Deep joint demosaicking and denoising

Deep Joint Demosaicking and Denoising

Michaël Gharbi
MIT CSAIL

Gaurav Chaurasia
MIT CSAIL

Sylvain Paris
Adobe

Frédo Durand
MIT CSAIL

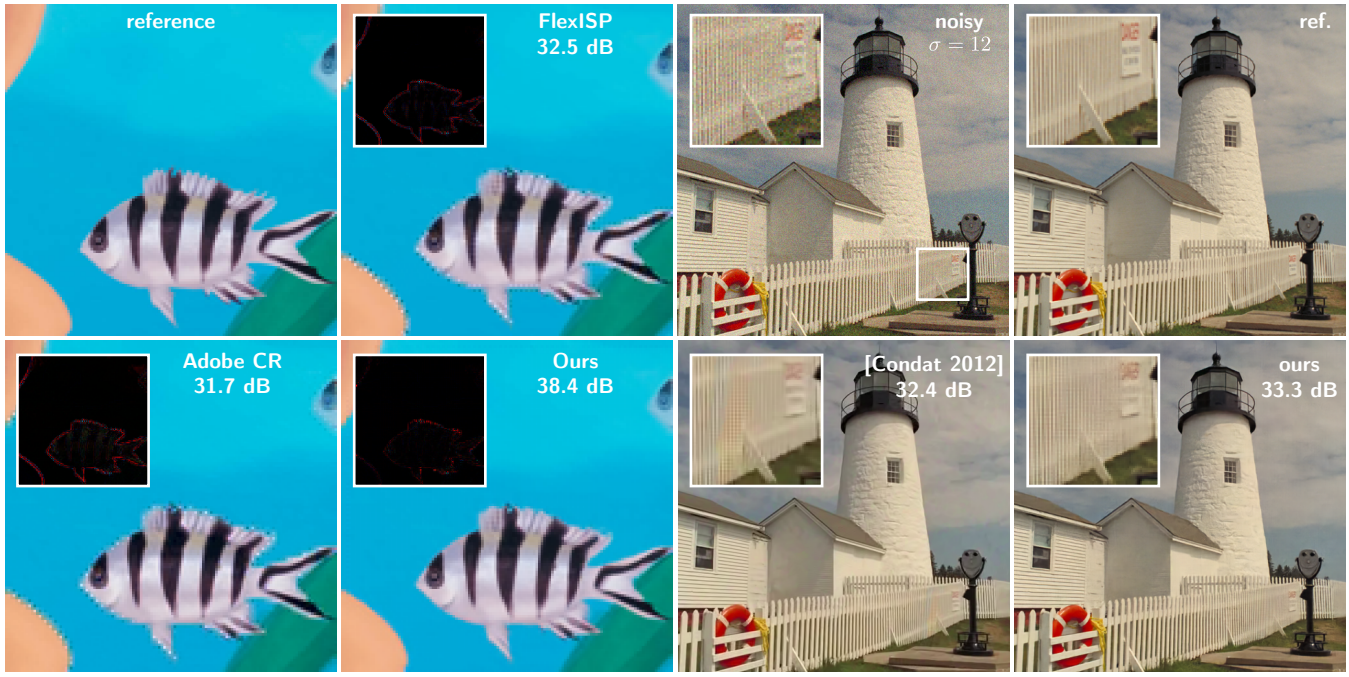

Figure 1: *We propose a data-driven approach for jointly solving denoising and demosaicking. By carefully designing a dataset made of rare but challenging image features, we train a neural network that outperforms both the state-of-the-art and commercial solutions on demosaicking alone (group of images on the left, insets show error maps), and on joint denoising–demosaicking (on the right, insets show close-ups). The benefit of our method is most noticeable on difficult image structures that lead to* moiré *or* zippering *of the edges.*

Abstract

Demosaicking and denoising are the key first stages of the digital imaging pipeline but they are also a severely ill-posed problem that infers three color values per pixel from a single noisy measurement. Earlier methods rely on hand-crafted filters or priors and still exhibit disturbing visual artifacts in hard cases such as moiré or thin edges. We introduce a new data-driven approach for these challenges: we train a deep neural network on a large corpus of images instead of using hand-tuned filters. While deep learning has shown great success, its naive application using existing training datasets does not give satisfactory results for our problem because these datasets lack hard cases. To create a better training set, we present metrics to identify difficult patches and techniques for mining community photographs for such patches. Our experiments show that this network and training procedure outperform state-of-the-art both on noisy and noise-free data. Furthermore, our algorithm is an order of magnitude faster than the previous best performing techniques.

Keywords: deep learning, demosaicking, denoising, data driven methods, convolutional neural networks

Concepts: •**Computing methodologies** → **Computational photography;** *Image processing;*

SA '16 Technical Papers,, December 05-08, 2016, Macao
ISBN: 978-1-4503-4514-9/16/12
DOI: http://dx.doi.org/10.1145/2980179.2982399

1 Introduction

Demosaicking and denoising are simultaneously the crucial first steps of most digital camera pipelines. They are quintessentially ill-posed reconstruction problems: at least two-thirds of the data is missing and the existing data is corrupted with noise. Furthermore, complex aliasing issues arise because the red, green and blue channels are sampled at different locations and at different rates. while most image areas are easy to address, the rare challenging regions can still lead to catastrophic failure and visually disturbing artifacts such as checkerboard patterns, zippering around edges, and moiré.

For modularity, demosaicking and denoising are often solved independently and sequentially. This unfortunately leads to error accumulation because demosaicking needs to cope with unreliable samples and denoising suffers from the non-linear and variable per-pixel noise introduced by demosaicking. It has long been recognized that exploiting the regularity of natural images is key to lifting underdetermination. Traditional techniques have hard-coded hand-crafted heuristics into local filters [Cok 1987; Laroche and Prescott 1994;

Buades et al. 2009]. Heide et al. [2014] proposed a joint solution to denoising and demosaicking by embedding a non-local natural image prior into an optimization approach. However, their prior is still hand-crafted and the combination of optimization and a non-local prior leads to a steep increase in computation cost.

In contrast, we address demosaicking and denoising jointly using a data-driven local filtering approach for efficiency. We train our model on a large set of ground truth data to optimally leverage regularities found in natural images. We build on the success of deep learning and convolutional neural networks, e.g. [LeCun et al. 2015]. While data-driven local-filtering has been explored previously [Klatzer et al. 2016; Tian et al. 2014; Lansel and Wandell 2011], assembling a quality training set is always key and we found that the characteristics of demosaicking and denoising make this a challenge, in particular because catastrophically hard inputs are rare and because salient artifacts are not well captured by standard image metrics. Another challenge is that deep learning often requires to train a new network or to fine-tune an existing one for even slightly different instances of a problem. This is particularly problematic for issues such as sensor noise, whose strength varies with the ISO setting, and other imaging characteristics.

Our contributions to joint denoising-demosaicking are a Convolutional Neural Network capable of handling a *wide range* of noise levels and a procedure to build a training set rich in challenging images prone to moiré and artifacts. We demonstrate that our approach enables higher-quality results than previous work and runs faster on both CPU and GPU.

2 Related Work

Demosaicking is a well-studied problem and most algorithms perform well in flat regions of the image. But all tend to struggle around strong edges and textured areas (Figure 1). This leads to conspicuous artifacts such as *zippering*, color moiré and loss of detail. Many approaches derive edge-adaptive interpolation schemes to control such artifacts [Laroche and Prescott 1994]. A popular solution is to design nonlinear filters that avoid interpolating across the strong local edges [Li et al. 2008]. The key ingredient for demosaicking is to leverage cross-channel dependencies to recover details beyond the Nyquist frequency of each channel. Correlations between color channels can be captured by the *smooth hue* prior [Cok 1987] where color ratios or differences are modeled as smoothly varying signals. Algorithms based on this heuristic interpolate channels sequentially starting with the luminance component i.e. green channel [Zhang et al. 2009; Chang and Tan 2004]. The demosaicked green channel is then used to guide the chrominance interpolation. In these techniques, image quality is adversely affected when the smooth hue heuristic does not hold, leading to false color (Figure 1). Hirakawa et al. [2005] use median filtering on color differences to mitigate the effect. But such post-processing techniques have drawbacks like excessive blurring, and do not fundamentally change the issue of color moiré. We propose to replace hand-crafted filters by a machinery that can jointly interpolate the three color channels, is fully trainable and can learn to disambiguate error-prone patterns directly from natural images without relying on hard-coded heuristics.

Self-similarity and data-driven demosaicking Recent methods overcome the ill-posedness of demosaicking by exploiting local self-similarity in natural images and fill in the missing color information from similar neighboring patches [Buades et al. 2009; Zhang et al. 2011]. He et al. [2012] use SVM regression to learn on-line a demosaicking process tailored to the input image. Another approach to the demosaicking problem is to employ machine-learning. Kwan et al. [2004] adopt a classification approach to select one of two discrete directions of interpolation with hand-designed features. Some techniques employ fully connected shallow neural network architectures with small spatial footprints [Go et al. 2000; Kapah and Hel-Or 2000]. Early data-driven techniques used simple architectures and hard-coded heuristics. They were trained on small datasets of up to a few hundred images, and do not compare favorably with the state-of-the-art. This has been attributed to the lack of appropriate training datasets [Zhang et al. 2009]. Learning-based methods enable experimentation with new sensor designs and alternative mosaick patterns [Lansel and Wandell 2011; Tian et al. 2014]. In this work, we gather a dataset of millions of difficult patches from online photo collections according to the severity of the artifacts produced by a baseline demosaicking method. We train a model directly from the input mosaick to the final color image and achieve state-of-the art quality.

Joint denoising and demosaicking Demosaicking is further complicated by the presence of noise. Estimates of edge orientation in noisy data are less reliable which leads to noticeable artifacts in the demosaicked image. The techniques that perform these steps sequentially usually start with denoising [Park et al. 2009]. A notable exception, Akiyama et al. [2015] first denoise the Bayer array viewed as a four-channels quarter-resolution image. Recent attempts have shown the advantages of joint approaches [Hirakawa and Parks 2006; Condat and Mosaddegh 2012]. Jeon and Dubois [2013] optimize a set of filters for discrete noise levels. Heide et al [2014] use a global primal-dual optimization with a self-similarity prior. The nearest neighbor search and the iterative nature of the algorithm makes it slow and somewhat impractical. Khashabi et al. [2014] demonstrate a learning approach that generalizes to non-Bayer mosaick patterns. Klatzer et al. [2016] use a sequential energy minimization approach which can be interpreted as a convolutional network with trainable activation functions and where intermediate layers are constrained to output a color image. Klatzer et al. can learn a noise model from data but this model is tailored to a single noise level and fixed after training. Instead, we expose a runtime parameter and train our network so it adapts to a wide range of noise levels.

Neural networks for image processing Convolutional neural networks (CNN) have revolutionized classification problems in computer vision [Krizhevsky et al. 2012; Szegedy et al. 2015; Simonyan and Zisserman 2014]. They are also rapidly becoming a mainstream tool in image processing tasks like pixel-wise object segmentation [Long et al. 2015; Badrinarayanan et al. 2015; Noh et al. 2015], depth and normals estimation from a single image [Eigen et al. 2014; Wang et al. 2015], view interpolation [Flynn et al. 2016], deconvolution [Xu et al. 2014], filter approximation [Xu et al. 2015], image colorization [Cheng et al. 2015; Zhang et al. 2016; Larsson et al. 2016; Iizuka et al. 2016], style-transfer [Gatys et al. 2016], optical flow [Dosovitskiy et al. 2015a], image inpainting [Eigen et al. 2013; Pathak et al. 2016] and image synthesis [Dosovitskiy et al. 2015b].

3 Convolutional Neural Network for Joint Demosaicking and Demoising

Demosaicking and denoising have traditionally been addressed using nonlinear filter design, incorporating prior heuristics about inter- and intra-channel correlation, behavior around edges, and exploiting intra-image patch similarity. A convolutional network seems a natural choice for the problem in this context. First, it enables discovery of natural correlations in the data. Second, the network can represent a superset of the pipelines implemented by many previous techniques while all its parameters are optimized jointly to minimize a single objective.

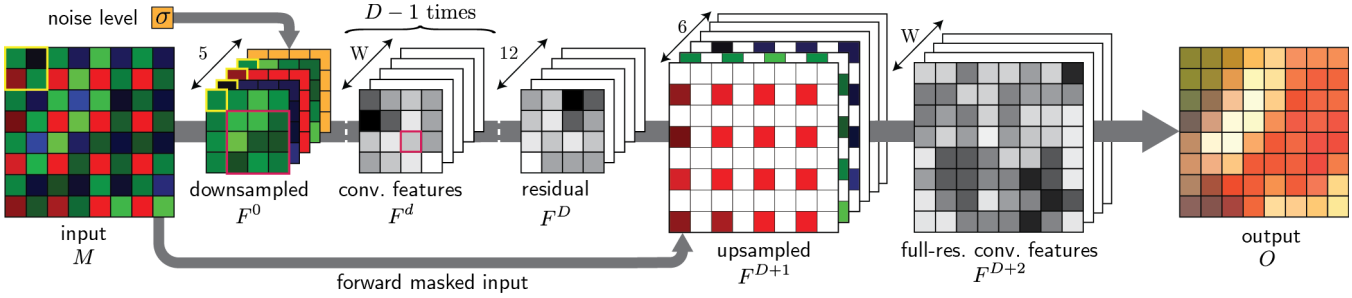

Figure 2: *Our proposed architecture. The first layer of the network packs 2×2 blocks in the Bayer image into a 4D vector to restore translation invariance and speed up the processing. We augment each vector with the noise parameter σ to form 5D vectors. Then, a series of convolutional layers filter the image to interpolate the missing color values. We finally unpack the 12 color samples back to the original pixel grid and concatenate a masked copy of the input mosaick. We perform a last group of convolutions at full resolution this time to produce the final features. We linearly combine them to produce the demosaicked output.*

A network alone is not sufficient to tackle denoising/demosaicking. We will see in § 4 that the choice of training data has critical impact, especially because difficult inputs are rare yet cause visually disturbing artifacts.

We cast joint denoising and demosaicking as a supervised learning problem: we train our algorithm on a set of input measurements for which the desired output is known. We create the training set from millions of sRGB images, generating the corresponding mosaicked arrays by leaving out two color channels per pixel and adding noise. We then build a convolutional neural network and train it in an end-to-end fashion. The inputs are the mosaicked array M with a single channel per pixel and an estimate σ of the noise level; the output is an image O of the same size with a RGB triplet per pixel. We start our exposition focusing on demosaicking and then discuss noise.

3.1 Network architecture

We use a standard feed-forward network architecture to implement our demosaicking operator (Figure 2). Our network is composed of $D+1$ convolutional layers. Each convolution layer has W outputs and uses kernels of size $K \times K$. We denote by F^d the feature map of the d-th layer. In addition to the input mosaick M, the network takes as input an estimate of the noise level σ. We first describe the general architecture of the network. Details on how σ comes into play can be found in § 3.2. Since the Bayer mosaick is ubiquitous, we specialize our network to exploit its structure. We show however in § 5 that our approach generalizes to non-Bayer patterns.

We first rearrange the samples of the Bayer input mosaick to obtain a quarter-resolution multi-channel image which makes the spatial pattern translation invariant with a period of 1 pixel and reduces the computational cost of the subsequent steps. The first layer F^0 extracts 2×2 patches from M and packs them as a 4 channel feature map indexed by c.

$$F_c^0(x,y) = \mathrm{M}\left(2x + (c \bmod 2),\ 2y + \left\lfloor \frac{c}{2} \right\rfloor\right) \tag{1}$$

The bulk of the processing is performed at this lower resolution by the next D layers. They share the same structure and consist in convolutions with a bank of filters of spatial footprint $K \times K$ followed by a point-wise ReLU non-linearity $f(\cdot) = \max(0, \cdot)$.

$$F_c^d = f\left(b_c^d + \sum_{c'=1}^{W} w_{cc'}^d * F_{c'}^{d-1}\right) \text{ for } c \in \{1 \dots W\} \tag{2}$$

b_c^d is a scalar bias for the c-th channel of layer d, and $w_{cc'}^d$ is a two-dimensional convolution kernel of size $K \times K$. Each layer uses a total of W^2 such filters. The final low-resolution feature map F^D has 12 channels instead of W (and accordingly uses $12W$ filters). These final features correspond to the color samples of a 2×2 neighborhood. We upsample them back to full-resolution, reversing the process of Equation 1. We also concatenate masked copies of the input mosaick M as channels in F^{D+1}. The masks m_c effectively isolate the RGB color samples on three distinct channels.

$$F_c^{D+1}(x,y) = m_c(x,y)\mathrm{M}(x,y) \text{ for } c \in \{1 \dots 3\} \tag{3}$$

$$F_c^{D+1}(x,y) = F_{c'}^D\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor\right) \text{ for } c \in \{4 \dots 6\} \tag{4}$$

Here $c' = 4(c-4)+1+(x \bmod 2)+2(y \bmod 2)$. This implements a form of residual network: we fast-forward the *identity mapping* to deeper layers, thereby allowing the network to learn a residual instead of the absolute mapping. Propagating the identity through many non-linear layers is harder an uses more parameters than with this shorcut [He et al. 2016]. However, we do not force the network to use both the fast-forwarded identity and the non-linear stack in fixed proportions but rather let it learn the appropriate mix: we perform a last convolution (Equation (2)), at full resolution this time, to produce F^{D+2}. The final output O of the network is an affine combination of the last feature maps F^{D+2}.

$$\mathrm{O}_c(x,y) = b_c^{\mathrm{O}} + \sum_{c'} w_{cc'}^{\mathrm{O}} F_{c'}^{D+2}(x,y) \tag{5}$$

Overall we opted for a thin (small W), deep (large D) architecture that is most similar to that of [Simonyan and Zisserman 2014]. We experimented with networks of depth from $D = 5$ up to 20. For each convolutional layer, we used kernels with spatial footprint $K = 3$. The network thus implements a non-linear filter with a receptive field of $2D(K-1)+K+1$ pixels with respect to the input's resolution. We pad the input of each convolution layer by $\frac{K-1}{2}$ pixels on each side so that the spatial dimension does not decrease with depth. The network thus also learns the boundary condition and does not reduce the dimensions of the input image, which would happen if we were to keep only the valid part of the convolutions. While processing the image at full-resolution with the color masks of Equation (4) directly applied to the input mosaick is possible (§ 5), it incurs a higher computational cost since the network then processes four times as many pixels. This also reduces the receptive field of the final layer. We did not find this alternative approach to significantly affect the denoising/demosaicking performances.

3.2 Joint denoising with multiple noise levels

A combination of Poisson and Gaussian noises in *linear* space accurately models camera noise [Foi et al. 2008]. Because we work with white-balanced gamma-corrected sRGB images, we use an additive Gaussian noise model as [Jeon and Dubois 2013] recommends.

We want to alleviate the need for a specialized network for each noise level. Instead, we train a single network on a continuous range of noise levels and explicitly add the noise level as an input parameter to the network. At training time and for each new input M, we randomly sample a noise level $\sigma \in [\sigma_1, \sigma_2]$. We corrupt M with a centered additive Gaussian noise of variance σ^2 before feeding it to the network. We also provide the network with the scalar estimate of the noise level σ as extra input (Figure 2). In practice, since camera model and settings are stored alongside the raw data and one can rely on offline noise calibration, the noise level is typically known and used to inform demosaicking. In order to incorporate this new information into the convolutional architecture, we spatially replicate the noise level to match the input dimensions of the first layer F^0 and concatenate it as an extra channel: F^0 now has 5 channels (Figure 2). [Burger et al. 2012] used a similar approach for denoising-only in a non-convolutional setup.

3.3 Training procedure

At training time, we use a dataset $\mathcal{D} = \{(\sigma_i, \mathrm{M}_i, \mathrm{I}_i)\}_i$ of mosaicked/ground-truth image patches where M_i is generated from I_i and corrupted with additive white Gaussian noise of variance σ_i^2 on-line. We optimize the weights and biases by minimizing the normalized L_2 loss on this training set:

$$\mathcal{L}\left(\left\{w^{(d)}, b^{(d)}\right\}_d\right) = \frac{1}{p^2|\mathcal{D}|}\sum_i ||\mathrm{O}_i - \mathrm{I}_i||^2 \qquad (6)$$

In all our experiments, we use a patch size $p = 128$ pixels for the training samples. The filter weights $w^{(d)}$ are initialized according to [He et al. 2015] and the biases $b^{(d)}$ are first set to 0. The optimization is carried out by ADAM [Kingma and Ba 2014], a flavor of stochastic gradient descent that maintains an adaptive estimate of the first and second order moments of the gradient and uses them to be independent of any diagonal rescaling of the gradient. We use a batch size of 64, and an initial learning rate of 10^{-4}. We used an L_2 weight decay of 10^{-8} on $w^{(d)}$. All other parameters are left at the value recommended by the authors. As learning progresses, we decrease the learning rate by a factor 10 whenever the validation error on an independent dataset stalls, typically twice, after 10 epochs. In our experience, higher initial learning rates fail to converge to a good solution. The training is performed with a customized version of CAFFE [Jia et al. 2014] on a NVIDIA Titan X, and usually takes 2-3 weeks.

4 Training Data

When trained on standard datasets, our neural network works well on average but produces disturbing artifacts on a number of hard cases, the common plague of demosaicking and denoising. These challenges are due to two important issues. First, hard cases are rare and get diluted by the vastly more common easy areas. Second, metrics such as L^2 or PSNR fail to notice demosaicking artifacts that are salient to humans.

We now present an algorithm for detecting challenging patches and focus the training on them using a combination of adaptive training based on human visual difference predictors and a new metric optimized to detect moiré artifacts.

We first train a network on standard datasets and use it to demosaick and denoise millions of ground-truth photographs in order to mine for hard cases. We look for two classes of artifacts frequently missed by the network: luminance artifacts and color moiré. Inspired by curriculum learning [Bengio et al. 2009], we adaptively build a new dataset composed of these artifact-prone patches. We use this dataset to fine-tune or train the network from sratch. This improves the model's performance on difficult cases and can be seen as reweighting the loss function to give more weight to artifact-prone patches. Below, we discuss our selection strategy and the metrics we use.

4.1 Ground-truth and mosaicked image

We start with a large number of sRGB images downloaded from the web to generate ground truth data. We create a mosaick from each image, add noise, and use this pair for training. We restrict our selection to images with at least 16 Mpix to favor higher quality images. To avoid biasing the network towards distortions caused by the camera pipeline that first created the downloaded images, we downsample them by a factor 4 using bicubic interpolation and use this as ground-truth. While more complex downsampling techniques are possible [Khashabi et al. 2014], they do not help in our context: our training images are JPEG-compressed and come from unknown and diverse sources.

We create the mosaicked and noisy image M by retaining only one color channel per pixel according to the Bayer pattern from the sRGB image. We also augment patches from the training set with random rotations in 90°steps, random left-right mirror images, and 1-pixel shifted copies in either dimension. This augments the training data by a factor $32\times$ and provides some rotational and translational invariance.

4.2 Challenging patches are rare

Publicly available demosaicking datasets contain a few hundred images which is insufficient for training the thousands of parameters of a deep network. Instead, we trained our first network using 1.3 million images from Imagenet [Deng et al. 2009] and 1 million images from MirFlickr [Huiskes and Lew 2008]. While this network matches the PSNR statistics of previous work, a closer inspection reveals artifacts near thin edges and complex textures (see Figure 3). Large quantity of training samples do not guarantee convincing demosaicking.

A random selection of images is mainly composed of smooth patches as these dominate natural images [Levin et al. 2012]. Challenging structures only make up a small fraction, shown as the tail of the patch distribution in Figure 4. Smooth patches account for a majority of the training time, even though results on such cases are already perceptually indistinguishable from ground truth. We compensate for this by assembling a training set with more difficult patches.

4.3 Mining hard patches

We create a database of difficult patches by applying the first network (trained on Imagenet) to millions of new patches we download from the web and retaining the failure cases. We detect patches that pose two specific challenges: luminance artifacts around thin structures (e.g. *zipper*) and color moiré. We use separate metrics to detect these cases. Rejecting trivial cases effectively reweighs the loss function (Eq. (6)) towards challenging ones.

Salient luminance artifacts We first use the perceptually based HDR-VDP2 [Mantiuk et al. 2011] to detect luminance artifacts

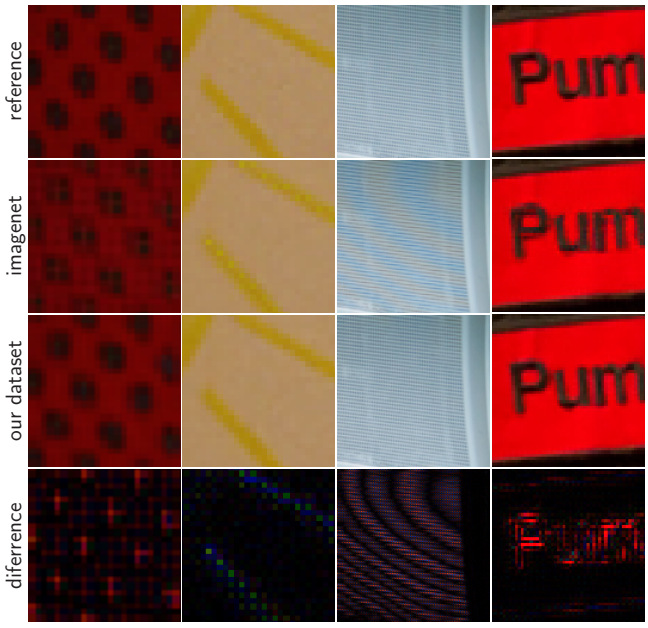

Figure 3: *A network trained on a standard image dataset (second row) creates noticeable artifacts in its output such as the zippering on the thin yellow line, confusion around curves in the first and last example, and moiré in the third example. When training the same network on our new dataset of difficult cases, these artifacts are mostly gone (third row). The last row shows the difference map between the two network outputs, and the first row is the ground-truth image. (best viewed in digital form)*

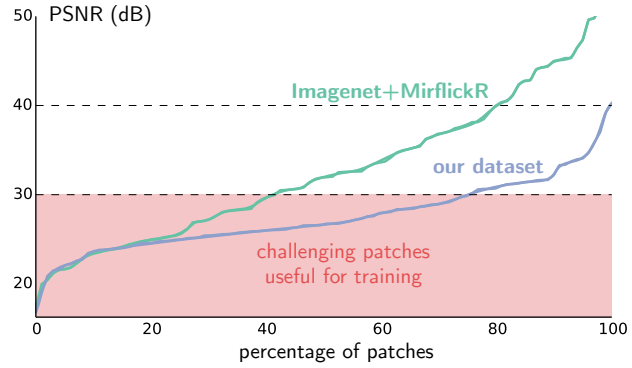

Figure 4: *Most of the patches in a generic training dataset (here Imagenet and Mirflickr) are easy cases for modern demosaicking algorithms (in this figure, we measure the PSNR of AHD [Hirakawa and Parks 2005]). For a network to perform well in challenging situations, it needs to be trained on challenging patches, i.e., on data that lies on the tail of the patch distribution. This plot shows that our dataset contains more such patches. Further, not shown in this figure is the fact that demosaicking failures on our patches lead to more visually unpleasant artifacts: we explicitly selected the patches for this reason.*

around thin edges. We have found that standard metrics like PSNR, SSIM, S-CIELAB do not capture perceptual artifacts as convincingly as HDR-VDP2. It has also been empirically shown to correlate well with human judgement for simpler demosaicking [Sergej and Mantiuk 2014]. It compares the visibility of local artifacts as well as overall image quality to a reference. It models the response of

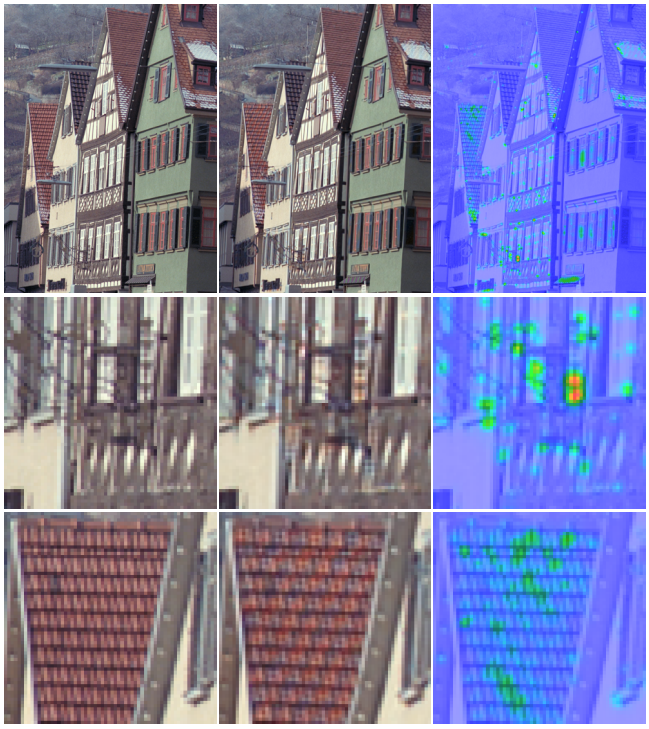

Figure 5: *HDR-VDP run on the demosaicked output of LDI-NAT [Zhang et al. 2011]. First, row full image. Second row, HDR-VDP correctly detects the zipper pattern due to luminance variations. It signals some anomalies in the output image but with a low probability of detection. It works only on luminance, therefore misses the chrominance moiré artifacts.*

the human visual system including phenomena such as spectral sensitivity, luminance adaptation and frequency masking and is calibrated against contrast sensitivity measurements. For each new image, we apply demosaicking using the pre-trained network. We then compare the network's output to the ground truth using HDR-VDP and compute a probability of artifacts at each pixel. We smooth the probability map using Gaussian blur ($\sigma = 3$) and extract up to 30 local maxima if the artifact probability exceeds 0.1. This resulted in 2,489,180 problematic patches from 1,393,107 15.2 Mpix images ($\sim 3\%$ of total pixels). We adjusted the metric to approximate the response of a human viewing a 2560×1600 30-inch sRGB display from a distance of 1m. HDR-VDP detects high-frequency luminance artifacts (Fig. 5a); training our network on these patches yielded drastically improved results. The metric however misses color moiré artifacts because it only analyzes the luminance channel (Fig. 5, bottom row).

Moiré and aliasing Moiré is an interference pattern caused by aliasing. Repetitive details close to or smaller than the resolution of the sampling grid can give rise to artificial low frequency patterns. Mosaic images have their color channels at spatial offsets; moiré appears as distracting false color bands (Fig.6b) after demosaicking because of erroneous interpolation of color samples. The effect of aliasing is best understood in the Fourier domain because it introduces undesirable frequencies. We quantify moiré artifacts by measuring the change in frequency content from the ground-truth I to the demosaicked image O image. We first convert both I and O to the *Lab* space and compute the 2D Fourier transform of each channel $\mathcal{F}_{\mathrm{I}}(\omega)$ and $\mathcal{F}_{\mathrm{O}}(\omega)$ respectively. We then compute the gain

(a) *input* **(b)** *demosaicked output* **(c)** *error* **(d)** *amplitude gain in the Fourier domain*

Figure 6: *Frequency Gain due to moire*

of the demosaicked image with respect to the input at each frequency.

$$\rho(\omega) = \begin{cases} \log\left(\frac{|\mathcal{F}_0(\omega)|^2+\eta}{|\mathcal{F}_1(\omega)|^2+\eta}\right) & \text{if } |\omega| \leq r \\ 1 & \text{otherwise} \end{cases} \quad (7)$$

We only compare the gain in frequencies lower than r to mitigate boundary effects and high-frequency noise. We smooth the gain map with Gaussian blur and mark the patch as aliased if the maximum gain value across all channels and frequencies exceeds a threshold t. For 128×128 patches, we set the the low-pass radius $r = 0.95\pi$, the standard deviation of the Gaussian kernel to 3, and gain map threshold to $t = 2$. This criterion consistently selects moiré-prone patches. Figure 6 shows the gain map for an aliased patch.

These moiré patches are rare; they lie at the end of the tail of natural patch distribution (Figure 4). We found 0.05% of patches from 2 million images patches are aliased. Nonetheless, these artifacts are important because they can still affect large areas of an image (e.g. a 128×128 patch), making it unusable.

5 Results

We evaluate our network in various conditions. Unless stated otherwise, all the experiments in this section use a network with $D = 15$ layers (each with $W = 64$ 3×3 filters) trained from scratch on $2{,}590{,}186$ 128×128 hard patches. The network has $559{,}776$ trainable parameters. We stop the training when the error on a separate validation set of 4000 images stops decreasing. We test all techniques on another dataset of 2000 images. All three datasets are independent and have been mined in the same fashion, as described in Section 4. Half of the test set was mined using the HDR-VDP metric (we refer to this half as the *vdp* test set). The other half was assembled using the moiré metric (we refer to it as *moiré*). The parameters of competing techniques are set to the values recommended by their authors, often tuned on the Kodak/McMaster datasets included in our comparison. Our main metric is PSNR where the error is averaged over pixels *and* color channels before taking the logarithm.

First, we compare our algorithm against previous work on the demosaicking-only task with noise-free sRGB images (Table 1, in particular no denoising is applied). This evaluation illustrates that high PSNR statistics can obscure subtle perceptual artifacts: we demonstrate this on hard cases from our testing dataset (Figure 9). We then present our results on demosasicking noisy inputs, which we refer to as joint denoising and demosasicking (Figure 7). Although our network is trained on 8-bits sRGB data, we also evaluate our network on linear RGB data (Table 2) and non-Bayer mosaicks. This shows that our approach generalizes to other demosaicking conditions. We finally describe implementation details and show that our algorithm is faster than the previous best-performing methods on both CPU and GPU.

Demosaicking noise-free images We first evaluate our algorithm on noise-free inputs from two common demosaicking datasets: McMaster [Zhang et al. 2011] and Kodak [Li et al. 2008]. Table 1 (first two columns) show that our network outperforms the previous techniques on these datasets. These results alone however are not sufficient because these datasets are known to have flaws and to misrepresent the statistics of digital images [Levin et al. 2012]. To provide a more accurate depiction of the demosaicking challenges, we also compare our technique with the state of the art on a testing set of 2000 hard cases not seen during training (Table 1 third and fourth columns). Our method produces consistently better results quantitatively and the improvement is also visually significant (Figure 9). Our network (trained on difficult cases) successfully handles complex patterns and generates artifact-free results. We also compare to the widely used Adobe Camera Raw software. Results for all the datasets and techniques can be found in the supplemental material. Since the test images are noise-free, no denoising has been applied in this experiment.

	kodak	mcm	vdp	moiré
bilinear	32.9	32.5	25.2	27.6
Adobe Camera Raw 9	33.9	32.2	27.8	29.8
Klatzer* [2016]	35.3	30.8	28.0	30.3
Gunturk [2002]	35.8	33.2	29.3	31.3
Lu [2010]	36.0	33.4	29.4	31.4
Li [2005]	36.1	33.1	29.2	31.5
Hirakawa [2005]	36.1	33.8	28.6	30.8
Condat [2011]	35.5	33.3	28.4	30.9
Condat [2012]	36.1	33.6	29.6	31.9
Jeon [2013]	36.4	34.0	27.8	30.4
Hirakawa [2006]	36.5	33.9	30.0	32.1
Hamilton [1997]	36.9	35.2	28.9	30.9
Zhang [2005]	37.3	34.7	30.3	32.4
Buades [2009]	37.3	35.5	29.7	31.7
Zhang (NLM) [2011]	37.9	36.3	30.1	31.9
Getreuer [2011]	38.1	36.1	30.8	32.5
Heide [2014]	40.0	38.6	27.1	34.9
ours	**41.2**	**39.5**	**34.3**	**37.0**

Table 1: PSNR *comparison of our approach to state of the art techniques on the demosaicking-only scenario. First and second column show evaluation on standard datasets. Third and fourth column show comparisons on our datasets containing images prone to luminance artifacts and color moiré respectively. No denoising is applied for any of the competing methods. (*) For Klatzer et al., we used the published model which was trained on linear data and ran it on* linearized *images. The training code was not available at the time of publication.*

Training set and training time We initially trained our network on the 1.3 million images from Imagenet [Deng et al. 2009] and 1 million from MirFlickr [Huiskes and Lew 2008] instead of our dataset of difficult cases. Despite reaching competitive PSNR levels (on-par with FlexISP [Heide et al. 2014]), the network produced noticeable artifacts, mainly along thin structures and moiré-prone textures. We believe that this is due to the inherent bias of these standard datasets towards trivial cases like smooth patches or unambiguous edges. Training a network on the difficult cases significantly improves visual quality (Figure 3). We found that fine-tuning the *Imagenet+MirFlickr* network or retraining from scratch worked equally well. All the results we report are trained from scratch on the hard examples only. Accuracy is numerically competitive after a day of training but image quality improves with longer training.

Week-long training is common with deep networks and has no impact on the practicality of our approach since it is done only once before the algorithm is deployed.

Joint denoising and demosaicking results We now present results for joint denoising and demosaicking (Figure 10). We train on images corrupted with continuous levels of noise $\sigma \in [0; 20]$. Similar to previous work, we model noise in the white-balanced gamma-corrected images as signal-independent white Gaussian noise [Jeon and Dubois 2013]. During evaluation, we tested images at 6 levels of noise within the range used for training (Fig. 7). Our results consistently outperform previous techniques on all noise levels.

We also experimented with networks trained on a single noise level instead of continuous levels and did not observe noteworthy change in result quality (Fig. 7). This suggests that the network is already optimally trained, and does not require fine-tuning for each noise level.

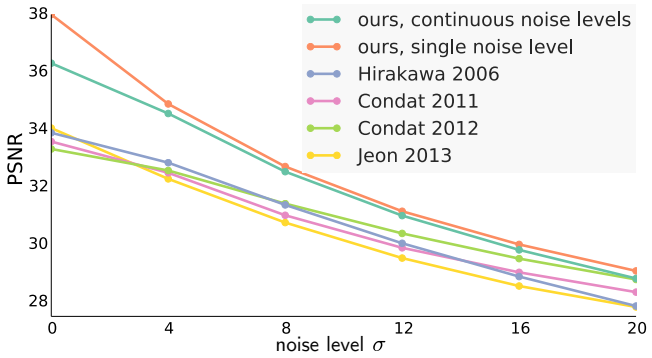

Figure 7: PSNR *comparison joint denoising and demosaicking at different levels of Gaussian noise with standard deviation σ. The metric is averaged across all four datasets from Table 1: mcm, kodak, vdp, moiré.*

Processing linear data Khashabi et al. [2014] suggest that demosaicking should be evaluated on raw RGB data with an affine noise model [Foi et al. 2008; Hasinoff et al. 2010]. In previous experiments, we instead trained and evaluated on sRGB to facilitate comparisons with state-of-the-art techniques that choose to do the same. Without any further training on linear data or affine noise models, our sRGB trained network outperforms the best techniques on the MSR 16-bits linear Panasonic testing set [Khashabi et al. 2014] (Table 2). Since noise parameters for individual images are not provided in this dataset, we estimate the average noise variance and use it as our noise parameter. We also fine-tuned our network on our dataset of hard cases *linearized* from sRGB and observed virtually the same performance. This shows that our network is not restricted to sRGB data and generalizes well to linear data. Real RAW training data would be ideal, but available datasets do not contain enough challenging cases: we observed no quality improvement when training on MIT5k [Bychkovsky et al. 2011] or the MSR training set. Figure 8 shows the output of our algorithm on real (linear) RAW images captured by a Canon 5D mark II at various ISO levels.

Alternative mosaick patterns With a few simple modifications, our method generalizes to non-Bayer patterns. We experimented with the Fuji X-Trans pattern. Compared to the Bayer network illustrated in Figure 2, we no longer process the image at quarter resolution. Instead, the mosaicked input RGB values are kept at full-resolution on separate planes: we remove layers F^0 and F^{D+1}. The X-Trans pattern is 6x6 pixels; this would imply a much more aggressive downsampling. At train time, we apply a color mask to the ground truth that converts it to the non-Bayer mosaick. We trained this modified network from scratch for three days on our dataset of hard-cases. We evaluate this new network on the MSR Panasonic X-Trans dataset [Khashabi et al. 2014]. The table below shows that our algorithm consistently performs better than previous techniques.

	linear	sRGB
Khashabi [2014]	36.9	30.6
Klatzer [2016]	39.6	33.1
ours	**39.7**	**33.2**

	noise-free		with noise	
	linear	sRGB	linear	sRGB
bilinear	30.9	24.9	–	–
Hamilton [1997]	36.7	30.0	–	–
Hirakawa [2005]	37.2	31.3	–	–
Zhang (NAT) [2011]	37.6	31.6	–	–
Gunturk [2002]	38.2	31.0	–	–
Lu [2010]	38.3	31.0	–	–
Zhang (NLM) [2011]	38.4	32.1	–	–
Zhang [2005]	38.8	31.7	–	–
Getreuer [2011]	39.4	32.9	–	–
Khashabi [2014]	39.4	32.6	37.8	31.5
Heide* [2014]	40.0	33.8	–	–
Klatzer [2016]	40.9	34.6	**38.8**	**32.6**
ours	41.6	35.3	38.4	32.5
ours (f.t.)	**42.7**	**35.9**	38.6	**32.6**

Table 2: *Evaluation on linear data for both noise-free and noisy data. We report* PSNR *in both linear and sRGB space. We feed a single estimate of the average noise level to our network a test time. We also fine tuned our network to linearized sRGB. Among all competing techniques, only Khashabi[2014] and Klatzer [2016] techniques were specifically designed for linear data. Results on noisy images are excluded from the table for methods that do not attempt to denoise.*

Variations on the network configuration Using as few as $D = 7$ layers has a minimal impact on general accuracy, but patches prone to moiré are significantly degraded: they benefit from the large spatial footprint of the deeper network. $W = 64$ filters per layer worked well, using $W = 128$ was superfluous and $W = 32$ decreased the PSNR.

Running time Unless stated otherwise, we benchmark all methods with 1MPix images on an Intel Core i7-3770K and GeForce Titan 700 and report the average time over 10 runs. Our technique is linear in the pixel count. Superlinear competitors can be made to run in linear-time by processing the image in tiles. We use the Halide image processing language [Ragan-Kelley et al. 2013] to implement our network. Our CPU implementation of a $D = 15$ layers network is $8\times$ faster than ATLAS Caffe [Jia et al. 2014] and $3.5\times$ faster than Caffe with Intel's Math Kernel Library. It processes image at 3s/Mpix on a modern desktop CPU. Our approach is up to two orders of magnitude faster than previous high-quality techniques that use global optimization like FlexISP [Heide et al. 2014] and other non-local techniques [Zhang et al. 2011] (Table 3).

Figure 8: *Our algorithm trained on sRGB data generalizes to real (linear) RAW images. It successfully removes color moiré on the fabric at various noise levels whereas dcraw does not (first and second rows). In comparison, Klatzer et al. [2016] does not generalize well to widely different noise levels: it denoises too aggressively the ISO100 image (first row), and produces artifacts on the ISO6400 image (second row). This is particularly visible on the smooth wall. Our output is free of checkerboard patterns and staircase artifacts (third row). DCRaw exhibits these artifacts on the red lettering and Klatzer et al. has checkboard on the blue line on the right.*

	CPU (ms/Mpix)	GPU (ms/Mpix)
bilinear	127	–
Hamilton [1997]	385	–
Condat [2011]	566	–
Lu [2010]	737	–
Li [2005]	1117	–
Hirakawa [2006]	1618	–
Gunturk [2002]	1991	–
Hirakawa [2005]	2998	–
Condat [2012]	11,211	–
Jeon [2013]	14,728	–
Zhang [2005]	30,642	–
Khashabi [2014]	36,157*	–
Zhang (NLM) [2011]	264,243	–
Zhang (NAT) [2011]	1,700,510	–
Heide [2014]	1,815,111	3000*
Klatzer [2016]	3,560,510	1600*
ours	**2,932**	**325**

Table 3: *Runtime of different demosaicking algorithms in their publicly available implementations. Our approach is faster than previous high quality techniques like FlexISP [Heide et al. 2014]. Timings with an asterisk (*) are reported from the respective original paper.*

Limitations Our approach relies on image metrics to detect challenging patches and build a ground-truth dataset. We used HDR-VDP for luminance artifacts, but it is not perfect and we can benefit from a better metric. Also, if the sRGB ground truth is corrupted with color moiré, our network will learn the corruption; a no-reference moiré detector is required to alleviate this.

6 Conclusion

We demonstrated that a joint approach based on a deep neural network can significantly improve the quality of demosaicking and denoising. It can resolve even challenging situations that usually result in zippering or moiré artifacts. However, traditional supervised learning must be adapted because the vast majority of image regions are easy to address and the real hard cases do not occur enough and and are not well characterized by even advanced perceptual image metrics. We proposed an adaptive approach as well as a new moiré detection metric to tackle these challenges. Our method outperforms state-of-the-art solutions in terms of both perceptual and statistical visual quality, while being an order of magnitude faster.

Acknowledgements

We thank the SIGGRAPH reviewers for their constructive comments. We gratefully acknowledge NVIDIA for the generous donation of a Tesla K40 GPU. Thanks to Eric Chan for his invaluable expertise and precious feedback. Thanks to Sebastian Nowozin, Felix Heide and Jan Kautz for help with the comparison. Thanks to Tiam Jaorensri for help with the hardware. This work was partially funded by a gift from Adobe.

References

AKIYAMA, H., TANAKA, M., AND OKUTOMI, M. 2015. Pseudo four-channel image denoising for noisy cfa raw data. In *Image Processing (ICIP), 2015 IEEE International Conference on*, 4778–4782.

BADRINARAYANAN, V., HANDA, A., AND CIPOLLA, R. 2015. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*.

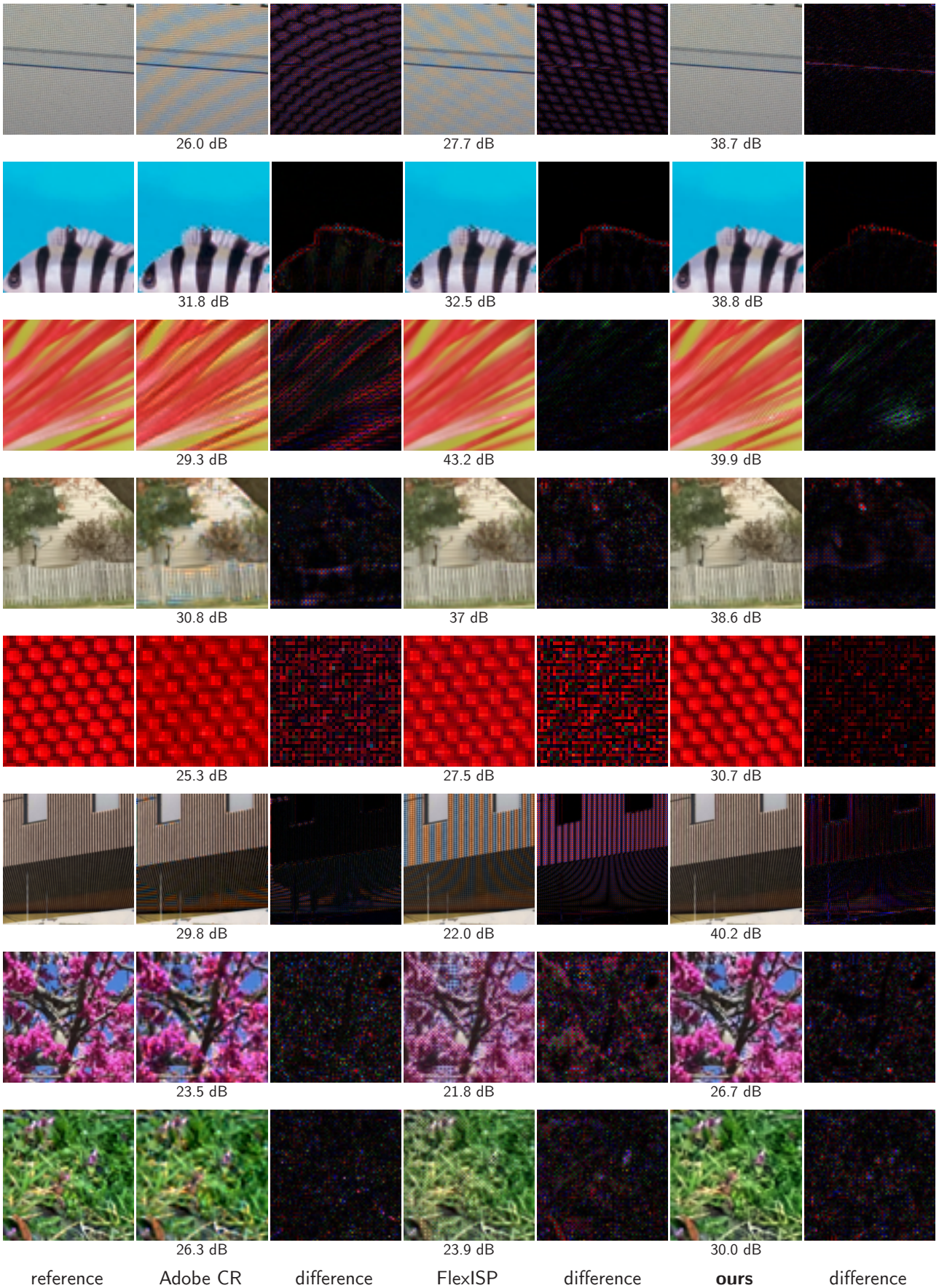

Figure 9: *Comparison of our approach with Adobe Camera Raw, FlexISP [Heide et al. 2014] on noise-free images. Exhaustive results can be found in supplementary material.*

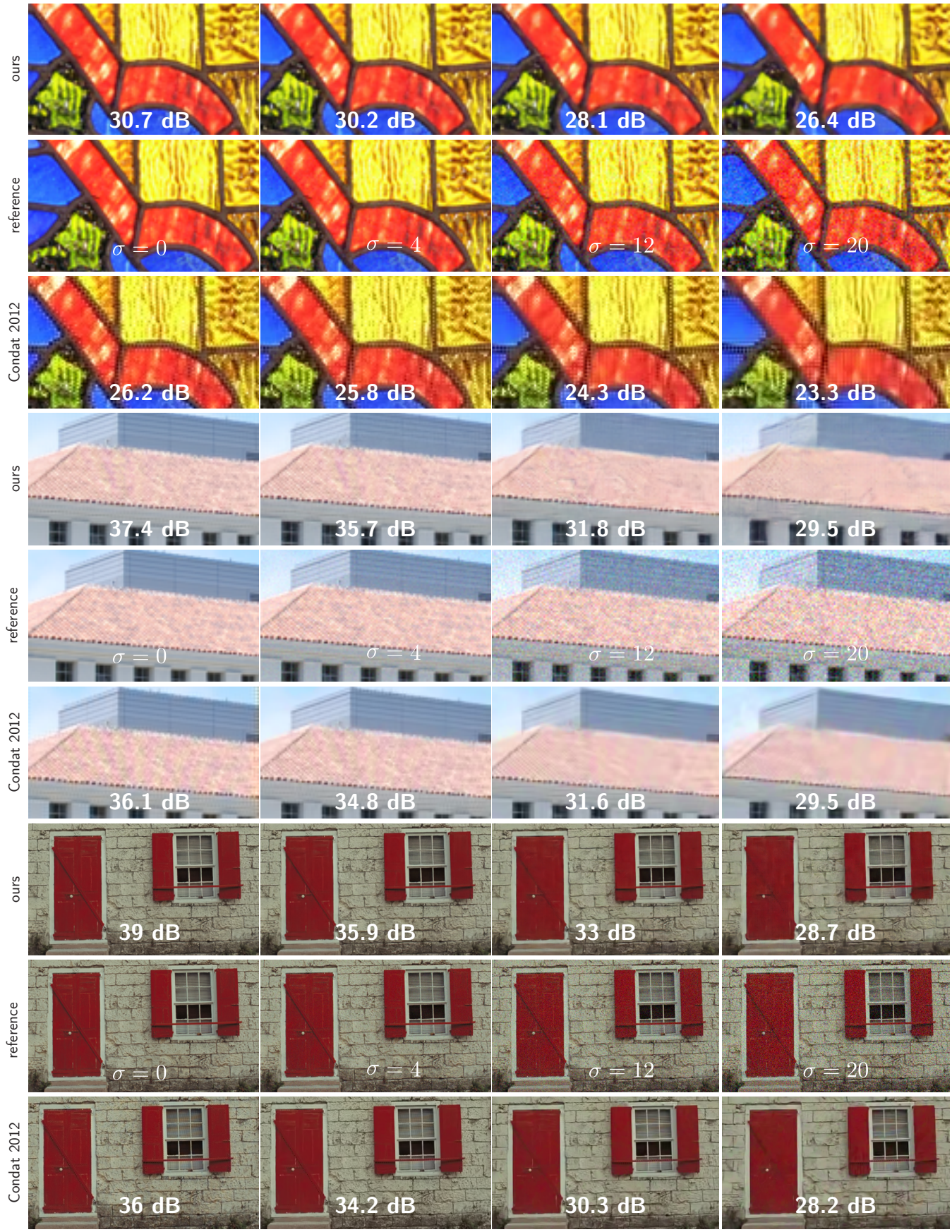

Figure 10: *Joint denoising and demosaicking results. Our approach outperforms previous best techniques on noisy data in challenging images on different levels of Gaussian noise with standard deviation σ. Exhaustive results can be found in supplementary material.*

BENGIO, Y., LOURADOUR, J., COLLOBERT, R., AND WESTON, J. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, New York, NY, USA, ICML '09, 41–48.

BUADES, A., COLL, B., MOREL, J.-M., AND SBERT, C. 2009. Self-similarity driven color demosaicking. *Image Processing, IEEE Transactions on 18*, 6, 1192–1202.

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

BYCHKOVSKY, V., PARIS, S., CHAN, E., AND DURAND, F. 2011. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*.

CHANG, L., AND TAN, Y.-P. 2004. Effective use of spatial and spectral correlations for color filter array demosaicking. *Consumer Electronics, IEEE Transactions on 50*, 1, 355–365.

CHENG, Z., YANG, Q., AND SHENG, B. 2015. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, 415–423.

COK, D. R., 1987. Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal, Feb. 10. US Patent 4,642,678.

CONDAT, L., AND MOSADDEGH, S. 2012. Joint demosaicking and denoising by total variation minimization. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, IEEE, 2781–2784.

CONDAT, L. 2011. A new color filter array with optimal properties for noiseless and noisy color image acquisition. *Image Processing, IEEE Transactions on 20*, 8, 2200–2210.

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.

DOSOVITSKIY, A., FISCHER, P., ILG, E., HAUSSER, P., HAZIRBAS, C., GOLKOV, V., V.D. SMAGT, P., CREMERS, D., AND BROX, T. 2015. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*.

DOSOVITSKIY, A., TOBIAS SPRINGENBERG, J., AND BROX, T. 2015. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1538–1546.

EIGEN, D., KRISHNAN, D., AND FERGUS, R. 2013. Restoring an image taken through a window covered with dirt or rain. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, 633–640.

EIGEN, D., PUHRSCH, C., AND FERGUS, R. 2014. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2366–2374.

FLYNN, J., NEULANDER, I., PHILBIN, J., AND SNAVELY, N. 2016. Deepstereo: Learning to predict new views from the world's imagery. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

FOI, A., TRIMECHE, M., KATKOVNIK, V., AND EGIAZARIAN, K. 2008. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *Image Processing, IEEE Transactions on 17*, 10, 1737–1754.

GATYS, L. A., ECKER, A. S., AND BETHGE, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2414–2423.

GETREUER, P. 2011. Color demosaicing with contour stencils. In *Digital Signal Processing (DSP), 2011 17th International Conference on*, IEEE, 1–6.

GO, J., SOHN, K., AND LEE, C. 2000. Interpolation using neural networks for digital still cameras. *Consumer Electronics, IEEE Transactions on 46*, 3, 610–616.

GUNTURK, B. K., ALTUNBASAK, Y., AND MERSEREAU, R. M. 2002. Color plane interpolation using alternating projections. *Image Processing, IEEE Transactions on 11*, 9, 997–1013.

HAMILTON JR, J. F., AND ADAMS JR, J. E., 1997. Apparatus for utilizing a digitized image signal, May 13. US Patent 5,629,734.

HASINOFF, S. W., DURAND, F., AND FREEMAN, W. T. 2010. Noise-optimal capture for high dynamic range photography. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 553–560.

HE, F.-L., WANG, Y.-C. F., AND HUA, K.-L. 2012. Self-learning approach to color demosaicking via support vector regression. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, IEEE, 2765–2768.

HE, K., ZHANG, X., REN, S., AND SUN, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*.

HE, K., ZHANG, X., REN, S., AND SUN, J. 2016. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

HEIDE, F., STEINBERGER, M., TSAI, Y.-T., ROUF, M., PAJKAK, D., REDDY, D., GALLO, O., LIU, J., HEIDRICH, W., EGIAZARIAN, K., ET AL. 2014. Flexisp: a flexible camera image processing framework. *ACM Transactions on Graphics (TOG) 33*, 6, 231.

HIRAKAWA, K., AND PARKS, T. W. 2005. Adaptive homogeneity-directed demosaicing algorithm. *Image Processing, IEEE Transactions on 14*, 3, 360–369.

HIRAKAWA, K., AND PARKS, T. W. 2006. Joint demosaicing and denoising. *Image Processing, IEEE Transactions on 15*, 8, 2146–2157.

HUISKES, M. J., AND LEW, M. S. 2008. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, ACM, 39–43.

IIZUKA, S., SIMO-SERRA, E., AND ISHIKAWA, H. 2016. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016) 35*, 4.

JEON, G., AND DUBOIS, E. 2013. Demosaicking of noisy bayer-sampled color images with least-squares luma-chroma demultiplexing and noise level estimation. *Image Processing, IEEE Transactions on 22*, 1, 146–156.

JIA, Y., SHELHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J., GIRSHICK, R., GUADARRAMA, S., AND DARRELL, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, ACM, New York, NY, USA, MM '14, 675–678.

KAPAH, O., AND HEL-OR, H. Z. 2000. Demosaicking using artificial neural networks. In *Electronic Imaging*, International Society for Optics and Photonics, 112–120.

KHASHABI, D., NOWOZIN, S., JANCSARY, J., AND FITZGIBBON, A. W. 2014. Joint demosaicing and denoising via learned non-parametric random fields. *Image Processing, IEEE Transactions on 23*, 12, 4968–4981.

KINGMA, D., AND BA, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

KLATZER, T., HAMMERNIK, K., KNOBELREITER, P., AND POCK, T. 2016. Learning joint demosaicing and denoising based on sequential energy minimization. In *2016 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 1–11.

KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

KWAN, C., AND XIAOLIN, W. 2004. A classification approach to color demosaicking. In *International Conference on Image Processing*.

LANSEL, S., AND WANDELL, B. 2011. Local linear learned image processing pipeline. In *Imaging Systems and Applications*, Optical Society of America, IMC3.

LAROCHE, C. A., AND PRESCOTT, M. A., 1994. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients, Dec. 13. US Patent 5,373,322.

LARSSON, G., MAIRE, M., AND SHAKHNAROVICH, G. 2016. Learning representations for automatic colorization. In *European Conference on Computer Vision (ECCV)*.

LECUN, Y., BENGIO, Y., AND HINTON, G. 2015. Deep learning. *Nature 521*, 7553, 436–444.

LEVIN, A., NADLER, B., DURAND, F., AND FREEMAN, W. T. 2012. Patch complexity, finite pixel correlations and optimal denoising. In *Computer Vision–ECCV 2012*. Springer, 73–86.

LI, X., GUNTURK, B., AND ZHANG, L. 2008. Image demosaicing: A systematic survey. In *Electronic Imaging 2008*, International Society for Optics and Photonics, 68221J–68221J.

LI, X. 2005. Demosaicing by successive approximation. *Image Processing, IEEE Transactions on 14*, 3, 370–379.

LONG, J., SHELHAMER, E., AND DARRELL, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

LU, Y. M., KARZAND, M., AND VETTERLI, M. 2010. Demosaicking by alternating projections: theory and fast one-step implementation. *Image Processing, IEEE Transactions on 19*, 8, 2085–2098.

MANTIUK, R., KIM, K. J., REMPEL, A. G., AND HEIDRICH, W. 2011. HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. In *ACM Transactions on Graphics (TOG)*, vol. 30, ACM, 40.

NOH, H., HONG, S., AND HAN, B. 2015. Learning deconvolution network for semantic segmentation. *CoRR abs/1505.04366*.

PARK, S. H., KIM, H. S., LANSEL, S., PARMAR, M., AND WANDELL, B. A. 2009. A case for denoising before demosaicking color filter array data. In *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, IEEE, 860–864.

PATHAK, D., KRÄHENBÜHL, P., DONAHUE, J., DARRELL, T., AND EFROS, A. 2016. Context encoders: Feature learning by inpainting.

RAGAN-KELLEY, J., BARNES, C., ADAMS, A., PARIS, S., DURAND, F., AND AMARASINGHE, S. 2013. Halide: A language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. *SIGPLAN Not. 48*, 6 (June), 519–530.

SERGEJ, T., AND MANTIUK, R. 2014. Perceptual evaluation of demosaicing artefacts. In *Image Analysis and Recognition*. Springer, 38–45.

SIMONYAN, K., AND ZISSERMAN, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

TIAN, Q., LANSEL, S., FARRELL, J. E., AND WANDELL, B. A. 2014. Automating the design of image processing pipelines for novel color filter arrays: Local, linear, learned (l3) method. In *IS&T/SPIE Electronic Imaging*, International Society for Optics and Photonics, 90230K–90230K.

WANG, X., FOUHEY, D., AND GUPTA, A. 2015. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 539–547.

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.

XU, L., REN, J., YAN, Q., LIAO, R., AND JIA, J. 2015. Deep edge-aware filters. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 1669–1678.

ZHANG, L., AND WU, X. 2005. Color demosaicking via directional linear minimum mean square-error estimation. *Image Processing, IEEE Transactions on 14*, 12, 2167–2178.

ZHANG, F., WU, X., YANG, X., ZHANG, W., AND ZHANG, L. 2009. Robust color demosaicking with adaptation to varying spectral correlations. *Image Processing, IEEE Transactions on 18*, 12, 2706–2717.

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, R., ISOLA, P., AND EFROS, A. A. 2016. Colorful image colorization. *ECCV*.