

# Real Image Denoising Based on Multi-Scale Residual Dense Block and Cascaded U-Net with Block-Connection

Long Bao\*, Zengli Yang\*, Shuangquan Wang, Dongwoon Bai, Jungwon Lee  
SOC R&D, Samsung Semiconductor, Inc.

{long.bao, zengli.y, shuangquan.w, dongwoon.bai, jungwon2.lee}@samsung.com

## Abstract

Benefiting from the recent real image dataset, learning-based approaches have achieved good performance for real-image denoising. To further improve the performance for Bayer raw data denoising, this paper introduces two new networks, which are multi-scale residual dense network (MRDN) and multi-scale residual dense cascaded U-Net with block-connection (MCU-Net). Both networks are built upon a newly designed multi-scale residual dense block (MRDB), and MCU-Net uses MRDB to connect the encoder and decoder of the U-Net. To better exploit the multi-scale feature of the images, the MRDB adds another branch of atrous spatial pyramid pooling (ASPP) based on residual dense block (RDB). Compared to the skip connection, the block-connection using MRDB can adaptively transform the features of the encoder and transfer them to the decoder of the U-Net. In addition, a novel noise permutation algorithm is introduced to avoid model overfitting. The superior performance of these new networks in removing noise within Bayer images has been demonstrated by comparison results on the SIDD benchmark, and the top ranking of SSIM in the NTIRE 2020 Challenge on Real Image Denoising - Track1: rawRGB.

## 1. Introduction

As a fundamental topic of image processing, image denoising removes the presence of noise, reconstructs the structural content details, and then generates high-quality images. As a key component in many practical applications and commercialization products, such as cameras and smartphones, the research of image denoising attracts attentions from both academic and industry. Traditional image denoising research mainly focuses on removing noise within sRGB data. In a recent decade, due to a better understanding about noise, it is revealed that denoising within Bayer raw data will be much more efficient than denoising within sRGB data. As an example in Fig. 1, for an image signal processing (ISP) pipeline within cameras to render sRGB images from Bayer raw sensor data, a

\* These authors have equal contributions.

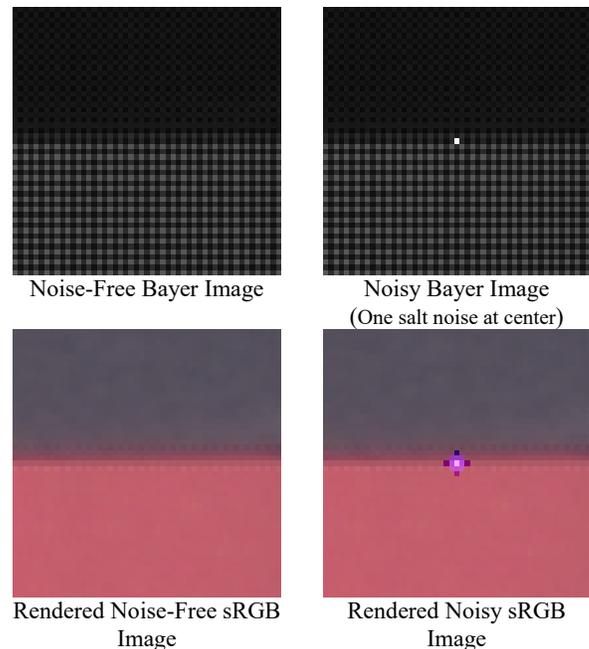


Figure 1: Example to show how an image signal processing (ISP) pipeline (bypassing denoising) makes the denoising problem much more complex and difficult in the sRGB data: one salt noisy pixel at the center of Bayer image will lead to several noisy pixels in the sRGB image.

simple salt noise within a Bayer image will alter other neighbor pixels in the sRGB image if bypassing its denoising process. That's the reason why camera designers think it is simpler and more efficient to remove noise in the Bayer image than reconstructing corrupted pixels in the sRGB image. Hence, this paper focuses on the Bayer image denoising to solve the problem at its early stage.

Most developed sRGB denoising algorithms can be applied in removing the noise within the Bayer data. Compared to traditional handcrafted methods, such as non-local mean [1] and BM3D [2], convolutional neural network (CNN) started a new chapter for the research of image denoising. Recently, the learning-based image denoising methods have achieved remarkable performance thanks to the large image datasets and well-studied deep learning techniques [3]-[8].

Image datasets for denoising can be divided into two categories: synthetic image dataset [5]-[7] and real image dataset [9]-[14], based on the source of the provided noisy images within dataset. Synthetic image dataset is usually built by: 1) first collecting high-quality images as noise-free images by downsampling a high-resolution image or post-processing a low-ISO image [12]; 2) then adding synthetic noise based on statistic noise models (including Gaussian noise model or Poissonian-Gaussian noise model [15]) to get noisy images. Real image dataset is generated in another way: 1) First collecting multiple real noisy images in a short time to ensure minimal image content change, such as scene luminance change or scene object movements; 2) then fusing these multiple images to generate a synthetic noise-free image.

Compared to the synthetic image dataset, the real image dataset is closer to real data processed in practical applications. Hence, this paper focuses on real image denoising. However, even though many researchers and scientists continue in making their efforts to build large real image datasets such as SIDD dataset [16] and DND dataset [12], there is still a challenge of overfitting problem in the learning-based methods due to the limitation of training data size. To handle this challenge, this paper introduces a new noise permutation, which can generate more synthetic noisy data by utilizing real content and real noise information.

For deep learning, many architectures and techniques have been proposed and tested in this research topic [17]-[26]. Most of the state-of-the-art real image denoising networks can be classified into two structures: the residual structure and encoder-decoder structure. The residual structure mainly utilizes spatial features by processing different neural blocks on the input features, whereas the encoder-decoder structure mainly focuses on processing features at different scales. The residual-type methods [19]-[21], such as DnCNN [9], IrCNN [22], and RIDNet [23], focus on learning the difference between the ground-truth image and noisy image. The encoder-decoder structure can be further divided into three types: U-Net [25], Down-up scaling network [26], and Dual-Domain network [3]. U-Net follows its original design by using multiple downsamplings and upsamplings to capture multi-scale features. Down-up scaling network has only one downsampling and one upsampling, and depends on a complex backbone network to restore information. Different from the first two structures in the spatial domain, the Dual-Domain network utilizes multiple U-Net to exploit the features in both spatial and frequency domains. Even though many different encoder-decoder structures have been proposed, the cascaded U-Net structure has not been fully explored.

Besides new network structure designs, researchers also pay attention to the neural network block design. Instead of using traditional convolution layer blocks, existing state-of-

the-art methods utilize the complex residual dense block (RDB) [27], which is inspired by the residual network and dense network. RDB is further extended to be dense connected residual block (DCR) by removing the catenation layer in RDB [17]. Furthermore, several RDBs can be rearranged to build a more complex group residual dense block (GRDB) [26]. All these methods show good performance in NTIRE 2019 Real Image Denoising Challenge [3]. However, these blocks didn't take the multi-scale feature into consideration. Hence, this paper proposes a new multi-scale residual dense block (MRDB), inspired by atrous spatial pyramid pooling (ASPP) [28] and RDB, which shows good performance in reconstructing the texture details while removing the noise.

Overall, this paper introduces two new real-image denoising networks, MRDN and MCU-Net. The novelty of these two networks includes: 1) using MRDB for the multi-scale feature in the neural block design; 2) using the block-connection to replace the skip connection for the multi-layer feature; 3) using noise permutation for data augmentation to avoid model overfitting. All these new methods and networks have been demonstrated their excellent performance for Bayer image denoising in experimental and comparison results based on the SIDD benchmark and the NTIRE 2020 Real Image Denoising Challenge-Track 1: rawRGB.

## 2. Proposed Methods

### 2.1. Multi-scale Residual Dense Network

The Multi-scale Residual Dense Network (MRDN) is based on a new basic module, the Multi-scale Residual Dense Block (MRDB), as shown in Fig. 2 (a). MRDB combines multi-scale features from the ASPP and other features from the traditional residual dense block (RDB).

As shown in Fig. 2 (b), the ASPP [28] contains four parallel network blocks including *conv 1×1*, *conv Rate 6*, *conv Rate 12* and *pooling*. The *conv Rate 6* and *conv Rate 12* denote the 3×3 dilated convolutions with the dilation rate of 6 and 12, respectively. *Conv Rate 6*, *conv Rate 12* and image *pooling* can well capture the multi-scale features of the block input. The features outputted from the ASPP are concatenated and compressed to be combined with other features from the RDB. To have a seamless local residual connection, this concatenated feature is compressed with another *conv 1×1* before an element-wise adder.

The output of the MRDB preserves the same number of channels of its input to avoid the exponential complexity increase. With the MRDB as a building module, the MRDN constructs the network using the similar way as the residual dense network (RDN) [26] by cascading the MRDBs with dense connections. Specifically, the outputs of the MRDBs are concatenated and compressed with a *conv 1×1*, and a global residual connection is adopted to obtain clean features.

## 2.2. MRDB Cascaded U-Net with Block-Connection

The traditional U-Net utilizes the skip connection to jump over layers across the encoder and decoder. Instead of the skip connection, the multi-scale residual dense cascaded U-Net with block-connection (MCU-Net) in Fig. 3 uses

MRDB as the block-connection shown in Fig. 3 (b). This block-connection using MRDB can adaptively transform the features of the encoder and transfer them to the decoder of the U-Net. Also, to enrich its capability and robustness, the MCU-Net adopts a cascaded structure in Fig. 3 (a).

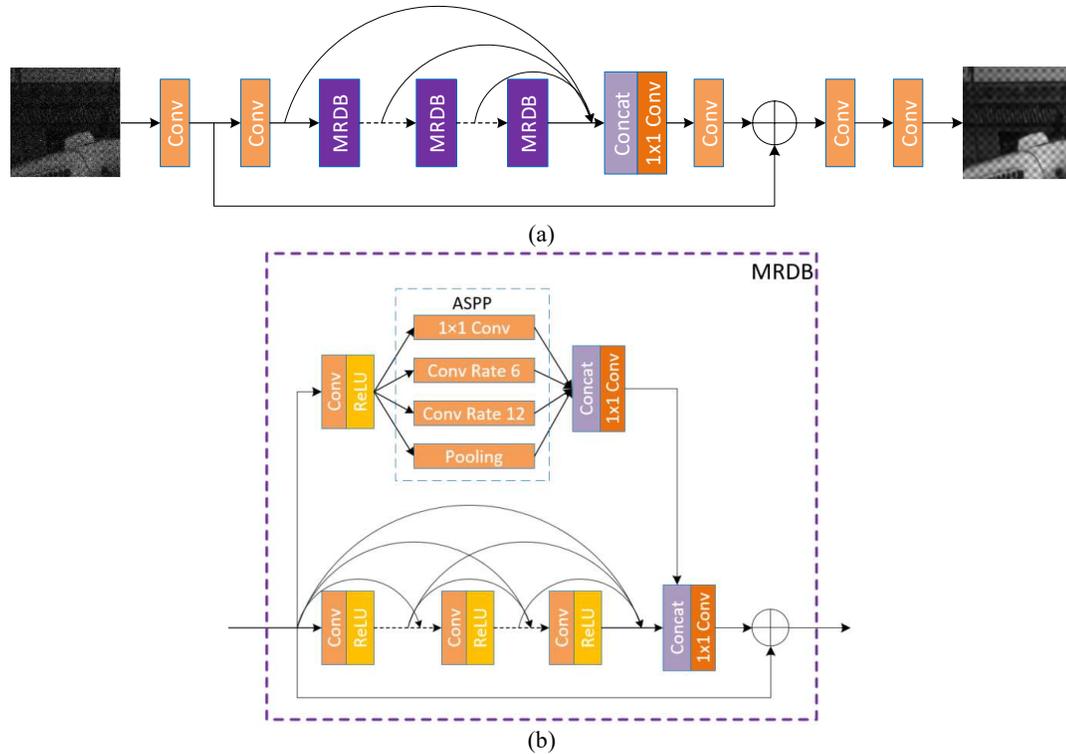


Figure 2: Multi-scale Residual Dense Network (MRDN): (a) Diagram of MRDN; (b) Diagram of MRDB.

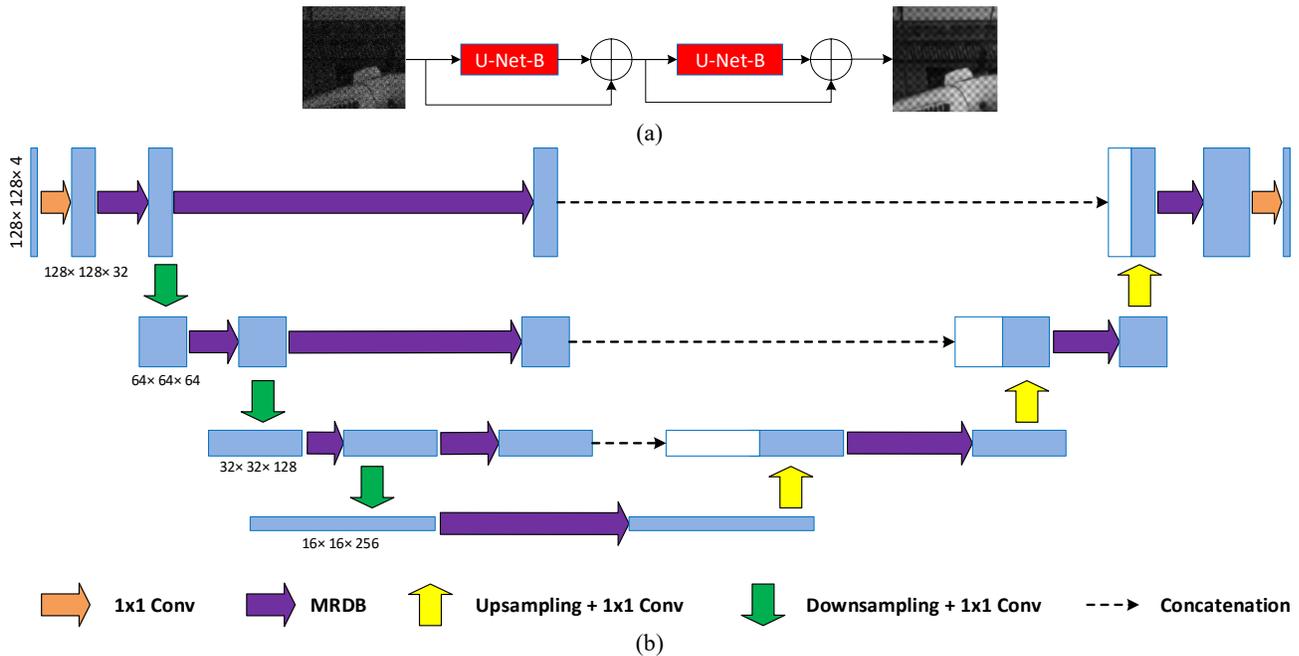


Figure 3: Multi-scale residual dense Cascade U-Net with Block-connection (MCU-Net): (a) MCU-Net; (b) U-Net with Block-connection (U-Net-B).

Each U-Net with the block-connection (U-Net-B) has three scales by using three downsamplings and upsamplings. The  $1\times 1$  convolutional layer is used to compress or expand the number of feature channels. To enforce the network only learning the difference between the input and output, a residual connection is applied. By this way, the network is able to learn how to cancel the presence of noise and get clean images.

### 2.3. Noise Permutation

Data augmentation is an efficient technique to help neural networks to avoid the overfitting problem. The commonly used luminance/contrast/saturation jittering will change the noise characteristics of real noisy images, which should be avoided for image denoising. Further, other applicable image augmentations, such as image flipping/rotation, cannot be directly utilized due to the special property of Bayer data. These traditional data augmentation will generate low-quality images because of mismatched Bayer patterns after augmentation [30]. To handle this problem, Bayer augmentation was proposed [30], which only takes content augmentation into consideration. However, the noise diversity is as important as the content diversity. Hence, some researchers developed learning-based methods including cERGAN generator [26] and Noise Flow [31]. Different from these two approaches generating artificial noise for noise-free images, this paper introduces a new data permutation to utilize real noise from real noisy images. By changing the spatial distribution of real noise, more training samples are generated with real content and noise.

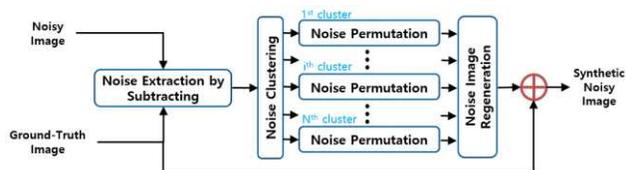


Figure 4: Framework of noise permutation.

As shown in Fig. 4, the first step of this method is to generate the noise image data by subtracting the ground-truth image from its corresponding noisy image. For the noise data, a noise-clustering process divides the data into  $N$  clusters based on their corresponding ground-truth intensity values. Then, within each cluster, a random permutation is performed to swap the positions of those noises. After the permutation, a new synthetic noise image is generated and added back to its corresponding ground-truth image to generate a new synthetic noisy image.

The advantages of this noise permutation include: 1) it doesn't introduce artificial noise based on some statistical noise models; 2) it largely preserves the signal dependency property of the noise in the rawRGB space with proper  $N$ ; 3) it provides more training samples with different near-real

noisy images for a given ground-truth image. Hence, this methods shows benefits in avoiding the model overfitting.

## 3. Experimental Results

### 3.1. Datasets

We used training images released by the NTIRE 2020 Real Image Denoising Challenge-Track1: rawRGB, which are from the SIDD dataset [32]. These training images come from 160 different scene instances, each scene instance has two pairs of high-resolution images and each pair includes one noisy image and its corresponding ground-truth image. In total, there are 320 training image pairs. These images were captured in different physical environments by different smartphone cameras including Samsung Galaxy S6 Edges, Apple iPhone7, Google Pixel, Motorola Nexus 6 and LG G4. For these images, we divided them into the training group (302 images, 151 scene instances) and the validation group (18 images, 9 scene instances).

A smaller input size of our network during training leads to a faster speed and a lower memory requirement of GPUs. Hence, instead of feeding an entire image to our network, we extracted different  $256\times 256$  Bayer patches from each high-resolution image. These patches come from dividing the entire image directly and additional 25 random croppings. All these patches will be rearranged into the same Bayer pattern [30]. To test the performance of the trained model, we downloaded the SIDD benchmark data from the SIDD official website. These SIDD benchmark data were extracted from another 40 noisy images provided by the SIDD benchmark organizers. For each image, they extracted 32 patches with the size of 256 by 256 as benchmark data. For these downloaded SIDD benchmark patch data, we generated their corresponding denoised results with our trained model. By submitting the denoised results to the SIDD benchmark website, we received the PSNR and SSIM metrics reports.

### 3.2. Implementation Details

The networks are trained with Adam optimizer [33] with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , using  $L_1$  loss. The learning rate is set to 0.0001, and the weight decay parameter is  $10^{-8}$ . The initial training was performed on the whole dataset to get the pretrained model. With the pretrained model, we selected the hard patches whose PSNR is smaller than 50 dB to further fine-tune the model with learning rate  $10^{-5}$ . In addition, the Bayer data augmentation such as flipping and/or rotating image and noise permutation were adopted during the training. We used Python with PyTorch to implement these networks and trained them on two Nvidia Tesla V100 GPUs with the batch size as 8. Approximately, MCU-Net took 3~4 days for training and MRDN took ~7 days for training.

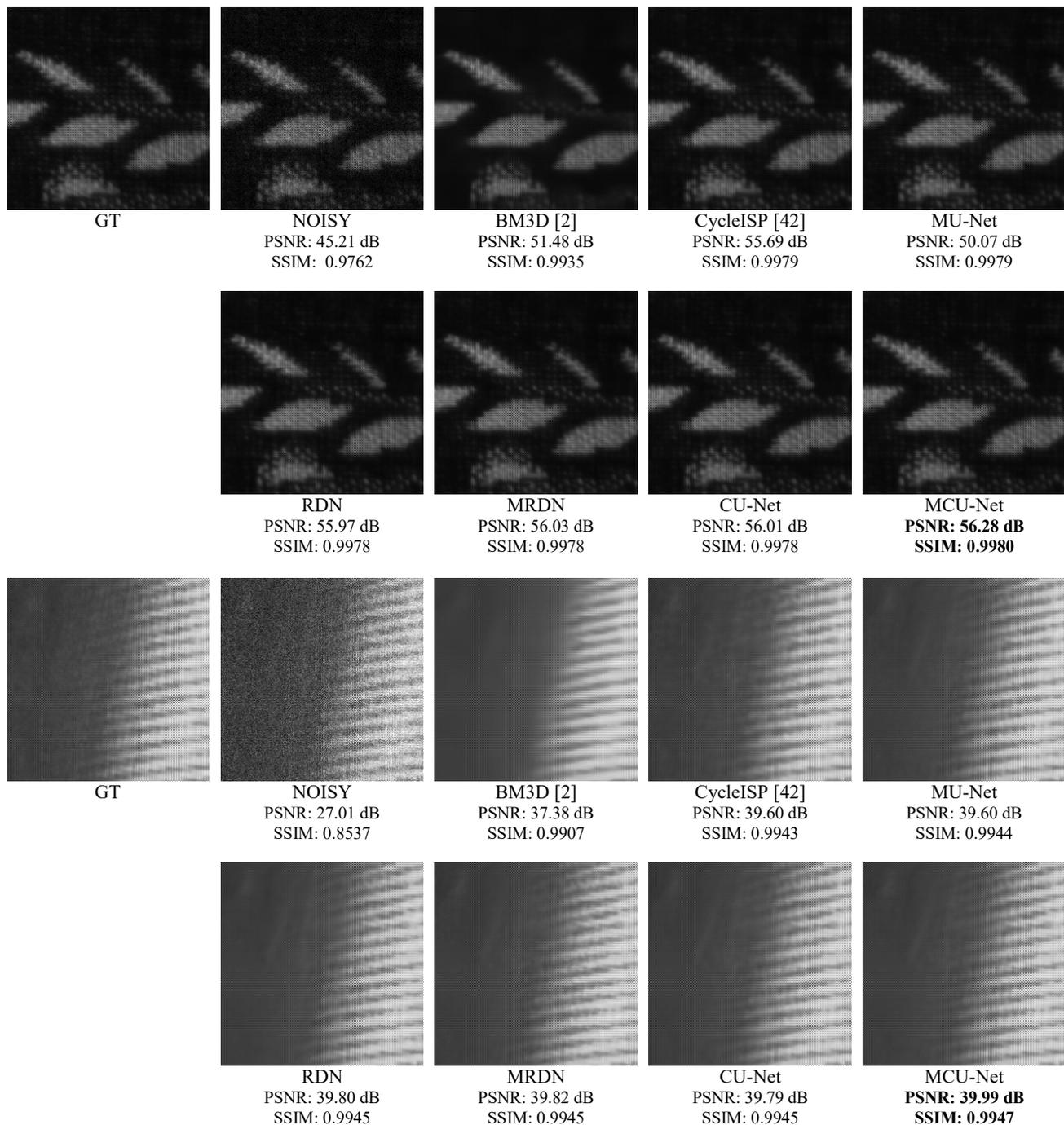


Figure 5: Comparison results of BM3D [2], CycleISP [42], RDN, MRDN, CU-Net, MU-Net and MCU-Net (To better visualize the dark Bayer image, their intensity values are multiplied by a scale value for proper visualization; PSNR and SSIM are calculated on Bayer images; and top ranking in PSNR/SSIM is in **bold**).

### 3.3. Experimental Results

This paper focuses on Bayer image denoising. But the Bayer images are too dark to be displayed. Hence, to visually check the denoised result, we present our denoised results in two ways.

One way is to scale up the intensity values within Bayer image data, as shown in Fig. 5. Another way is to utilize a simple and light-weight camera image signal processing (ISP) pipeline to render the sRGB images from the Bayer images, as shown in Fig. 6. This ISP pipeline code is provided by the SIDD dataset, which can be directly

downloaded from its official website. PSNR and SSIM are also calculated based on the Bayer raw data for Fig. 5 and sRGB data for Fig. 6.

From the images in Figs. 5 and 6, one can observe that the proposed MRDN and MCU-Net models can

successfully remove the noise while maintaining the detail structural and texture details, and generate the near-ground-truth (GT) images. These can also be verified by the high PSNR and SSIM metrics of MCU-Net and MRDN.



Figure 6: Comparison results of BM3D [2], CycleISP [42], RDN, MRDN, CU-Net, MU-Net and MCU-Net (To better visualize the dark Bayer image, a simplified image singal processing pipeline system from SIDD dataset [16] is applied on these denoised Bayer images to render their sRGB images; PSNR and SSIM are calculated on sRGB images; and top ranking in PSNR/SSIM is in bold).

Table 1: Experiments of MRDB vs RDB on SIDD benchmark.

	MRDN	RDN	MCU-Net	CU-Net
PSNR on raw (dB)	48.68	48.67	48.80	48.73
SSIM on raw	0.990	0.990	0.990	0.990
PSNR on sRGB (dB)	36.42	36.41	36.54	36.45
SSIM on sRGB	0.875	0.874	0.875	0.874

Table 2: Experiments of block-connection vs skip connection in U-Net framework on SIDD benchmark.

	MCU-Net (Block-connection)	MU-Net (Skip connection)
PSNR on raw (dB)	48.80	48.71
SSIM on raw	0.990	0.990
PSNR on sRGB (dB)	36.54	36.43
SSIM on sRGB	0.875	0.874

Table 3: Comparison results in SIDD benchmark (top 3 in **bold**, top 1 in **bold** and Underline).

	PSNR on raw (dB)	SSIM on raw	PSNR on sRGB (dB)	SSIM on sRGB
EPLL [40]	40.73	0.935	25.19	0.842
GLIDE [38]	41.87	0.949	25.98	0.816
KSVD-G [41]	42.50	0.969	28.13	0.781
KSVD-DCT [41]	42.70	0.970	28.21	0.784
TNRD [39]	42.77	0.945	26.99	0.744
LPG-PCA [35]	42.79	0.974	30.01	0.854
FoE [36]	43.13	0.969	27.18	0.812
MLP [21]	43.17	0.965	27.52	0.788
KSVD [34]	43.26	0.969	27.41	0.832
DnCNN [9]	43.30	0.965	28.24	0.829
NLM [1]	44.06	0.971	29.39	0.846
WNNM [37]	44.85	0.975	29.54	<b>0.888</b>
BM3D [2]	45.52	<b>0.980</b>	30.95	0.863
CycleISP [42]	47.93	<b>0.985</b>	35.44	0.856
aRID	<b>48.05</b>	<b>0.980</b>	<b>35.78</b>	<b>0.902</b>
MRDN	<b>48.68</b>	<u>0.990</u>	<b>36.42</b>	<b>0.875</b>
MCU-Net	<u>48.80</u>	<u>0.990</u>	<u>36.54</u>	<b>0.875</b>

### 3.4. Experiments on MRDB vs RDB

As one of the main contributions within this paper, the MRDB is proposed to introduce the multi-scale feature in the traditional residual dense block. To demonstrate the performance of the MRDB, two comparison groups are created. For MRDN, we replaced its MRDB with RDB to get a comparable RDN. For MCU-Net, we did the same replacement and name it as CU-Net for the ablation study.

To evaluate their performance, SIDD benchmark results are provided in Table 1. From this table, MRDN has 0.01 dB gain on raw PSNR compared to RDN, and MCU-Net has almost 0.07 dB gain on raw PSNR compared to CU-

Net.

Beside the comparison among the Bayer raw data, we also render their corresponding sRGB data by using SIDD's ISP pipeline and calculate the PSNR and SSIM metrics on sRGB data. From Table 1, MRDN has 0.01 dB gain on sRGB PSNR compared to RDN, and MCU-Net has almost 0.09 dB gain on sRGB PSNR compared to CU-Net.

These gains demonstrate the advantage of MRDB over RDB. This advantage can be also observed among the results of MRDN vs RDN and MCU-Net vs CU-Net in Figs. 5 and 6.

### 3.5. Experiments on Block-Connection vs Skip Connection

As another contribution of this paper, the block-connection is introduced to replace the traditional skip connection for better performance in extracting features. To demonstrate its advantage, we designed an ablation test by comparing MU-Net and MCU-Net. Here, MU-Net is modified based on MCU-Net by replacing the block-connection with the skip connection. Both networks are evaluated over the SIDD benchmark.

The test results are shown in Table 2, where one can see that the block-connection can bring in 0.09 dB gain over the skip connection in raw PSNR and 0.12 dB in sRGB PSNR. Similar improvements can be visually observed among the denoised images by MCU-Net and MU-Net in Figs. 5 and 6.

### 3.6. Comparison Results

To evaluate the performance of the proposed MCU-Net and MRDN, their SIDD benchmark scores are shown in Table 3 together with the scores of other prior arts, which include BM3D [2], NLM [1], KSVD [34], LPG-PCA [35], FoE [36], MLP [21], WNNM [37], GLIDE [38], TNRD [39], EPLL [40], DnCNN [9], KSVD [41], CycleISP [42] and aRID. Here, aRID is a new submission to SIDD benchmark website without disclosing its detail. For the methods whose SIDD benchmark score have been published, we directly copied their scores from the SIDD website<sup>1</sup>. For CycleISP, it has no published score on the website, and its score is calculated by the benchmark website based on the results generated by the pretrained model and testing script they shared in their official GitHub website<sup>2</sup>.

To highlight the ranking, the top 3 methods are marked in bold and the top method is underlined. From Table 3, MCU-Net and MRDN achieve better performance than prior arts on the SIDD benchmark data. This can be also justified by the visual comparison of results from our models, CycleISP [42] and BM3D [2] in Figs. 5 and 6.

<sup>1</sup> <https://www.eecs.yorku.ca/~kamel/sidd/benchmark.php>.

<sup>2</sup> <https://github.com/swz30/CycleISP>

### 3.7. NTIRE 2020 Challenge on Real Image Denoising - Track1: rawRGB

Beside the SIDD benchmark score, we also tested our new network models on the validation data of NTIRE 2020 Challenge on Real Image Denoising - Track1: rawRGB [32]. Due to the limited number of submissions allowed for each user, we only tested our new models and an ensemble model of RDN, MRDN and CU-Net. The received metrics including PSNR and SSIM are shown in Table 4. These high PSNR and SSIM scores show superior performance of our models. Especially, the ensemble model achieves the top ranking of SSIM in the NTIRE 2020 Challenge on Real Image Denoising - Track1: rawRGB [32].

Table 4: PSNR and SSIM evaluations of our methods on the validation data of NTIRE 2020 Challenge on Real Image Denoising - Track1: rawRGB.

Method	PSNR (dB)	SSIM
MRDN	52.66	0.9960
MCU-Net	52.63	0.9960
Our ensemble model	<b>52.75</b>	<b>0.9960</b>

### 4. Conclusion

This paper proposed two new networks, which are MRDN and MCU-Net, based on new MRDB, and new block-connection within U-Net. Also, to avoid the model overfitting, a novel noise permutation is proposed to generate synthetic noisy images which combine real content information of the ground-truth images and noise information of noisy images with different spatial distribution. Experimental and comparison results on the SIDD and NTIRE 2020 Challenge on Real Image Denoising - Track1: rawRGB demonstrate the superior performance of these new approaches in generating high-quality denoised Bayer images.

### References

- [1] A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.
- [2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE TIP*, 16(8): 2080-2095, 2007.
- [3] A. Abdelhamed, et al. NTIRE 2019 challenge on real image denoising: Methods and results. In *CVPR Workshops*, 2019.
- [4] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron. Unprocessing images for learned raw denoising. In *CVPR*, 2019.
- [5] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, et al. NTIRE 2017 challenge on single image superresolution: Methods and results. In *CVPR Workshops*, 2017.
- [6] 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.
- [7] R. Franzen. Kodak lossless true color image suite. *Source: http://r0k.us/graphics/kodak*.
- [8] S. Gu, Y. Li, L. Gool, and R. Timofte. Self-guided network for fast image denoising. In *ICCV*, 2019.
- [9] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE TIP*, 26(7): 3142-3155, 2017.
- [10] J. Anaya and A. Barbu. Renoir - a dataset for real low-light noise image reduction. *Journal of Visual Communication and Image Representation*, 51: 144-154, 2018.
- [11] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *CVPR*, 1683-1691, 2016.
- [12] T. Plotz, and S. Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017.
- [13] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019.
- [14] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang. Real-world noisy image denoising: A new benchmark. *IEEE TPAMI*, 2018.
- [15] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE TIP*, 2008.
- [16] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018.
- [17] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image restoration. *preprint arXiv:1812.10477*, 2020.
- [18] S. Zini, S. Bianco, and R. Schettini. Deep residual autoencoder for quality independent JPEG restoration. *preprint arXiv:1903.06117*, 2019.
- [19] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *CVPR*, 2017.
- [20] S. Anwar, C. P. Huynh, and F. Porikli. Chaining identity mapping modules for image denoising. *preprint arXiv:1712.02933*, 2017.
- [21] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, 2012.
- [22] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep CNN denoiser prior for image restoration. In *CVPR*, 2017.
- [23] S. Anwar, and N. Barnes. Real image denoising with feature attention. In *ICCV*, 2019.
- [24] M. Haris, G. Shakhnarovich, and N. Ukita. Deep backprojection networks for super-resolution. In *CVPR*, 2018.
- [25] B. Park, S. Yu, and J. Jeong. Densely connected hierarchical network for image denoising. In *CVPR*, 2019.
- [26] D. W. Kim, J. Ryun Chung, and S. W. Jung. GRDN: Grouped residual dense network for real image denoising and GAN-based real-world noise modeling. In *CVPR*, 2019.
- [27] Y. Zhang, Y. Tian, Y. Kong, B. Zhong and Y. Fu. Residual dense network for image super-resolution. In *CVPR*, pp. 2472-2481, 2018.
- [28] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 801-818, 2018.
- [29] X. Xia and B. Kulis. W-Net: A deep model for fully unsupervised image segmentation. *arXiv:1711.08506*, 2017.

- [30] J. Liu, C. H. Wu, Y. Wang, Q. Xu, Y. Zhou, H. Huang, C. Wang, S. Cai, Y. Ding, H. Fan and J. Wang. Learning raw image denoising with bayer pattern unification and bayer preserving augmentation. In *CVPR*, 2019.
- [31] A. Abdelhamed, M. A. Brubaker and M. S. Brown. Noise Flow: Noise modeling with conditional normalizing flows. In *ICCV*, 2019.
- [32] A. Abdelhamed, et al., NTIRE 2020 challenge on real image denoising: Dataset, methods and results. In *CVPR Workshops*, 2020.
- [33] D. P. Kingma, and J. Ba. Adam: A method for stochastic optimization. *preprint arXiv:1412.6980*, 2014.
- [34] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE TSP*, 54(11):4311–4322, 2006.
- [35] L. Zhang, W. Dong, D. Zhang, and G. Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43(4):1531–1549, 2010
- [36] S. Roth and M. J. Black. Fields of experts. *IJCV*, 82(2):205–229, 2009
- [37] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014.
- [38] H. Talebi and P. Milanfar. Global image denoising. *IEEE TIP*, 23(2):755–768, 2014.
- [39] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE TPAMI*, 39(6):1256–1272, 2017.
- [40] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011.
- [41] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE TIP*, 15(12):3736–3745, 2006.
- [42] S. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao. CycleISP: Real image restoration via improved data Synthesis. In *CVPR*, 2020.