IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 2, 2019, accepted August 31, 2019, date of publication September 5, 2019, date of current version September 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939578

Color Filter Array Demosaicking Using Densely Connected Residual Network

BUMJUN PARK[1] AND JECHANG JEONG[2], (Member, IEEE)
[1]Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, South Korea
[2]Department of Electronic Engineering, Hanyang University, Seoul 04763, South Korea

Corresponding author: Jechang Jeong (jjeong@hanyang.ac.kr)

This work was supported by the research fund of Signal Intelligence Research Center supervised by the Defense Acquisition Program Administration and Agency for Defense Development of Korea.

ABSTRACT Deep convolutional neural networks have been used extensively in recent image processing research, exhibiting drastically improved performance. In this study, we apply convolutional neural networks to color filter array demosaicking, which plays an essential role in single-sensor digital cameras. Contrary to conventional convolutional neural network-based demosaicking models, the proposed model does not require any initial interpolation step for mosaicked input images, which increases the computational complexity. Using a mosaicked image as input, the proposed model is trained in an end-to-end manner to generate demosaicked images outputs. Many deep neural networks experience vanishing-gradient problem, which makes models hard to be trained. To solve this problem, we apply residual learning and densely connected convolutional neural network. Moreover, we apply block-wise convolutional neural networks to consider local features. Finally, we apply a sub-pixel interpolation layer to generate demosaicked output images more efficiently and accurately. Experimental results show that our proposed model outperforms conventional solutions and state-of-the-art models.

INDEX TERMS Demosaicking, color filter array interpolation, deep learning, convolutional neural network.

I. INTRODUCTION

Digital color image pixels consist of three color components: red, green, and blue. To obtain exact information of these components, digital cameras need three sensors, which make cameras expensive and bulky. For this reason, most digital cameras use a single-sensor architecture with a color filter array (CFA). After penetrating the CFA, the sensor of the camera only takes one color information per pixel, according to the arrangement of the color in the CFA. There are many CFA patterns, but the Bayer pattern [1] is the most widely used. Figure 1 shows an example of the Bayer pattern. The number of green pixels in the Bayer pattern is twice that of red or blue pixels. This is because the human eye perceives spatial details chiefly from luminance information, and the luminosity function is similar to the CIE 1931 green matching function [2]. As there is only information of one color per pixel in single-sensor digital cameras, information of the other two colors should be interpolated. This interpolation process, referred to as demosaicking, is CFA interpolation.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin.

Many demosaicking algorithms have been proposed [3]–[5]. In the past, demosaicking was implemented with simple interpolation algorithms: nearest neighbor [6], bilinear [7], or bicubic interpolation [8]. However, these simple interpolations resulted in many false color artifacts: blurring, chromatic aliases, zippering, and purple fringing [9]. To overcome this, Zhang and Wu proposed color demosaicking via directional linear minimum mean square-error estimation (DLMMSE) [10]. Dengwen and Xiaoliu proposed colour demosaicking with directional filtering and weighting (DDFW) [11]. Pekkucuksen and Altunbasak proposed edge strength filter based demosaicking (ESF) [12] and multi gradients-based demosaicking (MSG) [13]. Recently, many demosaicking algorithms using residual interpolation have been proposed. Monno *et al.* proposed adaptive residual interpolation for demosaicking (ARI) [14] and Kim and Jeong proposed four-direction residual interpolation for demosaicking (FDRI) [15].

Despite many proposals, there still existed false color artifacts in the result images of conventional algorithms. Recently, deep convolutional neural networks (CNNs) have been applied to many image processing algorithms, including

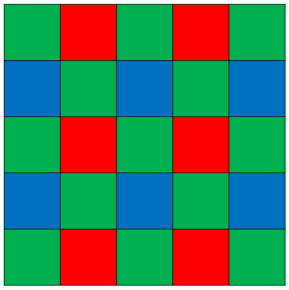


FIGURE 1. Example of the Bayer pattern.

classification [16], [17], super resolution [18]–[20], high dynamic range [21], deblurring [22], denoising [23]–[25], dehazing [26], and deraining [27], and exhibited significantly improved performance. There are also many proposed demosaicking methods based on CNN architecture [28]–[30]. Tan *et al.* proposed color image demosaicking via deep residual learning (DDRL) [31] and Tan *et al.* proposed image demosaicking via multiple CNN (DDemo) [32].

Inspired by these CNN-based image processing solutions, we propose a CFA demosaicking method using a densely connected residual network (DRDN). Many conventional demosaicking solutions based on CNN architecture require substantial pre-processing: initial interpolation, initial demosaicking, and training image clustering [32]. Initial interpolation and initial demosaicking are applied by almost every conventional CNN-based demosaicking solution to equalize the resolution of the input image with the output image. Although they allow CNN models to easily generate output images with the desired resolution, they drastically increase the computational complexity and the memory use of the CNN model, as it receives three times more input data with two times higher resolution. Furthermore, the pre-processing method itself considerably consumes the computational complexity if conventional demosaicking algorithms are applied for their initial demosaicking process. The clustering of the training images also increases the computational complexity, as it needs to train the CNN model for each cluster, even after the clustering process.

To overcome these issues, our proposed model is trained in an end-to-end manner, which does not require any initial interpolation or demosaicking process to generate demosaicked output images with the desired resolution. Instead of the initial interpolation or demosaicking process, the proposed model divides the mosaicked image into four color layers, which are a quarter of the mosaicked image for the input of the proposed network. The layers consist of one red layer, one blue layer, and two green layers. There are twice as many green layers as red or blue layers, because the Bayer pattern [1] has twice as many green pixels as red or blue pixels. This input data modification enables our proposed network to consume less computational complexity and memory than conventional demosaicking networks. Then, we train our proposed CNN model to generate demosaicked images with the same size as the original mosaicked image. To generate demosaicked images more accurately and efficiently, we apply a sub-pixel interpolation layer [33] that learns to generate demosaicked images with desired resolution while our proposed CNN model is trained. Moreover, our proposed DRDN applies the residual learning [17] on the densely connected CNN [16] to avoid the vanishing-gradient problem, and block-wise CNN to consider local features. Finally, we apply a self-ensemble method [19], [34]. It enables our proposed network to exhibit better performance by applying an ensemble method without additional training or clustering processes.

The rest of this paper is organized as follows. Section II briefly describes the conventional networks that inspired us, and summarizes our contributions. Section III describes our proposed CNN model in detail. Section IV shows how we trained our proposed model and present the experimental results. Finally, section V outlines the conclusion of this study.

II. RELATED WORKS

A. CONVENTIONAL DEMOSAICKING NETWORKS

A lot of effort has been made to apply the CNN architecture to demosaicking. Tan et al. proposed the DDRL, which estimates the intermediate green channel to use it as a guidance for the reconstruction of the red and blue channels. However, the DDRL interpolates the input CFA image by bilinear interpolation [7] to make the resolution of the input image same with the desired output image resolution. This causes their network to consume more memory and computational complexity than they need, as they force their network to handle larger resolution feature maps. Tan et al. proposed the DDemo, which uses initial demosaicking and clustering methods for their network. As mentioned above, the initial demosaicking method itself consumes a lot of computational complexity. Besides, it also makes their network more computationally complex, and consumes more memory. Moreover, the clustering of the training images makes their method even more complex, because they need to train a model for each image cluster while clustering itself also consumes computational complexity. To reduce the unnecessary computational complexity, we modify the input mosaicked image into four color layers, which are a quarter the size of the original image. Because there are twice as many green pixels than red or blue pixels, we generate two green layers, one red layer, and one blue layer. Owing to the reduction of the input images resolution due to the input image modification, our proposed network can use memory efficiently, and considerably reduce the computational complexity.

B. RESIDUAL DENSE NETWORK FOR IMAGE RESTORATION

He et al. proposed deep residual learning for image recognition (ResNet) [17]. As CNN architectures became deeper, many models experienced the vanishing-gradient problem.

ResNet solves this problem by applying a skip connection that enables the network to learn residual functions with reference to the input layers. In the following year, Huang *et al.* proposed densely connected convolutional networks (DenseNet) [16]. They connected each layer to every other layer in a feed-forward fashion to solve the vanishing-gradient problem and induce the network to reuse the information of the previous layers. Recently, Zhang *et al.* proposed residual dense network for image restoration (RDN) [20], which combines the idea of ResNet and DenseNet. They applied global residual learning to the DenseNet and local residual learning to the dense block to solve the vanishing-gradient problem. Inspired by these models, we propose the DRDN, which is optimized for demosaicking.

C. SUB-PIXEL INTERPOLATION

Shi *et al.* proposed super resolution using efficient sub-pixel convolutional neural network (ESPCN) [33]. Their proposed sub-pixel interpolation layer learns an upscaling filters to upscale their final output. This enables their network to reduce the computational complexity, as the input of their network is an original low resolution image instead of an interpolated high resolution image, owing to the sub-pixel interpolation layer. Inspired by this model, we apply their sub-pixel interpolation layer to reduce the computational complexity and increase accuracy of the interpolation.

D. SELF-ENSEMBLE METHOD

The self-ensemble method was introduced in [19] and [34]. In the conventional methods, they use a multi-model structure to apply the ensemble method [32]. However, the multi-model structure costs a lot of time and computational complexity, as it needs to train additional networks. On the other hand, the self-ensemble method, which averages the outputs of the transformed input images, only needs one trained network. By applying the self-ensemble method, we can increase the performance of our proposed network without training additional networks.

E. CONTRIBUTIONS

In this paper, we propose an input data modification method that enables our proposed network to avoid using initial interpolation or demosaicking processes, and reduce the computational complexity and memory consumption. Next, we propose a demosaicking network that combines the idea of ResNet [17] and DenseNet [16] to solve the vanishing-gradient problem while successfully interpolating the missing pixels of the mosaicked image. Moreover, our network applies the idea of the sub-pixel interpolation layer [33] to demosaicking solution, which enables our proposed network to generate demosaicked images more efficiently and accurately. Finally, we apply the self-ensemble method, which shows significant performance enhancement without additional time or computational complexity consumptions.

FIGURE 2. Data flow of the proposed demosaicking solution.

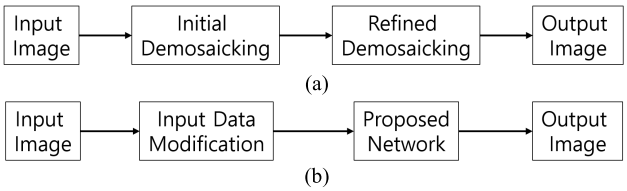


FIGURE 3. Demosaicking process of conventional methods and the proposed method: (a) conventional methods and (b) the proposed method.

III. PROPOSED METHOD

Conventional demosaicking solutions that apply CNN architectures consist of two steps: initial interpolation or demosaicking and a CNN network that refines the result of the initial demosaicking. However, the initial interpolation or demosaicking of the input image not only consumes considerable computational complexity itself, but also increases the computational complexity and memory use of the CNN network. To solve this critical problem, we propose an input data modification method that enables our proposed network to avoid applying initial interpolation or demosaicking process. The proposed input data modification process is discussed in subsection A. After the modification of the input data, our proposed CNN model generates demosaicked images. The details of our proposed network will be discussed in subsection B. Figure 2 shows a data flow of the proposed demosaicking solution, and Figure 3 shows a comparison of the process between the conventional demosaicking solutions and the proposed method.

A. INPUT DATA MODIFICATION

As shown in Figure 1, mosaicked images with the Bayer pattern have information of one color channel per pixel, and there are twice as many green pixels as red or blue pixels. By reordering these pixels, we generate four color layers that have a quarter of the size of the original mosaicked image. Accordingly, our input data modification process generates two green layers, one red layer, and one blue layer. Using this input data modification, we can reduce the computational complexity and memory consumption of our demosaicking solution in two ways. First, our proposed demosaicking solution can avoid applying initial interpolation or demosaicking process, which considerably consumes computational complexity. Because our input data modification only needs a simple reordering process, we can achieve a drastic computational complexity reduction compared with conventional demosaicking methods. Second, because our input data modification makes our input image into a quarter of the size of the original image, our CNN model can reduce its computational

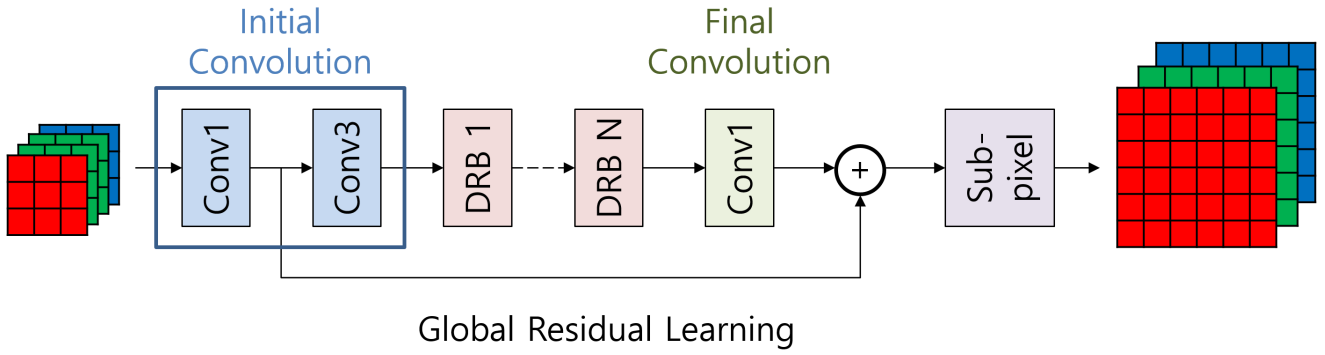


FIGURE 4. Structure of the proposed network. Conv1 denotes 1 × 1 convolution layer and conv3 denotes 3 × 3 convolution layer.

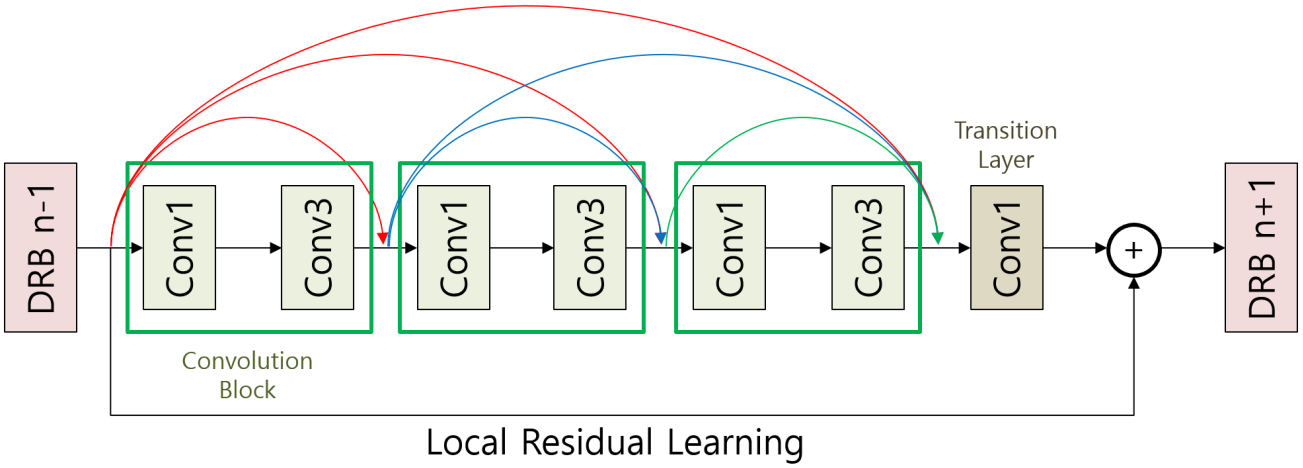


FIGURE 5. Structure of the proposed densely connected residual block. Red, blue, and green arrows indicate the dense connectivity.

complexity and memory consumption. Because the resolution of the input image is reduced, the size of the feature maps is also reduced, which results in a reduction of the memory consumption for storing the information of the feature maps. Moreover, the number of convolution operations also reduces, which determines the computational complexity and training time of the network. Therefore, we can reduce the computational complexity and memory consumption of our proposed network.

B. NETWORK STRUCTURE

Inspired by [16], [17], and [20], our proposed DRDN consists of four parts: initial convolution block, densely connected residual blocks (DRBs), final convolution layer, and sub-pixel interpolation layer. Figure 4 shows the structure of our proposed DRDN, where conv1 indicates 1 × 1 convolution layers and conv3 indicates 3 × 3 convolution layers. Note that there exist activation functions after every convolution layer. The initial convolution block consists of two convolution layers: a 1 × 1 convolution layer and a 3 × 3 convolution layer. The 1 × 1 convolution layer generates 12 feature maps to apply global residual learning before the sub-pixel interpolation layer. Then, the 3 × 3 convolution layer generates 64 feature maps for DRB. After the convolution operations of the initial convolution block, our proposed DRDN proceeds N DRBs. The structure of the DRB will be discussed in subsection C. Next, our proposed network proceeds to the final convolution layer, which produces the output of the DRBs into 12 feature maps for the sub-pixel interpolation layer. Finally, the sub-pixel interpolation layer generates demosaicked images with the desired resolution.

C. DENSELY CONNECTED RESIDUAL BLOCK

Inspired by [16], [17], and [20], our proposed DRB consists of two parts: convolution blocks and a transition layer. Figure 5 shows the structure of our proposed DRB. Note that there exist activation functions after every convolution layer. The convolution blocks consist of two convolution layers, which are the same as the initial convolution block of the proposed network structure. However, in the convolution block, the number of filters of the convolution layers is different. According to [16], the 3 × 3 and 1 × 1 convolution layers generate k and 4k feature maps, respectively, where k denotes the growth rate of the DRB. In our proposed DRB, there exist three convolution blocks, whose outputs are connected by dense connectivity. After extracting the feature maps by

TABLE 1. Size of the convolution layers of the proposed network.

Layers			Size
Initial Convolution		Conv1	$1 \times 1 \times 12$
		Conv3	$3 \times 3 \times 64$
DRB	Convolution	Conv1	$1 \times 1 \times 4k$
		Conv3	$3 \times 3 \times k$
	Transition	Conv1	$1 \times 1 \times 64$
Final Convolution		Conv1	$1 \times 1 \times 12$

the convolution blocks, the transition layer generates 64 feature maps to apply the local residual learning in the DRB. Table 1 summarizes the size of the convolution layers of our proposed DRDN. Finally, the output of the previous DRB is used for the input of the next DRB.

IV. EXPERIMENTS

A. TRAINING DETAILS

As many image processing solutions that apply CNN architecture have been proposed, many datasets have been used for training networks. Recently, Agustsson et al. released the DIV2K training and validation datasets [35], which include high quality images. The DIV2K training dataset consists of 800 images where the resolution of each image is similar to the FHD resolution (1920×1080). The DIV2K validation dataset consists of 100 images, where the resolution of each image is similar to the training dataset. Given the high quality of the images in the DIV2K, many state-of-the-art image processing methods use this dataset and show improved performance. Thus, we train our proposed network with the DIV2K training and validation sets. When training our network, we use patches that is extracted from the training dataset where the width and height of the patches are set to 64 pixels. To augment the training patches, we randomly rotate and flip the input patches before entering the proposed network. We set the batch size of the training patches to 64, and train our proposed network for 300 epochs. We use the Adam optimizer [36] with an initial learning rate of 10^{-4}, and divided it by 10 for every 100 epochs. For the activation function, we used the leaky rectified linear unit (leaky ReLU), where α is set to 0.1. We use the mean square error for the loss function. We set the number of DRB (N) to 15 and the growth rate (k) to 32. For the test sets, we use the Kodak [37] and McMaster [38] dataset, which are widely used as test sets for demosaicking solutions. The Kodak dataset consists of 24 images where the resolution of each image is 768×512.

TABLE 2. CPSNR (dB) and SSIM results of the conventional demosaicking methods and the proposed method for the Kodak dataset.

	1		2		3		4		5		6		7		8	
	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM
DLMMSE	38.52	0.9865	42.61	0.9843	40.09	0.9859	43.53	0.9851	34.81	0.9809	37.03	0.9847	39.87	0.9765	43.83	0.9890
DDFW	39.72	0.9904	42.89	0.9872	40.81	0.9888	44.29	0.9889	35.79	0.9854	37.74	0.9872	40.09	0.9791	44.43	0.9912
ESF	39.91	0.9902	42.60	0.9869	40.69	0.9891	43.80	0.9884	36.11	0.9859	36.11	0.9843	39.27	0.9767	44.77	0.9914
MSG	39.79	0.9905	43.30	0.9879	41.23	0.9897	44.32	0.9890	35.96	0.9862	37.77	0.9874	40.22	0.9797	44.58	0.9913
ARI	38.81	0.9863	41.62	0.9812	39.62	0.9837	43.18	0.9863	35.20	0.9806	37.54	0.9839	38.59	0.9700	43.24	0.9888
FDRI	34.75	0.9689	40.80	0.9764	37.13	0.9767	41.44	0.9797	30.43	0.9517	36.18	0.9754	37.90	0.9652	39.79	0.9793
DDRL	41.44	0.9892	43.92	0.9811	41.99	0.9845	45.12	0.9855	37.22	0.9840	39.94	0.9848	41.92	0.9820	45.20	0.9863
DDemo	41.96	0.9931	43.75	0.9869	42.22	0.9899	45.26	0.9894	37.66	0.9903	40.26	0.9904	42.08	0.9842	45.43	0.9917
DRDN	42.70	0.9942	44.52	0.9888	42.64	0.9912	45.54	0.9904	37.83	0.9908	40.69	0.9915	42.80	0.9859	45.73	0.9924
DRDN+	42.86	0.9943	44.64	0.9889	42.79	0.9914	45.68	0.9906	38.02	0.9911	40.82	0.9918	42.88	0.9863	45.91	0.9927
	9		10		11		12		13		14		15		16	
	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM
DLMMSE	41.85	0.9884	37.45	0.9753	40.90	0.9816	40.93	0.9659	41.27	0.9731	39.17	0.9755	38.46	0.9715	43.30	0.9737
DDFW	42.10	0.9893	37.95	0.9799	41.40	0.9854	41.15	0.9768	41.76	0.9760	39.69	0.9809	38.54	0.9751	43.49	0.9802
ESF	41.95	0.9890	37.72	0.9800	41.49	0.9863	40.82	0.9764	41.52	0.9783	40.30	0.9843	38.41	0.9762	42.45	0.9835
MSG	42.46	0.9899	38.22	0.9830	41.99	0.9876	41.59	0.9791	42.13	0.9792	40.26	0.9850	39.00	0.9786	43.89	0.9859
ARI	41.25	0.9866	36.90	0.9751	40.61	0.9818	39.67	0.9693	40.86	0.9759	39.24	0.9747	38.05	0.9730	43.16	0.9813
FDRI	39.26	0.9798	34.46	0.9664	38.06	0.9711	39.03	0.9663	39.30	0.9705	36.40	0.9694	37.18	0.9689	42.47	0.9799
DDRL	42.77	0.9850	38.92	0.9815	42.41	0.9834	41.95	0.9753	42.96	0.9738	41.30	0.9797	40.17	0.9778	44.91	0.9820
DDemo	42.76	0.9887	39.04	0.9847	42.55	0.9880	42.15	0.9799	42.96	0.9810	41.50	0.9853	39.95	0.9821	44.96	0.9872
DRDN	43.17	0.9904	39.45	0.9859	42.89	0.9888	42.64	0.9829	43.27	0.9813	41.65	0.9838	40.69	0.9845	45.42	0.9886
DRDN+	43.32	0.9903	39.43	0.9860	43.04	0.9890	42.68	0.9832	43.37	0.9814	41.81	0.9846	40.85	0.9849	45.48	0.9887
	17		18		19		20		21		22		23		24	
	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM
DLMMSE	35.52	0.9838	42.75	0.9859	41.09	0.9776	38.10	0.9876	40.27	0.9867	42.39	0.9866	36.08	0.9852	42.86	0.9852
DDFW	35.44	0.9858	43.57	0.9896	41.29	0.9831	38.85	0.9905	40.95	0.9900	43.23	0.9906	37.15	0.9886	43.45	0.9868
ESF	35.34	0.9858	42.55	0.9885	40.45	0.9814	37.55	0.9878	41.22	0.9904	42.15	0.9893	37.19	0.9879	42.96	0.9859
MSG	35.72	0.9872	43.63	0.9898	41.42	0.9838	38.94	0.9906	41.15	0.9903	43.39	0.9908	37.39	0.9891	43.71	0.9871
ARI	35.36	0.9843	42.67	0.9869	40.61	0.9798	38.26	0.9887	40.51	0.9878	42.67	0.9881	34.97	0.9800	41.89	0.9716
FDRI	33.00	0.9760	41.09	0.9810	39.74	0.9731	36.24	0.9842	36.24	0.9757	41.92	0.9869	33.24	0.9715	40.43	0.9662
DDRL	37.03	0.9841	45.02	0.9853	42.85	0.9820	40.57	0.9886	42.17	0.9864	44.89	0.9867	38.60	0.9853	44.03	0.9807
DDemo	36.59	0.9889	44.82	0.9899	42.84	0.9864	40.55	0.9931	42.46	0.9914	44.89	0.9916	38.10	0.9902	44.08	0.9856
DRDN	37.87	0.9905	45.26	0.9912	43.81	0.9890	41.24	0.9938	42.88	0.9921	42.03	0.9903	39.12	0.9911	44.40	0.9850
DRDN+	38.10	0.9908	45.44	0.9915	43.89	0.9892	41.44	0.9940	43.09	0.9924	44.19	0.9921	39.34	0.9914	44.66	0.9866

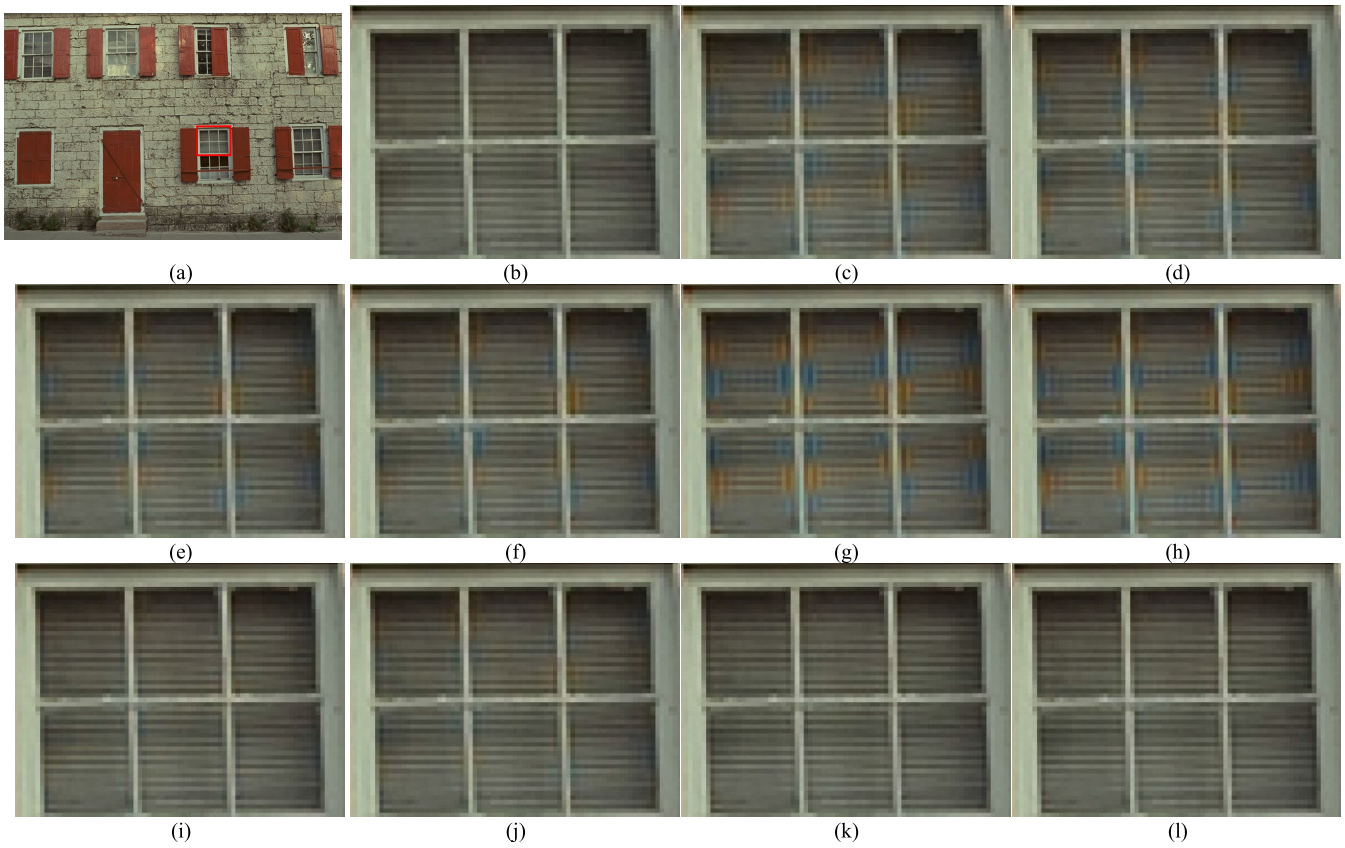


FIGURE 6. Result images of the conventional demosaicking methods and the proposed method for the Kodak dataset: (a) Original image, (b) red block of the original image, (c) DLMMSE, (d) DDFW, (e) ESF, (f) MSG, (g) ARI, (h) FDRI, (i) DDRL, (j) DDemo, (k) DRDN, and (l) DRDN+.

The McMaster dataset consists of 18 images where each image is extracted from high resolution images with a size of 500×500.

B. PERFORMANCE COMPARISON

We compare the performance of our proposed network with six non-CNN-based algorithms (DLMMSE [10], DDFW [11], ESF [12], MSG [13], ARI [14], and FDRI [15]) and two state-of-the-art CNN-based demosaicking models (DDRL [31] and DDemo [32]). For the objective comparison, we calculate the color peak signal to noise ratio (CPSNR) [39] and structural similarity (SSIM) [40] of the result images. When calculating the CPSNR, we remove 10 pixels around the borders of the resulting images and ground truth images to avoid boundary artifacts. Table 2 and 3 show the objective comparison results of the Kodak dataset, and Table 4 and 5 show the objective comparison results of the McMaster dataset. Note that DRDN+ denotes the result of the proposed network which employed the self-ensemble method [19], [34]. In the tables, we highlighted the highest value with red, and the second highest value with blue. As expected, the CNN-based demosaicking models show better performances for most of the sequences than the non-CNN based algorithms when comparing CPSNR and SSIM.

For the Kodak dataset in Table 2, the proposed methods with and without the self-ensemble method show the highest and second highest results for most of the sequences. However, it is interesting to note that the non-CNN-based conventional methods show good performance on the SSIM for some sequences. This implies that there is room for further development of demosaicking methods that employ deep neural networks. From the average performances comparison in Table 3, it can be seen that the proposed method outperforms the conventional methods by up to 4.66 dB in terms of CPSNR results, and by up to 0.0156 in terms of SSIM results. The proposed method achieves even better performance when it applies the self-ensemble method. The proposed method shows improved CPSNR and SSIM results up to 4.89 dB and 0.0160, respectively. For the McMaster dataset in Table 4, the proposed method and the proposed method with the self-ensemble method achieves the highest and the second highest performance for the most of the sequences. The DDRL, which applies deep neural networks for demosaicking shows good performance on the CPSNR for some sequences; however, overall, the proposed method exhibits a superior performance than the conventional methods. The proposed method outperforms the conventional methods by up to 5.05 dB in terms of average CPSNR, and by up to 0.0464 in terms of average SSIM as shown in Table 5.

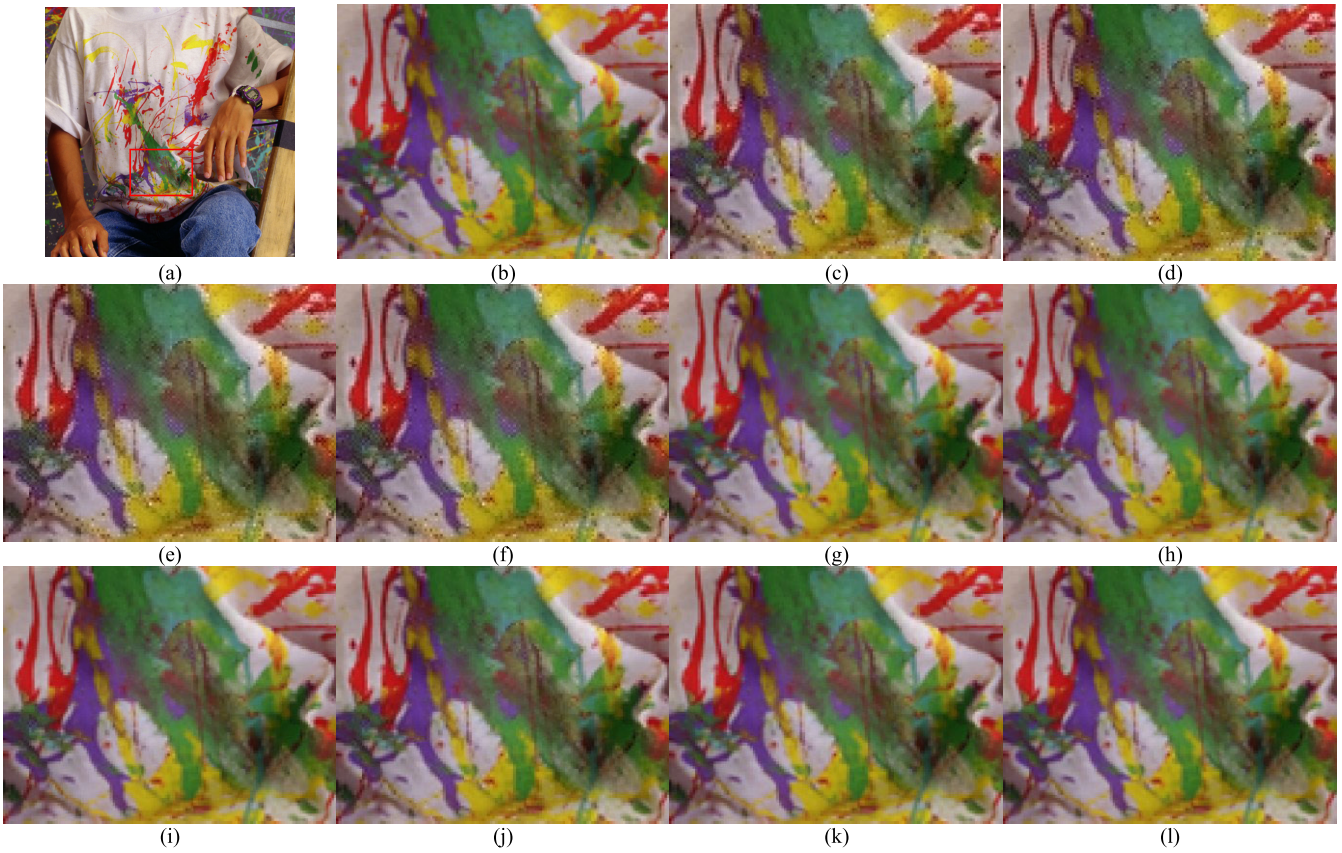


FIGURE 7. Result images of the conventional demosaicking methods and the proposed method for the McMaster dataset: (a) Original image, (b) red block of the original image, (c) DLMMSE, (d) DDFW, (e) ESF, (f) MSG, (g) ARI, (h) FDRI, (i) DDRL, (j) DDemo, (k) DRDN, and (l) DRDN+.

TABLE 3. Average CPSNR (dB) and SSIM results of the conventional demosaicking methods and the proposed method for the Kodak dataset.

	DLMMSE	DDFW	ESF	MSG	ARI
CPSNR	40.11	40.66	40.31	40.92	39.77
SSIM	0.9815	0.9853	0.9852	0.9866	0.9811
	FDRI	DDRL	DDemo	DRDN	DRDN+
CPSNR	37.77	41.97	42.03	42.43	42.66
SSIM	0.9733	0.9831	0.9879	0.9889	0.9893

When the self-ensemble method is employed, the proposed method exhibits improved average CPSNR and SSIM up to 5.19 dB and 0.0471, respectively. These results show that our proposed method outperforms the conventional methods significantly, and exhibits even more enhanced performance by applying the self-ensemble method.

Figure 6 and 7 present the result images of the Kodak and McMaster datasets, respectively, for the subjective comparison. When comparing the performances of the demosaicking methods, it is important to compare whether there exist any artifacts such as zippering or false color artifacts. In Figure 6, we compare the result images of the Kodak dataset. As shown in the figure, the conventional demosaicking methods show false color artifacts, both with and without the CNN architecture. However, the proposed method interpolates the pixel values accurately and does not present any artifacts. Figure 7 presents the resulting images of the McMaster dataset. As shown in the figure, the conventional methods produce false color artifacts in the form of black dots. However, the proposed method does not exhibit any false color artifacts, which establishes its excellent performance.

C. ABLATION STUDY

To exhibit the effect of global residual learning (GR), local residual learning (LR), and dense connectivity (CONC), we performed an ablation study. Table 6 shows the results of the ablation study. When we removed all the components from our proposed network, it was unable to train the network. This exhibits that the vanishing-gradient problem hindered our network from training the parameters without the components. When we applied one of the three components, our network could overcome the vanishing-gradient

TABLE 4. CPSNR (dB) and SSIM results of the conventional demosaicking methods and the proposed method for the McMaster dataset.

	1		2		3		4		5		6	
	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM
DLMMSE	26.98	0.8544	36.34	0.9355	37.25	0.9254	36.60	0.9272	38.78	0.9213	37.23	0.9247
DDFW	27.14	0.8613	36.93	0.9443	37.50	0.9342	36.83	0.9377	38.64	0.9300	37.14	0.9314
ESF	26.07	0.8531	35.57	0.9458	36.44	0.9328	35.89	0.9471	38.28	0.9414	36.66	0.9386
MSG	27.05	0.8695	36.47	0.9509	37.28	0.9396	36.80	0.9499	38.83	0.9420	37.13	0.9409
ARI	29.63	0.9245	39.24	0.9719	40.20	0.9726	40.02	0.9642	40.59	0.9500	39.02	0.9568
FDRI	29.38	0.9188	38.99	0.9715	39.98	0.9723	39.92	0.9671	40.97	0.9561	39.29	0.9585
DDRL	31.05	0.9292	40.90	0.9732	41.61	0.9728	41.40	0.9605	42.23	0.9515	40.17	0.9588
DDemo	29.26	0.9133	39.66	0.9726	40.86	0.9728	39.38	0.9663	41.50	0.9590	39.09	0.9581
DRDN	31.13	0.9371	40.69	0.9761	41.69	0.9762	41.07	0.9669	41.78	0.9566	40.11	0.9616
DRDN+	31.26	0.9387	40.83	0.9765	41.77	0.9766	41.19	0.9674	41.88	0.9575	40.19	0.9623
	7		8		9		10		11		12	
	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM
DLMMSE	37.27	0.9176	30.46	0.8736	29.31	0.8642	33.92	0.9282	33.68	0.9082	32.59	0.9550
DDFW	37.32	0.9225	30.05	0.8813	29.79	0.8730	33.81	0.9324	33.54	0.9093	32.63	0.9560
ESF	36.70	0.9306	28.98	0.8700	28.35	0.8556	33.49	0.9362	33.07	0.9171	32.31	0.9549
MSG	37.19	0.9329	30.18	0.8866	29.30	0.8730	34.10	0.9409	33.67	0.9206	32.93	0.9584
ARI	39.38	0.9590	35.64	0.9670	34.69	0.9620	36.43	0.9657	35.21	0.9454	34.63	0.9725
FDRI	39.42	0.9599	35.41	0.9650	33.75	0.9553	36.19	0.9638	35.14	0.9456	33.53	0.9687
DDRL	40.37	0.9561	36.50	0.9635	36.24	0.9669	37.93	0.9656	36.11	0.9491	36.23	0.9715
DDemo	39.47	0.9571	34.99	0.9527	34.30	0.9548	35.70	0.9542	34.94	0.9429	34.94	0.9730
DRDN	40.38	0.9627	37.01	0.9710	36.08	0.9688	38.03	0.9706	36.00	0.9500	36.30	0.9784
DRDN+	40.46	0.9631	37.11	0.9718	36.31	0.9704	38.19	0.9715	36.15	0.9514	36.45	0.9790
	13		14		15		16		17		18	
	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM
DLMMSE	34.32	0.9825	31.27	0.9091	33.84	0.9122	38.64	0.9737	37.45	0.9660	34.41	0.9270
DDFW	34.72	0.9832	31.11	0.9117	33.75	0.9186	38.98	0.9750	37.33	0.9670	34.78	0.9334
ESF	34.66	0.9837	30.30	0.9097	32.10	0.9039	38.80	0.9751	37.31	0.9667	33.95	0.9349
MSG	35.49	0.9852	31.12	0.9168	33.56	0.9195	39.17	0.9761	37.61	0.9669	34.69	0.9388
ARI	37.89	0.9892	35.42	0.9628	39.75	0.9724	39.70	0.9791	39.51	0.9784	37.88	0.9631
FDRI	38.03	0.9894	34.82	0.9591	38.88	0.9693	35.83	0.9603	38.14	0.9759	37.52	0.9639
DDRL	40.41	0.9775	36.66	0.9600	40.67	0.9678	41.34	0.9790	41.00	0.9774	40.17	0.9660
DDemo	38.69	0.9901	35.09	0.9561	39.40	0.9689	40.81	0.9821	40.39	0.9797	38.64	0.9657
DRDN	40.09	0.9917	36.44	0.9656	40.77	0.9753	41.44	0.9836	40.91	0.9795	39.95	0.9689
DRDN+	40.35	0.9919	36.54	0.9663	40.94	0.9759	41.52	0.9838	41.21	0.9803	40.08	0.9697

TABLE 5. Average CPSNR (dB) and SSIM results of the conventional demosaicking methods and the proposed method for the McMaster dataset.

	DLMMSE	DDFW	ESF	MSG	ARI
CPSNR	34.46	34.56	33.83	34.59	37.49
SSIM	0.9225	0.9279	0.9276	0.9338	0.9643
	FDRI	DDRL	DDemo	DRDN	DRDN+
CPSNR	36.96	38.94	37.62	38.88	39.02
SSIM	0.9623	0.9637	0.9622	0.9689	0.9697

TABLE 6. Ablation study results of the GR, LR, and CONC (CPSNR (dB) / SSIM).

GR	X	O	X	X	O	O	X	O
LR	X	X	O	X	O	X	O	O
CONC	X	X	X	O	X	O	O	O
Kodak	15.31 / 0.3045	27.03 / 0.8223	42.12 / 0.9882	42.39 / 0.9885	38.65 / 0.9764	42.03 / 0.988	42.42 / 0.9888	42.43 / 0.9889
McMaster	12.87 / 0.2114	27.58 / 0.8524	38.63 / 0.9676	38.75 / 0.9682	37.39 / 0.9615	38.57 / 0.9675	38.80 / 0.9686	38.88 / 0.9689

problem except for the case when we only applied GR. This indicates that GR could reduce the vanishing-gradient problem to some extent. The most successful component for solving the vanishing-gradient problem was the CONC followed by LR. They showed powerful performance completely solving the vanishing-gradient problem. When we applied two of the three components, our proposed network could be trained in every condition, and the CONC was found to be the most effective for the performance of our network. When we applied all the components, our proposed network could achieve the most powerful performance.

D. DIFFERENCE TO RDN

Although the network architecture of our proposed method is similar to that of RDN [20], there are three major differences: First, the architecture of the blocks is different. Unlike RDN, that uses six 3×3 convolution layers in its blocks, our DRDN uses three sets of 1×1 and 3×3 convolution layers; our experimental analysis indicates that using 1×1 convolution layer achieves better demosaicking performance. Second, RDN concatenates the output information of every block at the end of the network, thereby consuming a lot of memory, producing only a slight increase in the demosaicking performance. On the contrary, our DRDN does not use the output information of every block at the end of the network. Finally, we applied a different number of the blocks and growth rates, which are optimized for demosaicking and memory consumption.

V. CONCLUSION

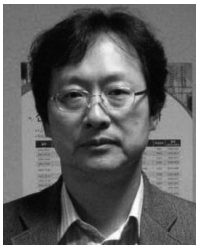
In this paper, we proposed a densely connected residual demosaicking network which successfully applied dense connectivity and residual learning on the demosaicking solution. Conventional demosaicking methods applied initial interpolation or demosaicking process, which made their method computationally complex. To solve this problem, our proposed network was trained in an end-to-end manner without any pre-processes. Moreover, our proposed network generated four color layers with a quarter of the size of the original mosaicked image by input data modification. This enabled our network to reduce the computational complexity and memory consumptions. Next, we trained our proposed network, which applied residual learning on densely connected CNN to avoid the vanishing-gradient problem. Additionally, our proposed network applied the sub-pixel interpolation layer which learns to generate demosaicked images with desired resolution while our proposed CNN model is trained. This enabled our network to generate the demosaicked images more efficiently and accurately. Finally, we applied the self-ensemble method, which enabled our proposed network to achieve even better performances without additional training or clustering process. The experimental results exhibited that our proposed network outperforms the conventional demosaicking methods for both objective and subjective comparisons.

REFERENCES

[1] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065 A, Jul. 20, 1976.
[2] T. Smith and J. Guild, "The C.I.E. colorimetric standards and their use," *Trans. Opt. Soc.*, vol. 33, no. 3, pp. 73–134, 1931.
[3] J. Li, C. Bai, Z. Lin, and J. Yu, "Optimized color filter arrays for sparse representation-based demosaicking," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2381–2393, May 2017.
[4] Y. Niu, J. Ouyang, W. Zuo, and F. Wang, "Low cost edge sensing for high quality demosaicking," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2415–2427, May 2019.
[5] R. Azizi and A. Latif, "Cross-channel regularisation for joint demosaicking and intrinsic lens deblurring," *IET Image Process.*, vol. 12, no. 9, pp. 1683–1691, Sep. 2018.
[6] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.
[7] K. T. Gribbon and D. G. Bailey, "A novel approach to real-time bilinear interpolation," in *Proc. 2nd IEEE Int. Workshop Electron. Design, Test Appl. (DELTA)*, Jan. 2004, pp. 126–131.
[8] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.
[9] R. A. Maschal, Jr., S. S. Young, J. Reynolds, K. Krapels, J. Fanning, and T. Corbin, "Review of Bayer pattern CFA demosaicing with new quality assessment algorithms," *Proc. SPIE*, vol. 7662, p. 766215, Apr. 2010.
[10] D. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2167–2178, Dec. 2005.
[11] Z. Dengwen, S. Xiaoliu, and D. Weiming, "Colour demosaicking with directional filtering and weighting," *IET Image Process.*, vol. 6, no. 8, pp. 1084–1092, Nov. 2012.
[12] I. Pekkucuksen and Y. Altunbasak, "Edge strength filter based color filter array interpolation," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 393–397, Jan. 2012.
[13] I. Pekkucuksen and Y. Altunbasak, "Multiscale gradients-based color filter array interpolation," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 157–165, Jan. 2013.
[14] Y. Monno, D. Kiku, M. Tanaka, and M. Okutomi, "Adaptive residual interpolation for color image demosaicking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3861–3865.
[15] Y. Kim and J. Jeong, "Four-direction residual interpolation for demosaicking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 5, pp. 881–890, May 2016.
[16] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[18] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.
[19] B. Lim, S. Son, H. Kim, S. Nah, and K. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, Jul. 2017, pp. 1132–1140.
[20] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2472–2481.
[21] G. Eilertsen, J. Kronander, G. Denes, R. Mantiuk, and J. Urger, "HDR image reconstruction from a single exposure using deep CNNs," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–15, 2017.
[22] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8183–8192.
[23] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[24] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3929–3938.
[25] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.
[26] H. Zhang and V. M. Patel, "Densely connected pyramid dehazing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3194–3203.
[27] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 695–704.
[28] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Trans. Graph.*, vol. 35, no. 6, 2016, Art. no. 191.
[29] N. S. Syu, Y. S. Chen, and Y. Y. Chuang, "Learning deep convolutional network for demosaicing," 2018, *arXiv:1802.03769*. [Online]. Available: https://arxiv.org/abs/1802.03769
[30] F. Kokkinos and S. Lefkimmiatis, "Deep image demosaicking using a cascade of convolutional residual denoising networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 303–319.

[31] R. Tan, K. Zhang, W. Zuo, and L. Zhang, "Color image demosaicking via deep residual learning," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 793–798.
[32] D. S. Tan, W.-Y. Chen, and K.-L. Hua, "DeepDemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2408–2419, May 2018.
[33] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.
[34] R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1865–1873.
[35] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.
[36] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[37] R. Franzen. (1999). *Kodak Lossless True Color Image Suite*. [Online]. Available: http://r0k.us/graphics/kodak
[38] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *J. Electron. Imag.*, vol. 20, no. 2, p. 023016, 2011.
[39] A. Horé and D. Ziou, "Is there a relationship between peak-signal-to-noise ratio and structural similarity index measure?" *IET Image Process.*, vol. 7, no. 1, pp. 12–24, Feb. 2013.
[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

BUMJUN PARK is currently pursuing the combined master's and Ph.D. degrees with the Department of Electronics and Computer Engineering, Hanyang University, Seoul, South Korea. His current research interests include frame rate up-conversion, denoising, and demosaicking based on deep learning.



JECHANG JEONG received the B.S. degree in electronic engineering from Seoul National University, South Korea, in 1980, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, in 1982, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1990. From 1982 to 1986, he was with the Korean Broadcasting System, where he helped develop teletext systems. From 1990 to 1991, he was a Postdoctoral Research Associate with the University of Michigan, Ann Arbor, where he helped to develop various signal-processing algorithms. From 1991 to 1995, he was with Samsung Electronics Company, South Korea, where he was involved in the development of HDTV, digital broadcasting receivers, and other multimedia systems. Since 1995, he has been conducting research at Hanyang University, Seoul, South Korea. He has published numerous technical articles. His research interests include digital signal processing, digital communication, and image and audio compression for HDTV and multimedia applications. He was a recipient of the Scientist of the Month Award, in 1998 from the Ministry of Information and Communication of Korea.