sensors

Article

Noise-Resistant Demosaicing with Deep Image Prior Network and Random RGBW Color Filter Array

Edwin Kurniawan [1], Yunjin Park [2] and Sukho Lee [1,*]

[1] Department of Computer Engineering, Dongseo University, Busan 47011, Korea; edwin.kurniawan.93@gmail.com
[2] Department of Mathematics, Ewha University, Seoul 03760, Korea; yunjjiin@naver.com
[*] Correspondence: petra@gdsu.dongseo.ac.kr; Tel.: +82-51-320-1744

Abstract: In this paper, we propose a deep-image-prior-based demosaicing method for a random RGBW color filter array (CFA). The color reconstruction from the random RGBW CFA is performed by the deep image prior network, which uses only the RGBW CFA image as the training data. To our knowledge, this work is a first attempt to reconstruct the color image with a neural network using only a single RGBW CFA in the training. Due to the White pixels in the RGBW CFA, more light is transmitted through the CFA than in the case with the conventional RGB CFA. As the image sensor can detect more light, the signal-to-noise-ratio (SNR) increases and the proposed demosaicing method can reconstruct the color image with a higher visual quality than other existing demosaicking methods, especially in the presence of noise. We propose a loss function that can train the deep image prior (DIP) network to reconstruct the colors from the White pixels as well as from the red, green, and blue pixels in the RGBW CFA. Apart from using the DIP network, no additional complex reconstruction algorithms are required for the demosaicing. The proposed demosaicing method becomes useful in situations when the noise becomes a major problem, for example, in low light conditions. Experimental results show the validity of the proposed method for joint demosaicing and denoising.

Keywords: color filter array; deep image prior; demosaicing; deep learning

check for updates

Citation: Kurniawan, E.; Park, Y.; Lee, S. Noise Resistant Demosaicing with Deep Image Prior Network and Random RGBW Color Filter Array. *Sensors* **2022**, *22*, 1767. https://doi.org/10.3390/s22051767

Academic Editor: Ludovic Macaire

Received: 20 January 2022
Accepted: 21 February 2022
Published: 24 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Current digital imaging systems often consist of monochromatic image sensors equipped with color filter arrays to capture color information. Color filters are required because the photosensors response to the intensity of the light and cannot distinguish the color information. Therefore, a small filter is coated in front of each pixel so that the small filter filters the light by wavelength range to obtain information about the specific color of light for each pixel. For example, the Bayer CFA pattern [1] passes through only one of the primary colors (Red, Green and Blue) at each pixel, where the ratio of the numbers of the R, G, and B pixels in this pattern is 1:2:1. The other two missing colors have to be reconstructed by demosaicing algorithms [2–11].

However, the problem of low resolution arises with the Bayer CFA because information about two color components is lost for each pixel. Moreover, as much of the light is absorbed by the color filters, the proportion of the light energy to the noise energy decreases, which again results in a decrease of the signal-to-noise-ratio (SNR). Therefore, the reconstructed (demosaiced) color image becomes noisy, especially, in low illumination environments when the light energy is low. The problem of noise intensifies as the resolution of mobile photos in smartphones grows since the small sized sensors receive less light and which causes a loss of detail, resulting in relatively high noise levels. The noise problem deepens as the resolution of mobile photos in smartphones increases, reducing the size of image sensors and the light energy they detect. Therefore, to allow more of the incident light to be detected rather than absorbed, CFAs using secondary colors (Cyan, Yellow, and Magenta) have been proposed in various forms, including CYGM (Cyan, Yellow, Green, Magenta), the CMY (Cyan, Magenta, and Yellow), and the CMYW (Cyan, Magenta, Yellow, and White) CFAs [12,13]. Other demosaicing methods use the RGBW (Red, Green, Blue, and White) CFA, which is a CFA that contains 'White' pixels that allow for the penetration of all color intensities [14,15]. There is a tradeoff between the number of R, G, and B pixels and the number of W pixels in demosaicing methods using the RGBW CFA as a higher number of W pixels results in a larger sensitivity of the W sensor, which makes it more robust to the noise, but this also results in a degradation of the resolution as white pixels do not contain color information. In this paper, we propose a deep image prior (DIP)-based demosaicing method that can reconstruct colors from a random RGBW CFA without the use of any complex reconstruction algorithm other than the training of the DIP with the single CFA image. To train the DIP network, we propose a new loss function that is tailored for the demosaicing of the random RGBW CFA. The loss function differs from existing ones used in DIP-based image restoration problems in the respect that for the white pixels, it imposes a linear constraint that the generated color components should obey instead of specifying the values of the color components. This is the first time that the DIP is used for the demosaicing of the CFA, which contains white pixels.

2. Related Works

2.1. Color Filter Arrays in Digital Imaging Systems

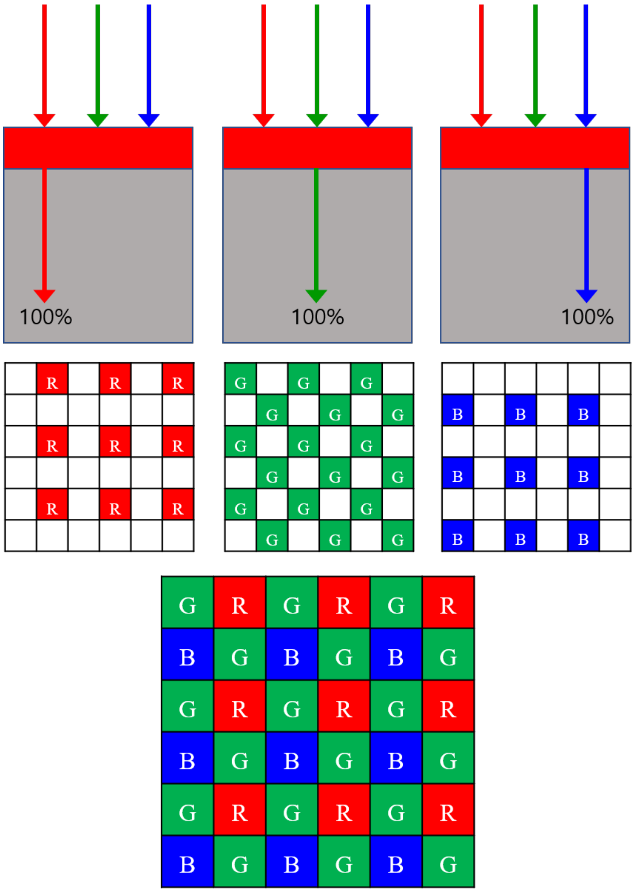
Current digital imaging systems often constitute monochrome image sensors that are overlaid with color filter arrays(CFAs) to capture color information. The most commonly used CFA is the Bayer CFA, which consists of Red, Green, and Blue color filters so that only one color component can be obtained at each pixel location, as shown in Figure 1. Let Ω be the two-dimensional spatial domain of the image, $\mathbf{I}_{orig}[\mathbf{k}] = [R_{orig}[\mathbf{k}], G_{orig}[\mathbf{k}], B_{orig}[\mathbf{k}]]^T$ be the true color image triple at $\mathbf{k}$, and $\mathbf{c}[\mathbf{k}] = [c_R[\mathbf{k}], c_G[\mathbf{k}], c_B[\mathbf{k}]]^T$ be the action of the CFA. Then, the intensity value $I_s[\mathbf{k}]$ of the Red (or Green/Blue) component sensed at the position $\mathbf{k} \in \Omega$ in the Bayer CFA can be expressed as the inner product of $\mathbf{I}_{orig}[\mathbf{k}]$ and $\mathbf{c}[\mathbf{k}]$:

$$I_s[\mathbf{k}] = \mathbf{c}[\mathbf{k}]^T \mathbf{I}_{orig}[\mathbf{k}]. \tag{1}$$

For the Bayer CFA, the components of $\mathbf{c}[\mathbf{k}]$ are defined as

$$\begin{array}{ll} c_R[\mathbf{k}] = 1, c_G[\mathbf{k}] = 0, c_B[\mathbf{k}] = 0 & \text{if } \mathbf{k} \in S_R \\ c_R[\mathbf{k}] = 0, c_G[\mathbf{k}] = 1, c_B[\mathbf{k}] = 0 & \text{if } \mathbf{k} \in S_G \\ c_R[\mathbf{k}] = 0, c_G[\mathbf{k}] = 0, c_B[\mathbf{k}] = 1 & \text{if } \mathbf{k} \in S_B \end{array} \tag{2}$$

where S_R, S_G, and S_B denote the sets of the R, G, and B pixels, respectively. The problem of demosaicing is to restore $\mathbf{I}_{orig}$ from I_s, which is an ill-posed problem since $\mathbf{I}_{orig}$ is a three-channel image, while the sensed image I_s is a one-channel monochrome image. Therefore, additional constraints have to be imposed to solve the demosaicing problem, such as applying an additional assumption that adjacent pixels have similar colors. By applying this assumption, in conventional demosaicing methods, the missing two color components are interpolated from the spatially adjacent CFA data.



G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G

Figure 1. A 6 × 6 partial cut of the Bayer color filter array. R: Red pixels, G: Green pixels, B: Blue pixels.

2.2. *RGBW Color Filter Array*

In [16], the proportions of the R, G, and B components in the white pixel are computed based on the assumption that a linear relationship exists between the white component and the three primary color components (red, green, and blue). Based on this assumption, the components in RGBW CFA matrices, which contain R, G, B, and W (white) pixels, can be defined as:

$$\begin{array}{rl} c_R[\mathbf{k}] = 1, c_G[\mathbf{k}] = 0, c_B[\mathbf{k}] = 0 & \text{if } \mathbf{k} \in S_R \\ c_R[\mathbf{k}] = 0, c_G[\mathbf{k}] = 1, c_B[\mathbf{k}] = 0 & \text{if } \mathbf{k} \in S_G \\ c_R[\mathbf{k}] = 0, c_G[\mathbf{k}] = 0, c_B[\mathbf{k}] = 1 & \text{if } \mathbf{k} \in S_B \\ c_R[\mathbf{k}] = \alpha_R, c_G[\mathbf{k}] = \alpha_G, c_B[\mathbf{k}] = \alpha_B & \text{if } \mathbf{k} \in S_W \end{array} \tag{3}$$

where S_W denotes the set of the white pixels, and α_R, α_G, and α_B are the proportions of the R, G, and B components, respectively, in the white pixel. The values for α_R, α_G, and α_B differ for different sensors. For example, in [16], the authors calculated α_R, α_G, and α_B by optimization to obtain $\alpha_R = 0.2936$, $\alpha_G = 0.4905$, and $\alpha_B = 0.2159$ using a typical RGBW sensor, i.e., the VEML6040 sensor developed by the Vishay company [17]. In this paper, we use the same values, α_R, α_G, and α_B, as calculated in [16] but emphasize the fact that the proposed method works with any values of α_R, α_G, and α_B.

It can be seen from (3) that the demosaicing problem can no longer be solved by mere interpolation as with the Bayer CFA because the intensity values sensed at the W pixels become linear combinations of the R, G, and B color components. Therefore, as a set of linear equations have to be solved, more complex demosaicing methods have to be applied, such as those in [12,13] in conventional non-neural network approaches. Meanwhile, neural network approaches for demosaicing use an end-to-end approach where the input is the mosaiced image and the output the ground-truth image. Therefore, the loss function is simple where the output of the network is to be compared with the ground-truth image. For example, the work of [18] uses the loss function defined as

$$\mathcal{L} = \sum_i \|N_\theta(\mathbf{I}_i^{in}) - \mathbf{I}_i^t\|^2, \tag{4}$$

where $N_\theta(\cdot)$ denotes the neural network, and $\mathbf{I}_i^{in}$ and $\mathbf{I}_i^t$ are the i's mosaiced and ground-truth images, respectively. Other tasks, such as the work of [19,20] we used in the comparison in the experiments, divide the entire image into smaller patches but apply the same loss function as shown in (4) to all the patches:

$$\mathcal{L} = \sum_i \sum_{p_i^k} \|N_\theta(\mathbf{I}_{p_i^k}^{in}) - \mathbf{I}_{p_i^k}^t\|^2, \tag{5}$$

However, conventional neural network approaches require a large dataset of mosaic/ground-truth images for the training of the neural network, and the training process usually takes several days. In comparison, we propose the use of a deep image prior (DIP) network that can be trained on the single sensed CFA image and can simultaneously obtain the color components for the R, G, B, and W pixels based on a loss function tailored for this problem.

2.3. Deep-Image-Prior-Based Image Restoration

Recently, Ulyanov et al. proposed the use of the DIP for image restoration [21]. The DIP resembles an auto encoder but is trained with a single image $\mathbf{I}$, i.e., the image to be restored. The DIP converts a noise tensor $\mathbf{z}$ derived from a uniform distribution into a restored image $\mathbf{g}_\theta(\mathbf{z})$, which is the output of the deep image prior network with parameters θ and input $\mathbf{z}$. In [21], the DIP is applied to the task of inpainting by minimizing the following loss function:

$$\mathcal{L} = \|(\mathbf{g}_\theta(\mathbf{z}) - \mathbf{I}) \odot \mathbf{m}\|^2, \tag{6}$$

where $\odot$ is the Hadamard's product(element-wise multiplication operator) and $\mathbf{m} \in \{0,1\}^{H \times W}$ is a binary mask with a value equal to zero corresponding to the missing pixels to be inpainted and a value of one corresponding to the pixels to be preserved. It has been shown in [21] that the minimization of $\mathcal{L}$ in (6) with respect to θ results in a DIP network that can inpaint an image. Inpainting and demosaicing are similar in that they fill in the missing information but differ in the fact that in inpainting the existing pixels have full channel information while in demosaicing they have not. That is, while all R, G, and B values are available for existing pixels in inpainting, in demosaicing, existing pixels have only one of the R, G, and B values. In [22], Park et al. use the variational DIP for the joint task of demosaicing and denoising. However, so far the DIP has never been used for the demosaicing of CFA images which contain white pixels.

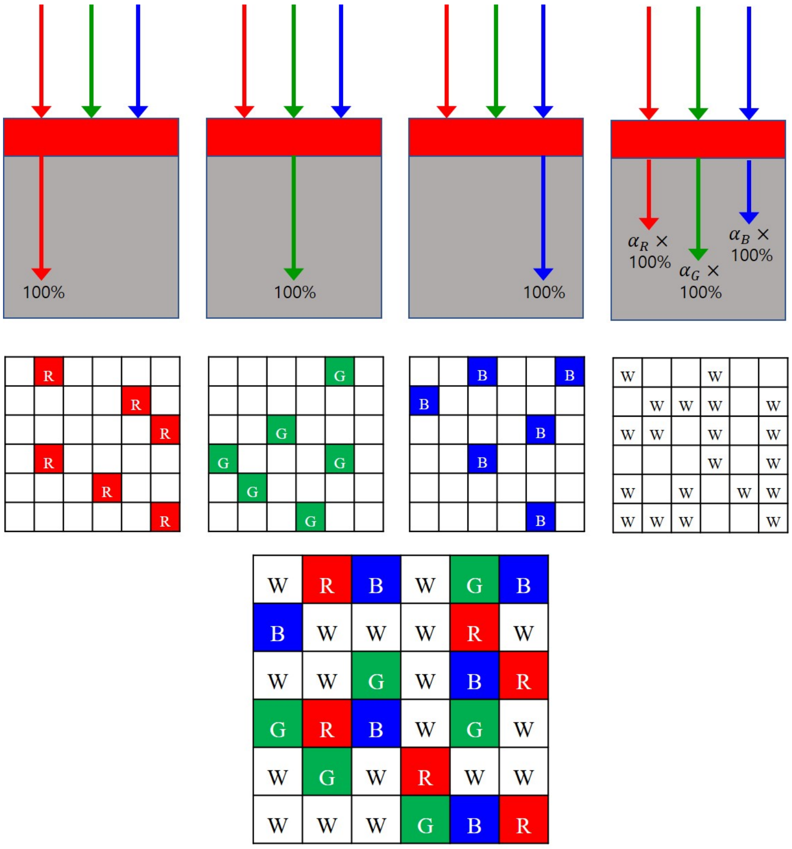
3. Proposed Method

In this section, we explain in detail the proposed method. We first introduce the random RGBW color filter array. Then, we propose a method how to reconstruct the colors from the monochrome CFA image sensed with the random RGBW CFA.

3.1. Random RGBW Color Filter Array

In neural network-based demosaicing methods, it is the CFA that determines the parameters of the neural network because the CFA defines the loss function. A different CFA results in a different loss function, which means that the resulting parameters of the network become different for different CFAs. The reason that we come up with a random pattern is based on the guess that teaching a network to generate random colors at random locations would make it easy to find the most meaningful parameters for the backpropagation because there is no bias for specific colors at certain locations. Furthermore, the reason that we used equal numbers of R, G, and B pixels instead of the widely used ratio of 1:2:1 for R, G, and B pixels is that in the reconstructed image, the color seed should serve as a reference point for the R, G, and B colors of pixels around the color seed to revive. When the number of R, G, and B color seeds is unbalanced, then the color will revive itself disproportionately, as we will show in the experimental section.

Figure 2 shows a 6 × 6 partial cut of the proposed random RGBW CFA pattern. Fifty percent of the whole RGBW CFA consists of white pixels, and the remaining 50% is divided equally between R, G and B pixels. The value of the sensed intensity value $I_s[\mathbf{k}]$ contains, according to (1), components of $c_R[\mathbf{k}], c_G[\mathbf{k}], c_B[\mathbf{k}]$ defined as in (3). The components of $c_R[\mathbf{k}], c_G[\mathbf{k}], c_B[\mathbf{k}]$ are constrained to lie in [0, 1] for physical feasibility since they correspond to the opacity rates. It is known that the white pixels let more of the light energy through than the R, G, and B pixels. As more light energy is detected at the white pixels, the signal-to-noise ratio is larger than at the R, G, and B pixels. This makes the random RGBW CFA more robust against the noise.



	R				
				R	
					R
	R				
			R		
					R

				G	
		G			
G				G	
	G				
			G		

		B			B
B					
				B	
		B			
				B	

W			W		
	W	W	W		W
W	W		W		W
			W		W
W		W		W	W
W	W	W			W

W	R	B	W	G	B
B	W	W	W	R	W
W	W	G	W	B	R
G	R	B	W	G	W
W	G	W	R	W	W
W	W	W	G	B	R

Figure 2. Proposed random RGBW CFA. R: Red pixels, G: Green pixels, B: Blue pixels, W: White pixels.

3.2. DIP-Based Demosaicing of the Random RGBW-CFA

In this section, we propose a DIP-based demosaicing method for the demosaicing of the random RGBW CFA. We denote by $\mathbf{g}_\theta(\mathbf{z})$ the three-dimensional output of the DIP with network parameters θ and $\mathbf{z}$ as the input. The size of $\mathbf{g}_\theta(\mathbf{z})$ is $H \times W \times 3$, where H and W are the height and the width of the demosaiced image, respectively, and the number of channels is 3, corresponding to the R, G, and B channels. At every pixel $\mathbf{k} \in \Omega$, where Ω is the two-dimensional spatial domain of $\mathbf{g}_\theta(\mathbf{z})$, the color is defined by an RGB triplet $\mathbf{g}_\theta(\mathbf{z})[\mathbf{k}]$, which is a three-element vector that specifies the intensities of the Red, Green, and Blue components of the color. For the purpose of simplification, we will denote $\mathbf{g}_\theta(\mathbf{z})[\mathbf{k}]$ as $\mathbf{g}_\theta[\mathbf{k}]$. Thus,

$$\mathbf{g}_\theta(\mathbf{z})[\mathbf{k}] = \mathbf{g}_\theta[\mathbf{k}] = \left[g_\theta^R[\mathbf{k}], g_\theta^G[\mathbf{k}], g_\theta^B[\mathbf{k}]\right]^T, \tag{7}$$

where $g_\theta^R[\mathbf{k}]$, $g_\theta^G[\mathbf{k}]$, and $g_\theta^B[\mathbf{k}]$ are the Red, Green, and Blue components of the three-element vector $\mathbf{g}_\theta(\mathbf{z})[\mathbf{k}]$. The loss function used in the training of the DIP is

$$\mathcal{L} = \sum_{\mathbf{k}\in\Omega} \|\mathbf{c}[\mathbf{k}]^T \mathbf{g}_\theta[\mathbf{k}] - I_s[\mathbf{k}]\|^2 \tag{8}$$

where the components of $\mathbf{c}[\mathbf{k}] = [c_R[\mathbf{k}], c_G[\mathbf{k}], c_B[\mathbf{k}]]^T$ are defined as in (3). The minimization of the loss function constrains the output $\mathbf{g}_\theta[\mathbf{k}]$ to obey the physical constraint imposed on the sensed monochrome CFA image, i.e., the constraint that the sensed monochrome pixel is a weighted sum of the R, G, and B components, where the weights are determined by the ratio of opacity of the color filters in the CFA. It should be noticed that the form of the loss function is different from those designed for other image restoration purposes with the DIP. In conventional DIPs, pixel-wise comparisons are made between the intensities of the generated pixels and the data pixels. In comparison, the loss function in (8) does not make a pixel-wise comparison between the intensities but imposes a constraint that the generated pixel values should obey. The reason that the original colors can be restored by minimizing the loss function in (8) lies in the fact that the components of $\mathbf{c}[\mathbf{k}]$ vary for neighboring pixels and that the DIP works as a prior that favors a natural image. If the vectors $\mathbf{c}[\mathbf{k}]$ are the same for all $\mathbf{k}$, numerous images of different colors will be exist that minimize the loss in (8), making it difficult to restore the original colors. However, as the vectors $\mathbf{c}[\mathbf{k}]$ are different for neighboring pixels, there exists a unique solution that satisfies both the constraint of making the loss function in (8) minimum and the image prior constraint imposed by the DIP. However, even though there exists a unique solution, it is not guaranteed that the DIP can find this solution. The R (or G, B) pixels act as anchor points that hold the true R (or G, B) color components of the sensed image. As the other components of $\mathbf{c}[\mathbf{k}]$ at these pixels are zero, the true R color components will be reconstructed using these pixels when minimizing the loss in (8). The other color components will use these reconstructed true color components as clues and will be reconstructed by satisfying the constraints imposed by the loss in (8) and the DIP.

Writing the loss function in (8) in matrix form results in

$$\mathcal{L} = \|\widehat{RGB2Gray}(\mathbf{c} \odot \mathbf{g}_\theta(\mathbf{z})) - I_s\|^2 \tag{9}$$

where $\odot$ denotes the element-wise product and $\widehat{RGB2Gray}(\cdot)$ denotes the operation similar to the RGB to Gray image conversion, i.e., the operation that produces a monochrome image by adding the R, G, and B components. Here, the operation of $\widehat{RGB2Gray}(\cdot)$ is adding the three components in the three channels of $\mathbf{c} \odot \mathbf{g}_\theta(\mathbf{z})$, resulting in a one-channel image. It should be noted that for the R, G, and B pixels, i.e., for $\mathbf{k} \in S_R \cup S_G \cup S_B$ (where $\cup$ denotes the union operator), only one component in $\mathbf{c}[\mathbf{k}] \odot \mathbf{g}_\theta(\mathbf{z})[\mathbf{k}]$ is non-zero as $\mathbf{c}[\mathbf{k}]$ has only one non-zero component. This is why we use the notation $\widehat{RGB2Gray}(\cdot)$ instead of $RGB2Gray(\cdot)$.

However, if the loss function in (9) is minimized directly, the colors in the demosaiced image will fade. This is due to the fact that the minimizing of the loss function in (9) reaches the local minimum without sufficiently reconstructing the color components inherent in the white pixels. To prevent this phenomenon, we split up the loss function in (9) into four terms and apply different weights to them:

$$\mathcal{L} = \lambda_W \mathcal{L}_W + \lambda_R \mathcal{L}_R + \lambda_G \mathcal{L}_G + \lambda_B \mathcal{L}_B. \tag{10}$$

Here, $\mathcal{L}_W$ is related to the white pixels, and $\mathcal{L}_R$, $\mathcal{L}_G$, and $\mathcal{L}_B$ are related to the R, G, and B pixels, respectively, where $\mathcal{L}_W$ is defined as

$$\mathcal{L}_W = \|M_W \odot [\widehat{RGB2Gray}(\mathbf{c} \odot \mathbf{g}_\theta(\mathbf{z})) - I_s]\|^2 \tag{11}$$

and

$$\mathcal{L}_R = \|M_R \odot [\widehat{RGB2Gray}(\mathbf{c} \odot \mathbf{g}_\theta(\mathbf{z})) - I_s]\|^2, \tag{12}$$

$$\mathcal{L}_G = \|M_G \odot [\widehat{RGB2Gray}(\mathbf{c} \odot \mathbf{g}_\theta(\mathbf{z})) - I_s]\|^2, \tag{13}$$

$$\mathcal{L}_B = \|M_B \odot [\widehat{RGB2Gray}(\mathbf{c} \odot \mathbf{g}_\theta(\mathbf{z})) - I_s]\|^2. \tag{14}$$

Here, the mask M_W has values of 1 at the white pixel positions and 0 at the other positions. Similarly, M_R (M_G,M_B) has a value of 1 at the Red (Green, Blue) pixel positions and 0 at the other pixel positions. The sizes of M_W, M_R, M_G, and M_B are all $H \times W$.

If $\lambda_W = \lambda_R = \lambda_G = \lambda_B = 1$, the losses in (9) and (10) are the same. To avoid the color fading artifact, we apply different values of $\lambda_W, \lambda_R, \lambda_G$ and λ_B for different iteration number t:

$$\begin{array}{lllll} \lambda_W = 1, & \lambda_R = 1, & \lambda_G = 0, & \lambda_B = 0 & t < 500 \\ \lambda_W = 1, & \lambda_R = 0, & \lambda_G = 1, & \lambda_B = 0 & 500 \leq t < 1000 \\ \lambda_W = 1, & \lambda_R = 0, & \lambda_G = 0, & \lambda_B = 1 & 1000 \leq t < 1500 \\ \lambda_W = 1.5, & \lambda_R = 0.5, & \lambda_G = 0.5, & \lambda_B = 0.5 & 1500 \leq t \leq 2100 \end{array} \tag{15}$$

For the first 500 iterations, the color image is reconstructed using only the R pixels and the White pixels, i.e., by minimizing only $\lambda_W \mathcal{L}_W + \lambda_R \mathcal{L}_R$ with $\lambda_W = \lambda_R = 1$. By doing so, we expect that the red colors are fully reconstructed at least at the R pixels. After that, we reconstruct the colors for the next 500 iterations according to the sensed G pixels and White pixels, i.e., $\lambda_W \mathcal{L}_W + \lambda_G \mathcal{L}_G$ with $\lambda_W = \lambda_G = 1$, and then, for the next 500 iterations, according to the sensed B pixels and White pixels, i.e., $\lambda_W \mathcal{L}_W + \lambda_B \mathcal{L}_B$ with $\lambda_W = \lambda_B = 1$. At the end of 1500 iterations, the R, G, and B color components are highly saturated at all pixels. We then run another 600 iterations with $\lambda_W = 1.5, \lambda_R = 0.5, \lambda_G = 0.5$, and $\lambda_B = 0.5$. This imposes a strong linear constraint on the white pixels according to the minimization of the loss function in (11) while also trying to maintain the reconstructed colors to some extent. The number of iterations has been determined by experiments.

Figure 3 shows the overall diagram of the proposed method and how the input noise is gradually turning into the demosaiced image as the optimization process progresses according to the minimization of the loss function in (8). We use a noise image $\mathbf{z}$ as the input to the DIP, which is sum of a constant noise ($\mathbf{z}_c$) and a variable noise ($\mathbf{z}_{v(t)}$), as in the variational DIP in [22]. The variable noise $\mathbf{z}_{v(t)}$ is newly generated for each step of the training and is multiplied by 0.1 before being added to $\mathbf{z}_c$:

$$\mathbf{z} = \mathbf{z}_c + 0.1\mathbf{z}_{v(t)}. \tag{16}$$

The constant noise $\mathbf{z}_c$ remains unchanged throughout the training. After the training of the DIP is finished, i.e., at the test time, only the constant noise $\mathbf{z}_c$ is put into the DIP to obtain the final denoised and demosaiced color image. The reason that $\mathbf{z}_{v(t)}$ is added to $\mathbf{z}_c$ in the training is to obtain a denoising effect, as explained in [22], i.e., the extra variable noise $\mathbf{z}_{v(t)}$ prevents the output of the DIP from being noisy.

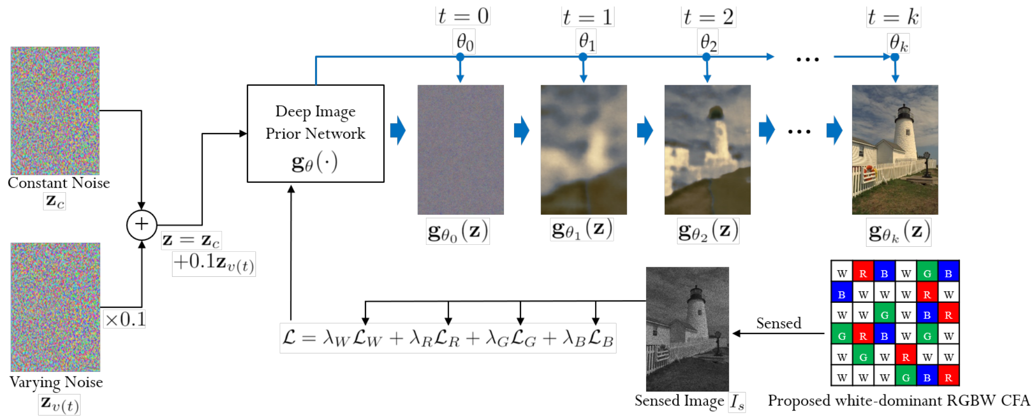

$t = 0$
θ_0
$t = 1$
θ_1
$t = 2$
θ_2
$t = k$
θ_k
Constant Noise
$\mathbf{z}_c$
Deep Image Prior Network
$\mathbf{g}_\theta(\cdot)$
$\mathbf{g}_{\theta_0}(\mathbf{z})$
$\mathbf{g}_{\theta_1}(\mathbf{z})$
$\mathbf{g}_{\theta_2}(\mathbf{z})$
$\mathbf{g}_{\theta_k}(\mathbf{z})$
$\mathbf{z} = \mathbf{z}_c$
$+0.1\mathbf{z}_{v(t)}$
$\times 0.1$
Varying Noise
$\mathbf{z}_{v(t)}$
$\mathcal{L} = \lambda_W \mathcal{L}_W + \lambda_R \mathcal{L}_R + \lambda_G \mathcal{L}_G + \lambda_B \mathcal{L}_B$
Sensed
Sensed Image I_s
W R B W G B
B W W W R W
W W G W B R
G R B W G W
W G W R W W
W W W G B R
Proposed white-dominant RGBW CFA

Figure 3. Overall diagram of the proposed method.

4. Experiments and Discussion

4.1. Experimental Settings

We experimented on the noise-added Kodak [23] dataset, which contains 24 images in bmp image file format with size of 768 × 512, and the McMaster dataset, which is used in the experiments in [24] consisting of 18 color images with resolution of 500 × 500. The noise is derived from a zero-mean Gaussian distribution with different standard deviations for each color channel. We set the standard deviations as 0.0463, 0.0294, 0.0322, and 0.0157 for the R, G, B, and W channels, respectively. The standard deviations are determined according to actual physical measurements of the amount of noises in the R, G, B, and W channels acquired under low light condition. The amount of noise is different for each channel since each color filter absorbs different light energy.

4.2. Network Structure

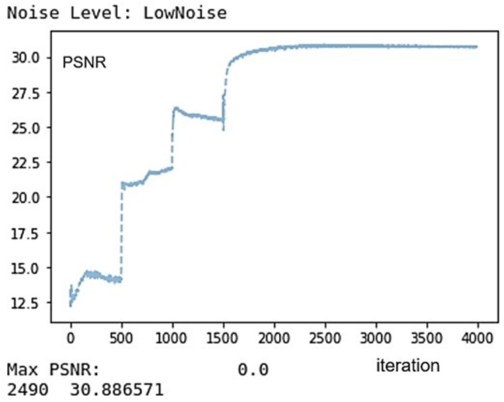
We use the same structure of the DIP network as in [21] for inpainting. The network has an encoder-decoder type structure that consists of five down-sampling and five up-sampling convolutional layers with skip connections, where each convolutional layer consists of 128 feature maps obtained by 3 × 3 convolutions followed by a Leaky ReLU activation. The weights in the DIP are all initialized with a Gaussian noise with standard deviation of 0.03. For the backpropagation optimization, we used the Adam optimizer with a learning rate of 0.001.

4.3. Experimental Results

We compared the proposed method with other joint demosaicing and denoising neural networks such as the Sequential Energy Minimization (SEM) method [18], the DNet [19], and the LCNN model [20]. The SEM method is one of the first data-driven local-filtering methods to use a deep learning based model for joint demosaicing and denoising. The DNet further improves the SEM model by adopting the convolutional neural network in its structure, while the LCNN model is a lightweight convolutional neural network to adopt residue learning and aggregated residual transformations.

These methods use a lot of images for the training of the network and use the conventional Bayer CFA. For the comparison of RGBW-CFA-based demosaicing, we compared with the demosaicing method developed by the Sony corporation [15] for the Sony RGBW CFA, with the Paul's method [14] for the Paul's RGBW CFA. We also compare with the residual interpolation method [6] as a representative of classic RGB demosaicing methods. We made quantitative comparisons in terms of the CPSNR (Color Peak Signal-to-Noise Ratio) the SSIM (Structural Similarity Index Measure), and the FSIMc (Feature Similarity Index for Color images).

Figure 4 shows how the CPSNR value changes in the process of reconstructing the Kodak No. 17 image as the iteration progresses. It can be seen that after iteration 2100, the CPSNR value converges. Therefore, we normally terminate the reconstruction process at iteration 2100. Furthermore, it can be observed that the PSNR value changes significantly at iterations 500, 1000, and 1500 because the loss function changes at this time, as explained in Section 3.2.

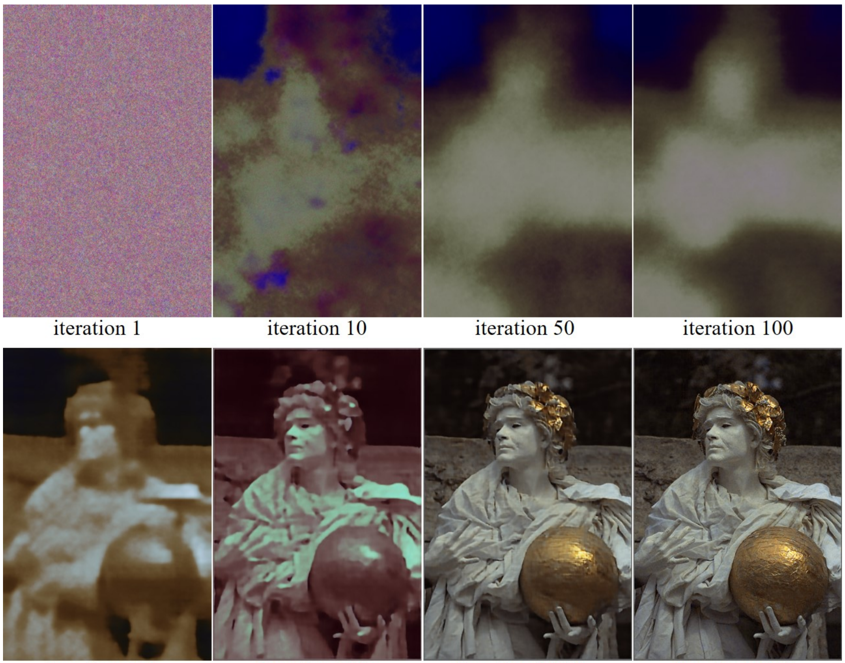


```
Noise Level: LowNoise
```

```
Max PSNR:              0.0
2490  30.886571
```

Figure 4. Showing the CPSNR values in the process of reconstructing the Kodak No. 17 image as the iteration progresses.

Figures 5 and 6 show the reconstruction of the Kodak No. 17 and No. 25 images according to the progress of the iteration, respectively. Again, it can be observed that there is a large change in the colors, before and after 500, 1000, and 1500 iterations, respectively, as the loss function changes at iteration 500, 1000, and 1500, respectively. It can be seen that the all the colors are well reconstructed at iteration 2100.



iteration 1 iteration 10 iteration 50 iteration 100

iteration 400 iteration 900 iteration 1400 iteration 2100

Figure 5. Showing the process of reconstructing the Kodak No. 17 image as the iteration progresses.

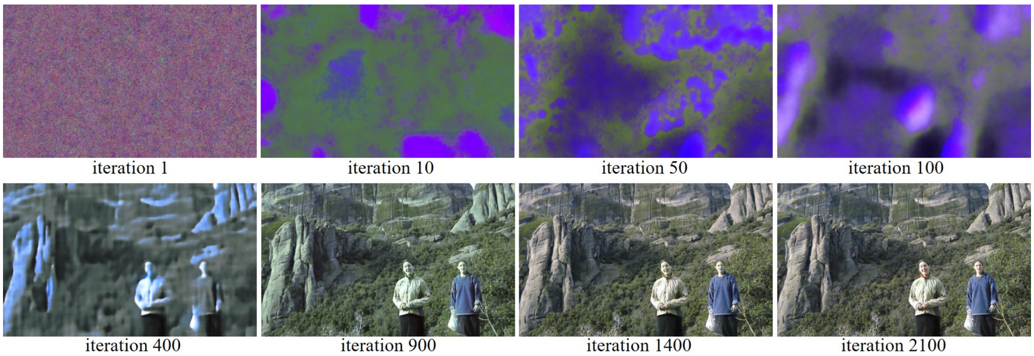


Figure 6. Showing the process of reconstructing the Kodak No. 25 image as the iteration progresses.

To show the effectiveness of using the same ratio R, G, and B pixels as opposed to the 1:2:1 ratio of R, G, and B pixels commonly used in conventional demosaicing methods, we compared, in Figure 3, the reconstruction results with the RGBW CFAs having a ratio 1:2:1:4 of R, G, B, W pixels, and a ratio of 1:1:1:3 of R, G, B, W pixels, respectively. To show the validity of using equal ratio R, G, and B pixels as opposed to the ratio of 1:2:1 of R, G, and B pixels normally used in conventional demosaicing methods, we compared in Figure 3 the reconstruction results with the proposed method on the random RGBW CFA on the Kodak No. 17 and No. 25 images with a ratio of 1:2:1:4 of R, G, B, W pixels, and a ratio of 1:1:1:3 of R, G, B, W pixels, respectively. Due to the larger proportion G pixels in the RGBW CFA with ratio of 1:2:1:4, the reconstructed images appear a bit greenish, as can be observed especially in the bottom image of Figure 7a. Furthermore, due to the smaller number of R and B pixels, some of the colors are not reconstructed, such as the red in the scarf in the bottom image of Figure 7a. However, with the proposed RGBW pattern, all the colors are well reconstructed.

Table 1 summarizes the CPSNR, SSIM, and the FSIMc values of the various demosaicing methods, which are the average values for all the images in the Kodak and the McMaster datasets, respectively. The proposed method shows the highest results for all evaluation indicators. Figures 8 and 9 show the visual comparison on the different methods on the Kodak No.19 image. The fence region in Kodak No.19 image is used a lot in the comparison between different demosaicing methods because it is difficult to reconstruct the fence bars while preventing the aliasing artifact. As can be observed in Figure 9b,c, the RI and the Sony CFAs with their corresponding demosaicking methods show aliasing artifacts (color artifacts) on the fence region. Paul's method and SEM method can deal with aliasing artifacts to some extent but some aliasing artifacts still remain. The DNet and the LCNN methods can deal better with the aliasing artifacts but are not so efficient in eliminating the noise, as can be seen in Figure 9. Due to the smaller noise at white pixels, the effective reconstruction by the proposed loss function, and the inherent denoising property, the proposed method shows good denoising effects as can be seen in Figure 9h. However, one of the drawbacks of the proposed method is that the color fades a little, as can be seen in Figure 9h. There is still room for improvement if other network structures are used for the DIP, which may be one of the further study topics.

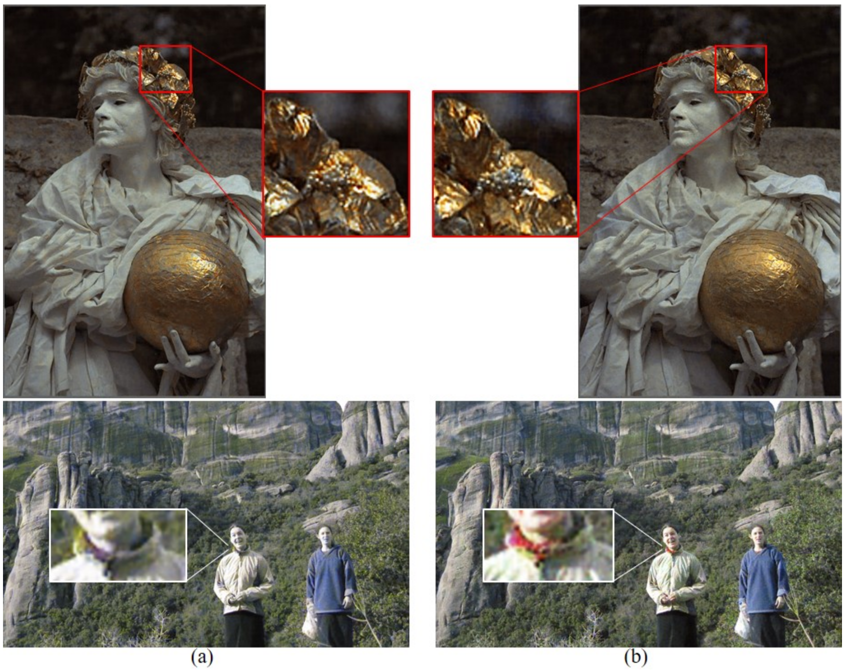


Figure 7. Comparing the reconstructions results with the proposed method on the random RGBW CFA on the Kodak No. 17 and No. 25 images with (**a**) a ratio of 1:2:1:4 of R, G, B, and W pixels (**b**) a ratio of 1:1:1:3 of R, G, B, and W pixels.

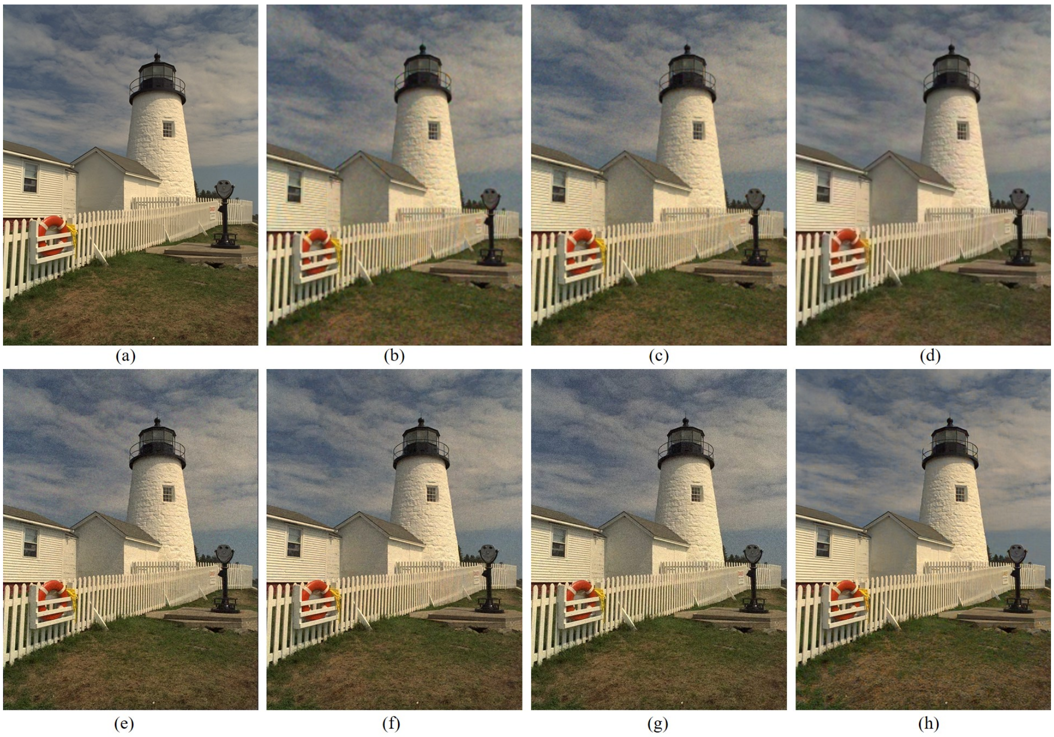


Figure 8. Visual comparison of the details between the results of different demosaicing methods on the Kodak No. 19 image: (**a**) original (**b**) RI [6] (**c**) Sony [15] (**d**) Paul's [14] (**e**) DNet [19] (**f**) SEM [18] (**g**) LCNN [20] (**h**) Proposed.

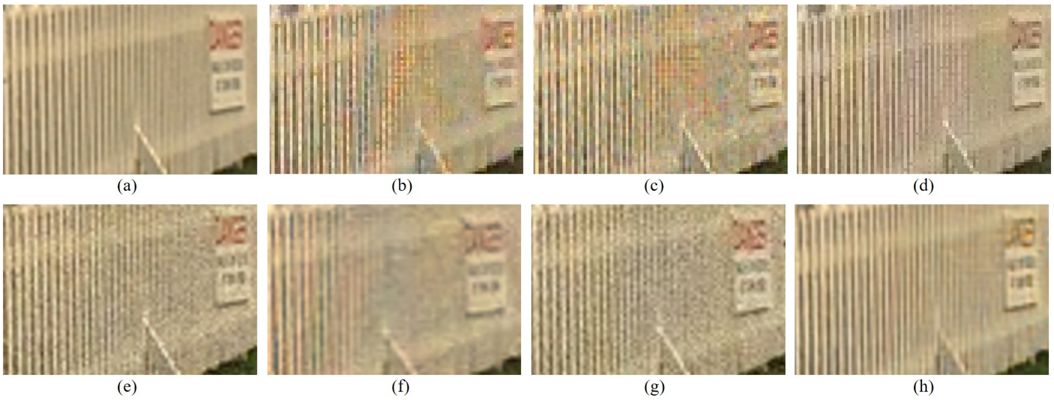


Figure 9. Visual comparison of the details in Figure 8 between the results of different demosaicing methods on the Kodak No. 19 image: (**a**) original (**b**) RI [6] (**c**) Sony [15] (**d**) Paul's [14] (**e**) DNet [19] (**f**) SEM [18] (**g**) LCNN [20] (**h**) Proposed.

Table 1. Comparison of the CPSNR, SSIM, and the FSIMc values between the various demosaicing methods on the Kodak and the McMaster image datasets with Low Noise Level. The largest values are in bold fonts.

Dataset	Method	CPSNR	SSIM	FSIMc
Kodak	SEM [18]	30.39	0.9961	0.9792
	DNet [19]	28.79	0.9540	0.9743
	LCNN [20]	29.00	0.9948	0.9932
	RI [6]	28.69	0.9517	0.9448
	Sony [15]	29.17	0.9615	0.9502
	Paul et al. [14]	30.72	0.9794	0.9566
	Proposed	**30.80**	**0.9964**	**0.9953**
McMaster	SEM [18]	29.26	0.9940	0.9773
	DNet [19]	28.88	0.9579	0.9756
	LCNN [20]	28.94	0.9943	0.9922
	RI [6]	28.59	0.9563	0.9456
	Sony [15]	28.80	0.9641	0.9515
	Paul et al. [14]	27.79	0.9715	0.9571
	Proposed	**31.61**	**0.9963**	**0.9936**

5. Conclusions

In this paper, we proposed a DIP-based joint demosaicing and denoising method tailored for the demosaicing of a white-dominant RGBW color filter array. The demosaicing is performed by training the DIP network with a single noisy color filter array (CFA) image. For this aim, we proposed a loss function different from those used in other DIP-based image restoration applications. We proposed how to reconstruct colors from white pixels that have no explicit color information and showed in the experimental results that the noise in the white pixels as well as in the R, G, and B pixels are well removed by the regression process inherent in the minimization of the loss function. The proposed method can easily be applied to the demosaicing to other color filter arrays.

Author Contributions: Conceptualization, E.K., Y.P., S.L.; methodology, E.K., Y.P.; software, E.K., Y.P., S.L.; validation, E.K., S.L.; formal analysis, E.K., S.L.; investigation, S.L.; resources, E.K., Y.P., S.L.; writing—original draft preparation, E.K., S.L.; writing—review and editing, E.K., S.L.; visualization, Y.P.; supervision, S.L.; project administration, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Dongseo University, "Dongseo Cluster Project Research Fund of 2021" grant number DSU-20210001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This work was supported by Dongseo University, "Dongseo Cluster Project" Research Fund of 2021 (DSU-20210001).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bayer, B. Color Imaging Array. U.S. Patent 3,971,065, 20 July 1976.
2. Alleysson, D.; Susstrunk, S.; Herault, J. Linear demosaicing inspired by the human visual system. *IEEE Trans. Image Process.* **2005**, *14*, 439–449. [CrossRef] [PubMed]
3. Dubois, E. Frequency-domain methods for demosaicking of Bayer-sampled color images. *IEEE Signal Process. Lett.* **2005**, *12*, 847–850. [CrossRef]
4. Gunturk, B.K.; Altunbasak, Y.; Mersereau, R.M. Color plane interpolation using alternating projections. *IEEE Trans. Image Process.* **2002**, *11*, 997–1013. [CrossRef] [PubMed]
5. Gunturk, B.K.; Glotzbach, J.; Altunbasak, Y.; Schafer, R. Demosaicking: Color filter array interpolation. *Signal Process. Mag. IEEE* **2005**, *22*, 44–54. [CrossRef]
6. Kiku, D.; Monno, Y.; Masayuki, T.; Okutomi, M. Beyond Color Difference: Residual Interpolation for Color Image Demosaicking. *IEEE Trans. Image Process.* **2016**, *25*, 1288–1300. [CrossRef] [PubMed]
7. Kimmel, R. Demosaicing: Image reconstruction from color CCD samples. *IEEE Trans. Image Process.* **1999**, *8*, 1221–1228. [CrossRef] [PubMed]
8. Daniele, M.; Giancarlo, C. Color image demosaicking: An overview. *Signal Process. Image Commun.* **2011**, *26*, 518–533. [CrossRef]
9. Pei, S.C.; Tam, I.K. Effective color interpolation in CCD color filter arrays using signal correlation. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 503–513. [CrossRef]
10. Pekkucuksen, I.; Altunbasak, Y. Multiscale Gradients-Based Color Filter Array Interpolation. *IEEE Trans. Image Process.* **2013**, *22*, 157–165. [CrossRef]
11. Lei, Z.; Xiaolin, W. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Trans. Image Process.* **2005**, *14*, 2167–2178. [CrossRef]
12. Lukac, R.; Plataniotis, K.N. Color filter arrays: Design and performance analysis. *IEEE Trans. Consum. Electron.* **2005**, *51*, 1260–1267. [CrossRef]
13. Hirakawa, K.; Wolfe, P.J. Spatio-Spectral Color Filter Array Design for Optimal Image Recovery. *IEEE Trans. Image Process.* **2008**, *17*, 1876–1890. [CrossRef] [PubMed]
14. Oh, P.; Lee, S.; Kang, M.G. Colorization-Based RGB-White Color Interpolation using Color Filter Array with Randomly Sampled Pattern. *Sensors* **2017**, *17*, 1523. [CrossRef] [PubMed]
15. Tachi, M. Image Processing Device, Image Processing Method, and Program Pertaining to Image Correction. U.S. Patent 8,314,863, 20 November 2012.
16. Rafinazari, M.; Dubois, E. Demosaicking algorithms for RGBW color filter arrays. In *SPIE Electronic Imaging 2016*; Eschbach, R., Gabriel G., Marcu, A.R., Eds.; Society for Imaging Science and Technology: Bellingham, WA, USA, 2015; Volume 9395, pp. 1–6. [CrossRef]
17. RGBW Filter Spectral Responses of VEML6040 Sensor. Available online: http://www.vishay.com/docs/84276/veml6040.pdf (accessed on 19 January 2022).
18. Klatzer, T.; Hammernik, K.; Knobelreiter, P.; Pock, T. Learning joint demosaicing and denoising based on sequential energy minimization. In Proceedings of the 2016 IEEE International Conference on Computational Photography (ICCP), Evanston, IL, USA, 13–15 May 2016; pp. 1–11. [CrossRef]
19. Gharbi, M.; Chaurasia, G.; Paris, S.; Durand, F. Deep joint demosaicking and denoising. *ACM Trans. Graph.* **2016**, *35*, 1–12. [CrossRef]
20. Huang, T.; Wu, F.F.; Dong, W.; Shi, G.; Li, X. Lightweight Deep Residue Learning for Joint Color Image Demosaicking and Denoising. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 127–132. [CrossRef]
21. Dmitry, U.; Andrea, V.; Victor, L. Deep Image Prior. *Int. J. Comput. Vis.* **2020**, *128*, 1867–1888. [CrossRef]
22. Park, S.W.; Kang, M.G. Generalized color interpolation scheme based on intermediate quincuncial pattern. *J. Electron. Imaging* **2014**, *23*, 030501. [CrossRef]
23. The Kodak Color Image Dataset. Available online: http://r0k.us/graphics/kodak/ (accessed on 19 January 2022)
24. Zhang, L.; Wu, X.; Buades, A.; Li, X. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *J. Electron. Imaging* **2011**, *20*, 023016.