A High-Quality Denoising Dataset for Smartphone Cameras

Abdelrahman Abdelhamed
York University
kamel@eecs.yorku.ca

Stephen Lin
Microsoft Research
stevelin@microsoft.com

Michael S. Brown
York University
mbrown@eecs.yorku.ca

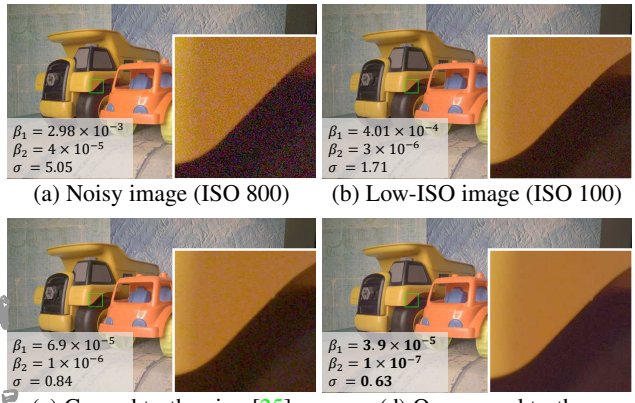
Abstract

The last decade has seen an astronomical shift from imaging with DSLR and point-and-shoot cameras to imaging with smartphone cameras. Due to the small aperture and sensor size, smartphone images have notably more noise than their DSLR counterparts. While denoising for smartphone images is an active research area, the research community currently lacks a denoising image dataset representative of real noisy images from smartphone cameras with high-quality ground truth. We address this issue in this paper with the following contributions. We propose a systematic procedure for estimating ground truth for noisy images that can be used to benchmark denoising performance for smartphone cameras. Using this procedure, we have captured a dataset – the Smartphone Image Denoising Dataset (SIDD) *– of ~30,000 noisy images from 10 scenes under different lighting conditions using five representative smartphone cameras and generated their ground truth images. We used this dataset to benchmark a number of denoising algorithms. We show that CNN-based methods perform better when trained on our high-quality dataset than when trained using alternative strategies, such as low-ISO images used as a proxy for ground truth data.*

1. Introduction

With over 1.5 billion smartphones sold annually,[1] it is unsurprising that smartphone images now vastly outnumber images captured with DSLR and point-and-shoot cameras. But while the prevalence of smartphones makes them a convenient device for photography, their images are typically degraded by higher levels of noise due to the smaller sensors and lenses found in their cameras. This problem has heightened the need for progress in image denoising, particularly in the context of smartphone imagery.

A major issue towards this end is the lack of an established benchmarking dataset for real image denoising representative of smartphone cameras. The creation of such a dataset is essential both to focus attention on denoising of smartphone images and to enable standardized evaluations of denoising techniques. However, many of the approaches used to produce noise-free ground truth images are not fully sufficient, especially for the case of smartphone cameras. For example, the common strategy of using low ISO and long exposure to acquire a "noise free" image [2, 26] is not applicable to smartphone cameras, as noise is still significant on such images even with the best camera settings (e.g., see Figure 1). Recent work in [25] moved in the right direction by globally aligning and post-processing low-ISO images to match their high-ISO counterparts. This approach gives excellent performance on DSLR cameras; however, it is not entirely applicable to smartphone images. In particular, post-processing of a low-ISO image does not sufficiently remove the remaining noise, and the reliance on a global translational alignment has proven inadequate for aligning smartphone images.



(a) Noisy image (ISO 800) (b) Low-ISO image (ISO 100)

(c) Ground truth using [25] (d) Our ground truth

Figure 1: An example scene imaged with an LG *G4* smartphone camera: (a) a high-ISO noisy image; (b) same scene captured with low ISO – this type of image is often used as ground truth for (a); (c) ground truth estimated by [25]; (d) our ground truth. Noise estimates (β_1 and β_2 for noise level function and σ for Gaussian noise – see Section 3.2) indicate that our ground truth has significantly less noise than both (b) and (c). Images shown are processed in raw-RGB, while sRGB images are shown here to aid visualization.

[1] Source: Gartner Reports, 2017

Contribution This work establishes a much needed image dataset for smartphone denoising research. To this end, we propose a systematic procedure for estimating ground truth for real noisy images that can be used to benchmark denoising performance on smartphone imagery. Using this procedure, we captured a dataset of ~30,000 real noisy images using five representative smartphone cameras and generated their ground truth images. Using our dataset, we benchmarked a number of denoising methods to gauge the relative performance of various approaches, including patch-based methods and more recent CNN-based techniques. From this analysis, we show that for CNN-based methods, notable gains can be made when using our ground truth data versus conventional alternatives, such as low-ISO images.

2. Related Work

We review work related to ground truth image estimation for denoising evaluation. Given the wide scope of denoising research, only representative works are cited.

Ground truth for noisy real images The most widely used approach for minimizing random noise is by image averaging, where the average measurement of a scene point statistically converges to the noise-free value with a sufficiently large number of images. Image averaging has become a standard technique in a broad range of imaging applications that are significantly affected by noise, including fluorescence microscopy at low light levels and astronomical imaging of dim celestial bodies. The most basic form of this approach is to capture a set of images of a static scene with a stationary camera and fixed camera settings, then directly average the images. This strategy has been employed to generate noise-free images used in evaluating a denoising method [34], in evaluating a noise estimation method [5], in comparing algorithms for estimating noise level functions [21, 22], and for determining the parameters of a cross-channel noise model [24]. While per-pixel averaging is effective in certain instances, it is not valid in two common cases: (1) when there is misalignment in the sequence of images, which leads to a blurry mean image, and (2) when there are clipped pixel intensities due to low-light conditions or over-exposure, which causes the noise to be non-zero-mean and direct averaging to be biased [13, 25]. These two cases are typical of smartphone images, and to the best of our knowledge, no prior work has addressed ground truth estimation through image averaging under these settings. We show how to accurately estimate noise-free images for these cases as part of our ground truth estimation pipeline in Section 4.

Another common strategy is to assume that images from online datasets (e.g., the TID2013 [26] and PASCAL VOC datasets [12]) are noise-free and then synthetically generate noise to add to these images. However, there is little evidence to suggest the selected images are noise-free, and denoising results obtained in this manner are highly dependent on the accuracy of the noise model used.

Denoising benchmark with real images There have been, to the best of our knowledge, two attempts to quantitatively benchmark denoising algorithms on real images. One is the RENOIR dataset [2], which contains pairs of low/high-ISO images. This dataset lacks accurate spatial alignment, and the low-ISO images still contain noticeable noise. Also, the raw image intensities are linearly mapped to 8-bit depth, which adversely affects the quality of the images.

More closely related to our effort is the work on the Darmstadt Noise Dataset (DND) [25]. Like the RENOIR dataset, DND contains pairs of low/high-ISO images. By contrast, the work in [25] post-processes the low-ISO images to (1) spatially align them to their high-ISO counterparts, and (2) overcome intensity changes due to changes in ambient light or artificial light flicker. This work was the first principled attempt at producing high-quality ground truth images. However, most of the DND images have relatively low levels of noise and normal lighting conditions. As a result, there is a limited number of cases of high noise levels or low-light conditions, which are major concerns for image denoising and computer vision in general. Also, treating misalignment between images as a global translation is not sufficient for cases including lens motion, radial distortion, or optical image stabilization.

In our work on ground truth image estimation, we investigate issues that are pertinent for smartphone cameras and have not been properly addressed by prior strategies, such as the effect of spatial misalignment among images due to lens motion (i.e., optical stabilization) and radial distortion, and the effect of clipped intensities due to low-light conditions or over-exposure. In addition, we examine the impact of our dataset on recent deep learning-based methods and show that training with real noise and our ground truth leads to appreciably improved performance of such methods.

3. Dataset

In this section, we describe details regarding the setup and protocol followed to capture our dataset. Then, we discuss the image noise estimation.

3.1. Image Capture Setup and Protocol

Our image capture setup is as follows. We capture static indoor scenes to avoid misalignments caused by scene motion. In addition, we use a direct current (DC) light source to avoid the flickering effect of alternating current (AC) lights [28]. Our light source allows adjustments of illumination brightness and color temperature (ranging from 3200K to 5500K). We used five smartphone cameras (Apple iPhone 7, Google Pixel, Samsung Galaxy S6 Edge, Motorola Nexus 6, and LG G4).

We captured our dataset using the following protocol. We captured each scene multiple times using different cameras, different settings, and/or different lighting conditions. Each combination of these is called a *scene instance*. For each scene instance, we capture a *sequence* of successive images, with a 1–2-second time interval between subsequent images. While capturing an image sequence, all camera settings (e.g., ISO, exposure, focus, white balance, exposure compensation) are fixed throughout the process.

We captured 10 different scenes using five smartphone cameras under four different combinations (on average) of the following settings and conditions:

- 15 different ISO levels ranging from 50 up to 10,000 to obtain a variety of noise levels (the higher the ISO level, the higher the noise).
- Three illumination temperatures to simulate the effect of different light sources: 3200K for tungsten or halogen, 4400K for fluorescent lamps, and 5500K for daylight.
- Three light brightness levels: low, normal, and high.

For each scene instance, we captured a sequence of 150 successive images. Since noise is a random process, each image contains a random sample from the sensor's noise distribution. Therefore, the total number of images in our dataset – the Smartphone Image Denoising Dataset (SIDD) – is ~30,000 (10 scenes × 5 cameras × 4 conditions × 150 images). For each image, we generate the corresponding ground truth image (Section 4) and record all settings with the raw data in DNG/Tiff files. Figure 2 shows some example images from our dataset under different lighting conditions and camera settings.

Throughout this paper, we denote a sequence of images of the same scene instance as

$$\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}, \tag{1}$$

where $\mathbf{x}_i$ is the i^{th} image in the sequence, N is the number of images in the sequence, and $\mathbf{x}_i \in \mathbb{R}^M$, where M is the number of pixels in each image. Since we are considering images in raw-RGB space, we have only one mosaicked channel per image. However, images shown throughout the paper are rendered to sRGB to aid visualization.

3.2. Noise Estimation

It is often useful to have an estimate of the noise levels present in an image. To provide such estimates for our dataset, we use two common measures. The first is the signal-dependent noise level function (NLF) [21, 14, 20], which models the noise as a heteroscedastic signal-dependent Gaussian distribution where the variance of noise is proportional to image intensity. For low-intensity pixels, the heteroscedastic Gaussian model is still valid since the sensor noise (modeled as a Gaussian) dominates [18]. We denote the NLF-squared for a noise-free image $\mathbf{y}$ as

$$\boldsymbol{\beta}^2(\mathbf{y}) = \beta_1 \mathbf{y} + \beta_2, \tag{2}$$

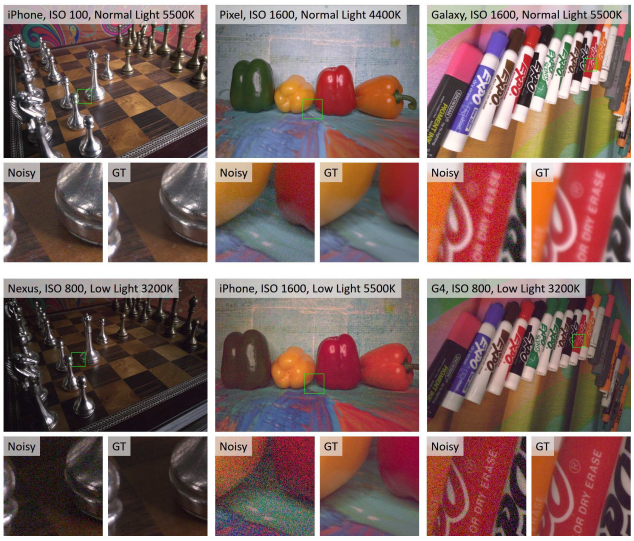


Figure 2: Examples of noisy images from our SIDD dataset captured under different lighting conditions and camera settings. Below each scene, zoomed-in regions from both the noisy image and our estimated ground truth (Section 4) are provided.

where β_1 is the signal-dependent (multiplicative component of the noise (the Poisson or shot noise), and β_2 is the independent additive Gaussian component of the noise. Then, the corresponding noisy image $\mathbf{x}$ clipped to $[0, 1]$ would be

$$\mathbf{x} = \min\Big(\max\big(\mathbf{y} + \mathcal{N}(\mathbf{0}, \boldsymbol{\beta}), \mathbf{0}\big), \mathbf{1}\Big). \tag{3}$$

For our noisy images, we report the NLF parameters provided by the camera device through the Android Camera2 API [15], which we found to be accurate when matched with [14]. To assess the quality of our ground truth images, we measure their NLF using [14]. The second measure of noise we use is the homoscedastic Gaussian distribution of noise that is independent of image intensity, usually denoted by its standard deviation σ. To measure σ for our images, we use the method in [7]. We include this latter measure of noise because many denoising algorithms require it as an input parameter along with the noisy image.

4. Ground Truth Estimation

This section provides details on the processing pipeline for estimating ground truth images along with experimental validation of the pipeline's efficacy. Figure 3 provides a diagram of the major steps:

1. Capture a sequence of images following our capture setup and protocol from Section 3;
2. Correct defective pixels in all images (Section 4.1);
3. Omit outlier images and apply intensity alignment of all images in the sequence (Section 4.2);
4. Apply dense local image alignment of all images with respect to a single reference image (Section 4.3);

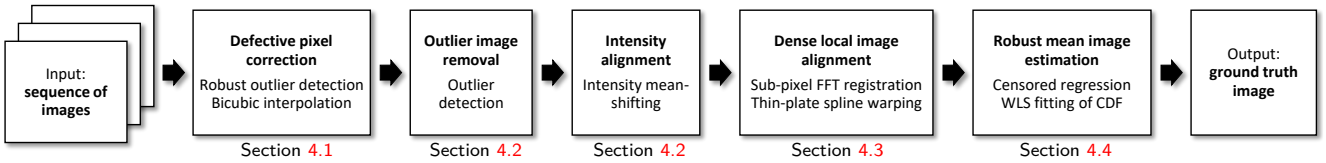


Figure 3: A block diagram illustrating the main steps in our procedure for ground truth image estimation. The respective sections for each step are shown.

5. Apply robust regression to estimate the underlying true pixel intensities of the ground-truth image (Section 4.4).

4.1. Defective Pixel Correction

Defective pixels can affect the accuracy of the ground-truth estimation as they do not adhere to the same underlying random process that generates the noise at normal pixel locations. We consider two kinds of defective pixels: (1) hot pixels that produce higher signal readings than expected; and (2) stuck pixels that produce fully saturated signal readings. We avoid altering image content by applying a median filter to remove such noise and instead apply the following procedure.

First, to detect the locations of defective pixels on each camera sensor, we capture a sequence of 500 images in a light-free environment. We record the mean image denoted as $\mathbf{x}_a$, and then we estimate a Gaussian distribution with mean μ_{dark} and standard deviation σ_{dark} over the distribution of pixels in the mean image $\mathbf{x}_a$. Ideally, μ_{dark} would be the *dark level* of the sensor and σ_{dark} would be the level of dark current noise. Hence, we consider all pixels having intensity values outside a 99.9% confidence interval of $\mathcal{N}(\mu_{dark}, \sigma_{dark})$ as defective pixels.

We use weighted least squares (WLS) fitting of the cumulative distribution function (CDF) to estimate the underlying Gaussian distribution of pixels. We use WLS to avoid the effect of outliers (i.e., the defective pixels), which can be up to 2% of the total pixels in the camera sensor. Also, the non-defective pixels normally have much smaller variance in their values compared to the defective pixels. This leads us to use a weighted approach to robustly estimate the underlying distribution.

After detecting the defective pixel locations, we use bicubic interpolation to estimate the correct intensity values at such locations. Figure 4 shows an example of a ground truth image where we apply our defective pixel correction method versus a directly estimated mean image. In the cameras we used, the percentage of defective pixels ranged from 0.05% up to 1.86% of the total number of pixels.

4.2. Intensity Alignment

Despite the controlled imaging environment, there is still a need to account for slight changes in scene illumination and camera exposure time due to potential hardware imprecision. To address this issue, we first estimate the average intensity of all images in the sequence where μ_i is the mean intensity of image i. Then, we calculate the mean μ_a and standard deviation σ_a of the distribution of all μ_i and consider all images outside a 99.9% confidence interval of $\mathcal{N}(\mu_a, \sigma_a)$ as outliers and remove them from the sequence. Finally, we re-calculate μ_a and perform intensity alignment by shifting all images to have the same mean intensity:

$$\mathbf{x}_i = \mathbf{x}_i - \mu_i + \mu_a. \quad (4)$$

The total number of outlier images we found in our entire dataset is only 231 images. These images were typically corrupted by noticeable brightness change.

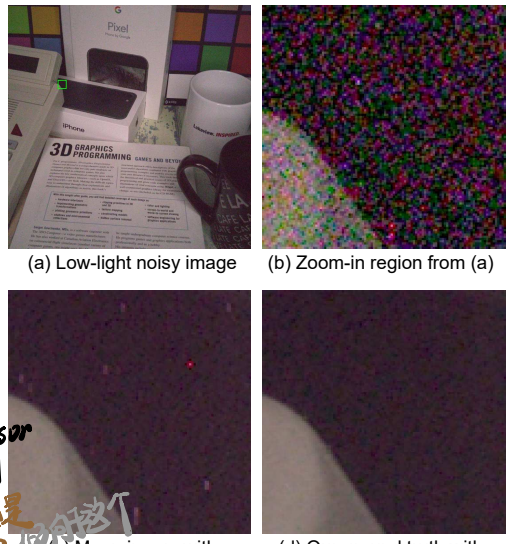
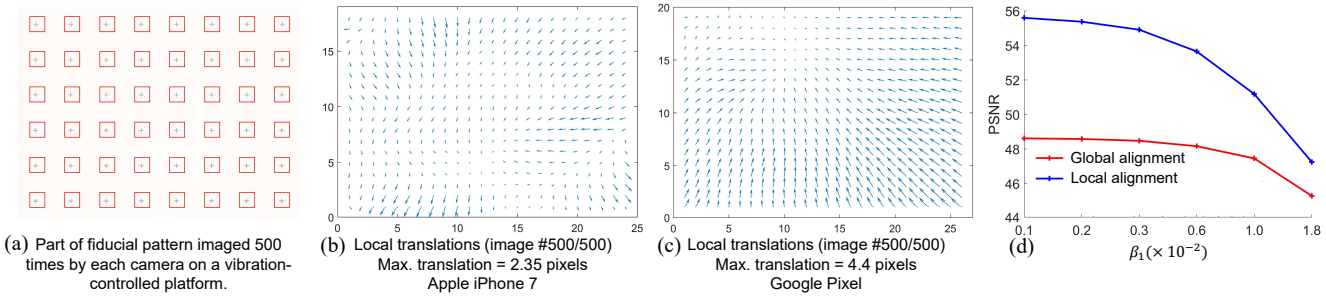


Figure 4: An example of a mean image (c) computed over a sequence of low-light images where defective pixels are present, and our corresponding ground truth (d) where defective pixels were corrected. One of the images from the sequence is shown in (a) and zoomed-in in (b).

4.3. Dense Local Spatial Alignment

While capturing image sequences with smartphones, we observed a noticeable shift in image content over the image sequence. To examine this problem further, we placed the smartphones on a vibration-controlled optical table (to rule out environmental vibration) and imaged a planar scene with fixed fiducials, as shown in Figure 5a. We tracked these fiducials over a sequence of 500 images to reveal a spatially varying pattern that looks like a combination of lens coax-



(a) Part of fiducial pattern imaged 500 times by each camera on a vibration-controlled platform.
(b) Local translations (image #500/500) Max. translation = 2.35 pixels Apple iPhone 7
(c) Local translations (image #500/500) Max. translation = 4.4 pixels Google Pixel
(d)

Figure 5: (a) A part of a static planar chart with fiducials imaged on a vibration-free optical table. The quiver plots of the observed and measured pixel drift between the first and last (500^{th}) image in a sequence of 500 images are shown for (b) iPhone 7 and (c) Google Pixel. (d) The effect of replacing our local alignment technique by a global 2D translation to align a sequence of images after synthesizing the local pixel drift from (b). We applied both techniques after synthesizing signal-dependent noise from a range of the β_1 parameter of the NLF estimated by the camera devices.

ial shift and radial distortion, as shown in Figure 5b for the iPhone 7 and Figure 5c for the Google Pixel. With a similar experiment, we found that DSLR cameras do not produce such distortions or shifts.

On further investigation, we found that this is caused by optical image stabilization (OIS) that cannot be disabled, either through API calls, or because it was part of the underlying camera hardware. [2] As a result, we had to perform local dense alignment of all images before estimation of the ground truth images. To do this, we adopted the following method for robust local alignment of the noisy images (we repeat this process for each image in the sequence):

1. Choose one image $\mathbf{x}_{\text{ref}}$ to be a reference for the alignment of all the other images in the sequence.
2. Divide each image into overlapping patches of size 512×512 pixels. We choose large enough patches to account for the higher noise levels in the images; the larger the patch, the more accurate our estimate of the local translation vector. We denote the centers of these patches as the destination landmarks which we use for the next registration step.
3. Use an accurate Fourier transform-based method [17] to estimate the local translation vector for each patch in each image $\mathbf{x}_i$ with respect to the corresponding patch from the reference image $\mathbf{x}_{\text{ref}}$. In this way, we obtain the source landmarks for each image.
4. Having the corresponding local translation vectors from the source landmarks in each image $\mathbf{x}_i$ to the destination landmarks in the reference image $\mathbf{x}_{\text{ref}}$, we apply 2D thin-plate spline image warping based on the set of arbitrary landmark correspondences [3] to align each image to the reference image. We found our adopted technique to be much more accurate than treating the misalignment problem as a global 2D translation.

Figure 5d shows the effect of replacing our local alignment technique by a global 2D translation. We applied both techniques on a sequence of synthetic images that includes synthesized local pixel shifts and signal-dependent noise. The synthesized local pixel shift is the same as the shift measured from real images (Figure 5b and 5c). The synthesized noise is based on the NLF parameters (β_1 and β_2) estimated by the camera devices and extracted using the Camera2 API. Our technique for local alignment consistently yields higher PSNR values over a range of realistic noise levels versus a 2D global alignment.

In our ground truth estimation pipeline, we warp all images in a sequence to a reference image for which we desire to estimate the ground truth. To estimate ground truth for another image in the sequence, we re-apply the spatial alignment process using that image as a reference. This way, we have a different ground truth for each image in our dataset.

4.4. Robust Mean Image Estimation

Once images are aligned, the next step is to estimate the mean image. The direct mean will be biased due to the clipping effects of the under-illuminated or over-exposed pixels [13]. To address this, we propose a robust technique that accounts for such clipping effects. Considering all observations of a pixel at position j throughout a sequence of N images, denoted as

$$\chi_j = \{x_{1j}, \ldots, x_{Nj}\}, \tag{5}$$

we need to robustly estimate the underlying noise-free value $\hat{\mu}_j$ of this pixel with the existence of censored observations due to the sensor's minimum and maximum measurement limits. As a result, instead of simple calculation of the mean value of χ_j, we apply the following method for robust estimation of $\hat{\mu}_j$:

1. Remove the possibly censored observations whose intensities are equal to 0 or 1 in normalized linear raw-RGB space:

$$\chi'_j \leftarrow \{x_{ij} \mid x_{ij} \in (0,1)\}_{i=1}^{N}, \tag{6}$$

where $|\chi_j|$ becomes $N' \leq N$.

[2] Google's Pixel camera does not support OIS; however, the underlying sensor, Sony's Exmor RS IMX378, includes gyro stabilization.

2. Define the empirical cumulative distribution function (CDF) of χ'_j as

$$\mathbf{\Phi}_e(t \mid \chi'_j) = \sum_{i=1}^{N'} \{x_{ij} \mid x_{ij} \le t\} / \sum_{i=1}^{N'} x_{ij}. \quad (7)$$

3. Define a parametric cumulative distribution function of a normal distribution with mean μ_p and standard deviation σ_p as

$$\mathbf{\Phi}_p(t \mid \mu_p, \sigma_p) = \int_{-\infty}^{t} \mathcal{N}(t' \mid \mu_p, \sigma_p)\, dt'. \quad (8)$$

4. Define an objective function that represents a weighted sum of square errors between $\mathbf{\Phi}_e$ and $\mathbf{\Phi}_p$ as

$$\psi(\mu_p, \sigma_p) = \sum_{t \in \chi'_j} w_t \left(\mathbf{\Phi}_p(t \mid \mu_p, \sigma_p) - \mathbf{\Phi}_e(t \mid \chi'_j)\right)^2, \quad (9)$$

where we choose the weights w_t to represent a convex function such that the weights compensate for the variances of the fitted CDF values, which are lowest near the mean and highest in the tails of the distribution:

$$w_t = \left(\mathbf{\Phi}_e(t \mid \chi'_j)\left(1 - \mathbf{\Phi}_e(t \mid \chi'_j)\right)\right)^{-\frac{1}{2}}. \quad (10)$$

5. Estimate the mean $\hat{\mu}_j$ and standard deviation $\hat{\sigma}_j$ of χ'_j by minimizing Equation 9:

$$(\hat{\mu}_j, \hat{\sigma}_j) = \arg\min_{\mu_p, \sigma_p} \psi(\mu_p, \sigma_p) \quad (11)$$

using a derivative-free simplex search method [20].

To evaluate our adopted WLS method for estimating mean images affected by intensity clipping, we conduct an experiment on synthetic images with synthetic noise added and intensity clipping applied. We used NLF parameters estimated from real images to synthesize the noise. We then apply our method to estimate the mean image. We compared the result with maximum likelihood estimation (MLE) with censoring, which is commonly used for censored data regression, as shown in Figure 6. We repeated the experiment over a range of numbers of images (Figure 6a) and a range of synthetic NLFs (Figure 6b). For reference, we plot the error of the direct calculation of the mean image before (green line) and after (black line) applying the intensity clipping. Our adopted WLS method achieves much lower error than MLE, almost as low as the direct calculation of the mean image before clipping.

Quality of our SIDD dataset vs the DND dataset In order to assess the quality of ground truth images estimated by our pipeline compared to the DND post-processing [25], we asked the authors of DND to post-process five of our low/high-ISO image pairs. We then estimated the inherent noise levels in these images using [7] and compared them to our ground truth of the same five scenes as shown in Figure 7a. Our pipeline yields lower noise levels, and hence, higher-quality images, in four out of five images. Also, Figure 7b shows the distribution of noise levels in our dataset compared to the DND dataset. The wider range of noise levels in our dataset makes it a more comprehensive benchmark for testing on different imaging conditions and more representative for smartphone camera images.

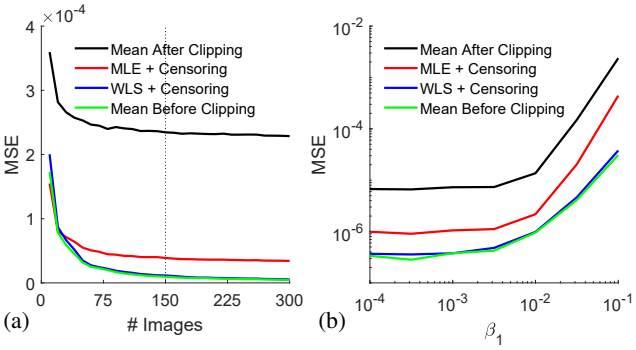


Figure 6: Comparison between methods used for estimating the mean image (a) over a range of number of images and (b) over a range of the first parameter of signal-dependent noise (β_1). The adopted method, WLS fitting of the CDF with censoring, yields the lowest MSE.

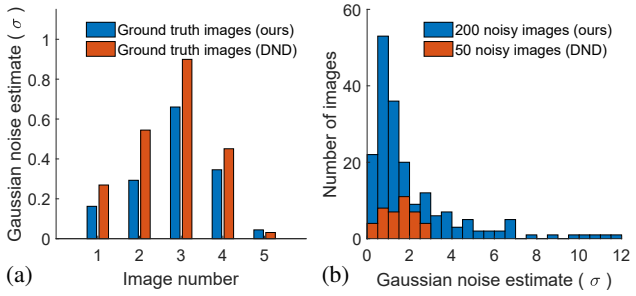


Figure 7: (a) Comparison between noise levels in our ground truth images versus the ground truth estimated by [25] for five scenes. Our ground truth has lower noise levels in four out of five images. (b) Comparison of noise levels in our dataset versus DND dataset.

5. Benchmark

In this section we benchmark a number of representative and state-of-the-art denoising algorithms to examine their performance on real noisy images with our recovered ground truth. We also show that the performance of CNN-based methods can be significantly improved if trained on real noisy images with our ground truth instead of synthetic noisy images and/or low-ISO images as ground truth.

5.1. Setup

For the purpose of benchmarking, we picked 200 ground truth images, one for each scene instance from our SIDD dataset. From these 200 images, we carefully selected a representative subset of 40 images for evaluation experiments in this paper and for a benchmarking website to be released

	Applied/ Evaluated	BM3D [10]	NLM [4]	KSVD [1]	KSVD-DCT [11]	KSVD-G [11]	LPG-PCA [32]	FoE [27]	MLP [6]	WNNM [16]	GLIDE [29]	TNRD [8]	EPLL [35]	DnCNN [33]
PSNR	Raw/Raw	**45.52**	44.06	43.26	42.70	42.50	42.79	43.13	43.17	44.85	41.87	42.77	40.73	43.30
	Raw/sRGB	**30.95**	29.39	27.41	28.21	28.13	30.01	27.18	27.52	29.54	25.98	26.99	25.19	28.24
	sRGB/sRGB	25.65	26.75	26.88	**27.51**	27.19	24.49	25.58	24.71	25.78	24.71	24.73	27.11	23.66
SSIM	Raw/Raw	**0.980**	0.971	0.969	0.970	0.969	0.974	0.969	0.965	0.975	0.949	0.945	0.935	0.965
	Raw/sRGB	0.863	0.846	0.832	0.784	0.781	0.854	0.812	0.788	**0.888**	0.816	0.744	0.842	0.829
	sRGB/sRGB	0.685	0.699	0.842	0.780	0.771	0.681	0.792	0.641	0.809	0.774	0.643	**0.870**	0.583
Time	Raw	34.3	210.7	2243.9	133.3	153.6	438.1	6097.2	131.2	1975.8	12440.5	**15.2**	653.1	51.7
	sRGB	**27.4**	621.9	9881.0	96.3	92.2	2004.3	12166.8	564.8	8882.2	36091.6	45.1	1996.4	158.9

Table 1: Denoising performance PSNR (dB), SSIM, and denoising time (seconds) per 1 Mpixel image (1024×1024 pixels) for benchmarked methods averaged over the 40 images form our SIDD benchmark. The top three methods are indicated with colors (green, blue, and red) in top-down order of performance, with best results in bold. For reference, the mean PSNRs of benchmark images in raw-RGB and sRGB are 36.70 dB and 19.71 dB, respectively, and the mean SSIM values are 0.832 and 0.397 in raw-RGB and sRGB, respectively. It is worth noting that the mean PSNRs of the noisy images in [25] were reported as 39.39 dB (raw-RGB) and 29.98 (sRGB), which indicate lower noise levels than in our dataset.

as well, while the other 160 noisy images and their ground truth images will be made available for training purposes. Since many denoisers are computationally expensive (some taking more than one hour to denoise a 1-Mpixel image), we expedite comparison by applying denoisers on 32 randomly selected non-overlapping image patches of size 256×256 pixels from each of the 40 images, for a total of 1280 image patches. The computation times of the benchmaked algorithms were obtained by running all of them single-threaded on the same machine equipped with an Intel® Xeon® CPU E5-2637 v4 @ 3.50GHz with 128GB of memory.

The algorithms benchmarked are: BM3D [10], NLM [4], KSVD [1], LPG-PCA [32], FoE [27], MLP [6], WNNM [16], GLIDE [29], TNRD [8], EPLL [35], and DnCNN [33]. For BM3D [10], we applied Anscombe-BM3D [23] in raw-RGB space and CBM3D [9] in sRGB space. For KSVD, we benchmark two variants of the original algorithm [11], one using the DCT over-complete dictionary, denoted here as KSVD-DCT, and the other using a global dictionary of natural image patches, denoted here as KSVD-G. For benchmarking the learning-based algorithms (e.g., MLP, TNRD, and DnCNN), we use the available trained models for the sake of fair comparison against other algorithms; however, in Section 5.3 we show the advantage of training DnCNN on our dataset. We applied all algorithms in both raw-RGB and sRGB spaces. However, the denoising in raw-RGB space is evaluated in both raw-RGB and after conversion to sRGB. In all cases, we evaluate performance against our ground truth images. For raw-RGB images, we denoise each CFA channel separately. To render images from raw-RGB to sRGB, we simulate the camera processing pipeline [19] using metadata from DNG files.

Most of the benchmarked algorithms require, as an input parameter, an estimate of the noise present in the image in the form of either the standard deviation of a uniform-power Gaussian distribution (σ) or the two parameters (β_1 and β_2) of the signal-dependent noise level function. We follow the same procedure from Section 3.2 to provide such estimates of the noise as input to the algorithms.

5.2. Results and Discussion

Table 1 shows the performance of the benchmarked algorithms in terms of peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [31], and denoising time. Our discussion, however, will be focused on the PSNR-based ranking of methods, as the top-performing methods tend to have similar SSIM scores, especially in raw-RGB space. From the PSNR results, we can see that classic patch-based and optimization-based methods (e.g., BM3D, KSVD, LPG-PCA, and WNNM) are outperforming learning-based methods (e.g., MLP, TNRD, and DnCNN) when tested on real images. This finding was also observed in [25]. We additionally benchmarked a number of methods not examined in [25], and make some interesting observations. One is that the two variants of the classic KSVD algorithm, trained on DCT and global dictionaries, achieve the best and second best PSNRs for the case of denoising in sRGB space. This is mainly because the underlying dictionaries well represent the distribution of small image patches in the sRGB space. Another observation is that denoising in the raw-RGB space yields higher quality with faster denoising compared to denoising in the sRGB space, as shown in Table 1. Also, we can see that BM3D is still one of the fastest denoising algorithms in the literature along with TNRD and dictionary-based KSVD, followed by other discriminative methods (e.g., DnCNN and MLP) and NLM. Furthermore, this examination of denoising times raises concerns about the applicability of some denoising methods. For example, though WNNM is one of the best denoisers, it is also among the slowest. Overall, we find the BM3D algorithm to remain one of the best performers in terms of denoising quality and computation time combined.

	# patches	# training	# testing	$[\sigma_{\min}, \sigma_{\max}]$	σ_μ
Subset A	5,120	4,096	1,024	[1.62, 5.26]	2.62
Subset B	10,240	8,192	2,048	[4.79, 23.5]	9.73

Table 2: Details of the two subsets of raw image patches used for training DnCNN. The terms $\sigma_{\min}$, $\sigma_{\max}$, and σ_μ indicate minimum, maximum, and mean noise levels.

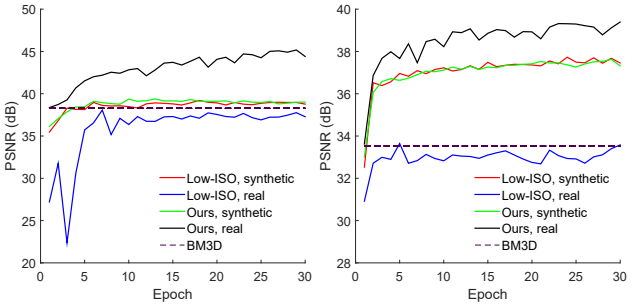
		Low-ISO		Ours	
		Synthetic	Real	Synthetic	Real
Subset A	β_1	4.66×10^{-3}	2.75×10^{-3}	2.88×10^{-3}	$\mathbf{1.01 \times 10^{-3}}$
	β_2	$\mathbf{1.90 \times 10^{-4}}$	3.94×10^{-4}	6.26×10^{-4}	8.05×10^{-4}
	σ	1.24	8.02×10^{-1}	8.95×10^{-1}	$\mathbf{4.62 \times 10^{-1}}$
Subset B	β_1	3.06×10^{-3}	2.20×10^{-3}	2.42×10^{-3}	$\mathbf{1.05 \times 10^{-3}}$
	β_2	9.67×10^{-4}	1.88×10^{-3}	$\mathbf{3.18 \times 10^{-4}}$	5.96×10^{-4}
	σ	1.03	1.04	6.97×10^{-1}	$\mathbf{4.18 \times 10^{-1}}$

Table 3: Mean noise estimates (β_1, β_2, and σ) of the denoised testing image patches using the four DnCNN models trained on subsets A and B. Training on our ground truth with real noise mostly yields higher-quality images.

5.3. Application to CNN Training

To further investigate the usefulness of our high-quality ground truth images, we use them to train the DnCNN denoising model [33] and compare the results with the same model trained on post-processed low-ISO images [25] as another type of ground truth. For each type of ground truth, we train DnCNN with two types of input: our real noisy images and our ground truth images with synthetic Gaussian noise added. For synthetic noise, we use the mean noise level (σ_μ), as estimated from the real noisy images, to synthesize the noise. We found that using noise levels higher than σ_μ for training yields lower testing performance. To further assess the four training cases, we test on two subsets of randomly selected raw-RGB image patches, one with low noise levels, and the other having medium to high noise levels, as shown in Table 2. Since we had access to only five low-ISO images post-processed by [25], we used them in subset A, whereas for subset B, we had to post-process additional low-ISO images using our own implementation of [25]. In all four cases of training, we test the performance against our ground truth images.

Figure 8 shows the testing results of DnCNN using two types of ground truth for training (post-processed low-ISO vs our ground truth images) and two types of noise (synthetic and real). Results are shown for both subsets A and B. We can see that training on our ground truth using real noise yields the highest PSNRs, whereas using low-ISO ground truth with real noise yields lower PSNRs. One reason for this is the remaining noise in the low-ISO images. Also, the post-processing may not sufficiently undo the intensity and spatial misalignment between low- and high-ISO images. Furthermore, the models trained on synthetic noise perform



(a) Subset A (b) Subset B

Figure 8: Testing results of DnCNN [33] using two types of ground truth (post-processed low-ISO and our ground truth images) and two types of noise (synthetic and real) on two random subsets of our dataset (see Table 2). Training with our ground truth on real noise yields the highest PSNRs.

similarly regardless of the underlying ground truth. This is because both models are trained on the same Gaussian distribution of noise and therefore learn to model the same distribution. Additionally, BM3D performs comparably on low noise levels (subset A), while DnCNN trained on our ground truth images significantly outperforms BM3D on all noise levels (both subsets). To investigate if there is a bias for using our ground truth as the reference of evaluation, we compare the no-reference noise estimates (β_1, β_2, and σ) of the denoised patches from the four models. As shown in Table 3, training on our ground truth with real noise mostly yields the highest quality, especially for β_1, which is the dominant component of the signal-dependent noise [30].

6. Conclusion

This paper has addressed a serious need for a high-quality image dataset for denoising research on smartphone cameras. Towards this goal, we have created a public dataset of ~30,000 images with corresponding high-quality ground truth images for five representative smartphones. We have provided a detailed description of how to capture and process smartphone images to produce this ground truth dataset. Using this dataset, we have benchmarked a number of existing methods to reveal that patch-based methods still outperform learning-based methods trained using conventional ground truthing methods. Our preliminary results on training CNN-based methods using our images (in particular, DnCNN [33]) suggest that CNN-based methods can outperform patch-based methods when trained on proper ground truth images. We believe our dataset and our associated findings will be useful in advancing denoising methods for images captured with smartphones.

Acknowledgments This study was funded in part by a Microsoft Research Award, the Canada First Research Excellence Fund for the Vision: Science to Applications (VISTA) programme, and The Natural Sciences and Engineering Research Council (NSERC) of Canada's Discovery Grant.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. 7

[2] J. Anaya and A. Barbu. RENOIR - a benchmark dataset for real noise reduction evaluation. *arXiv preprint*, 1409, 2014. 1, 2

[3] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE TPAMI*, 11(6):567–585, 1989. 5

[4] A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 7

[5] A. Buades, Y. Lou, J.-M. Morel, and Z. Tang. Multi image noise estimation and denoising. *MAP5 2010-19*, 2010. 2

[6] H. Burger, C. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *CVPR*, 2012. 7

[7] G. Chen, F. Zhu, and P. Ann Heng. An efficient statistical method for image noise level estimation. In *ICCV*, 2015. 3, 6

[8] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE TPAMI*, 39(6):1256–1272, 2017. 7

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In *IEEE ICIP*, volume 1, pages I–313, 2007. 7

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE TIP*, 16(8):2080–2095, 2007. 7

[11] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE TIP*, 15(12):3736–3745, 2006. 7

[12] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 2

[13] A. Foi. Clipped noisy images: Heteroskedastic modeling and practical denoising. *Signal Processing*, 89(12):2609–2629, 2009. 2, 5

[14] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE TIP*, 17(10):1737–1754, 2008. 3

[15] Google. Android Camera2 API. https://developer.android.com/reference/android/hardware/camera2/package-summary.html. Accessed: March 28, 2018. 3

[16] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014. 7

[17] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup. Efficient subpixel image registration algorithms. *Optics letters*, 33(2):156–158, 2008. 5

[18] S. W. Hasinoff. Photon, Poisson noise. In *Computer Vision*. 2014. 3

[19] H. Karaimer and M. S. Brown. A software platform for manipulating the camera imaging pipeline. In *ECCV*, 2016. 7

[20] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998. 6

[21] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE TPAMI*, 30(2):299–314, 2008. 2, 3

[22] X. Liu, M. Tanaka, and M. Okutomi. Practical signal-dependent noise parameter estimation from a single noisy image. *IEEE TIP*, 23(10):4361–4371, 2014. 2

[23] M. Makitalo and A. Foi. Optimal inversion of the Anscombe transformation in low-count Poisson image denoising. *IEEE TIP*, 20(1):99–109, 2011. 7

[24] S. Nam, Y. Hwang, Y. Matsushita, and S. Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *CVPR*, 2016. 2

[25] T. Plötz and S. Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 1, 2, 6, 7, 8

[26] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C.-C. J. Kuo. Image database TID2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, 30:57–77, 2015. 1, 2

[27] S. Roth and M. J. Black. Fields of experts. *IJCV*, 82(2):205–229, 2009. 7

[28] M. Sheinin, Y. Y. Schechner, and K. N. Kutulakos. Computational imaging on the electric grid. In *CVPR*, 2017. 2

[29] H. Talebi and P. Milanfar. Global image denoising. *IEEE TIP*, 23(2):755–768, 2014. 7

[30] H. J. Trussell and R. Zhang. The dominance of Poisson noise in color digital cameras. In *IEEE ICIP*, pages 329–332, 2012. 3, 8

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 7

[32] L. Zhang, W. Dong, D. Zhang, and G. Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43(4):1531–1549, 2010. 7

[33] K. Zhang et al. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE TIP*, 2017. 7, 8

[34] F. Zhu, G. Chen, and P.-A. Heng. From noise modeling to blind image denoising. In *CVPR*, 2016. 2

[35] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011. 7