

Chaining Identity Mapping Modules for Image Denoising

Saeed Anwar, *Student Member, IEEE*, Cong Phuoc Huynh, *Member, IEEE* and Fatih Porikli, *Fellow, IEEE*

Abstract—We propose to learn a fully-convolutional network model that consists of a Chain of Identity Mapping Modules (CIMM) for image denoising. The CIMM structure possesses two distinctive features that are important for the noise removal task. Firstly, each residual unit employs identity mappings as the skip connections and receives pre-activated input in order 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.

After being trained on the BSD400 dataset, 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 conventional benchmark images, the BSD68 dataset and real-world images from Darmstadt Noise Dataset (DND).

Index Terms—Denoising, External datasets, Convolutional neural network, Identity mapping, Modular structure, multimedia applications.

I. 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, the image denoising has become an essential process for various computer vision and image analysis applications. Few notable methods benefiting from image denoising on multimedia applications are detection [3], face recognition [4], image Inpaiting [5], artifact removal [6], image deblurring [7], super-resolution [8] and feature learning in multimedia [9]. In the past few years, the research focus in this area has been shifted to how to make the best use of image priors. To this end, several approaches attempted to exploit non-local self similar (NSS) patterns [10, 11], sparse models [12, 13, 14], gradient models [15, 16], Markov random field models [17], external denoising [18, 19, 20] and convolutional neural networks [1, 2, 21].

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 [10] and BM3D [11], a number of their variants [22, 23, 24] were

S. Anwar and F. Porikli are with the Research School of Engineering, Australian National University, and CSIRO Data61. This research was done while C. P. Huynh was with the Research School of Engineering, Australian National University. This research was supported under Australian Research Council’s Discovery Projects funding scheme (project number DP150104645). E-mail: (saeed.anwar, fatih.porikli, cong.huynh)@anu.edu.au

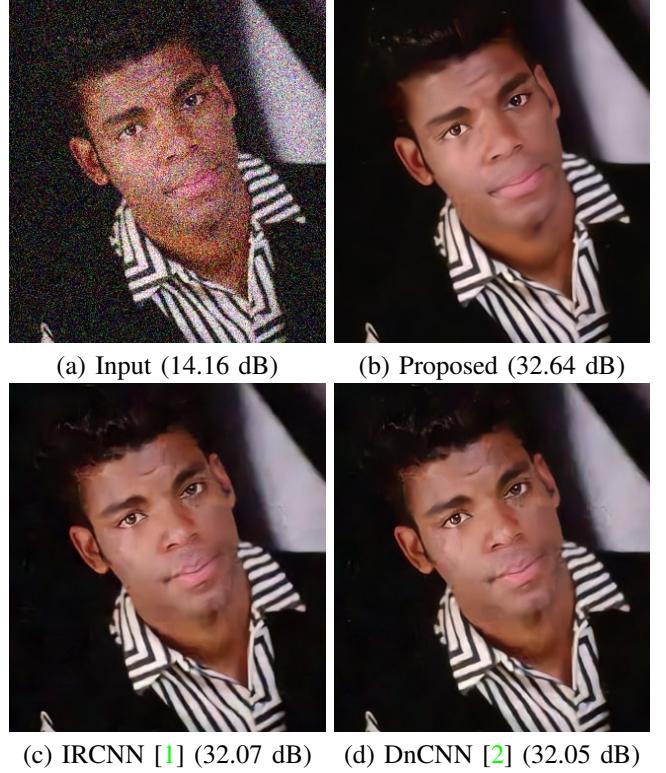


Fig. 1. Denoising results for an image corrupted by the Gaussian noise with $\sigma = 50$. Our result has the best PSNR score, and unlike other methods, it does not have over-smoothing or over-contrasting artifacts. Best viewed in color on high-resolution display.

also proposed to execute the search for similar patches in similar transform domains.

Complementing above, use of external priors for denoising has been motivated by the pioneering studies in [25, 26], 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.* [27] proposed 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 [28, 29, 30], which learn over-complete dictionaries from a set of external natural clean images and then enforce patch self-similarity through sparsity. Similarly, the work in [31] imposed a group residual representation between the sparse representation of the noise contaminated

image and that of its prefiltered version to minimize the error.

Towards an efficient fusion of external datasets, many previous works [32, 33, 34] 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 [35, 36], modified Zoran *et al.* [32]’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 [20], CSID [19] and CID [37]; however, all of these have limited applicability in denoising of generic (from an unspecific class) images.

As an alternative, CSF [38] 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. The pixel values of the shrinkage mappings are learned discriminatively. 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 [1, 2, 21, 38, 39]. Using deep neural networks, IrCNN [1] and DnCNN [2] learn the residual present in the contaminated image by using the noise in the loss function instead of the clean image as the ground-truth. The architectures of IrCNN [1] and DnCNN [2] 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. Besides, they also learn batch normalization parameters after every convolutional layer.

TRND [40] incorporated a field-of-experts prior [17] 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 [21] which exploits the non-local self-similarity using deep networks. Non-local variational schemes have motivated the design of the NLNet model [21] 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 [1] and DnCNN [2], as it inherits the limitations associated with the NSS priors as not all patches recur in an image.

FormResNet is proposed by [41] which builds upon DnCNN

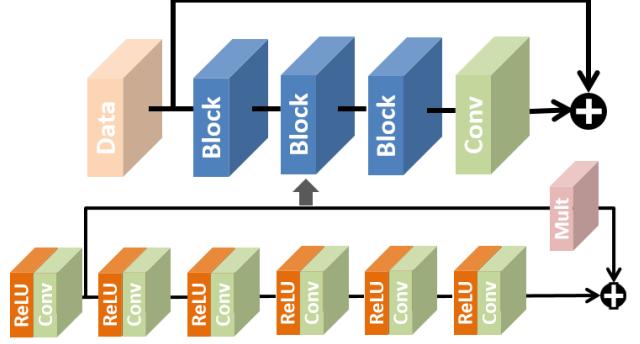


Fig. 2. The proposed network architecture, which consists of multiple modules with similar structures. Each module is composed of a series of pre-activation-convolution layer pairs.

[2]. This model is composed of two networks; both networks are similar to DnCNN [2]; however, the difference lies in the loss layers. The first network termed as “Formatting layer” has incorporated the Euclidean and perceptual loss into one. The Classical algorithms such as BM3D can also replace this formatting layer. Although the second deep network is exactly similar to DnCNN [2]; however, the authors named it “DiffResNet” and input to this network is fed from the first one. The stated formatting layer removes high-frequency corruption in uniform areas, while DiffResNet learns the structured regions. FormResNet [41] improves upon the results of DnCNN [2] by a small margin.

Recently, wavelet domain deep network denoising architecture for CNN is proposed by [42]. The motivation behind this CNN based network is persistent homology analysis [43]. The network takes wavelet transformed images as input and learns the features in transformed manifold rather than original image manifold. The proposed network has high number of channels as compared to DnCNN [2] and FormResNet [41]; hence, the marginal increase in PSNR can be attributed to the number of channels employed for learning features.

A. Inspiration & Motivation

Existing convolutional neural network based image denoising methods [1, 2, 39] 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 mentioned CNN based denoising methods, it will fall into gradients vanishing problem and impel it further [44]. This property of increasing the size of the network is important and helps in performance boost [45]. Therefore, our goal is to propose a model that overcomes this deficiency.

Another reason is the lack of true color denoising. Most of the current denoising systems are either for grayscale image denoising or treat each color channel separately ignoring the relationship between the color channels. Only a handful of works [2, 19, 21, 46] approached color image denoising in its own context.

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 CNNs, including 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 network where we call each module as a mapping modules (MM). The mapping modules can be replicated and easily extended to any arbitrary depth for performance enhancement.

B. Contributions

The contributions of this work can be summarized as follows:

- An effective CNN architecture that consists of a Chain of Identity Mapping modules (CIMM) 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.
- A single denoising network that can handle various noise levels.

II. CHAIN OF IDENTITY MAPPING MODULES

This section presents our approach to image denoising by learning a Convolutional Neural Network consisting of a Chain of Identity Mapping Modules (CIMM). Each module is composed of a series of pre-activation units followed by convolution functions, with residual connections between them. Section II-B formulates the learning objective. Subsequently, the meta-structure of the CIMM network in Section II-A.

A. Network Design

Residual learning has recently delivered state of the art results for object classification [47, 48] and detection [49], while offers training stability. Inspired by the Residual Network variant with identity mapping [48], we adopt a modular design for our denoising network. The design consists of a Chain of Identity Mapping modules (CIMM).

1) *Network elements*: Figure 2 depicts the entire architecture, where identity mapping modules are shown as a blue blocks, which are in turn composed of basic ReLU (orange) and convolution (green) layers. The output of each module is a summation of the identity function and the residual function. In our experiments, we typically employ 64 filters of size 3×3 in each convolution layer.

The meta-level structure of the network is governed by three parameters: the number of identity modules (*i.e.* M), the number of pre-activation-convolution pairs in each module (*i.e.* L), and the number of output channels (*i.e.* C), which we fixed across all the convolution layers.

The high-level structure of the network can be viewed as a chain of identity mapping modules, where the output of each module is fed directly into the subsequent one. Subsequently, 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 noise-free image/patch is then subtracted from 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 main responsibility is to learn a set of convolution filters to predict image noise. In addition, the identity mapping branch in each module allows the propagation of the loss gradients in both directions without any bottleneck.

2) *Justification of design*: For image denoising, several previous works have adopted a fully convolutional network design, without any pooling mechanism [1, 2, 50]. 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 learning 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 [10, 11]. Instead of using a unit 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 [48]. In a similar fashion, our design employs ReLU before each convolution layer. This design differs from existing neural network architectures for denoising [2, 21, 50]. The pre-activation helps training to converge more easily, by 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.

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

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

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

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

TABLE I
DENOISING PERFORMANCE (IN PSNR) ON THE BSD68 DATASET [51] 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

TABLE II

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	4	6	8
29.28	29.34	29.35	29.36

where $\mathbf{z}_{m,0}$ is the output of the first ReLU-convolution pair. 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 $h(\mathbf{y}, \mathcal{W}) = f(\mathbf{y} + \sum_{m=1}^M \mathbf{r}_m, w_{m+1})$.

B. Learning to Denoise

Our convolutional neural network (CNN) is trained on image patches or regions rather than at the image-level. This decision is driven by a number of reasons. Firstly, it offers random sampling of a large number of training samples at different locations from various images. 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 success in approaches that learns image patch priors from external data for image denoising [32].

From a set of noise-free training images, we randomly crop a number of training patches $\mathbf{x}_i, i = 1, \dots, N$ as the groundtruth. 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 $\mathbf{y}_i, i = 1, \dots, N$. With this setup, our image denoising network (described in Section II-A) is aimed to reconstruct a patch $\mathbf{x}_i^* = h(\mathbf{y}_i, \mathcal{W})$ from the input patch \mathbf{y}_i .

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

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

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 III-B.

III. EXPERIMENTS

A. Datasets and Baselines

We performed experimental validation on the widely used classical images (same number and images as [2]). Similarly, we also use DnD datasets [52] consists of real 1000 images and BSD68 dataset [17] composed of 68 images. It is to be noted

TABLE III
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

here, that our BSD400 dataset [51] for training and BSD68 dataset [17] for testing are disjoint. To generate noisy test images, we corrupt the images by additive white Gaussian noise with standard deviations (std) of $\sigma_n = 15, 25, 50, 70$, as employed by [1, 2, 21]. For evaluation purposes, we use the Peak Signal-to-Noise Ratio (PSNR) index as the error metric. We compare our proposed method with numerous state-of-the-art methods, including BM3D [11], WNNM [12], MLP [39], EPLL [32], TNRD [40], IRCNN [1], DnCNN [2] and NLNET [21]. To ensure a fair comparison, we use the default setting provided by the respective authors.

B. Training Details

The training input to our network is noisy and noise-free patch pairs of size 40×40 cropped randomly from the BSD400 dataset [51]. 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 400 images of BSD400 dataset.

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 model as noise-specific model. Alternatively, we train a single model for the noise range [1, 50] (similar to [2]) and we refer to this model as noise-agnostic model. At each update of training, we construct a batch by randomly selecting noisy patches with noise levels between 1 and 50.

We implement the denoising method in the Caffe framework on two Tesla P100 GPUs, and employ the Adam optimization algorithm [53] 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 ten epochs. Our network takes six hours and 40 epochs to train a model. The input to our network are patches of 40×40 of mini-batches of size 64. We train our network from scratch by a random initialization of the convolution weights according to the method in [54] and a regularization strength, *i.e.* weight decay, of 0.0001.

C. Boosting Denoising Performance

To boost the performance of the trained model, we use the late fusion/geometric transform strategy as adopted by [55]. 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 $i = 1, \dots, 8$. From these geometrically transformed images, we estimate corresponding denoised images $\{\hat{x}_1^t, \hat{x}_2^t, \dots, \hat{x}_8^t\}$, where $\hat{x}_i^t = h(\hat{y}_i^t, W)$ using our model. To generate the final denoised image \hat{x} , we perform the corresponding inverse geometric transform $\tilde{x}_i^{-t} = \Gamma_i^{-1}(\hat{x}_i^t)$

TABLE IV
DETAILED ARCHITECTURE OF AN IDENTITY MAPPING MODULE.

Parameters	Mapping Module Layers					
	1 st	2 nd	3 rd	4 th	5 th	6 th
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

TABLE V
PSNR REPORTED ON THE BSD68 DATASET FOR $\sigma_n = 25$ WHEN DIFFERENT FEATURES ARE ADDED TO THE BASELINE (FIRST ROW).

Dilation	Identity	Boosting	PSNR
No	No	No	29.24
Yes	No	No	29.23
No	Yes	No	29.28
Yes	Yes	No	29.32
Yes	Yes	Yes	29.34

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 have 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.

D. Identity Mapping Modules

The structure of the mapping modules used in our experiments is depicted in Table IV. Each module consists of a series 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 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 pair ReLU + Conv as shown in figure 2

E. Ablation Studies

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 I shows the average PSNR on BSD68 [17] 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 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.

2) *Number of modules:* We show the effect of the number of modules on denoising results. As mentioned earlier, each module \mathbf{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 II. It is clear from the results that making the network deeper increase the average PSNR. However, since fast restoration is desired, we prefer a small network of three modules *i.e.* $\mathbf{M} = 3$, which achieves better performance than other methods.

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 [1, 2]. This helps the restoration of noisy structures in the image. The

usage of traditional 3×3 filters is popular in deeper networks. However, using dilated filters there is a tradeoff between the number of layers and the size of the dilated filters without effecting denoising results. In Table III, we present three experimental settings to show the tradeoff between the dilated filter size and the depth of network. In the first experiment as shown in the first column of Table III, we use a traditional filter of size 3×3 and depth of 18 to cover the receptive field of training patch of size 40. In the next experiment, we keep the size of the filter same but enlarge the filter using a dilation factor of two. This increases the size of the filter to 5×5 but having nine non-zero entries 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 a dilation of three would result in the depth of each module to be six. As in Table III, all three trained models result in similar denoising performance, with the obvious 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 to 22MB to 6.5MB.

4) *Network structure Analysis:* In Table V, we show the performance on 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 V, PSNR is 29.32 dB without boosting and 29.34 dB (last row) if we average the output from eight transformed images.

F. Grayscale Image Denoising

In this section, first we demonstrate how our method performs on classical images and then report results on the BSD68 dataset.

1) *Classical Images:* For completeness, we compare our algorithm to several state-of-the-art denoising methods using grayscale classical images shown in Figure 3 and reported in Table VI.

In Table VI, we present the average PSNR for the denoised images. Our network is the best performer for almost all classical images except “Barbara”. The reason for this may be the repetitive structures in the mentioned image, which makes it easy for BM3D [11] and WNNM [12] to find and employ patches with great similarity to the noisy input, hence providing better results.

Subsequently, we depict an example from the classical images. The visual quality of our recovered images, as shown in Figure 3, is better than all others. This also illustrates that our network restores aesthetically pleasing textures. Small and noticeable features restored by our network include the sharpness and the clarity of the subtle textures around the fore and hind wings, mouth, and antennas of the butterfly.

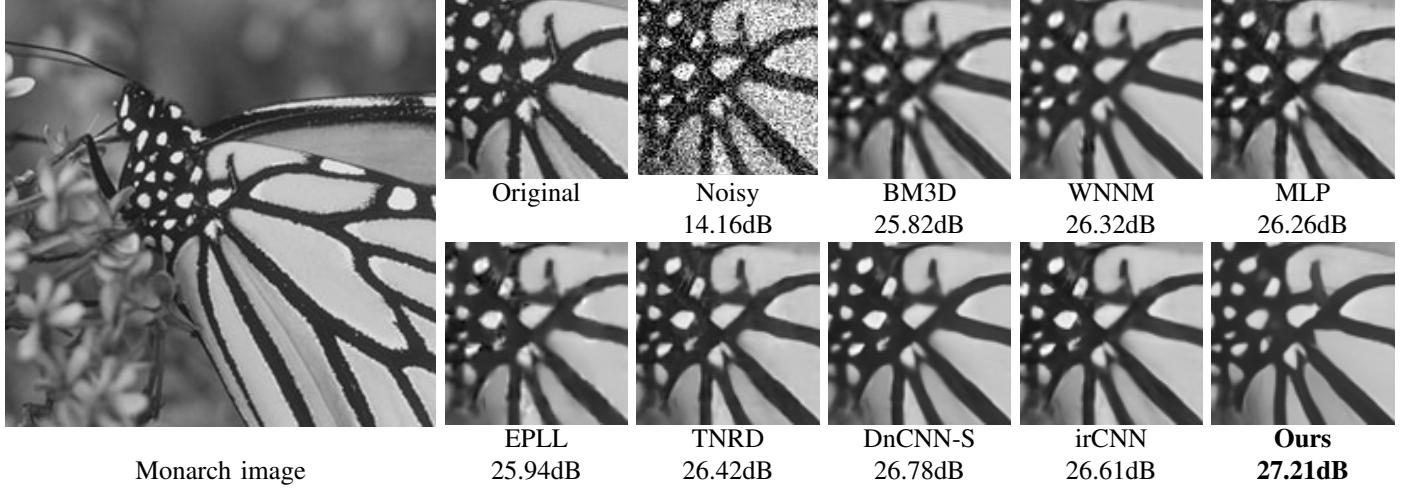


Fig. 3. Denoising quality comparison on a sample image with strong edges and texture, selected from classical image set for noise level $\sigma_n = 50$. The visual quality, *i.e.* sharpness of the edges on the wings and small textures reproduced by our method is the best among all.

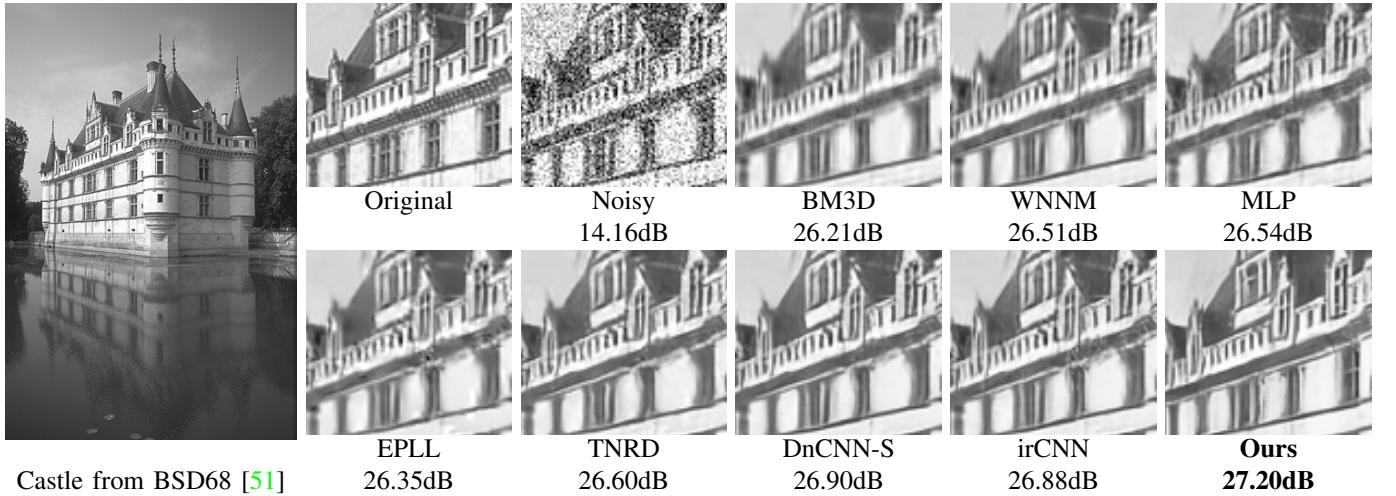


Fig. 4. Comparison on a sample image from BSD68 dataset [51] for $\sigma_n = 50$. Our network is able to recover fine textures on the castle

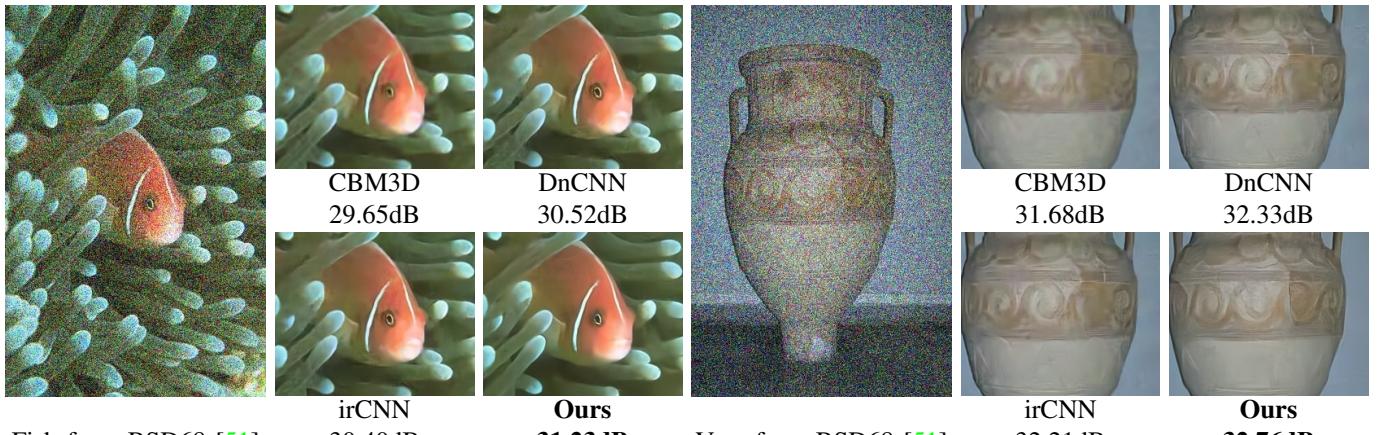


Fig. 5. Denoising performance for state-of-the-art versus the proposed method on sample color images from the dataset in [51], where the noise standard deviation σ_n is 50. The image we recover is more natural, contains less contrast artifacts and is closest to the ground-truth.



Fig. 6. A sample color image with rich textures, selected from the BSD68 dataset [51] for $\sigma_n = 25$.

TABLE VI

PERFORMANCE COMPARISON BETWEEN IMAGE DENOISING ALGORITHMS ON WIDELY USED CLASSICAL IMAGES, IN TERMS OF PSNR (IN dB). THE BEST RESULTS ARE HIGHLIGHTED WITH RED COLOR WHILE THE BLUE COLOR REPRESENTS THE SECOND BEST DENOISING RESULTS.

	Cman	House	Peppers	Starfish	Monar	Airpl	Parrot	Lena	Barbara	Boat	Man	Couple	Average
$\sigma_n = 15$													
BM3D [11]	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10	32.372
WNNM [12]	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	33.60	32.27	32.11	32.17	32.696
EPLL [32]	31.85	34.17	32.64	31.13	32.10	31.19	31.42	33.92	31.38	31.93	32.00	31.93	32.138
CSF [38]	31.95	34.39	32.85	31.55	32.33	31.33	31.37	34.06	31.92	32.01	32.08	31.98	32.318
TNRD [40]	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.23	32.11	32.502
DnCNNS [2]	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47	32.859
DnCNNB [2]	32.10	34.93	33.15	32.02	32.94	31.56	31.63	34.56	32.09	32.35	32.41	32.41	32.680
IrCNN [1]	32.55	34.89	33.31	32.02	32.82	31.70	31.84	34.53	32.43	32.34	32.40	32.40	32.769
Ours-agnostic	32.11	35.10	33.28	32.31	33.07	31.58	31.80	34.67	32.48	32.42	32.40	32.50	32.812
Ours-specific	32.61	35.21	33.21	32.35	33.33	31.77	32.01	34.69	32.74	32.44	32.50	32.52	32.950
$\sigma_n = 25$													
BM3D [11]	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.969
WNNM [12]	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	31.24	30.03	29.76	29.82	30.257
EPLL [32]	29.26	32.17	30.17	28.51	29.39	28.61	28.95	31.73	28.61	29.74	29.66	29.53	29.692
MLP [39]	29.61	32.56	30.30	28.82	29.61	28.82	29.25	32.25	29.54	29.97	29.88	29.73	30.027
CSF [38]	29.48	32.39	30.32	28.80	29.62	28.72	28.90	31.79	29.03	29.76	29.71	29.53	29.837
TNRD [40]	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71	30.055
DnCNNS [2]	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12	30.436
DnCNNB [2]	29.94	33.05	30.84	29.34	30.25	29.09	29.35	32.42	29.69	30.20	30.09	30.10	30.362
IrCNN [1]	30.08	33.06	30.88	29.27	30.09	29.12	29.47	32.43	29.92	30.17	30.04	30.08	30.384
Ours-agnostic	29.87	33.34	30.94	29.68	30.39	29.08	29.38	32.65	30.17	30.27	30.08	30.20	30.505
Ours-specific	30.26	33.44	30.87	29.77	30.62	29.23	29.61	32.66	30.29	30.30	30.18	30.24	30.624
$\sigma_n = 50$													
BM3D [11]	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.722
WNNM [12]	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	27.79	26.97	26.94	26.64	27.052
EPLL [32]	26.10	29.12	26.80	25.12	25.94	25.31	25.95	28.68	24.83	26.74	26.79	26.30	26.471
MLP [39]	26.37	29.64	26.68	25.43	26.26	25.56	26.12	29.32	25.24	27.03	27.06	26.67	26.783
TNRD [40]	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50	26.812
DnCNNS [2]	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90	27.178
DnCNNB [2]	27.03	30.02	27.39	25.72	26.83	25.89	26.48	29.38	26.38	27.23	27.23	26.91	27.206
IrCNN [1]	26.88	29.96	27.33	25.57	26.61	25.89	26.55	29.40	26.24	27.17	27.17	26.88	27.136
Ours-agnostic	27.03	30.48	27.57	26.01	27.03	25.84	26.53	29.77	26.89	27.28	27.29	27.06	27.398
Ours-specific	27.25	30.70	27.54	26.05	27.21	26.06	26.53	29.65	26.62	27.36	27.26	27.24	27.457
$\sigma_n = 70$													
BM3D [11]	24.62	27.91	25.07	23.56	24.24	23.75	24.49	27.57	25.47	25.40	25.56	25.00	25.221
WNNM [12]	24.86	28.59	25.25	23.78	24.62	24.00	24.64	27.85	26.17	25.58	25.68	25.18	25.517
EPLL [32]	24.60	27.32	25.03	23.52	24.19	23.72	24.44	27.11	23.20	25.27	25.50	24.80	24.891
DnCNNS [2]	25.37	28.22	25.50	23.97	25.10	24.34	24.98	27.85	23.97	25.76	25.91	25.31	25.523
Ours-specific	25.83	29.19	25.90	24.28	25.66	24.59	25.12	28.25	25.06	26.00	26.02	25.78	25.974

Furthermore, a magnified view of the results in Figures 3 for methods such as [1, 11, 21] shows artifacts and failures in the smooth areas. Our CNN network also outperforms [1, 2, 21], which are trained using deep neural networks.

2) *BSD68 Dataset*: We present the average PSNR scores for the estimated denoised images in Table VII. The IR-CNN [1] and DnCNN [2] network structures are similar, hence produce nearly similar results. On the other hand, our method reconstructs the images accurately, achieving higher PSNR than completing methods on all four levels of noise. Furthermore, the difference in PSNR between our method and the state-of-the-art techniques at the higher noise levels.

For a comprehensive evaluation, we demonstrate the visual results on a selected grayscale image from BSD68 [17] dataset in Figure 4. In our results, the image details are more similar to the ground-truth details, and our quantitative results are numerically higher than the others. Our method outperforms the second best method by several orders of magnitude (PSNR is computed in the logarithmic scale). Also, note that the de-

noising results of other CNN based algorithms are comparable to each other.

In Table IX, we compare our method with Bae *et al.* [42], Jiao *et al.* [41] and DnCNN [2] in terms of the training data, model capacity and denoising performance. Our method outperforms the alternatives with a model capacity comparable to DnCNN [2] and Jiao *et al.* [41], and much lower than Bae *et al.* [42].

G. Color Image Denoising

For noisy color images, we train our network with the noisy RGB input patches of size 40×40 with the corresponding clean ground-truth patches. We only modify the first and last convolution layer of the grayscale network to input and output three channels instead of one channel, keeping all other parameters same as the grayscale network.

We present the quantitative results in Table VIII and qualitative results in Figures 5 and 6 against benchmark methods including the latest CNN based state-of-the-art color image

TABLE VII

PERFORMANCE COMPARISON BETWEEN OUR METHOD AND EXISTING ALGORITHMS ON THE GRayscale VERSION OF THE BSD68 DATASET [51]. THE MISSING DENOISING RESULTS, INDICATED BY “-”, OCCURS WHEN THE METHOD IS NOT TRAINED TO DEAL WITH THE INPUT NOISY IMAGES.

Noise Level	BM3D	WNNM	EPLL	TNRD	DenoiseNet	Methods				
						DnCNNs	IrCNN	NLNet	Ours-Agnostic	Ours-Specific
15	31.08	31.32	31.19	31.42	31.44	31.73	31.63	31.52	31.68	31.81
25	28.57	28.83	28.68	28.92	29.04	29.23	29.15	29.03	29.18	29.34
50	25.62	25.83	25.67	26.01	26.06	26.23	26.19	26.07	26.31	26.40
70	24.44	-	24.43	-	-	24.90	-	-	-	25.13

TABLE VIII

THE SIMILARITY BETWEEN THE DENOISED COLOR IMAGES AND THE GROUND-TRUTH COLOR IMAGES OF BSD68 DATASET FOR OUR NETWORK AND EXISTING ALGORITHMS MEASURED BY PSNR (IN DB) REPORTED FOR NOISE LEVELS OF $\sigma=15, 25$, AND 50 .

Noise Levels	CBM3D [46]	MLP [39]	TNRD [40]	DnCNN [2]	Methods				
					IrCNN [1]	CNLNet [21]	Ours-agnostic	Ours-specific	
15	33.50	-	31.37	33.89	33.86	33.69	33.96	34.12	
25	30.69	28.92	28.88	31.33	31.16	30.96	31.32	31.42	
50	27.37	26.00	25.94	27.97	27.86	27.64	28.05	28.19	

TABLE IX

COMPARISONS WITH STATE-OF-THE-ART METHODS ON BSD68 WITH $\sigma_n = 50$, AND BSD100 WITH $\sigma_n = 25$. THE RESULTS OF [42] AND [41] ARE TAKEN FROM THEIR RESPECTIVE PAPERS.

	Jiao [41]	Bae [42]	DnCNN [2]	Ours
Kernel Size	3x3	3x3	3x3	3x3
Patch Size	40x40	40x40	40x40	40x40
Channels	64	320	64	64
Training data	BSD400	BSD400 +Urban100	BSD400	BSD400
BatchNorm	Yes	Yes	Yes	No
Conv. layers	20	20	17	19
No. parameters	671k	16629k	566k	630k
BSD68	-	26.35dB	26.23dB	26.40dB
BSD100	29.08dB	-	29.05dB	29.25dB

denoising techniques [1, 2, 11]. It can be observed that our algorithm attains an improved average PSNR on all three different noise levels for the color version of BSD68 dataset [17]. As shown, our method restores true colors closer to their authentic values while others fail and induce false colorizations in certain image regions. Furthermore, a close look reveals that our network reproduces the local texture with much less artifacts and sufficiently sharp details.

H. Darmstadt Noise Dataset: Real-world images

So far, state-of-the-art denoising methods, such as FormResNet [41], DnCNN [2], IrCNN [1] and BM3D [11] etc. have normally been evaluated on classical images and the BSD68 dataset. Recently, [52] proposed the Darmstadt Noise Dataset (DND) benchmark for denoising algorithms which consists of 50 images. The dataset is composed of images with interesting and challenging structures. The images are converted to sRGB and gamma correction is applied. The size of each image is in Megapixels; therefore, each image is cropped at 20 locations and each composed of 512×512 pixels yielding 1000 test crops, and overlap between the images is about 10%. Only these test images are provided, there are no images for either training or validation. Therefore, we use the same model which is trained on the synthetic BSD68 dataset [17].

The quantitative results in PSNR and SSIM averaged over all the images for real-world DnD is presented in Table X. It can be observed that our method is the best performer followed by BM3D. Previously, the classical method BM3D

is considered to be outperformed by most of the state-of-the-art algorithms on synthetic datasets; however, this is not the case when using the real-world Darmstadt Noise Dataset. It is to be noted that our method does not require to know the noise level in advance unlike BM3D and other state-of-the-art techniques. Furthermore, we visually compare our method with a few recent algorithms as shown on several samples from [52] in Figure 7¹. It can be observed that both CBM3D [11], as well as DnCNN [2], are unable to remove the noise from the images. On the other hand, it can be seen that our method eliminates the noise and preserve the structures.

IV. CONCLUSIONS

To sum up, we employ residual learning and identity mapping to predict the denoised image using a five-module and five-layer deep network of 26 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.

In synthetic images case, we have provided ample examples and have shown that our network outperforms classical state-of-the-art denoising algorithms that are intended for use on natural images. Furthermore, we have compared against the current convolutional neural networks both visually and numerically. Our network gain is about **0.1dB** on BSD68 dataset [51] and results are visually pleasing.

On real images of Darmstadt Noise Dataset (DND) [52], we have shown that our method provides visually pleasing results and a gain of about **1.43dB** of PSNR. The real images appear less grainy after passing through our proposed network and preserving fine image structures. Furthermore, competitive denoising algorithms require information about the noise in the image while on the contrary, our network does not require any information about the noise present in the images.

¹PSNR for individual images are not available as [52]’s system only provide average PSNR

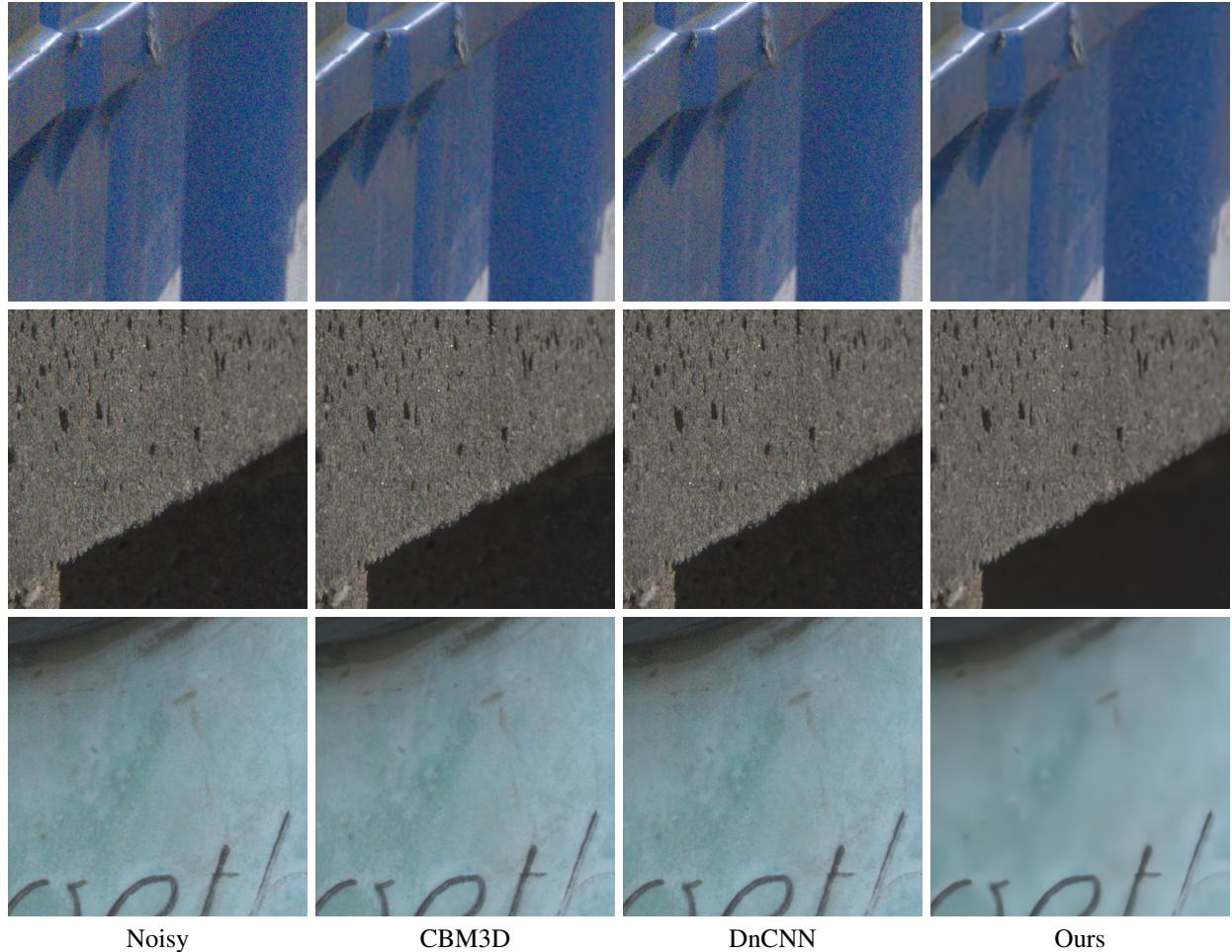


Fig. 7. Comparison of our method against the state-of-the-art algorithms on real images containing Gaussian noise from Darmstadt Noise Dataset (DND) benchmark for different denoising algorithms [52]. Difference can be better viewed in magnified view.

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

For the time being our approach is only applicable to Gaussian noise removal. However, we would like to train our model with different noise types such as Poisson, astronomical *etc.* and examine its performance on these specific noise types. It should be noted that other state-of-the-art methods, for example, BM3D [11], WNNM [12] are only applicable to Gaussian noise and may not be readily adapted to handle different noise types.

All CNN approaches performance on images with regular, and repeating structures such as ‘‘Barbara’’ is relatively less in terms of PSNR compared to classical denoising methods. This phenomenon is due to the design of traditional denoising methods to exploit the regular and repeating structures. To overcome this issue, either block-matching scheme can be incorporated into our CNN approach or relying on consolidating the outcome of various denoising algorithms with our CNN approach.

REFERENCES

- [1] K. Zhang, W. Zuo, S. Gu, and L. Zhang, ‘‘Learning deep cnn denoiser prior for image restoration,’’ *CVPR*, 2017. [1](#), [2](#), [3](#), [4](#), [5](#), [7](#), [8](#)
- [2] 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](#), [4](#), [5](#), [7](#), [8](#), [10](#)
- [3] M. Liu, H. Liu, and C. Chen, ‘‘Robust 3d action recognition through sampling local appearances and global distributions,’’ *TMM*, 2018. [1](#)
- [4] L. Liu, C. Xiong, H. Zhang, Z. Niu, M. Wang, and S. Yan, ‘‘Deep aging face verification with large gaps,’’ *TMM*, 2016. [1](#)
- [5] J. Liu, S. Yang, Y. Fang, and Z. Guo, ‘‘Structure-guided image inpainting using homography transformation,’’ *TMM*, 2018. [1](#)
- [6] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, ‘‘Deep universal generative adversarial compression artifact removal,’’ *TMM*, 2019. [1](#)
- [7] X. Yu, F. Xu, S. Zhang, and L. Zhang, ‘‘Efficient patch-wise non-uniform deblurring for a single image,’’ *TMM*, 2014. [1](#)
- [8] J. Xie, R. S. Feris, S. Yu, and M. Sun, ‘‘Joint super resolution and denoising from a single depth image,’’ *TMM*, 2015. [1](#)
- [9] X. Yang, T. Zhang, and C. Xu, ‘‘Cross-domain feature learning in multimedia,’’ *TMM*, 2015. [1](#)
- [10] A. Buades, B. Coll, and J.-M. Morel, ‘‘A non-local algorithm for image denoising,’’ in *CVPR*, 2005, pp. 60–65. [1](#), [3](#)
- [11] K. Dabov, A. F. V. Katkovnik, and K. Egiazarian, ‘‘Image denoising by sparse 3-D transform-domain collaborative filtering,’’ 2007, pp. 2080–2095. [1](#), [3](#), [4](#), [5](#), [7](#), [8](#), [9](#), [10](#)
- [12] S. Gu, L. Zhang, W. Zuo, and X. Feng, ‘‘Weighted nuclear norm minimization with application to image denoising,’’ in *CVPR*, 2014, pp. 2862–2869. [1](#), [4](#), [5](#), [7](#), [9](#), [10](#)
- [13] 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*, pp. 2233–2246, 2012. [1](#)
- [14] D. Huang, L. Kang, Y. F. Wang, and C. Lin, ‘‘Self-learning based image decomposition with applications to single image denoising,’’ *TMM*, 2014. [1](#)
- [15] J. Xu and S. Osher, ‘‘Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoising,’’ *TIP*, pp. 534–544, 2007. [1](#)

TABLE X
MEAN PSNR AND SSIM OF THE DENOISING METHODS EVALUATED ON THE REAL IMAGES DATASET BY [52].

Metrics	Methods							
	WNNM [12]	EPLL [32]	BM3D [11]	MLP [39]	DenoiseNet [56]	TNRD [40]	DnCNN [2]	Ours-agnostic
PSNR	34.44	33.51	34.61	34.14	35.08	29.92	32.43	36.04
SSIM	0.8646	0.8244	0.8507	0.8331	0.8680	0.8306	0.7900	0.9136

- [16] Y. Weiss and W. T. Freeman, “What makes a good model of natural images?” in *CVPR*, 2007, pp. 1–8. 1
- [17] S. Roth and M. J. Black, “Fields of experts,” *IJCV*, 2009. 1, 2, 4, 5, 7, 8
- [18] 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
- [19] S. Anwar, F. Porikli, and C. P. Huynh, “Category-specific object image denoising,” *TIP*, pp. 5506–5518, 2017. 1, 2
- [20] E. Luo, S. H. Chan, and T. Q. Nguyen, “Adaptive image denoising by targeted databases,” *TIP*, pp. 2167–2181, 2015. 1, 2
- [21] S. Lefkimiatis, “Non-local color image denoising with convolutional neural networks,” *CVPR*, 2016. 1, 2, 3, 4, 5, 7, 8
- [22] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images,” *TIP*, pp. 1395–1411, 2007. 1
- [23] M. Lebrun, A. Buades, and J.-M. Morel, “A nonlocal bayesian image denoising algorithm,” *SIAM Journal on Imaging Sciences*, 2013. 1
- [24] B. Goossens, H. Luong, A. Pizurica, and W. Philips, “An improved non-local denoising algorithm,” in *IP*, 2008, p. 143. 1
- [25] A. Levin and B. Nadler, “Natural image denoising: Optimality and inherent bounds,” in *CVPR*, 2011, pp. 2833–2840. 1
- [26] P. Chatterjee and P. Milanfar, “Is denoising dead?” *TIP*, 2010. 1
- [27] S. H. Chan, T. Zickler, and Y. M. Lu, “Monte carlo non-local means: Random sampling for large-scale image filtering,” *TIP*. 1
- [28] M. Elad and D. Datsenko, “Example-based regularization deployed to super-resolution reconstruction of a single image,” *Comput. J.*, 2009. 1
- [29] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in *ICCV*, 2009. 1
- [30] W. Dong, X. Li, D. Zhang, and G. Shi, “Sparsity-based image denoising via dictionary learning and structural clustering,” in *CVPR*, 2011. 1
- [31] Q. W. Y. B. Zhiyuan Zha, Xinggan Zhang and L. Tang, “Group sparsity residual constraint for image denoising,” in *preprint*, 2017. 1
- [32] D. Zoran and Y. Weiss, “From learning models of natural image patches to whole image restoration,” in *ICCV*, 2011, pp. 479–486. 2, 4, 7, 10
- [33] L. Z. F. Chen and H. Yu, “External Patch Prior Guided Internal Clustering for Image Denoising,” in *ICCV*, 2015. 2
- [34] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, “Patch Group Based Nonlocal Self-Similarity Prior Learning for Image Denoising,” in *ICCV*, 2015, pp. 1211–1218. 2
- [35] L. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, “Patch group based nonlocal self-similarity prior learning for image denoising,” 2015. 2
- [36] F. Chen, L. Zhang, and H. Yu, “External patch prior guided internal clustering for image denoising,” 2015. 2
- [37] H. Yue, X. Sun, J. Yang, and F. Wu, “Image denoising by exploring external and internal correlations,” *TIP*, pp. 1967–1982, 2015. 2
- [38] U. Schmidt and S. Roth, “Shrinkage fields for effective image restoration,” in *CVPR*, 2014. 2, 7
- [39] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image denoising: Can plain neural networks compete with bm3d?” in *CVPR*, 2012. 2, 4, 7, 8, 10
- [40] Y. Chen and T. Pock, “Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration,” *TPAMI*, pp. 1256–1272, 2017. 2, 4, 7, 8, 10
- [41] J. Jiao, W.-C. Tu, S. He, and R. W. Lau, “Formresnet: Formatted residual learning for image restoration,” in *CVPRW*, 2017. 2, 7, 8
- [42] W. Bae, J. Yoo, and J. C. Ye, “Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification,” in *CVPRW*, 2017. 2, 7, 8
- [43] H. Edelsbrunner and J. Harer, “Persistent homology-a survey,” *Contemporary mathematics*, pp. 257–282, 2008. 2
- [44] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *TNN*, 1994. 2
- [45] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778. 2
- [46] 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. 2, 8
- [47] K. H., X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, 2015. 3
- [48] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” *ECCV*, 2016. 3
- [49] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature pyramid networks for object detection,” *CoRR*, 2016. 3
- [50] J. Kim, J. Kwon Lee, and K. Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in *CVPR*, 2016. 3
- [51] 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. 4, 6, 8
- [52] T. Plötz and S. Roth, “Benchmarking denoising algorithms with real photographs,” *arXiv preprint arXiv:1707.01313*, 2017. 4, 8, 9, 10
- [53] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, 2014. 4
- [54] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *CoRR*, 2015. 4
- [55] R. Timofte, R. Rothe, and L. Van Gool, “Seven ways to improve example-based single image super resolution,” in *CVPR*, 2016. 4
- [56] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, “Deep class-aware image denoising,” in *International Conference on Sampling Theory and Applications*, 2017. 10



Saeed Anwar received Bachelor degree in Computer Systems Engineering with distinction from University of Engineering and Technology (UET), Pakistan, in July 2008, and Master degree in Erasmus Mundus Vision and Robotics (Vibot), Europe, in August 2010 with distinction. During his masters, he carried out his thesis at Toshiba Medical Visualization Systems Europe, Scotland. He has also been a visiting research fellow at Pal Robotics, Barcelona. Since 2014, he is a PhD student at ANU and Data61/CSIRO. He has also been working as a Lecturer and Assistant Professor at NUCES, Pakistan. His major research interests are low-level vision, enhancement, restoration, & optimization.



Cong Phuoc Huynh is a research scientist at Amazon Lab126, and concurrently an adjunct research fellow at ANU. He has co-authored a book on imaging spectroscopy for scene analysis and over 20 journal articles and conference papers in computer vision and pattern recognition. He is an inventor of eight patents on spectral imaging. He is a co-recipient of a DICTA Best Student’s paper Award in 2013. Previously, he was a computer vision researcher at NICTA. He received a B.Sc. degree (Hons) in Computer Science and Software Engineering from the University of Canterbury, New Zealand in 2006, and M.Sc. and Ph.D. degrees in Computer Science from ANU in 2007 and 2012.



Fatih Porikli is an IEEE Fellow and a Professor in the Research School of Engineering, ANU. He is also managing the Computer Vision Research Group at Data61/CSIRO. He has received his PhD from New York University in 2002. Previously he served Distinguished Research Scientist at Mitsubishi Electric Research Laboratories. Prof. Porikli is the recipient of the R&D 100 Scientist of the Year Award in 2006. He won 4 best paper awards at premier IEEE conferences and received 5 other professional prizes. Prof. Porikli authored more than 150 publications and invented 66 patents. He is the co-editor of 2 books. He is serving as the Associate Editor of 5 journals for the past 8 years. He was the General Chair of AVSS 2010 and WACV 2014, and the Program Chair of WACV 2015 and AVSS 2012. His research interests include computer vision, deep learning, manifold learning, online learning, and image enhancement with commercial applications in video surveillance, car navigation, intelligent transportation, satellite, and medical systems.