; Jia, X.; Lu, H.; and Liu, J. 2021. Neighbor2Neighbor: Self-Supervised Denoising from Single Noisy Images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 14776–14785.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, 448–456.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Int. Conf. Learn. Represent.*

Kostadin Dabov, V. K., Alessandro Foi; and Egiazarian, K. 2007. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Process.*, 16(8): 2080–2095.

Krishnan, D.; and Fergus, R. 2009. Dark Flash Photography. *ACM Trans. Graph.*, 28(3): 1–11.

Krull, A.; Buchholz, T.-O.; and Jug, F. 2019. Noise2Void - Learning Denoising From Single Noisy Images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2124–2132.

Lehtinen, J.; Munkberg, J.; Hasselgren, J.; Laine, S.; Karras, T.; Aittala, M.; and Aila, T. 2018. Noise2Noise: Learning Image Restoration without Clean Data. In *Int. Conf. Mach. Learn.*, volume 80, 2971–2980.

Li, Y.; Huang, J.-B.; Narendra, A.; and Yang, M.-H. 2016. Deep Joint Image Filtering. In *Eur. Conf. Comput. Vis.*, 154–169.

Marinč, T.; Srinivasan, V.; Gül, S.; Hellge, C.; and Samek, W. 2019. Multi-Kernel Prediction Networks for Denoising of Burst Images. In *IEEE Int. Conf. Image Process.*, 2404–2408.

Mildenhall, B.; Barron, J. T.; Chen, J.; Sharlet, D.; Ng, R.; and Carroll, R. 2018. Burst Denoising With Kernel Prediction Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2502–2510.

Petschnigg, G.; Szeliski, R.; Agrawala, M.; Cohen, M.; Hoppe, H.; and Toyama, K. 2004. Digital Photography with Flash and No-Flash Image Pairs. *ACM Trans. Graph.*, 23(3): 664–672.

Takeda, H.; Farsiu, S.; and Milanfar, P. 2007. Kernel Regression for Image Processing and Reconstruction. *IEEE Trans. Image Process.*, 16(2): 349–366.

Tomasi, C.; and Manduchi, R. 1998. Bilateral filtering for gray and color images. In *Int. Conf. Comput. Vis.*, 839–846.

Wang, Z.; Bovik, A.; Sheikh, H.; and Simoncelli, E. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4): 600–612.

Wang, Z.; Cun, X.; Bao, J.; Zhou, W.; Liu, J.; and Li, H. 2022. Uformer: A General U-Shaped Transformer for Image Restoration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 17662–17672.

Weickert, J. 1998. *Anisotropic Diffusion in Image Processing*. Teubner Stuttgart.

Xia, Z.; Gharbi, M.; Perazzi, F.; Sunkavalli, K.; and Chakrabarti, A. 2021. Deep Denoising of Flash and No-Flash Pairs for Photography in Low-Light Environments. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2063–2072.

Yan, Q.; Shen, X.; Xu, L.; Zhuo, S.; Zhang, X.; Shen, L.; and Jia, J. 2013. Cross-Field Joint Image Restoration via Scale Map. In *Int. Conf. Comput. Vis.*, 1537–1544.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; and Yang, M. 2022. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 5718–5729.

Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.*, 26(7): 3142–3155.

Zhang, K.; Zuo, W.; and Zhang, L. 2018. FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. *IEEE Trans. Image Process.*, 27(9): 4608–4622.