Pattern Recognition 157 (2025) 110929

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Joint learning of RGBW color filter arrays and demosaicking

Chenyan Bai [a], Faqi Liu [a], Jia Li [b,*]

[a] *College of Information Engineering, Capital Normal University, Beijing, 100048, China*
[b] *School of Artificial Intelligence, Beijing Normal University, Beijing, 100875, China*

ARTICLE INFO

Keywords:
Demosaicking
Color filter array (CFA)
RGBW
End-to-end

ABSTRACT

RGBW color filter arrays (CFAs) have gained widespread attention for their superior performance in low-light conditions. Most existing demosaicking methods are tailored for specific RGBW CFAs or involve manual joint design with CFAs. The close relationship between sampling and reconstruction means that restricting the search space through predefined CFAs and demosaicking methods severely limits the ability to achieve optimal performance. In this paper, we propose a new approach for joint learning of RGBW CFA and demosaicking. This approach can simultaneously learn optimal CFAs and demosaicking methods of any size, while also being capable of reconstructing mosaicked images of any size. We use a surrogate function and arbitrary CFA interpolation to ensure end-to-end learning of the RGBW CFA. We also propose a dual-branch fusion reconstruction network that utilizes the W channel to guide the reconstruction of R, G, and B channels, reducing color errors while preserving more image details. Extensive experiments demonstrate the superiority of our proposed method. Our code is available at: https://github.com/fql0528/learncfa.

1. Introduction

Digital color imaging devices usually employ color filter arrays (CFAs) that are integrated on the sensor surface to capture color images. The CFA has the same size as the sensor, with each pixel of the CFA allowing only a single color of light to pass through. Consequently, the resulting mosaicked image is composed of pixels with only one color component, which does not have to be red (R), green (G), or blue (B). However, color images are commonly composed of RGB three color components. Therefore, it is necessary to reconstruct RGB color images from mosaicked images, a process known as demosaicking [1,2]. It is an essential step of the image signal processing (ISP) pipelines.

Color imaging with a CFA is a sampling process, while demosaicking is the reconstruction process. Therefore, both the CFA and the corresponding demosaicking method are crucial for obtaining high-quality demosaicked images. To date, the Bayer CFA [3] has been the most widely used CFA in the industry (see Fig. 1(a)). This CFA is constructed with the smallest repeating pattern of 2 × 2, where each pattern comprises one R pixel, two diagonally paired G pixels, and one B pixel. While the Bayer CFA is commonly used, it blocks most incident light [4]. In low-light conditions, this reduces the signal-to-noise ratio (SNR) and presents a significant challenge in reconstructing color images.

While many demosaicking methods for the Bayer CFA have achieved significant advancements in enhancing the quality of color imaging in low-light conditions, there is a failure to fully exploit the incident light. To overcome this limitation of the Bayer CFA, researchers have suggested various CFAs that include panchromatic (or white (W)) pixels. These CFAs comprise only R, G, B, and W pixels, and hence called RGBW CFAs. On the one hand, W pixels transmit light in all colors. So W has a higher spectral sensitivity than R, G, and B [4,5]. On the other hand, the demosaiced images are RGB images. Including RGB pixels assists in establishing the parameters of ISP. More importantly, the RGBW CFAs are easy to manufacture and have been used in surveillance cameras and mobile phones [2]. The greater the number of W pixels in a CFA, the higher the light utilization efficiency of the CFA. However, W pixels do not capture RGB color information and may result in a decrease in color fidelity. So a trade-off exists between the ratios of RGB and W pixels in an RGBW CFA. Previous works focus on developing demosaicking methods for a given RGBW CFA [2,5], creating RGBW CFAs for a specific demosaicking method [7,8], or manually co-designing the RGBW CFA and demosaicking method [4]. As a result, the available search space is significantly limited, which hinders the achievement of optimal performance due to the inherent relationship between sampling and reconstruction.

Jointly learning the CFA and demosaicking method is a promising solution. For example, Henz et al. [9] constrained each color filter of the CFA to a linear combination of the RGB primary filters, resulting in a differentiable problem. Then they performed joint design of the CFA and demosaicking using a convolutional neural network. They

[*] Corresponding author.
E-mail address: jiali@bnu.edu.cn (J. Li).

https://doi.org/10.1016/j.patcog.2024.110929
Received 23 February 2024; Received in revised form 1 August 2024; Accepted 21 August 2024
Available online 23 August 2024

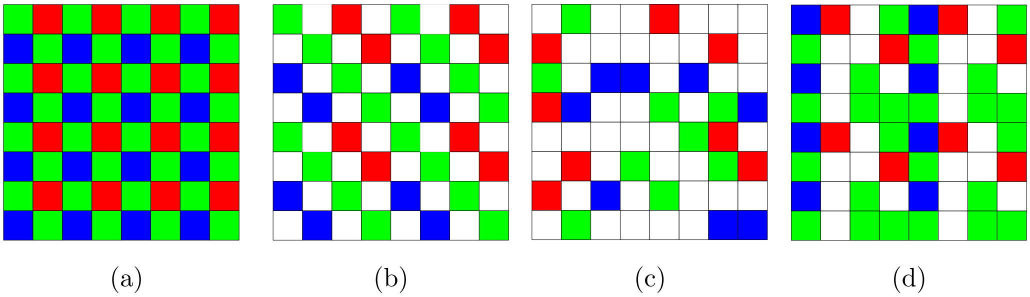

Fig. 1. The Bayer CFA and three RGBW CFAs. (a) is the Bayer CFA [3]. (b) is the Kodak CFA [5]. (c) is the Chakrabarti CFA [6]. (d) is our learned CFA.

demonstrated that their approach could achieve higher CPSNR values. However, their approach cannot be used to learn RGBW CFAs and it is physically infeasible to develop their learned CFAs [2]. Chakrabarti [6] jointly learned the RGBW CFA and demosaicking (see Fig. 1(c)). Designing an RGBW CFA involves solving a non-differentiable channel selection problem. To address this, they used a softmax function with a temperature coefficient to approximate the non-differentiable channel selection. They also trained the CFA pattern and demosaicking method together using a dual-branch reconstruction network. However, *their approach needed the CFA pattern to match the training image patches (see Section 2.3), limiting the flexibility of joint learning. More importantly, it required the input images for both training and testing are of the same size, making it impossible to train on image patches and test on full-size images. We argue that the main challenges lie in conducting channel selection and handling missing values (see Section 2.4).*

To address the limitations mentioned above, we propose a new approach to simultaneously learn the RGBW CFA and demosaicking. We model the joint learning approach as an autoencoder, which consists of an encoder network and a decoder network (see Fig. 2). The encoder network mimics the color imaging with an RGBW CFA and generates an interpolated image with RGBW channels. The decoder network reconstructs the interpolated image into an RGB image. More specifically, our solution includes three sequential components: channel selection, arbitrary CFA interpolation, and reconstruction network. For each pixel of the RGBW CFA, channel selection selects one from R, G, B and W, which is non-differentiable and cannot be trained using gradient-based methods. To address this, we borrow the idea of training binary neural networks (see Section 3.2). Throughout the training process, the learned RGBW CFA is constantly changing. We utilize arbitrary CFA interpolation to restore the missing values of the changing CFA (see Section 3.2). The reconstruction network must leverage the CFA's characteristics to achieve superior performance with a relatively small network. To achieve this, we introduce a dual-branch fusion reconstruction network to further enhance network performance (see Section 3.3). Our learned RGBW CFA is shown in Fig. 1(d).

The contributions of this paper are:

- We propose a novel joint learning approach for RGBW CFA design and demosaicking, which is modeled as an autoencoder. It is capable of learning CFA patterns of arbitrary sizes and reconstructing mosaicked images of different sizes. Our approach outperforms the existing solutions for both noise-free and noisy scenarios.
- We use the maximum indicator operator to simulate non-differentiable channel selection. Following the spirit of training binary neural networks, we employ the softmax function with a temperature coefficient as the surrogate function to mimic gradient propagation, ensuring the end-to-end learnability of RGBW CFA design.
- We propose a simple interpolation method for any RGBW CFAs. It adds the values of neighboring known pixels together, with each weighted by their inverse distance to the point of interest. It is very important for end-to-end learning of an RGBW CFA.
- We propose a dual-branch reconstruction network that is lightweight and high-performing, taking into consideration the higher ratio and SNR of W pixels compared to RGB pixels.

2. Related work

2.1. CFA design

Various RGBW CFAs have been proposed for different considerations. The Kodak CFA [5] is the most frequently used RGBW CFA. It is a 4 × 4 CFA pattern and contains 2 × 2 groups of pixels (see Fig. 1(b)). Honda et al. [10] suggested the 4 × 4 CFA pattern with 75% W pixels, where the RGB pixels are assigned in a downsampled checkerboard. Hamilton and Compton [11] developed a 8 × 8 CFA pattern with 50% W pixels. Based on the frequency structure [12], Li et al. [7] formulated the CFA design with W pixels as a multi-objective optimization problem, which is fully automatic. Kaizu [13] suggested a 6 × 6 CFA pattern with 50% W pixels [13], which contains an equal percentage of R, G, and B pixels. Chakrabarti et al. [4] designed a CFA pattern by placing the 2 × 2 Bayer CFA pattern into 6 × 6 W patterns. For more details on RGBW CFAs, we refer the readers to [2].

2.2. Demosaicking methods

Many demosaicking methods have been proposed for RGBW CFAs. They are based on reasonable assumptions [2], designed for W-dominant CFAs [4], based on pansharpening [14], in the frequency domain [7], and based on the reconstruction strategy [15]. Deep neural networks (DNNs) have made remarkable progress in Bayer demosaicking [16]. Sharif and Jung [8] presented a deep learning-based demosaicking method for a CFA with 99% W pixels, which consists of two sequential steps. Notably, they developed a prototype system. However, their method requires heavy computational resources. Recently, researchers have realized that DNNs need hard training images to learn effective models for demosaicking. Gharbi et al. [16] employed a pretrained network and appropriate metrics to select hard training images. Sun et al. [17] introduced a data-driven approach to automatically select informative images from a large-scale dataset.

2.3. Joint design of RGBW CFA and demosaicking

The design of an RGBW CFA is a channel selection task for each pixel. It is non-differentiable and cannot be directly learned by gradient-based optimization methods, e.g., SGD and Adam [18]. Chakrabarti [6] asymptotically approximated the channel selection using a softmax function with a temperature coefficient. Their reconstruction network has two paths. One path uses a fully-connected layer without a bias term in the log-domain. The other path uses cascade of convolutional layers. Since the reconstruction network uses fully-connected layers, it only works with the situation that the CFA pattern matches the to-be-reconstructed image, which restricts the flexibility of joint learning.

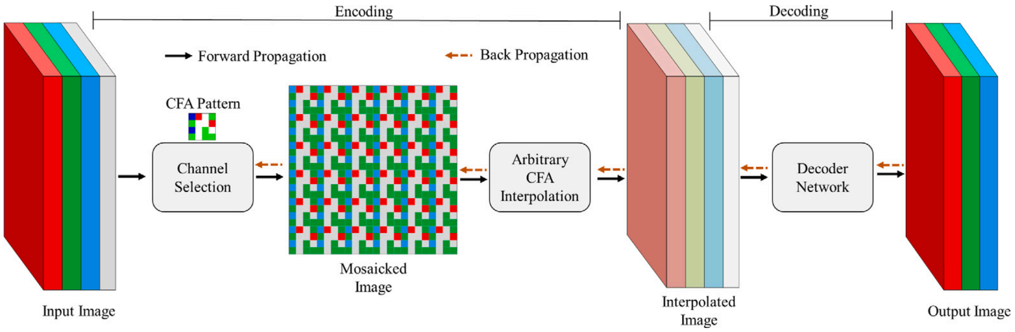

Fig. 2. Our autoencoder architecture for joint learning of the RGBW CFA pattern and demosaicking method. The encoder network optimizes the non-differentiable channel selection while also performing the interpolation for arbitrary RGBW CFA. The decoder network is responsible for reconstructing the demosaicked images. *In comparison to [6], our autoencoder can learn CFA patterns of any size, and is able to train on image patches and test on full-size images.* Here we use a 4 × 4 RGBW CFA as an example.

2.4. Handling missing values

When applying DNNs to reconstruct a periodic mosaicked image, the initial step is to address missing values, which can be achieved through techniques such as rearrangement [16], strided convolution [6], or interpolation [19]. The rearrangement technique rearranges the mosaicked image into a multi-channel image scaled down by the CFA pattern size. Since RGBW CFAs are typically larger than 4 × 4, which weakens the full-size spatial information in the mosaicked image, thus increasing the difficulty of subsequent network reconstruction. The strided convolution technique uses the standard convolution with a CFA pattern size stride, which reduces the dimensionality drastically. The interpolation technique recovers missing values for each channel by exploring spatial correlation within the same channel. It maintains the spatial information in the mosaicked image but can lead to spatial–spectral aliasing in the interpolated image. As the interpolation technique compromises spatial and spectral correlations, we choose it to address missing values. However, most existing interpolation methods are bilinear and tailored for the Bayer CFA. For the joint learning approach in [6], each pixel value of the mosaicked image is derived from a convex combination of RGBW filters. Each pixel in the CFA pattern should be split into different channels. Known pixels in the same channel are too distant, making interpolation challenging.

3. Proposed method

In this section, we first give an overview of the proposed autoencoder architecture for joint learning of the RGBW CFA pattern and demosaicking method (see Fig. 2). Then we provide a detailed description of the encoder network (see Fig. 3) and decoder network (see Fig. 5).

3.1. Method overview

Our approach to jointly learning RGBW CFA and demosaicking involves training an autoencoder, comprising an encoder network and a decoder network (see Fig. 2). The encoder network projects the input RGBW image based on the current RGBW CFA, which is a non-differentiable channel selection problem. The result is a mosaicked image, where each pixel is selected from R, G, B, and W. Since the CFA changes during training, we perform arbitrary CFA interpolation. Our linear interpolation method produces an initial estimate of the input RGBW image, which enhances results and accelerates convergence. Finally, the interpolated image, without missing values, is fed into the decoder network, which outputs the demosaicked image with RGB channels. This constitutes the forward propagation of our autoencoder.

The loss function measures the difference between the ground truth and the demosaicked image through the mean squared error (MSE). Throughout training, we can learn two trainable parameters, the CFA pattern and the decoder network weights, to optimize the autoencoder's performance. However, the encoder network encounters the challenge of non-differentiable channel selection and cannot be trained using conventional gradient-based methods. To address this, we draw inspiration from training binary neural networks and employ the softmax function with a temperature coefficient as a substitute for gradient propagation, ensuring the end-to-end trainability of the encoder network.

The encoder is limited by physical constraints imposed by the actual CFA construction. It does not utilize fully-connected layers. The decoder network only employs convolutional and nonlinear activation layers. Therefore, the encoder and decoder networks support images of varying sizes. We train the autoencoder using small image patches and can still reconstruct images with different resolutions without the need to break the image into patches.

3.2. Encoder network

The encoder network learns the RGBW CFA and performs arbitrary RGBW CFA interpolation. We demonstrate our encoder network in Fig. 3.

3.2.1. Channel selection

Channel selection is to select a color from R, G, B, and W for each pixel of the RGBW CFA, which is non-differentiable. Suppose the size of the to-be-learned CFA pattern is $m \times n$. Then we have $N = m \times n$ different color filters. For the ith filter, we use the weight vector $P_i = (P_i^1, P_i^2, P_i^3, P_i^4)^T \in \mathbb{R}^4$ to denote the importance of R, G, B, and W. We choose the largest one in P_i as the color of the ith filter, which is computed as

$$X_i^j = \begin{cases} 1, & P_i^j = \max\,(P_i); \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We denote $X_i = f(P_i) = (X_i^1, X_i^2, X_i^3, X_i^4)^T$. There is only one element in X_i is 1, and the others are 0. X_i can be used to mimic the ith pixel of the RGBW CFA. We can see that the maximum indicator function $X_i = f(P_i)$ is non-differentiable w.r.t. P_i. So we cannot directly update P_i using gradient-based methods.

CFA is created by periodically tiling a CFA pattern. We utilize disjoint binary masks to enforce pattern repetition and to handle images of any size. These masks can also impose additional constraints. For example, they can be used to enforce designs that adhere to specific patterns. In this paper, we do not impose such constraints. X_i has a corresponding binary mask pattern with the same size as the CFA pattern, where the ith element is 1 and the rest are 0. We replicate the mask pattern to align with the CFA size and get Mask_i. Then we have the mosaicked image generated by the ith color filter:

$$\text{Submosaick}_i = (X_i \odot I_{\text{input}}) \odot \text{Mask}_i, \tag{2}$$

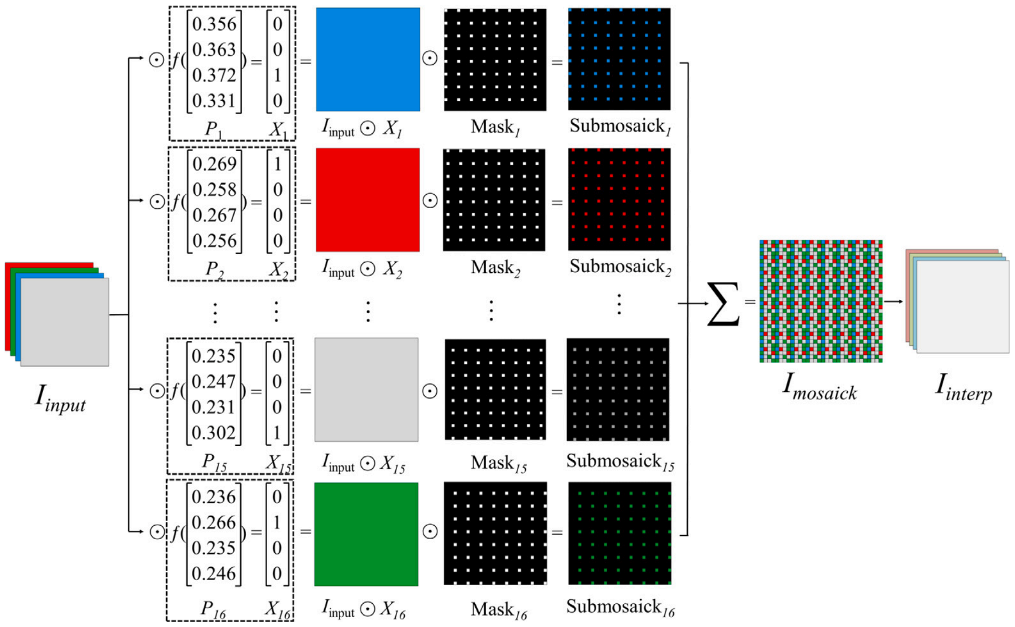
I_{input}
$\odot f(\begin{bmatrix}0.356\\0.363\\0.372\\0.331\end{bmatrix}) = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$
P_1
X_1
$I_{\text{input}} \odot X_1$
Mask_1
Submosaick_1
$\odot f(\begin{bmatrix}0.269\\0.258\\0.267\\0.256\end{bmatrix}) = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$
P_2
X_2
$I_{\text{input}} \odot X_2$
Mask_2
Submosaick_2
$\odot f(\begin{bmatrix}0.235\\0.247\\0.231\\0.302\end{bmatrix}) = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$
P_{15}
X_{15}
$I_{\text{input}} \odot X_{15}$
Mask_{15}
Submosaick_{15}
$\odot f(\begin{bmatrix}0.236\\0.266\\0.235\\0.246\end{bmatrix}) = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$
P_{16}
X_{16}
$I_{\text{input}} \odot X_{16}$
Mask_{16}
Submosaick_{16}
$\sum =$
$I_{mosaick}$
I_{interp}

Fig. 3. Our proposed encoder network. We use the 4 × 4 CFA pattern size for illustration. $P_i = (P_i^1, P_i^2, P_i^3, P_i^4)^T$ represents the weights of the ith pixel of the current RGBW CFA pattern ($i = 1, \ldots, 16$). f represents the maximum indicator function in Eq. (1). The ith color filter is represented by a binary vector X_i and a binary mask Mask$_i$, which generates the sub-mosaicked image Submosaick$_i$. The mosaicked image $I_{mosaick}$ is obtained by adding up all the sub-mosaicked images. An RGBW image I_{interp} without missing values are obtained by our interpolation method.

where I_{input} represents the input RGBW image, and $\odot$ denotes the element-wise multiplication with the broadcasting rule in Pytorch. Therefore, the mosaicked image generated by the CFA with N color filters is calculated as:

$$I_{\text{mosaick}} = \sum_{i=1}^{N} \text{Submosaick}_i. \tag{3}$$

It is worth to note that we initialize every element of P_i to a small constant (e.g., 0.001) to facilitate quick and efficient learning of the optimal CFA pattern. During training, P_i and X_i are continuously updated to minimize the reconstruction error, while Mask$_i$ remains constant.

Optimizing P_i involves a non-differentiable function f, which makes the standard back-propagation inapplicable. To solve this problem, we borrow inspiration from training binary neural networks [20] and use the Straight-Through Estimator (STE) [21]. It is an effective approach to address this problem to construct a non-trivial search direction by modifying the chain rule appropriately. This can be achieved by replacing the zero derivative with a related surrogate. We utilize the softmax function with a temperature coefficient as the surrogate function:

$$\phi(x)_i = \frac{\exp(\alpha x_i)}{\Sigma_{j=1}^{k} \exp(\alpha x_j)}, \tag{4}$$

where $x = (x_1, x_2, \ldots, x_k)^T$, and $\alpha > 0$ is the temperature coefficient. For a given x, $\phi(x)$ becomes a zero–one vector as $\alpha \to \infty$. Note that we use a fixed α during training. So $\phi(x)$ is an appropriate surrogate function to f. In the forward pass, we use f to simulate the color imaging with an RGBW CFA. In the backward pass, we employ ϕ in Eq. (4) to get a non-trivial gradient for updating P_i. This guarantees end-to-end learning of our autoencoder.

3.2.2. Arbitrary CFA interpolation

To feed the mosaicked image into a decoder network, it must first be interpolated to obtain a full-size estimation without missing values (see Section 2.4). However, the CFA pattern undergoes constant changes during the learning process, making the interpolation method for a specific RGBW CFA pattern obsolete. To solve this issue, we propose an interpolation method that utilizes the properties of the Gaussian function, allowing it to be applied universally to any CFA. Our arbitrary CFA interpolation for each channel is computed as:

$$Z = g(Y) \oslash g(M), \tag{5}$$

where Y is one channel of the mosaicked image and all the missing values are set to 0, M is the binary mask that indicates known pixels in Y, $\oslash$ is the element-wise division, and g is a Gaussian filter that follows a Gaussian distribution with mean 0 and variance σ^2. The values in the Gaussian filter g are used to represent the importance of interpolation weights, where the closer a point is to the center, the greater its weight. The window size of g can be given, such as 5 × 5 or 7 × 7, but the window size should ensure that it can cover the maximum distance of unknown pixels in each channel. In the experiments, we simply set $\sigma^2 = (w + 1)/6$, where w is the window size.

The scattering-weighted interpolation method proposed by Nguyen et al. [22] can address the interpolation problem for any CFA. However, it does not utilize the information of missing pixels during interpolation. This results in the gradients of zero values in X_i always being zero during backpropagation. As a result, the corresponding values in P_i cannot be updated, leading to the failure of learning of the CFA pattern (see Fig. 4). Our interpolation method can handle any CFA interpolation problem and ensure end-to-end learning of RGBW CFA (see Fig. 4), with greater efficiency compared to scattering-weighted interpolation.

3.3. Decoder network

The decoder network must effectively utilize intra- and inter-channel dependencies. In a mosaicked image, the number and SNR of W pixels are generally higher than those of R, G, and B pixels. Thus, the W channel is more reliable in the interpolated image. Additionally, the W channel shares similar edges with the R, G, and B channels. Therefore, making full use of the W channel can aid in the recovery of missing pixel values. This leads to the traditional sequential methods [2]. They first interpolate the W channel and then recover the RGB channels with reference to the interpolated W channel by exploiting the inter-channel correlation. To this end, we propose a dual-branch fusion decoder network (see Fig. 5 (a)), which consists of dual-branch feature

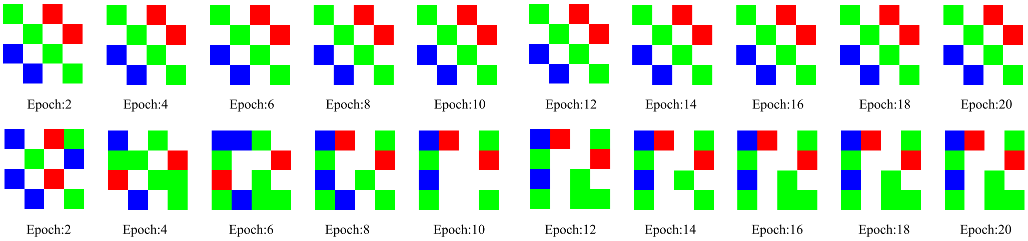

Fig. 4. Evolution of CFA patterns during training with two interpolation methods. The top row shows scattering-weighted interpolation, and the bottom row shows our interpolation method. It is evident that the CFA pattern in the top row cannot be updated, whereas the CFA pattern in the bottom row can be updated.

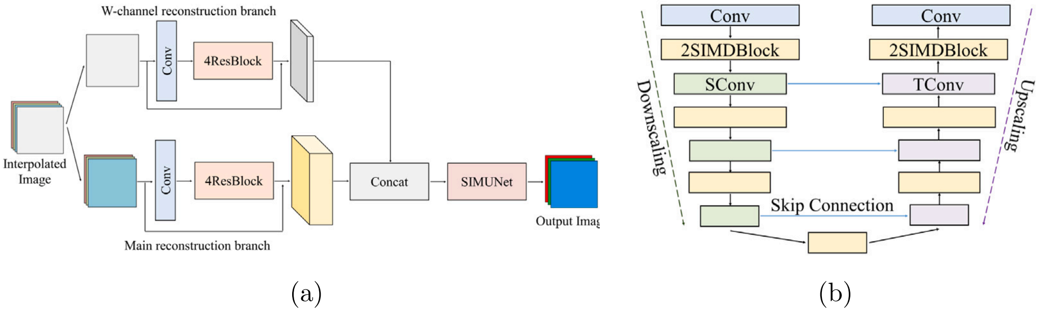

Fig. 5. Our decoder network. (a) Decoder network with two branches. (b) SIMUNet.

extraction, fusion module, and reconstruction module. This not only enhances SNR values but also diminishes false color artifacts, further improving the quality of demosaicked images.

The dual-branch feature extraction extracts shallow features from the W channel and uses it to guide the reconstruction of RGB channels. The fusion module concatenates the extracted shallow features of the W channel and RGB channels. Shallow features contain abundant image geometric information [23], such as edges and textures, which are crucial for recovering fine-grained image details. Therefore, we only extract the shallow features of the interpolated image in the dual branches for guidance and fusion, and then send them to the reconstruction module for deep feature extraction and reconstruction. The W channel is fed into the W-channel reconstruction branch to extract shallow features, while the RGB channels are processed by the main reconstruction branch. The two branches consist of multiple residual blocks (ResBlocks). The ResBlock includes residual connections, regular convolutions, and LeakyReLU activation functions, which can fully explore and extract shallow features of the W channel and RGB channels.

The proposed reconstruction module is the SIMUNet (see Fig. 5 (b)). For the architecture design, we employ the famous UNet [24] and a simplified information multi-distillation block (SIMDBlock) [25]. We evaluate other state-of-the-art (SOTA) lightweight blocks, like the residual local feature block [26], and surprisingly discover that SIMDBlock outperforms the rest in our scenario. UNet is widely used for various image reconstruction tasks. The SIMDBlock is a simplified version of the information multi-distillation block (IMDBlock) developed by Hui et al. [25], which has demonstrated exceptional performance in lightweight super-resolution. It comprises the Progressive Refinement Module (PRM) and the Selective Fusion Module (SFM). The PRM extracts features at a granular level, preserving partial information and further processing other features at each refinement layer. The SFM primarily employs Contrastive Channel Attention (CCA) layers to aggregate the features extracted through distillation steps, enhancing the collected refinement information and utilizing more useful features (e.g., edges, corners, and textures) to restore high-quality images. Inspired by the super-resolution network architecture in [27], we adopt a simplified multi-distillation module that only consists of the progressive refinement module. This SIMDBlock not only restores high-quality images but also possesses lightweight characteristics. Therefore, we integrate the SIMDBlock into UNet to efficiently reconstruct high-quality color images and handle various levels of noise. The backbone of the SIMUNet is the UNet architecture, which includes a contraction path (encoding) and an expansion path (decoding). Our reconstruction module comprises four scales, each scale composed of two consecutive SIMDBlocks and one upsampling or downsampling module. Unlike UNet, we use 2 × 2 stride convolution for downsampling. There is a skip connection between each scale's downsampling and upsampling operations. The number of channels in each layer varies from the first scale to the fourth scale, being 64, 128, 256, and 512, respectively. Inspired by the super-resolution network architecture design from [28], there are no activation functions following the first and last convolution (Conv) layers, after the SConv and TConv layers. Note that SConv and TConv denote the 2 × 2 strided convolution downscaling and 2 × 2 transposed convolution upscaling operations, respectively. Additionally, each SIMDBlock uses the LeakyReLU activation function.

4. Experiments

In this section, we first validate the effectiveness of our joint learning approach. Then we evaluate the performance of our decoder network with the Kodak CFA [5]. Next, we test the advantage of our encoder network with our decoder network. Finally, we conduct ablation studies and discuss the limitations and possible extensions of our approach.

4.1. Experimental settings

4.1.1. Dataset and metrics

Since there is no public RGBW dataset, we have to simulate the W channels. Following the approach in [6], we simulate the W channel by summing the R, G, and B channels. We use the Gehler-Shi dataset.[1] It consists of 568 linear RGB images that have not been processed by ISP. As in [6], we use 461 images as the training set, 51 images

[1] https://www2.cs.sfu.ca/~colour/data/shi_gehler/.

Table 1
Comparison of our joint learning approach. The average CPSNR, SSIM, and S-CIELAB values for different noise levels are given for images normalized in [0,1]. Note that "STD = 0" means the noise-free case.

	CFA pattern+Method	STD = 0	STD = 0.01	STD = 0.015
CPSNR↑	Bayer CFA [3]+RI [31]	58.15	49.87	47.07
	Kodak CFA [5]+Guided filter [2]	57.23	53.12	51.83
	Chakrabarti's approach [6]	56.00	51.98	50.96
	Our approach	**60.40**	**55.33**	**53.99**
SSIM↑	Bayer CFA [3]+RI [31]	0.9979	0.9855	0.9714
	Kodak CFA [5]+Guided filter [2]	0.9975	0.9934	0.9885
	Chakrabarti's approach [6]	0.9957	0.9898	0.9870
	Our approach	**0.9990**	**0.9964**	**0.9947**
S-CIELAB↓	Bayer CFA [3]+RI [31]	**0.00039**	0.00110	0.00160
	Kodak CFA [5]+Guided filter [2]	0.00063	0.00086	0.00096
	Chakrabarti's approach [6]	0.00068	0.00130	0.00150
	Our approach	0.00053	**0.00076**	**0.00079**

as the validation set to tune hyperparameters, and the remaining 56 images as the test set. We extract non-overlapping image patches of size 128×128 for training without any augmentation. We test the trained networks on the full-size images. We utilize the CPSNR, SSIM [29], and S-CIELAB [30] metrics to evaluate the performance. A higher CPSNR or SSIM value, or a lower S-CIELAB value, may indicate a potentially better image reconstruction quality. We also conduct visual evaluations.

4.1.2. Training details

We employ the same training procedure as in [6]. We add different levels of zero mean Gaussian noise to simulate low-light imaging. We learn the RGBW CFA with a moderate noise level, where the standard deviation (STD) is 0.01 for images normalized in [0,1]. Then we fix the learned CFA pattern and train a decoder network for each noise level. Our autoencoder is implemented in PyTorch and runs on an NVIDIA TITAN RTX GPU. We use the Adam optimizer [18] (β_1=0.9, β_2=0.999, ϵ=1×10^{-2}) with a batch size of 8. We train our autoencoder and decoder network for 20 epochs. While training the autoencoder, the learning rates for the encoder and decoder networks are set to 1×10^{-3} and 2×10^{-4}, respectively. When using a fixed RGBW CFA, the learning rate for the decoder network is set to 2×10^{-4}.

4.2. Evaluation of our joint learning approach

Chakrabarti's joint learning approach [6] is the most reasonable competitor. Due to it is limited to image patches for training and testing, it reconstructs a full-size image in a patch-by-patch manner. Since we adopt the same experimental setup as in [6], we directly utilize their pretrained models for testing.[2] We use the tailored demosaicking method RI [31] for the Bayer CFA [3], and employ the guided filter-based demosaicking method [2] for the Kodak CFA [5].

The average CPSNR, SSIM, and S-CIELAB values for different noise levels are shown in Table 1. The best values are in boldface. In terms of S-CIELAB, we can see that the Bayer CFA demosaicked by RI outperforms the other approaches in the noise-free case. In the remaining scenarios, our joint learning approach achieves the best performance. Note that the values of SSIM and S-CIELAB begin to differ after two or three decimal places, so we will not report their values in subsequent part of the paper.

The visual quality is typically the most important evaluation for image demosaicking. We present parts of the visual comparison in Fig. 6. It can be observed that the visual quality reconstructed by our approach is superior to that of other approaches, particularly in terms of color fidelity and texture detail recovery. Chakrabarti's approach [6] suffers from block artifacts due to patch-by-patch reconstruction.

Table 2
Comparison of our decoder network against other image restoration networks with the Kodak CFA [5]. The average CPSNR values for different noise levels are given for images normalized in [0,1]. Note that "STD = 0" means the noise-free case.

Method	Parameters	STD = 0	STD = 0.01	STD = 0.015
NAFNet [35]	51.38M	59.08	53.73	52.22
Uformer [36]	50.88M	59.45	54.15	52.73
RDNet [34]	40.65M	59.37	54.02	52.58
UNet [24]	31.03M	59.21	53.56	52.35
RIDNet [37]	25.36M	58.09	52.74	51.42
LAN [41]	24.39M	59.32	53.95	52.46
DeamNet [38]	24.12M	58.37	53.02	51.96
ADFNet [40]	23.56M	58.79	53.44	52.08
MSANet [39]	22.36M	58.00	52.65	51.28
Ours	23.25M	**60.26**	**54.91**	**53.61**

4.3. Evaluation of the decoder network

Without access to the source codes of current RGBW demosaicking networks [8], we are unable to compare with them. Since our encoder network involves an interpolation process, there is no need to handle missing pixels. Therefore, we can compare our decoder network with the latest networks for image restoration [32,33]. For example, RDNet [34], NAFNet [35], and Uformer [36] are advanced image restoration networks. RIDNet [37], DeamNet [38], MSANet [39], ADFNet [40], and LAN [41] are the SOTA image denoising networks. UNet [24] is a common image restoration network. So we also include it for comparison. Since the Kodak CFA [5] is the most widely used RGBW CFA, we use it as the predefined CFA. Then we evaluate the performance of our decoder network by comparing with aforementioned image restoration networks. We directly use the source codes or executable codes provided by the respective authors, and then train them on our dataset.

The average CPSNR values are shown in Table 2. The best values are in boldface. We can see that our decoder network achieves the best reconstruction performance in both noise-free and noisy conditions, with a relatively small number of parameters. Although our encoder network is not specifically designed for denoising, it demonstrates superior denoising capabilities, surpassing even advanced denoising networks [37,40,41]. This demonstrates that leveraging the characteristics of RGBW CFAs is beneficial for creating lightweight and high-performing network architectures.

4.4. Evaluation of the learned CFAs

With our decoder network, we evaluate the effectiveness of our learned CFA pattern and the existing RGBW CFA patterns. Table 3 shows the average CPSNR values at noise levels with STDs ranging from 0 to 0.04 for images normalized in [0,1]. The best values are in boldface. We train our CFA pattern at the noise level of STD = 0.01. However, we can see that it achieves the highest CPSNR values across

[2] https://github.com/ayanc/learncfa.

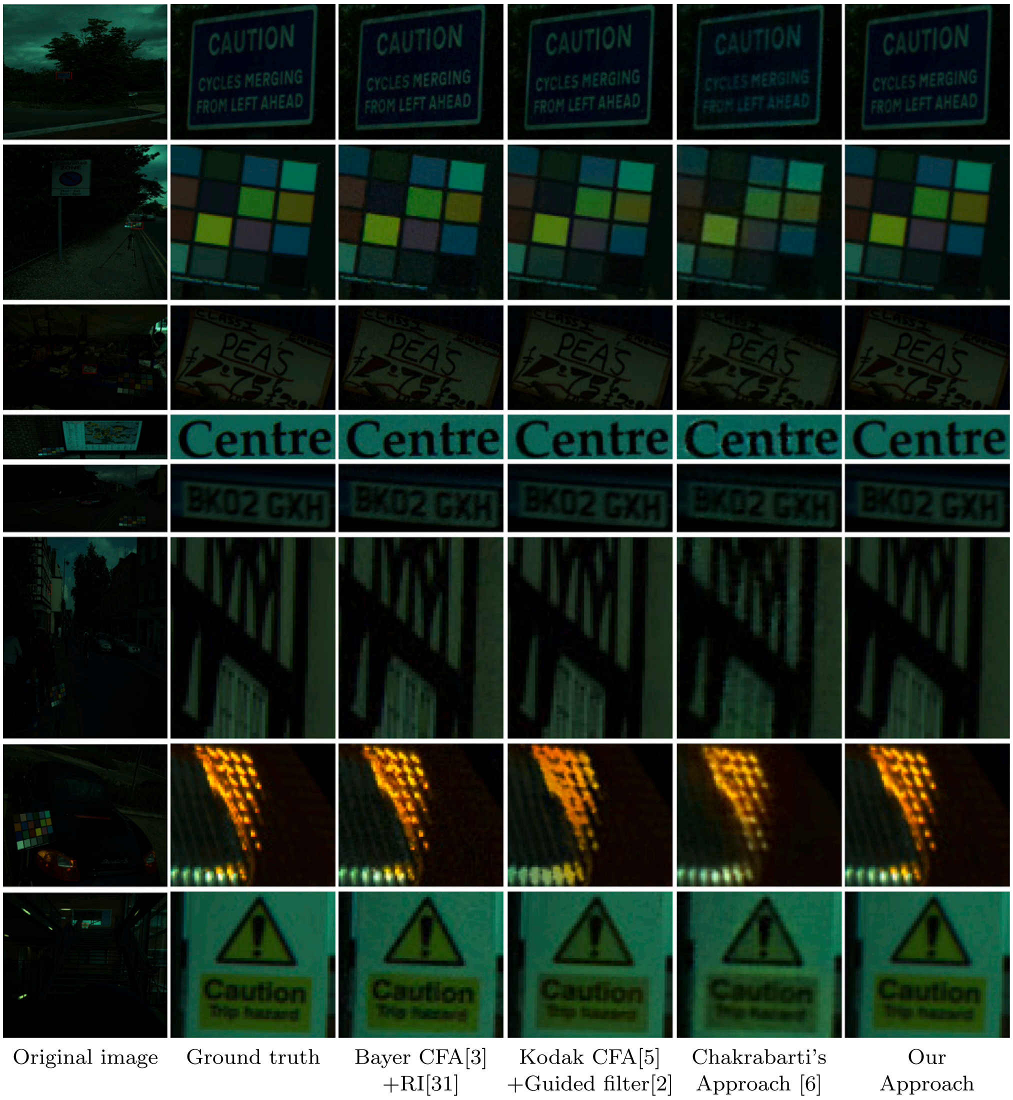

Fig. 6. Blowups of some demosaicked images by different approaches in Table 1 at a noise level of STD=0.01 for images normalized in [0,1]. The demosaicked images produced by our joint learning approach have better visual quality, while the texture details and colors of other images show some distortions.

Table 3
Comparison of different RGBW CFA patterns with various noise levels using our decoder network for images normalized in [0,1].

CFA pattern	STD = 0	STD = 0.005	STD = 0.01	STD = 0.015	STD = 0.02	STD = 0.03	STD = 0.04
CFZ CFA [4]	49.19	47.86	44.21	43.14	42.86	42.19	41.56
Kaizu CFA [13]	54.89	50.86	48.11	47.53	46.86	46.13	45.28
Wang CFA [42]	54.99	53.85	52.92	51.48	50.68	49.23	48.43
Chakrabarti CFA [6]	57.73	55.08	54.09	53.23	51.75	49.86	49.02
Gindele CFA [43]	59.60	55.89	54.32	53.22	51.86	50.06	49.23
Hamilton CFA [11]	59.78	56.26	54.31	53.08	52.06	50.78	49.83
Honda CFA [10]	59.92	56.45	54.72	53.73	52.52	51.26	50.31
Luo CFA [44]	60.11	56.45	54.82	53.58	52.63	51.28	50.36
Kodak CFA [5]	60.26	56.57	54.91	53.61	52.67	51.36	50.43
Our CFA	**60.40**	**56.83**	**55.33**	**53.99**	**53.12**	**52.23**	**51.29**

a wide range of noise levels. At different noise levels, our learned CFA pattern consistently outperforms other RGBW CFA patterns, validating the superiority of our learned CFA pattern. The Kodak CFA [5] can also produce very competitive performance. We present parts of the visual comparison with the Kodak CFA in Fig. 7. We can see that our learned CFA has advantages in noise reduction.

Note that while our approach can learn the optimal RGBW CFA pattern for any size, experiments show that different CFA pattern sizes have impacts on image reconstruction performance. We investigate the size of CFA patterns. We test the CFA pattern sizes of $3\times3, 4\times4, 5\times5, 6\times6, 7\times7$, and 8×8, and simultaneously train the RGBW CFA pattern and the demosaicking method. By evaluating the average CPSNR values of the demosaicked images, we obtain that the optimal CFA pattern size is 4×4. So we use the 4×4 CFA pattern in all experiments.

4.5. Ablation studies

4.5.1. Influence of surrogate functions

To validate the effectiveness of our surrogate function, we include different surrogate functions for comparison. Table 4 provides the

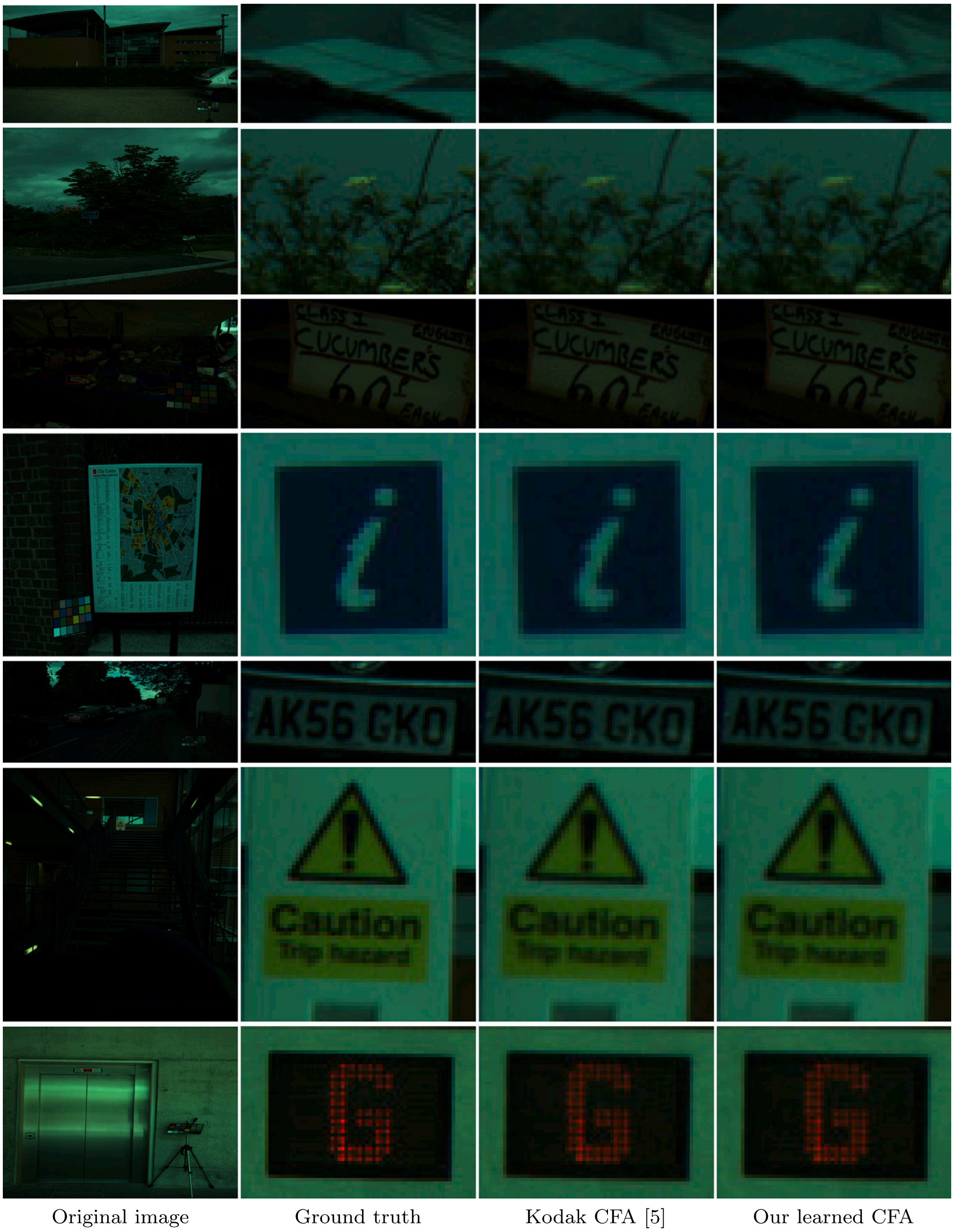

Fig. 7. Blowups of some demosaicked images with the kodak [5] and our learned CFAs using our decoder network at a noise level of STD=0.01 for images normalized in [0,1]. It can be seen that our learned CFA performs better in terms of noise removal.

Table 4
Ablation on different surrogate functions at a moderate noise level of STD = 0.01.

Surrogate function	CPSNR
Linear function	54.19
Hard-sigmoid function	54.38
α softmax function(α = 0.5)	54.63
α softmax function(α = 0.8)	54.71
α softmax function(α = 1.3)	**55.28**
α softmax function(α = 1.5)	54.82
α softmax function(α = 2.0)	54.52

average CPSNR using different surrogate functions at a moderate noise level of STD = 0.01. The best values are in boldface. The “Linear function” represents the linear function, i.e., $\phi(x) = x$. The “Hard-sigmoid function” denotes the truncation function, i.e., $\phi(x) = \text{clip}((x+1)/2, 0, 1)$. The “$\alpha$ softmax function” refers to the softmax function with a temperature coefficient α in Eq. (4), where α can take different values. As α increases, the α softmax function is closer to channel selection, and updating CFA pattern becomes more difficult. From Table 4, we can see that the best performance can be achieved using the softmax function with a temperature coefficient. The value of α affects the updating of the CFA pattern and the quality of demosaicked images. In this paper, we use $\alpha = 1.3$ for all experiments.

4.5.2. *Influence of components of decoder network*

To evaluate the effectiveness of the dual-branch fusion module and SIMUNet in our decoder network, we conduct an ablation study at the noise level of STD = 0.01. Table 5 presents the average CPSNR of the decoder network with different components. The best values are in boldface. “Single branch SIMUNet” refers to a single-branch decoder

Table 5
Ablation on the components of our decoder network at a moderate noise level of STD = 0.01.

Method	CPSNR
Single branch SIMUNet	54.32
SIMUNet→UNet	54.30
SIMUNet→RLFUNet	54.56
Ours	**55.20**

network obtained by removing the dual-branch fusion module from our decoder network. "SIMUNet → UNet" represents replacing SIMUNet with UNet in our decoder network. "SIMUNet → RLFUNet" denotes replacing SIMBlock with RLFBlock in our decoder network. "Ours" represents our decoder network. It is evident that removing the dual-branch fusion module reduces CPSNR by 0.88 dB. Replacing SIMUNet with UNet and RLFUNet reduces the CPSNR by 0.90 dB and 0.64 dB, respectively. SIMUNet helps to eliminate noise and restore texture, demonstrating the superiority of our decoder network components.

4.6. Discussion

Our approach has two notable limitations. First, we use of a simulated dataset. Improving demosaicking performance can be algorithm-centric or data-centric. Although our approach belongs to algorithm-centric demosaicking, there is a simulated-to-real gap. Collecting pairwise real training images for RGBW CFAs is more challenging than the Bayer CFA. For the 2×2 Bayer CFA, we can use downsampling or pixel shift technique [45,46] to obtain ground-truth RGB images. However, RGBW CFAs are usually larger than 4×4. There will be serious aliasing artifacts in downsampled images, and pixel shift technique is limited to (quasi) static scenes and certain advanced cameras with the Bayer CFA. As a result, we need more realistic simulations for color imaging with RGBW CFAs, which include accurate spectral response relationships between W and RGB channels, sophisticated noise modeling [47], crosstalk, point-spread function (PSF), and overexposure [48]. Second, our use of local 3×3 convolutions in the encoder network may explain why the optimal CFA is 4×4 rather than a larger size, highlighting the importance of global receptive fields.

Our approach can be further explored in three aspects. First, introducing appropriate priors (e.g., symmetry) may be helpful to make efficient use of training images and mitigate overfitting. Currently, we impose no constraints on the CFA pattern, which is learned entirely with training images. Second, the design of RGBW CFAs requires a trade-off between color fidelity and SNR, which depends on the application. Surveillance cameras prioritize low-light sensitivity, typically resulting in a higher W pixel ratio, whereas mobile phone cameras emphasize color accuracy, often using a lower W pixel ratio. In joint learning approaches, the learned CFA is mainly influenced by noise levels in training images and the reconstruction network. Noisy training images tend to produce higher W pixel ratios, and the network's ability to exploit both local and global features affects the final learned CFA. Third, data selection is crucial for demosaicking, and it is noteworthy that automatic data selection can be integrated into CFA learning process.

5. Conclusion

In this paper, we have presented a new approach for jointly learning the RGBW CFA and demosaicking, using an autoencoder model. The approach includes an encoder network and a decoder network, allowing for end-to-end learning of CFA design and demosaicking. It has the capability to learn CFA patterns of any size and reconstruct mosaicked images of any size. Extensive experiments on both noise-free and noisy images have shown the effectiveness of our approach.

Our results indicate that jointly learning an one-hot encoded sampling pattern and a reconstruction network is both feasible and beneficial. It can also be applied to learn other one-hot encoded sampling patterns, such as multispectral filter arrays, RGB+NIR CFAs, and polarization filter arrays.

CRediT authorship contribution statement

Chenyan Bai: Writing – review & editing, Writing – original draft, Supervision, Methodology, Funding acquisition. **Faqi Liu:** Writing – original draft, Validation, Software, Methodology, Data curation. **Jia Li:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data.

Acknowledgments

The authors would like to thank the Associate Editor and reviewers for their constructive comments and suggestions. J. Li is supported by NSF of China (grant nos. 62131003 and 62102034). C. Bai is supported by the science and technology plan of Beijing Education Commission (grant no. KM202010028009).

References

[1] R. Lukac, K.N. Plataniotis, Universal demosaicking for imaging pipelines with an RGB color filter array, Pattern Recognit. 38 (11) (2005) 2208–2212.

[2] J. Li, C. Bai, H. Huang, Universal demosaicking for interpolation-friendly RGBW color filter arrays, IEEE Trans. Image Process. 32 (2023) 3592–3605.

[3] B. Bayer, Color imaging array, 1976, United States Patent, no. 3971065.

[4] A. Chakrabarti, W.T. Freeman, T. Zickler, Rethinking color cameras, in: Proceedings of the IEEE International Conference on Computational Photography, IEEE, 2014, pp. 1–8.

[5] J.T. Compton, J.F. Hamilton Jr., Image sensor with improved light sensitivity, 2012, U.S. Patent 20 070 024 931 A1.

[6] A. Chakrabarti, Learning sensor multiplexing design through back-propagation, Adv. Neural Inf. Process. Syst. 29 (2016) 3081–3089.

[7] J. Li, C. Bai, Z. Lin, J. Yu, Automatic design of high-sensitivity color filter arrays with panchromatic pixels, IEEE Trans. Image Process. 26 (2) (2017) 870–883.

[8] S. Sharif, Y.J. Jung, Deep color reconstruction for a sparse color sensor, Opt. Express 27 (17) (2019) 23661–23681.

[9] B. Henz, E.S. Gastal, M.M. Oliveira, Deep joint design of color filter arrays and demosaicing, Comput. Graph. Forum 37 (2) (2018) 389–399.

[10] H. Honda, Y. Iida, Y. Egawa, H. Seki, A color CMOS imager with 4x4 white-RGB color filter array for increased low-illumination signal-to-noise ratio, IEEE Trans. Electron Devices 56 (11) (2009) 2398–2402.

[11] J.F. Hamilton Jr., J.T. Compton, Processing color and panchromatic pixels, 2012, US Patent 8, 274, 715.

[12] P. Hao, Y. Li, Z. Lin, E. Dubois, A geometric method for optimal design of color filter arrays, IEEE Trans. Image Process. 20 (3) (2011) 709–722.

[13] S. Kaizu, Image processing apparatus, imaging device, image processing method, and program for reducing noise or false colors in an image, 2017, US Patent 9, 699, 429.

[14] C. Kwan, J. Larkin, B. Ayhan, Demosaicing of CFA 3.0 with applications to low lighting images, Sensors 20 (12) (2020) 3423.

[15] C. Bai, J. Li, Convolutional sparse coding for demosaicking with panchromatic pixels, Signal Process., Image Commun. 77 (2019) 20–27.

[16] M. Gharbi, G. Chaurasia, S. Paris, F. Durand, Deep joint demosaicking and denoising, ACM Trans. Graph. 35 (6) (2016) 1–12.

[17] S. Sun, L. Chen, G. Slabaugh, P. Torr, Learning to sample the most useful training patches from images, 2020, arXiv preprint arXiv:2011.12097.

[18] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the International Conference for Learning Representations, 2014.

[19] K. Cui, Z. Jin, E. Steinbach, Color image demosaicking using a 3-stage convolutional neural network structure, in: Proceedings of the IEEE International Conference on Image Processing, IEEE, 2018, pp. 2177–2181.
[20] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, N. Sebe, Binary neural networks: A survey, Pattern Recognit. 105 (2020) 107281.
[21] T. Tieleman, G. Hinton, et al., Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, Coursera: Neural Netw. Mach. Learn. 4 (2) (2012) 26–31.
[22] C.M. Nguyen, J.N. Martel, G. Wetzstein, Learning spatially varying pixel exposures for motion deblurring, in: Proceedings of the IEEE International Conference on Computational Photography, 2022, pp. 1–11.
[23] F. Kong, M. Li, S. Liu, D. Liu, J. He, Y. Bai, F. Chen, L. Fu, Residual local feature network for efficient super-resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 766–776.
[24] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.
[25] Z. Hui, X. Gao, Y. Yang, X. Wang, Lightweight image super-resolution with information multi-distillation network, in: Proceedings of the ACM International Conference on Multimedia, 2019, pp. 2024–2032.
[26] F. Kong, M. Li, S. Liu, D. Liu, J. He, Y. Bai, F. Chen, L. Fu, Residual local feature network for efficient super-resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2022, pp. 766–776.
[27] K. Zhang, S. Gu, R. Timofte, Z. Hui, X. Wang, X. Gao, D. Xiong, S. Liu, R. Gang, N. Nan, et al., AIM 2019 challenge on constrained super-resolution: Methods and results, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop, 2019, pp. 3565–3574.
[28] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 136–144.
[29] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[30] X. Zhang, B.A. Wandell, A spatial extension of CIELAB for digital color-image reproduction, J. Soc. Inf. Disp. 5 (1) (1997) 61–63.
[31] D. Kiku, Y. Monno, M. Tanaka, M. Okutomi, Beyond color difference: Residual interpolation for color image demosaicking, IEEE Trans. Image Process. 25 (3) (2016) 1288–1300.
[32] S. Zhai, C. Ren, Z. Wang, X. He, L. Qing, An effective deep network using target vector update modules for image restoration, Pattern Recognit. 122 (2022) 108333.
[33] W. Wu, S. Liu, Y. Xia, Y. Zhang, Dual residual attention network for image denoising, Pattern Recognit. 149 (2024) 110291.
[34] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image restoration, IEEE Trans. Pattern Anal. Mach. Intell. 43 (7) (2020) 2480–2495.
[35] L. Chen, X. Chu, X. Zhang, J. Sun, Simple baselines for image restoration, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 17–33.
[36] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, H. Li, Uformer: A general U-shaped transformer for image restoration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 17683–17693.
[37] S. Anwar, N. Barnes, Real image denoising with feature attention, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3155–3164.
[38] C. Ren, X. He, C. Wang, Z. Zhao, Adaptive consistency prior based deep network for image denoising, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8596–8606.
[39] Y. Gou, P. Hu, J. Lv, J.T. Zhou, X. Peng, Multi-scale adaptive network for single image denoising, Adv. Neural Inf. Process. Syst. 35 (2022) 14099–14112.
[40] H. Shen, Z.-Q. Zhao, W. Zhang, Adaptive dynamic filtering network for image denoising, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 2227–2235.
[41] J. Li, Z. Zhang, X. Liu, C. Feng, X. Wang, L. Lei, W. Zuo, Spatially adaptive self-supervised learning for real-world image denoising, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 9914–9924.
[42] J. Wang, C. Zhang, P. Hao, New color filter arrays of high light sensitivity and high demosaicking performance, in: Proceedings of the International Conference on Image Processing, IEEE, 2011, pp. 3153–3156.
[43] E.B. Gindele, A.C. Gallagher, Sparsely sampled image sensing device with color and luminance photosites, 2002, US Patent 6, 476, 865.
[44] G. Luo, Color filter array with sparse color sampling crosses for mobile phone image sensors, in: Proceedings of the International Image Sensor Workshop, 2007, pp. 162–165.
[45] G. Qian, Y. Wang, J. Gu, C. Dong, W. Heidrich, B. Ghanem, J.S. Ren, Rethinking learning-based demosaicing, denoising, and super-resolution pipeline, in: Proceedings of the IEEE International Conference on Computational Photography, IEEE, 2022, pp. 1–12.
[46] T. Zhang, Y. Fu, C. Li, Deep spatial adaptive network for real image demosaicing, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 3326–3334.
[47] L. Sun, J. Lin, W. Dong, X. Li, J. Wu, G. Shi, Learning real-world heterogeneous noise models with a benchmark dataset, Pattern Recognit. (2024) 110823.
[48] D. Guo, Y. Cheng, S. Zhuo, T. Sim, Correcting over-exposure in photographs, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 515–521.

Chenyan Bai received the B.E. and M.S. degrees from Hebei University, Baoding, China, in 2008 and 2011, respectively, and the Ph.D. degree from Beijing Jiaotong University, Beijing, China, in 2016, all in computer science. Currently, she is a Lecturer at the College of Information Engineering, Capital Normal University, Beijing, China. Her research interests include image processing and machine learning.

Faqi Liu received the B.S. degree in Financial Big Data from Shandong University of Finance and Economics in 2021. He is currently pursuing the M.S. degree in the College of Information Engineering, Capital Normal University. His research interests include image processing and machine learning.

Jia Li received his Ph.D. degree in computer science from Beijing Jiaotong University in 2017. He was a Postdoctoral Researcher at Peking University from 2018 to 2020. He is currently a Lecturer in the School of Artificial Intelligence, Beijing Normal University. His research interests include machine learning, computer vision, and image processing.