Supplemental Information
Dirty Pixels: Towards End-to-End Image Processing and Perception

STEVEN DIAMOND*, Stanford University
VINCENT SITZMANN*, Stanford University, MIT
FRANK JULCA-AGUILAR*, Algolux
STEPHEN BOYD, Stanford University
GORDON WETZSTEIN, Stanford University
FELIX HEIDE, Princeton University

ACM Reference Format:
Steven Diamond, Vincent Sitzmann, Frank Julca-Aguilar, Stephen Boyd, Gordon Wetzstein, and Felix Heide. 2021. Supplemental Information Dirty Pixels: Towards End-to-End Image Processing and Perception. *ACM Trans. Graph.* 1, 1, Article 1 (January 2021), 10 pages. https://doi.org/10.1145/3446918

This document describes additional details about our image formation model, implementations, and results that complement the main manuscript.

1 CALIBRATION

In this section we discuss the details of the calibration of our image formation model.

1.1 PSF calibration

PSFs (point spread functions) offer a compact description of the aberrations of an optical system. For realistic optical systems, PSFs are spatially varying. In order to perform non-blind deconvolution, the PSF of the optical system has to be calibrated. PSF calibration is done with pictures taken under illumination conditions that ensure that no clipping occurs, allowing us to consider the unclipped image formation model (we illustrate the setup in Fig. 2). Furthermore, we have to take the Bayer pattern subsampling into account when estimating our PSF from raw images. For the purpose of PSF calibration we use the following image formation model:

$$y = S\left(v\left(g\left(i\right)\right) * k\right) + \eta. \tag{1}$$

where i is a target scene, η denotes additive noise, and y denotes the observation. Formation of y is a projection of i (i.e., the geometric distortion function of the lens and scene-camera projection) denoted by $g(.)$. It also depends on the optical vignetting function of the imaging system denoted by $v(.)$. In the imaging model (1), k represents

*These authors contributed equally to this research.

Authors' addresses: Steven Diamond, Stanford University, diamond@cs.stanford.edu; Vincent Sitzmann, Stanford University, MIT, sitzmann@mit.edu; Frank Julca-Aguilar, Algolux, frank.julca-aguilar@algolux.com; Stephen Boyd, Stanford University, boyd@stanford.edu; Gordon Wetzstein, Stanford University, gordonwz@stanford.edu; Felix Heide, Princeton University, fheide@cs.princeton.edu.

0730-0301/2021/1-ART1 $15.00
https://doi.org/10.1145/3446918

the lens PSF, and $*$ denotes the 2D convolution operator. $S(.)$ represents the sampling operator according to Bayer's pattern.

With some modifications on the charts introduced in [Mosleh et al. 2015], we use a chart that includes 0.5 expectation Bernoulli noise patterns and some checkerboard features for the camera-scene alignment shown in Fig. 1(a). A print of this chart is used as our synthetic scene i. Its picture is used as the observation y (Fig. 1(b)). Checkerboard corners in the picture of i are used to determine the projection function $g(.)$. We estimate local intensity, which is varying due to lens shading, and use that as a weighting factor in the estimation. Hence, we can form a sharp version of the scene as $u = v\,(g(i))$. Given n observations of the scene $x_1 \dots x_n$ and their sharp correspondences $u_1 \dots u_n$, we estimate a PSF k for each channel by finding the solution to

$$\hat{\mathbf{k}} = \arg\max_{\mathbf{k}} \left\| \begin{bmatrix} \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_n \end{bmatrix} \mathbf{k} - \begin{bmatrix} \mathbf{S}^\top \mathbf{y}_1 \\ \vdots \\ \mathbf{S}^\top \mathbf{y}_n \end{bmatrix} \right\|_2^2, \text{ s.t. } \mathbf{k} \geqslant 0 \tag{2}$$

where $\mathbf{y}_j \in \mathbb{R}^{NM/4 \times 1}$ and $\mathbf{k} \in \mathbb{R}^{R^2 \times 1}$ denote the $N/2 \times M/2$ observation x_j and the $R \times R$ channel PSF k in vector form, respectively. The $N \times M$ sharp correspondence u_j of each observation in a convolution matrix form is denoted by $\mathbf{U}_j \in \mathbb{R}^{NM \times R^2}$. In order to account for the super-resolved version of the PSF, each observation is transformed into the sensor resolution space using $\mathbf{S}^\top$ ($\mathbf{S} \in \mathbb{R}^{NM/4 \times NM}$ denotes the sampling matrix form of $S(.)$, and $\top$ denotes the matrix transpose). In our experiments, we use 10 observations (and 10 sharp correspondences), i.e., $n = 10$.

The lens PSF varies spatially in camera space. Therefore, the field-of-view of the camera is divided into non-overlapping blocks and the PSF estimation is carried out for each block individually. Fig. 1(c) shows a set of estimated PSFs for the entire field-of-view of a Nexus 5 camera.

1.2 Noise calibration

We consider the Poisson-Gauss noise model which is simple yet accurate for our use cases. Fig. 4 illustrates the calibration setup. We follow the modeling framework of [Foi 2009] in which the image y is given by

$$y = x + s(x)\xi, \quad x = \mathrm{E}(y),\ s(x) = \mathrm{std}(y) \tag{3}$$

ξ is an independent random noise such that,

$$s(x)\xi = \eta_\mathrm{p}(x) + \eta_\mathrm{g} \tag{4}$$

with $\eta_\mathrm{p}(x)$ and η_g the Poissonian and Gaussian components of the noise which follow the distributions

$$\eta_\mathrm{p}(x) \sim \alpha \mathcal{P}(\alpha^{-1} x)$$
$$\eta_\mathrm{g} \sim \mathcal{N}(0, \sigma^2)$$

Following [Foi 2009], the amplification by gain θ is modeled as a scaling $x = \theta \mathring{x}$, where $\mathring{x}$ is the corresponding clean signal without amplification.

Parameters α and σ depend on the hardware and the gain θ,

$$\alpha = \mathring{\alpha}\theta, \quad \sigma^2 = \theta \mathring{\sigma}^2 + \mathring{\sigma}'^2, \tag{5}$$

$\mathring{\sigma}^2 + \mathring{\sigma}'^2$ and $\mathring{\alpha}$ are the variance of the Gaussian noise and the value of α in the case of no amplification. $\mathring{\sigma}^2$ is the variance of the Gaussian noise introduced up to the amplification circuitry and $\mathring{\sigma}'^2$ is the variance of the Gaussian noise introduced thereafter. To estimate the noise parameters, we take calibration pictures of a noise chart (*e.g.* [ISO 12233:2014 2014]) at various gains. The estimation procedure from [Foi 2009] yields α and σ for

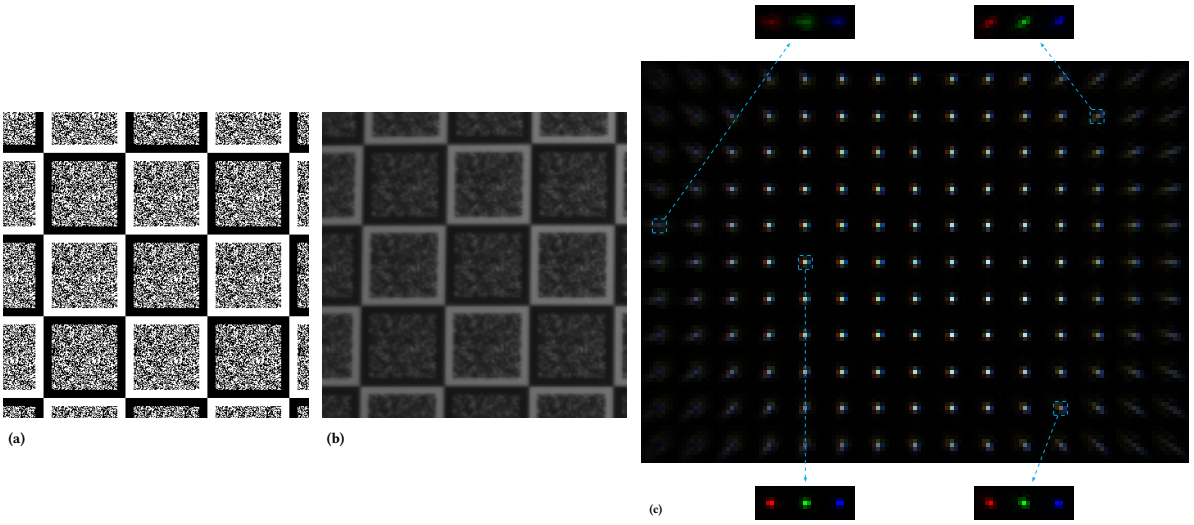

Fig. 1. (a) Synthetic pattern (i) used in the PSF estimation. (b) An observation b of the synthetic pattern. (Images in (a) and (b) are scaled to have a similar resolution for a better display here. They have different resolutions in practice.) (c) Lens PSFs measured for a Nexus 5 camera.

each gain, from which we deduce estimates for $\mathring{\alpha}$, $\mathring{\sigma}$ and $\mathring{\sigma}'$. In our use case we verified that we can assume within a good approximation that the Gaussian noise is created only before and up to the amplification, such that

$$\alpha = \mathring{\alpha}\theta, \quad \sigma = \theta\mathring{\sigma} \tag{6}$$

In the case of a clipped observation the model can be written (see [Foi 2009])

$$\hat{y} = \hat{x} + \hat{s}(\hat{x})\hat{\xi}, \quad \mathrm{E}(\hat{y}) = \hat{x},\ \hat{s}(\hat{x}) = \mathrm{std}(\hat{y}) \tag{7}$$

Fig. 3 shows plots of $s(x)$ and $\hat{s}(\hat{x})$ for the Nexus 6P at various ISO levels. The unit of both x-axis and y-axis of these plots is *digital unit* scaled by the maximal possible value of the sensor. Digital unit is the unit of the raw signal y, see [EMVA 1288 2016] for more details.

2 IMPLEMENTATION DETAILS

We built all models in the TensorFlow framework [Abadi et al. 2015]. We first pretrained the MobileNet-v1 on ImageNet for classification. We then trained our end-to-end architecture with the higher-level loss, using the pretrained MobileNet as initialization. We used RMSProp [Tieleman and Hinton 2012] with a decay of 0.9, $\epsilon = 1.0$, and a learning rate of $4.5e^{-4}$, exponentially decayed by a factor of 0.94 every epoch. We used data augmentation via image reflection, rotation, and random croping. Both our models and the baseline models are trained til convergence on 4 NVIDIA Tesla K80 GPUs per model. We will publish the trained models upon acceptance.

For the proposed image reconstruction network for human viewing, we trained and evaluated our proposed image reconstruction network using the same training, validation, and test splits defined by [Chen et al. 2018].

3 LEARNED DEEP ISP INTERMEDIATES

As described in Section 5 of the main manuscript, we also compared the proposed approach against image preprocessing using the U-Net network proposed by Chen et al. [2018] for imaging in low light scenarios. Figure 5

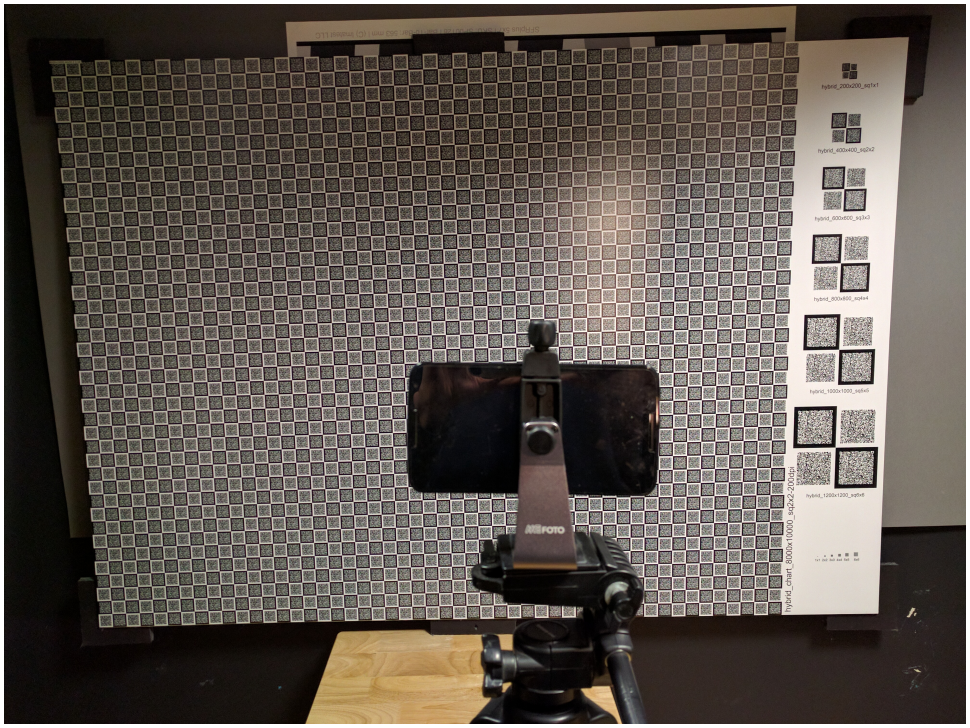

Fig. 2. PSF calibration setup.

shows some examples of raw images processed using the fine-tuned network from [Chen et al. 2018] that we use for our evaluations in Section 5 of the main manuscript. This network was trained with an ℓ_1 loss as proposed in [Chen et al. 2018]. We note that while this learned deep ISP network is able to reduce noise and recover perceptually pleasing color images, this process also remove fine details of the scene objects.

4 LEARNED DEEP ISP WITH ADDITIONAL PERCEPTUAL LOSS

The evaluation in the main paper also includes a separate variant of Chen et al. [2018] trained with an additional perceptual loss. For these experiments we initialize the training with the pre-trained network from above, which was trained with an ℓ_1 loss as proposed in [Chen et al. 2018]. We then add a perceptual loss with manually tuned weight to this loss in a second fine-tuning stage. Following [Johnson et al. 2016] we place an ℓ_2 loss over features extracted from the output image. Specifically, we use a feature reconstruction loss at *Relu2_2* of a VGG-16 model trained on ImageNet. Once this fine-tuning with perceptual loss had converged, we finetuned a MobileNet-v1 classifier over the images processed with this perceptual denoiser (now discarding all intermediate image losses). The results of this experiment are listed in Table 1 of the main manuscript.

5 CAPTURED DATASET RESULTS

In this section, we discuss the motivation behind the experiments in Section 5.2 in greater detail, and include additional results that complement our claims.

When operating in the wild, classification architectures are confronted with images from a wide range of light levels. However, as Table 1 demonstrates, classification pipelines trained for one noise level or on clean images

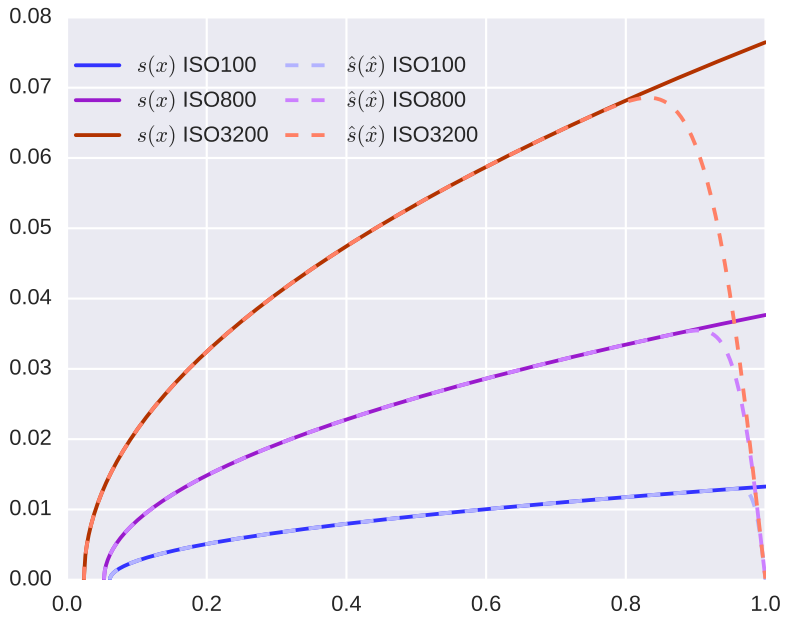

Fig. 3. std($\tilde{y}$) against E($\tilde{y}$) and std(y) against E(y) for the Nexus 6P at several ISO levels.

cannot generally be assumed to generalize well to other capturing conditions. There are several approaches to addressing this challenge. One can train separate models for a variety of different light levels and then choose the correct model at test time; however, this requires an explicit light level estimation and is inefficient, as several models have to be trained. A preferable solution is to develop architectures that are robust to varying light levels.

We thus investigate the capability of our baseline and proposed models to generalize to capturing conditions different from their training conditions. In addition to the results presented in Section 5, we evaluate a fine-tuned

Top-1 Accuracy

	No Blur	Center PSF	Offaxis PSF	Periphery PSF
3 lux	45.54%	47.37%	43.68%	19.65%
6 lux	63.57%	64.01%	61.07%	37.69%
12 lux	71.27%	71.49%	69.38%	53.29%
24 lux	75.03%	74.88%	73.10%	61.58%
48 lux	76.97%	76.49%	74.77%	64.92%
96 lux	78.10%	77.24%	75.52%	66.38%
No Noise	80.20%	78.02%	76.46%	67.57%

Top-5 Accuracy

	No Blur	Center PSF	Offaxis PSF	Periphery PSF
3 lux	68.53%	69.94%	65.81%	35.50%
6 lux	84.51%	84.57%	82.03%	59.06%
12 lux	90.18%	90.05%	88.45%	75.13%
24 lux	92.37%	92.12%	90.86%	82.18%
48 lux	93.41%	93.06%	91.96%	84.88%
96 lux	94.13%	93.48%	92.48%	85.84%
No Noise	95.20%	94.01%	93.00%	86.78%

Table 1. Top-1 and Top-5 accuracies of the pretrained Inception-v4 network evaluated on simulated datasets for different light levels and point spread function blurs.

Fig. 4. Noise calibration setup.

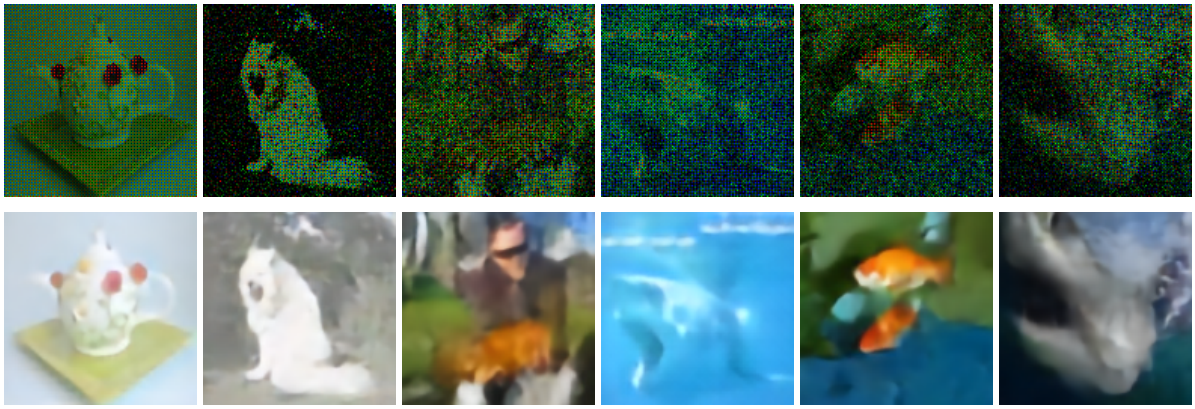

Fig. 5. Output examples of raw noisy images (top), and images processed with U-Net from Chen et al. [2018] fine-tuned on the noisy data. Left three rows correspond to images from 2-to-200 lux range, right three columns correspond to our 6 lux data.

Inception-v4 and the proposed joint model trained, also with a Inception-v4 classifier, on 3 lux simulated data on our captured 6 lux dataset and vice versa.

	Avg. 3 Lux		Avg. 6 Lux	
	Top-1	Top-5	Top-1	Top-5
Pretrained Inception-v4	20.14%	33.06%	30.07%	53.17%
Fine-tuned Inception-v4	37.32%	59.36%	46.01%	63.77%
Joint Architecture	**44.43%**	**65.64%**	**48.28%**	**71.65%**

Table 2. Results on data captured in the wild with a Google Pixel phone rear camera for models trained on the same light level.

	Avg. 3 Lux		Avg. 6 Lux	
	Top-1	Top-5	Top-1	Top-5
Pretrained Inception-v4	20.14%	33.06%	30.07%	53.17%
Fine-tuned Inception-v4	21.56%	35.55%	30.53%	50.91%
Joint Architecture	**31.64%**	**54.50%**	**45.29%**	**68.03%**

Table 3. Robustness to noise level on raw data. Results for models trained on *different* light level (*i.e.*, the 3 lux models evaluated on the 6 lux data and vice versa). The same data set as in Table 2 was used. The results validate that the proposed architecture generalizes well to unseen noise levels (close enough to the one of the synthetic training data), while fine-tuned networks are specialized to the exact noise distribution from the training.

Tables 2 and 3 show the performance of the trained models on the datasets corresponding to the training light level and the respective other, "wrong" light level. While the fine-tuned Inception-v4 models outperform the pretrained baseline on the light level they were fine-tuned on, they only perform on par for the respective "wrong" light level. This suggests that fine-tuning the inception architecture leads to models that are highly specialized to the training light level.

The proposed joint architectures outperform the fine-tuned Inception-v4 baselines on the light level they were trained for by up to 7.1%. However, they also outperform the pretrained Inception-v4 baseline on the other, unseen light level by tens of percent, even performing on par with the fine-tuned Inception-v4 baseline on the 6 lux dataset. This suggests that the low-level pipeline enables generalization across light levels, making the proposed architecture much more robust to classification in the wild than the alternative approaches.

Note that these results are different from the results over 2-to-20 and 2-to-200 lux low-light ranges showed in Table 1 of the main manuscript. In the light-level range results, the trained and test images include different levels of noise but both are randomly sampled from the corresponding range. On the other hand, in the results showed in Table 3, the specific noise levels of the training and test data are different. In both cases, our proposed architecture is able to generalize better across light levels, compared to conventional methods and models trained from scratch or fine-tuned.

6 COMPARISON TO TSENG ET AL. [2019]

We evaluated the first-order ISP approximation method from [Tseng et al. 2019] trained on our 2-to-200 low-light range data. In this experiment, as suggested by the authors of [Tseng et al. 2019], we used their ARM Mali-C71 hardware ISP (processing kindly provided by the authors) to generate 10000 ISP outputs by uniformly sampling the ISP hyperparameter space. The ARM Mali-C71 ISP hyperparameter space consists of 31 unique hyperparameters for denoising, demosaicking, white balancing, edge enhancement, gamma and tone correction blocks of the ISP. We generated the RAW inputs for this ISP as described in the main manuscript. The resulting RAW/post-ISP

image pairs with corresponding parameter settings were used to approximate the ISP using Tseng et al. [2019]. However, due to the high variation of the ISP in low light, the proxy network *failed to approximate the ISP on the highly noisy scenes as evidenced in Figure 6.* When using the flawed proxy network nevertheless for ISP tuning with a MobileNet-v1 classifier as in Section 5 of the main paper, the authors of [Tseng et al. 2019] confirmed that optimized parameters were out of bounds for the hardware ISP. This experiment validates the difficulty of training an ISP approximation network to cover real world scenes over variable low-light levels. As such, ISP parameter tuning using Tseng et al. [2019] is not a viable alternative to the proposed method, even when not taking into account that Tseng et al. [2019] do not learn the network parameters of the higher-level network, in contrast to the proposed method.

Similarly, Tseng et al. [2019] also failed on the Movidius Myriad 2 ISP images provided by the authors of this work for the same low-light scenario. We note that the Mali-C71 and Myriad 2 hardware ISP pipelines are not designed for such extreme low-light captures and, as such, it is not surprising that Tseng et al. [2019] struggles to fit their ISP approximator to the highly variable ISP output for such low-light captures. We also confirmed this with the authors of Tseng et al. [2019].

7 ADDITIONAL QUALITATIVE COMPARISONS

In Fig. 7, we show qualitative comparisons of classification results on low-light images in the wild. We show both the classification results of the pretrained Inception-v4 network, traditional denoiser (BM3D) + pretrained Inception-v4, and our proposed joint architecture trained on data simulated for a light level of 3 lux. The images were captured using the mobile prototype described in Section 6 of the paper, using a Google Pixel phone. The images were captured with a wide variety of light levels. The proposed joint architecture succeeds in predicting the correct class even under these inconsistent conditions, demonstrating generalization capabilities to light levels different from the training light level. The pretrained Inception-v4 network largely fails on these difficult cases.

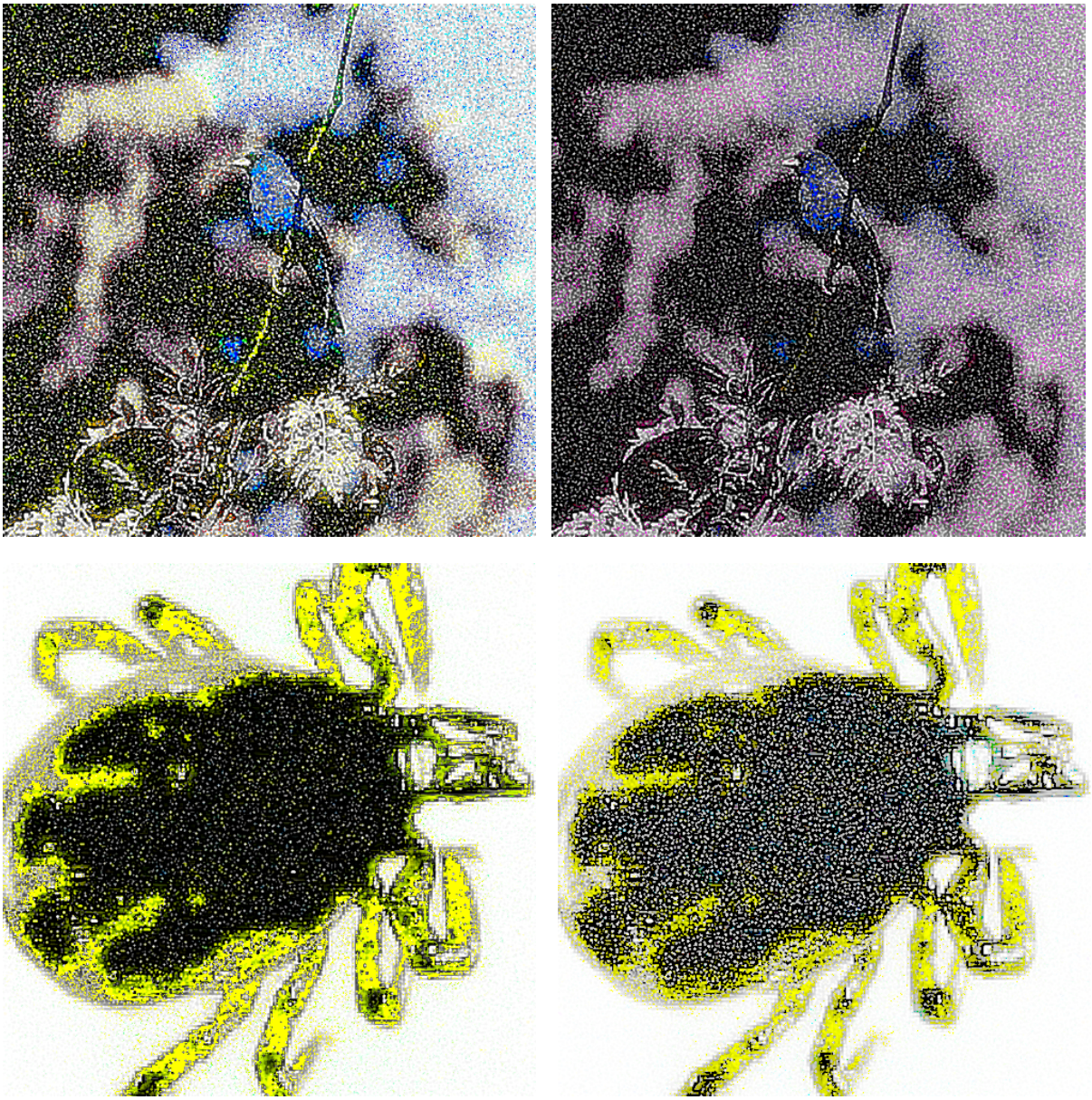

Fig. 6. ARM Mali-C71 hardware (left) and ISP approximation output (right) images using Tseng et al. [2019]. The ISP proxy from [Tseng et al. 2019] fails to fit to the hardware ISP in the low-light scenarios considered in our work and, hence, ISP hyperparameter optimization [2019] does not offer an alternative to the proposed method. Using the flawed ISP proxy for hyperparameter training with a MobileNet-v1 classifier resulted in out-of-bounds parameters using the method from Tseng et al. [2019].

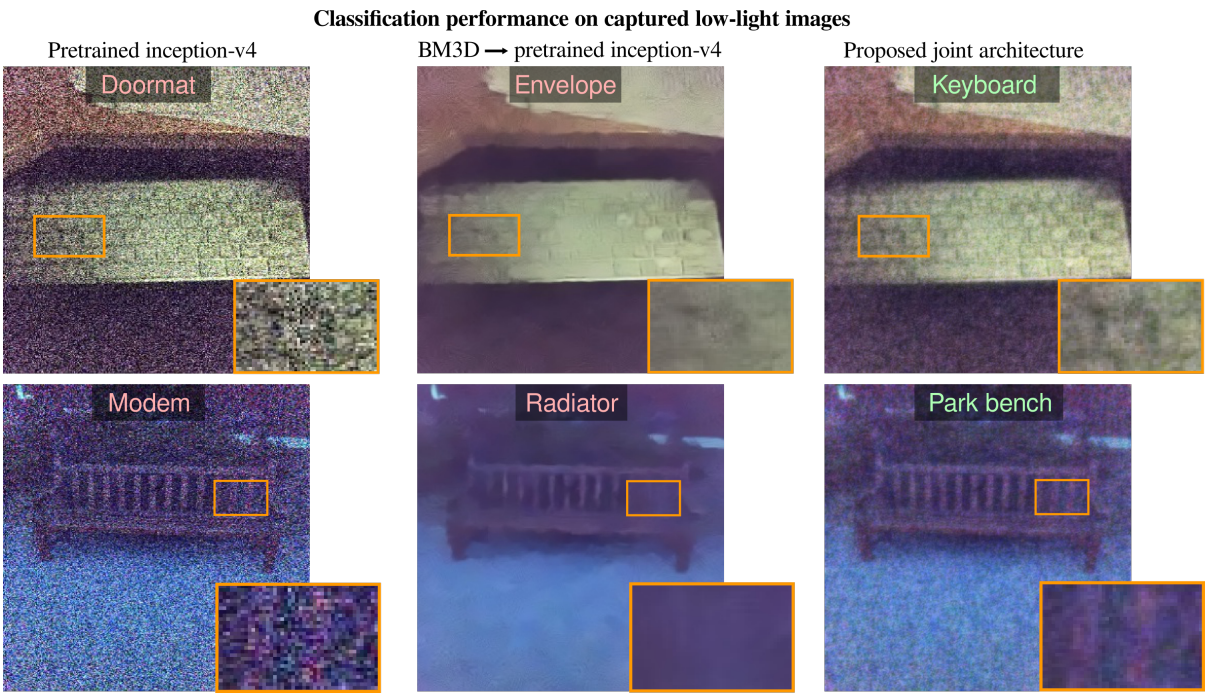

Fig. 7. Here we show images taken with a Google Pixel rear-facing camera in low light conditions (gamma-corrected for display). (Left) The noisy images are classified incorrectly by the pretrained Inception-v4 network. (Center) Denoising the images with BM3D leads to poor results due to the extreme noise level. The pretrained Inception-v4 network still cannot classify them correctly. (Right) The proposed joint architecture preserves fine detail at the expense of more noise and artifacts than for BM3D, leading to a correct classification result.

REFERENCES

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.

C. Chen, Q. Chen, J. Xu, and V. Koltun. 2018. Learning to See in the Dark. *ArXiv e-prints* (May 2018). arXiv:1805.01934

EMVA 1288 2016. EMVA 1288 Standard for Characterization of Image Sensors and Cameras.

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

ISO 12233:2014 2014. ISO 12233:2014 Photography – Electronic still picture imaging – Resolution and spatial frequency responses.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*. Springer, 694–711.

A. Mosleh, P. Green, E. Onzon, I. Begin, and J.M. Pierre Langlois. 2015. Camera Intrinsic Blur Kernel Estimation: A Reliable Framework. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

T. Tieleman and G. Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning* 4, 2 (2012).

Ethan Tseng, Felix Yu, Yuting Yang, Fahim Mannan, Karl ST Arnaud, Derek Nowrouzezahrai, Jean-François Lalonde, and Felix Heide. 2019. Hyperparameter optimization in black-box image processing using differentiable proxies. *ACM Transactions on Graphics (SIGGRAPH)* 38, 4 (2019), 27.