Identity Enhanced Residual Image Denoising

Saeed Anwar[1,2], Cong Phuoc Huynh[2], Fatih Porikli[2]
[1]Data61-CSIRO, [2]The Australian Natinal University
saeed.anwar@data61.csiro.au

Abstract

We propose to learn a fully-convolutional network model that consists of a Chain of Identity Mapping Modules and residual on the residual architecture for image denoising. Our network structure possesses three distinctive features that are important for the noise removal task. Firstly, each unit employs identity mappings as the skip connections and receives pre-activated input to preserve the gradient magnitude propagated in both the forward and backward directions. Secondly, by utilizing dilated kernels for the convolution layers in the residual branch, each neuron in the last convolution layer of each module can observe the full receptive field of the first layer. Lastly, we employ the residual on the residual architecture to ease the propagation of the high-level information. Contrary to current state-of-the-art real denoising networks, we also present a straightforward and single-stage network for real image denoising.

The proposed network produces remarkably higher numerical accuracy and better visual image quality than the classical state-of-the-art and CNN algorithms when being evaluated on the three conventional benchmark and three real-world datasets.

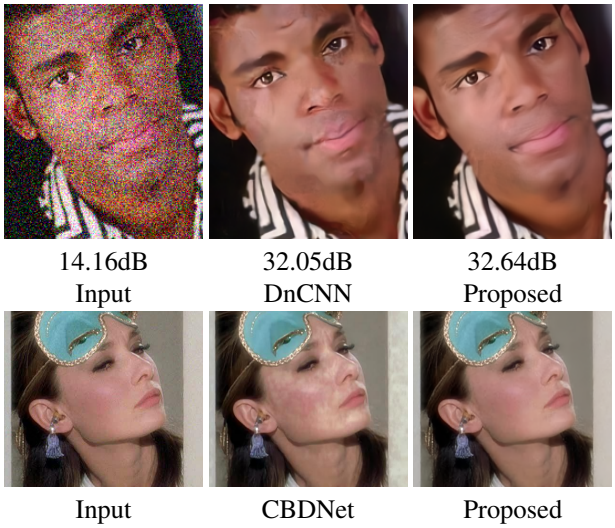

Figure 1. Denoising results: In the first row, an image corrupted by the Gaussian noise with $\sigma = 50$ from the BSD68 dataset ([42]). In the second row, a sample image from RNI15 ([32]) real noisy dataset. Our results have the best PSNR score for synthetic images, and unlike other methods, it does not have over-smoothing or over-contrasting artifacts. Best viewed in color on a high-resolution display.

1. Introduction

In recent years, the amount of multimedia content is growing at an enormous rate, for example, online videos, audios, and photos due to hand-held devices and other types of multimedia devices. Thus, image processing, specifically, image denoising, has become an essential process for various computer vision and image analysis applications. A few notable methods benefiting from image denoising are detection ([43]), face recognition ([28]), super-resolution ([53]), *etc*. In the past few years, the research in this area has shifted its focus on how to make the best use of image priors. To this end, several approaches attempted to exploit non-local self similar (NSS) patterns ([9, 15]), sparse models ([22, 39]), gradient models ([48, 46]), Markov random field models ([42]), external denoising ([54, 6, 36]) and convolutional neural networks ([56, 33, 57]).

The non-local matching (NLM) of self-similar patches and block matching with 3D filtering (BM3D) in a collaborative manner have been two prominent baselines for image denoising for almost a decade now. Due to popularity of NLM ([9]) and BM3D ([15]), a number of their variants ([20, 31, 21]) were also proposed to execute the search for similar patches in similar transform domains.

The use of external priors for denoising has been motivated by the pioneering studies of [34, 12], which showed that selecting correct reference patches from a large external image dataset of clean samples can theoretically suppress additive noise and attain infinitesimal reconstruction error. However, directly incorporating patches from an external database is computationally prohibitive even for a single image. To overcome this problem, Chan *et al*. [11] pro-

Code available at https://github.com/saeed-anwar/IERD

posed efficient sampling techniques for large databases but still the denoising is impractical as it takes hours to search patches for one single image if not days. An alternative to these methods can be considered as the dictionary learning based approaches [18, 37, 16], which learn over-complete dictionaries from a set of external natural clean images and then enforce patch self-similarity through sparsity.

Aiming at improving the use of external datasets, many previous works such as [59, 19, 51] investigated the use of maximum likelihood frameworks to learn Gaussian mixture models of natural image patches or group patches for clean patch estimation. Several studies, including [52, 13], modified Zoran *et al*. [59]'s statistical prior for reconstruction of class-specific noisy images by capturing the statistics of noise-free patches from a large database of same category images through the Expectation-Maximization algorithm. Other similar methods on external denoising include TID [36], CSID [6] and CID [55]; however, all of these have limited applicability in denoising of generic (from an unspecific class) images.

As an alternative, CSF [44] learns a single framework based on unification of random-field based model and half-quadratic optimization. The role of the shrinkage in wavelet image restoration is to attenuate small values towards zero due to the assumption of these values being the product of noise instead of the signal values.These predictions are then chained to form a cascade of shrinkage fields of Gaussian conditional random Fields. The CSF algorithm considers the data term to be quadratic and must have a closed-form solution based on discrete Fourier transform.

With the rise of convolutional neural networks (CNN), a significant performance boost for image denoising has been achieved [56, 57, 33, 10, 44]. Using deep neural networks, IrCNN [57] and DnCNN [56] learn to predict the residual noise present in the contaminated image by using the ground-truth noise in the loss function instead of the clean image. The architectures of IrCNN [57] and DnCNN [56] are very simple as it only stacks of convolutional, batch normalization and ReLU layers. Although both models were able to report favorable results, their performance depends heavily on the accuracy of noise estimation without knowing the underlying structures and textures present in the image.

TRND [14] incorporated a field-of-experts prior [42] into its convolutional network by extending conventional nonlinear diffusion model to highly trainable parametrized linear filters and influence functions. It has shown improved results over more classical methods; however, the imposed image priors inherently impede its performance, which highly rely on the choice of hyper-parameter settings, extensive fine-tuning and stage-wise training.

Another notable deep learning-based work is non-local color image denoising (abbreviated as NLNet), presented by [33] which exploits the non-local self-similarity using deep networks. Non-local variational schemes have motivated the design of the NLNet model [33] and employ the non-local self-similarity property of natural images for denoising. The performance heavily depends on coupling discriminative learning and self-similarity. The restoration performance is comparatively better to several earlier state-of-the-art. Though, this model improves on classical methods but lagging behind IrCNN [57] and DnCNN [56], as it inherits the limitations associated with the NSS priors as not all patches recur in an image.

Currently, the trend changed from synthetic denoising towards real-image denoising ([41, 23, 8, 5]). Although, the algorithms, for example, DnCNN, *etc*. trained a single model for synthetic datasets; however, it failed to achieve satisfactory results on real images. Commonly, real-image denoising is a two-stage process. The first step involves the prediction of the noise variance, while the second stage employs the predicted noise-level to denoise the image. As an example, Noise Clinic proposed (NC) by [32] first predicts the noise, which is dependent on the signal's frequency and then used non-local Bayes (NLB) ([31]) to denoise it.

Similarly, [58] trains FFDNet, a non-blind denoising network based on Gaussian noise. The mentioned network achieves partial success in denoising the real noisy images. However, FFDNet requires manual settings in case of high noise variance. More recently, [23] proposed CBDNet, a blind network for real-noisy images. The system is composed of two subnets: one for prediction of noise and the second to denoise photographs using the predicted noise. Furthermore, CBDNet uses multiple losses and exploits synthetic and real images alternatively to train the model. The authors also report the use of high noise variance to denoise a low noisy image. Moreover, to improve results, the system may require manual intervention. More recently, Anwar & Barnes presented denoising real images via attention mechanism, known as RIDNet [5]. The modules are carefully designed to learn features differently. In this work we present a straightforward end-to-end structure that delivers results on real noisy images using a single-stage network without requiring any intervention or attention mechanism.

1.1. Inspiration & Motivation

Existing convolutional neural network image denoising methods ([10, 56, 57]) connect weight layers consecutively and learn the mapping by brute force. One problem with such an architecture is the addition of more weight layers to increase the depth of the network. Even if the new weight layers are added to the above mentioned CNN based denoising methods, it will suffer from the vanishing gradients problem and make it worse ([7]). This increase in the depth of the network is essential to attain the performance boost ([26]). Therefore, our goal is to propose a model that

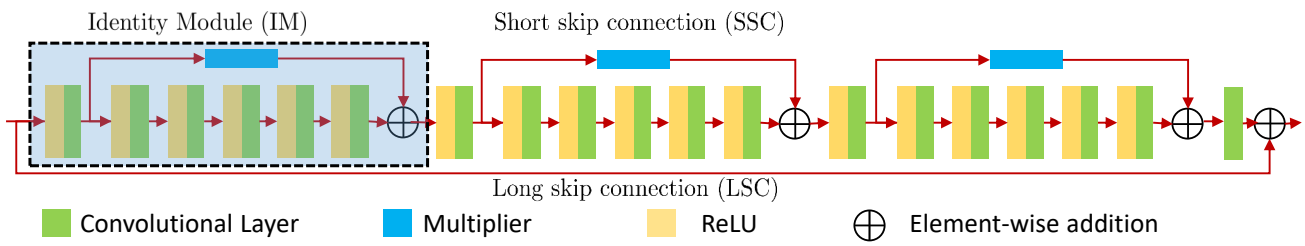

Figure 2. The proposed network architecture, which consists of multiple modules with similar structures. Each module is composed of a series of pre-activation-convolution layer pairs. The multiplier block negates the input block features to be summed at the end of the mapping module.

overcomes this deficiency. Another reason is the lack of single-stage real image denoising. Most of the current denoising systems are either for synthetic image denoising or treat noise estimation and denoising separately, ignoring the relationship between the noise and the image structures.

To provide a solution, our choice is the convolutional neural networks in a discriminative prior setting for image denoising. There are many advantages of using single-stage CNNs for synthetic and real images, which include efficient inference, incorporation of robust priors, integration of local and global receptive fields, regressing on nonlinear models, and discriminative learning capability. Furthermore, we propose a modular single-stage network where we call each module as a identity module (IM). The identity module can be replicated and easily extended to any arbitrary depth for performance enhancement.

1.2. Contributions

The contributions of this work can be summarized as follows:

- An effective CNN architecture that consists of a Chain of Identity Mapping modules for image denoising. These modules share a common composition of layers, with residual connections between them to facilitate training stability.
- The use of dilated convolutions for learning suitable filters to denoise at different levels of spatial extent and residual on the residual architecture for the ease of flow of the high-frequency details.
- A low-weight single-stage real image denoiser without any complex modules.
- Extensive evaluation on six datasets (three synthetic and three real) against more than 20 state-of-the-art denoising methods.

2. Identity Enhanced Residual Denoising

This section presents our approach to image denoising by learning a Convolutional Neural Network consisting of a series of Identity Mapping Modules. Each module is composed of a series of pre-activation units followed by convolution functions, with residual connections between them. The meta-structure of our Identity Enhanced Residual Denoising (IERD) network is explained in Section 2.1 followed by the formulation of the learning objective in Section 2.2.

2.1. Network Design

Residual learning has recently delivered state of the art results for object classification ([24, 27]) and detection ([35]), while offers training stability. Inspired by the Residual Network variant with identity mapping ([27]), we adopt a modular design for our denoising network. The design consists of a series of Identity Mapping modules.

2.1.1 Network Elements

Figure 2 depicts the entire architecture, where identity mapping modules are shown as blue blocks, which are, in turn, composed of basic ReLU and convolution layers. The output of each module is a summation of the identity function and the residual function.

Three parameters govern the meta-level structure of the network: $\mathbf{M}$ is the number of identity modules, $\mathbf{L}$ is the number of pairs of pre-activation and convolution layers in each module, and $\mathbf{C}$ is the number of output channels, which we fixed across all the convolution layers.

The high-level structure of the network can be viewed as a chain of identity modules, where the output of each module is fed directly into the succeeding one. Consequently, the output of this chain is fed to a final convolution layer to produce a tensor with the same number of channels as the input image. At this point, the final convolution layer directly predicts the noise component from a noisy image. The noisy image/patch is then added to the input to recover the noise-free image.

The identity mapping modules are the building blocks of the network, which share the following structure. Each module consists of two branches: a residual branch and an

identity mapping branch. The residual branch of each module contains a series of layers pairs, *i.e.* a nonlinear pre-activation (typically ReLU) layer, followed by a convolution layer. Its primary responsibility is to learn a set of convolution filters to predict image noise. Besides, the identity mapping branch in each module allows the propagation of loss gradients in both directions without any bottleneck.

2.1.2 Justification of the network design

Several previous image denoising works have adopted a fully convolutional network design, without any pooling mechanism ([56, 29]). This is necessary in order to preserve the spatial resolution of the input tensor across different layers. We follow this design by using only non-linear activations and convolution layers across our network.

Furthermore, we aim to design the network in such a way where convolution layers neurons in the last layer of each identity mapping (IM) module observe the full spatial receptive field in the first convolution layer. This design helps to learn to connect input neurons at all spatial locations to the output neurons, in much the same way as well-known non-local mean methods such as ([15, 9]). Instead of using a unit dilation stride within each layer, we also experimented with dilated convolutions to increase the receptive fields of the convolution layers. By this design, we can reduce the depth of each IM module while the final layer's neurons can still observe the full input spatial extent.

Pre-activation has been shown to offer the highest performance for classification when used together with identity mapping ([27]). In a similar fashion, our design employs ReLU before each convolution layer. This design differs from existing neural network architectures for denosing ([29, 33]). The pre-activation helps training to converge more easily, while the identity function preserves the range of gradient magnitudes. Also, the resulting network generalizes better as compared to the post-activation alternative. This property enhances the denoising ability of our network.

2.1.3 Formulation

Now we formulate the prediction output of this network structure for a given input patch $\mathbf{y}$. Let $\mathcal{W}$ denote the set of all the network parameters, which consists of the weights and biases of all constituting convolution layers. Specifically, we let $w_{m,l}$ denote both the kernel and bias parameters of the l-th convolution layer in the residual branch of the m-th module.

Within such a branch, the intermediate output of the l-th ReLU-convolution pair and of the m-th module is a composition of two functions

$$\mathbf{z}_{m,l} = f(g(\mathbf{y}_{m,l}); w_{m,l}), \tag{1}$$

where f and g are the notation for the convolution and the ReLU functions, $\mathbf{z}_{m,l}$ is the output of the l-th ReLU-convolution pair of m-th module. By composing the series of ReLU-convolution pairs, we obtain the output of the m-th residual branch as

$$\mathbf{r}_m = -\mathbf{z}_{m,0} + f(g(\ldots f(g(\mathbf{y}_{m,0}; w_{m,0}))\ldots); w_{m,l}), \tag{2}$$

where $\mathbf{z}_{m,0}$ is the output of the first ReLU-convolution pair, and $\mathbf{r}_m$ is the residual output of the corresponding module. Chaining all the identity mapping modules, we obtain the output as $\sum_{m=1}^{M} \mathbf{r}_m$. Finally, the output of this chain is convolved with a final convolution layer with learnable parameters w_{m+1} to predict the noise component as $\text{IERD}(\mathbf{y}, \mathcal{W}) = f(\mathbf{y} + \sum_{m=1}^{M} \mathbf{r}_m, w_{m+1})$.

2.2. Learning to Denoise

Our network is trained on image patches or regions rather than at the entire image. A number of reasons drive this decision. Firstly, it offers a random sampling of a large number of training samples at different locations from various images. The random shuffling of training samples is well-known to be a useful technique to stabilize the training of deep neural networks. Therefore, it is preferable to batch training patches with a random, diverse mixture of local structures, patterns, shapes, and colors. Secondly, there has been a success in approaches that learns image patch priors from external data for image denoising ([59]).

From a set of noise-free training images, we randomly crop several training patches $\mathbf{x}_i, i = i, \ldots, N$ as the ground-truth. The noisy version of these patches is obtained by adding (Gaussian) noise to the ground truth training images. Let us denote the set of noisy patches corresponding to the former as $\mathbf{y}_i, i = i, \ldots, N$. With this setup, our image denoising network is aimed to reconstruct a patch $\mathbf{x}_i^* = \text{IERD}(\mathbf{y}_i, \mathcal{W})$ from the input patch $\mathbf{y}_i$.

The learning objective is to minimize the following sum of squares of ℓ_2-norms

$$\mathcal{L} \triangleq \frac{1}{N} \sum_{i=1}^{N} \|\text{IERD}(\mathbf{y}_i, \mathcal{W}) - \mathbf{x}_i\|^2. \tag{3}$$

To train the proposed network on a large dataset, we minimize the objective function in Equation 3 on mini-batches of training examples. Training details for our experiments are described in Section 3.2.

3. Experiments

3.1. Datasets

We performed experimental validation on three widely used publicly available synthetically generated noisy datasets (in supplementary materials) and three real noisy image datasets described below.

Table 1. Detailed architecture of an identity mapping module.

Parameters	Identity Module Layers					
	1st	2nd	3rd	4th	5th	6th
Padding	1	3	3	3	3	3
Dilation	1	3	3	3	3	3
Kernel Size	3	3	3	3	3	3
Channels	64	64	64	64	64	64

- **DnD:** Recently, [40] proposed the Darmstadt Noise Dataset (DND) to benchmark the denoising algorithms. The dataset is composed of images with interesting and challenging structures. The size of each image is in Megapixels; therefore, each image is cropped at 20 locations of size 512 $\times$ 512 pixels yielding 1000 test crops. Only these test images are provided; there are no images for training or validation.

- **RNI15:** RNI15 proposed by [32] consists of 15 real noisy images. There are no ground-truth images available for this dataset.

- **SIDD:** Smartphone Image Denoising Dataset (SIDD) proposed by [1] is the largest collection of real-noisy images. A total of 30k noisy images are gathered from ten different scenes under different lighting conditions via five smartphone cameras with their ground truth images.

For evaluation purposes, we use the Peak Signal-to-Noise Ratio (PSNR) index as the error metric. We compare our proposed method with around 20+ state-of-the-art methods on the above six datasets. To ensure a fair comparison, we use the default setting provided by the respective authors.

3.2. Training Details

The training input to our network is noisy, and noise-free patch pairs cropped randomly from the BSD400 dataset ([38]) for synthetic denoising while for real noisy images, we use cropped patches from SSID ([1]), Poly([47]), and RENOIR ([4]). Note that there is no overlap between the training and evaluation datasets. We also augment the training data with horizontally and vertically flipped versions of the original patches and those rotated at an angle of $\frac{\pi n}{2}$, where $n = 1, 2, 3$. The training patches are randomly cropped on the fly from the images of the mentioned datasets.

We offer two strategies for handling different noise levels. The first one is to train a network for each specific noise level, and we call this model as "noise-specific" model. Alternatively, we train a single model for the any noise, and we refer to this model as a "noise-agnostic" model. At each update of training, we construct a batch of 32 by randomly selecting noisy patches with different noise levels.

Table 2. Denoising performance (in PSNR) on the BSD68 dataset ([38]) for different sizes of training input patches for $\sigma_n = 25$, keeping all other parameters constant.

Training patch size					
20	30	40	50	60	70
29.13	29.30	29.34	29.36	29.37	29.38

We implement the denoising method in the PyTorch framework on two Tesla P100 GPUs and employ [30]'s Adam optimization algorithm for training. The initial learning rate was set to 10^{-4}, and the momentum parameter was 0.9. We scheduled the learning rate such that it is halved after every 10^5 iterations. We train our network from scratch by a random initialization of the convolution weights according to the method in [25] and a regularization strength, *i.e.* weight decay, of 10^{-3}.

3.3. Boosting Denoising Performance

To boost the performance of the trained model, we use the late fusion/geometric transform strategy as adopted by [45]. During the evaluation, we perform eight types of augmentation (including identity) of the input noisy images y as $y_i^t = \Gamma_i(y)$ where $i = 1, \cdots, 8$. From these geometrically transformed images, we estimate corresponding denoised images $\{\hat{x}_1^t, \hat{x}_2^t, \cdots, \hat{x}_8^t\}$, where $\hat{x}_i^t = \text{IERD}(\hat{y}_i^t, W)$ using our model. To generate the final denoised image $\hat{x}$, we perform the corresponding inverse geometric transform $\tilde{x}_i^{-t} = \Gamma_i^{-1}(\tilde{x}_i^t)$ and then take the average of the outputs as $\tilde{x} = \frac{1}{8}\sum_{i=1}^{8} \tilde{x}_i^t$. This strategy is beneficial as it saves training time and has a small number of parameters as compared to individually trained eight models. We also found empirically that this fusion method gives approximately the same performance as the models trained individually with geometric transform. The boosted version is denoted

3.4. Structure of Identity Modules

The structure of the identity modules used in our experiments is depicted in Table 1. Each module consists of a series of layers of "ReLU + Conv" pair. All the convolution layers have a kernel size of 3×3 and 64 output channels. The kernel dilation and padding are the same in each layer and vary between 1 and 3. The skip connection connects the output of the first pair of "ReLU + Conv" to the last "Conv" as shown in figure 2

3.5. Ablation Studies

3.5.1 Influence of the patch size

In this section, we show the role of the patch size and its influence on the denoising performance. Table 2 shows the average PSNR on BSD68 ([42]) for $\sigma_n = 25$ with respect to the increase in size of the training patch. It is obvious that there is a marginal improvement in PSNR as the patch

Table 3. The average PSNR of the denoised images for the BSD68 dataset, with respect to different number of modules **M**. The higher the number of modules, the higher is the accuracy.

Number of modules				
2	3	4	6	8
29.28	29.34	29.34	29.35	29.36

Table 4. Denoising performance for different network settings to dissect the relationship between kernel dilation, number of layers and receptive field.

No of layers	18	9	6
Kernel dilation	1	2	3
	29.34	29.34	29.34

size increases. The main reason for this phenomenon is the size of the receptive field, with a larger patch size network learns more contextual information, hence able to predict local details better.

3.5.2 Number of modules

We show the effect of the number of modules on denoising results. As mentioned earlier, each module **M** consists of six convolution layers, by increasing the number of modules, we are making our network deeper. In this settings, all parameters are constant, except the number of modules, as shown in Table 3. It is clear from the results that making the network deeper increases the average PSNR. However, since fast restoration is desired, we prefer a small network of three modules *i.e.* $\mathbf{M} = 3$, which still achieves better performance than competing methods.

3.5.3 Kernel dilation and number of layers

It has been shown that the performance of some networks can be improved either by increasing the depth of the network or by using large convolution filter size to capture the context information ([57, 56]). This helps the restoration of noisy structures in the image. The usage of traditional 3×3 filters is popular in deeper networks. However, there is a tradeoff between the number of layers and the size of the dilated filters without effecting denoising results. In Table 4, we present three experimental settings to show the tradeoff between the dilated filter size and the depth of the network. In the first experiment, as shown in the first column of Table 4, we use a traditional filter of size 3×3 and depth of 18 to cover the receptive field of training patch.

In the next experiment, we keep the size of the filter the same but enlarge the filter using a dilation factor of two. Although this increases the size of the filter to 5×5; however, still having only nine non-zero entries similar to the above experiment, and it can be interpreted as a sparse filter. Therefore, the receptive field of the training patch can now be covered by nine non-linear mapping layers, contrary to the 18-layers depth per module. Similarly, by expanding the filter by dilation of three would result in the depth of each module to be six. As in Table 4, all three trained models result in similar denoising performance, with the apparent advantage of the shallow network being the most efficient. The number of parameters reduced from 1954k to 663k; similarly, the memory usage for one input patch is reduced from 22MB to 6.5MB.

Table 5. PSNR reported on the BSD68 dataset for $\sigma_n = 25$ when different features are added to the baseline (first row).

Dilation	Identity	Boosting	PSNR
			29.24
✓			29.23
	✓		29.28
✓	✓		29.32
✓	✓	✓	29.34

3.5.4 Network structure Analysis

In Table 5, we show the performance on the BSD68 dataset when adding different features, including a kernel dilation of three across all convolution layers, identity skip connection, or boosting via geometric transformation to the DnCNN baseline which is reported in the first row. The improvement over DnCNN is observed with the introduction of identity skip connections. Applying a dilation of three over 17 or 19 convolutional layers of DnCNN (row 2) does not appear to be effective. However, using dilated convolution in a short chain of six layers, such as row 3, improves the performance further. In Table 5, PSNR is 29.32 dB without boosting and 29.34 dB (last row) if we average the output from eight transformed images.

3.6. Real-world images

So far, state-of-the-art denoising methods, such as DnCNN ([56]), IrCNN ([57]) and BM3D ([15]) *etc*. usually have been evaluated on classical images, and the BSD68 dataset but their performance is limited on real image datasets. Furthermore, real image denoising is becoming popular; hence, we compare our method against recent state-of-the-art [5, 23, 58] algorithms.

3.6.1 Darmstadt Noise Dataset

We visually compare our method with a few recent algorithms, as shown on several samples from [40] in Figure 3. It can be observed that synthetic denoiser such as CBM3D ([15]), DnCNN ([56]) *etc*., and real image denoisers such as CBDNet ([23]) and FFDNet ([58]), are unable to remove the noise from the images. On the other hand, it can be seen that our method eliminates noise and preserve the structures.

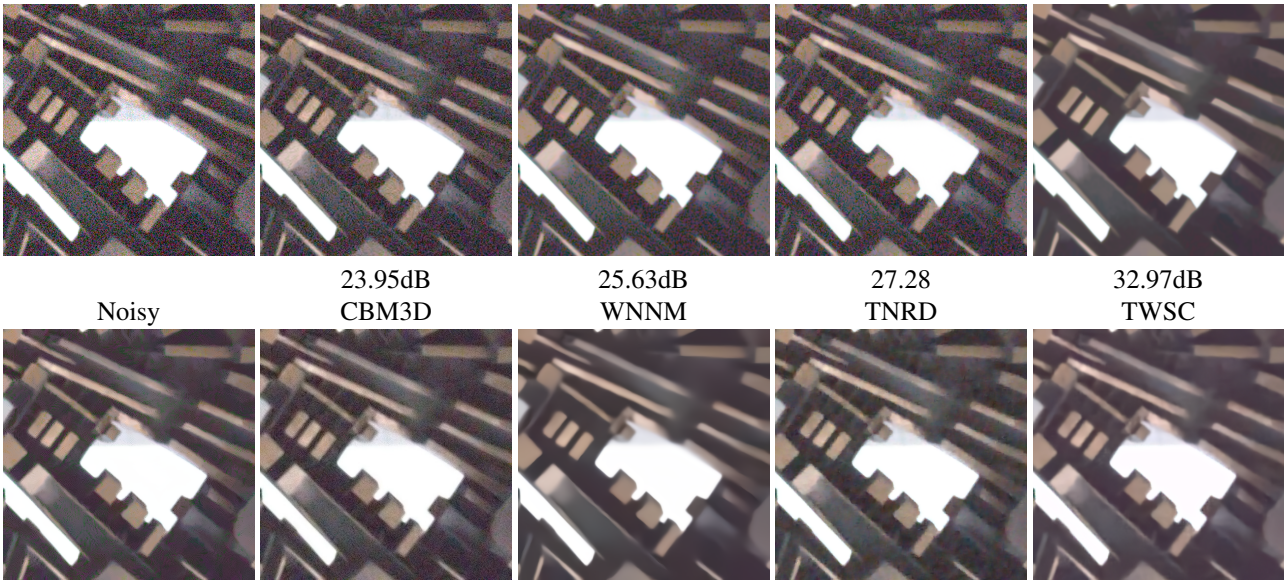

Figure 3. Comparison of our method against the state-of-the-art algorithms on real images containing Gaussian noise from Darmstadt Noise Dataset (DND) benchmark for different denoising algorithms. Difference can be better viewed in magnified view.

Table 6. Mean PSNR and SSIM of the denoising methods evaluated on the real images dataset by [40].

Method	Blind/Non-blind	PSNR	SSIM
CDnCNNB ([56])	Blind	32.43	0.7900
EPLL ([59])	Non-blind	33.51	0.8244
TNRD ([14])	Non-blind	33.65	0.8306
NCSR ([17])	Non-blind	34.05	0.8351
MLP ([10])	Non-blind	34.23	0.8331
FFDNet ([58])	Non-blind	34.40	0.8474
BM3D ([15])	Non-blind	34.51	0.8507
FoE ([42])	Non-blind	34.62	0.8845
WNNM ([22])	Non-blind	34.67	0.8646
NC ([32])	Blind	35.43	0.8841
NI ([2])	Blind	35.11	0.8778
KSVD ([3])	Non-blind	36.49	0.8978
MCWNNM ([50])	Non-blind	37.38	0.9294
TWSC ([49])	Non-blind	37.96	0.9416
FFDNet+ ([58])	Non-blind	37.61	0.9415
CBDNet ([23])	Blind	38.06	0.9421
IERD (Ours)	Blind	**39.20**	**0.9524**
RIDNET [5]	Blind	**39.25**	**0.9528**
IERD+ (Ours)	Blind	**39.30**	**0.9531**

The quantitative results in PSNR and SSIM averaged over all the images for real-world DnD is presented in Table 6. Our method is the best performer, followed by CBDNet. Our method is also able to improve significantly on NI ([2]), a software which is part of coral draw and photoshop. It is to be noted that our method does not require to know the noise level in advance, like [15]'s BM3D and does not require to estimate it separately, like [23]'s CBDNet.

3.6.2 RNI15

The ground-truth images for RNI15 ([32]) are not publicly available; therefore, we present the visual comparison only in Figure 4. In the first example, we can see that there are artifacts on the face for the output of FFDNet ([58]) and CBDNet ([23]) while our method is able to remove the noise without introducing any artifacts. In the second example (given in the second row), our method smooths out the noise and can produce crisp edges while the competing method fails to produce any results without noise. The noise structures are very prominent in the second image near the eyes, as well as the gloves. This shows the robustness of our method against challenging images.

3.6.3 SSID

We utilize the SIDD real noise dataset ([1]) as the final dataset for comparison. Table 7 shows the average PSNR on the validation dataset where our method improves upon FFDNet ([58]) and CBDNet ([23]) with a margin of 9.62dB and 8.04dB. Next, we show the sample visual denoise images from SIDD for various competing algorithms in Figure 5. Our results are resembling the ground-truth image colors while the previous state-of-the-art images produce color casts and artificial colors.

Figure 4. Sample visual examples from RNI15 ([32]). Our method annihilates the noise and preserves the essential details while the competing methods fail to deliver satisfactory results *i.e.* unable to remove noise. Best viewed on high-resolution display.

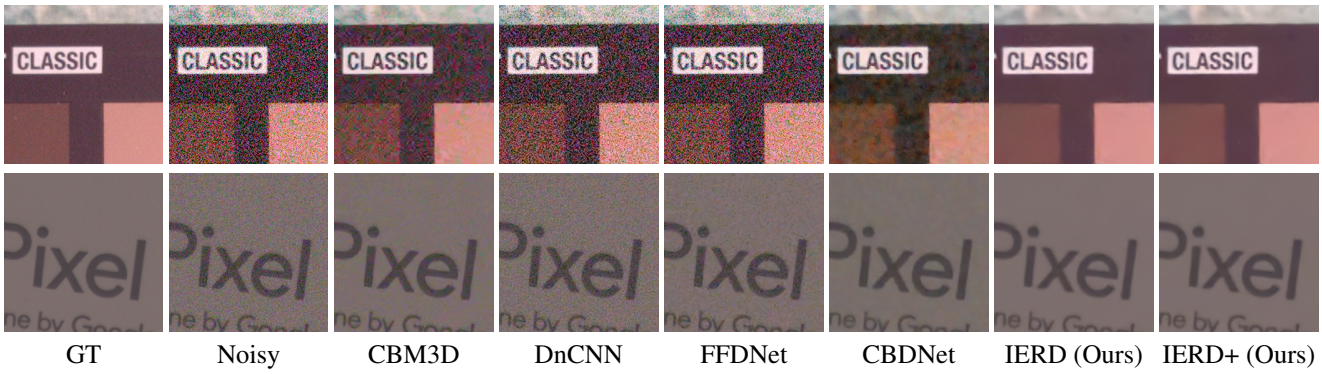

Figure 5. A few challenging examples from SSID dataset ([1]). Our method can restore true colors and remove noise.

Table 7. The quantitative results (in PSNR (dB)) for the SSID dataset ([1]).

Methods					
BM3D	DnCNN	FFDNet	CBDNet	RIDNet	IERD+
30.88	26.21	29.20	30.78	**38.71**	**38.82**

4. Conclusions

To sum up, we employ residual learning and identity mapping to predict the denoised image using a three-module and six-layer deep network of 19 weight layers with dilated convolutional filters without batch normalization. Our choice of network is based on the ablation studies performed in the experimental section of this paper.

This is the first modular framework to predict the denoised output without any dependency on the pre- or post-processing. Our proposed network removes the potentially authentic image structures while allowing the noisy observations to go through its layers, and learns the noise patterns to estimate the clean image.

On real images, we have shown that our method provides visually pleasing results and a gain of about **1.2**dB on Darmstadt Noise Dataset, **9.62**dB on smartphone image denoising dataset (SIDD) in terms of PSNR. The real images appear less grainy after passing through our proposed network and preserving fine image structures. Furthermore, competitive denoising algorithms either require information about the noise in advance or estimate it in a disjoint stage while, on the contrary, our network does not require any information about the noise present in the images.

In the future, we aim to generalize our denoising network to other image restoration and enhancement tasks such as deblurring, color correction, JPEG artifact removal, rain removal, dehazing, and super-resolution *etc*.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 5, 7, 8

[2] ABSoft. Neat image. 7

[3] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *TIP*, 2006. 7

[4] Josue Anaya and Adrian Barbu. Renoir–a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 2018. 5

[5] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *ICCV*, pages 3155–3164, 2019. 2, 6, 7

[6] Saeed Anwar, Fatih Porikli, and Cong Phuoc Huynh. Category-specific object image denoising. *TIP*, pages 5506–5518, 2017. 1, 2

[7] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *TNN*, 1994. 2

[8] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *CVPR*, 2019. 2

[9] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *CVPR*, pages 60–65, 2005. 1, 4

[10] Harold Christopher Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, 2012. 2, 7

[11] Stanley H Chan, Todd Zickler, and Yue M Lu. Monte carlo non-local means: Random sampling for large-scale image filtering. *TIP*, 2014. 1

[12] P. Chatterjee and P. Milanfar. Is denoising dead? *TIP*, 2010. 1

[13] Fei Chen, Lei Zhang, and Huimin Yu. External patch prior guided internal clustering for image denoising. 2015. 2

[14] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *TPAMI*, pages 1256–1272, 2017. 2, 7

[15] Kostadin Dabov, Alessandro F., Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. pages 2080–2095, 2007. 1, 4, 6, 7

[16] Weisheng Dong, Xin Li, D. Zhang, and Guangming Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *CVPR*, 2011. 2

[17] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *TIP*, 2012. 7

[18] Michael Elad and Dmitry Datsenko. Example-based regularization deployed to super-resolution reconstruction of a single image. *Comput. J.*, 2009. 2

[19] L. Zhang F. Chen and H. Yu. External Patch Prior Guided Internal Clustering for Image Denoising. In *ICCV*, 2015. 2

[20] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *TIP*, pages 1395–1411, 2007. 1

[21] Bart Goossens, Hiêp Luong, Aleksandra Pizurica, and Wilfried Philips. An improved non-local denoising algorithm. In *IP*, page 143, 2008. 1

[22] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, pages 2862–2869, 2014. 1, 7

[23] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. *arXiv preprint arXiv:1807.04686*, 2018. 2, 6, 7

[24] K. H., Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, 2015. 3

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, 2015. 5

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *ECCV*, 2016. 3, 4

[28] Erik Hjelmås and Boon Kee Low. Face detection: A survey. *Computer vision and image understanding*, 83(3):236–274, 2001. 1

[29] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 4

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014. 5

[31] M Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 2013. 1, 2

[32] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. The noise clinic: a blind image denoising algorithm. *IPOL*, 2015. 1, 2, 5, 7, 8

[33] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. *CVPR*, 2016. 1, 2, 4

[34] A. Levin and B. Nadler. Natural image denoising: Optimality and inherent bounds. In *CVPR*, pages 2833–2840, 2011. 1

[35] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, 2016. 3

[36] Enming Luo, Stanley H Chan, and Truong Q Nguyen. Adaptive image denoising by targeted databases. *TIP*, pages 2167–2181, 2015. 1, 2

[37] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009. 2

[38] 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*, 2001. 5

[39] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *TPAMI*, pages 2233–2246, 2012. 1

[40] Tobias Plötz and Stefan Roth. Benchmarking denoising algorithms with real photographs. *CVPR*, 2017. 5, 6, 7

[41] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NIPS*, 2018. 2

[42] Stefan Roth and Michael J Black. Fields of experts. *IJCV*, 2009. 1, 2, 5, 7

[43] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 137:24–37, 2015. 1

[44] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *CVPR*, 2014. 2

[45] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *CVPR*, 2016. 5

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

[47] Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018. 5

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

[49] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *ECCV*, 2018. 7

[50] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *ICCV*, 2017. 7

[51] Jun Xu, Lei Zhang, Wangmeng Zuo, David Zhang, and Xiangchu Feng. Patch Group Based Nonlocal Self-Similarity Prior Learning for Image Denoising. In *ICCV*, pages 1211–1218, 2015. 2

[52] L Xu, L Zhang, W Zuo, D Zhang, and X Feng. Patch group based nonlocal self-similarity prior learning for image denoising. 2015. 2

[53] Wenhan Yang, Jiashi Feng, Guosen Xie, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Video super-resolution based on spatial-temporal recurrent residual networks. *Computer Vision and Image Understanding*, 168:79–92, 2018. 1

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

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

[56] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 1, 2, 4, 6, 7

[57] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. *CVPR*, 2017. 1, 2, 6

[58] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *TIP*, 2018. 2, 6, 7

[59] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486, 2011. 2, 4, 7