Residual Learning for Effective joint Demosaicing-Denoising

Yu Guo 1 Qiyu Jin 1 Gabriele Facciolo 2 Tieyong Zeng 3 Jean-Michel Morel 2
1 School of Mathematical Science, Inner Mongolia University, China
2 Centre Borelli, ENS Paris-Saclay, CNRS, France
3 Department of Mathematics, The Chinese University of Hong Kong, Satin, Hong Kong
{yuguomath, qyjin2015}@aliyun.com, gabriele.facciolo@ens-paris-saclay.fr,
zeng@math.cuhk.edu.hk, moreljeanmichel@gmail.com

Abstract

Image demosaicing and denoising are key steps for color image production pipeline. The classical processing sequence consists in applying denoising first, and then demosaicing. However, this sequence leads to oversmoothing and unpleasant checkerboard effect. Moreover, it is very difficult to change this order, because once the image is demosaiced, the statistical properties of the noise will be changed dramatically. This is extremely challenging for the traditional denoising models that strongly rely on statistical assumptions. In this paper, we attempt to tackle this prickly problem. Indeed, here we invert the traditional CFA processing pipeline by first applying demosaicing and then using an adapted denoising. In order to obtain high-quality demosaicing of noiseless images, we combine the advantages of traditional algorithms with deep learning. This is achieved by training convolutional neural networks (CNNs) to learn the residuals of traditional algorithms. To improve the performance in image demosaicing we propose a modified Inception architecture. Given the trained demosaicing as a basic component, we apply it to noisy images and use another CNN to learn the residual noise (including artifacts) of the demosaiced images, which allows to reconstruct full color images. Experimental results show clearly that this method outperforms several state-of-the-art methods both quantitatively as well as in terms of visual quality.

1. Introduction

The objective of demosaicing is to build a full color image from four spatially undersampled color channels. Indeed, digital cameras can only capture one color information through a single monochrome sensor on each pixel, and most of them use color filter arrays (CFA) such as Bayer pattern to obtain images. The raw data collected in this way is missing two-thirds of pixels and is contaminated by noise. Hence, image demosaicing, i.e. the task of reconstructing a full color image from the incomplete raw data is a typical ill-posed problem. In the common image processing pipeline, it is taken for granted that the raw data should be denoised first and then demosaiced. Because denoising algorithms are commonly built on certain statistical priors, once the raw data is demosaiced, these priors could be seriously disrupted. Moreover, most standard demosaicing algorithms with good performance are designed based on the critical noise-free condition. Therefore, this pipeline sequence has been widely used without question [18, 25, 32].

However, recently Jin et al. [17] have found experimentally that the sequence of demosaicing first and then denoising can achieve better image quality. This conclusion breaks the previous convention. Since the CFA images are different from the normal grayscale or full color images, CFA denoising will often subsample them into half-size four-channel RGGGB images. This can result in loss of image details due to the reduced resolution, but the relative spatial positions of R, G and B pixels are also lost, making it easier to produce a checkerboard effect [6]. If we want to modify this order, we must first solve a thorny problem, how to remove the noise whose statistical properties are changed by a complex interpolation. This is almost impossible for traditional denoising algorithms, but current data-driven deep learning method brings hope to our problems. In recent years, deep convolutional neural networks have achieved great success in the fields of computer vision and image processing. In image classification and recognition [14, 24, 36, 37], denoising [12, 15, 34, 43, 44], demosaicing [10, 27, 38, 40], super-resolution [9, 11, 26, 28, 42] and other high-level and low-level visual tasks, the effect of deep learning greatly surpasses the traditional methods.

In this work, we first combine convolutional neural networks (CNNs) and traditional algorithms to obtain a demosaicing algorithm. Using this demosaicing as a base, we use another CNN to remove the demosaiced noise, whose sta-
istical properties have been changed. The four main contributions of this work are:

- We present a convolutional neural network with a new mechanism for image demosaicing. This network generates a full color image by learning the residuals between the color image obtained by the GBTF algorithm [33] and the ground truth.

- In order to better carry out the cross-channel information fusion and improve the receptive field of convolution to reduce artifacts caused by image reconstruction, we modified the traditional Inception architecture. Subsequent experiments confirm that the proposed Inception architecture has a powerful effect in image demosaicing.

- We use the same network architecture to learn the residual noise between the demosaiced CFA images and the ground truth for different noise levels. With such a data-driven method, the neural network can fit efficiently the noise whose statistical properties has been dramatically changed after the complex interpolation.

- Quantitative and visual experimental results on Kodak, McMaster and WED + Flickr datasets confirm that our model outperforms the state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 presents related work on demosaicing and denoising. The demosaicing and denoising model is comprehensively introduced in Section 3. Section 4 gives a comparison of quantitative results and visual effects with the state-of-the-art method. The concluding remarks are given in Section 5.

2. Related Work

Demosaicing is a problem with vast literatures. All authors agree that the key to attain a good demosaicing is to restore the image areas with high-frequency content. This is because smooth areas are easy to interpolate from the available samples. Since green samples represent 50% of the information captured with the Bayer pattern and because of the correlation between RGB channels, many traditional algorithms first reconstruct the green channel by using the gradient information to determine the image edges. Then the green channel is used as a guide to reconstruct the red and blue channels, as done in HA [1], GBTF [33], RI [19], MLRI [20], MLRI+wei [21], and ARI [31]. Note that image data will inevitably be contaminated by noise during the collection. Since denoising and demosaicing are both ill-posed problems, in order to reduce the artifacts caused by error accumulation, some studies have proposed to jointly perform demosaicing and denoising. For instance, in [3] the authors propose a joint method based on total variation minimization, while the authors of [39] propose a joint demosaicing and denoising based on the alternating direction method of multipliers (ADMM).

Thanks to the widespread adoption of data-driven deep learning methods in the field of image processing, image demosaicing and denoising have also made new advances. Gharbi et al. designed a joint demosaicing and denoising network (JCNN) using convolutional neural networks [10]. In order to improve the reconstruction on the difficult parts of the images they also established a training dataset. Kokkinos et al. proposed an iterative network (C-RCNN) [23]. Tan et al. proposed the CDM-CNN algorithm using green plane guidance [40], and then Cui et al. extended their work and proposed the CDM-3-Stage algorithm [4]. Tan et al. used residual learning and multi-model fusion to propose an adaptive demosaicing network [38]. Huang et al. proposed a lightweight joint demosaicing and denoising network (LCNN-DD) [16]. Syu et al. compared the effects of convolution kernels of different sizes on the reconstruction and designed a new CFA pattern using a data-driven approach [35]. Mei et al. proposed HighEr-Resolution Network (HERN) in the AIM2019 RAW competition, which can be trained using high-resolution images to learn global information [30]. Ehret et al. proposed to train the network using raw data without ground truth and they found that fine-tuning the network allowed to improve the reconstruction quality [8]. Liu et al. introduced density-map guidance into the convolutional neural networks, and used the edge features of the image to reconstruct the image [27].

3. Residual learning for demosaicing and denoising

According to a scheme proposed by Jin et al. [17], we perform the reconstruction of the CFA image in two stages. First, we design a demosaicing algorithm that combines traditional methods and deep learning to process noise-free images. When applied to a noisy CFA image this network produces a noisy full-color image whose noise statistical properties have been changed after complex interpolation. Therefore, a second convolutional neural network is used to remove this noise.

The noisy CFA model is written as

$$Y = M \ast (X + \epsilon),$$

(1)

where $X$ is a original full color image, $Y$ is the noisy CFA (or mosaicked) image, $\epsilon$ is Gaussian noise with mean 0 and standard deviation $\sigma$, the operator $\cdot \ast$ denotes the array element-wise multiplication and $M$ denotes the CFA mask. The CFA mask $M$ and its inverse mask are defined as

$$M = \begin{bmatrix} M_R \\ M_G \\ M_B \end{bmatrix} \quad \text{and} \quad IM = \begin{bmatrix} 1 - M_R \\ 1 - M_G \\ 1 - M_B \end{bmatrix},$$

(2)
\[ M_R(i, j) = \begin{cases} 0, & \text{if } (i, j) \notin \Omega_R; \\ 1, & \text{if } (i, j) \in \Omega_R, \end{cases} \]

\[ M_G(i, j) = \begin{cases} 0, & \text{if } (i, j) \notin \Omega_G; \\ 1, & \text{if } (i, j) \in \Omega_G, \end{cases} \]

\[ M_B(i, j) = \begin{cases} 0, & \text{if } (i, j) \notin \Omega_B; \\ 1, & \text{if } (i, j) \in \Omega_B, \end{cases} \]

where \( 1(\cdot) \) is an indicator function, \( \Omega \) denotes the set of CFA image pixels, \( \Omega_R, \Omega_G, \Omega_B \subseteq \Omega \) are disjoint sets of pixels, which respectively record red, green and blue values in the CFA image, and satisfy \( \Omega_R \cup \Omega_G \cup \Omega_B = \Omega \).

The first stage considers only the noise-free CFA model:

\[ Y = M \ast X, \quad (3) \]

where \( X \) is a full color image, \( Y \) is the noise-free CFA (or mosaicked) image. We first use GBTF to get a raw demosaiced image \( \hat{X}_{GBTF} = GBTF(Y) \) and a residual \( R_{GBTF} = X - \hat{X}_{GBTF} \). We modify the Inception architecture to achieve better performance in learning the residual and get a estimator \( \hat{R}_{GBTF} \) (see Figure 1). The final full color image is obtained as

\[ \hat{X}_{DM} = IM \ast (\hat{X}_{GBTF} + \hat{R}_{GBTF}) + M \ast X. \quad (4) \]

The first term in the equation above is the demosaiced image estimated by the CNN and evaluated on the inverse CFA mask \( IM \), while the second term are unaltered input CFA samples on the mask \( M \). The resulting CNN is adapted to demosaic noise-free image. So, applying it to a noisy CFA image, produces a noisy demosaiced image.

To handle noisy CFA images, another stage is needed to remove the noise. Given the trained demosaicing as a basic component, we apply it to model (1) and obtain a noisy full color image \( \hat{X}_{DM} \) which can be decomposed as

\[ \hat{X}_{DM} = X + \varepsilon_{DM}. \quad (5) \]

Here, \( \varepsilon_{DM} \) is the residual noise (including artifacts) of the demosaiced image, which is not independent identically distributed (I.I.D.) anymore, and it is difficult to characterize its statistical properties. This is extremely challenging for the traditional denoising models that strongly rely on statistical assumptions, then we use another CNN to learn the residual noise \( \varepsilon_{DM} \) and obtain the estimator \( \hat{\varepsilon}_{DM} \) (see Figure 1). The final full color image is reconstructed as follows

\[ \hat{X}_{DMDN} = \hat{X}_{DM} - \hat{\varepsilon}_{DM}. \quad (6) \]

There are several advantages in training separate demosaicing and denoising networks. First, the noise-free demosaicing focuses on reconstructing the architecture and details in the image without concessions. In addition, the demosaicing network does not need to be adapted to each noise level. Second, the demosaiced result facilitates the task of the denoiser which has to adapt only to the noise and demosaicing artifacts. As we will see later, training a joint denoising and demosaicing with equivalent capacity as the demosaicing and denoising networks indeed leads to lower quality results.

### 3.1. Noise-free demosaicing

The CFA images are different from ordinary images as the values of adjacent pixels represent the intensity of different colors. Many of the existing deep learning algorithms subsample the CFA images to four-channel RGGB images and send them to the network. However, such subsampling operation reduces the image resolution. Therefore, the network needs to perform functions similar to super-resolution, and cannot only focus on image demosaicing. In order to improve this situation, some algorithms use bilinear interpolation as preprocessing. However, the bilinear interpolation cannot reconstruct the image well and this affects the performance of the convolutional network. In this work, we use the gradient based threshold free (GBTF) method [33], which has superior performance compared to the bilinear interpolation.

After the CFA image is preprocessed, we use a convolutional neural network for residual learning. The network architecture is shown in Figure 1. Syu et al. pointed out in their work [35] that convolution kernels of different sizes will affect the reconstruction accuracy. The larger the size of the convolution kernel, the higher the reconstruction accuracy. However, the number of parameters using a \( 5 \times 5 \) convolution kernel is 2.7 times that of using a \( 3 \times 3 \) convolution kernel. Therefore, we still use the \( 3 \times 3 \) convolution kernel, but want to make the network have a larger receptive field. In the image demosaicing task, due to the lack of color information, the full color image reconstruc-
tion must make full use of the correlation of the three RGB channels. Therefore, the degree of cross-channel information fusion determines the performance of the demosaicing algorithm. In order to get a better cross-channel fusion and a larger receptive field, we propose to modify the architecture of GoogleNet Inception-ResNet [36] and adapt the Inception block. The proposed network has 16 Inception blocks. The architecture of the Inception block is shown in Figure 2. In the Inception block, we use $1 \times 1$ convolution kernels to fuse and compress the channels, and use three-way branches to learn different residual features, and finally concatenate the three-way branches. Since the Inception block has good scalability, we have designed a lightweight Inception block, which will be denoted by (-) in what follows. With roughly the same number of parameters as a $3 \times 3$ Conv-BN-ReLU block for 64-layer feature maps, the proposed Inception block increases the network depth (3 non-linearities) and has a larger receptive field ($5 \times 5$). Moreover, the Inception(-) uses about 50% of the parameters of the $3 \times 3$ Conv-BN-ReLU. The parameter comparison data are shown in Table 1.

| Input the number of feature layers | Inception | Inception(-) | Conv-BN-ReLU |
|-----------------------------------|-----------|--------------|--------------|
| First branch                      | 64        | 64           | 64           |
| Second branch                     | 32(1 $\times$ 1) 16(1 $\times$ 1) | 16(1 $\times$ 1) | 3 $\times$ 3 |
| Third branch                      | 32(1 $\times$ 1) 32(3 $\times$ 3) 16(3 $\times$ 3) | 16(1 $\times$ 1) 16(3 $\times$ 3) | |
| Output the number of feature layers | 64        | 64           | 64           |
| Number of parameters              | 39584     | 19584        | 37056        |

Table 1. Inception architecture and number of parameters. The depth of Conv-BN-ReLU is 1, the receptive field is 3, but the depth of the Inception is 3, and receptive field is 5. And Inception(-) has the same properties and the number of parameters is only 52.8% of Conv-BN-ReLU.

### 3.2. Denoising after demosaicing

After obtaining an effective demosaicing algorithm, we continue to use the data-driven convolutional neural network to fit the complex noise that is changed by demosaicing. In order to overcome the limitation that the joint network can only handle a low level of noise, we train the denoising network separately according to the standard deviation of the noise.

For the denoising network, we continue to use the same Inception block architecture as the demosaicing network shown in Figure 1. In the denoising stage, the convolutional neural network is not learning the usual residuals, but learning the noise distribution of the image. We use our proposed denoising algorithm to process the noisy CFA image. And then use the denoising network to remove the noise of the noisy image.

### 3.3. Training procedure

Our approach is a two-stage method and each stage has its loss function. These functions are both evaluated using the classical mean square error (MSE) loss.

In the first stage, the network is trained on a noise-free data set. The loss for the noise-free demosaicing stage is

$$L_{DM} (\Theta_{DM}) = \frac{1}{2N} \sum_{i=1}^{N} \left\| \hat{X}_{DM}^i - X^i \right\|^2,$$  \hspace{0.5cm} (7)
\[
\hat{X}_{DM} = IM \ast \left(\hat{X}_{GBTF}^i + F(\hat{X}_{GBTF}^i; \Theta_{DM})\right) + M \ast \hat{X}^i, \tag{8}
\]

where \( F(\hat{X}_{GBTF}^i; \Theta_{DM}) \) is the output of the demosaicing network to estimate the residual \( R_{GBTF} \) (see (4)).

After the demosaicing network is trained, we apply it to noisy CFA images (see model (1)) to produce a noisy full color images (see model (5)). The goal of this stage is to remove residual noise \( \varepsilon_{DM} \), then the loss for the stage is

\[
\mathcal{L}_{DN}(\Theta_{DN}) = \frac{1}{2N} \sum_{i=1}^{N} \left\| \hat{X}_{DM}^{i} - X^{i} \right\|_2^2, \tag{9}
\]

\[
\hat{X}^{i}_{DM} = \hat{X}_{DM}^i - G(\hat{X}_{DM}^i; \Theta_{DN}), \tag{10}
\]

where \( G(\hat{X}_{DM}^i; \Theta_{DN}) \) is the output of the denoising network, which works as an estimator of \( \varepsilon_{DM} \).

For training the joint demosaicing and denoising, Gharbi et al. provided a dataset of two million \( 128 \times 128 \) images (MIT Dataset) [10]. Ma et al. established the Waterloo Exploration Database (WED) with 4,744 high-quality natural images [29] and Syu et al. provided the Flickr500 with 500 high-quality images [35]. We use these datasets to build our training and test sets. Indeed, 100,000 images are randomly selected from MIT Dataset. And 4653 images in WED and 491 images in Flickr500 are randomly cropped into 100,000 images \( 128 \times 128 \). These 200,000 patches \( 128 \times 128 \) constitute our training set. Furthermore, 91 images in WED and 9 images in Flickr500 compose our test set. During the training time, the patch is flipped and rotated \( 180^\circ \) with a 50\% probability for data augmentation.

For training the denoising model we start by adding Gaussian white noise to the CFA images sampled from the training set (see Table 3 for the standard deviation \( \sigma \) of the noise) and apply the demosaicing network to the noisy CFA images. The color images generated by the demosaicing network at each noise level are then used as noisy training set.

The network architecture was implemented in PaddlePaddle. The network weights are initialized using [13] and the biases are first set to 0. The optimization is performed by the ADAM optimizer [22] using the default parameters. The batch size is set to 64, and the initial learning rate to \( 10^{-2} \). The learning rate decay strategy is the exponential decay method, and the learning rate decayed by 0.9 every 3000 iterations. Our model is trained on a NVIDIA Tesla V100 and each training takes 70 epochs, where the demosaicing algorithm usually takes 4-5 days, and the denoising algorithm for each noise level takes 3-4 days.
Table 3. This table lists the experimental results of demosaicing and denoising on the Kodak dataset, McMaster dataset and our test set. The best value is marked in bold. The noise level that the algorithm cannot handle is indicated by x in the table.

| Algorithm | JCNN | LCNN-DD | ADMM | Ours+CBM3D | Ours |
|-----------|------|---------|------|------------|------|
|            | PSNR/SSIM | PSNR/SSIM | PSNR/SSIM | PSNR/SSIM | PSNR/SSIM |
| Kodak      |                                                |                                                |                                                |                                                |                                                |
| $\sigma$   |                                                |                                                |                                                |                                                |                                                |
| 5          | 35.98/0.9444 | 33.83/0.8700 | 31.71/0.8764 | 36.42/0.9480 | 37.01/0.9518 |
| 10         | 33.10/0.9005 | 28.34/0.6705 | 31.13/0.8574 | 33.27/0.9057 | 33.99/0.9135 |
| 15         | 31.24/0.8587 | 24.96/0.5201 | 30.23/0.8351 | 31.40/0.8682 | 32.16/0.8798 |
| 20         | 29.84/0.8166 | x          | 29.32/0.8115 | 30.09/0.8340 | 30.89/0.8502 |
| 40         | x          | x          | 25.71/0.6793 | 26.87/0.7236 | 28.06/0.7637 |
| 60         | x          | x          | 24.22/0.6255 | 24.80/0.6531 | 26.51/0.7096 |
| McMaster   |                                                |                                                |                                                |                                                |                                                |
| $\sigma$   |                                                |                                                |                                                |                                                |                                                |
| 5          | 35.15/0.9327 | 33.24/0.8586 | 32.37/0.8994 | 35.44/0.9344 | 36.30/0.9412 |
| 10         | 32.93/0.8979 | 28.44/0.6752 | 31.58/0.8739 | 32.78/0.8959 | 33.89/0.9108 |
| 15         | 31.20/0.8579 | 25.30/0.5323 | 30.43/0.8417 | 31.01/0.8570 | 32.25/0.8831 |
| 20         | 29.77/0.8156 | x          | 29.28/0.8085 | 29.65/0.8196 | 31.01/0.8579 |
| 40         | x          | x          | 25.15/0.6661 | 25.92/0.6977 | 28.12/0.7916 |
| 60         | x          | x          | 22.94/0.5966 | 23.36/0.6160 | 26.37/0.7399 |
| WED + Flickr|                                                |                                                |                                                |                                                |                                                |
| $\sigma$   |                                                |                                                |                                                |                                                |                                                |
| 5          | 34.88/0.9450 | 33.02/0.8657 | 31.19/0.9007 | 35.56/0.9488 | 36.29/0.9560 |
| 10         | 32.59/0.9085 | 28.13/0.6758 | 30.63/0.8812 | 32.79/0.9112 | 33.79/0.9284 |
| 15         | 30.86/0.8723 | 24.97/0.5364 | 29.70/0.8584 | 30.96/0.8781 | 32.09/0.9033 |
| 20         | 29.47/0.8358 | x          | 28.72/0.8345 | 29.60/0.8488 | 30.86/0.8808 |
| 40         | x          | x          | 24.93/0.7173 | 25.92/0.7527 | 27.83/0.8086 |
| 60         | x          | x          | 22.84/0.6636 | 23.40/0.6838 | 26.17/0.7594 |

Figure 4. Results of the various comparison between state-of-the-arts and our method for demosaicing and denoising in image 3 of McMaster with noise $\sigma = 10$.

Figure 5. Results of the various comparisons between the state-of-the-arts and our approach for demosaicing and denoising in image 19 of Kodaka with noise $\sigma = 20$.

4. Experiments

4.1. Datasets

We choose the classic Kodak [45] and McMaster [7] datasets for evaluating our algorithm on the demosaicing and denoising task. The Kodak dataset consists of 24 images ($768 \times 512$). The McMaster dataset consists of 18 images ($500 \times 500$), which are cropped from the $2310 \times 1814$ high-resolution images. At the same time, we conduct experiments on our test set, which consists of 100 images from the WED and Flickr500 datasets.

4.2. Quantitative and qualitative comparison

We use the peak signal-to-noise ratio (PSNR) [2] and structural similarity (SSIM) [41] to evaluate the performance of the algorithm. After we saved each image in png format of uint8, we clip the image border by 10 pixels to calculate the PSNR value of the image. Compared with
### Table 4. Ablation study. This table shows the effect of using different preprocessing methods and network architectures.

| Method       | Kodak    | McMaster | WED+Flickr |
|--------------|----------|----------|------------|
| PSNR/SSIM    | PSNR/SSIM| PSNR/SSIM|            |
| GBTF+        | 42.19/0.9882 | 39.12/0.9702 | 39.65/0.9815 |
| Conv-BN-ReLU |          |          |            |
| HA+Inception | 42.23/0.9878 | 39.37/0.9714 | 39.82/0.9818 |
| Bil+Inception| 42.50/0.9886 | 39.55/0.9723 | 40.07/0.9827 |
| Ours(-)      | 42.49/0.9888 | 39.25/0.9702 | 39.84/0.9820 |
| Ours         | 42.76/0.9893 | 39.61/0.9725 | 40.22/0.9831 |

### Table 5. Comparison with end-to-end training. This table shows the comparison of the performance of a one-stage joint demosaicking and denoising network (with 32 Inception blocks) and the proposed two-stage demosaicking first (16 Inception blocks) and then denoising (16 Inception blocks). The results are for a noise level of $\sigma = 20$.

| Method     | Kodak    | McMaster | WED+Flickr |
|------------|----------|----------|------------|
| PSNR/SSIM  | PSNR/SSIM| PSNR/SSIM|            |
| One-stage  | 30.12/0.8259 | 30.14/0.8367 | 29.93/0.8560 |
| Two-stage  | 30.89/0.8502 | 31.01/0.8579 | 30.86/0.8808 |

### Table 6. Average running time of demosaicing for 50 images ($500 \times 500$) on a PC with Intel Core i7-9750H 2.60GHz, 16GB memory, and Nvidia GTX-1650 GPU.

| Method      | CPU(s) | GPU(s) |
|-------------|--------|--------|
| GBTF        | 2.76   | -      |
| MLRI+wei    | 1.35   | -      |
| ARI         | 25.58  | -      |
| C-RCNN      | 156.39 | 2.26   |
| JCNN        | 11.78  | 0.22   |
| CDM-CNN     | 1.71   | 0.33   |
| CDM-3-Stage | 7.65   | 0.90   |
| LCNN-DD     | 1.64   | 0.21   |
| Ours(-)     | 4.41   | 0.36   |
| Ours        | 7.46   | 0.51   |

### Table 7. Average running time of demosaicing and denoising for 50 images ($500 \times 500$) with noise level $\sigma = 10$ on a PC with Intel Core i7-9750H 2.60GHz, 16GB memory, and Nvidia GTX-1650 GPU. DM indicates the demosaicing stage of our method, and DN indicates the denoising stage.

| Method      | CPU(s)  | GPU(s)  |
|-------------|---------|---------|
| ADMM        | 472.27  | -       |
| JCNN        | 11.84   | 0.23    |
| LCNN-DD     | 1.64    | 0.21    |
| Ours(DM+DN) | 14.79(7.42+7.37) | 0.92(0.48+0.44) |

Figure 3 illustrates a challenging case in which existing algorithms always produce color distortions (in the necklace part), while the proposed algorithm present no distortions. In order to better observe the reconstruction effect of the algorithm, we show the residual image between the reconstructed image generated by all the algorithms and the ground truth. It can be seen that the visual effect is consistent with the numerical evaluation. Figures 4 and 5 show the comparison of visual effects when the standard deviation is $\sigma = 10, 20$. When the standard deviation is $\sigma = 20$ (Figure 5), we note that the fence is more pleasant and has fewer checkerboard artifacts. At the same time, we find that using the traditional CBM3D algorithm denoising under the premise of a good demosaicking can also obtain a better effect than the deep learning algorithms (such as JCNN).

### 4.3. Ablation study and running time

In order to verify the effectiveness of our proposed method, we conduct some ablation experiments. We trained and compared the following models: (a) Using the GBTF algorithm [33] for preprocessing, while the demosaicking network is built using the classic Conv-BN-ReLU blocks. (b) Using HA algorithm [1] for preprocessing, while the network uses our proposed Inception block. (c) Using bilinear interpolation for preprocessing, while the network uses our proposed Inception block. The performance of the...
above three cases on the three datasets is shown in Table 4. This shows that the using the GBTF algorithm for preprocessing and the architecture of Inception block are more effective for image demosaicing.

In order to verify the necessity of noise-free demosaicing, we train a one-stage joint demosaicing and denoising network. We fixed the noise level of the data to $\sigma = 20$ and trained a network architecture with 32 Inception blocks. This matches the capacity of the two networks of demosaicing and denoising (16 blocks each). The network still uses the GBTF algorithm for preprocessing and residual learning. Table 5 shows the differences between the two strategies. We can see that the joint network is not as effective as the two-stage network. This highlights the importance of training the demosaicing network on noise-free data.

In order to estimate the computational complexity of these algorithms, we test the average time consumed by all algorithms to process 50 images ($500 \times 500$) on a PC with Intel Core i7-9750H 2.60GHz, 16GB memory, and Nvidia GTX-1650 GPU. For the deep learning algorithms, only the actual network processing time is calculated, not including the network loading time. The time consumed by the algorithm in noise-free image demosaicing is shown in Table 6. The time consumed for the demosaicing and denoising of CFA images with noise level $\sigma = 10$ is shown in Table 7. Since our network is composed of independent demosaicing and denoising stages, the time consumed can be calculated separately. In Table 7, DM denotes the demosaicing stage of our algorithm, and DN the denoising stage. It can be seen that the processing time of our algorithm is comparable to the other deep learning algorithms. And it is better than the traditional algorithms that require iteration, such as ARI [31] and ADMM [39].

5. Conclusion

In this paper, we changed the traditional image processing pipeline of first denoising and then demosaicing to first demosaicing and then denoising. Combining the advantages of the traditional algorithms and deep learning, we used the GBTF algorithm for preprocessing and used the convolutional neural network to learn the residuals in the denoising stage. In order to better handle the cross-channel fusion problem, we proposed an Inception block and extended it to a lightweight version. In the denoising stage, we used the same convolutional neural network architecture to learn the noise whose statistical properties are changed by complex interpolation. This solved the difficulties that traditional models have been unable to solve. Experiments on the Kodak, McMaster and WED + Flickr datasets showed clearly that our algorithm has superiority compared with the state-of-the-art demosaicing algorithms and joint demosaicing and denoising algorithms.

6. Acknowledgments

Jin has been supported by the National Natural Science Foundation of China (Grants No. 61661039, Natural Science Fund of Inner Mongolia Autonomous Region (Grant No. 2020MS01002), China Scholarship Council for a one year visiting at Ecole normale supérieure Paris-Saclay (No. 201806810001). Work partly financed by Office of Naval research grant N00014-17-1-2552 and DGA Astrid project n° ANR-17-ASTR-0013-01.

References

[1] James E Adams Jr and John F Hamilton Jr. Adaptive color plane interpolation in single sensor color electronic camera, July 29 1997. US Patent 5,652,621.
[2] D. Alleysson, S. Susstrunk, and J. Herault. Linear demosaicing inspired by the human visual system. IEEE Transactions on Image Processing, 14(4):439–449, 2005.
[3] L. Condat and S. Mosaddegh. Joint demosaicing and denoising by total variation minimization. In 2012 19th IEEE International Conference on Image Processing (ICIP), pages 2781–2784, 2012.
[4] K. Cui, Z. Jin, and E. Steinbach. Color image demosaicing using a 3-stage convolutional neural network structure. In 2018 25th IEEE International Conference on Image Processing (ICIP), pages 2177–2181, 2018.
[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In 2007 IEEE International Conference on Image Processing, volume 1, pages I – 313–I – 316, 2007.
[6] A. Danielyan, M. Vehvilainen, A. Foi, V. Katkovnik, and K. Egiazarian. Cross-color bm3d filtering of noisy raw data. In 2009 International Workshop on Local and Non-Local Approximation in Image Processing, pages 125–129, 2009.
[7] E. Dubois. Frequency-domain methods for denoising of bayer-sampled color images. IEEE Signal Processing Letters, 12(12):847–850, 2005.
[8] T. Ehret, A. Davy, P. Arias, and G. Facciolo. Joint demosaicing and denoising by fine-tuning of bursts of raw images. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 8867–8876, 2019.
[9] F. Fang, J. Li, and T. Zeng. Soft-edge assisted network for single image super-resolution. IEEE Transactions on Image Processing, 29:4656–4668, 2020.
[10] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédéric Durand. Deep joint demosaicing and denoising. ACM Transactions on Graphics (TOG), 35(6):191, 2016.
[11] J. Gu, H. Lu, W. Zuo, and C. Dong. Blind super-resolution with iterative kernel correction. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1604–1613, 2019.
[12] S. Gu, Y. Li, L. Van Gool, and R. Timofte. Self-guided network for fast image denoising. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 2511–2520, 2019.
[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet.
classification. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 1026–1034, 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.

[15] Zhiwei Hong, Xiaoqiang Fan, Tao Jiang, and Jianxing Feng. End-to-end unpaired image denoising with conditional adversarial networks. In Thirty-Third AAAI Conference on Artificial Intelligence, AAAI'20, pages 4140–4149, 2020.

[16] T. Huang, F. F. Wu, W. Dong, G. Shi, and X. Li. Lightweight deep residue learning for joint color image demosaicking and denoising. In 2018 24th International Conference on Pattern Recognition (ICPR), pages 127–132, 2018.

[17] Q. Jin, G. Facciolo, and J. Morel. A review of an old dilemma: Demosaicking first, or denoising first? In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 2169–2179, 2020.

[18] Ossi Kalevo and Henry Rantanen. Noise reduction techniques for bayer-matrix images. In Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications III, volume 4669, pages 348–359, International Society for Optics and Photonics, 2002.

[19] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi. Residual interpolation for color image demosaicking. In 2013 IEEE International Conference on Image Processing (ICIP), pages 2304–2308, 2013.

[20] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Minimized-laplacian residual interpolation for color image demosaicking. In Digital Photography X, volume 9023, page 90230L, International Society for Optics and Photonics, 2014.

[21] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi. Beyond color difference: Residual interpolation for color image demosaicking. IEEE Transactions on Image Processing, 25(3):1288–1300, 2016.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[23] Filippos Kokkinos and Stamatios Lefkimmiatis. Deep image demosaicking using a cascade of convolutional residual denoising networks. In 2018 European Conference on Computer Vision (ECCV), pages 317–333, 2018.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS), pages 1097–1105, 2012.

[25] Min Seok Lee, Sang Wook Park, and Moon Gi Kang. Denoising algorithm for cfa image sensors considering interchannel correlation. Sensors, 17(6):1236, 2017.

[26] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1132–1140, 2017.

[27] L. Liu, X. Jia, J. Liu, and Q. Tian. Joint demosaicing and denoising with self guidance. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2237–2246, 2020.

[28] C. Ma, Y. Rao, Y. Cheng, C. Chen, J. Lu, and J. Zhou. Structure-preserving super resolution with gradient guidance. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 7766–7775, 2020.

[29] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang. Waterloo exploration database: New challenges for image quality assessment models. IEEE Transactions on Image Processing, 26(2):1004–1016, 2017.

[30] K. Mei, J. Li, J. Zhang, H. Wu, J. Li, and R. Huang. Higher-resolution network for image demosaicing and enhancing. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pages 3441–3448, 2019.

[31] Yusuke Monno, Daisuke Kiku, Masayuki Tanaka, and Masatoshi Okutomi. Adaptive residual interpolation for color and multispectral image demosaicking. Sensors, 17(12):2787, 2017.

[32] S. H. Park, H. S. Kim, S. Lansel, M. Parmar, and B. A. Wandell. A case for denoising before demosaicking color filter array data. In 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, pages 860–864, 2009.

[33] I. Pekkuuksen and Y. Altunbasak. Gradient based threshold free color filter array interpolation. In 2010 IEEE International Conference on Image Processing (ICIP), pages 137–140, 2010.

[34] Y. Quan, M. Chen, T. Pang, and H. Ji. Self2self with dropout: Learning self-supervised denoising from single image. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1887–1895, 2020.

[35] Nai-Sheng Syu, Yu-Sheng Chen, and Yung-Yu Chuang. Learning deep convolutional networks for demosaicing. arXiv preprint arXiv:1802.03769, 2018.

[36] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty-First AAAI Conference on Artificial Intelligence, AAAI’17, page 42784284, 2017.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–9, 2015.

[38] D. S. Tan, W. Chen, and K. Hua. Deepdemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks. IEEE Transactions on Image Processing, 27(5):2408–2419, 2018.

[39] H. Tan, X. Zeng, S. Lai, Y. Liu, and M. Zhang. Joint demosaicing and denoising of noisy bayer images with admm. In 2017 IEEE International Conference on Image Processing (ICIP), pages 2951–2955, 2017.

[40] Runjie Tan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Color image demosaicking via deep residual learning. In Proc. IEEE Int. Conf. Multimedia Expo (ICME), pages 793–798, 2017.

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.
[42] Zhijie Wen, Jiawei Guan, Tieyong Zeng, and Ying Li. Residual network with detail perception loss for single image super-resolution. *Computer Vision and Image Understanding*, 199:103007, 2020.

[43] Z. Xia, F. Perazzi, M. Gharbi, K. Sunkavalli, and A. Chakrabarti. Basis prediction networks for effective burst denoising with large kernels. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11841–11850, 2020.

[44] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[45] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, 20(2):023016, 2011.