sensors

MDPI

Article

Joint Demosaicing and Denoising Based on a Variational Deep Image Prior Neural Network

Yunjin Park [1,†], Sukho Lee [2,†], Byeongseon Jeong [3,†] and Jungho Yoon [1,*]

[1] Department of Mathematics, Ewha W. University, Seoul 03760, Korea; yunjjin92@gmail.com
[2] Division of Computer Engineering, Dongseo University, Busan 47011, Korea; petrasuk@gmail.com
[3] Institute of Mathematical Sciences, Ewha W. University, Seoul 03760, Korea; bjeong_ewha@ewha.ac.kr
* Correspondence: yoon@ewha.ac.kr; Tel.: +82-2-3277-2293
† These authors contributed equally to this work.

Received: 2 March 2020; Accepted: 20 May 2020; Published: 24 May 2020

check for updates

Abstract: A joint demosaicing and denoising task refers to the task of simultaneously reconstructing and denoising a color image from a patterned image obtained by a monochrome image sensor with a color filter array. Recently, inspired by the success of deep learning in many image processing tasks, there has been research to apply convolutional neural networks (CNNs) to the task of joint demosaicing and denoising. However, such CNNs need many training data to be trained, and work well only for patterned images which have the same amount of noise they have been trained on. In this paper, we propose a variational deep image prior network for joint demosaicing and denoising which can be trained on a single patterned image and works for patterned images with different levels of noise. We also propose a new RGB color filter array (CFA) which works better with the proposed network than the conventional Bayer CFA. Mathematical justifications of why the variational deep image prior network suits the task of joint demosaicing and denoising are also given, and experimental results verify the performance of the proposed method.

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

1. Introduction

Nowadays, many digital imaging systems use a single monochrome sensor with color filter array (CFA) to capture a color image. Without color filters, the monochrome camera sensor would give only brightness or luminance information and could not recover the colors of the light that fall on each pixel. To obtain color information, every pixel is covered with a color filter that only lets through a certain color of light: red, green or blue. These sampled red, green and blue channels are then interpolated to fill in the missing information at the pixels for which certain colors could not be sampled. This procedure is called the demosaicing procedure, for which many methods have been proposed [1–9].

Current image sensors come in CCD (charge-coupled device) or CMOS (complementary metal–oxide–semiconductor) types, which are both sensitive to thermal noise. Therefore, CFA pattern images taken in low illumination suffer from noise of Poisson distribution. The noise in the CFA pattern image has a large effect on the reconstruction of the color image, as the noise in the noisy pixels spreads out to neighboring regions by the demosaicing process. The denoising is also a challenging task since at least two-thirds of the data are missing. Complex aliasing problems can occur in the demosaicing process if a poor denoising is applied beforehand. As most digital camera pipelines are sequential, quite often the demosaicing and denoising are also performed in an independent and sequential way. This leads to an irreversible error accumulation, since both the demosaicing and the denoising are ill-posed problems and the error occurring in one of the procedures cannot be undone in the other procedure. It has been shown that simultaneous coping with the errors in both the denoising and the demosaicing has advantages, and some joint demosaicing and denoising methods based on optimization techniques have been developed [10–12]. Recently, inspired by the success of the convolutional neural network (CNN) in many image processing tasks, methods which use the CNN in joint demosaicing and denoising have been proposed [13–15]. The work of [13] is a first attempt to apply a learning approach for demosaicing, and the work in [14] is a first attempt to use a convolutional network for joint demosaicing and denoising, but works only on a single noise level. The work in [15] exposes a runtime parameter and trains the network so that it adapts to a wider range of noise levels, but still can only work with a relatively low level of noise. The work of [16] proposes a residue learning neural network structure for the joint demosaicing and denoising problem based on the analysis of the problem using sparsity models, and that of [17] presents a method to learn demosaicing directly from mosaiced images without requiring ground truth RGB data, and showed that a specific burst improves the fine-tuning of the network. Furthermore, the work in [18] proposes a demosaicing network which can be described as an iterative process, and proposes a principled way to design a denoising network architecture. Such CNN-based methods need a lot of data to be trained, and normally, work poorly with varying noise.

In this paper, we propose a deep image prior based method which needs only the noisy image as the training data for the demosaicing. The proposed method uses as the input a sum of a constant and varying noise. We give mathematical justifications as to why the added varying input noise results in the denoising of the demosaiced image. Furthermore, we propose a color filter array which suits the proposed demosaicing method and show experimentally that the proposed method yields good joint demosaicing and denoising results.

2. Related Works

The following works are related to the proposed method. The proposed method can be seen as a variation of the following works fitted to the joint demosaicing and denoising problem.

2.1. Deep Image Prior

Recently, in [19], a deep image prior (DIP) has been proposed for image restoration. The DIP is a type of convolutional neural network which resembles an auto encoder, but which is trained with a single image $\mathbf{x}_0$; i.e., only with the image to be restored. The original DIP converts a 3D tensor $\mathbf{z}$ into a restored image $f_\theta(\mathbf{z})$, where $f_\theta(\cdot)$ denotes the deep image prior network with parameters θ. The tensor $\mathbf{z}$ is filled with random noise from a uniform distribution. The DIP can be trained to inpaint an image with a loss function as follows:

$$L = D\left(m \odot f_\theta(\mathbf{z}), m \odot \mathbf{x}_0\right), \tag{1}$$

where $m \in \{0,1\}^{H \times W}$ is a binary mask with values of zero corresponding to the missing pixels to be inpainted and values of one corresponding to the existing pixels which have to be kept, and $\odot$ is an element-wise multiplication operator. Here, D is a distance measure, which is normally set as the square of the L_2 difference operator; i.e., $D(a,b) = \|a-b\|_2^2$. The minimization of L in Equation (1) with respect to the parameters θ of the DIP network has been shown to be capable of inpainting an image; i.e., the minimization results in an inpainted image $f_\theta(\mathbf{z})$. Inpainting and demosaicing are similar in that they try to fill in missing pixels. The difference is that in inpainting the existing pixels have full channel information, i.e., all R, G and B values are available, whereas in the demosaicing the existing pixels have only one of the R, G and B values.

2.2. Variational Auto Encoder

The variational auto-encoder [20] is a stochastic spin of the auto-encoder which consists of an encoder $q_\theta(\mathbf{z} \mid \mathbf{x})$ and a decoder $p_\phi(\mathbf{x} \mid \mathbf{z})$, where both the encoder and decoder are neural networks with parameters θ and ϕ, respectively. Given an image $\mathbf{x}$ as the input, the encoder $q_\theta(\mathbf{z} \mid \mathbf{x})$ outputs parameters to a Gaussian probability distribution. After that, samples are drawn from this distribution to get a noise input $\mathbf{z}$ to the decoder $p_\phi(\mathbf{x} \mid \mathbf{z})$. The space from which $\mathbf{z}$ is sampled is stochastic and of lower dimension than the space of $\mathbf{x}$. By sampling different samples each time, the variational auto-encoder learns to generate different images. The proposed method also samples noise from a Gaussian distribution, but, unlike the variational auto-encoder, the noise is constituted of constant noise and varying noise and is an input into the network and not used as an intermediate input stage to the decoder.

3. Variational Deep Image Prior for Joint Demosaicing and Denoising

In this section, we propose a variational deep image prior (DIP) for joint demosaicing and denoising. We denote by f_θ the DIP network, and use the same network structure as the DIP for inpainting as defined in [19]; i.e., a U-Net type network which is downsampled five times and upsampled five times. The loss function for the variational DIP differs from that of the original DIP as follows:

$$L = \begin{cases} \|m_r \odot (f_\theta(\mathbf{z}_c) - \mathbf{y}^k)\|^2] & \text{for } k < P \\ \\ \|m_r \odot (f_\theta(\mathbf{z}_c + \mathbf{z}_v) - \mathbf{y}^k)\|^2] & \text{for } k \geq P. \end{cases} \tag{2}$$

Here, $\mathbf{z}_c$ and $\mathbf{z}_v$ denote the constant noise and the varying noise, respectively, both derived from a Gaussian distribution, and m_r is the binary mask corresponding to the proposed random CFA; i.e., it consists of three channels. Each channel constitutes of 33% of random pixels with the value one and 66% of random pixels having the value zero. Unlike the inpainting problem, the positions of the pixels having the value one are different for each channel. The input to f_θ is a constant noise ($\mathbf{z}_c$) until the $(P-1)$-th training step. Then, after the $(P-1)$-th training step, the input becomes the sum of a constant noise ($\mathbf{z}_c$) and a varying noise ($\mathbf{z}_v$), where the noise $\mathbf{z}_v$ is newly generated and differs for each training step. The effect of adding this varying noise $\mathbf{z}_v$ will be explained later. The target image $\mathbf{y}^k$ also differs for the different iteration steps:

$$\mathbf{y}^k = \begin{cases} \mathbf{x}_0 & \text{for } k < P \\ \\ \underset{\mathbf{y}}{\operatorname{argmin}}[(1-\alpha-\beta)\|m_r \odot (\mathbf{y} - \mathbf{y}^{k-1})\|^2 + \alpha\|m_r \odot (\mathbf{y} - \mathbf{x}_0)\|^2 & \\ \quad +\beta\|m_r \odot (\mathbf{y} - f_\theta(\mathbf{z}_c))\|^2] & \text{for } k \geq P \end{cases} \tag{3}$$

For the steps $k < P$, $\mathbf{y}^k = \mathbf{x}_0$, $\mathbf{x}_0$ is a three channel image, wherein each channel contains 33% of either the R, G or B intensity values at the pixel positions where the R, G and B values are sensed by the random CFA, respectively, and 66% of zero values at the remaining positions. Furthermore, we assume that $\mathbf{x}_0$ contains noise. Therefore, if $\mathbf{y}^k = \mathbf{x}_0$ for all steps k, then the reconstructed image f_θ will converge to a noisy demosaiced image. To avoid this, after the step $k = P$, the target image $\mathbf{y}^k$ becomes a weighted average of the previous target image $\mathbf{y}^{k-1}$, the given noisy image $\mathbf{x}_0$ and $f_\theta(\mathbf{z}_c)$. The weights between these images are controlled by α and β. In the experiments, we let $\alpha = 0.003$ and $\beta = 0.007$ for all images. It should be noted that $f_\theta(\mathbf{z}_c)$ differs from the current output $f_\theta(\mathbf{z}_c + \mathbf{z}_v)$. The image $f_\theta(\mathbf{z}_c)$ is a denoised version of $\mathbf{y}^{k-1}$, and the adding of it denoises the target image $\mathbf{y}^k$. The adding of $\mathbf{x}_0$ has the effect of adding back the noise, a trick which is widely used in denoising algorithms to restore the fine details back to the target image. The adding of the previous target image $\mathbf{y}^{k-1}$ keeps the record of the denoised target image. By using the improved target image $\mathbf{y}^k$, the network is not trained toward the given noisy image $\mathbf{x}_0$ any longer, which results in a better denoised output image. Figure 1 shows the working flow of the proposed method. Here, the orange bullets and the solid lines refer to the neural network used in the computation with the specific parameters of $\theta_0, \theta_1, \theta_2, \ldots$, and the dashed lines refer to the inputs to the network. Again, it can be observed that $\mathbf{z}_c$ remains constant, while $\mathbf{z}_{v_P}$, $\mathbf{z}_{v_{P+1}}, \ldots$ are changing over the time.

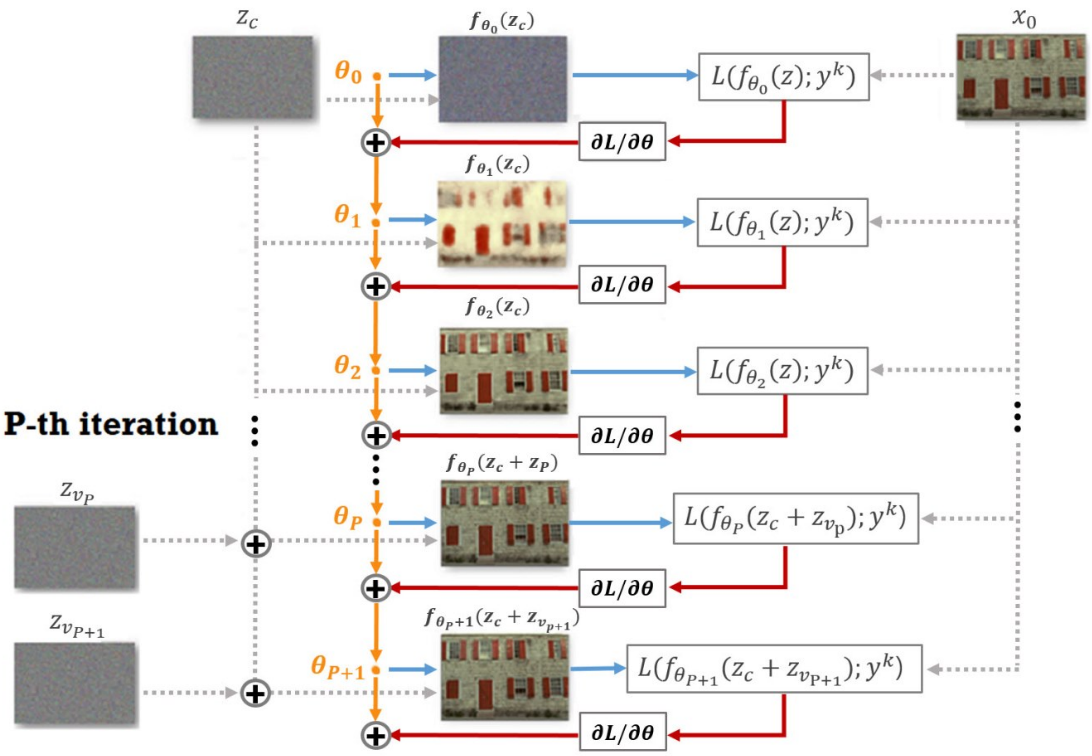
z_c
$f_{\theta_0}(z_c)$
x_0
θ_0
$L(f_{\theta_0}(z); y^k)$
$\partial L/\partial\theta$
$f_{\theta_1}(z_c)$
θ_1
$L(f_{\theta_1}(z); y^k)$
$\partial L/\partial\theta$
$f_{\theta_2}(z_c)$
θ_2
$L(f_{\theta_2}(z); y^k)$
$\partial L/\partial\theta$
P-th iteration
$f_{\theta_P}(z_c + z_P)$
z_{v_P}
θ_P
$L(f_{\theta_P}(z_c + z_{v_P}); y^k)$
$\partial L/\partial\theta$
$f_{\theta_P+1}(z_c + z_{v_{p+1}})$
$z_{v_{P+1}}$
θ_{P+1}
$L(f_{\theta_{P+1}}(z_c + z_{v_{P+1}}); y^k)$
$\partial L/\partial\theta$

Figure 1. Diagram of the proposed method. The variational noise $\mathbf{z}_{v_P}$, $\mathbf{z}_{v_{P+1}}$, ... is added after the (P-1)-th iteration.

Next, we give mathematical justifications on the joint demosaicing and denoising property of the variational DIP. First, the reason which explains why the DIP performs demosaicing can be found in [21], where it is proven that an auto-encoder with an insufficient number of input channels performs an approximation of the target signal under the Hankel structured low-rank constraint:

$$\begin{array}{ll} \min_{f \in \mathbb{R}^n} & \|f^* - f\|^2 \\ \text{subject to} & \mathrm{RANK}(\mathbb{H}_{d|p}(f)) \leq r < pd \end{array} \tag{4}$$

where $\mathbb{H}_{d|p}(f)$ denotes the Hankel matrix of f with p input channels and a convolution filter of size d, and f^* is the target signal. The above approximation can be easily extended to the two-dimensional case. Thus, letting $f = f_\theta(\mathbf{z}_c)$ and $f^* = \mathbf{y}^k$ for the two-dimensional case, we see that we get a low rank approximation of $\mathbf{y}^k$. It has been already shown in [22], that a low rank approximation can perform a reconstruction of missing pixels. When applied to the CFA patterned image, this results in a demosaicing. The Hankel structured low-rank approximation in Equation (4) performs a better approximation than the method in [22], since in [22] the low rank approximation is with respect to the Fourier basis, whereas in Equation (4) this is with respect to a learned basis which best reconstructs the given image, and therefore, yields a better demosaicing result.

Now, to consider the effect of adding the varying noise $\mathbf{z}_v$ to the constant noise $\mathbf{z}_c$, we consider a multi-variable vector-valued function $f_\theta(\mathbf{z})$ which can be expressed as a set of M multi-variable scalar-valued function $f_{\theta_i}(\mathbf{z})$:

$$f_\theta(\mathbf{z}) = \left[f_{\theta_1}(\mathbf{z}), \cdots, f_{\theta_M}(\mathbf{z})\right]^T.$$

The Taylor expansion of the i-th ($i = 1, \cdots, M$) component is

$$f_{\theta_i}(\mathbf{z} + \delta\mathbf{z}) = f_{\theta_i}(\mathbf{z}) + \mathbf{g}^T\delta\mathbf{z} + \frac{1}{2}\delta\mathbf{z}^T\mathbf{H}\delta\mathbf{z} + \epsilon(\delta\mathbf{z}^3) \tag{5}$$

where $\delta\mathbf{z} = [\delta\mathbf{z}_1, \cdots, \delta\mathbf{z}_N]^T$ is a vector. The gradient vector $\mathbf{g}$ and the Hessian matrix $\mathbf{H}$ are the first and second order derivatives of the function $f_{\theta_i}(\mathbf{z})$ defined as:

$$\mathbf{g} = g_{f_{\theta_i}}(\mathbf{z}) = \nabla f_{\theta_i}(\mathbf{z}) = \frac{d}{dz} f_{\theta_i}(\mathbf{z}) = \begin{bmatrix} \frac{\partial f_{\theta_i}}{\partial \mathbf{z}_1} \\ \vdots \\ \frac{\partial f_{\theta_i}}{\partial \mathbf{z}_N} \end{bmatrix}$$

$$\mathbf{H} = H_{f_{\theta_i}}(\mathbf{z}) = \frac{d}{dz} g_{f_{\theta_i}}(\mathbf{z}) = \begin{bmatrix} \frac{\partial^2 f_{\theta_i}}{\partial^2 z_1} & \cdots & \frac{\partial^2 f_{\theta_i}}{\partial \mathbf{z}_1 \partial \mathbf{z}_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f_{\theta_i}}{\partial \mathbf{z}_N \partial \mathbf{z}_1} & \cdots & \frac{\partial^2 f_{\theta_i}}{\partial^2 z_N} \end{bmatrix}.$$

We can extend Equation (5) to $f_\theta(\mathbf{z})$ as a vector form. The first two terms can be written as

$$\mathbf{f}_\theta(\mathbf{z} + \delta\mathbf{z}) \approx \begin{bmatrix} f_{\theta_1}(\mathbf{z}) \\ \vdots \\ f_{\theta_M}(\mathbf{z}) \end{bmatrix} + \begin{bmatrix} \frac{\partial f_{\theta_1}}{\partial z_1} & \cdots & \frac{\partial f_{\theta_1}}{\partial z_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{\theta_M}}{\partial z_1} & \cdots & \frac{\partial f_{\theta_M}}{\partial z_N} \end{bmatrix} \begin{bmatrix} \delta z_1 \\ \vdots \\ \delta z_N \end{bmatrix} = \mathbf{f}_\theta(\mathbf{z}) + \mathbf{J}_{f_\theta}(\mathbf{z})\delta\mathbf{z}$$

where $\mathbf{J}_{f_\theta}(\mathbf{z})$ is the Jacobian matrix defined over $f_\theta(\mathbf{z})$. The second order term requires a tensor form to be expressed which is difficult to write in vector form, and therefore, we replace the second order term with the error term $\epsilon(\|\mathbf{z}_v\|^2)$. Then, we can express $f_\theta(\mathbf{z}_c + \mathbf{z}_v)$ by the Taylor expansion

$$f_\theta(\mathbf{z}_c + \mathbf{z}_v) = f_\theta(\mathbf{z}_c) + \mathbf{J}_{f_\theta}(\mathbf{z}_c)\mathbf{z}_v + \epsilon(\|\mathbf{z}_v\|^2).$$

This results in

$$\begin{aligned} |f_\theta(\mathbf{z}_c + \mathbf{z}_{v_2}) - f_\theta(\mathbf{z}_c + \mathbf{z}_{v_1})| &= |\mathbf{J}_{f_\theta}(\mathbf{z}_c)(\mathbf{z}_{v_2} - \mathbf{z}_{v_1}) + \epsilon(\|\mathbf{z}_{v_2}\|^2) - \epsilon(\|\mathbf{z}_{v_1}\|^2)| \\ &= M\|\mathbf{z}_{v_2} - \mathbf{z}_{v_1}\|_2 \end{aligned} \tag{6}$$

where

$$M = \left| \mathbf{J}_{f_\theta}(\mathbf{z}_c) \frac{(\mathbf{z}_{v_2} - \mathbf{z}_{v_1})}{\|\mathbf{z}_{v_2} - \mathbf{z}_{v_1}\|_2} + \frac{\epsilon(\|\mathbf{z}_{v_2}\|^2)}{\|\mathbf{z}_{v_2} - \mathbf{z}_{v_1}\|_2} - \frac{\epsilon(\|\mathbf{z}_{v_1}\|^2)}{\|\mathbf{z}_{v_2} - \mathbf{z}_{v_1}\|_2} \right| \tag{7}$$

and $\mathbf{J}_{f_\theta}(\mathbf{z})$ is the Jacobian matrix defined over the vector function $f_\theta(\mathbf{z})$:

$$\mathbf{J}_{f_\theta}(\mathbf{z}) = \frac{d}{d\mathbf{z}} f_\theta(\mathbf{z})$$

Equation (6) implies the fact that if $M \neq 0$ and $\mathbf{z}_{v_2} \neq \mathbf{z}_{v_1}$, then $f_\theta(\mathbf{z}_c + \mathbf{z}_{v_2}) \neq f_\theta(\mathbf{z}_c + \mathbf{z}_{v_1})$; i.e., the outputs $f_\theta(\mathbf{z}_c + \mathbf{z}_{v_2})$ and $f_\theta(\mathbf{z}_c + \mathbf{z}_{v_1})$ cannot be the same. This is contradictory to the loss function in Equation (2), which minimization forces the outputs $f_\theta(\mathbf{z}_c + \mathbf{z}_{v_k})$ to converge to the same image $\mathbf{y}^0$ for all different inputs $\mathbf{z}_{v_k}, k \geq P$. It should be noted that, with very high probability, $M \neq 0$ and $\mathbf{z}_{v_1} \neq \mathbf{z}_{v_2}$, since $\mathbf{z}_{v_1}$ and $\mathbf{z}_{v_2}$ are random noises. Therefore, the different inputs of $\mathbf{z}_{v_k}$ act as regularizers which eliminate the components with small L_2 norm energy from $f_\theta(\mathbf{z}_c + \mathbf{z}_{v_k})$. As the components with small energy will be mostly the noise, this will result in a noise removal of $f_\theta(\mathbf{z}_c + \mathbf{z}_{v_k})$.

Furthermore, if we take the expectation of the different outputs $f_\theta(\mathbf{z}_c + \mathbf{z}_{v_k})$ with respect to $\mathbf{z}_{v_k}$, we get

$$\begin{aligned} \mathbb{E}_{\mathbf{z}_{v_k}}\{f_\theta(\mathbf{z}_c + \mathbf{z}_{v_k})\} &\approx \mathbb{E}_{\mathbf{z}_{v_k}}\{f_\theta(\mathbf{z}_c)\} + \mathbb{E}_{\mathbf{z}_{v_k}}\{\mathbf{J}_{f_\theta}(\mathbf{z}_c)\mathbf{z}_{v_k}\} \\ &= \mathbb{E}_{\mathbf{z}_{v_k}}\{f_\theta(\mathbf{z}_c)\} + \mathbf{J}_{f_\theta}(\mathbf{z}_c)\mathbb{E}_{\mathbf{z}_{v_k}}\{\mathbf{z}_{v_k}\} = \mathbb{E}_{\mathbf{z}_{v_k}}\{f_\theta(\mathbf{z}_c)\} = f_\theta(\mathbf{z}_c), \end{aligned} \tag{8}$$

which shows the fact that if we put $\mathbf{z}_c$ as the input after the DIP has been trained, the output $f_\theta(\mathbf{z}_c)$ will be approximately the average of the outputs $f_\theta(\mathbf{z}_c + \mathbf{z}_{v_k})$ for different $\mathbf{z}_{v_k}$. This averaging has a further denoising effect which will remove the remaining noise.

The fact that different inputs of $\mathbf{z}_{v_k}$ result in different outputs can also be shown by the mean value theorem. According to the mean value theorem, there always exists a point $\tilde{\mathbf{z}}$ between $\mathbf{z}_{v_1}$ and $\mathbf{z}_{v_2}$ such that the following equality holds:

$$f_\theta(\mathbf{z}_c + \mathbf{z}_{v_1}) - f_\theta(\mathbf{z}_c + \mathbf{z}_{v_2}) = \nabla_{\mathbf{z}} f_\theta(\tilde{\mathbf{z}}) \cdot (\mathbf{z}_{v_1} - \mathbf{z}_{v_2}). \tag{9}$$

When $\mathbf{z}_{v_1} \neq \mathbf{z}_{v_2}$, then the right-hand side of Equation (9) is, with very high probability, non-zero, since it is almost unlikely that $\nabla_{\mathbf{z}} f_\theta(\tilde{\mathbf{z}})$ and $(\mathbf{z}_{v_1} - \mathbf{z}_{v_2})$ are orthogonal to each other. Therefore, with very high probability,

$$|f_\theta(\mathbf{z}_c + \mathbf{z}_{v_1}) - f_\theta(\mathbf{z}_c + \mathbf{z}_{v_2})| = |\nabla_{\mathbf{z}} f_\theta(\tilde{\mathbf{z}})||\mathbf{z}_{v_1} - \mathbf{z}_{v_2}|cosine(\gamma) \neq 0, \tag{10}$$

where γ is the angle between $\nabla_{\mathbf{z}} f_\theta(\tilde{\mathbf{z}})$ and $\mathbf{z}_{v_1} - \mathbf{z}_{v_2}$. This means that if there is a difference between the inputs, then the outputs of the DIP cannot be the same, so there will be an averaging which removes the noise.

Next, we propose a CFA pattern, which we think works well with the proposed demosaicing method. The proposed CFA consists of randomly distributed pixels, where the pixels corresponding to the R, G and B channels take up 33% of the whole CFA pattern each. The design of the proposed CFA is not based on a rigorous analysis, as done in classical CFA designs [23–25], but on simple reasoning and experimental results. We reason that if the filters are to learn to generate the R, G and B pixels without any bias for a specific color or a specific position, the best training method would be to train the filters to generate any random color at any random position. For example, if the CFA pattern has, for example, 50% green pixels, as in the Bayer format, the convolutional filters will be trained mainly how to generate the green pixels from the noise. When trained like this, the same convolutional filters may be less effective in generating the R or B pixels. Therefore, we reason that the amount of information should be the same for all three channels; i.e., the CFA should consist of 33% of R, G and B pixels each. In the same manner, we reason that if the filters are to learn to generate the R, G and B pixels without any bias for a specific position, it would be good to train the filters to generate any random color at any random position, which is why we propose a pattern with randomly distributed color pixels. Experimental results show that the randomly patterned CFA works better with the proposed demosaicing method than the Bayer pattern or the Fuji X-Trans pattern. Figure 2 shows the different color filter arrays (CFAs) which are used in the experiments including the proposed CFA.

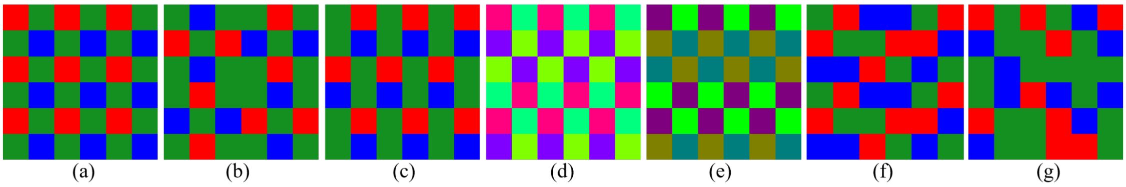

Figure 2. Different color filter arrays (CFAs): (**a**) Bayer CFA [26]; (**b**) Fuji X-Trans CFA [27]; (**c**) Lukac [23]; (**d**) Hirakawa Pattern-A [25]; (**e**) Hirakawa Pattern-B [25]; (**f**) proposed CFA (1:1:1); (**g**) proposed CFA (1:2:1).

4. Experimental Results

We compared the proposed method with other deep-learning-based demosaicing methods on three different datasets. We added different noises generated from Gaussian distributions with different standard deviations of σ_R, σ_G and σ_B, for the R, G and B channels, respectively. This is due to the fact that the R, G and B filters absorb different light energy. We compared the proposed method with the method in [28] as a representative of the non-deep learning demosaicing method; the sequential energy minimization (SEM) method [14]; the DemosaicNet [15] with two different CFAs, i.e., the DemosaicNet with Bayer CFA(DNetB) and the DemosaicNet with the Fuji X-Trans CFA(DNetX); and the plain DIP [19]. We made quantitative comparisons with the PSNR (peak signal-to-noise ratio), the CPSNR (color peak signal-to-noise ratio), the SSIM (structural similarity index), the FSIM (feature similarity index) and the FSIMc (feature similarity index chrominance) measures and summarized the results in tables 2–4. The values in the tables are the average values for the Kodak images and the McMaster (also known as the IMAX) images, respectively. Furthermore, the values corresponding to the red, green and blue channels are the average values for those particular channels, respectively. The parameters for α and β in Equation (3) were set to 0.003 and 0.007, respectively, throughout all the experiments.

Table 1 shows the results of performance comparison when the proposed method was applied for the different CFA patterns; i.e, the Bayer [26], the Fuji X-Trans [27], the Lucak [23], the Hirakawa [25] and the proposed CFA patterns with different noise levels. For the Hirakawa CFA we used the pattern-A and pattern-B patterns which have RGB ratios of 1:1:1 and 1:2:1, respectively. Likewise, for the proposed random patterned CFA, we used the Random1 pattern (RGB ratio of 1:1:1) and the Random2 pattern (RGB ratio of 1:2:1). The CPSNR, SSIM and FSIMc values are the average values of the images in the Kodak image dataset. When the noise is low, the Hirakawa pattern-A CFA shows the largest CPSNR and SSIM values. However, when the noise increases, the proposed random pattern shows larger PSNR, SSIM and FSIM values. This is maybe due to the fact that when the noise increases, the tasks of demosaicing and denoising become similar—i.e., the task of removing the random noise and the task of filling in random colors become similar—so the finding of the parameters which do the demosaicing and denoising tasks simultaneously becomes an easier task with the proposed random CFA than with other CFAs. It can be seen that the proposed CFA pattern mostly shows the largest value, especially when the noise is large.

Table 1. Comparison of the CPSNR, SSIM and FSIMc values for the different CFA patterns used with the proposed method on the Kodak image dataset.

Noise Level	Measure	Bayer	Xtrans	Lukac	HirakawaA	HirakawaB	Random1	Random2
	CPSNR	31.4386	32.1435	31.8650	**32.5018**	32.0552	32.1918	32.1410
σ_R = 9.83	SSIM-R	0.8543	0.8719	0.8596	**0.8827**	0.8750	0.8718	0.8709
σ_G = 6.24	SSIM-G	0.8870	0.8769	0.8839	**0.8972**	0.8823	0.8861	0.8863
σ_B = 6.84	SSIM-B	0.8530	0.8744	0.8590	0.8854	**0.8863**	0.8719	0.8749
	FSIMc	0.9771	**0.9789**	0.9759	0.9368	0.9740	0.9775	0.9775
	CPSNR	29.2582	29.3572	29.4057	**29.7132**	29.5584	29.3914	29.3710
σ_R = 19.67	SSIM-R	0.7724	0.7719	0.7745	0.8257	**0.8275**	0.7750	0.7927
σ_G = 12.48	SSIM-G	0.8217	0.8207	0.8244	**0.8452**	0.8402	0.8230	0.8206
σ_B = 13.67	SSIM-B	0.7948	0.7992	0.7992	**0.8328**	0.8239	0.8028	0.8084
	FSIMc	0.9537	0.9541	**0.9548**	0.9532	0.9516	**0.9548**	**0.9548**
	CPSNR	28.0037	28.0340	28.0733	27.9030	27.5895	27.9799	**28.2210**
σ_R = 26.22	SSIM-R	0.7184	0.7126	0.7180	**0.7682**	0.7631	0.7127	0.7013
σ_G = 16.64	SSIM-G	0.7801	0.7743	0.7782	0.7682	0.7785	0.7751	**0.7809**
σ_B = 18.23	SSIM-B	0.7556	0.7517	0.7543	0.7645	0.7596	0.7542	**0.7670**
	FSIMc	0.9364	0.9354	0.9358	0.9415	**0.9616**	0.9354	0.9372

Figure 3 shows the results of the different demosaicing methods on the first dataset with color noise of standard deviations $\sigma_R = 9.83$, $\sigma_G = 6.24$ and $\sigma_B = 6.84$ for the R, G and B channels, respectively. The parameter P in Equation (3) is set to 1200 for the experiments with this color noise, and to 500 for all the other experiments. When the noise is light, the SEM, the DNetB and the DNetX also produce good joint demosaicing and denoising results. The SEM shows the best quantitative results in the PSNR values for the Kodak dataset, as can be seen in Table 2. However, the proposed method achieves the best results in the SSIM and the FSIM measures for all datasets, and the best PSNR values for the McMaster dataset. Figures 4 and 5 and Tables 2–4 show the results on the dataset with color noise of standard deviations $\sigma_R = 19.67$, $\sigma_G = 12.48$ and $\sigma_B = 13.67$. The ADMM method got the highest value of cPSNR for the McMaster dataset; that was due to the fact that the ADMM method incorporates the powerful BM3D denoising [29] and the total variation minimization into a single framework, which results in a large denoising power. Therefore, we experimented also on a combination of the proposed method and the BM3D. In this case, the proposed training method can focus more on finding the parameters for the demosaicing task, leaving a large part of denoising to the BM3D, which results in finding effective parameters for demosaicing. The results of the SEM, DNetB and DNetX are those without using the external BM3D denoising method. The proposed + BM3D outperforms the other methods on the Kodak dataset with respect to the PSNR and SSIM measures. As the noise increases, the ADMM, SEM, DNetB and DNetX result in severe color artifacts, as can be observed from the fence regions in the enlarged images in Figure 5b–e. However, the DIP and the proposed method overcome such color artifacts due to the inherent rank minimization property. The figures are selected according to the best PSNR values, which is why the figures for the DIP are a little more blurry than the figures for the proposed method. The DIP reconstructs the noise when reconstructing the high frequency components while the proposed method does not. Finally, Figures 6 and 7 and Tables 2–4 show the results on the dataset with color noise of standard deviations $\sigma_R = 26.22$, $\sigma_G = 16.64$ and $\sigma_B = 18.23$. For this dataset, the non-deep-learning ADMM method outperforms all the deep-learning-based methods, including the proposed method, in the quantitative measures. However, the proposed method outperforms all other deep-learning-based methods. Furthermore, while the ADMM shows large aliasing artifacts, as can be seen in Figure 8b, the proposed method is free from such artifacts. Again, it should be taken into account that this is the result of training with the noisy pattern CFA image only. Furthermore, we fixed all the hyper-parameters of the network for all the different noise levels, which means that the proposed method is not sensitive to the noise levels.

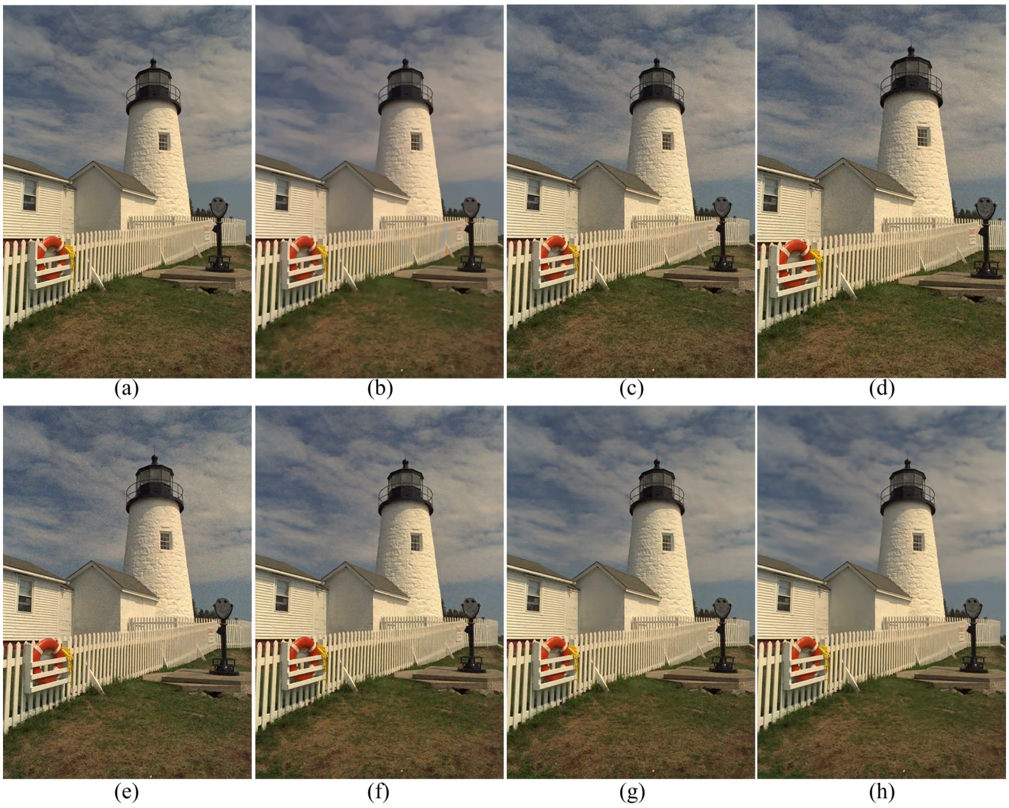

Figure 3. Reconstruction results for the Kodak number 19 image with noise levels $\sigma_R = 9.83$, $\sigma_G = 6.24$ and $\sigma_B = 6.84$. (**a**) Original, (**b**) ADMM, [28] (**c**) SEM, [14] (**d**) DNetB, [15] (**e**) DNetX, [15] (**f**) DIP, [19] (**g**) proposed and (**h**) proposed + BM3D.

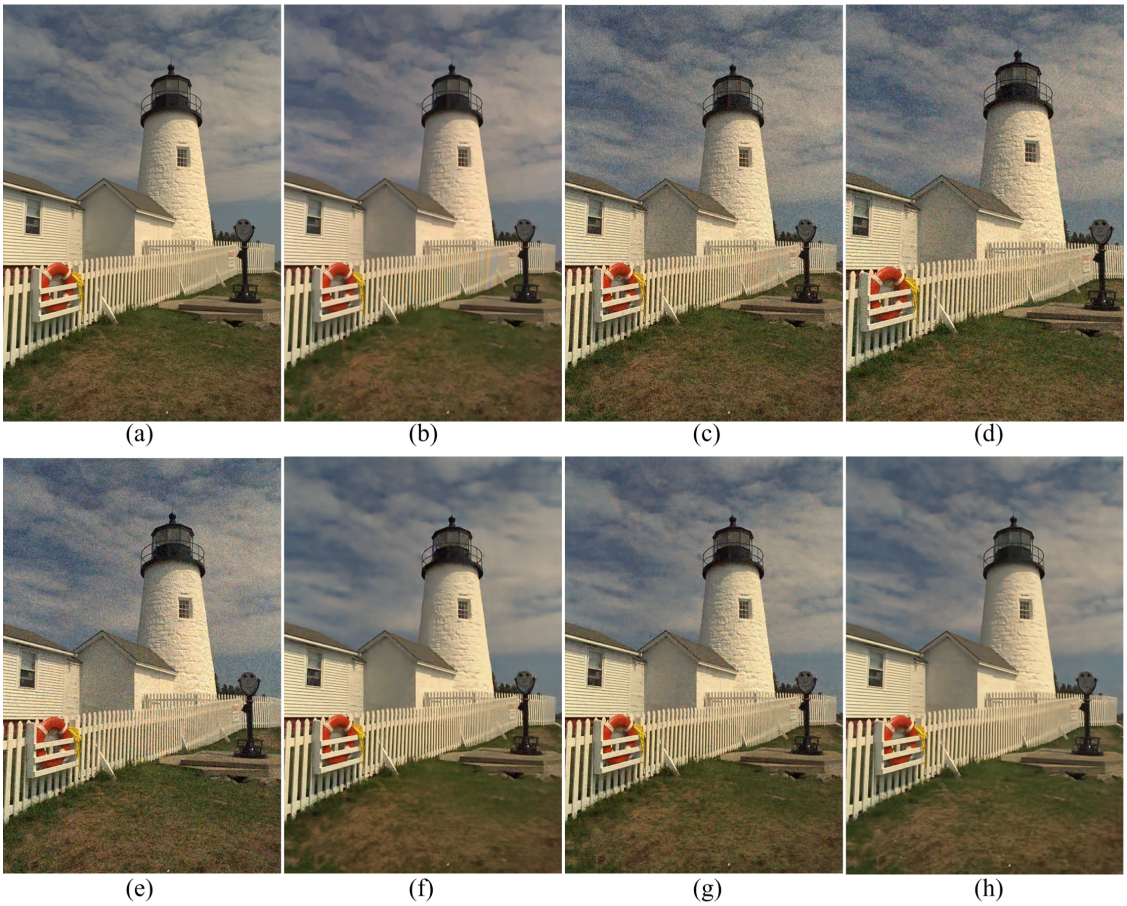

Figure 4. Reconstruction results for the Kodak number 19 image with noise levels $\sigma_R = 19.67$, $\sigma_G = 12.48$ and $\sigma_B = 13.67$. (**a**) Original, (**b**) ADMM, [28] (**c**) SEM, [14] (**d**) DNetB, [15] (**e**) DNetX, [15] (**f**) DIP, [19] (**g**) proposed and (**h**) proposed + BM3D.

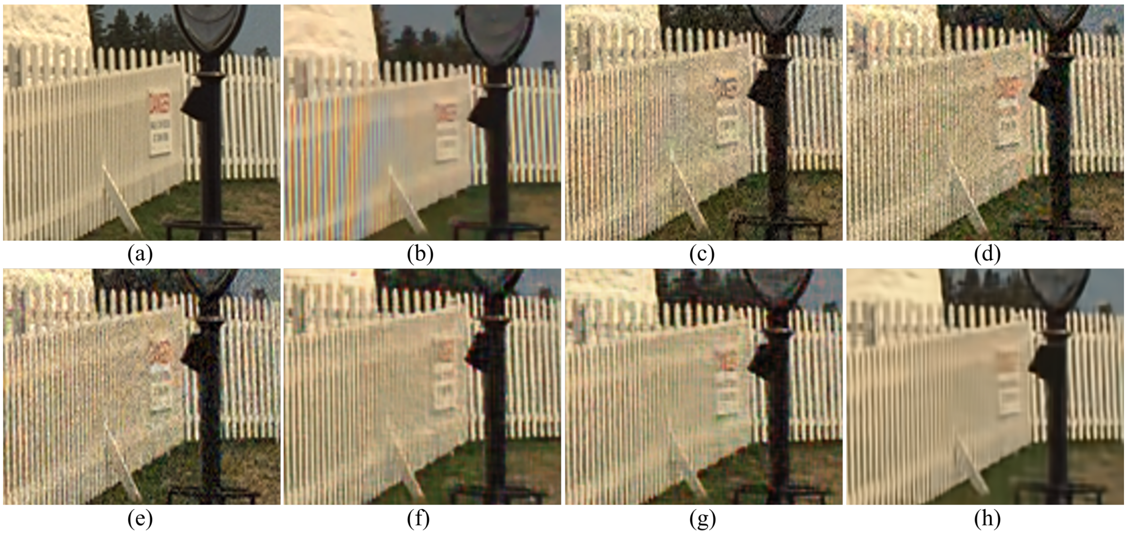

Figure 5. Enlarged regions of Figure 4. (**a**) Original, (**b**) ADMM, [28] (**c**) SEM, [14] (**d**) DNetB, [15] (**e**) DNetX, [15] (**f**) DIP, [19] (**g**) proposed and (**h**) proposed + BM3D.

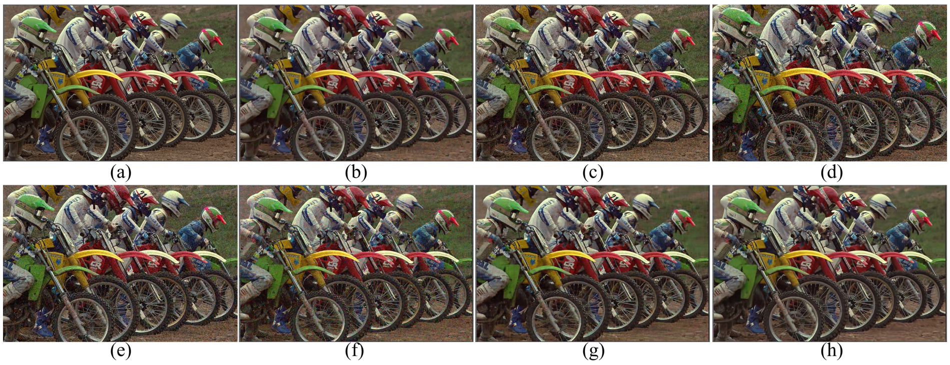

Figure 6. Reconstruction results for the Kodak number 5 image with noise levels $\sigma_R = 26.22$, $\sigma_G = 16.64$ and $\sigma_B = 18.23$. (**a**) Original, (**b**) ADMM, [28] (**c**) SEM, [14] (**d**) DNetB, [15] (**e**) DNetX, [15] (**f**) DIP, [19] (**g**) proposed and (**h**) proposed + BM3D.

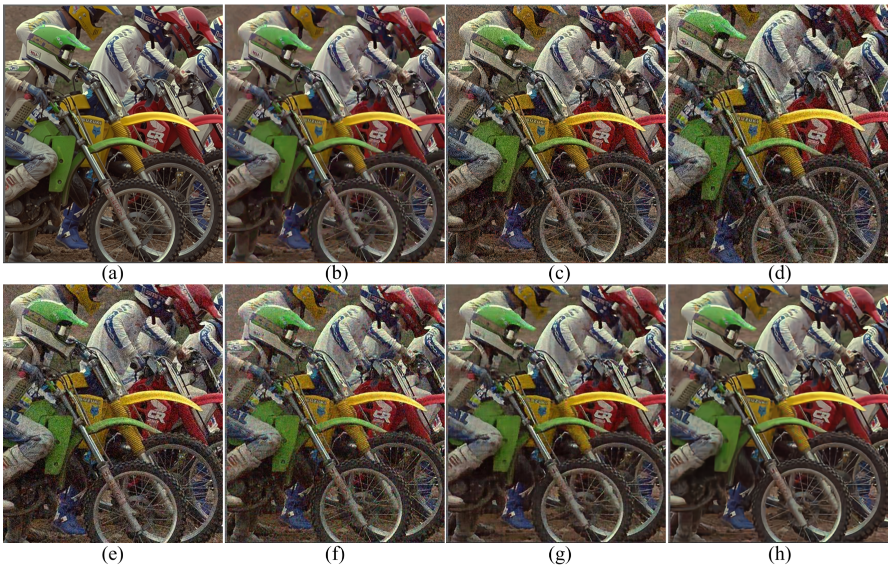

Figure 7. Enlarged regions of Figure 6. (**a**) Original, (**b**) ADMM, [28] (**c**) SEM, [14] (**d**) DNetB, [15] (**e**) DNetX, [15] (**f**) DIP, [19] (**g**) proposed and (**h**) proposed + BM3D.

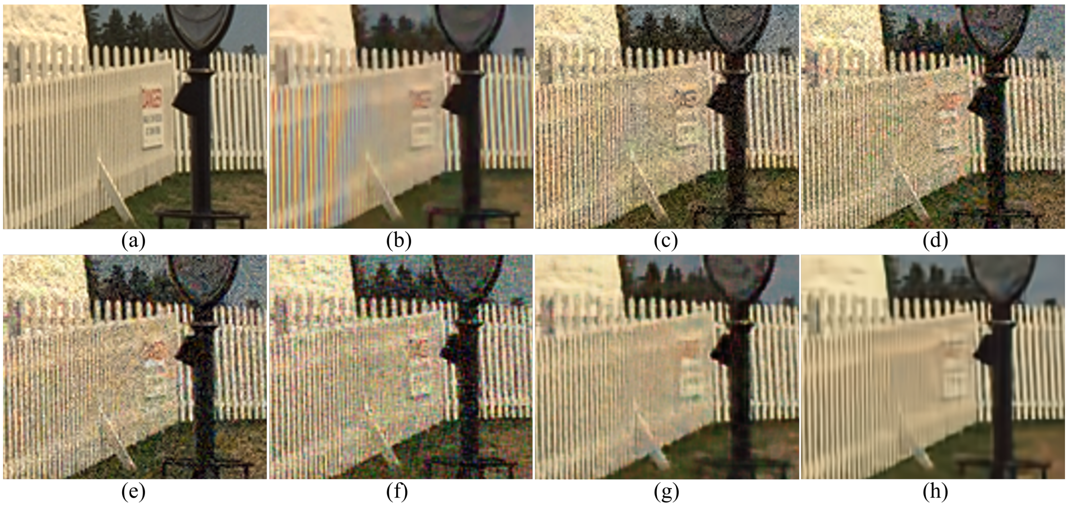

Figure 8. Enlarged regions of the denoising results on the Kodak number 19 image with noise levels $\sigma_R = 26.22$, $\sigma_G = 16.64$ and $\sigma_B = 18.23$. (**a**) Original, (**b**) ADMM, [28] (**c**) SEM, [14] (**d**) DNetB, [15] (**e**) DNetX, [15] (**f**) DIP, [19] (**g**) proposed and (**h**) proposed + BM3D.

Table 2. Comparison of the PSNR values among the various demosaicing methods on the Kodak and the McMaster image datasets.

Noise Level	Dataset	Measure	ADMM	SEM	DNetB	DNetX	DIP	Proposed	Proposed +BM3D
$\sigma_R = 9.83$ $\sigma_G = 6.24$ $\sigma_B = 6.84$	**Kodak**	cPSNR	31.3370	32.5468	31.4053	31.3766	30.4512	32.1410	**32.6103**
		PSNR-R	30.9196	31.6351	30.3037	30.4766	29.5487	31.5406	**32.2743**
		PSNR-G	31.9553	**33.2664**	32.1838	32.0917	31.1663	32.5779	32.8726
		PSNR-B	31.2722	**32.9731**	32.0131	31.7628	30.9852	32.4079	32.7272
	McMaster	cPSNR	32.1258	30.7908	31.1171	31.0155	29.6490	32.6603	**33.0659**
		PSNR-R	31.8458	29.7472	29.9868	29.9864	28.7221	32.2605	**32.9728**
		PSNR-G	33.2163	32.1750	32.2558	32.1366	30.1651	33.1813	**33.4180**
		PSNR-B	31.6661	30.9114	31.5195	31.2861	30.4179	32.8239	**33.0647**
$\sigma_R = 19.67$ $\sigma_G = 12.48$ $\sigma_B = 13.67$	**Kodak**	cPSNR	30.1883	26.1682	25.9266	25.8636	27.2001	29.3710	**30.2026**
		PSNR-R	29.5430	25.1485	24.5928	24.9497	26.1787	28.7261	**29.6902**
		PSNR-G	**30.7728**	26.4776	26.6385	26.2463	27.8791	29.9587	30.4681
		PSNR-B	30.3904	27.1772	26.9603	26.5889	27.9003	29.9483	**30.5204**
	McMaster	cPSNR	**30.5794**	25.7280	26.1234	26.0321	26.7356	29.7984	30.4533
		PSNR-R	29.9344	24.4154	24.6482	24.8721	25.5419	28.9401	**29.9461**
		PSNR-G	**31.5879**	26.5727	27.0705	26.6617	27.3464	30.4840	30.9327
		PSNR-B	30.5402	26.6814	27.2079	26.9079	27.8159	30.3375	**30.7383**
$\sigma_R = 26.22$ $\sigma_G = 16.64$ $\sigma_B = 18.23$	**Kodak**	cPSNR	**29.3247**	23.2843	23.4976	23.4724	25.6570	28.2210	29.0672
		PSNR-R	**28.5560**	22.3131	22.0491	22.5529	24.7341	27.3588	28.4363
		PSNR-G	**29.9424**	23.5352	24.2448	23.7847	26.1365	28.7616	29.3628
		PSNR-B	**29.6403**	24.3010	24.6938	24.2836	26.2839	28.8031	29.5067
	McMaster	cPSNR	**29.5312**	23.2554	23.8799	23.7807	25.4785	28.2050	29.0961
		PSNR-R	**28.6716**	21.8944	22.2896	22.6013	24.2930	27.1305	28.4053
		PSNR-G	**30.5938**	23.9948	24.8180	24.2958	26.0823	29.0621	29.6725
		PSNR-B	**29.6901**	24.3931	25.1801	24.8034	26.5224	28.9340	29.4845

Table 3. Comparison of the SSIM values among the various demosaicing methods on the Kodak and the McMaster image datasets.

Noise Level	Dataset	Measure	ADMM	SEM	DNetB	DNetX	DIP	Proposed	Proposed +BM3D
$\sigma_R = 9.83$ $\sigma_G = 6.24$ $\sigma_B = 6.84$	**Kodak**	SSIM-R	0.8613	0.8581	0.7774	0.7834	0.7977	0.8709	**0.8871**
		SSIM-G	0.8840	0.8785	0.8267	0.8154	0.8625	0.8863	**0.8891**
		SSIM-B	0.8524	**0.8802**	0.8288	0.8209	0.8469	0.8749	0.8794
	McMaster	SSIM-R	0.8846	0.8016	0.7612	0.7662	0.8017	0.8805	**0.8993**
		SSIM-G	**0.9132**	0.8577	0.8222	0.8134	0.8657	0.9013	0.9074
		SSIM-B	0.8624	0.8192	0.8068	0.8000	0.8407	0.8758	**0.8822**
$\sigma_R = 19.67$ $\sigma_G = 12.48$ $\sigma_B = 13.67$	**Kodak**	SSIM-R	0.8253	0.5630	0.5372	0.5405	0.6490	0.7927	**0.8300**
		SSIM-G	**0.8535**	0.6007	0.6087	0.5762	0.7501	0.8206	0.8372
		SSIM-B	0.8264	0.6256	0.6218	0.5931	0.7320	0.8084	**0.8270**
	McMaster	SSIM-R	**0.8416**	0.5258	0.5320	0.5385	0.6529	0.8104	0.8378
		SSIM-G	**0.8817**	0.6083	0.6185	0.5919	0.7725	0.8467	0.8588
		SSIM-B	**0.8288**	0.5941	0.6209	0.6007	0.7485	0.8241	0.8221
$\sigma_R = 26.22$ $\sigma_G = 16.64$ $\sigma_B = 18.23$	**Kodak**	SSIM-R	**0.7972**	0.4296	0.4240	0.4319	0.5881	0.7013	0.7948
		SSIM-G	**0.8315**	0.4642	0.4972	0.4657	0.6875	0.7809	0.8074
		SSIM-B	**0.8058**	0.4858	0.5093	0.4819	0.6706	0.7670	0.7978
	McMaster	SSIM-R	**0.8091**	0.4066	0.4289	0.4364	0.5915	0.7463	0.7961
		SSIM-G	**0.8579**	0.4828	0.5173	0.4864	0.7156	0.8116	0.8265
		SSIM-B	**0.8031**	0.4672	0.5213	0.4962	0.6863	0.7839	0.7873

Figure 9 compares the convergence of the PSNR values according to the training iterations of the plain DIP and the proposed variational DIP, respectively. As can be seen, the plain DIP converges to a lower PSNR value as the training step iterates, which is due to the fact that the noise in the target image is reconstructed. In comparison, with the proposed variational DIP, the noise is not reconstructed, due to the reasons explained in the previous section. Therefore, the final output image converges to a joint demosaiced and denoised image, which results in a convergence to a higher PSNR value.

Table 5 shows the computational time costs of the different methods. All the methods have been run on a PC with an Intel Core i9-9900K Processor, NVIDIA GeForce RTX 2080 Ti and 32 GB RAM. The proposed method is the slowest of all the methods, which is due to the fact that the proposed method uses a training step for each incoming CFA image. The computational time can be reduced if the proposed method is combined with the meta learning approach. One of the possible methods would be to initialize the neural network with good initial parameters obtained by some pre-training with many images. This should be one of the major topics of further studies.

Table 4. Comparison of the FSIM values among the various demosaicing methods on the Kodak and the McMaster image datasets.

Noise Level	Dataset	Measure	ADMM	SEM	DNetB	DNetX	DIP	Proposed	Proposed +BM3D
σ_R = 9.83 σ_G = 6.24 σ_B = 6.84	**Kodak**	FSIMc	0.9722	**0.9792**	0.9743	0.9730	0.9666	0.9775	0.9764
		FSIM-R	0.9580	0.9635	0.9469	0.9471	0.9469	0.9665	**0.9705**
		FSIM-G	0.9729	**0.9794**	0.9738	0.9734	0.9639	0.9736	0.9737
		FSIM-B	0.9574	**0.9728**	0.9650	0.9607	0.9606	0.9704	0.9708
	McMaster	FSIMc	0.9807	0.9788	0.9756	0.9746	0.9708	0.9824	**0.9824**
		FSIM-R	0.9650	0.9565	0.9478	0.9481	0.9540	0.9716	**0.9756**
		FSIM-G	**0.9802**	0.9773	0.9743	0.9743	0.9667	0.9779	0.9784
		FSIM-B	0.9629	0.9600	0.9613	0.9587	0.9638	0.9749	**0.9754**
σ_R = 19.67 σ_G = 12.48 σ_B = 13.67	**Kodak**	FSIMc	**0.9598**	0.9232	0.9225	0.9175	0.9276	0.9548	0.9562
		FSIM-R	0.9399	0.8936	0.8654	0.8738	0.8978	**0.9548**	0.9463
		FSIM-G	**0.9603**	0.9266	0.9217	0.9210	0.9270	0.9514	0.9540
		FSIM-B	0.9471	0.9259	0.9068	0.9014	0.9245	0.9490	**0.9520**
	McMaster	FSIMc	**0.9695**	0.9290	0.9304	0.9268	0.9378	0.9625	0.9656
		FSIM-R	0.9462	0.8911	0.8749	0.8803	0.9086	0.9467	**0.9525**
		FSIM-G	**0.9683**	0.9315	0.9277	0.9284	0.9345	0.9582	0.9618
		FSIM-B	0.9509	0.9197	0.9085	0.9056	0.9320	0.9554	**0.9591**
σ_R = 26.22 σ_G = 16.64 σ_B = 18.23	**Kodak**	FSIMc	**0.9502**	0.8811	0.8835	0.8776	0.9007	0.9372	0.9433
		FSIM-R	0.9255	0.8479	0.8148	0.8272	0.8687	0.9042	**0.9305**
		FSIM-G	**0.9508**	0.8868	0.8848	0.8832	0.8981	0.9352	0.9417
		FSIM-B	0.9383	0.8876	0.8676	0.8607	0.8969	0.9329	**0.9406**
	McMaster	FSIMc	**0.9601**	0.8919	0.8971	0.8929	0.9164	0.9470	0.9530
		FSIM-R	0.9309	0.8462	0.8291	0.8379	0.8855	0.9258	**0.9351**
		FSIM-G	**0.9588**	0.8982	0.8946	0.8964	0.9132	0.9437	0.9500
		FSIM-B	0.9402	0.8869	0.8727	0.8700	0.9092	0.9403	**0.9473**

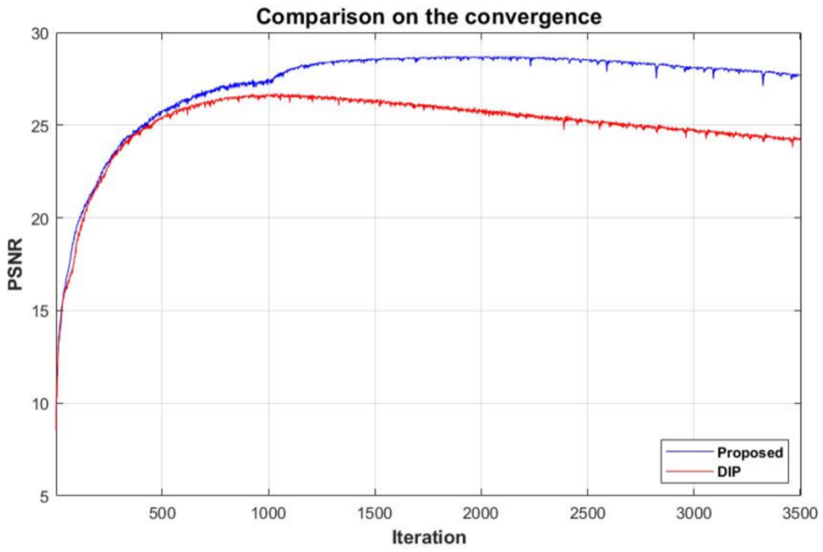

Figure 9. Comparison of the convergence between the DIP and the proposed variational DIP.

Table 5. Showing the computational time costs of the different joint demosaicing and denoising methods.

Method	ADMM	SEM	DNetB	DNetX	DIP	Proposed
Time Cost	567 s	465 s	8 s	8 s	525 s	647 s
Software Tool	MATLAB	Python	Python	Python	Pytorch	Pytorch

5. Conclusions

In this paper, we proposed a variational deep image prior for the joint demosaicing and denoising of the proposed random color filter array. We mathematically explained why the variational model results in a demosaicing and denoising result, and experimentally verified the performance of the proposed method. The experimental results showed that the proposed method is superior to other deep-learning-based methods, including the deep image prior network. How to apply the proposed method on the demosaicing of color filter arrays including channels other than the three primary color channels could be the topic of further studies.

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

Funding: The work of J.Y. was supported in part by the National Research Foundation of Korea under grant NRF-2020R1A2C1A01005894 and the work of S.L. was supported by the Basic Science Research Program through the National Research Foundation of Korea under grant NRF-2019R1I1A3A01060150.

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

References

1. Pekkucuksen, I.; Altunbasak, Y. Multiscale gradients-based color filter array interpolation. *IEEE Trans. Image Process.* **2013**, *22*, 157–165. [CrossRef] [PubMed]
2. Kiku, D.; Monno, Y.; Tanaka, M.; Okutomi, M. Beyond color difference: Residual interpolation for color image demosaicking. *IEEE Trans. Image Process.* **2016**, *25*, 1288–1300. [CrossRef] [PubMed]
3. Gunturk, B.K.; Altunbasak, Y.; Mersereau, R.M. Color plane interpolation using alternating projections. *IEEE Trans. Image Process.* **2002**, *11*, 997–1013. [CrossRef] [PubMed]
4. Alleysson, D.; Süsstrunk, S.; Hérault, J. Linear demosaicing inspired by the human visual system. *IEEE Trans. Image Process.* **2005**, *14*, 439–449. [PubMed]
5. Gunturk, B.K.; Glotzbach, J.; Altunbasak, Y.; Schafer, R.W.; Mersereau, R.M. Demosaicking: Color filter array interpolation. *IEEE Signal Process. Mag.* **2005**, *22*, 44–54. [CrossRef]
6. Kimmel, R. Demosaicing: Image reconstruction from color ccd samples. *IEEE Trans. Image Process.* **1999**, *8*, 1221–1228. [CrossRef] [PubMed]
7. 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.
8. Menon, D.; Calvagno, G. Color image demosaicking: An overview. *Signal Process Image* **2011**, *26*, 518–533. [CrossRef]
9. Dubois, E. Frequency-domain methods for demosaicking of bayer-sampled color images. *IEEE Signal Process. Lett.* **2005**, *12*, 847–850. [CrossRef]
10. Hirakawa, K.; Parks, T.W. Joint demosaicing and denoising. *IEEE Trans. Image Process.* **2006**, *15*, 2146–2157. [CrossRef] [PubMed]
11. Jeon, G.; Dubois, E. Demosaicking of noisy bayer sampled color images with least-squares luma-chroma demultiplexing and noise level estimation. *IEEE Trans. Image Process.* **2013**, *22*, 146–156. [CrossRef] [PubMed]
12. Buades, A.; Duran, J. CFA Video Denoising and Demosaicking Chain via Spatio-Temporal Patch-Based Filtering. *IEEE Trans. Circuits Syst. Video Technol.* **2019**. [CrossRef]
13. Khashabi, D.; Nowozin, S.; Jancsary, J.; Fitzgibbon, A.W. Joint demosaicing and denoising via learned nonparametric random fields. *IEEE Trans. Image Process.* **2014**, *23*, 4968–4981. [CrossRef] [PubMed]
14. 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.
15. Gharbi, M.; Chaurasia, G.; Paris, S.; Durand, F. Deep Joint Demosaicking and Denoising. *ACM Trans. Graph. (TOG)* **2016**, *35*, 1–12. [CrossRef]

16. Huang, T.; Wu, F.; Dong, W.; Guangming, S.; Li, X. Lightweight Deep Residue Learning for Joint Color Image Demosaicking and Denoising. In Proceedings of the 2018 International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 127–132.
17. Ehret, T.; Davy, A.; Arias, P.; Facciolo, G. Joint Demosaicking and Denoising by Fine-Tuning of Bursts of Raw Images. In Proceedings of the 2019 International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 8868–8877.
18. Kokkinos, F.; Lefkimmiatis, S. Iterative Joint Image Demosaicking and Denoising Using a Residual Denoising Network. *IEEE Trans. Image Process.* **2019**, *28*, 4177–4188. [CrossRef] [PubMed]
19. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Deep Image Prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 9446–9454.
20. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the 2nd International Conference on Learning Representations(ICLR 2014), Banff, AB, Canada, 14–16 April 2014.
21. Ye, J.C.; Han, Y.S. Deep Convolutional Framelets: A General Deep Learning for Inverse Problems. *SIAM J. Imaging Sci.* **2017**, *11*, 991–1048. [CrossRef]
22. Zhou, J.; Kwan, C.; Ayhan, B. A High Performance Missing Pixel Reconstruction Algorithm for Hyperspectral Images. In Proceedings of the 2nd International Conference on Applied and Theoretical Information Systems, Taipei, Taiwan, 10–12 February 2012; pp. 1–10.
23. Lukac, R.; Konstantinos, N.P. Color Filter Arrays: Design and Performance Analysis. *IEEE Trans. Consum. Electron.* **2005**, *51*, 1260–1267. [CrossRef]
24. Vaughn, I.J; Alenin, A.S.; Tyo, J.S. Focal plane filter array engineering I: Rectangular lattices. *Opt. Express* **2017**, *25*, 11954–11968. [CrossRef] [PubMed]
25. 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]
26. Bayer, B. Color Imaging Array. U.S. Patent 3,971,065, 20 July 1976.
27. Fujifilm X-Pro1. Available online: http://www.fujifilmusa.com/products/digital_cameras/x/fujifilm_x_pro1/features (accessed on 23 May 2020).
28. Tan, H.; Zeng, X.; Lai, S.; Liu, Y.; Zhang, M. Joint demosaicing and denoising of noisy bayer images with ADMM. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP 2017), Beijing, China, 17–20 September 2017; pp. 2951–2955.
29. Kostadin, D.; Alessandro, F.; Vladimir, K.; Karen, E. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095.