Dirty Pixels:
Optimizing Image Classification Architectures for Raw Sensor Data

Steven Diamond   Vincent Sitzmann   Stephen Boyd   Gordon Wetzstein   Felix Heide

Noisy image → Inception-v4   Noisy image → BM3D → Inception-v4   Noisy image → Proposed joint architecture

Figure 1: *The performance of deep networks trained for high-level computer vision tasks such as classification degrades under noise, blur, and other imperfections present in raw sensor data. (Left) An image of jelly beans corrupted by noise characteristic of low-light conditions is misclassified as a library by the Inception-v4 classification network. Cleaning up raw data using conventional low-level image processing does not necessarily improve performance. (Center) The image denoised with BM3D is still misclassified, now as a vending machine. We propose an end-to-end differentiable architecture for joint denoising, deblurring, and classification that makes classification robust to realistic noise and blur. The proposed architecture learns a denoising pipeline optimized for classification that enhances fine detail at the expense of more noise and artifacts. (Right) The learned denoiser recovers the jellybean texture, and the image is correctly classified in the "confection" category. The proposed architecture has a principled and modular design inspired by formal optimization methods that generalizes to other combinations of image formation models and high-level computer vision tasks.*

Abstract

Real-world sensors suffer from noise, blur, and other imperfections that make high-level computer vision tasks like scene segmentation, tracking, and scene understanding difficult. Making high-level computer vision networks robust is imperative for real-world applications like autonomous driving, robotics, and surveillance. We propose a novel end-to-end differentiable architecture for joint denoising, deblurring, and classification that makes classification robust to realistic noise and blur. The proposed architecture dramatically improves the accuracy of a classification network in low light and other challenging conditions, outperforming alternative approaches such as retraining the network on noisy and blurry images and preprocessing raw sensor inputs with conventional denoising and deblurring algorithms. The architecture learns denoising and deblurring pipelines optimized for classification whose outputs differ markedly from those of state-of-the-art denoising and deblurring methods, preserving fine detail at the cost of more noise and artifacts. Our results suggest that the best low-level image processing for computer vision is different from existing algorithms designed to produce visually pleasing images. The principles used to design the proposed architecture easily extend to other high-level computer vision tasks and image formation models, providing a general framework for integrating low-level and high-level image processing.

Keywords: computer vision, computational photography, machine learning, digital image processing

1   Introduction

Recent progress in deep learning has made it possible for computers to perform high-level tasks on images, such as classification, segmentation, and scene understanding. High-level computer vision is useful for many real-world applications, including autonomous driving, robotics, and surveillance. Applying deep networks trained for high-level computer vision tasks to the outputs of real-world imaging systems can be difficult, however, because raw sensor data is often corrupted by noise, blur, and other imperfections.

What is the correct way to apply high-level networks to raw sensor data? Do effects such as noise and blur degrade network performance? If so, can the lost performance be regained by cleaning up the raw data with traditional image processing algorithms or by retraining the high-level network on raw data? Or is an entirely new approach to combining low-level and high-level image processing necessary to make deep networks robust?

We examine these questions in the context of image classification under realistic camera noise and blur. We show that realistic noise and blur can substantially reduce the performance of a classification architecture, even after retraining on noisy and blurry images or preprocessing the images with standard denoising and deblurring algorithms. We introduce a new architecture for combined denoising, deblurring, and classification that improves classification performance in difficult scenarios. The proposed architecture is end-to-end differentiable and based on a principled and modular approach to combining low-level image processing with deep architectures. The architecture could be modified to handle a different image formation model or high-level computer vision task. We obtain superior performance by training the low-level image processing pipeline together with the classification network. The images output by the low-level image processing pipeline optimized for classification are qualitatively different from the images output by conventional denoising and deblurring algorithms, scoring worse on traditional reconstruction metrics such as peak signal-to-noise ratio (PSNR).

The proposed architecture for joint denoising, deblurring, and classification makes classification robust and effective in real-world ap-

plications. The principles used to design the proposed architecture can be applied to make other high-level computer vision tasks robust to noise and blur, as well as to handle raw sensor data with more complex image formation models, such as RGB-D cameras and general sensor fusion. More broadly, the idea of combining low-level and high-level image processing within a jointly trained architecture opens up new possibilities for all of computational imaging.

Our contributions in this paper are the following:

- We introduce a dataset of realistic noise and blur models calibrated from real-world cameras.

- We evaluate a classification architecture on images with realistic noise and blur and show substantial loss in performance.

- We propose a new end-to-end differentiable architecture that combines denoising and deblurring with classification, based on a principled and modular design inspired by formal optimization that can be applied to other image formation models and high-level tasks.

- We demonstrate that the proposed architecture, tuned on noisy and blurry images, substantially improves on the classification accuracy of the original network. The joint architecture outperforms alternative approaches such as fine-tuning the classification architecture alone and preprocessing images with a conventional denoiser or deblurrer.

- We highlight substantial qualitative differences between the denoised and deblurred images output by the proposed architecture and those output by conventional denoisers and deblurrers, which suggest that the low-level image processing that is best for high-level computer vision tasks like classification is different than that which is best for producing visually pleasing images.

We evaluate the performance of the proposed architecture primarily in low-light conditions. We focus on classification in low-light both because it is important for real-world applications, such as autonomous driving and surveillance at night, and because out of the broad range of light levels for which we evaluated the classification network we found the largest drop in accuracy in low light (both with and without blur). If we can mitigate the effects of noise and blur under the most challenging conditions, then we can certainly do so for easier scenarios.

2 Related Work

Effects of noise and blur on high-level networks A small body of work has explored the effects of noise and blur on deep networks trained for high-level computer vision tasks. Dodge and Karam evaluated a variety of state-of-the-art classification networks under noise and blur and found a substantial drop in performance [2016]. Vasiljevic et al. similarly showed that blur decreased classification and segmentation performance for deep networks, though much of the lost performance was regained by fine-tuning on blurry images [2016]. Several authors demonstrated that preprocessing noisy images with trained or classical denoisers improves the performance of trained classifiers [Tang and Eliasmith 2010; Tang et al. 2012; Agostinelli et al. 2013; Jalalvand et al. 2016; da Costa et al. 2016]. Chen et al. showed that training a single model for both denoising and classification can improve performance on both tasks [2016]. To the best of our knowledge we are the first to jointly train a denoiser or deblurrer combined with a high-level computer vision network in a pipeline architecture.

Unrolled optimization algorithms The low-level image processing in the proposed joint architecture is based on unrolled optimization algorithms. Unrolled algorithms take classical iterative optimization methods, such as forward-backward splitting [Bruck 1975], ISTA and FISTA [Beck and Teboulle 2009], Cremers-Chambolle-Pock [Pock et al. 2009; Chambolle and Pock 2011], the alternating direction method of multipliers [Glowinski and Marroco 1975; Boyd et al. 2001], and half-quadratic splitting [Geman and Yang 1995], and fix the number of iterations. If each iteration is differentiable in its output with respect to its parameters, the parameters of the unrolled algorithm can be optimized for a given loss through gradient based methods. Ochs et al. developed an unrolled primal-dual algorithm with Bregman distances and showed an application to segmentation [2015; 2016]. Schmidt and Roth trained image denoising and deblurring models using unrolled half-quadratic splitting [2014]. Similarly, Chen et al. trained models for denoising and other tasks using an unrolled forward-backward algorithm [2015; 2015]. Both Schmidt and Roth and Chen et al. parameterized their models using a field-of-experts prior [Roth and Black 2005].

Structured neural networks Unrolled optimization algorithms can be interpreted as structured neural networks, in which the network architecture encodes domain knowledge for a particular task [Wang et al. 2016]. Structured neural networks have been proposed for deblurring [Xu et al. 2014; Schuler et al. 2014; Chakrabarti 2016; Zhang et al. 2016a], denoising [Zhang et al. 2016b], and demosaicking [Gharbi et al. 2016]. Conventional fully-connected or convolutional neural networks have also been successfully applied to low-level image processing tasks (see, *e.g.*, [Jain and Seung 2009; Xie et al. 2012; Burger et al. 2012; Dong et al. 2014; Kim et al. 2016]). Another approach to linking traditional optimization methods and neural networks is to train a network for image reconstruction on data preprocessed with an iterative reconstruction algorithm [Schuler et al. 2013; Jin et al. 2016].

Camera image processing pipelines Most digital cameras perform low-level image processing such as denoising and demosaicking in a hardware image signal processor (ISP) pipeline based on efficient heuristics [Ramanath et al. 2005; Zhang et al. 2011; Shao et al. 2014]. Heide et al. showed in FlexISP that an approach based on formal optimization outperforms conventional ISPs on denoising, deblurring, and other tasks [2014]. Heide et al. later organized the principles of algorithm design in FlexISP into ProxImaL, a domain specific language for optimization based image reconstruction [2016].

3 Realistic image formation model

3.1 Image formation

We consider the image formation for each color channel as

$$\tilde{y} \sim \alpha \mathcal{P}(k * x / \alpha) + \mathcal{N}(0, \sigma^2)$$
$$y = \Pi_{[0,1]}(\tilde{y}),$$

where x is the target scene, y is the measured image, $\alpha > 0$ and $\sigma > 0$ are parameters in a Poisson and Gaussian distribution, respectively, k represents the lens point spread function (PSF), $*$ denotes 2D convolution, and $\Pi_{[0,1]}$ denotes projection onto the interval $[0, 1]$. The measured image thus follows the simple but physically accurate Poisson-Gaussian noise model with clipping described by Foi et al. [2008; 2009].

For simplicity we did not include subsampling of color channels, as in a Bayer pattern, in the image formation model. Subsampling

Figure 2: *A raw frame captured in daylight with a Nexus 5 rear camera (after demosaicking). The image was taken at ISO 3000 with a 30 ms exposure time. The noise in the image is clearly visible.*

amplifies the effects of noise and blur, so whatever negative impact noise and blur have on classification accuracy would only be greater if subsampling was taken into account. Nonetheless, we intend to expand the proposed joint denoising, deblurring, and classification architecture to include demosaicking in future work.

3.2 Calibration

We calibrated the parameters k, α, and σ of the image formation model from Sec. 3.1 for a variety of real-world cameras. Specifically, the PSFs k are estimated using a Bernoulli noise chart with checkerboard features, following Mosleh et al. [2015]. The lens PSF varies spatially in the camera space, so we divided the field-of-view of the camera into non-overlapping blocks and carried out the PSF estimation for each individual block. Fig. 3(a) shows our PSF calibration setup. Fig. 3(c) shows PSFs for the entire field-of-view of a Nexus 5 rear camera.

To estimate the noise parameters α and σ, we took calibration pictures of a chart containing patches of different shades of gray (*e.g.*, [ISO 2014]) at various gains and applied Foi's estimation method [2009]. Fig. 3(b) shows our noise calibration setup. Fig. 3(d) shows plots of $s(x) = \text{std}(\tilde{y})$ versus $E[\tilde{y}]$ and $\hat{s}(\hat{x}) = \text{std}(y)$ versus $E[y]$ for different ISO levels on a Nexus 6P rear camera. The parameters α and σ at a given light level are computed from the $s(x)$ and $\hat{s}(\hat{x})$ plots.

The noise under our calibrated image formation model can be quite high, especially for low light levels. The noisy image in Fig. 1 is an example. Fig. 2 shows a typical capture of a Nexus 5 rear camera captured in low light. This image was acquired for ISO 3000 and a 30 ms exposure time. The only image processing performed on this image was demosaicking. The severe levels of noise present in the image demonstrate that low and medium light conditions represent a major challenge for imaging and computer vision systems. Note that particularly inexpensive low-end sensors will exhibit drastically worse performance compared to higher end smartphone camera modules.

An in-depth description of our calibration procedure is provided in the supplement. Upon acceptance, we will publically release our dataset of camera PSFs and noise curves.

4 Image Classification under Noise and Blur

We evaluated classification performance under the image formation model from Sec. 3.1, calibrated for a Nexus 5 rear camera. We used PSFs from the center, offaxis, and periphery regions of the camera space. The three PSFs are highlighted in Fig. 3(c). We used noise parameters for a variety of lux levels, ranging from moonlight to standard indoor lighting, derived from the ISO noise curves in Fig. 3(d).

We simulated the image formation model for the chosen PSFs and lux levels on the ImageNet validation set of 50, 000 images [Deng et al. 2009]. We then applied the Inception-v4 classification network, one of the state-of-the-art models, to each noised and blurred validation set [Szegedy et al. 2016]. Table 1 shows Inception-v4's Top-1 and Top-5 classification accuracy for each combination of light level and PSF. The drop in performance for low light levels and for the periphery blur is dramatic. Relative to its performance on the original validation set, the network scores almost 60% worse in both Top-1 and Top-5 accuracy on the combination of the lowest light level and the periphery blur.

The results in Table. 1 clearly show that the Inception-v4 network is not robust to realistic noise and blur under low-light conditions. We consider three approaches to improving the classification network's performance in difficult scenarios:

1. We fine-tune the network on training data passed through the image formation model.

2. We denoise and deblur images using standard algorithms before feeding them into the network.

3. We train a novel architecture that combines denoising, deblurring, and classification, which we describe in Sec. 5.

We evaluate all three approaches in Sec. 7.

5 Differentiable Denoising, Deblurring, and Classification Architecture

In this section, we describe the proposed architecture for joint denoising, deblurring, and classification, illustrated in Fig. 4. The architecture combines low-level and high-level image processing units in a pipeline that takes raw sensor data as input and outputs image labels. Our primary contribution is to make the architecture end-to-end differentiable through a principled approach based on formal optimization, allowing us to jointly train low-level and high-level image processing using efficient algorithms such as stochastic gradient descent (SGD). Existing pipeline approaches, such as processing the raw sensor data with a camera ISP before applying a classification network, are not differentiable in the free parameters of the low-level image processing unit with respect to the pipeline output.

We base the low-level image processing unit on the shrinkage fields model, a differentiable architecture for Gaussian denoising and deblurring that achieves near state-of-the-art reconstruction quality [Schmidt and Roth 2014]. We modify the shrinkage fields model using ideas from convolutional neural networks (CNNs) in order to increase the model capacity and make it better suited for training with SGD. We also show how the model can be adapted to handle Poisson-Gaussian noise while preserving differentiability using the generalized Anscombe transform [Foi and Makitalo 2013].

Any differentiable classification network can be used in the proposed pipeline architecture. We use the Inception-v4 convolutional neural network (CNN) evaluated in Sec. 4 [Szegedy et al. 2016].
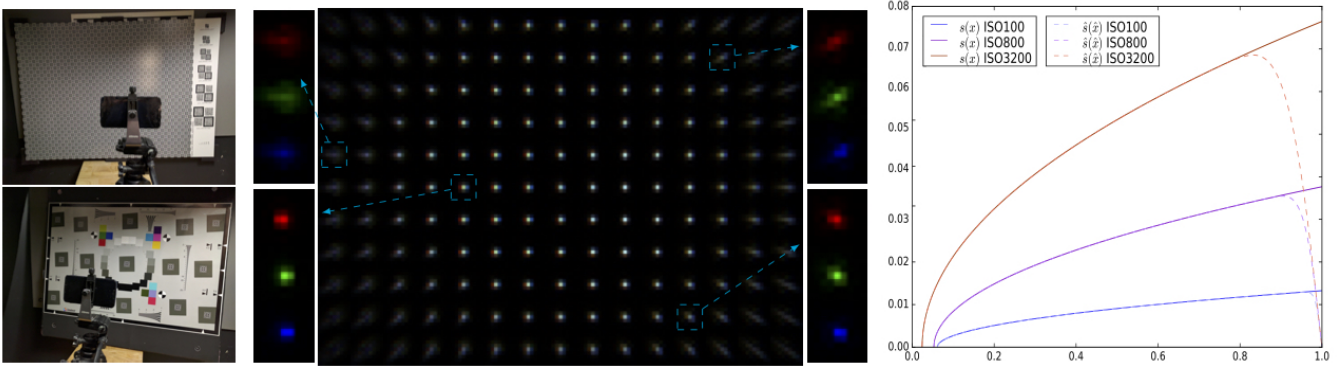
Figure 3: *(a) The PSF calibration setup. (b) The noise calibration setup. (c) The PSFs for the entire field-of-view of a Nexus 5 rear camera. Two center PSFs, an offaxis PSF, and a periphery PSF are magnified. (d) $s(x) = \text{std}(\tilde{y})$ versus $E[\tilde{y}]$ and $\hat{s}(\hat{x}) = \text{std}(y)$ versus $E[y]$ for different ISO levels on a Nexus 6P rear camera. The noise parameters α and σ at a given light level are computed from the $s(x)$ and $\hat{s}(\hat{x})$ plots.*

| | **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%

Light Level

Table 1: *We evaluated pretrained Inception-v4 on the ImageNet validation set passed through the image formation model from Sec. 3.1, calibrated for a Nexus 5 rear camera, under a range of illumination levels and PSFs. Noise increases as light level decreases. The Top-1 and Top-5 classification accuracy decrease substantially in low-light scenarios and with blur from a PSF from the periphery of the camera space.*

The proposed architecture can be adapted to other high-level computer vision tasks such as segmentation, object detection, tracking, and scene understanding by replacing the classification network with a network for the given task.

The outline of the section is as follows. In Sec. 5.1, we motivate the shrinkage fields algorithm through a connection to Bayesian models and formal optimization. In Sec. 5.2, we discuss the previously proposed the shrinkage fields algorithm. In Sec. 5.3, we explain how we modify shrinkage fields to incorporate ideas from CNNs. In Sec. 5.4 and 5.5, we present the low-level image processing units for Poisson-Gaussian denoising and joint denoising and deconvolution. In Sec. 5.6, we explore the connections between the proposed low-level image processing units and structured neural networks.

5.1 Background and motivation

Bayesian model The proposed low-level image processing unit and the shrinkage fields model are inspired by the extensive literature on solving inverse problems in imaging via maximum-a-posteriori (MAP) estimation under a Bayesian model. In the Bayesian model, an unknown image x is drawn from a prior distribution $\Omega(\theta)$ with parameters θ. The sensor applies a linear operator A to the image x, and then measures an image y drawn from a noise distribution $\omega(Ax)$.

Let $P(y|Ax)$ be the probability of sampling y from $\omega(Ax)$ and $P(x;\theta)$ be the probability of sampling x from $\Omega(\theta)$. Then the probability of an unknown image x yielding an observation y is

proportional to $P(y|Ax)P(x;\theta)$.

The MAP estimate of x is given by

$$x = \underset{x}{\arg\max}\, P(y|Ax)P(x;\theta),$$

or equivalently

$$x = \underset{x}{\arg\min}\, f(y, Ax) + r(x, \theta), \quad (1)$$

where the data term $f(y, Ax) = -\log P(y|Ax)$ and prior $r(x, \theta) = -\log P(x;\theta)$ are negative log-likelihoods. Computing x thus involves solving an optimization problem [Boyd and Vandenberghe 2004, Chap. 7].

Unrolled optimization Many algorithms have been developed for solving problem (1) efficiently for different data terms and priors (*e.g.*, FISTA [Beck and Teboulle 2009], Cremers-Chambolle-Pock [Chambolle and Pock 2011], ADMM [Boyd et al. 2001]). The majority of these algorithms are iterative methods, in which a mapping $\Gamma(x^k, A, y, \theta) \to x^{k+1}$ is applied repeatedly to generate a series of iterates that converge to the solution $x^\star$, starting with an initial point x^0.

Iterative methods are usually terminated based on a stopping condition that ensures theoretical convergence properties. An alternative approach is to execute a pre-determined number of iterations N, also known as unrolled optimization. Fixing the number of iterations allows us to view the iterative method as an explicit function
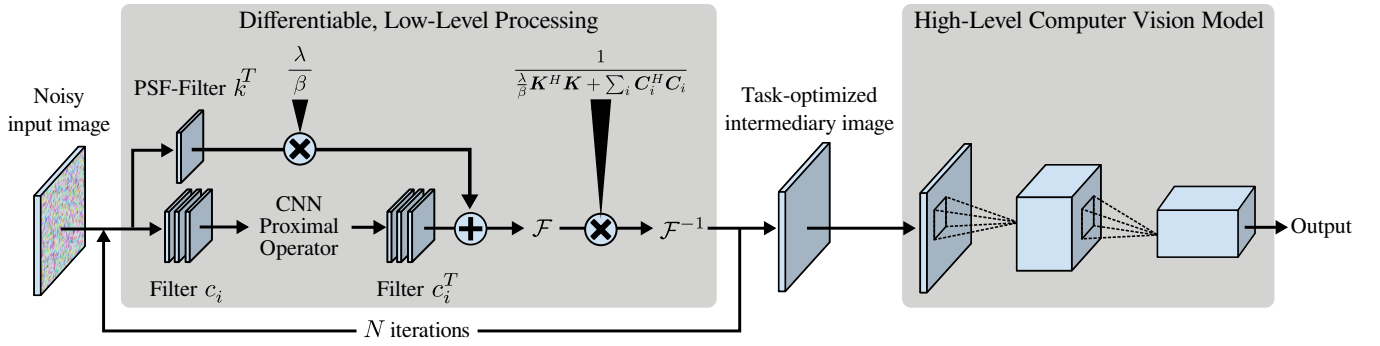
Figure 4: *The proposed architecture for joint denoising, deblurring, and classification combines low-level and high-level image processing in a single pipeline that takes in raw sensor data and outputs image labels. The low-level image processing unit applies an inverse filtering operation based on the image formation model and a learned proximal operator in a recurrent manner. The high-level computer vision model takes the output of the low-level unit and applies a standard classification network. The parameters shown are the filter k with Fourier transform $\mathbf{K}$, learned filters c_i with Fourier transform $\mathbf{C}_i$, and scalars $\lambda, \beta > 0$. For denoising k is the identity, while for joint denoising and deblurring k is the PSF. The operators $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the FFT and inverse FFT. For denoising we apply a slight modification shown in Fig. 7.*

$\Gamma^N(\cdot, A, y, \theta) \to x^N$ of the initial point x^0. Parameters such as θ may be fixed across all iteration or vary by iteration. One can interpret varying parameters as adaptive step sizes or as applying a single iteration of N different algorithms.

If each iteration of the unrolled optimization is differentiable, the gradient of θ and other parameters with respect to a loss function on x^N can be computed efficiently through backpropagation. We can thereby optimize the algorithm for a reconstruction metric such as PSNR or even the loss of a high-level network that operates on x^N (such as Inception-v4).

Parameterization The choice of data term $f(y, Ax)$ is based on the physical characteristics of the sensor, which determine the image formation and noise model. The choice of prior $r(x, \theta)$ is far less clear and has been the subject of extensive research. Classical priors are based on sparsity in a particular (dual) basis, *i.e.*, $r(x, \theta) = g(Cx, \theta)$ for some linear operator C and some norm (or pseudo-norm) $g(\cdot, \theta)$. For example, when C is the discrete gradient operator and $g(\cdot, \theta) = \|\cdot\|_1$, $r(x, \theta)$ is an anisotropic total-variation prior [Rudin et al. 1992]. Other widely used hand-crafted bases include the discrete cosine transform (DCT) and wavelets [Ahmed et al. 1974; Daubechies 1992].

Hand-crafted bases have few if any parameters. We need a richer parameterization in order to learn C. The most flexible parameterization for images assumes that C can be partitioned as

$$Cx = \begin{bmatrix} C_1 x \\ \vdots \\ C_k x \end{bmatrix},$$

where $C_1, \ldots, C_k$ are linear and translation invariant. It follows that each C_i is given by convolution with some filter c_i. Learning C from data means learning the filters $c_1, \ldots, c_k$.

The norm g can also be learned from data. Many iterative methods do not evaluate g directly, but instead access g via its (sub)gradient or proximal operator. The proximal operator $\mathbf{prox}_g$ is defined as

$$\mathbf{prox}_g(y) = \underset{z}{\operatorname{argmin}}\, g(z) + \frac{1}{2}\|z - y\|_2^2.$$

It can thus be simpler to learn the gradient or proximal operator of g

directly and define g implicitly. For ease of exposition, we assume from now on that we learn $\mathbf{prox}_g$.

A common assumption in the literature is that $\mathbf{prox}_g$ is fully separable, meaning given a multi-channel image $z \in \mathbf{R}^{m \times n \times p}$, $\mathbf{prox}_g(z)_{ijk}$ is a function only of z_{ijk} [Roth and Black 2005]. Under this assumption, $\mathbf{prox}_g$ can be parameterized using radial basis functions (RBFs) or any other basis for univariate functions. It is also common to assume that $\mathbf{prox}_g$ is uniform across pixels. In other words, for a given channel k, the function $\mathbf{prox}_g(\cdot)_{ijk}$ is the same for all (i, j). We then only need one parameterization per channel, and $\mathbf{prox}_g$ does not depend on the height and width of the image. The parameterization of C and $\mathbf{prox}_g$ described above is known as the field-of-experts [Roth and Black 2005].

5.2 Shrinkage fields

The shrinkage fields model is an unrolled version of the half-quadratic splitting (HQS) algorithm with the field-of-experts parameterization of C and $\mathbf{prox}_g$ described in Sec. 5.1 [Schmidt and Roth 2014; Geman and Yang 1995]. Fig. 5 illustrates the model. HQS is an iterative method to solve problem (2) when $f(y, Ax) = \frac{\lambda}{2}\|Ax - y\|_2^2$ where $\lambda > 0$, *i.e.*, the noise model is Gaussian. HQS is ideal for unrolled optimization because it can converge in far fewer iterations than other iterative methods (less than 10). HQS lacks the robustness and broad asymptotic convergence guarantees of other iterative methods, but these deficiencies are irrelevent for unrolled optimization with learned parameters.

The HQS algorithm as applied to the optimization problem

$$\text{minimize} \quad \frac{\lambda}{2}\|Ax - y\|_2^2 + g(Cx, \theta), \quad (2)$$

with optimization variable x, is given in Algorithm (1). The idea is to relax problem (2) to the problem

$$\text{minimize} \quad \frac{\lambda}{2}\|Ax - y\|_2^2 + \frac{\beta}{2}\|Cx - z\|_2^2 + g(z, \theta), \quad (3)$$

with optimization variables x and z, and alternately minimize over x and z each iteration while increasing β. Minimizing over z is computing the proximal operator of $\lambda g/\beta$. Minimizing over x is a least-squares problems, whose solution $\hat{x}$ is given by

$$\hat{x} = \left(\frac{\lambda}{\beta}A^T A + C^T C\right)^{-1}\left(\frac{\lambda}{\beta}A^T y + C^T z\right).$$
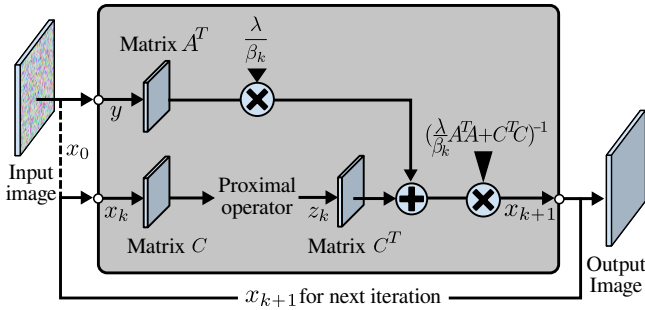
5

Figure 5: *The shrinkage fields model unrolls the HQS algorithm into a recurrent series of iterations. Each iteration solves a least-squares problem based on the data term and applies a proximal operator based on the prior.*

Figure 6: *The color channels can be merged in the proximal operator by summing the input filter responses for each channel and copying the operator output onto each channel before applying the adjoint filters.*

When A is a convolution, $\hat{x}$ can be computed efficiently through inverse filtering.

Algorithm 1 HQS to solve Problem (2)

1: Initialization: $\lambda > 0, 0 < \beta^1 < \cdots < \beta^N, x^0$.
2: **for** $k = 1$ to N **do**
3: $\quad z^k \leftarrow \mathbf{prox}_{\lambda g/\beta^k}(Cx^{k-1})$.
4: $\quad x^k \leftarrow \left(\frac{\lambda}{\beta^k} A^T A + C^T C\right)^{-1} \left(\lambda A^T y + C^T z^k\right)$.
5: **end for**

5.3 CNN proximal operator

RBF parameterization Schmidt and Roth achieve near state-of-the-art denoising and deblurring results by optimizing the shrinkage fields model for average reconstruction PSNR using the L-BFGS algorithm. Their RBF parameterization of $\mathbf{prox}_{\lambda g/\beta^k}$, however, suffers from several deficiencies. The most significant problem is that the RBF parameterization has low representational capacity. Since the RBF parameterization of $\mathbf{prox}_{\lambda g/\beta^k}$ is fully separable, it cannot exploit cross-channel correlations. A small number of basis functions is sufficient to represent arbitrary univariate functions over a fixed range with reasonable precision. Therefore, increasing the number of basis functions or adding additional RBF layers has minimal benefit, so we cannot trade-off computational complexity and representational capacity. A more subtle issue is that storing the gradient of $\mathbf{prox}_{\lambda g/\beta^k}$ with respect to the RBF parameters is memory intensive, which makes optimization via backpropagation challenging on GPUs. We discuss the details of memory usage in the supplement.

CNN parameterization In order to correct the deficiencies of the RBF parameterization, we propose a CNN parameterization of the proximal operator. In particular, we parameterize $\mathbf{prox}_g$ as a CNN with 1×1 kernels with stride 1 and ReLu nonlinearities. This is the same as iterating a fully connected network on the channels over the pixels. The proposed representation is separable and uniform across pixels, but not separable across channels. In other words, $\mathbf{prox}_g(z)_{ijk}$ is a function of $(z_{(i,j,1)}, \ldots, z_{(i,j,p)})$.

A CNN parameterization of the proximal operator has far greater representational capacity than an RBF representation. A CNN can exploit cross-channel correlations, and we can trade-off computational complexity and representational capacity by adding more layers or more units to the hidden layers. We could also increase the representational power of the CNN by breaking our assumption
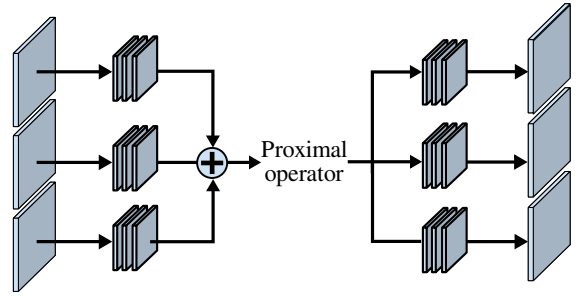
that the proximal operator is separable across pixels and expanding the CNN kernel size.

A further advantage of the CNN representation is that we benefit from the prolific research on optimizing CNNs. Questions such as how to initialize the kernel weights, how to regularize the weights, what optimization algorithm to use, and how to design the network so gradient information flows through it effectively have been explored in depth.

Merging color channels Applying the shrinkage fields architecture with either a RBF or CNN proximal operator to grayscale images is straightforward. If the image has multiple color channels, then multiple approaches are possible. One approach is to apply the architecture to each color channel separately with the same parameters. One could also use different parameters for different channels.

A more sophisticated approach is to merge the color channels within the proximal operator by summing the channel's filter responses Cx^k before applying $\mathbf{prox}_{\lambda g/\beta^k}$ The output z^k of the proximal operator is copied onto each color channel and the adjoint filter response $C^T z^k$ is computed for that channel. Figure 6 shows the proximal operator with merged color channels. Merging the color channels is equivalent to representing C as a single filter with multiple channels. We experiment in Sec. 7 with both keeping the color channels separate and merging the color channels.

5.4 Denoising

For denoising, the image formation model is $A = I$ and the noise model is the Poisson-Gaussian noise discussed in Sec. 3.1. To specialize the shrinkage fields architecture to denoising, we take a three step approach. First we apply the generalized Anscombe transform $\mathcal{A}$ to the measured image y to convert the Poisson-Gaussian noise into IID Gaussian noise [Foi and Makitalo 2013]. Then we apply the shrinkage fields architecture with $A = I$ and a CNN representation of $\mathbf{prox}_{\lambda g/\beta^k}$. The operations inside the low-level image processing unit in Fig. 4 illustrate an iteration of shrinkage fields for denoising. Lastly, we apply the inverse generalized Anscombe transform $\mathcal{A}^{-1}$ to convert the image back into its original domain. The full denoising unit is depicted in Fig. 7. The generalized Anscombe transform and its inverse are differentiable functions, so the Poisson-Gaussian denoising unit is differentiable. Note that the linear operator $\left(\frac{\lambda}{\beta^k} A^T A + C^T C\right)^{-1}$ from Fig. 5 is computed through inverse filtering, since $A = I$ is a convolution.
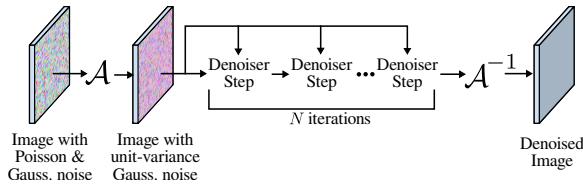
Figure 7: *The full denoising unit applies the generalized Anscombe transform $\mathcal{A}$, followed by N iterations of shrinkage fields specialized to denoising, and finally applies the inverse Anscombe transform $\mathcal{A}^{-1}$.*

5.5 Joint deblurring and denoising

For joint denoising and deblurring, the image formation model is $Ax = k * x$ for a known PSF k and the noise model is the Poisson-Gaussian noise discussed in Sec. 3.1. We cannot apply the generalized Anscombe transform because it would make the image formation model nonlinear. Instead we approximate the Poisson-Gaussian noise as Gaussian noise and apply the shrinkage fields architecture with $A = k*$ and a CNN representation of $\mathbf{prox}_{\lambda_g/\beta^k}$. The joint denoising and deblurring architecture is the low-level image processing unit depicted in Fig. 4. Inverse filtering is used to solve the HQS least-squares subproblem, as in the denoising unit.

5.6 Structured neural network interpretation

The proposed denoising and joint denoising and deblurring units can be viewed either as unrolled optimization algorithms or as structured neural networks, in which the network architecture encodes domain knowledge for a particular task [Wang et al. 2016]. The advantage of structured neural networks is they can exploit domain knowledge, in our case the image formation model. However, there is no standard template for encoding domain knowledge as network structure.

The proposed low-level image processing unit offers just such a template for image reconstruction tasks. We designed our architecture as an unrolled HQS algorithm, which assumes a quadratic data term, but one could combine the CNN parameterization of the prior's proximal operator (or gradient) with other unrolled optimization algorithms that make fewer assumptions. Potential applications exist across all of computational imaging, including depth estimation, sensor fusion, scientific imaging, and medical imaging, to name a few.

For many image reconstruction and image understanding tasks general purpose neural network architectures are sufficient. We believe there are some cases, however, where encoding domain knowledge in the network at a minimum accelerates training and reduces the risk of overfitting and in the best case improves overall performance.

6 Implementation

We built all models in the TensorFlow framework [Abadi et al. 2015]. For the low-level image processing unit, we used only 1 layer of unrolled HQS due to computational constraints. We used 24 5×5 filters with stride 1 for C and 3 convolutional layers with 24 channels for the CNN proximal operator. We kept the color channels separate for the denoising unit but merged them in the CNN proximal operator for the joint deblurring and denoising unit.

We initialized the joint architecture with pretrained parameters for both the low-level image processing unit and Inception-v4. We

pretrained the denoising units by optimizing for average PSNR on 2000 100×100 grayscale image patches from the BSDS300 training dataset [Martin et al. 2001]. Similarly, we pretrained the joint denoising and deblurring units by optimizing for average PSNR on 200 180×180 color image patches from BSDS300. We used 400 iterations of the L-BFGS optimization algorithm in all cases. We discuss the L-BFGS initialization for the low-level image processing units and further details of the units' parameterization in the supplement.

The Inception-v4 and joint architectures were fine-tuned on the full ImageNet training set passed through the image formation model in Sec. 3.1. We used RMSProp [Tieleman and Hinton 2012] with a decay of 0.9, $\epsilon = 1.0$, and a learning rate of $4.5e^{-3}$, exponentially decayed by a factor of 0.94 every epoch. We trained all models for 2 epochs. Fine-tuning took 1 day for the Inception-v4 models, 5 days for the joint denoising and clasification models, and 3 days for the joint denoising, deblurring, and classification models. We used 4 NVIDIA Tesla K80 GPUs per model.

7 Evaluation

7.1 Results

We selected four combinations of light level and camera region from Table 1 for which the classification accuracy of the pretrained Inception-v4 network on the ImageNet validation set dropped substantially relative to its performance without noise or blur. We evaluated three methods of improving the classification accuracy:

1. We applied conventional denoising and deblurring algorithms to the noised and blurred validation set images, then evaluated the pretrained network on the processed images.

2. We fine-tuned Inception-v4 on the 1.2 million ImageNet training images passed through the image formation model for the given light level and PSF.

3. We fine-tuned the joint denoising, deblurring, and classification architecture described in Sec. 5 on the ImageNet training images passed through the image formation model.

Table 2 summarizes the results. For the two cases with noise but no blur, denoising the images with non-local means (NLM) [Buades et al. 2005] decreased Top-1 and Top-5 classification accuracy by over 10%, while denoising with BM3D [Danielyan et al. 2012] increased Top-1 and Top-5 accuracy by a few percent. For the combination of 6 lux illumination and offaxis blur, denoising and deblurring the images with HQS [Geman and Yang 1995] decreased Top-1 and Top-5 classification accuracy by over 10%. For the combination of 6 lux illumination and periphery blur, preprocessing the images with HQS decreased accuracy by a few percent. Overall, denoising and deblurring the images with conventional algorithms at best marginally increased classification accuracy and in many cases substantially decreased performance.

For all four cases, fine-tuning Inception-v4 on images passed through the image formation model improved classification accuracy substantially, by 10s of percent. The Top-1 and Top-5 classification accuracy were still worse than in the noise and blur free case, but the gap was reduced dramatically.

The highest classification accuracy, however, was obtained by the joint architecture. Top-1 accuracy was up to 5.1% higher than the fine-tuned classifier, and Top-5 accuracy was up to 3.7% higher. The benefit of tuning the denoiser and deblurrer with the classification network was far greater than that of combining a conventional denoiser or deblurrer with the pretrained network.

	3 lux		6 lux		6 lux + Offaxis PSF		6 lux + Periphery PSF	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Pretrained Inception-v4	45.54%	68.53%	63.57%	84.51%	61.07%	82.03%	37.69%	59.06%
Fine-Tuned Inception-v4	64.49%	85.73%	73.19%	91.32%	68.32%	88.44%	61.02%	83.32%
Joint Architecture	**69.58%**	**89.30%**	**74.79%**	**92.34%**	**71.08%**	**90.24%**	**66.03%**	**87.01%**
NLM + Pretrained	32.19%	52.12%	56.04%	77.85%	-	-	-	-
BM3D + Pretrained	47.43%	70.35%	67.00%	87.47%	-	-	-	-
HQS + Pretrained	-	-	-	-	48.64%	70.98%	36.21%	57.04%

Table 2: *We evaluated three approaches to improving classification performance in four challenging scenarios with low-light and blur: fine-tuning Inception-v4 on noisy and blurry images, training the proposed joint denoising, deblurring, and classification architecture, and cleaning up the images with standard denoising and deblurring algorithms. Fine-tuning improved Top-1 and Top-5 classification accuracy substantially over the pretrained baseline, but the joint architecture achieved the best results in all four scenarios, improving on the fine-tuned Top-1 accuracy by up to 5.1% and the Top-5 accuracy by up to 3.7%. Denoising and deblurring the images with conventional algorithms at best improved classification accuracy by a few percent over the pretrained baseline and in many cases substantially decreased accuracy.*

7.2 Interpretation

The results in Table 2 raise the question of why the jointly tuned denoising and deblurring algorithms were so much more helpful to the classifier than conventional algorithms. The images in Figure 8 suggest an answer. Figure 8 shows an image for each of the four combinations of noise and blur that was incorrectly classified by the pretrained Inception-v4 network but correctly classified by the joint architecture. The noised and blurred image is shown, as well as the noised and blurred image denoised and deblurred by a conventional algorithm and by the jointly tuned denoising and deblurring unit. The label assigned by the classifier is given in each instance, as well as the PSNR relative to the original image.

The images output by the conventional denoising and deblurring algorithms contain far less noise than the original noisy and blurry image. Fine details in many regions are blurred out, however. The image output by the jointly tuned denoising and deblurring unit, by contrast, contains more noise but preserves more fine detail.

By conventional metrics of restoration quality such as PSNR and visual sharpness, the jointly tuned denoising and deblurring unit is worse than conventional algorithms. We can see though that it preserves fine detail that is useful to the classification network, while still denoising and deblurring the image to an extent. The qualitative results suggest that the reconstruction algorithms and metrics used to make images visually pleasing to humans are not appropriate for high-level computer vision architectures.

8 Discussion

In summary, we showed that the performance of a classification architecture decreases significantly under realistic noise and blur scenarios. We make classification robust to noise and blur by introducing a new fully-differentiable architecture that combines denoising, deblurring, and classification. The architecture is based on unrolled optimization algorithms and can easily be adapted to a different high-level task or image formation model.

We demonstrate that the proposed architecture dramatically improves classification accuracy under noise and blur, surpassing other approaches such as fine-tuning the classification network on blurry and noisy images and preprocessing images with a conventional denoising or deblurring algorithm. We highlight major qualitative differences between the denoised and deblurred images produced as intermediate representations in the proposed architecture and the output of conventional denoising and deblurring algorithms. Our results suggest that the image processing most helpful to a deep network for classification or some other high-level task is different from the traditional image processing designed to produce visually pleasing images.

Limitations As discussed in Sec. 3, our image formation model is an accurate representation of real world cameras. The only major simplification in our model is that we do not handle demosaicking. We intend to add demosaicking to our low-level image processing unit in future work. Another obstacle to real world deployment of the proposed architecture is that the noise level and PSF must be known at runtime, both because they are hard-coded into the low-level image processing unit and also because the architecture is trained on a specific noise level and PSF.

A simple solution to the dependence on the noise level and PSF is to train an ensemble of models for different noise levels and PSFs, run them all when classifying an image, and assign the label given the highest confidence. Another possible approach that we will explore in future work is to parameterize the model with the noise level so that retraining for different noise levels is not required.
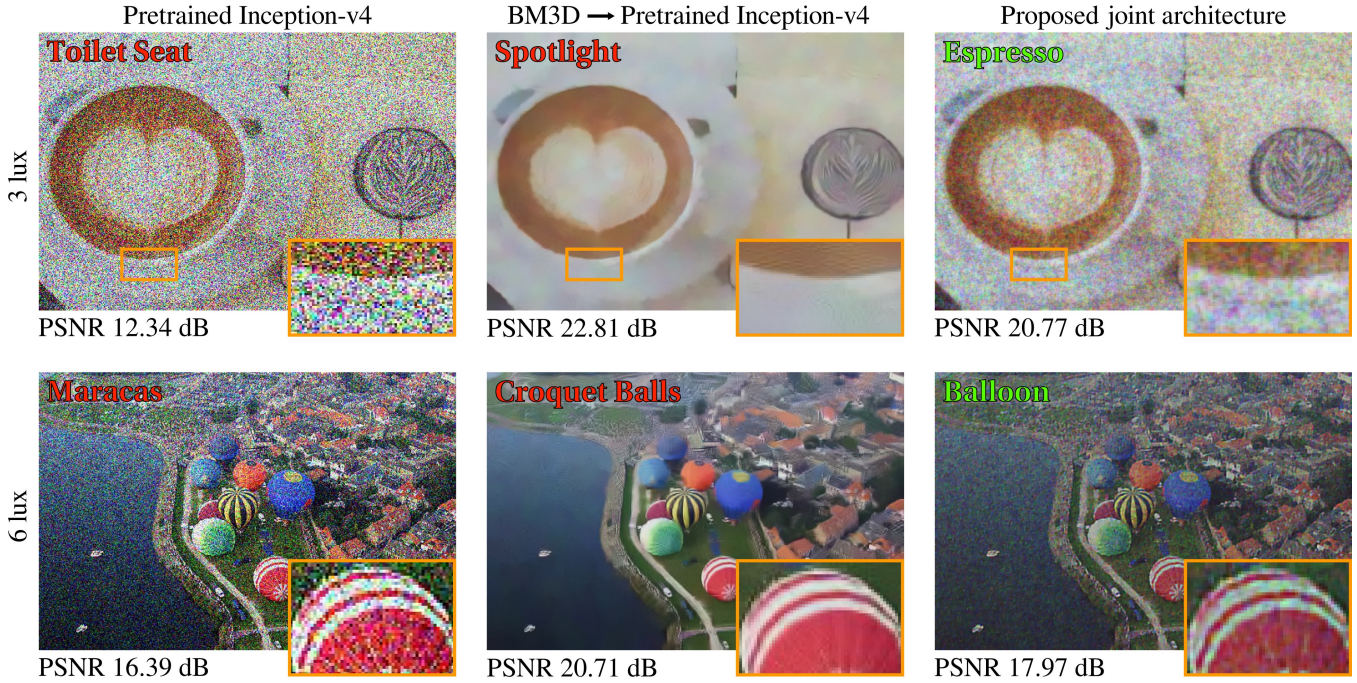
Future Work In the future, we will apply the principles outlined in this paper for designing joint architectures that combine low-level and high-level image processing to many other problems in computational photography and computational imaging. We believe unrolled optimization algorithms with CNN proximal operators (or gradients) can achieve state-of-the-art results for many generative and discriminative tasks.

We will also expand the proposed architecture to model the camera lens and sensors. Just as we optimized the denoiser and deblurrer for classification, we aim to optimize the lens, color subsampling pattern, and other elements of the imaging system for the given high-level vision task. We plan on investigating a similar approach to optimizing the optical system for other imaging modalities as well.

9 Conclusion

In the future most images taken by cameras and other imaging systems will be consumed by high-level computer vision architectures, not by humans. We must reexamine the foundational assumptions of image processing in light of this momentous change. Image re-

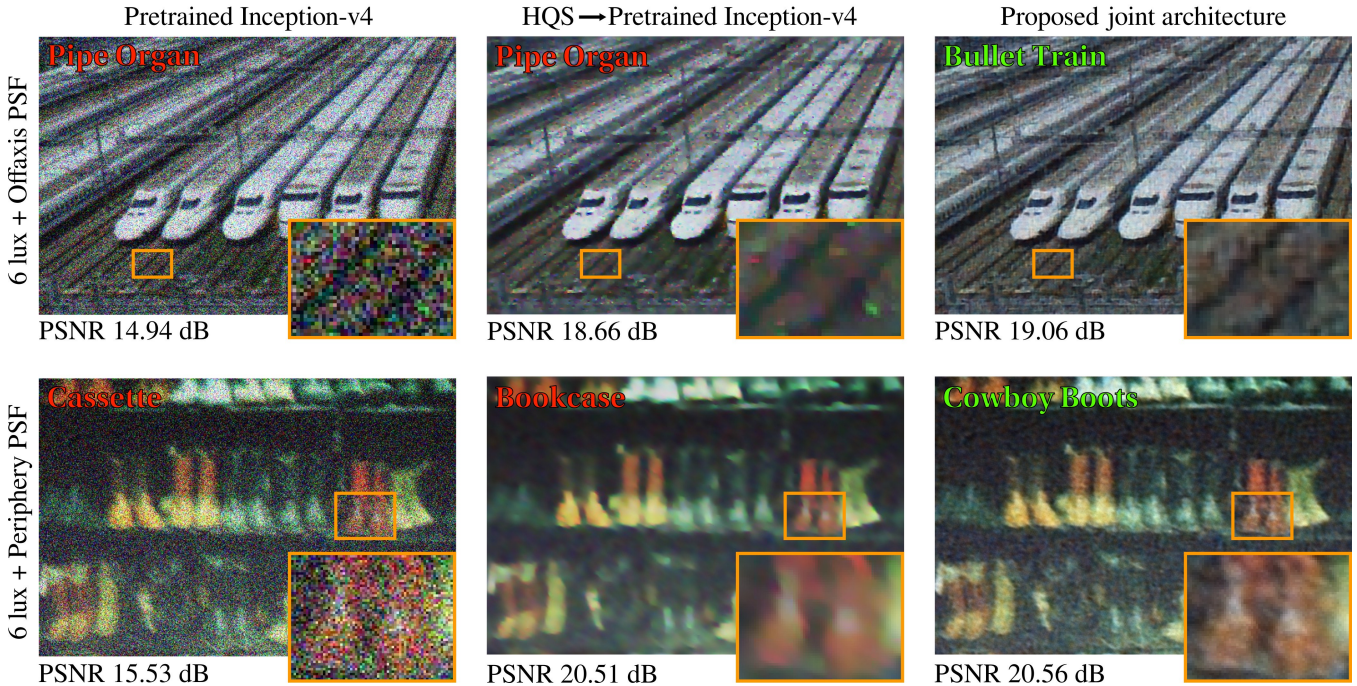**Classification performance on noisy images**

Figure 8: *We show images with simulated noise and blur for each of the four combinations of light level and PSF evaluated in Table 2. (Left) The noisy and blurry images are misclassified by the pretrained Inception-v4 network. (Center) Denoising the images with BM3D or jointly denoising and deblurring with HQS does not lead to correct classifications. (Right) The proposed jointly trained denoising, deblurring, and classification architecture learns low-level image processing pipelines optimized for classification that enhance fine detail at the expense of more noise and artifacts. The images processed with the learned pipeline are correctly classified, yet score poorly on conventional metrics of reconstruction quality such as PSNR.*

construction algorithms designed to produce visually pleasing images for humans are not necessarily appropriate for computer vision pipelines. We have proposed one approach to redesigning low-level image processing to better serve high-level imaging tasks, in a way that incorporates and benefits from knowledge of the physical image formation model. But ours is only the first, not the final word in a promising new area of research.

Acknowledgements

References

ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G. S., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., GOODFELLOW, I., HARP, A., IRVING, G., ISARD, M., JIA, Y., JOZEFOWICZ, R., KAISER, L., KUDLUR, M., LEVENBERG, J., MANÉ, D., MONGA, R., MOORE, S., MURRAY, D., OLAH, C., SCHUSTER, M., SHLENS, J., STEINER, B., SUTSKEVER, I., TALWAR, K., TUCKER, P., VANHOUCKE, V., VASUDEVAN, V., VIÉGAS, F., VINYALS, O., WARDEN, P., WATTENBERG, M., WICKE, M., YU, Y., AND ZHENG, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

AGOSTINELLI, F., ANDERSON, M. R., AND LEE, H. 2013. Adaptive multi-column deep neural networks with application to robust image denoising. In *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., 1493–1501.

AHMED, N., NATARAJAN, T., AND RAO, K. 1974. Discrete cosine transform. *IEEE Transactions on Computers C-23*, 1, 90–93.

BECK, A., AND TEBOULLE, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences 2*, 1, 183–202.

BOYD, S., AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press.

BOYD, S., PARIKH, N., CHU, E., PELEATO, B., AND ECKSTEIN, J. 2001. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning 3*, 1, 1–122.

BRUCK, R. 1975. An iterative solution of a variational inequality for certain monotone operators in Hilbert space. *Bulletin of the American Mathematical Society 81*, 5 (Sept.), 890–892.

BUADES, A., COLL, B., AND MOREL, J.-M. 2005. A non-local algorithm for image denoising. In *Proc. IEEE CVPR*, vol. 2, 60–65.

BURGER, H., SCHULER, C., AND HARMELING, S. 2012. Image denoising: Can plain neural networks compete with BM3D? In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2392–2399.

CHAKRABARTI, A. 2016. A neural approach to blind motion deblurring. In *Proceedings of the European Conference on Computer Vision*.

CHAMBOLLE, A., AND POCK, T. 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision 40*, 1, 120–145.

CHEN, Y., AND POCK, T. 2015. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *arXiv preprint arXiv:1508.02848*.

CHEN, Y., YU, W., AND POCK, T. 2015. On learning optimized reaction diffusion processes for effective image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5261–5269.

CHEN, G., LI, Y., AND SRIHARI, S. 2016. Joint visual denoising and classification using deep learning. In *Proceedings of the IEEE International Conference on Image Processing*, 3673–3677.

DA COSTA, G. B. P., CONTATO, W. A., NAZARE, T. S., NETO, J. E., AND PONTI, M. 2016. An empirical study on the effects of different types of noise in image classification tasks. *arXiv preprint arXiv:1609.02781*.

DANIELYAN, A., KATKOVNIK, V., AND EGIAZARIAN, K. 2012. BM3D frames and variational image deblurring. *IEEE Trans. Image Processing 21*, 4, 1715–1728.

DAUBECHIES, I. 1992. *Ten lectures on wavelets*, vol. 61. SIAM.

DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K., AND FEI-FEI, L. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 248–255.

DODGE, S., AND KARAM, L. 2016. Understanding how image quality affects deep neural networks. *arXiv preprint arXiv:1604.04004*.

DONG, C., LOY, C. C., HE, K., AND TANG, X. 2014. Learning a deep convolutional network for image super-resolution. In *Proceedings of the European Conference on Computer Vision*, 184–199.

FOI, A., AND MAKITALO, M. 2013. Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise. *IEEE Trans. Image Process. 22*, 1, 91–103.

FOI, A., TRIMECHE, M., KATKOVNIK, V., AND EGIAZARIAN, K. 2008. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Trans. Image Process. 17*, 10, 1737–1754.

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

GEMAN, D., AND YANG, C. 1995. Nonlinear image recovery with half-quadratic regularization. *IEEE Trans. Image Processing 4*, 7, 932–946.

GHARBI, M., CHAURASIA, G., PARIS, S., AND DURAND, F. 2016. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG) 35*, 6, 191.

GLOWINSKI, R., AND MARROCO, A. 1975. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique 9*, 2, 41–76.

HEIDE, F., STEINBERGER, M., TSAI, Y.-T., ROUF, M., PAJAK, D., REDDY, D., GALLO, O., LIU, J., HEIDRICH, W., EGIAZARIAN, K., KAUTZ, J., AND PULLI, K. 2014. FlexISP:

A flexible camera image processing framework. *ACM Trans. Graph. (SIGGRAPH Asia) 33*, 6.

HEIDE, F., DIAMOND, S., NIESSNER, M., RAGAN-KELLEY, J., HEIDRICH, W., AND WETZSTEIN, G. 2016. ProxImaL: Efficient image optimization using proximal algorithms. *ACM Trans. Graph. 35*, 4.

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

JAIN, V., AND SEUNG, S. 2009. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, 769–776.

JALALVAND, A., NEVE, W. D., DE WALLE, R. V., AND MARTENS, J. 2016. Towards using reservoir computing networks for noise-robust image recognition. In *Proceedings of the International Joint Conference on Neural Networks*, 1666–1672.

JIN, K. H., MCCANN, M. T., FROUSTEY, E., AND UNSER, M. 2016. Deep convolutional neural network for inverse problems in imaging. *arXiv preprint arXiv:1611.03679*.

KIM, J., LEE, J., AND LEE, K. 2016. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1646–1654.

MARTIN, D., FOWLKES, C., TAL, D., AND MALIK, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, 416–423.

MOSLEH, A., GREEN, P., ONZON, E., BEGIN, I., AND PIERRE LANGLOIS, J. 2015. Camera intrinsic blur kernel estimation: A reliable framework. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

OCHS, P., RANFTL, R., BROX, T., AND POCK, T. 2015. Bilevel optimization with nonsmooth lower level problems. In *International Conference on Scale Space and Variational Methods in Computer Vision*, Springer, 654–665.

OCHS, P., RANFTL, R., BROX, T., AND POCK, T. 2016. Techniques for gradient-based bilevel optimization with non-smooth lower level problems. *Journal of Mathematical Imaging and Vision*, 1–20.

POCK, T., CREMERS, D., BISCHOF, H., AND A.CHAMBOLLE. 2009. An algorithm for minimizing the Mumford-Shah functional. In *Proceedings of the IEEE International Conference on Computer Vision*, 1133–1140.

RAMANATH, R., SNYDER, W., YOO, Y., AND DREW, M. 2005. Color image processing pipeline in digital still cameras. *IEEE Signal Processing Magazine 22*, 1, 34–43.

ROTH, S., AND BLACK, M. J. 2005. Fields of experts: A framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, IEEE, 860–867.

RUDIN, L., OSHER, S., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena 60*, 14, 259 – 268.

SCHMIDT, U., AND ROTH, S. 2014. Shrinkage fields for effective image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2774–2781.

SCHULER, C. J., CHRISTOPHER BURGER, H., HARMELING, S., AND SCHOLKOPF, B. 2013. A machine learning approach for non-blind image deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

SCHULER, C., HIRSCH, M., HARMELING, S., AND SCHÖLKOPF, B. 2014. Learning to deblur. In *NIPS 2014 Deep Learning and Representation Learning Workshop*.

SHAO, L., YAN, R., LI, X., AND LIU, Y. 2014. From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms. *IEEE Transactions on Cybernetics 44*, 7, 1001–1013.

SZEGEDY, C., IOFFE, S., AND VANHOUCKE, V. 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*.

TANG, Y., AND ELIASMITH, C. 2010. Deep networks for robust visual recognition. In *Proceedings of the International Conference on Machine Learning*, 1055–1062.

TANG, Y., SALAKHUTDINOV, R., AND HINTON, G. 2012. Robust boltzmann machines for recognition and denoising. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2264–2271.

TIELEMAN, T., AND HINTON, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning 4*, 2.

VASILJEVIC, I., CHAKRABARTI, A., AND SHAKHNAROVICH, G. 2016. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*.

WANG, S., FIDLER, S., AND URTASUN, R. 2016. Proximal deep structured models. In *Advances in Neural Information Processing Systems 29*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. 865–873.

XIE, J., XU, L., AND CHEN, E. 2012. Image denoising and inpainting with deep neural networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, 341–349.

XU, L., REN, J. S., LIU, C., AND JIA, J. 2014. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, 1790–1798.

ZHANG, L., WU, X., BUADES, A., AND LI, X. 2011. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging 20*, 2, 023016–023016.

ZHANG, J., PAN, J., LAI, W.-S., LAU, R., AND YANG, M.-H. 2016. Learning fully convolutional networks for iterative non-blind deconvolution. *arXiv preprint arXiv:1611.06495*.

ZHANG, K., ZUO, W., CHEN, Y., MENG, D., AND ZHANG, L. 2016. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *arXiv preprint arXiv:1608.03981*.