Demosaicking Algorithm Using Deep Residual Convolutional Network

Jin Wang[1], Siyou Guo[2], Qilei Li[3], David Camacho[4], Gwanggil Jeon[1]*

[1] The Department of Embedded Systems Engineering, Incheon National University, Incheon, 22012 (South Korea)
[2] The School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo 255000 (China)
[3] The School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS (United Kingdom)
[4] The Department of Computer Systems Engineering, Technical University of Madrid, Madrid (Spain)

* Corresponding author: gjeon@inu.ac.kr

Received 13 February 2025 | Accepted 4 May 2025 | Early Access 24 February 2026

Abstract

Single-sensor imaging systems are widely deployed in portable devices including digital cameras, smartphones, and personal digital assistants (PDAs) for real-time image acquisition. While convolutional neural networks (CNNs) have demonstrated exceptional capabilities in various image processing tasks, their potential for demosaicking applications remains underexplored. This paper presents a demosaicking framework utilizing a Deep Residual Convolutional Neural Network (DRCNN) architecture. Firstly, we initialize the mosaicked images using conventional demosaicking algorithms and learn the DRCNN for three color channels. The proposed DRCNN architecture innovatively integrates three core components: Binary Convolution Units (BCUs) for computational efficiency, Efficient Layer Aggregation Networks (ELAN) for multi-scale feature fusion, and Dense Residual Blocks (DRBs) for enhanced gradient flow. Comprehensive evaluations demonstrate that the proposed algorithms outperform existing approaches in PSNR, computational complexity, and visual quality.

Keywords

Color Channel, Convolutional Neural Network, Demosaicking, Residual Learning.

DOI: 10.9781/ijimai.2026.6568

I. Introduction

In most digital cameras, images are captured by a single image sensor, its surface overlaid with a color filter array (CFA). This design choice is primarily driven by the need to reduce both cost and size, as the image sensor represents the most expensive component of the camera system [1]. To correctly restore missing color channels, demosaicking methods [2] using color differences between green (G) channels and red/blue (R/B) channels have been studied. Certain approaches first perform edge detection, subsequently interpolating missing color values along the detected edges rather than across them. One such technique, adaptive color plane interpolation (ACPI), employs an edge-sensing interpolation method [3]. ACPI generates three distinct predictors and selects the optimal one via an edge classifier, which utilizes the Laplacian second-order term for the R or B planes and the gradient term for the G plane. Another noteworthy method, effective color interpolation (ECI), relies on color differences between the G and R/B channels to estimate the missing values [4]. Building on this concept, enhanced ECI (EECI) refines the interpolation by incorporating pixel differences as weights [5]. Meanwhile, the adaptive homogeneity-directed (AHD) method capitalizes on color homogeneity within the CIELAB color space [6], and an adaptive filtering approach has also been applied to CFA demosaicking [7]. Additionally, a demosaicking technique based on the variance of color differences (VCD) has been introduced to identify reliable edges [8].

In this approach, the missing G values are categorized into two distinct regions: the plane region and the edge region. To determine whether a pixel lies on an edge, the gradient variation within an $N \times N$ window is computed. The directional filtering and a posteriori decision method (DFPD) utilizes the sum of the gradients in both the horizontal and vertical directions within a 5×5 window [9]. By combining DFPD with enhanced effective color interpolation (EECI), the color demosaicking with directional filtering and weighting (CDDFW) method was introduced [10].

The high-order interpolation (HOI) method, which employs a Taylor series expansion for interpolation, alongside a weighted median filter to select the optimal predictor, has also been proposed [11]. The effective demosaicking based on subband correlation (EDUSC) method utilizes discrete wavelet transforms to classify edge pixels [12]. In the effective demosaicking method based on edge properties (EDAEP), fixed weights are added to the ECI method to enhance interpolation quality [13]. Further advancements are found in the work of Pekkucuksen et al., who introduced demosaicking methods that leverage edge strength filters and multiscale gradients [14], [15]. Kim et al. proposed a novel method incorporating geometric

Please cite this article as:
J. Wang, S. Guo, Q. Li, D. Camacho, G. Jeon. Demosaicking Algorithm Using Deep Residual Convolutional Network, *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 9, no. 7, pp. 6-15, 2026, http://doi.org/10.9781/ijimai.2026.6568

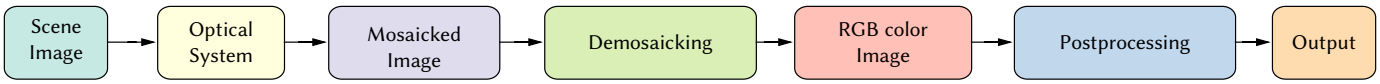


Fig. 1. Digital image acquisition using a digital camera.

duality and dilated directional differentiation for more accurate color reconstruction [16]. Several techniques also explore the frequency domain of CFA images. The regularization approach to demosaicking (RAD) employs frequency properties to evaluate the luminance component, offering spatially adaptive regularization [17]. Dubois *et al.* introduced a luma-chroma demultiplexing algorithm [18] and further refined it by applying a least-squares strategy to design an optimal filter [19], [20].

Recently, considerable attention has been devoted to convolutional neural networks (CNNs) within the fields of computer vision and image processing. CNN-based approaches have led to significant breakthroughs in various tasks, including image recognition, segmentation, super-resolution, and denoising [21]–[34]. The very deep convolutional network (VDSR) [33], which introduced the concept of residual learning, marked a major advancement in image super-resolution, yielding substantial performance improvements. Inspired by VDSR, we propose a deep residual CNN-based demosaicking method. In recent years, numerous demosaicking algorithms leveraging CNNs have emerged [35]–[41]. Wang *et al.* introduced a compact demosaicking neural network based on the UNet++ architecture [35], while joint denoising and demosaicking algorithms were explored by Gharbi et al. [36]. Several approaches have focused on color image demosaicking through deep residual learning (DDRL) [40], and a deep fully convolutional network for demosaicking was proposed by Tan *et al.* [41].

While deep-learning-based demosaicking has demonstrated significant improvements over traditional methods, existing approaches often suffer from large model sizes and inefficient architectures, making them less practical for real-time applications. Additionally, handling complex edges and fine textures remains a challenge due to insufficient feature aggregation and redundant network parameters.

To address these issues, we introduce a novel combination of binary convolution units (BCUs) [42], efficient layer aggregation networks (ELAN) [43], and dense residual blocks (DRBs). BCUs improve computational efficiency by approximating full-precision weights through linear combinations of binary weights, reducing memory usage while maintaining performance. ELAN enhances gradient propagation, allowing deeper networks to learn high-quality demosaicked images with improved feature fusion. Meanwhile, DRBs mitigate weight redundancy and strengthen feature reuse, which is particularly beneficial for preserving fine details in challenging demosaicking scenarios. By integrating these components, our method achieves high-quality color reconstruction with improved efficiency, making it well-suited for practical applications.

The rest of the paper is organized as follow: a brief review of related work is introduced in section II. In Section III, we present a detailed description of the proposed algorithm. Section IV presents experimental results and a discussion of the findings. Finally, Section V concludes the paper.

II. Related Work

Digital camera acquires digital images by a single monochromatic image sensor which processes only one-color channel at each pixel location shown in Fig. 1. Therefore, the acquired image is in the pattern of a mosaic. Fig. 2 shows 7 × 7 Bayer pattern CFA. The number of G color pixels is twice that of blue and red pixels, and the G channel retains more image details in the Bayer CFA pattern. To restore a three-channel full-color image from the mosaicked one, an interpolation method that reconstructs the missing color values is necessary. This algorithm is called demosaicking or CFA interpolation [2].

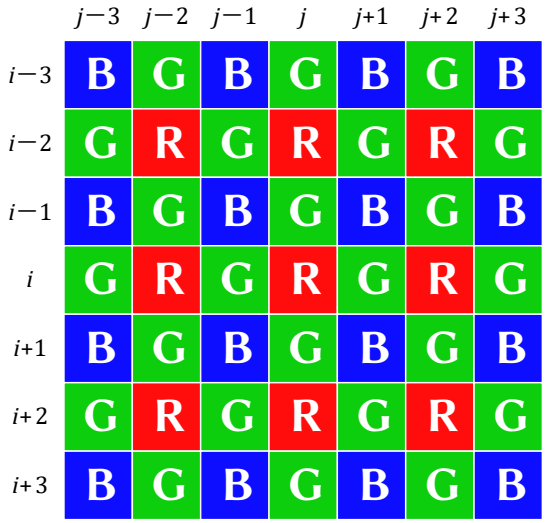


Fig. 2. Illustration of the 7 × 7 CFA Bayer pattern.

The conventional demosaicking algorithms are divided into three categories: one is weighted-average based methods using a directional filter; the other is dictionary learning methods; another is neural network based methods. Most weighted-average based methods reconstruct the G channel component and then estimate the red and blue channels using the reconstructed G color channel. The to-be-interpolated green color pixels are a weighted average of the given color pixels around the missing green pixels. Gharbi *et al.* [36] and Tan *et al.* [41] proposed deep convolutional neural network based demosaicking algorithms. Tan *et al.* [41] adopts a fully convolutional network which needs huge parameters to train the datasets. They proposed three-model training and weighted double interpolation fusion to reconstruct the color channels. In contrast, we just adopt two-model training and image-complex selection criteria.

III. The Proposed Deep Residual Convolutional Neural Network-based Demosaicking Algorithm

Gharbi *et al.* [36] directly rearranged the original mosaicked images as the input of the network, and Tan *et al.* [41] provided a two-step image demosaicking method which initialized the mosaicked images using gradient-corrected bilinear interpolation to reconstruct the demosaicked images as the input of the network. We conduct experiments in two ways: (1) using BDD [44] to reconstruct demosaicked images as input, and (2) directly using the original mosaicked images as the input of the network. All the training and testing conditions are the same for the two models, and we randomly choose some images to evaluate the two models. We can see from Table I that in our network, with the same training time, the initial demosaicked images as input achieve better performance than the original mosaicked images as input. Therefore, we chose the initial demosaicked images using BDD [44] as input for our network.

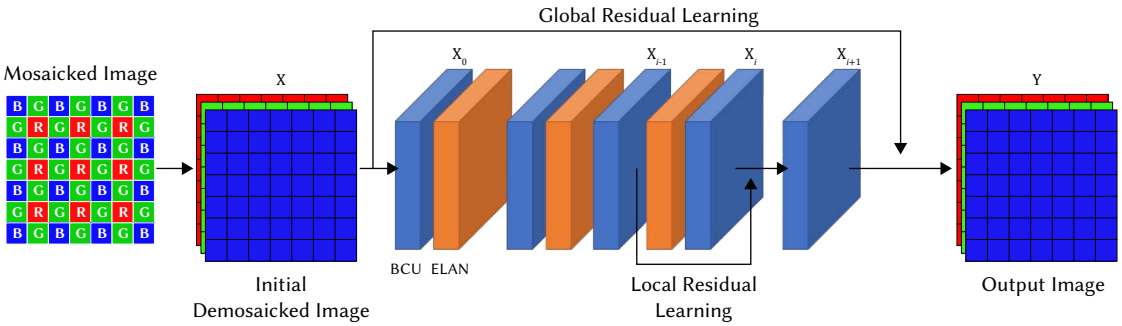


Fig. 3. The proposed DRCNN architecture. We initialized the mosaicked image with BDD [44] as the input of the network. We repeatedly cascade layers (each layer contains one BCU and one ELAN) and two residual learning layers (Local and global residual learning).

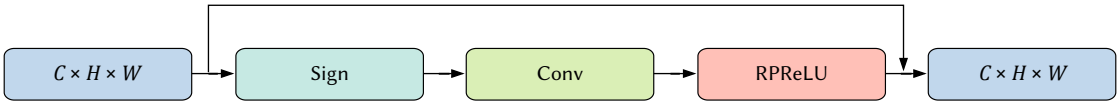


Fig. 4. Structure of BCU.

TABLE I. PSNR AND SSIM COMPARISONS OF TWO MODELS

Image Type	PSNR	SSIM
Initial demosaicked images	41.34	0.9926
Original mosaicked images	40.45	0.9908

A. Network Architecture

Our proposed DRCNN consists of a cascade of nineteen repeated layers, each composed of one BCU and an ELAN layer. The overall structure of our DRCNN architecture is shown in Fig. 3. Each convolutional layer has 64 feature maps with a filter size of 3 × 3. To prevent gradient vanishing and enhance model robustness, we add a nonlinear layer (RPRELU) after each convolutional layer. Residual learning has been shown to enable fast convergence and achieve strong performance in image processing.

BCU: BCU (as shown in Fig. 4) approximates the full-precision weights and activations to only 1 bit with a pre-defined binary function which can achieving high computational efficiency [42]. BCU can be expressed as

$$x_{i,j}^{b} = \mathrm{Sign}(x_{i,j}^{f}) = \begin{cases} +1, & \text{if } x_{i,j}^{f} > \alpha_{i,j} \\ -1, & \text{if } x_{i\,j}^{f} \le \alpha_{i,j} \end{cases} \tag{1}$$

Then the weights are approximated as

$$w_{i,j}^{b} = \frac{\| w_{j}^{f} \|_1}{n} \mathrm{Sign}(w_{i,j}^{f}) = \begin{cases} +\frac{\| w_{j}^{f} \|_1}{n}, & \text{if } w_{i,j}^{f} > 0 \\ -\frac{\| w_{j}^{f} \|_1}{n}, & \text{if } w_{i,j}^{f} \le 0 \end{cases} \tag{2}$$

where $x_{i,j}^{f}$ and $w_{i,j}^{f}$ are full-precision activations and weights in the $i-th$ channel of $j-th$ layer separately. α_{ij} is the learnable coefficient to control the threshold of the sign function for the $i-th$ channel of the $j-th$ layer.

RPReLU: The RPReLU [45] is proposed as the activation function in BCU defined as:

$$f(y_{i,j}) = \begin{cases} y_{i,j} - y_{i,j} + \delta_{i,j}, & \text{if } y_{i,j} > y_{i,j} \\ \beta_{i,j}(y_{i,j} - y_{i,j}) + \delta_{i,j}, & \text{if } y_{i,j} \le y_{i,j} \end{cases} \tag{3}$$

where $f(\cdot)$ is the RPReLU function, $y(i,j)$ is the input of the function and is the element of $i-th$ channel of $j-th$ layer, (i, j) and (i, j) are shifts for moving the distribution, and (i, j) is parameter of the slope of the negative parts.

ELAN: ELAN is an ensemble scaling method to choose a better trade-off of performance and efficiency as shown in Fig. 5. ELAN aims to make a more efficient neural network to converge quickly by guiding the shortest or longest gradient path.

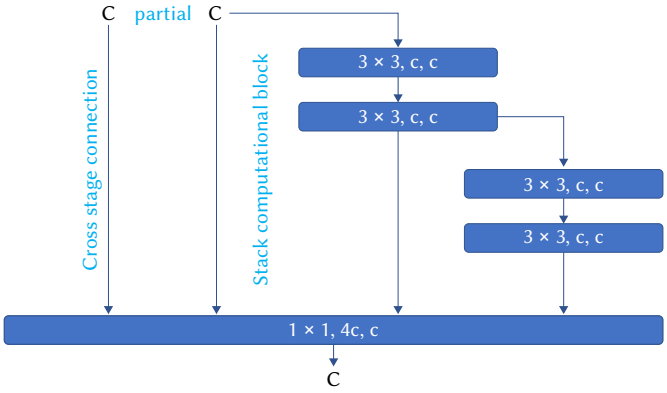


Fig. 5. Structure of ELAN.

Residual unit: Inspired by VDSR [33], at the end of the network, we provide two residual layers, one is local residual learning and the other is global residual learning, as seen in Fig. 3. Let X be the input and X_i represent the output of $i-th$ convolution. We have the expression as

$$X_i = \max(0, w_i * X_{i-1} + b_i) \tag{4}$$

The local residual learning can be expressed as

$$X_{i+1} = w_i * (X_{i-1} + X_i) + b_{i+1} \tag{5}$$

The local residual unit is adding the two feature maps X_{i-1} and X_i to the input of the next convolutional layer. Local residual learning has been proven to further improve the performance of image restoration, which can efficiently increase the ability of the model.

Global residual learning is shown as

$$Y = X_{i+1} + X \tag{6}$$

VDSR [33] introduced the global residual learning to avoid gradient vanishing. After the global residual learning we get the reconstructed demosaicking images.

Image complex selection: The choice of the training set is significant for the network. The different kinds of images have varied features, such as flat images, complex images, colorful images, and so

Fig. 6. 24 images of the Kodak dataset (up) and 18 images of the McM dataset (down).

on. Sorting all the images into a single category is not appropriate. The more similar the test images to the training sets, the better performance the results show. Based on the discussion, we provide an image complex selection criterion to extract the images with more complex regions. Our target is to increase the MSE of the image with low quality. As we know the image detail is difficult to reconstruct and causes image degradation during image reconstruction thereby most methods achieve low PSNR. Therefore, we simply utilize the PNSR between the reconstructed image and the original images to select the complex images. A threshold T is introduced to divide which training images belong to complex image sets.

Ensemble image fusion: With the training of the two image datasets, we simply use a weighted average to fuse the output of the two models as follows:

$$Y_O = wY_G + (1 - w)Y_c \tag{7}$$

In Table II, we provide the comparison of average PSNR and SSIM of Kodak and McM datasets and we can find out that the ensemble model improves both PSNR and SSIM compared with the two single models.

TABLE II. Comparison of Average PSNR and SSIM of Kodak and McM Datasets

Model	PSNR	SSIM
General model	40.921	0.99278
Complex model	40.925	0.99282
Esemble model	40.957	0.99283

IV. Experimental Results

In this section, the performance of the proposed algorithm is evaluated and compared with some reference methods. We assessed the objective performance in terms of color PSNR and structural similarity index (SSIM), which are widely adopted metrics in the literature. The parameters w and the threshold T are super parameters and they are chosen as $T = 35.5$ and $w = 0.4$.

A. Implementation Details

Datasets for training. DIV2K dataset [46] is a recently introduced high-quality image dataset with 2K resolution for image restoration and contains 800 training images. Flickr is another popular high-quality image dataset with 2K resolution. For the general model, we adopt 800 DIV2K images as the training dataset. For the complex region, we test all the DIV2K and Flickr images and choose the images with PSNR lower than $T = 35.5$. We get 997 images for complex models. The training data is firstly preprocessed by BDD [44] and then we cut the training RGB input image with patches size 64×64 with stride 64. Our proposed deep residual convolutional network-based demosaicking algorithm (DRCNN) was implemented using PyTorch and trained on an NVIDIA RTX 3090 GPU with 24GB VRAM. The training process took approximately 72 hours for convergence on a dataset of high-resolution images. For weight initialization, we applied the Xavier initialization to ensure stable gradient propagation in deep layers. Additionally, we experimented with He initialization for convolutional layers but found that Xavier provided more stable training dynamics. To enhance training efficiency and prevent overfitting, we froze the initial convolutional layers during early training epochs, allowing higher layers to focus on learning complex textures and edge details before fine-tuning the entire

TABLE III. Performance Comparison of PSNR and SSIM of Kodak Dataset (The Best Performances Are Marked in **Bold**). DDRCNN Is the Proposed Algorithm

Image	1		2		3		4		5		6	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AFD	37.56	0.9891	40.70	0.9978	42.69	0.9966	41.06	0.9968	38.04	0.9930	38.04	0.9910
VCD	38.42	0.9846	40.43	0.9927	42.54	0.9871	40.77	0.9897	37.78	0.9853	40.00	0.9882
HOI	36.56	0.9871	40.83	0.9980	42.71	0.9965	40.72	0.9966	37.63	0.9925	38.81	0.9926
ESF	39.91	0.9936	40.82	0.9979	42.55	0.9968	40.45	0.9967	37.55	0.9927	41.22	0.9955
EDAEP	34.92	0.9060	39.74	0.9334	41.75	0.9200	40.03	0.9231	35.78	0.9173	36.85	0.9087
ECI	33.96	0.9816	38.49	0.9971	40.76	0.9956	39.49	0.9959	35.31	0.9886	35.05	0.9886
GD	38.98	0.9924	41.07	0.9982	42.41	0.9970	40.96	0.9972	38.11	0.9939	40.76	0.9951
NAT	34.45	0.9772	39.78	0.9861	41.94	0.9928	39.67	0.9898	35.82	0.9864	35.80	0.9838
ARI	38.81	0.9911	39.66	0.9949	42.68	0.9959	40.70	0.9955	38.27	0.9932	40.52	0.9938
FR	39.66	0.9928	41.62	0.9976	44.00	0.9964	41.50	0.9967	39.27	0.9940	41.24	0.9949
BDD	38.75	0.9920	41.50	0.9982	43.22	0.9970	41.30	0.9971	38.56	0.9938	40.59	0.9949
DJDD	41.60	0.9955	41.70	0.9979	43.80	0.9967	42.30	0.9975	39.40	0.9947	41.70	0.9956
DFCD	41.96	0.9958	42.14	0.9976	44.83	0.9968	42.85	0.9976	40.56	0.9956	42.46	0.9960
DRCNN	42.58	0.9967	**42.56**	0.9985	**45.38**	0.9975	**43.63**	0.9979	**41.32**	0.9964	**42.92**	0.9966

Image	7		8		9		10		11		12	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AFD	42.91	0.9966	35.23	0.9873	42.57	0.9892	42.70	0.9916	39.34	0.9914	42.77	0.9955
VCD	42.13	0.9876	36.29	0.9828	42.85	0.9803	42.47	0.9815	39.89	0.9825	43.43	0.9912
HOI	42.72	0.9965	34.86	0.9860	42.39	0.9877	42.27	0.9906	39.13	0.9917	43.03	0.9958
ESF	42.15	0.9962	37.19	0.9906	42.96	0.9900	42.60	0.9919	40.69	0.9936	43.80	0.9966
EDAEP	41.28	0.9211	32.83	0.9002	40.90	0.9052	40.98	0.9103	37.48	0.9077	41.67	0.9144
ECI	40.30	0.9955	30.52	0.9806	39.66	0.9856	40.18	0.9888	36.20	0.9885	40.00	0.9944
GD	42.59	0.9967	36.93	0.9905	42.92	0.9904	42.89	0.9923	40.50	0.9937	43.77	0.9967
NAT	41.50	0.9919	32.61	0.9772	40.53	0.9846	40.84	0.9861	39.65	0.9850	41.02	0.9914
ARI	42.67	0.9954	34.98	0.9847	41.57	0.9785	41.55	0.9888	39.63	0.9904	43.17	0.9956
FR	43.75	0.9962	37.34	0.9907	43.65	0.9898	43.28	0.9916	41.16	0.9934	44.58	0.9965
BDD	43.10	0.9967	36.96	0.9903	43.29	0.9900	42.97	0.9919	40.71	0.9937	44.02	0.9967
DJDD	44.10	0.9968	37.70	0.9917	43.30	0.9891	43.50	0.9917	41.50	0.9938	44.50	0.9967
DFCD	44.90	0.9970	38.11	0.9926	44.09	0.9901	**43.75**	0.9923	42.23	0.9941	45.26	0.9970
DRCNN	**45.23**	**0.9975**	**39.41**	**0.9939**	**44.66**	**0.9913**	42.69	**0.9934**	**42.69**	**0.9950**	**45.90**	**0.9975**

Image	13		14		15		16		17		18	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AFD	33.77	0.9857	37.17	0.9908	39.84	0.9933	41.15	0.9889	41.39	0.9912	37.12	0.9867
VCD	34.80	0.9815	37.01	0.9834	39.70	0.9895	43.71	0.9823	41.52	0.9837	37.12	0.9830
HOI	32.77	0.9820	37.17	0.9903	39.05	0.9925	42.47	0.9917	40.85	0.9900	36.45	0.9845
ESF	36.11	0.9904	36.11	0.9916	39.27	0.9932	44.77	0.9945	41.95	0.9916	37.72	0.9859
EDAEP	31.48	0.9059	36.27	0.9154	38.81	0.9185	40.44	0.9055	39.63	0.9176	35.38	0.9162
ECI	31.34	0.9767	35.28	0.9867	38.73	0.9911	38.07	0.9874	39.31	0.9875	35.19	0.9818
GD	35.07	0.9888	36.78	0.9926	39.61	0.9936	44.18	0.9938	42.07	0.9919	37.79	0.9878
NAT	30.74	0.9706	36.32	0.9844	38.49	0.9886	39.04	0.9814	39.06	0.9864	34.83	0.9669
ARI	35.21	0.9877	37.55	0.9918	38.61	0.9893	43.25	0.9930	41.14	0.9900	36.87	0.9833
FR	35.75	0.9896	38.27	0.9926	40.28	0.9938	44.60	0.9936	42.29	0.9917	38.15	0.9865
BDD	34.95	0.9886	37.70	0.9926	39.61	0.9935	44.42	0.9941	42.02	0.9918	37.64	0.9873
DJDD	37.50	0.9929	39.20	0.9936	41.50	0.9940	44.70	0.9940	42.00	0.9909	39.30	0.9890
DFCD	37.67	0.9935	40.27	0.9944	42.09	0.9945	45.44	0.9946	42.76	0.9917	39.05	0.9895
DRCNN	**37 .89**	**0.9936**	**40.48**	**0.9954**	**42.67**	**0.9956**	**45.92**	**0.9954**	**43.13**	**0.9929**	**39.63**	**0.9902**

Image	19		20		21		22		23		24	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AFD	40.00	0.9911	41.12	0.9894	38.68	0.9911	38.51	0.9903	43.15	0.9961	34.83	0.9872
VCD	40.51	0.9830	41.13	0.9810	39.16	0.9852	38.07	0.9790	43.00	0.9876	35.10	0.9792
HOI	39.82	0.9902	40.63	0.9889	37.92	0.9904	38.03	0.9889	43.48	0.9959	34.16	0.9854
ESF	41.49	0.9913	41.52	0.9897	40.30	0.9929	38.41	0.9891	42.45	0.9951	35.34	0.9873
EDAEP	37.75	0.9119	39.26	0.9097	36.45	0.9133	37.17	0.9140	42.36	0.9240	33.73	0.9095
ECI	35.30	0.9877	38.62	0.9873	35.82	0.9874	36.37	0.9876	40.67	0.9955	33.18	0.9828
GD	41.38	0.9924	41.49	0.9902	39.37	0.9924	38.69	0.9907	42.87	0.9962	35.19	0.9881
NAT	37.51	0.9738	39.41	0.9788	35.92	0.9721	37.28	0.9767	42.28	0.9773	33.35	0.9735
ARI	40.46	0.9902	40.86	0.9867	39.24	0.9911	38.06	0.9878	43.37	0.9960	35.35	0.9872
FR	41.85	0.9884	42.32	0.9887	40.31	0.9893	39.48	0.9885	44.08	0.9943	35.65	0.9868
BDD	41.07	0.9917	41.69	0.9901	39.61	0.9926	38.82	0.9906	43.46	0.9961	34.92	0.9878
DJDD	42.20	0.9935	42.30	0.9904	41.00	0.9937	39.30	0.9918	44.10	0.9963	36.60	0.9900
DFCD	42.56	0.9933	42.97	0.9908	41.51	0.9940	39.96	0.9920	44.97	0.9965	36.58	0.9905
DRCNN	**43.15**	**0.9938**	**43.53**	**0.9921**	**41.99**	**0.9947**	**40.73**	**0.9930**	**45.52**	**0.9973**	**37.92**	**0.9922**

TABLE IV. Performance Comparison of Average PSNR and SSIM of Kodak Dataset (The Best Performances Are Marked in **Bold**). DDRCNN Is the Proposed Algorithm

Method	VCD	HOI	ESF	ECI	EDAEP	GD	BDD
PSNR	39.596	39.909	39.352	40.306	38.039	39.991	40.45
SSIM	0.9847	0.9909	0.9927	0.9137	0.9883	0.9930	0.9929
Method	NAT	ARI	CHID	FR	DIDD	DFCN	DRCNN
PSNR	37.714	39.75	40.82	41.00	41.145	42.04	42.64
SSIM	0.9818	0.9905	0.9923	0.9937	0.9944	0.9949	0.9949

TABLE V. Performance Comparison of PNSR and SSIM of McM Dataset (The Best Performances Are Marked in **Bold**). DDRCNN Is the Proposed Algorithm

Image	1		2		3		4		5		6	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AFD	27.35	0.9642	33.89	0.9724	33.08	0.9763	36.04	0.9876	31.73	0.9701	34.24	0.9655
VCD	26.90	0.9577	33.55	0.9677	32.46	0.9550	34.49	0.9607	31.25	0.9575	33.78	0.9523
HOI	27.61	0.9664	34.19	0.9733	32.89	0.9749	36.07	0.9876	31.91	0.9703	34.36	0.9647
ESF	26.07	0.9534	33.08	0.9677	32.31	0.9724	34.66	0.9852	30.30	0.9611	32.10	0.9481
EDAEP	27.61	0.8849	34.00	0.8951	32.07	0.8801	34.37	0.8822	32.11	0.8850	35.04	0.8823
ECI	26.96	0.9661	33.32	0.9731	31.73	0.9723	33.35	0.9845	31.74	0.9718	34.38	0.9699
GD	27.16	0.9633	33.88	0.9721	33.04	0.9756	35.83	0.9873	31.50	0.9687	33.80	0.9614
NAT	29.18	0.9466	38.78	0.9688	39.57	0.9667	38.88	0.9652	40.74	0.9643	38.70	0.9599
ARI	29.63	0.9763	39.25	0.9942	40.19	0.9936	40.03	0.9975	40.60	0.9969	39.02	0.9895
FR	29.80	0.9749	39.62	0.9919	40.41	0.9913	40.14	0.9939	41.23	0.9929	39.37	0.9865
BDD	28.55	0.9719	34.79	0.9761	33.00	0.9760	36.88	0.9887	33.47	0.9771	36.54	0.9759
DJDD	28.70	0.9741	34.80	0.9767	34.20	0.9793	37.00	0.9883	33.90	0.9800	38.60	0.9823
DFCD	29.26	0.9757	**39.66**	**0.9949**	**40.85**	**0.9944**	39.39	**0.9976**	**41.50**	**0.9975**	39.30	**0.9904**
DRCNN	**31.52**	**0.9841**	36.38	0.9814	36.73	0.9856	**40.79**	0.9925	36.67	0.9861	**41.14**	0.9884

Image	7		8		9		10		11		12	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AFD	37.89	0.9759	37.93	0.9784	35.42	0.9889	35.42	0.9918	37.63	0.9887	37.08	0.9961
VCD	38.54	0.9689	37.46	0.9711	34.37	0.9812	34.37	0.9867	36.96	0.9837	36.60	0.9916
HOI	37.17	0.9713	37.39	0.9768	35.31	0.9886	35.31	0.9919	37.81	0.9885	37.52	0.9964
ESF	38.80	0.9790	37.31	0.9767	33.96	0.9849	33.96	0.9893	36.44	0.9840	35.89	0.9951
EDAEP	36.24	0.8951	37.14	0.9081	35.25	0.9067	35.25	0.9131	37.85	0.9123	37.18	0.9118
ECI	36.47	0.9668	36.50	0.9760	34.35	0.9888	34.35	0.9919	37.10	0.9891	36.28	0.9962
GD	38.67	0.9790	37.97	0.9783	35.03	0.9881	35.03	0.9912	37.39	0.9878	36.95	0.9961
NAT	38.95	0.9651	33.84	0.9603	32.88	0.9572	35.10	0.9593	35.04	0.9520	32.60	0.9668
ARI	39.38	0.9934	35.63	0.9919	34.69	0.9879	36.42	0.9884	35.22	0.9779	34.64	0.9805
FR	39.65	0.9904	35.33	0.9876	34.21	0.9842	36.28	0.9840	35.61	0.9762	34.48	0.9669
BDD	36.23	0.9659	37.67	0.9774	36.67	0.9910	38.06	0.9930	38.73	0.9905	38.41	0.9968
DJDD	40.50	0.9852	40.00	0.9817	38.20	0.9930	38.60	0.9941	40.00	0.9931	38.50	0.9972
DFCD	39.48	**0.9943**	34.98	**0.9901**	34.30	0.9874	35.68	0.9860	34.95	0.9774	34.95	0.9818
DRCNN	**41.71**	0.9872	**41.48**	0.9827	**40.39**	**0.9951**	**41.13**	**0.9954**	**42.02**	**0.9954**	**41.52**	**0.9981**

Image	13		14		15		16		17		18	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AFD	39.26	0.9960	37.28	0.9874	37.55	0.9911	30.56	0.9790	30.65	0.9721	34.35	0.9820
VCD	38.85	0.9888	37.14	0.9833	37.10	0.9866	30.16	0.9712	29.35	0.9596	33.79	0.9744
HOI	39.40	0.9962	37.81	0.9883	37.81	0.9916	30.93	0.9789	29.80	0.9662	34.24	0.9810
ESF	38.28	0.9953	36.66	0.9858	36.71	0.9890	28.98	0.9689	28.35	0.9528	33.49	0.9766
EDAEP	39.37	0.9090	37.67	0.9066	37.78	0.9101	31.40	0.9039	30.60	0.8952	34.06	0.9005
ECI	38.56	0.9962	36.66	0.9880	37.26	0.9916	30.30	0.9812	29.97	0.9717	32.81	0.9813
GD	39.14	0.9960	37.34	0.9875	37.33	0.9909	30.50	0.9774	29.58	0.9648	34.29	0.9813
NAT	35.94	0.9848	34.09	0.9578	37.83	0.9592	35.96	0.9620	37.50	0.9733	37.00	0.9674
ARI	37.90	0.9899	35.43	0.9827	39.75	0.9857	39.71	0.9823	39.51	0.9814	37.88	0.9929
FR	39.13	0.9708	35.08	0.9765	39.46	0.9803	36.82	0.9633	39.44	0.9772	38.02	0.9907
BDD	40.38	0.9967	38.59	0.9892	38.63	0.9926	32.74	0.9841	31.77	0.9765	35.04	0.9835
DJDD	40.80	0.9971	38.70	0.9897	39.10	0.9937	34.30	0.9885	38.00	0.9837	35.80	0.9848
DFCN	38.70	0.9906	35.11	0.9825	39.40	0.9850	**40.82**	0.9857	**40.40**	0.9832	**38.64**	**0.9939**
DRCNN	**42.21**	**0.9977**	**40.33**	**0.9914**	**40.60**	**0.9950**	37.54	**0.9939**	36.63	**0.9918**	38.44	0.9911

TABLE VI. Performance Comparison of Average PSNR and SSIM of McM Dataset (The Best Performances Are Marked in **Bold**). DDRCNN Is the Proposed Algorithm

Method	VCD	HOI	ESF	ECI	EDAEP	GD	BDD
PSNR	34.936	34.388	34.962	33.830	34.820	34.109	35.90
SSIM	0.9721	0.9807	0.9759	0.8990	0.9809	0.9804	0.9835
Method	NAT	ARI	FR	CHID	DJDD	DFCN	DRCNN
PSNR	36.25	37.49	37.45	36.75	36.93	37.62	**39.28**
SSIM	0.9631	0.9879	0.9822	-	0.9868	0.9882	**0.9907**

network. The model was optimized using the Adam optimizer with an initial learning rate of 1e-4, which was progressively decayed using a cosine annealing schedule. We also incorporated data augmentation techniques such as random cropping, horizontal/vertical flipping, and slight color jittering to improve generalization.

Datasets for testing. In the demosaicking algorithm, the Kodak dataset and McM dataset were used for testing the performance of the proposed algorithm, which are most adopted for evaluating the demosaicking methods, shown in Fig. 6. The Kodak dataset consists of 24 images with a size of 768 × 512, and Zhang *et al.* [47] provided that Kodak dataset has very high spectral correlation and low saturation, and is smoother than general digital color images with high quality. Then McM dataset is introduced by Zhang for demosaicking methods. The McM dataset contains 18 images with size 500 × 500 which are cropped from high resolution images with original size 2310 × 1814. To prove the superiority of our algorithm, we test on both datasets.

Training details. The proposed demosaicking algorithm was trained using an Adam optimizer. The initial learning rate was set to 0.0001 and was halved every 10 epochs. Each model underwent 20 epochs of training, with the learning rate being reduced twice throughout the process. The MSE loss was used to train each model.

B. Comparison With State-of-the-Art Methods

A simulation was performed using standard test images from the Kodak and McM dataset to verify the superiority of the proposed method compared to the existing algorithms (adaptive filter based demosaicking (AFD) [7], VCD [8], HOI [11], Edge Strength Filter (ESF) [14], EDAEP [4], ECI [4], geometric duality (GD) [16], DFCN [41], deep joint demosaicking and denoising (DJDD) [36], fused regression (FR) [48], residual interpolation (ARI) [49], local directional and nonlocal adaptive thresholding (NAT) [47], BDD [44], and CHID [35]).

Objective Performance Analysis. In the paper, we present the objective performance of the proposed method measured in terms of PSNR and SSIM. Table III provides a comparison of the PSNR and SSIM of 24 Kodak images. We can see from Table III that except for image #10, we have the best PSNR and SSIM compared with the existing methods. For image #10, although we have a smaller PSNR, we have the largest SSIM. We have the best average PSNR and SSIM of the Kodak dataset as shown in Table IV. Table V shows the comparison of PSNR and SSIM of 18 McM images and Table VI shows the comparison of average PSNR and SSIM. We can find out that the proposed method achieves the best PSNR and SSIM, with improvements in the average PSNR when compared with AFD, VCD, HOI, ESF, EDAEP, ECI, GD, DFCN, DJDD, FR, ARI, NAT, and BDD.

C. Subjective Performance Analysis

The reconstructed images always have aliasing at image edges due to the signal reconstructed from samples is different from the original continuous signal. In order to illustrate the subjective performance evaluation of the resulting images in terms of visual effect, we show part of the perceived image quality of #6 image as shown in Fig. 7. As shown in Fig. 7, the proposed algorithm has the best results in improved visual quality in terms of the details of the yellow color and edge information compared to the other methods.

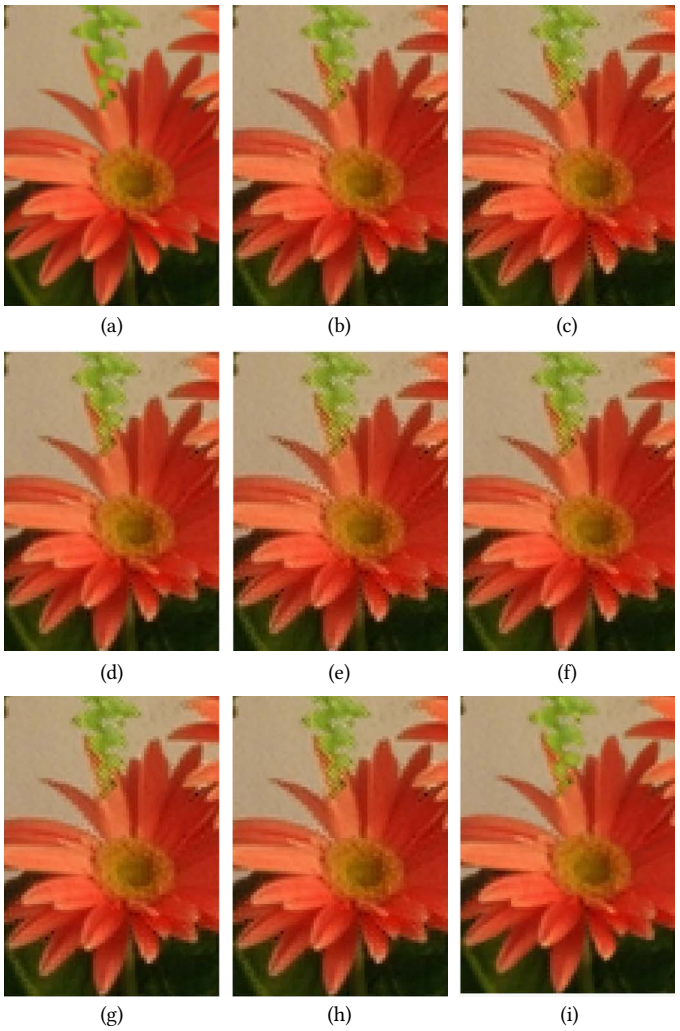


Fig. 7. (a) Part of the original McM #6 image. Perceived image quality comparison using various deinterlacing methods: (b) AFD, (c) VCD, (d) HOI, (e) ESF, (f) EDAEP, (g) ECI, (h) GD, and (i) proposed method.

D. Computional Efficiency Analysis

To achieve high computational efficiency, this paper adopts BCU and ELAN to reduce the network's complexity. As shown in Table VII, we compare the inference time of our proposed algorithm with several conventional methods (RSTCANet [50], RNAN [51], and DRUNet [52]) and observe that it achieves comparable performance.

TABLE VII. Performance Comparison of Inference Time

Method	Proposed	RSTCANet	TCPDNet	DRUNet
Inference time (ms)	72	120	75	226

E. CFA Pattern Analysis

We compare three different CFA patterns—Bayer, Quad, and Q×Q—using PSNR and SSIM, as shown in Tables VIII and IX. The results indicate that our algorithm achieves higher PSNR and SSIM scores compared to other deep learning methods (DFCN [41] and AirNet [53]).

TABLE VIII. Performance Comparison of Average PNSR of McM and Kodak Dataset With Different CFA Patterns

CFA	Proposed	DFCN	AirNet
Bayer	40.96	39.83	40.75
Quad	40.78	39.81	40.63
QxQ	40.73	39.80	40.62

TABLE IX. Performance Comparison of Average SSIM of McM and Kodak Dataset With Different CFA Patterns

CFA	Proposed	DFCN	AirNet
Bayer	0.9928	0.9913	0.9922
Quad	0.9923	0.9911	0.9920
QxQ	0.9912	0.9910	0.9915

F. Joint Demosaicing and Denoising Analysis

We compare the PSNR and SSIM of conventional algorithms and our proposed algorithm under Gaussian noise with = 20, as shown in Table X. The results demonstrate that our proposed algorithm significantly outperforms the other algorithms.

V. Conclusions

In this paper, we proposed an efficient adaptive BP neural network based demosaicking method adopting BCU and ELAN, which is a flexible model and can achieve better demosaicking tasks for different image structures and reduce the complexity of the network. Additionally, we utilize a few surrounding neighboring pixels and hidden layer neurons for training to achieve high efficiency. Our proposed method has a superior performance when compared to conventional methods in both objective and subjective performance.

While our approach demonstrates strong performance in the tested scenarios, it has certain limitations that warrant further exploration. First, the reliance on BDD for initial interpolation may introduce biases specific to that dataset, potentially limiting generalizability to other domains with different image characteristics. Additionally, the threshold-based approach for identifying "complex" images, while effective for Kodak and McM benchmarks, may not be robust across more diverse or unseen datasets. Future work should explore adaptive or learning-based strategies to enhance generalizability and reduce dependence on fixed heuristics. A broader evaluation across varied datasets and real-world conditions would also help clarify the method's applicability and limitations. Future work will focus on enhancing generalizability by incorporating more diverse training data and exploring learning-based approaches for complexity assessment. Additionally, integrating self-supervised or reinforcement learning strategies may help improve adaptability to real-world scenarios. Another promising direction is optimizing computational efficiency to enable real-time processing. By acknowledging these open questions and outlining future improvements, we aim to encourage further research in this area and contribute to the development of more robust and generalizable solutions.

CRediT Authorship Contribution Statement

Gwanggil Jeon: Conceptualization, Methodology, Writing – Original Draft, Writing – Review & Editing, Funding Acquisition, Supervision.

Jin Wang: Investigation, Data Curation, Formal Analysis, Validation.

David Camacho: Writing – Original Draft, Writing – Review & Editing.

Qilei Li: Investigation, Data Curation, Formal Analysis.

Siyou Guo: Investigation, Data Curation, Formal Analysis.

Data Statement

This study did not generate new datasets. The research focuses on methodological development and analysis based on available data and simulation settings described in the manuscript.

Declaration of Conflicts of Interest

We have no conflict of interest to declare.

Acknowledgment

This work was supported by Research Project Support Program for Excellence Institute (2022, ESL) in Incheon National University.

References

[1] R. Lukac, K. N. Plataniotis, D. Hatzinakos, "Color image zooming on the bayer pattern," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1475–1492, 2005.

[2] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, R. M. Mersereau, "Demosaicking: color filter array interpolation," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 44–54, 2005.

[3] J. F. Hamilton, J. E. Adams, "Adaptive color plane interpolation in single sensor color electronic camera." [Patent], 1997.

[4] S. C. Pei, I. K. Tam, "Effective color interpolation in ccd color filter arrays using signal correlation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 503–513, 2003.

[5] L. Chang, Y. P. Tan, "Effective use of spatial and spectral correlations for color filter array demosaicking," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 355–365, 2004.

[6] K. Hirakawa, T. W. Parks, "Adaptive homogeneity-directed demosaicing

TABLE X. Performance Comparison of Average PSNR and SSIM of McM Dataset (The Best Performances Are Marked in **Bold**). DDRCNN Is the Proposed Algorithm

Dataset	Method	VCD	HOI	ESF	BDD	EDAEP	DJDD	DRCNN
Kodak	PSNR	24.021	25.221	26.389	27.356	27.338	27.869	**28.563**
	SSIM	0.7301	0.7621	0.7653	0.7922	0.7910	0.7988	**0.8037**
McM	PSNR	24.369	25.337	26.387	27.661	27.897	28.902	**29.081**
	SSIM	0.7512	0.7635	0.7728	0.7936	0.7933	0.7983	**0.8033**

algorithm," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.

[7] N. X. Lian, L. Chang, Y. P. Tan, V. Zagorodnov, "Adaptive filtering for color filter array demosaicking," *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2515–2525, 2007.

[8] K. H. Chung, Y. H. Chan, "Color demosaicing using variance of color differences," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 2944–2955, 2006.

[9] D. Menon, S. Andriani, G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 132–141, 2006.

[10] Z. Dengwen, S. Xiaoliu, D. Weiming, "Colour demosaicking with directional filtering and weighting," *IET Image Processing*, vol. 6, no. 8, pp. 1084–1092, 2012.

[11] J. S. J. Li, S. Randhawa, "Color filter array demosaicking using high-order interpolation techniques with a weighted median filter for sharp color edge preservation," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 1946–1957, 2009.

[12] C. Y. Su, W. C. Kao, "Effective demosaicing using subband correlation," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 1, pp. 199–204, 2009.

[13] W. J. Chen, P. Y. Chang, "Effective demosaicking algorithm based on edge property for color filter arrays," *Digital Signal Processing*, vol. 22, no. 1, pp. 163–169, 2012.

[14] I. Pekkucuksen, Y. Altunbasak, "Edge strength filter based color filter array interpolation," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 393–397, 2011.

[15] I. Pekkucuksen, Y. Altunbasak, "Multiscale gradients-based color filter array interpolation," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 157–165, 2012.

[16] J. Kim, G. Jeon, J. Jeong, "Demosaicking using geometric duality and dilated directional differentiation," *Optics Communications*, vol. 324, pp. 194–201, 2014.

[17] D. Menon, G. Calvagno, "Regularization approaches to demosaicking," *IEEE transactions on image processing*, vol. 18, no. 10, pp. 2209–2220, 2009.

[18] E. Dubois, "Frequency-domain methods for demosaicking of bayer-sampled color images," *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 847–850, 2005.

[19] B. Leung, G. Jeon, E. Dubois, "Least-squares luma–chroma demultiplexing algorithm for bayer demosaicking," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1885–1894, 2011.

[20] G. Jeon, E. Dubois, "Demosaicking of noisy bayer-sampled color images with least-squares luma-chroma demultiplexing and noise level estimation," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 146–156, 2012.

[21] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[22] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[23] R. Zhang, P. Isola, A. A. Efros, "Colorful image colorization," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, 2016, pp. 649–666, Springer.

[24] S. Iizuka, E. Simo-Serra, H. Ishikawa, "Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 4, pp. 1–11, 2016.

[25] J. Long, E. Shelhamer, T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[26] H. Noh, S. Hong, B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[27] S. Xie, Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403.

[28] W. Shen, X. Wang, Y. Wang, X. Bai, Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3982–3991.

[29] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[30] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[31] K.-L. Hua, C.-H. Hsu, S. C. Hidayati, W.-H. Cheng, Y.-J. Chen, "Computer-aided classification of lung nodules on computed tomography images via deep learning technique," *OncoTargets and Therapy*, pp. 2015–2022, 2015.

[32] K. A. Jangtjik, M.-C. Yeh, K.-L. Hua, "Artist-based classification via deep learning with multi-scale weighted pooling," in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016, pp. 635–639.

[33] J. Kim, J. K. Lee, K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[34] K. Zhang, W. Zuo, L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3262–3271.

[35] S. Wang, M. Zhao, R. Dou, S. Yu, L. Liu, N. Wu, "A compact high-quality image demosaicking neural network for edge-computing devices," *Sensors*, vol. 21, no. 9, p. 3265, 2021.

[36] M. Gharbi, G. Chaurasia, S. Paris, F. Durand, "Deep joint demosaicking and denoising," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.

[37] N. S. Syu, Y. S. Chen, Y.-Y. Chuang, "Learning deep convolutional networks for demosaicing," *arXiv preprint arXiv:1802.03769*, 2018.

[38] F. Kokkinos, S. Lefkimmiatis, "Deep image demosaicking using a cascade of convolutional residual denoising networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 303–319.

[39] B. Park, J. Jeong, "Color filter array demosaicking using densely connected residual network," *IEEE Access*, vol. 7, pp. 128076–128085, 2019.

[40] R. Tan, K. Zhang, W. Zuo, L. Zhang, "Color image demosaicking via deep residual learning," in *Processing IEEE International Conference on Multimedia Expo (ICME)*, vol. 2, 2017, p. 6.

[41] D. S. Tan, W. Y. Chen, K. L. Hua, "Deepdemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2408–2419, 2018.

[42] B. Xia, Y. Zhang, Y. Wang, Y. Tian, W. Yang, R. Timofte, L. Van Gool, "Basic binary convolution unit for binarized image restoration network," *arXiv preprint arXiv:2210.00405*, 2022.

[43] C. Y. Wang, H. Y. M. Liao, I. H. Yeh, "Designing network design strategies through gradient path analysis," *arXiv preprint arXiv:2211.04800*, 2022.

[44] J. Wang, J. Wu, Z. Wu, G. Jeon, J. Jeong, "Bilateral filtering and directional differentiation for bayer demosaicking," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 726–734, 2016.

[45] Z. Liu, Z. Shen, M. Savvides, K. T. Cheng, "Reactnet: Towards precise binary neural network with generalized activation functions," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, 2020, pp. 143–159, Springer.

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

[47] L. Zhang, X. Wu, A. Buades, X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic Imaging*, vol. 20, no. 2, pp. 023016–023016, 2011.

[48] J. Wu, R. Timofte, L. Van Gool, "Demosaicing based on directional difference regression and efficient regression priors," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3862–3874, 2016.

[49] D. Kiku, Y. Monno, M. Tanaka, M. Okutomi, "Residual interpolation for color image demosaicking," in *2013 IEEE International Conference on Image Processing*, 2013, pp. 2304–2308, IEEE.

[50] W. Xing, K. Egiazarian, "Residual swin transformer channel attention network for image demosaicing," in *2022 10th European Workshop on Visual Information Processing (EUVIP)*, 2022, pp. 1–6, IEEE.

[51] Y. Zhang, K. Li, K. Li, B. Zhong, Y. Fu, "Residual non-local attention networks for image restoration," *arXiv preprint arXiv:1903.10082*, 2019.

[52] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6360–6376, 2021.

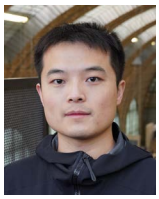
[53] B. Li, X. Liu, P. Hu, Z. Wu, J. Lv, X. Peng, "All-in- one image restoration for unknown corruption," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17452–17462.

Jin Wang

Jin Wang received a BS in mathematics and applied mathematics from Zhejiang University in 2007, and an MS and PhD in electronic communications engineering from Hanyang University. She was a professor at Xidian University. She is a professor at Incheon National University.

Siyou Guo

Siyou Guo is pursuing an M.S. degree at the School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo, China. His research interests include misinformation detection, deepfake detection, deep learning, and computer vision.



Qilei Li

Qilei Li received the Ph.D. in Computer Science from Queen Mary University of London. He previously earned an M.S. degree from Sichuan University in 2020. From June 2022 to April 2024, he worked as a machine learning scientist at Veritone Inc, where he focused on developing a scalable person search framework for retrieving individuals at different locations and times, as captured by various cameras. His current research interests lie in privacyaware machine learning, with a particular emphasis on learning domain-invariant knowledge representation from multimodal data captured in diverse environments. His research outcome has been recognized as ESI Highly Cited Paper (Top 1%). Additionally, he serves as an evaluator for the ELLIS PhD Program.

David Camacho

David Camacho is full professor at Computer Systems Engineering Department of Universidad Politécnica de Madrid (UPM), and the head of the Applied Intelligence and Data Analysis research group (AIDA: https://aida.etsisi.uam.es) at UPM. He holds a Ph.D. in Computer Science from Universidad Carlos III de Madrid in 2001 with honors (best thesis award in Computer Science). He has published more than 300 journals, books, and conference papers. His research interests include Machine Learning (Clustering/Deep Learning), Computational Intelligence (Evolutionary Computation, Swarm Intelligence), Social Network Analysis, Fake News and Disinformation Analysis. He has participated/led more than 50 research projects (Spanish and European: H2020, DG Justice, ISFP, and Erasmus+), related to the design and application of artificial intelligence methods for data mining and optimization for problems emerging in industrial scenarios, aeronautics, aerospace engineering, cybercrime/cyber intelligence, social networks applications, or video games among others. He has served as Editor in Chief of Wiley's Expert Systems since 2023 and sits on the Editorial Board of several journals, including Information Fusion, IEEE Transactions on Emerging Topics in Computational Intelligence (IEEE TETCI), Human-centric Computing and Information Sciences (HCIS), and Cognitive Computation among others.

Gwanggil Jeon

Gwanggil Jeon received the B.S., M.S., and Ph.D. (summa cum laude) degrees from the Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea, in 2003, 2005, and 2008, respectively. From 2009.09 to 2011.08, he was with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada, as a Post-Doctoral Fellow. From 2011.09 to 2012.02, he was with the Graduate School of Science and Technology, Niigata University, Niigata, Japan, as an Assistant Professor. From 2014.12 to 2015.02 and 2015.06 to 2015.07, he was a Visiting Scholar at Centre de Mathématiques et Leurs Applications (CMLA), École Normale Supérieure Paris-Saclay (ENS-Cachan), France. From 2019 to 2020, he was a Prestigious Visiting Professor at Dipartimento di Informatica, Università degli Studi di Milano Statale, Italy. He is currently a Full Professor Incheon National University, Incheon, Korea.