# Blind Universal Bayesian Image Denoising with Gaussian Noise Level Learning

Majed El Helou Sabine Süsstrunk
EPFL, Switzerland
{majed.elhelou, sabine.susstrunk}@epfl.ch

## Abstract

*Blind and universal image denoising consists of a unique model that denoises images with any level of noise. It is especially practical as noise levels do not need to be known when the model is developed or at test time. We propose a theoretically-grounded blind and universal deep learning image denoiser for Gaussian noise. Our network is based on an optimal denoising solution, which we call fusion denoising. It is derived theoretically with a Gaussian image prior assumption. Synthetic experiments show our network's generalization strength to unseen noise levels. We also adapt the fusion denoising network architecture for real image denoising. Our approach improves real-world grayscale image denoising PSNR results by up to* $0.7dB$ *for training noise levels and by up to* $2.82dB$ *on noise levels not seen during training. It also improves state-of-the-art color image denoising performance on every single noise level, by an average of* $0.1dB$*, whether trained on or not.*

## 1. Introduction

Image denoising is a fundamental image restoration task applied in any image processing pipeline. An image denoiser can also be part of deep network models to improve the training of high-level vision tasks [22]. However, being an ill-posed inverse problem, denoising is challenging [12].

After the development of the best analytical solution, BM3D [6], little improvement in denoising performance was achieved until the advent of deep learning denoisers [49]. Recent Convolutional Neural Network (CNN) based methods achieve state-of-the-art image denoising performance and are even faster than traditional optimization-based approaches [45]. Well-designed CNN architectures can also outperform adversarial training methods in image restoration tasks [38].

Neural networks can be deep and wide and thus have large capacity to model complex functions [46, 51] by leveraging network regularization or normalization [16] and residual learning [15]. However, the complex functions modeled by the networks are not interpretable and have little connection to stochastic denoising. This is a limitation for training general models for denoising different noise levels. Denoisers are *blind* when they require no information about the noise level at test time, and *universal* when a single model can handle all noise levels. Blind universal models are important since knowing the noise level, at test time or ahead of training, is not a practical scenario.

We mathematically derive a blind and universal denoising function under the theoretical assumption that the image prior is Gaussian. Our denoising function, which is optimal in stochastic expectation, is referred to as fusion denoising because it fuses the input with a prior weighted using the signal-to-noise ratio. Experimental results show that the state-of-the-art denoiser DnCNN [49] can model an optimal fusion denoising function. However, it only models it for noise levels that are seen by the network during training. For unseen levels, our synthetic experiment's fusion network, called *Fusion Net*, far outperforms DnCNN. We show on synthetic data our improved generalization results.

The assumption that the image prior is Gaussian does not necessarily apply to real-world images. Building on the foundations of our theoretical solution, we adapt our *Fusion Net* by *learning* a fusion function. We call this network *Blind Universal Image Fusion Denoiser (BUIFD)*. BUIFD improves state-of-the-art denoising performance on noise levels seen in training for grayscale and color images on the standard Berkeley test sets (BSD68 and CBSD68) [34]. Furthermore, we show that the generalization results to unseen noise levels obtained in our synthetic experiment extend to the denoising of the grayscale BSD68 test set. Indeed, the denoising performance on noise levels not trained on improves by up to $2.82dB$ in terms of PSNR.

Our main contributions are: (1) we theoretically derive an optimal fusion denoising function and integrate it into a deep learning architecture (Fusion Net), (2) we show on synthetic data that the fusion improves the network's generalization power, and (3) we develop a blind universal image fusion denoiser (BUIFD) adapted to real-world images, and show that it outperforms the state of the art on the standard Berkeley denoising test sets[1].

[1] Our code is available at: https://github.com/IVRL/BUIFD

## 2. Related Work

The image denoising approaches in the literature can be divided into classical methods and the more recent deep-learning-based methods. One common aspect is, however, the leveraging of image priors to improve denoising. For practical reasons, it is important for a denoiser to be blind and universal since the noise levels in noisy images might not be constant or known.

**Image priors.** Whether they are in the form of assumptions made on image gradients [18, 28, 35, 41], sparsity [13, 8], self-similarity within images [9, 3, 43], hybrid approaches [24], or neural network weights given a certain architecture [49, 2], image priors are essential for denoising. Even traditional methods based on diffusion or filtering (in space [30] or in other domains [37]) rely on some priors. They, in all their forms and for multiple image restoration problems, can be discovered and tested heuristically [18, 11], learned with dictionaries [13], with Markov random fields [34], or with deep neural networks [49]. In our network, the prior takes the explicit form of learned feature representations.

**Noise modeling.** Additive white Gaussian noise is not necessarily the best model in practical scenarios [31, 2], such as denoising raw images [2]. Nevertheless, a large part of the image denoising literature focuses on Gaussian denoising since it remains a fundamental problem. Images with noise following different, potentially data-dependent, distributions can be transformed into images with Gaussian noise, and transformed back [25, 31]. In addition, a Gaussian denoising solution can serve as a proximal [29, 21] for image regularizers. It can be a substitute for the costly step in half-quadratic splitting (HQS) optimization, typically responsible for non-differentiable regularization in image processing. This approach is taken in the recent HQS method that leverages the denoiser for image restoration [50]. We thus work with the assumption of an additive white Gaussian noise model.

**Image denoisers.** Having to know the exact noise level is a serious limitation in practice for denoisers, and to know it ahead of time, before training, is even more limiting. A fixed and known noise level is also a limitation when denoising images with spatially-varying noise level [51]. Not having a universal denoising model means that multiple models need to be trained and stored for different noise levels, and that noise level knowledge is required at test time. The recent method [50] that generalizes to image restoration tasks is a non-universal non-blind denoiser, where 25 denoising networks are used for noise levels below $50$, and even training parameters are chosen based on the noise level. Similarly, Remez *et al.* [32], who reach PSNR results on par with the state of the art, are another non-universal non-blind example. To leverage better priors, images are first classified into a set of classes and every single class has its specific deep network. The method is also not blind and is trained per noise level. Zhang *et al.* [52] present a universal non-blind network for multiple super-resolution degradations by denoising, deblurring, and super-resolving images. They report that although a blind version is more practical, their blind approach fails to perform consistently well since it cannot generalize.

**Blind universal denoisers.** The state-of-the-art Gaussian denoiser DnCNN is both universal and blind [49]. It is a deep network that is jointly trained on randomly-sampled noise level patches to generalize denoising to a range of noise levels. It has not been outperformed yet by other methods, whether blind or not [39, 14]. Only the recent FFDNet [51] improves on DnCNN for noise levels $50$ and $75$ by $0.06$ and $0.15dB$ respectively, on the Berkeley BSD68 set, while performing similarly or worse for other levels. It is, however, not a blind network as it requires a noise level map as input. Lefkimmiatis [21] recently studied universal denoising, building on prior work for modeling patch similarity in CNNs [20]. His methods are, strictly speaking, not universal as two networks are trained separately, one for low ($\leq 30$) and one for high noise levels ($\in [30, 55]$). They are thus non-blind since a noise-level-based choice must be made at inference time. Furthermore, the published results do not outperform the blind DnCNN denoising results. We thus conduct evaluation comparisons of our BUIFD method with the state-of-the-art DnCNN and the classic BM3D approach [6, 7], which is the best non-learning-based denoiser. It leverages image self-similarities by jointly filtering similar image patches. The authors also present a blind version of the BM3D algorithm, and we compare to both blind and non-blind versions.

Our proposed image denoiser BUIFD learns to disentangle a prior and a noise level feature representation. They serve as inputs to the fusion part of the network, responsible for general denoising. Disentangling the feature space is fundamental for interpretability [4], partial transfer learning [47], domain translation [44], domain adaptation [48], specific attribute manipulation [10, 23, 53] and multi-task networks [1]. In our case, it is fundamental for our theoretical denoising function since the disentangled representations serve as its inputs.

## 3. Image Fusion Denoising

### 3.1. Theoretical framework

Although some specific applications can have a more accurate modeling [19, 40], an additive white Gaussian noise model is often assumed in denoising tasks, as it models common acquisition channels [42]. We thus assume that the additive independent and identically distributed noise $n$ follows a Gaussian distribution $\mathcal{N}(0, \sigma_n^2)$, and is uncorrelated with the data $x$. The noise standard deviation $\sigma_n$ is

called noise level. In a Bayesian framework, the conditional probability distribution of the noiseless data $x$ given a noisy observation $y$ (where $y = x + n$) is given by the relation

$$p_{X|Y}(x|y) = \frac{p_{Y,X}(y,x)}{p_Y(y)} = \frac{p_{Y|X}(y|x)p_X(x)}{p_Y(y)}, \quad (1)$$

where $X$ and $Y$ are the random variables corresponding respectively to $x$ and $y$. We are interested in the conditional distribution as we search for the Maximum Aposteriori Probability (MAP) estimate $\hat{x}$ of $x$. The former is

$$\hat{x} = \arg\max_x p_{X|Y}(x|y). \quad (2)$$

We also model the data prior on $x$ as a Gaussian distribution $\mathcal{N}(\bar{x}, \sigma_x^2)$ centered at $\bar{x}$ [33]. We later modify this assumption in Sec. 4 to the practical case of real-world images. The conditional probability of $y$ given a noiseless $x$ value is

$$p_{Y|X}(y|x) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(y-x)^2}{2\sigma_n^2}}, \quad (3)$$

and the probability distribution of $y$ is the convolution of those of $x$ and $n$, given in the Gaussian case by

$$p_Y(y) = p_X(x) \circledast p_N(n) = \frac{e^{-\frac{(y-\bar{x})^2}{2(\sigma_x^2+\sigma_n^2)}}}{\sqrt{2\pi(\sigma_x^2+\sigma_n^2)}}, \quad (4)$$

where $\circledast$ is the convolution operator. With these probability distribution functions, we can obtain an expression for the conditional distribution of $x$ given its noisy observation $y$. Substituting Eq. (3) and Eq. (4) into Eq. (1), we obtain

$$p_{X|Y}(x|y) = \frac{e^{-\frac{(x-\bar{x})^2}{2\sigma_x^2} - \frac{(y-x)^2}{2\sigma_n^2} + \frac{(y-\bar{x})^2}{2(\sigma_x^2+\sigma_n^2)}}}{\sqrt{2\pi(\sigma_x^2\sigma_n^2)/(\sigma_x^2+\sigma_n^2)}}, \quad (5)$$

which can also be written in the following form of a Gaussian in $x$, given an observation $y$

$$p_{X|Y}(x|y) = \frac{1}{\sqrt{2\pi\hat{\sigma}_x^2}} e^{-\frac{(x-\hat{\mu})^2}{2\hat{\sigma}_x^2}}. \quad (6)$$

By matching Eq. (5) and Eq. (6) for all possible $x$ values, we obtain the expressions for $\hat{\mu}$ and $\hat{\sigma}^2$

$$\hat{\mu} = \frac{\sigma_n^2\bar{x} + \sigma_x^2 y}{\sigma_x^2 + \sigma_n^2}, \qquad \hat{\sigma}^2 = \frac{\sigma_x^2\sigma_n^2}{\sigma_x^2 + \sigma_n^2}. \quad (7)$$

For the Gaussian shown in Eq. (6), the MAP estimator is also the conditional expected value (mode and mean being equal) and it is hence given by

$$\hat{x} = \mathbf{E}[x|y] = \int_{-\infty}^{\infty} x \cdot p_{X|Y}(x|y)dx, \quad (8)$$

which, using Eq. (6), can be directly derived to be

$$\hat{x} = \frac{\bar{x}}{1+S} + \frac{y}{1+1/S}, \quad (9)$$

where $S \triangleq \sigma_x^2/\sigma_n^2$ and stands for Signal-to-Noise Ratio (SNR). We call this operation fusion denoising as it fuses the prior and the noisy image, based on the SNR.

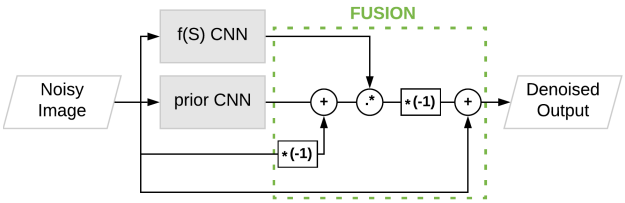
Image denoising models are typically trained to maximize PSNR or equivalently minimize Mean Squared Error (MSE) loss. This means that with close-to-optimal convergence of a neural network model (MSE loss $\rightarrow 0^+$), its output tends towards the minimum MSE estimator (MMSE). With our Gaussian modeling, this leads to the MAP estimator $\hat{x}$ of Eq. (9). Thus, an MSE reconstruction loss in a neural network leads it to the estimator $\hat{x}$, *iff* $S$ and $\bar{x}$ are correctly predicted and correctly used in the fusion with the noisy input $y$, as in Eq. (9). The optimal fusion, used as reference in our experimental evaluation in Sec. 3.5, is given the exact $S$ and $\bar{x}$ values for Eq. (9).

### 3.2. Fusion Net architecture

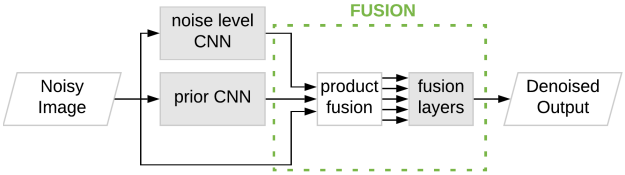
We incorporate the basic structure of the optimal fusion solution into the architecture of a neural network, which we call *Fusion Net*. We build the main blocks of our Fusion Net based on the blind DnCNN introduced in [49] and illustrated in Fig. 1(a). In Fig. 1, the noise-predicting CNN of DnCNN, the prior-predicting CNN, and the one predicting $f(S)$ (where $f(S) \triangleq \frac{1}{1+S}$) in our Fusion Net, all leverage the same DnCNN architecture design. The CNNs are all constituted of a sequence of convolution layers, rectified linear units (ReLU) [27] and batch normalization blocks [16]. Note that $f(S)$ is inversely-proportional to the SNR and proportional to the noise level. It is the factor multiplying the prior in Eq. (9).

Unlike the DnCNN that predicts the *noise* values in the input noisy image, then subtracts them from the noisy input to yield the final denoised output, our network learns optimal *fusion denoising* given by the function in Eq. (9), as illustrated in Fig. 1(b). The same depth and capacity of the DnCNN are retained to learn separately the image prior and the SNR function, $f(S)$, that is required for the weighted fusion of the prior and the noisy input image. Note that SNR learning also contains a form of prior knowledge, but of variance rather than of expectation. We subtract from the prior our noisy input image and multiply the result, pixel-wise, with the SNR function. This yields the noise prediction given a noisy input, which we subtract from the latter to obtain the denoised output. This architecture is mathematically equivalent to Eq. (9). However, the wiring of Fig. 1(b) allows us to clearly have a residual learning connection and to keep the parallelism between the two aforementioned networks.


(a) DnCNN




(b) Fusion Net




(c) BUIFD

Figure 1. (a) Schematic of the DnCNN residual learning approach for denoising. The network predicts the noise in an image. (b) Our Fusion Net that explicitly learns the SNR function for optimal fusion of the noisy image with the learned prior, following Eq. (9). (c) Our real-image fusion denoiser, BUIFD, where fusion is carried out with a pixel-wise product stage followed by three convolution layers for learning a general fusion function (Sec. 4.1).

### 3.3. Fusion Net feature disentangling

To mimic the optimal fusion between image prior and noisy image based on the SNR, as in Eq. (9), both the architecture and loss function are adapted. For the fusion, the network needs to predict the image prior $\bar{x}$ and $f(S)$ per pixel (Fig. 1(b)). We obtain, with close-to-zero MSE reconstruction loss of our Fusion Net, that the ground-truth target and the network output are approximately equal

$$\begin{cases} \bar{x} \cdot f(S) + y \cdot (1 - f(S)) \approx a \cdot b + y \cdot (1 - b) \\ \forall y \in \mathcal{D}^{\mathcal{T}}, \end{cases} \tag{10}$$

where $a$ and $b$ are feature representations in the Fusion Net, and $y$ is the noisy input. As Eq. (10) holds for all $y$ in the training dataset $\mathcal{D}^{\mathcal{T}}$, and as the dataset is assumed general enough, we can apply coefficient equating. We conclude that the network disentangled representations $\{a, b\}$ are respectively equal to the prior and the SNR function $\{\bar{x}, f(S)\}$, with close-to-zero MSE reconstruction loss.

We can further incorporate optimal denoising information in the Fusion Net, under the theoretical settings described in Sec. 3.1, through explicit SNR learning with a dedicated loss term. The fusion representations, i.e. the prior $\bar{x}$ and $f(S)$, are thus further enforced through a penalty term for predicting $f(S)$ in the loss function. The full loss function $\mathcal{L}_f$ of the Fusion Net is given by

$$\mathcal{L}_f = \alpha||a \cdot b + y \cdot (1-b) - x||_2^2 + (1-\alpha)||b - f(S)||_2^2, \tag{11}$$

where $\alpha$ is a weight parameter, the first term is the MSE reconstruction loss similar to that of the DnCNN, and the second term is a reconstruction loss for $f(S)$. Following Eq.(10), $a \cdot b + y \cdot (1 - b)$ is the denoised output of the Fusion Net.

The Fusion Net therefore minimizes the reconstruction loss over the denoised image by learning to predict the image prior and the SNR function values separately. Unlike the DnCNN residual learning network, which only leverages ground-truth noise-free images during training, the Fusion Net also leverages explicit SNR information.

### 3.4. Experimental setup

The networks are trained (and tested) with data generated synthetically according to the theoretical assumption of a Gaussian image prior as defined in Sec. 3.1. The training data is composed of over 200k patches of size $40 \times 40$ pixels. Image pixel intensities for the training data are drawn at random from $\mathcal{N}(127, 25^2)$, following the Gaussian image prior assumption, and all values are normalized to $[0, 1]$ before the training through division by 255 and clipping of all values outside the interval to the interval's closer bound when noise is added. For the testing data, 256 images of size $256 \times 256$ pixels are used, and they are created with the same procedure as that of the training data.

We train the networks for 50 epochs with batches of size 128. We use the Adam optimizer [17] with an initial learning rate of $0.001$ that is decayed by a factor of 10 every 30 epochs, the remaining parameters being set to the default values. The weight $\alpha$ in Eq. (11) is set to $0.1$. We train the networks with multiple levels of noise. The standard deviation of the additive Gaussian noise is chosen uniformly at random within the interval $[5, 25]$ during the training. At the end of every epoch, the noise components are re-sampled, following the same procedure, but not the ground-truth images. For the testing phase, the networks are evaluated on test images where the added noise is also Gaussian, with a given standard deviation.

### 3.5. Evaluation

PSNR results of DnCNN, our Fusion Net, as well as the optimal upper bound are presented in Table 1. The optimal upper bound denoising performance is that of the optimal mathematical solution in Eq. (9). We can see that both the DnCNN and the Fusion Net perform similarly on the training noise levels (left half of the table), and very close to optimal. To validate that the results are indeed statistically similar, we analyze the distribution of PSNR values across

| $\sigma$ | **5** | **10** | **15** | **20** | **25** | **30** | **40** | **50** | **60** | **70** |
|---|---|---|---|---|---|---|---|---|---|---|
| Optimal Fusion | 34.325 | 28.778 | 25.947 | 24.261 | 23.185 | 22.464 | 21.604 | 21.138 | 20.860 | 20.681 |
| DnCNN | 34.158 | 28.736 | 25.920 | 24.245 | 23.169 | 22.281 | 20.490 | 18.925 | 17.548 | 16.372 |
| Fusion Net | 34.158 | 28.734 | 25.922 | 24.249 | 23.173 | 22.346 | 21.310 | 20.908 | 20.609 | 19.669 |
| $p$-value | 0.760 | 0.568 | 0.465 | 0.100 | 0.053 | $\approx 0$ | $\approx 0$ | $\approx 0$ | $\approx 0$ | $\approx 0$ |

Table 1. Test set PSNR ($dB$) results for the noise standard deviations given in the top row. The networks are trained on multiple noise levels randomly chosen within the interval $[5, 25]$. On the other hand, noise levels in the right half of the table are not seen in the training. We also report the optimal mathematical denoising performance (Optimal Fusion). The bottom row shows the independent two-sample T-test results of the pair of networks for each of the testing noise levels. We report the two-tailed $p$-values validating the null hypothesis of equal average PSNR performances between DnCNN and the Fusion Net over the training noise levels, with a significance level of $0.05$.

the test set. A two-sided T-test (independent two-sample T-test) is used to evaluate the null hypothesis that the PSNR results of both networks have similar expected values. This test is chosen as we have the exact same sample sizes defined by the test dataset, and the variances of PSNR results are very similar. The T-test results are given in the bottom row of Table 1, and the null hypothesis holds for all configurations in the left half of the table (for a $0.05$ significance level, i.e., a $p$-value $\geq 0.05$). This proves that the Fusion Net, despite the modeling that mimics optimal denoising fusion and the additional training information to learn SNR values, performs similarly to the DnCNN. The latter has therefore enough capacity and learns an optimal denoising. This, however, only holds for the noise levels seen by the networks, shown in the left half of Table 1. The confidence in the null hypothesis decreases with increasing test noise levels. With a significance level above $0.053$, the null hypothesis would even be rejected for noise level $25$.

The evaluation results on noise levels larger than $25$, which are not trained on by any of the networks, are reported in the right half of Table 1. For these larger noise levels, the null hypothesis is very clearly rejected as there is a growing performance gap between DnCNN and our Fusion Net. The $p$-value quickly drops to zero when there is a PSNR gap, since variances are very small in our results. The Fusion Net generalizes better to unseen noise levels, even performing close to optimal up to noise level $60$. The further we increase the noise level, the larger is the performance gap between the Fusion Net and the DnCNN. Although both networks perform well for the training noise levels, the Fusion Net learns a more general model and clearly outperforms on unseen noise levels.

# 4. Denoising Real Images

## 4.1. Method

Here, our main objectives are to (1) design a *Blind Universal Image Fusion Denoiser (BUIFD)* for real images, by adapting the theoretical fusion strategy integrated in our Fusion Net, (2) evaluate the denoising performance of BUIFD on training noise levels, and (3) assess the generalization to unseen noise levels with real images.

Since a real image cannot be modeled with a simple Gaussian prior, our image fusion denoising network used for real images (BUIFD), shown in Fig. 1(c), is adapted from the theoretical Fusion Net, shown in Fig. 1(b), by modifying the fusion part. We replace the optimal mathematical fusion by a product fusion step followed by trainable convolution layers. We use three convolution layers to learn the data-dependent fusion function. The optimal fusion function $F$ is to be applied on the noisy input image $y$, the prior prediction, and the noise level prediction

$$\hat{x} = F(y,\ f_P(y, \theta_P),\ f_N(y, \theta_N)), \tag{12}$$

where the prior-predicting and noise-level-predicting network functions are respectively $f_P$ and $f_N$, with their corresponding learned parameters $\theta_P$ and $\theta_N$, and the denoised estimate is $\hat{x}$. The optimal fusion $F$ can be approximated by $\hat{F}$ modeled with three convolution layers. However, we expect $F$ to contain pixel-wise inter-input multiplications similar to the ones of Eq. (9). Since such pixel-wise multiplications cannot be replicated with convolutions, we pass two additional inputs into the convolution layers that model $\hat{F}$. These additional inputs are concatenated with those of Eq. (12) and are given by

$$f_P(y, \theta_P) \odot f_N(y, \theta_N), \quad y \odot (1 - f_N(y, \theta_N)), \tag{13}$$

where $\odot$ is pixel-wise multiplication. The two additional inputs reduce the learning burden of the convolution layers and improve the denoising performance. Note that we normalize $f_N(\cdot, \cdot) \in [0, 1]$. We call this pixel-wise multiplication step and the concatenation of the additional inputs the product fusion (shown in the pipeline of Fig. 1(c)). These two fusion steps, namely the product fusion and the three convolution layers, form $\hat{F}$ and realize point (1) above. The BUIFD's optimization loss is given by

$$\mathcal{L}_f = ||\hat{F}(\mathbf{C}) - x||_2^2 + ||f_N(y, \theta_N) - N||_2^2, \tag{14}$$

where $\mathbf{C}$ is the concatenation of the inputs $\{y,\ f_P(y, \theta_P),\ f_N(y, \theta_N),\ f_P(y, \theta_P) \odot f_N(y, \theta_N),\ y \odot (1 - f_N(y, \theta_N))\}$ listed in Eq. (12) and (13), $x$ is the ground-truth original image, and $f_N(y, \theta_N)$ and $N$ are respectively the predicted

| Method | Blind | Test noise level (standard deviation) 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| BM3D [6] | **No** | 37.56 | 33.26 | 30.97 | 29.44 | 28.31 | 27.42 | 26.65 | 25.98 |
| | Yes | 29.33 | 29.18 | 28.95 | 28.70 | 28.31 | 27.32 | 25.13 | 22.38 |
| $\text{DnCNN}_{55}$ [49] | Yes | 37.65 | 33.61 | 31.34 | 29.70 | 28.39 | 27.29 | 26.32 | 25.48 |
| **BUIFD**$_{55}$ | Yes | 37.41 | 33.55 | 31.40 | 29.91 | 28.77 | 27.81 | 27.00 | 26.26 |
| $\text{DnCNN}_{75}$ [49] | Yes | 37.64 | 33.62 | 31.38 | 29.79 | 28.54 | 27.52 | 26.63 | 25.87 |
| **BUIFD**$_{75}$ | Yes | 37.25 | 33.47 | 31.35 | 29.88 | 28.75 | 27.82 | 27.03 | 26.31 |
| | | **45** | **50** | **55** | **60** | **65** | **70** | **75** | **Mean** |
| BM3D [6] | **No** | 25.28 | 24.77 | 24.29 | 23.84 | 23.42 | 23.00 | 22.62 | 27.12 |
| | Yes | 19.95 | 18.16 | 16.83 | 15.76 | 14.87 | 14.11 | 13.45 | 22.16 |
| $\text{DnCNN}_{55}$ [49] | Yes | 24.70 | 23.99 | 23.34 | 22.62 | 21.47 | 19.78 | 18.15 | 26.25 |
| **BUIFD**$_{55}$ | Yes | 25.62 | 25.00 | 24.43 | 23.82 | 23.08 | 22.11 | 20.97 | 27.14 |
| $\text{DnCNN}_{75}$ [49] | Yes | 25.15 | 24.49 | 23.89 | 23.35 | 22.86 | 22.41 | 21.99 | 27.01 |
| **BUIFD**$_{75}$ | Yes | 25.67 | 25.10 | 24.55 | 24.05 | 23.56 | 23.10 | 22.67 | 27.37 |

Table 2. PSNR ($dB$) comparisons of *grayscale* image denoising on the BSD68 standard test set. We compare the non-blind BM3D, the blind BM3D, DnCNN, and our BUIFD. $\text{DnCNN}_\sigma$ or $\text{BUIFD}_\sigma$ indicates that the network sees noise levels *only* up to $\sigma$ during the training. Note: small deviations in reported PSNR values compared with the literature, notably on higher noise levels, are due to clipping noisy inputs (and outputs) to $[0, 255]$, as a practical consideration. Red indicates the best blind result, for each range of training noise levels.

| $\sigma_c$ | **15** | **25** | **40** | **55** | **65** |
|---|---|---|---|---|---|
| $\text{BM3D}_{NB}$ | 29.30 | 27.80 | 25.75 | 24.28 | 23.41 |
| BM3D | 28.94 | 27.80 | 21.63 | 16.78 | 14.85 |
| $\text{DnCNN}_{55}$ | 31.24 | 28.32 | 25.41 | 23.17 | 20.83 |
| **BUIFD**$_{55}$ | 31.38 | 28.74 | 26.22 | 24.33 | 22.81 |
| $\text{DnCNN}_{75}$ | 31.31 | 28.51 | 25.80 | 23.87 | 22.83 |
| **BUIFD**$_{75}$ | 31.34 | 28.73 | 26.29 | 24.52 | 23.53 |

Table 3. We evaluate PSNR values, with spatially-varying noise level, on BSD68. The noise level increases linearly within the image over the range $[\sigma_c - 10, \sigma_c + 10]$. The non-blind BM3D is given the central noise level $\sigma_c$, and we refer to it as $\text{BM3D}_{NB}$.

and ground-truth noise level values, normalized to $[0, 1]$. We discuss the relation between BUIFD (Fig. 1(c)) and our theoretical Bayesian network Fusion Net (Fig. 1(b)) in detail in the supplementary material.

### 4.2. Experimental setup

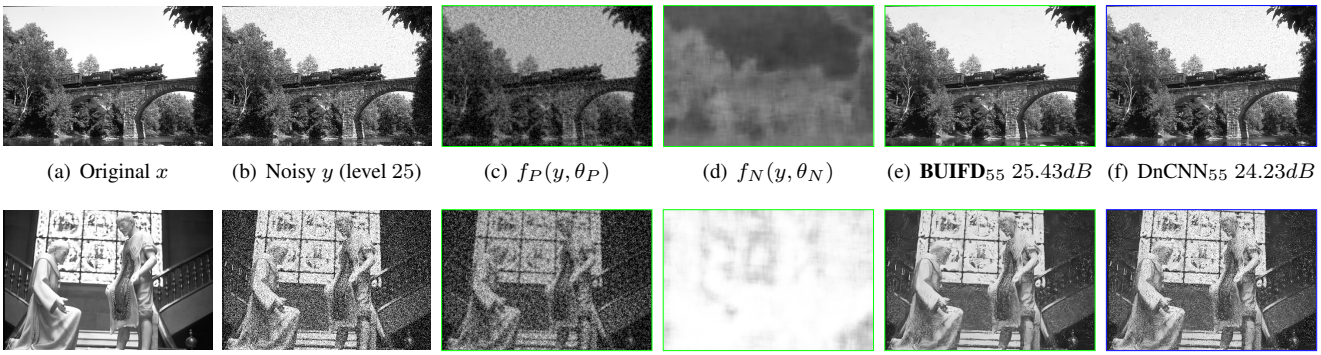
We use the implementation referenced by the authors of DnCNN and the same datasets[2]. As mentioned in Sec. 4.1, the architecture of our prior-predicting network is identical to that of DnCNN. All the network details are available in [49] and we omit the repetition. The same network depth and feature layers are thus used in the prior-predicting network (18 main blocks) in Fig. 1(c). The noise level network is a shallower one made up of 5 blocks similar to the ones used in the prior predictor. Each block is a convolution followed by a batch normalization and a ReLU, and we append to the noise level predictor a convolution followed by an application of the logistic sigmoid function to obtain the normalized $f_N(\cdot, \cdot) \in [0, 1]$. The noise level values are thus mapped during the training to the range $[0, 1]$ by dividing by the largest *training* noise level. The three convolution layers approximating the final fusion have 16 channels. Both the BUIFD and the DnCNN networks are trained with the same training parameters and optimization settings, similar to Sec. 3.4. The noise level predictor is jointly trained within the BUIFD, so both network branches always see the same training data and noise levels as each other in the experiments of Sec 4.3. We use the 400 Berkeley images [5, 36] for grayscale training and the 432 color Berkeley images for color training, as in [49]. The same architectures are retained for grayscale and color networks.

[2]https://github.com/SaoYan/DnCNN-PyTorch

### 4.3. Evaluation

Grayscale denoising evaluation is carried out over the standard Berkeley 68 image test set (BSD68) [34] taken from [26]. Table 2 reports the results of our fusion approach and of the state-of-the-art blind DnCNN, when they are both trained with noise levels up to 55 or up to 75. Note that for our fusion approach that is trained up to noise level 55, we map the maximum network prediction of 1, during training, to 55 and not to the maximum test noise level, for a more fair comparison. The results of the blind version of BM3D as well as those of the non-blind BM3D, which is given the correct test noise level at inference time, are also reported for reference. We restrict all noisy test images to the range $[0, 255]$, as having negative intensities, or values exceeding 255, is not a configuration encountered in practice.

Fig. 2 shows our intermediate feature results, the prior and the noise level values, along with denoising results. The denoised image is created by fusing the noisy input image with the network-derived prior and the noise level values. The fusion is carried out by the product fusion step and the



(a) Original $x$ (b) Noisy $y$ (level 25) (c) $f_P(y, \theta_P)$ (d) $f_N(y, \theta_N)$ (e) **BUIFD**$_{55}$ $25.43dB$ (f) DnCNN$_{55}$ $24.23dB$

(g) Original $x$ (h) Noisy $y$ (level 75) (i) $f_P(y, \theta_P)$ (j) $f_N(y, \theta_N)$ (k) **BUIFD**$_{55}$ $20.18dB$ (l) DnCNN$_{55}$ $17.22dB$

Figure 2. Left to right: original and noisy images, prior and noise level predictions of BUIFD, our fused denoising result and the DnCNN denoised image. Our denoising result is created by fusing the noisy image, the prior and the noise level values, for instance (e) is $\hat{F}((b), (c), (d))$. All the networks are trained on noise levels in $[0, 55]$. Whether the noise level is seen (25), or not seen (75), during training, our denoised results are more precise and smoother (sky in (e-f), window, wall and arms in (k-l)). Best viewed on screen.

| | | **Test noise level (standard deviation)** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Blind** | **5** | **10** | **15** | **20** | **25** | **30** | **35** | **40** |
| CBM3D [7] | **No** | 40.19 | 35.75 | 33.25 | 31.52 | 30.18 | 29.07 | 28.09 | 27.18 |
| | Yes | 28.17 | 28.08 | 27.94 | 27.74 | 27.49 | 27.21 | 26.90 | 26.58 |
| CDnCNN$_{55}$ [49] | Yes | 40.05 | 35.92 | 33.57 | 31.93 | 30.66 | 29.61 | 28.71 | 27.91 |
| **CBUIFD**$_{55}$ | Yes | 40.07 | 36.01 | 33.66 | 32.02 | 30.75 | 29.71 | 28.81 | 28.00 |
| CDnCNN$_{75}$ [49] | Yes | 39.75 | 35.74 | 33.46 | 31.86 | 30.62 | 29.59 | 28.70 | 27.91 |
| **CBUIFD**$_{75}$ | Yes | 40.05 | 35.98 | 33.66 | 32.02 | 30.76 | 29.71 | 28.81 | 28.01 |
| | | **45** | **50** | **55** | **60** | **65** | **70** | **75** | **Mean** |
| CBM3D [7] | **No** | 26.53 | 25.85 | 25.22 | 24.62 | 24.05 | 23.51 | 22.99 | 28.53 |
| | Yes | 26.23 | 25.85 | 25.41 | 24.83 | 24.05 | 23.07 | 21.93 | 26.10 |
| CDnCNN$_{55}$ [49] | Yes | 27.17 | 26.49 | 25.84 | 25.24 | 24.66 | 24.09 | 23.52 | 29.02 |
| **CBUIFD**$_{55}$ | Yes | 27.28 | 26.59 | 25.94 | 25.34 | 24.75 | 24.17 | 23.62 | 29.11 |
| CDnCNN$_{75}$ [49] | Yes | 27.19 | 26.52 | 25.87 | 25.27 | 24.70 | 24.13 | 23.59 | 28.99 |
| **CBUIFD**$_{75}$ | Yes | 27.28 | 26.60 | 25.95 | 25.34 | 24.75 | 24.18 | 23.63 | 29.12 |

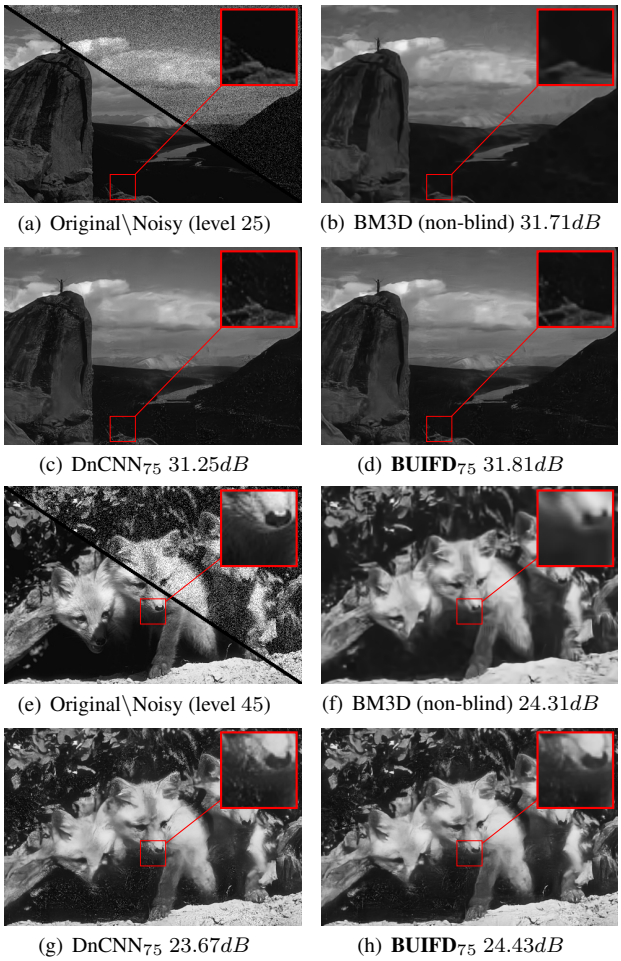
Table 4. PSNR ($dB$) comparisons of *color* image denoising, similar to Table 2, on the CBSD68 standard test set. Red also indicates the best blind result, for each range of training noise levels.

three convolution layers. As in practical scenarios, the denoised outputs are clipped to $[0, 255]$, as are the noisy input images. Our results are smoother compared with those of DnCNN over low frequency regions, and details are better reconstructed over the high frequency content.

As seen in Table 2, our fusion approach improves the PSNR at every single noise level starting from $15 - 20$, which includes seen levels for both training ranges. Comparing DnCNN$_{75}$ and BUIFD$_{75}$, which are trained on all noise levels, we also note with our approach an improvement of up to $0.7dB$ and an average improvement of $0.36dB$. We outperform even the non-blind version of BM3D by an average of $0.25dB$ with our version trained on all noise levels and we perform just as well as the non-blind BM3D when training only up to level 55. Comparing the results of DnCNN$_{55}$ and of BUIFD$_{55}$ in Table 2, for unseen noise levels in the range $(55, 75]$, we see that the generalization of the fusion approach to unseen noise levels indeed applies to real images. The improvement of $2.82dB$ for level 75 is consistent with that obtained in our synthetic experiment in Table 1.

The results in Table 3 illustrate denoising images with spatially-varying noise level, without re-training the networks. Noise is added across an image with a level that increases linearly with rows. For the non-blind BM3D, we input the average noise level as a guide. The BUIFD network can handle spatially-varying noise, which neither the prior nor the noise level predicting network branches are trained on. It outperforms DnCNN on all noise setups, whether the networks are trained on the full range or only up to level 55.

For color image denoising, we use the standard color version of BSD68 (CBSD68) for testing. PSNR results are reported in Table 4. The high inter-channel correlation between the RGB color channels [11] allows all methods to



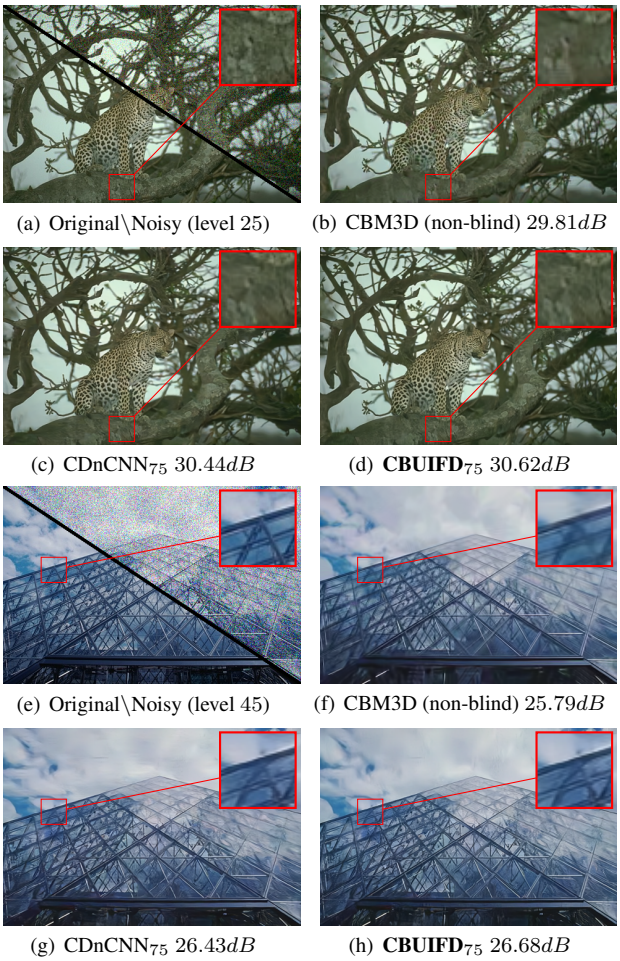
(a) Original\Noisy (level 25) (b) BM3D (non-blind) 31.71$dB$

(c) DnCNN$_{75}$ 31.25$dB$ (d) **BUIFD**$_{75}$ 31.81$dB$

(e) Original\Noisy (level 45) (f) BM3D (non-blind) 24.31$dB$

(g) DnCNN$_{75}$ 23.67$dB$ (h) **BUIFD**$_{75}$ 24.43$dB$

Figure 3. Grayscale image denoising from BSD68. All networks are trained on all noise levels $[0, 75]$ and we test on noise levels 25 and 45. Non-blind BM3D results are very smoothed, and details are lost. DnCNN preserves more details, but at the expense of PSNR. Our blind approach preserves details and outperforms the non-blind BM3D in terms of PSNR. Best viewed on screen.



(a) Original\Noisy (level 25) (b) CBM3D (non-blind) 29.81$dB$

(c) CDnCNN$_{75}$ 30.44$dB$ (d) **CBUIFD**$_{75}$ 30.62$dB$

(e) Original\Noisy (level 45) (f) CBM3D (non-blind) 25.79$dB$

(g) CDnCNN$_{75}$ 26.43$dB$ (h) **CBUIFD**$_{75}$ 26.68$dB$

Figure 4. Examples of color image denoising from CBSD68. All networks are trained on the full range of noise levels $[0, 75]$ and we test on noise levels 25 and 45. Best viewed on screen.

perform significantly better in terms of PSNR on color images compared with grayscale. We hypothesize that this correlation also enables the networks to implicitly learn the noise level prediction. High correlation implies that the network sees multiple approximately equal data samples with different noise instances drawn from the same distribution. Thus, it more easily learns an estimate of the noise variance. Each of the two networks therefore performs more or less the same when trained up to noise level 55 and when trained up to noise levels 75. Our fusion approach, however, consistently outperforms CDnCNN on every single noise level for both training noise ranges. Our average improvement over CDnCNN is about $0.1dB$. We also note that the networks outperform, on average, even the non-blind CBM3D by about $0.5dB$ for CDnCNN and $0.6dB$ for our CBUIFD.

Sample image denoising results for grayscale and color images are illustrated in Fig. 3 and Fig. 4 respectively, for the non-blind BM3D and the blind networks DnCNN and BUIFD trained on the full range of noise levels. More results are provided in the supplementary material.

### 4.4. Relation with the Bayesian framework

The Fusion Net in Fig. 1(b) explicitly models the relation with the Bayesian solution in the theoretical experiments. We discuss in what follows the relation between BUIFD (Fig. 1(c)) and the Bayesian solution Eq. (9). We first note that a Gaussian prior does not perfectly model real images, and thus, we expect that the real-image BUIFD network (Fig. 1(c)) deviates from the Fusion Net (Fig. 1(b)), from which it is inspired, to adapt to real images. However, as addressed in Sec. 4.1, the relation between BUIFD and the Bayesian framework is strongly pertinent.

First, the product fusion Eq. 13 explicitly creates *the same components* as in the Bayesian equation Eq. (9). This product fusion weighs noisy input and learned prior based

on SNR, as in the Bayesian fusion. The fusion layers are only 3 convolutional layers with no non-linearities, to ensure that mostly an additive fusion of our Bayesian terms takes place, with local smoothing, and the relation with the Bayesian solution is preserved as much as possible.

Second, we do not predict an image prior in the sense of a pixel intensity probability distribution, but only the expected mean of that *unknown* distribution. In the literature, priors are often probability distributions of image gradients, but our definition is quite distinctive. *Our prior is, per pixel, the expected value of the distribution out of which the pixel's intensity was sampled*. Even with noise-free images, one cannot exactly know that distribution (nor its mean), per pixel, to assess how much this definition is still respected in the BUIFD network with real images. However, all other Bayesian components are consistent, and the empirical results as well. Our improvement of $3.30dB$ at unseen noise level 70 in the theoretical experiment is paralleled by an improvement of $2.82dB$ at noise level 75 in the real image BSD68 experiment.

## 5. Conclusion

We define a theoretical framework under which we derive an optimal denoising solution that we call fusion denoising. We integrate it into a deep learning architecture and compare with the optimal mathematical solution and with the state-of-the-art blind universal denoiser DnCNN. Our synthetic experimental results show that our Fusion Net generalizes far better to higher unseen noise levels.

We learn a data-dependent fusion function to adapt our fusion denoising network to real images. Our blind universal image fusion denoising network BUIFD improves the state-of-the-art real image denoising performance both on training noise levels and on unseen noise levels.

## References


[1] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8):1798–1828, 2013. 2

[2] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron. Unprocessing images for learned raw denoising. *arXiv preprint arXiv:1811.11127*, 2018. 2

[3] A. Buades, B. Coll, and J.-M. Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008. 2

[4] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, pages 2172–2180, 2016. 2

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

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE TIP*, 16(8):2080–2095, 2007. 1, 2, 6

[7] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *ICIP*, pages 313–316, 2007. 2, 7

[8] W. Dong, L. Zhang, and G. Shi. Centralized sparse representation for image restoration. In *ICCV*, pages 1259–1266, 2011. 2

[9] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE TIP*, 22(4):1620–1630, 2013. 2

[10] M. El Helou, S. Mandt, A. Krause, and P. Beardsley. Mobile robotic painting of texture. In *ICRA*, 2019. 2

[11] M. El Helou, Z. Sadeghipoor, and S. Süsstrunk. Correlation-based deblurring leveraging multispectral chromatic aberration in color and near-infrared joint acquisition. In *ICIP*, pages 1402–1406, 2017. 2, 7

[12] M. Elad. *Image Denoising*, pages 273–307. Springer New York, New York, NY, 2010. 1

[13] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE TIP*, 15(12):3736–3745, 2006. 2

[14] C. Godard, K. Matzen, and M. Uyttendaele. Deep burst denoising. In *ECCV*, pages 538–554, 2018. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 1, 3

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[18] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NeurIPS*, pages 1033–1041, 2009. 2

[19] Y. Le Montagner, E. D. Angelini, and J.-C. Olivo-Marin. An unbiased risk estimator for image denoising in the presence of mixed Poisson–Gaussian noise. *IEEE TIP*, 23(3):1255–1268, 2014. 2

[20] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *CVPR*, pages 3587–3596, 2017. 2

[21] S. Lefkimmiatis. Universal denoising networks: A novel CNN architecture for image denoising. In *CVPR*, pages 3204–3213, 2018. 2

[22] D. Liu, B. Wen, X. Liu, Z. Wang, and T. S. Huang. When image denoising meets high-level vision tasks: a deep learning approach. In *IJCAI*, pages 842–848, 2018. 1

[23] Y.-C. Liu, Y.-Y. Yeh, T.-C. Fu, S.-D. Wang, W.-C. Chiu, and Y.-C. Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *CVPR*, pages 8867–8876, 2018. 2

[24] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, pages 2272–2279, 2009. 2

[25] M. Mäkitalo and A. Foi. Noise parameter mismatch in variance stabilization, with an application to Poisson–Gaussian noise estimation. *IEEE TIP*, 23(12):5348–5359, 2014. 2

[26] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–423, 2001. 6

[27] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *ICML*, pages 807–814, 2010. 3

[28] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005. 2

[29] N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014. 2

[30] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE TPAMI*, 12(7):629–639, 1990. 2

[31] T. Plötz and S. Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, pages 2750–2759, 2017. 2

[32] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein. Class-aware fully convolutional Gaussian and Poisson denoising. *IEEE TIP*, 27(11):5707–5722, 2018. 2

[33] S. Romdhani and T. Vetter. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *CVPR*, pages 986–993, 2005. 3

[34] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205, 2009. 1, 2, 6

[35] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 2

[36] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *CVPR*, pages 2774–2781, 2014. 6

[37] J.-L. Starck, E. J. Candès, and D. L. Donoho. The curvelet transform for image denoising. *IEEE TIP*, 11:670–684, 2002. 2

[38] M. Suganuma, M. Ozay, and T. Okatani. Exploiting the potential of standard convolutional autoencoders for image restoration by evolutionary search. In *ICML*, pages 4778–4787, 2018. 1

[39] C. Tian, Y. Xu, L. Fei, and K. Yan. Deep learning for image denoising: A survey. *arXiv preprint arXiv:1810.05052*, 2018. 2

[40] G. Wang, C. Lopez-Molina, and B. De Baets. Blob reconstruction using unilateral second order Gaussian kernels with application to high-ISO long-exposure image denoising. In *ICCV*, pages 4817–4825, 2017. 2

[41] Y. Weiss and W. T. Freeman. What makes a good model of natural images? In *CVPR*, pages 1–8, 2007. 2

[42] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *NeurIPS*, pages 341–349, 2012. 2

[43] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *ICCV*, pages 244–252, 2015. 2

[44] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2849–2857, 2017. 2

[45] J. Yongcheng, Y. Yezhou, F. Zunlei, Y. Jingwen, Y. Yizhou, and M. Song. Neural style transfer: A review. *arXiv preprint arXiv:1705.04058v6*, 2018. 1

[46] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016. 1

[47] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, pages 3712–3722, 2018. 2

[48] J. Zhang, W. Li, and P. Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In *CVPR*, pages 1859–1867, 2017. 2

[49] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE TIP*, 26(7):3142–3155, 2017. 1, 2, 3, 6, 7

[50] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep CNN denoiser prior for image restoration. In *CVPR*, pages 3929–3938, 2017. 2

[51] K. Zhang, W. Zuo, and L. Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE TIP*, 27(9):4608–4622, 2018. 1, 2

[52] K. Zhang, W. Zuo, and L. Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, pages 3262–3271, 2018. 2

[53] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee. Unsupervised discovery of object landmarks as structural representations. In *CVPR*, pages 2694–2703, 2018. 2