Real Image Denoising with Feature Attention

Anonymous ICCV submission

Paper ID 3246

Abstract

Deep convolutional neural networks perform better on images containing spatially invariant noise (synthetic noise); however, their performance is limited on real-noisy photographs and requires multiple stage network modeling. To advance the practicability of denoising algorithms, this paper proposes a novel single-stage blind real image denoising network (RIDNet) by employing a modular architecture. We use a residual on the residual structure to ease the flow of low-frequency information and apply feature attention to exploit the channel dependencies. Furthermore, the evaluation in terms of quantitative metrics and visual quality on three synthetic and four real noisy datasets against 19 state-of-the-art algorithms demonstrate the superiority of our RIDNet.

1. Introduction

Image denoising is a low-level vision task that is essential in a number of ways. First of all, during image acquisition, some noise corruption is inevitable and can downgrade the visual quality considerably; therefore, removing noise from the acquired image is a key step for many computer vision and image analysis applications [25]. Secondly, denoising is a unique testing ground for evaluating image prior and optimization methods from a Bayesian perspective [27, 63]. Furthermore, many image restoration tasks can be solved in the unrolled inference through variable splitting methods by a set of denoising subtasks, which further widens the applicability of image denoising [3, 30, 47, 60].

Generally, denoising algorithms can be categorized as model-based and learning-based. Model-based algorithms include non-local self-similarity (NSS) [16, 11, 18], sparsity [27, 45], gradient methods [43, 52, 50], Markov random field models [48], and external denoising priors [8, 57, 39]. The model-based algorithms are computationally expensive, time-consuming, unable to suppress the spatially variant noise directly and characterize complex image textures. On the other hand, discriminative learning aims to model the image prior from a set of noisy and ground-truth image sets. One technique is to learn the prior in steps in the context of truncated inference [15] while another approach is to employ brute force learning, for example, MLP [12] and CNN methods [59, 60]. CNN models [61, 28] improved denoising performance, due to its modeling capacity, network training, and designs. However, the performance of the current learning models is limited and tailored for a specific level of noise.

Noisy CBDNet [28] RIDNet (Ours)

Figure 1. A real noisy face image from RNI15 dataset [35]. Unlike other CBDNet [28], RIDNet does not have over-smoothing or over-contrasting artifacts (Best viewed in color on high-resolution display)

A practical denoising algorithm should be efficient, flexible, perform denoising using a single model and handle both spatially variant and invariant noise when the noise standard-deviation is known or unknown. Unfortunately, the current state-of-the-art algorithms are far from achieving all of these aims. We present a CNN model which is efficient and capable of handling synthetic as well as real-noise present in images. We summarize the contributions of this work in the following paragraphs.

1.1. Contributions

- Present CNN based approaches for real image denoising employ two-stage models; we present the first model that provides state-of-the-art results using only one stage.
- To best of our knowledge, our model is the first to incorporate feature attention in denoising.
- Most current models connect the weight layers consecutively; and so increasing the depth will not help improve performance [19, 38]. Also, such networks

can suffer from vanishing gradients [10]. We present a modular network, where increasing the number of modules helps improve performance.

- We experiment on three synthetic image datasets and four real-image noise datasets to show that our model achieves state-of-the-art results on synthetic and real images quantitatively and qualitatively.

2. Related Works

In this section, we present and discuss recent trends in the image denoising. Two notable denoising algorithms, NLM [11] and BM3D [16], use self-similar patches. Due to theer sucess, many variants were proposed, including SADCT [24], SAPCA [18], NLB [34], and INLM [26] which seek self-similar patches in different transform domains. Dictionary-based methods [22, 40, 20] enforce sparsity by employing self-similar patches and learning over-complete dictionaries from clean images. Many algorithms [63, 23, 55] investigated the maximum likelihood algorithm to learn a statistical prior, *e.g*. the Gaussian Mixture Model of natural patches or patch groups for patch restoration. Furthermore, Levin *et al*. [37] and Chatterjee *et al*. [14], motivated external denoising [8, 6, 39, 58] by showing that an image can be recovered with negligible error by selecting reference patches from a clean external database. However, all of the external algorithms are class-specific.

Recently, Schmidt *et al*. [49] introduced a cascade of shrinkage fields (CSF) which integrated half-quadratic optimization and random-fields. Shrinkage aims to suppress smaller values (noise values) and learn mappings discriminatively. The CSF assumes the data fidelity term to be quadratic and that it has a discrete Fourier transform based closed-form solution.

Currently, due to the popularity of convolutional neural networks (CNNs), image denoising algorithms [59, 60, 36, 12, 49, 7] have achieved a performance boost. Notable denoising neural networks, DnCNN [59], and IrCNN [60] predict the residue present in the image instead of the denoised image as the input to the loss function is ground truth noise as compared to the original clean image. Both networks achieved better results despite having a simple architecture where repeated blocks of convolutional, batch normalization and ReLU activations are used. Furthermore, IrCNN [60] and DnCNN [59] are dependent on blindly predicted noise *i.e*. without taking into account the underlying structures and textures of the noisy image.

Another essential image restoration framework is Trainable Nonlinear Reaction-Diffusion (TRND) [15] which uses a field-of-experts prior [48] into the deep neural network for a specific number of inference steps by extending the non-linear diffusion paradigm into a profoundly trainable parametrized linear filters and the influence functions. Although the results of TRND are favorable, the model requires a significant amount of data to learn the parameters and influence functions as well as overall fine-tuning, hyper-parameter determination, and stage-wise training. Similarly, non-local color net (NLNet) [36] was motivated by non-local self-similar (NSS) priors which employ non-local self-similarity coupled with discriminative learning. NLNet improved upon the traditional methods; but, it lags in performance compared to most of the CNNs [60, 59] due to the adaptaton of NSS priors, as it is unable to find the analogs for all the patches in the image.

Building upon the success of DnCNN [59], Jiao *et al*. proposed a network consisting of two stacked subnets, named "FormattingNet" and "DiffResNet" respectively. The architecture of both networks is similar, and the difference lies in the loss layers used. The first subnet employs total variational and perceptual loss while the second one uses ℓ_2 loss. The overall model is named as FormResNet and improves upon [60, 59] by a small margin. Lately, Bae *et al*. [9] employed persistent homology analysis [21] via wavelet transformed domain to learn the features in CNN denoising. The performance of the model is marginally better compared to [59, 32], which can be attributed to a large number of feature maps employed rather than the model itself. Recently, Anwar *et al*. introduced CIMM, a deep denoising CNN architecture, composed of identity mapping modules [7]. The network learns features in cascaded identity modules using dilated kernels and uses self-ensemble to boost performance. CIMM improved upon all the previous CNN models [59, 32].

Recently, many algorithms focused on blind denoising on real-noisy images. The algorithms [60, 59, 32] benefitted from the modeling capacity of CNNs and have shown the ability to learn a single-blind denoising model; however, the denoising performance is limited, and the results are not satisfactory on real photographs. Generally speaking, real-noisy image denoising is a two-step process: the first involves noise estimation while the second addresses non-blind denoising. Noise clinic (NC) [35] estimates the noise model dependent on signal and frequency followed by denoising the image using non-local Bayes (NLB). In comparison, Zhang *et al*. [61] proposed a non-blind Gaussian denoising network, termed FFDNet that can produce satisfying results on some of the real noisy images; however, it requires manual intervention to select high noise-level.

Very recently, CBDNet [28] trains a blind denoising model for real photographs. CBDNet [28] is composed of two subnetworks: noise estimation and non-blind denoising. CBDNet [28] also incorporated multiple losses, is engineered to train on real-synthetic noise and real-image noise and enforces a higher noise standard deviation for low noise images. Furthermore, [28, 61] may require manual intervention to improve results. On the other hand, we present an

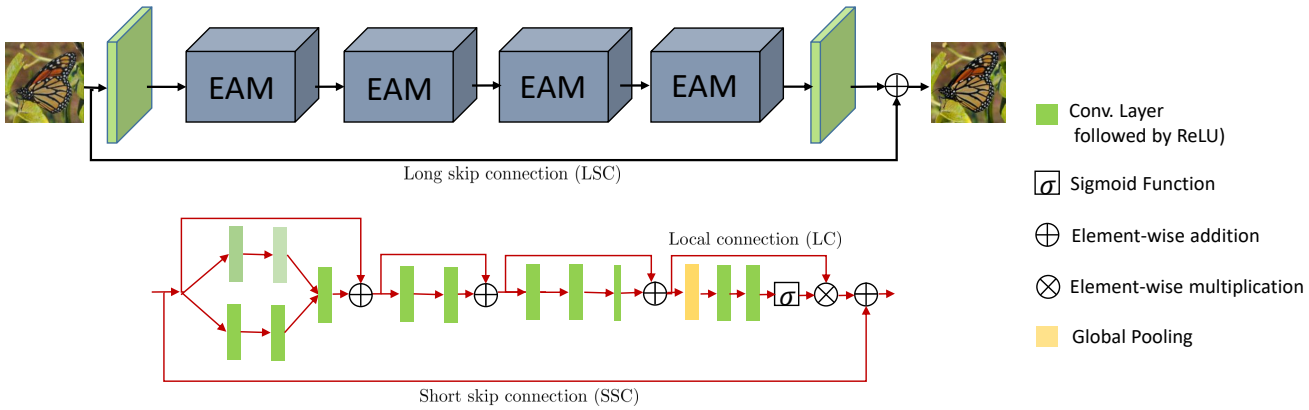

Figure 2. The architecture of the proposed network. Different green colors of the conv layers denote different dilations while the smaller size of the conv layer means the kernel is 1×1. The second row shows the architecture of each EAM.

end-to-end architecture that learns the noise and produces results on real noisy images without requiring separate subnets or manual intervention.

3. CNN Denoiser

3.1. Network Architecture

Our model is composed of three main modules *i.e.* feature extraction, feature learning residual on the residual module, and reconstruction, as shown in Figure 2. Let us consider x is a noisy input image and $\hat{y}$ is the denoised output image. Our feature extraction module is composed of only one convolutional layer to extract initial features f_0 from the noisy input:

$$f_0 = M_e(x), \tag{1}$$

where $M_e(\cdot)$ performs convolution on the noisy input image. Next, f_0 is passed on to the feature learning residual on the residual module, termed as M_{fl},

$$f_r = M_{fl}(f_0), \tag{2}$$

where f_r are the learned features and $M_{fl}(\cdot)$ is the main feature learning residual on the residual component, composed of enhancement attention modules (EAM) that are cascaded together as shown in Figure 2. Our network has small depth, but provides a wide receptive field through kernel dilation in each EAM initial two branch convolutions. The output features of the final layer are fed to the reconstruction module, which is again composed of one convolutional layer.

$$\hat{y} = M_r(f_r), \tag{3}$$

where $M_r(\cdot)$ denotes the reconstruction layer.

There are several choices available as loss function for optimization such as ℓ_2 [59, 60, 7], perceptual loss [32, 28], total variation loss [32] and asymmetric loss [28]. Some networks [32, 28] employs more than one loss to optimize the model, contrary to earlier networks, we only employ one loss *i.e.* ℓ_1. Now, given a batch of N training pairs, $\{x_i, y_i\}_{i=1}^N$, where x is the noisy input and y is the ground truth, the aim is to minimize the ℓ_1 loss function as

$$L(\mathcal{W}) = \frac{1}{N}\sum_{i=1}^{N} ||\text{RIDNet}(x_i) - y_i||_1, \tag{4}$$

where RIDNet$(\cdot)$ is our network and $\mathcal{W}$ denotes the set of all the network parameters learned. Our feature extraction M_e and reconstruction module M_r resemble the previous algorithms [19, 7]. We now focus on the feature learning residual on the residual block, and feature attention.

3.2. Feature learning Residual on the Residual

In this section, we provide more details on the enhancement attention modules that uses a Residual on the Residual structure with local skip and short skip connections. Each EAM is further composed of D blocks followed by feature attention. Due to the residual on the residual architecture, very deep networks are now possible that improve denoising performance; however, we restrict our model to four EAM modules only. The first part of EAM covers the full receptive field of input features, followed by learning on the features; then the features are compressed for speed, and finally a feature attention module enhances the weights of important features from the maps. The first part of EAM is realized using a novel merge-and-run unit as shown in Figure 2 second row. The input features branched and are passed through two dilated convolutions, then concatenated and passed through another convolution. Next, the features are learned using a residual block of two convolutions while compression is achieved by an enhanced residual block (ERB) of three convolutional layers. The last layer of ERB flattens the features by applying a 1×1 kernel. Finally, the output of the feature attention unit is added to the input of EAM.

In image recognition, residual blocks [29] are stacked

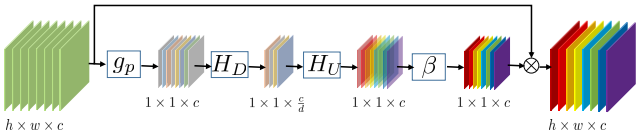

Figure 3. The feature attention mechanism for selecting the essential features.

together to construct a network of more than 1000 layers. Similarly, in image superresolution, EDSR [38] stacked the residual blocks and used long skip connections (LSC) to form a very deep network. However, to date, very deep networks have not been investigated for denoising. Motivated by the success of [62], we introduce the residual on the residual as a basic module for our network to construct deeper systems. Now consider the m-th module of the EAM is given as

$$f_m = EAM_m(EAMm - 1(\cdots(M_0(f_0))\cdots)), \quad (5)$$

where f_m is the output of the EAM_m feature learning module, in other words $f_m = EAM_m(f_{m-1})$. The output of each EAM is added to the input of the group as $f_m = f_m + f_{m-1}$. We have observed that simply cascading the residual modules will not achieve better performance, instead we add the input of the feature extractor module to the final output of the stacked modules as

$$f_g = f_0 + M_{fl}(\mathcal{W}_{w,b}), \quad (6)$$

where $\mathcal{W}_{w,b}$ are the weights and biases learned in the group. This addition *i.e.* LSC, eases the flow of information across groups. f_g is passed to reconstruction layer to output the same number of channels as the input of the network. Furthermore, we add another long skip connection to subtract the input image from the network output *i.e.* $\hat{y} = M_r(f_g) - x$, in order to learn the residual (noise) rather than the denoised image, as this technique helps in faster learning as compared to learning original image due to the sparse representation of the noise.

3.2.1 Feature Attention

This section provides information about the feature attention mechanism. Attention [56] has been around for some time; however, it has not been employed in image denoising. Channel features in image denoising methods are treated equally, which is not appropriate for many cases. To exploit and learn the critical content of the image, we focus attention on the relationship between the channel features; hence the name: feature attention (see Figure 3).

An important question here is how to generate attention differently for each channel-wise feature. Images generally can be considered as having low-frequency regions (smooth or flat areas), and high-frequency regions (*e.g.*, lines edges and texture). As convolutional layers exploit local information only and are unable to utilize global contextual information, we first employ global average pooling to express the statistics denoting the whole image, other options for aggregation of the features can also be explored to represent the image descriptor. Let f_c be the output features of the last convolutional layer having c feature maps of size $h \times w$; global average pooling will reduce the size from $h \times w \times c$ to $1 \times 1 \times c$ as:

$$g_p = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{i=1}^{w} f_c(i, j), \quad (7)$$

where $f_c(i, j)$ is the feature value at position (i, j) in the feature maps.

Furthermore as investigated in [31], we propose a self-gating mechanism to capture the channel dependencies from the descriptor retrieved by global average pooling. According to [31], the mentioned mechanism must learn the nonlinear synergies between channels as well as mutually-exclusive relationships. Here, we employ soft-shrinkage and sigmoid functions to implement the gating mechanism. Let us consider δ, and β are the soft-shrinkage and sigmoid operators, respectively. Then the gating mechanism is

$$r_c = \beta(H_U(\delta(H_D(g_p)))), \quad (8)$$

where H_D and H_U are the channel reduction and channel upsampling operators, respectively. The output of the global pooling layer g_p is convolved with a downsampling Conv layer, activated by the soft-shrinkage function. To differentiate the channel features, the output is then fed into an upsampling Conv layer followed by sigmoid activation. Moreover, to compute the statistics, the output of the sigmoid (r_c) is adaptively rescaled by the input f_c of the channel features as

$$\hat{f}_c = r_c \times f_c \quad (9)$$

3.3. Implementation

Our proposed model contains four EAM blocks. The kernel size for each convolutional layer is set to 3×3, except the last Conv layer in the enhanced residual block and those of the features attention units, where the kernel size is 1×1. Zero padding is used for 3×3 to achieve the same size outputs feature maps. The number of channels for each convolutional layer is fixed at 64, except for feature attention downscaling. A factor of 16 reduces these Conv layers; hence having only four feature maps. The final convolutional layer either outputs three or one feature maps depending on the input. As for running time, our method takes about 0.2 second to process a 512×512 image.

Long skip connection (LSC)		✓		✓				✓	✓
Short skip connection (SSC)			✓	✓			✓	✓	✓
Long connection (LC)						✓	✓		✓
Feature attention (FA)					✓	✓	✓	✓	✓
PSNR (in dB)	28.45	28.77	28.81	28.86	28.52	28.85	28.86	28.90	28.96

Table 1. Investigation of skip connections and feature attention. The best result in PSNR (dB) on values on BSD68 [48] in $\times 10^2$ iterations is presented.

4. Experiments

4.1. Training settings

To generate noisy synthetic images, we employ BSD500 [41], DIV2K [4], and MIT-Adobe FiveK [13], resulting in 4k images while for real noisy images, we use cropped patches of 512×512 from SSID [1], Poly [51], and RENOIR [5]. Data augmentation is performed on training images, which includes random rotations of 90, 180 , 270 and flipping horizontally. In each training batch, 32 patches are extracted as inputs with a size of 80×80. Adam [33] is used as the optimizer with default parameters. The learning rate is initially set to 10^{-4} and then halved after 10^5 iterations. The network is implemented in the Pytorch [44] framework and trained with an Nvidia Tesla V100 GPU. Furthermore, we use PSNR as evaluation metric.

4.2. Ablation Studies

4.2.1 Influence of the skip connections

Skip connections play a crucial role in our network. Here, we demonstrate the effectiveness of the skip connections. Our model is composed of three basic types of connections which includes long skip connection (LSC), short skip connections (SSC), and local connections (LC). Table 1 shows the average PSNR for the BSD68 [48] dataset. The highest performance is obtained when all the skip connections are available while the performance is lower when any connection is absent. We also observed that increasing the depth of the network in the absence of skip connections does not benefit performance.

4.2.2 Feature-attention

Another important aspect of our network is feature attention. Table 1 compares the PSNR values of the networks with and without feature attention. The results support our claim about the benefit of using feature attention. Since the inception of DnCNN [59], the CNN models have matured, and further performance improvement requires the careful design of blocks and rescaling of the feature maps. The two mentioned characteristics are present in our model in the form of feature-attention and the skip connections.

4.3. Comparisons

We evaluate our algorithm using the Peak Signal-to-Noise Ratio (PSNR) index as the error metric and compare against many state-of-the-art competitive algorithms which include traditional methods *i.e*. CBM3D [17], WNNM [27], EPLL [63], CSF [49] and CNN-based denoisers *i.e*. MLP [12], TNRD [15], DnCNN [59], IrCNN [60], CNLNet [36], FFDNet [61] and CBDNet [28]. To be fair in comparison, we use the default setting of the traditional methods provided by the corresponding authors.

4.3.1 Test Datasets

In the experiments, we test four noisy real-world datasets *i.e*. RNI15 [35], DND [46], Nam [42] and SSID [1]. Furthermore, we prepare three synthetic noisy datasets from the widely used 12 classical images, BSD68 [48] color and gray 68 images for testing. We corrupt the clean images by additive white Gaussian noise using noise sigma of 15, 25 and 50 standard deviations.

- RNI15 [35] provides 15 real-world noisy images. Unfortunately, the clean images are not given for this dataset; therefore, only the qualitative comparison is presented for this dataset.
- Nam [42] comprises of 11 static scenes and the corresponding noise-free images obtained by the mean of 500 noisy images of the same scene. The size of the images are enormous; hence, we cropped the images in 512×512 patches and randomly selected 110 from those for testing.
- DnD is recently propsed by Plotz *et al*. [46] which originally contains 50 pairs of real-world noisy and noise-free scenes. The scenes are further cropped into patches of size 512×512 by the providers of the dataset which resulted in 1000 smaller images. The near noise-free images are not publicly available, and the results (PSNR/SSIM) can only be obtained through the online system introduced by [46].
- SSID [1] (Smartphone Image Denoising Dataset) is recently introduced. The authors have collected 30k real noisy images and their corresponding clean images; however, only 320 images are released for training and 1280 images pairs for validation, as testing images are

Noise Level	Methods									
	BM3D	WNNM	EPLL	TNRD	DenoiseNet	DnCNN	IrCNN	NLNet	FFDNet	Ours
15	31.08	31.32	31.19	31.42	31.44	31.73	31.63	31.52	31.63	**31.81**
25	28.57	28.83	28.68	28.92	29.04	29.23	29.15	29.03	29.23	**29.34**
50	25.62	25.83	25.67	26.01	26.06	26.23	26.19	26.07	26.29	**26.40**

Table 2. The similarity between the denoised and the clean images of BSD68 dataset [48] for our method and competing measured in terms of average PSNR for σ=15, 25, and 50 on grayscale images.

Methods	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
BM3D [16]	32.37	29.97	26.72
WNNM [27]	32.70	30.26	27.05
EPLL [63]	32.14	29.69	26.47
MLP [12]	-	30.03	26.78
CSF [49]	32.32	29.84	-
TNRD [15]	32.50	30.06	26.81
DnCNN [59]	32.86	30.44	27.18
IrCNN [60]	32.77	30.38	27.14
FFDNet [61]	32.75	30.43	27.32
Ours	**32.91**	**30.60**	**27.43**

Table 3. The quantitative comparison between denoising algorithms on 12 classical images, (in terms of PSNR). The best results are highlighted as bold.

not released yet. We will use the validation images for testing our algorithm and the competitive methods.

4.3.2 Grayscale noisy images

In this subsection, we evaluate our model on the noisy grayscale images corrupted by spatially invariant additive white Gaussian noise. We compare against nonlocal self-similarity representative models *i.e*. BM3D [16] and WNNM [27], learning based methods *i.e*. EPLL, TNRD [15], MLP [12], DnCNN [59], IrCNN [60], and CSF [49]. In Tables 3 and 2, we present the PSNR values on Set12 and BSD68. It is to be remembered here that BSD500 [41] and BSD68 [48] are two disjoint sets. Our method outperforms all the competitive algorithms on both datasets for all noise levels; this may be due to the larger receptive field as well as better modeling capacity.

4.3.3 Color noisy images

Next, for noisy color image denoising, we keep all the parameters of the network similar to the grayscale model, except the first and last layer are changed to input and output three channels rather than one. Figure 4 presents the visual comparison and Table 4 reports the PSNR numbers between our methods and the alternative algorithms. Our algorithm consistently outperforms all the other techniques published in Table 4 for CBSD68 dataset [48]. Similarly, our network produces the best perceptual quality images as shown in Figure 4. A closer inspection on the vase reveals that our

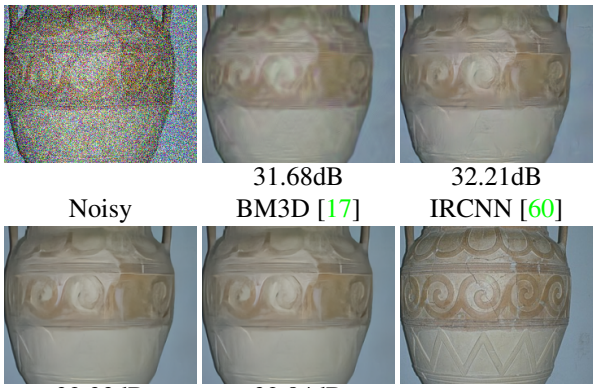

Figure 4. Denoising performance of our RIDNet versus state-of-the-art methods on a color images from [48] for $\sigma_n = 50$

network generates textures closest to the ground-truth with fewer artifacts and more details.

4.3.4 Real-World noisy images

To further assess the practicality of our model, we employ a real noise dataset. The evaluation is difficult because of the unknown level of noise, the various noise sources such as shot noise, quantization noise *etc*., imaging pipeline *i.e*. image resizing, lossy compression *etc*. Furthermore, the noise is spatially variant (non-Gaussian) and also signal dependent; hence, the assumption that noise is spatially invariant, employed by many algorithms does not hold for real image noise. Therefore, real-noisy images evaluation determines the success of the algorithms in real-world applications.

DnD: Table 5 presents the quantitative results (PSNR/SSIM) on the sRGB data for competitive algorithms and our method obtained from the online DnD benchmark website available publicly. The blind Gaussian denoiser DnCNN [59] performs inefficiently and is unable to achieve better results than BM3D [16] and WNNM [27] due to the poor generalization of the noise during training. Similarly, the non-blind Gaussian traditional denoisers are able to report limited performance, although the noise standard-deviation is provided. This may be due to the fact that these denoisers [16, 27, 63] are tailored for AWGN only and real-noise is different in characteristics to syn-

Noise Levels	Methods							
	CBM3D [17]	MLP [12]	TNRD [15]	DnCNN [59]	IrCNN [60]	CNLNet [36]	FFDNet [61]	Ours
15	33.50	-	31.37	33.89	33.86	33.69	33.87	**34.01**
25	30.69	28.92	28.88	31.33	31.16	30.96	31.21	**31.37**
50	27.37	26.00	25.94	27.97	27.86	27.64	27.96	**28.14**

Table 4. Performance comparison between our network and existing state-of-the-art algorithms on the color version of the BSD68 dataset [48].

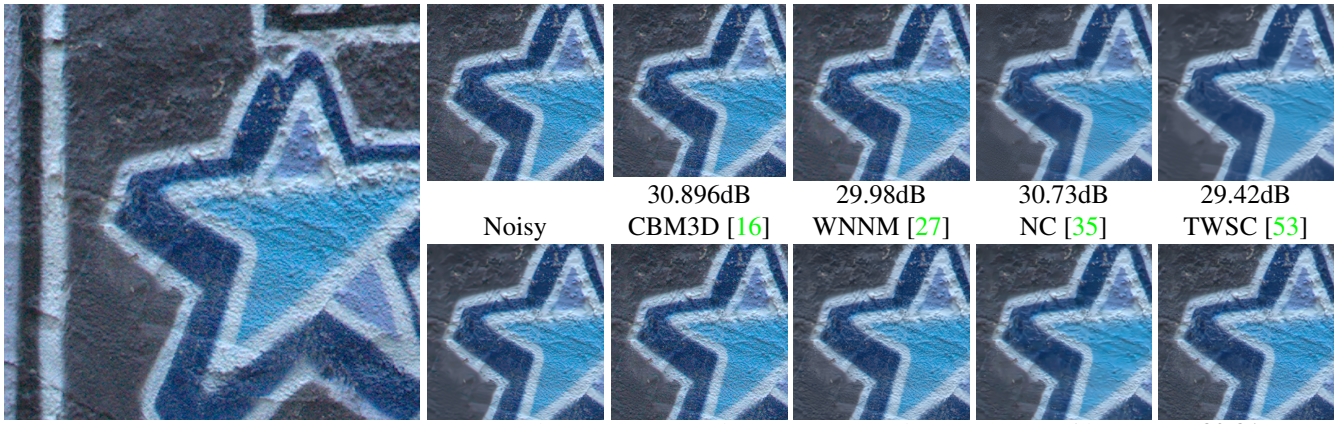

Figure 5. A real noisy example from DND dataset [46] for comparison of our method against the state-of-the-art algorithms.

Method	Blind/Non-blind	PSNR	SSIM
CDnCNN-B	Blind	32.43	0.7900
EPLL	Non-blind	33.51	0.8244
TNRD	Non-blind	33.65	0.8306
NCSR	Non-blind	34.05	0.8351
MLP	Non-blind	34.23	0.8331
FFDNet	Non-blind	34.40	0.8474
BM3D	Non-blind	34.51	0.8507
FoE	Non-blind	34.62	0.8845
WNNM	Non-blind	34.67	0.8646
NC	Blind	35.43	0.8841
NI	Blind	35.11	0.8778
CIMM	Non-blind	36.04	0.9136
KSVD	Non-blind	36.49	0.8978
MCWNNM	Non-blind	37.38	0.9294
TWSC	Non-blind	37.96	0.9416
FFDNet+	Non-blind	37.61	0.9415
CBDNet	Blind	38.06	0.9421
Ours	Blind	**39.23**	**0.9526**

Table 5. The Mean PSNR and SSIM denoising results of state-of-the-art algorithms evaluated on the DnD sRGB images [46]

thetic noise. Incorporating feature attention and capturing the appropriate characteristics of the noise through a novel module means our algorithm leads by large margin *i.e.* 1.17dB PSNR compared to the second performing method, CBDNet [28]. Furthermore, our algorithm only employs real-noisy images for training using only ℓ_1 loss while CBDNet [28] uses many techniques such as multiple losses

Figure 6. Comparison of our method against the other methods on a real image from RNI15 [35] benchmark containing spatially variant noise.

(*i.e.* total variation, ℓ_2 and asymmetric learning) and both real-noise as well as synthetically generated real-noise. As reported by the author of CBDNet [28], it is able to achieve 37.72 dB with real-noise images only. Noise Clinic (NC) [35] and Neat Image (NI) [2] are the other two state-of-the-art blind denoisers other than [28]. NI [2] is commercially available as a part of Photoshop and Corel PaintShop. Our network is able to achieve 3.82dB and 4.14dB more PSNR from NC [35] and NI [2], respectively.

Next, we visually compare the result of our method with the competing methods on the denoised images provided by the online system of Plotz *et al.* [46] in Figure 5. The PSNR and SSIM values are also taken from the website. From Figure 5, it is clear that the methods of [28, 61, 59] perform poorly in removing the noise from the star and in some cases the image is over-smoothed, on the other hand, our algorithm can eliminate the noise while preserving the finer details and structures in the star image.

Figure 7. A real high noise example from RNI15 dataset [35]. Our method is able to remove the noise in textured and smooth areas without introducing artifacts.

	Methods				
Datasets	BM3D	DnCNN	FFDNet	CBDNet	Ours
Nam [42]	37.30	35.55	38.7	39.01	**39.09**
SSID [1]	30.88	26.21	29.20	30.78	**38.71**

Table 6. The quantitative results (in PSNR (dB)) for the SSID [1] and Nam [42] datasets.

RNI15: On RNI15 [35], we provide qualitative images only as the ground-truth images are not available. Figure 6 presents the denoising results on a low noise intensity image. FFDNet [61] and CBDNet [28] are unable to remove the noise in its totality as can been seen near the bottom left of handle and body of the cup image. On the contrary, our method is able to remove the noise without the introduction of any artifacts. We present another example from the RNI15 dataset [35] with high noise in Figure 7. CDnCNN [59] and FFDNet [61] produce results of limited nature as some noisy elements can be seen in the near the eye and gloves of the Dog image. In comparison, our algorithm recovers the actual texture and structures without compromising on the removal of noise from the images.

Nam: We present the average PSNR scores of the resultant denoised images in Table 6. Unlike CBDNet [28], which is trained on Nam [42] to specifically deal with the JPEG compression, we use the same network to denoise the Nam images [42] and achieve favorable PSNR numbers. Our performance in terms of PSNR is higher than any of the current state-of-the-art algorithms. Furthermore, our claim is supported by the visual quality of the images produced by our model as shown in Figure 8. The amount of noise present after denoising by our method is negligible as compared to CDnCNN and other counterparts.

SSID: As a last dataset, we employ the SSID real noise dataset which has the highest number of test (validation) images available. The results in terms of PSNR are shown in the second row of Table 6. Again, it is clear that our method outperforms FFDNet [61] and CBDNet [28] by a margin of 9.5dB and 7.93dB, respectively. In Figure 9, we show the denoised results of a challenging image by different algorithms. Our technique recovers the true colors which are closer to the original pixel values while competing methods

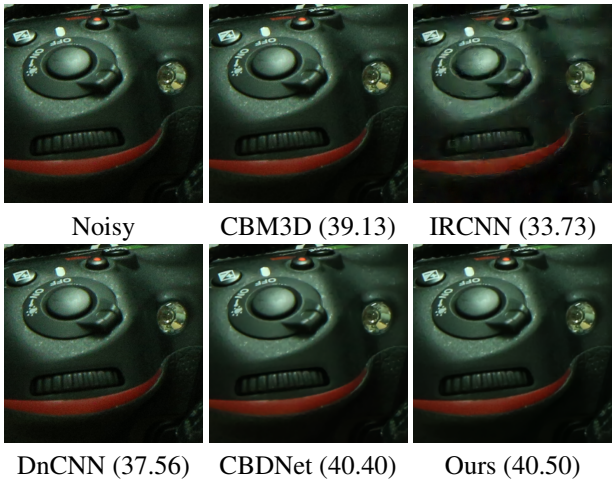

Figure 8. An image from Nam dataset [42] with JPEG compression. CBDNet is trained explicitly on JPEG compressed images; still, our method performed better.

Figure 9. A challenging example from SSID dataset [1]. Our method can remove noise and restore true colors.

are unable to restore original colors and in specific regions induce false colors.

5. Conclusion

In this paper, we present a new CNN denoising model for synthetic noise and real noisy photographs. Unlike previous algorithms, our model is a single-blind denoising network for real noisy images. We propose a novel restoration module to learn the features and to enhance the capability of the network further; we adopt feature attention to rescale the channel-wise features by taking into account the dependencies between the channels. We also use LSC, SSC, and SC to allow low-frequency information to bypass so the network can focus on residual learning. Extensive experiments on three synthetic and four real-noise datasets demonstrate the effectiveness of our proposed model.

References

[1] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018. 5, 8

[2] ABSoft. Neat image. 7

[3] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE transactions on image processing*, 19(9):2345–2356, 2010. 1

[4] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 5

[5] J. Anaya and A. Barbu. Renoir–a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018. 5

[6] S. Anwar, C. Huynh, and F. Porikli. Combined internal and external category-specific image denoising. In *British Machine Vision Conference*, 2017. 2

[7] S. Anwar, C. P. Huynh, and F. Porikli. Chaining identity mapping modules for image denoising. *arXiv preprint arXiv:1712.02933*, 2017. 2, 3

[8] S. Anwar, F. Porikli, and C. P. Huynh. Category-specific object image denoising. *TIP*, pages 5506–5518, 2017. 1, 2

[9] W. Bae, J. Yoo, and J. C. Ye. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *CVPR Workshops*, pages 145–153, 2017. 2

[10] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *TNN*, 1994. 2

[11] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, pages 60–65, 2005. 1, 2

[12] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, pages 2392–2399, 2012. 1, 2, 5, 6, 7

[13] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011*, pages 97–104. IEEE, 2011. 5

[14] P. Chatterjee and P. Milanfar. Is denoising dead? *TIP*, pages 895–911, 2010. 2

[15] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *TPAMI*, pages 1256–1272, 2017. 1, 2, 5, 6, 7

[16] K. Dabov, A. F., V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. pages 2080–2095, 2007. 1, 2, 6, 7

[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color image denoising via sparse 3-D collaborative filtering with grouping constraint in luminance-chrominance space. In *ICIP*, 2007. 5, 6, 7

[18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. BM3D image denoising with shape-adaptive principal component analysis. In *Signal Processing with Adaptive Sparse Structured Representations*, 2009. 1, 2

[19] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016. 2, 3

[20] W. Dong, X. Li, D. Zhang, and G. Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *CVPR*, pages 457–464, 2011. 2

[21] H. Edelsbrunner and J. Harer. Persistent homology-a survey. *Contemporary mathematics*, pages 257–282, 2008. 2

[22] M. Elad and D. Datsenko. Example-based regularization deployed to super-resolution reconstruction of a single image. *Comput. J.*, pages 15–30, 2009. 2

[23] L. Z. F. Chen and H. Yu. External Patch Prior Guided Internal Clustering for Image Denoising. In *ICCV*, pages 1211–1218, 2015. 2

[24] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *TIP*, pages 1395–1411, 2007. 2

[25] R. C. Gonzalez and P. Wintz. Digital image processing(book). *Reading, Mass., Addison-Wesley Publishing Co., Inc.(Applied Mathematics and Computation*, (13):451, 1977. 1

[26] B. Goossens, H. Luong, A. Pizurica, and W. Philips. An improved non-local denoising algorithm. In *IP*, page 143, 2008. 2

[27] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, pages 2862–2869, 2014. 1, 5, 6, 7

[28] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang. Toward convolutional blind denoising of real photographs. *arXiv preprint arXiv:1807.04686*, 2018. 1, 2, 3, 5, 7, 8

[29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[30] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, et al. Flexisp: A flexible camera image processing framework. *ACM Transactions on Graphics (TOG)*, 33(6):231, 2014. 1

[31] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 4

[32] J. Jiao, W.-C. Tu, S. He, and R. W. Lau. Formresnet: Formatted residual learning for image restoration. In *CVPR Workshops*, pages 1034–1042, 2017. 2, 3

[33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[34] M. Lebrun, A. Buades, and J.-M. Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, pages 1665–1688, 2013. 2

[35] M. Lebrun, M. Colom, and J.-M. Morel. The noise clinic: a blind image denoising algorithm. *Image Processing On Line*, 5:1–54, 2015. 1, 2, 5, 7, 8

[36] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. *CVPR*, 2016. 2, 5, 7

[37] A. Levin and B. Nadler. Natural image denoising: Optimality and inherent bounds. In *CVPR*, pages 2833–2840, 2011. 2

[38] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *Computer vision and pattern recognition workshops*, 2017. 2, 4

[39] E. Luo, S. H. Chan, and T. Q. Nguyen. Adaptive image denoising by targeted databases. *TIP*, pages 2167–2181, 2015. 1, 2

[40] 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

[41] 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. 5, 6

[42] S. Nam, Y. Hwang, Y. Matsushita, and S. Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1683–1691, 2016. 5, 8

[43] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, pages 460–489, 2005. 1

[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 5

[45] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *TPAMI*, pages 2233–2246, 2012. 1

[46] T. Plötz and S. Roth. Benchmarking denoising algorithms with real photographs. *arXiv preprint arXiv:1707.01313*, 2017. 5, 7

[47] Y. Romano, M. Elad, and P. Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017. 1

[48] S. Roth and M. J. Black. Fields of experts. *IJCV*, pages 205–229, 2009. 1, 2, 5, 6, 7

[49] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *CVPR*, pages 2774–2781, 2014. 2, 5, 6

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

[51] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018. 5

[52] J. Xu and S. Osher. Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoising. *TIP*, pages 534–544, 2007. 1

[53] J. Xu, L. Zhang, and D. Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–36, 2018. 7

[54] J. Xu, L. Zhang, D. Zhang, and X. Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1096–1104, 2017. 7

[55] 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 1211–1218, 2015. 2

[56] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 4

[57] H. Yue, X. Sun, J. Yang, and F. Wu. Cid: Combined image denoising in spatial and frequency domains using web images. In *CVPR*, pages 2933–2940, June 2014. 1

[58] H. Yue, X. Sun, J. Yang, and F. Wu. Image denoising by exploring external and internal correlations. *TIP*, pages 1967–1982, 2015. 2

[59] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 1, 2, 3, 5, 6, 7, 8

[60] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. *CVPR*, 2017. 1, 2, 3, 5, 6, 7

[61] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 1, 2, 3, 5, 6, 7, 8

[62] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. *arXiv preprint arXiv:1807.02758*, 2018. 4

[63] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486, 2011. 1, 2, 5, 6