Deep Spatial Adaptive Network for Real Image Demosaicing

Tao Zhang,[1] Ying Fu,[1*] Cheng Li[2]

[1]Beijing Institute of Technology
[2]Huawei Noah's Ark Lab
{tzhang,fuying}@bit.edu.cn, licheng89@huawei.com

Abstract

Demosaicing is the crucial step in the image processing pipeline and is a highly ill-posed inverse problem. Recently, various deep learning based demosaicing methods have achieved promising performance, but they often design the same nonlinear mapping function for different spatial locations and do not well consider the difference of mosaic pattern for each color. In this paper, we propose a deep spatial adaptive network (SANet) for real image demosaicing, which can adaptively learn the nonlinear mapping function for different locations. The weights of spatial adaptive convolution layer are generated by the pattern information in the receptive filed. Besides, we collect a paired real demosaicing dataset to train and evaluate the deep network, which can make the learned demosaicing network more practical in the real world. The experimental results show that our SANet outperforms the state-of-the-art methods under both comprehensive quantitative metrics and perceptive quality in both noiseless and noisy cases.

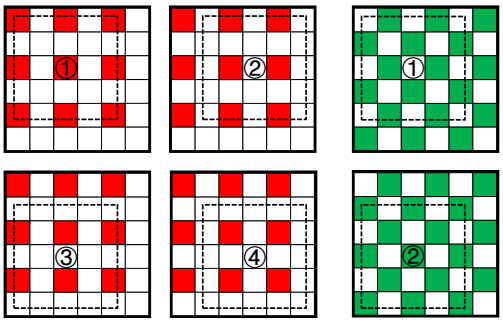


Figure 1: The interpolation filter for different locations in R, G, B channels. Note that the B channel is the same as R channel. We take filter with $5*5$ kernel size as example, and there are 4 types for R and B channels and 2 types for G channel, respectively.

Introduction

To reduce cost, most digital camera captures image through a single CCD/CMOS sensor with color filter array (CFA), e.g., RGGB Bayer pattern, where two-thirds of the information is lost and the rest one-third of the information may be perturbed by different kinds of noise. Modern digital camera employs Image Signal Processing (ISP) pipeline to create high-quality color image from the raw data. The first and most crucial step in the sequence of ISP steps is demosaicing. The recovery errors during the early step of ISP may negatively influence the visual appearance of final result.

Since demosaicing is under-determined, prior knowledge of the natural image is usually utilized to regularize the recovery. The traditional techniques encode the heuristic hand-crafted priors into local filter and interpolate the mosaic image (Cok 1987; Laroche 1994; Malvar, He, and Cutler 2004; Buades et al. 2009). These local filters are adaptively in terms of the local CFA information and/or image content. Besides, optimization approaches iteratively recovery color image by embedding hand-crafted prior into optimization, such as nonlocal prior (Heide et al. 2014). However, the hand-crafted priors are insufficient to represent the variety of the real-world noisy data, and some challenging high frequency regions appear some visually disturbing artifacts such as checkerboard patterns, zippering around edges, and moire (Gharbi et al. 2016).

Different from traditional methods which rely on hand-crafted priors, deep learning methods (Tan et al. 2017b; Tan, Chen, and Hua 2018; Gharbi et al. 2016; Kokkinos and Lefkimmiatis 2018; Liu et al. 2020; Chen, Wen, and Chan 2021) employ convolution neural network (CNN) to implicitly learn the prior from training dataset. Some methods (Gharbi et al. 2016; Liu et al. 2020; Chen, Wen, and Chan 2021) decompose a mosaic image with Bayer pattern into four-channel RGGB image and feed it to a CNN, which is similar to image super-resolution (Song et al. 2020; Pan et al. 2020). The other methods (Tan et al. 2017b; Tan, Chen, and Hua 2018; Kokkinos and Lefkimmiatis 2018) directly input the mosaic image to a CNN. All these methods employ the same nonlinear mapping function for different spatial locations. Nevertheless, as shown in Figure 1, different locations need different interpolation filters, *i.e.*, different mapping functions.

In addition, deep learning methods rely heavily on training dataset. There several datasets are utilized mostly, and they can be grouped into three categories. The first kind of

*Corresponding author
Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

dataset (Gharbi et al. 2016; Timofte et al. 2018, 2017) contains sRGB images, which have been nonlinearly processed. Nevertheless, demosaicing always works in linear representation of raw image in the real ISP, Besides, the sRGB image is demosaiced by existing algorithm, which may introduce undesirable artifacts. The second kind of dataset (Khashabi et al. 2014) contains linear RGB images downsampled from raw mosaic image, but it may change the structure of the signal. The third kind of dataset (Qian et al. 2019) contains linear full color images captured by camera with advanced pixel shift technique. Moreover, all these datasets only have clean RGB images, and the mosaic images are synthesized with CFA and gaussian noise. The synthetic data has domain gap to real raw data, and may limit the practice of trained demosaicing methods in the real world.

In this paper, we present a deep spatial adaptive network (SANet) to adaptively learn the mapping function for different spatial locations in the mosaic image, depicted in Figure 2. The architecture of SANet is based on UNet (Ronneberger, Fischer, and Brox 2015), and involves proposed spatial adaptive convolution layer and residual learning. In each spatial adaptive convolution layer, the kernel weights are generated by the pattern information in the receptive field. To ease the training of deep network, we further introduce residual learning to SANet, including global residual learning and local residual learning. Besides, we capture a real demosaicing dataset by the camera with advanced pixel shift technique under both noiseless and noisy cases. The captured dataset contains paired raw mosaic and raw RGB images, which makes the trained network more practical in the real world. The experimental results show that our SANet outperforms the state-of-the-art methods under both comprehensive quantitative metrics and perceptive quality in both noiseless and noisy cases.

In summary, our main contributions are that

- We propose a deep spatial adaptive network for real image demosaicing, which is learned on our captured paired real demosaicing dataset and can adaptively learn the mapping function for different spatial locations.
- We design a spatial adaptive convolution layer to replace the conventional convolution layer, whose weights are generated for each spatial location by the pattern information in the receptive filed.
- We capture a real demosaicing dataset with paired raw mosaic and RGB images, which makes the trained network more practical under both noiseless and noisy cases in the real world.

Related Work

In this section, we review the most relevant studies on image demosaicing, and spatial adaptive network.

Image Demosaicing

Image demosaicing aims to recover a full color image from a sub-sampled mosaic image with potential noise. Since image demosaicing is a highly ill-posed problem, prior knowledge of the natural image is utilized to regularize the recovery. Traditional interpolation-based methods (Cok 1987; Laroche 1994; Malvar, He, and Cutler 2004; Buades et al. 2009) encode the heuristic hand-crafted priors into local filter and interpolate the mosaic image. At the early stage, the local filter is designed to interpolate R, G and B channels separately. Later, to exploit the correlation between different color channels, various priors have been proposed to model inter-channel correlation, such as integrated gradient (Pekkucuksen and Altunbasak 2010), sparsity (Mairal, Elad, and Sapiro 2007; Yu, Sapiro, and Mallat 2011), self-similarity (Zhang and Wu 2005; Mairal et al. 2009) and residual interpolation (Kiku et al. 2016; Monno et al. 2017). However, the interpolation-based methods cannot handle noise in the mosaic image. The optimization-based methods (Heide et al. 2014; Tan et al. 2017a) embed hand-crafted priors into an optimization algorithm and iteratively recover the full color image from noisy mosaic image. Heide *et al.* (2014) proposed a primal dual optimization method with nonlocal prior. Tan *et al.* (2017a) integrated various hand-crafted priors, e.g., total variation prior and nonlocal prior, into alternating direction method of multipliers (ADMM) for image demosaicing.

Recently, deep learning methods (Tan et al. 2017b; Tan, Chen, and Hua 2018; Gharbi et al. 2016; Kokkinos and Lefkimmiatis 2018; Liu et al. 2020) have been proposed to automatically learn the desired prior for image demosaicing. Gharbi *et al.* (2016) proposed a deep convolution network to cover full color image from noisy mosaic image. Tan *et al.* (2017b) and Tan *et al.* (2018) first initially covered the full color image via bilinear interpolation, and then employed a CNN-based method to enhance the initialized result. Kokkinos *et al.* (2018) unfolded the majorization-minimization algorithm with a residual denoising network for image demosaicing. Liu *et al.* (2020) proposed a self-guidance network for image demosaicing by introducing green channel guidance and density map guidance.

The traditional methods employ hand-crafted priors, which often only model the linear characteristic and are insufficient to exploit the nonlinearity in natural image. The deep learning methods learn the same nonlinear prior for different locations in the mosaic image, but do not well consider the pattern information. In this work, we present an efficient CNN-based method for image demosaicing to learn spatial adaptive prior according to pattern information.

Adaptive Network

The naive CNN-based methods only employ convolution layer, nonlinear activation layer and/or normalization layer to model the nonlinear mapping function between input and output information. Once the network has been trained, the network parameters and the nonlinear mapping function are fixed. Recently, more and more researchers focus on developing adaptive network, which can learn the adaptive mapping function according to image features and/or extra input information.

The most well-known category of adaptive network is that with various attention mechanisms, such as nonlocal attention (Wang et al. 2018), channel attention (Hu, Shen, and Sun 2018) and so on. The attention mechanism adaptively calculates the correlation between different input informa-

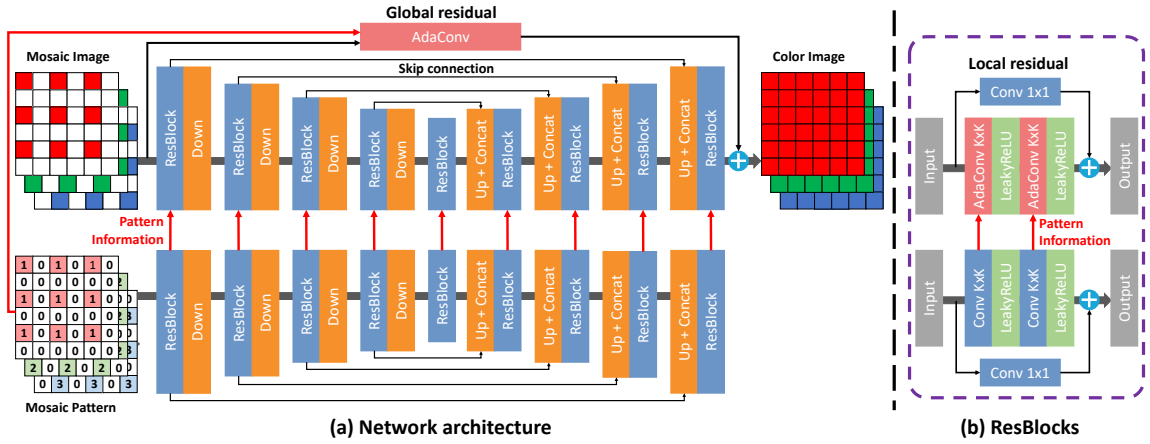


Figure 2: The overview of the proposed image demosaicing network SANet, employing Unet as the basic architecture. To ease the training, we introduce global and local residual learning into SANet, and employ residual block as fundamental block. In each residual block, we replace the conventional convolution layer by our proposed spatial adaptive convolution layer, whose weights are generated for each spatial location by the pattern information in the receptive filed through feeding the mosaic pattern into the same network architecture. The details of residual blocks and corresponding pattern information flow are indicated in the purple dotted box and shown in the right.

tion. Besides, deformable convolution (Dai et al. 2017) employs irregularly-shaped filers. Dynamic convolution (Chen et al. 2020) linearly combines a set of kernels in a convolution layer. Adaptive activation (Kligvasser, Shaham, and Michaeli 2018) and normalization (Huang and Belongie 2017) generate the parameters of these layers according to extra input information.

Our spatial adaptive network is more related to kernel prediction network (Bako et al. 2017) and hypernetwork (Ha, Dai, and Le 2016). The kernel prediction network generates spatial-varying kernels according to input images, meanwhile the hypernetwork produces spatial-consistent kernels according to extra input information. In this work, we propose a spatial adaptive convolution layer with spatial-varying kernels, whose parameters are generated according to CFA pattern information.

Spatial Adaptive Network for Image Demosaicing

In this section, we first formulate the problem for image demosaicing with noise, and describe the motivation of our method. Then, we introduce the spatial adaptive convolution, whose weights are generated by the pattern information in the receptive field. Finally, we describe the overall network architecture of SANet, which can adaptively learn the mapping function for each spatial location.

Formulation and Motivation

The aim of demosaicing is to recovery full color RGB image $\mathbf{X} \in \mathbb{R}^{3\times H\times W}$ from mosaic image $\mathbf{Y} \in \mathbb{R}^{1\times H\times W}$, where H and W are the number of height and width for the mosaic and RGB images. $\mathbf{N} \in \mathbb{R}^{1\times H\times W}$ denotes the additive noise. The relationship of mosaic and RGB images is generally linear and can be represented as

$$\mathbf{Y} = \mathcal{M}(\mathbf{X}) + \mathbf{N}, \tag{1}$$

where $\mathcal{M}$ is the mosaic mapping function.

To recover the full color RGB image, traditional methods (Cok 1987; Laroche 1994; Malvar, He, and Cutler 2004; Buades et al. 2009) employ various linear local filters regularized by heuristic hand-crafted priors to adaptively interpolate the mosaic image in different locations, while deep learning methods (Tan et al. 2017b; Tan, Chen, and Hua 2018; Gharbi et al. 2016; Kokkinos and Lefkimmiatis 2018) directly learn nonlinear mapping function with CNN to recover RGB image without considering the pattern information in each location of mosaic image. In this work, we present a spatial adaptive convolution layer to learn the adaptive nonlinear mapping function in terms of pattern information. Concretely, we design a spatial adaptive convolution layer, whose weights in each location are generated by the pattern information in the same receptive field. Figure 2 and Figure 3 show the proposed spatial adaptive network and spatial convolution layer, respectively.

Spatial Adaptive Convolution

Before describing spatial adaptive convolution, we first review the conventional convolution. Let $\mathbf{F}^I \in \mathbb{R}^{C_i\times H\times W}$ denote the input feature map, where C_i is the input channels. A set of C_o convolution kernels with size of $K \times K$ is denoted as $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{C_o\times C_i\times K\times K}$, where each kernel $\boldsymbol{\mathcal{K}}_o \in \mathbb{R}^{C_i\times K\times K}$ $(o = 1, 2, \cdots, C_o)$ is consist of C_i convolution filters $\boldsymbol{\mathcal{K}}_{o,i} \in \mathbb{R}^{K\times K}$ $(i = 1, 2, \cdots, C_i)$. Then, these filters are employed to operate the input feature map in a sliding window way to generate the output feature map $\mathbf{F}^O \in \mathbb{R}^{C_o\times H\times W}$, which can be expressed as

$$\mathbf{F}_o^O[p] = \sum_{i=1}^{C_i} \sum_{q\in\mathcal{N}(p)} \boldsymbol{\mathcal{K}}_{o,i}[p-q]\mathbf{F}_i^I[q], \tag{2}$$

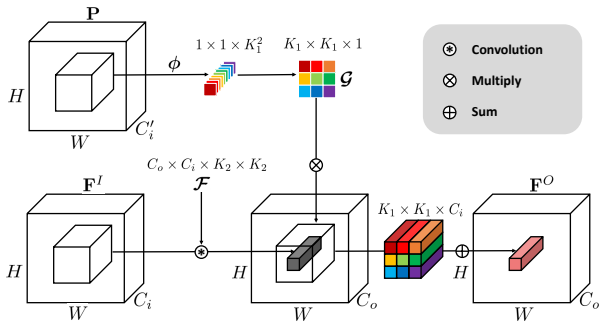


Figure 3: The spatial adaptive convolution. We decompose large spatial adaptive kernel $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{C_o \times C_i \times H \times W \times K \times K}$ into $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{H \times W \times K_1 \times K_1}$ and $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{C_o \times C_i \times K_2 \times K_2}$, and we set $K_1 + K_2 - 1 = K$ to keep the receptive field. The weights of $\boldsymbol{\mathcal{G}}$ for each feature location are generated by pattern information in the receptive field with the function $\boldsymbol{\phi}$.

where p denotes the spatial location for convenient representation, and $\mathcal{N}(p)$ is the neighboring pixels of p.

The conventional convolution shares kernel weights and learns a consistent mapping function for each spatial location. However, different locations in the mosaic image with different pattern information need different mapping functions. The spatial adaptive convolution can break the kernel spatial sharing property of conventional convolution, and each spatial location is operated by an independent kernel. The convolution kernel of spatial adaptive convolution is denoted as $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{C_o \times C_i \times H \times W \times K \times K}$, and the corresponding convolution operation can be represented as

$$\mathbf{F}_o^O[p] = \sum_{i=1}^{C_i} \sum_{q \in \mathcal{N}(p)} \boldsymbol{\mathcal{H}}_{o,i,p}[p-q]\mathbf{F}_i^I[q]. \tag{3}$$

Nevertheless, spatial adaptive convolution in this form occupies a very large memory, which is $H \times W$ times larger than conventional convolution and $C_o \times K \times K$ or $C_i \times K \times K$ times larger than input or output feature maps. The large memory occupation limits the application of spatial adaptive convolution in this form.

Previous researches show that a large convolution kernel can be decomposed into two small convolution kernels and keep the same receptive field (Szegedy et al. 2016). The kernel decomposition not only reduces computational cost, but also reduces the memory occupation. Here, we decompose the large kernel of spatial adaptive convolution $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{C_o \times C_i \times H \times W \times K \times K}$ into $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{H \times W \times K_1 \times K_1}$ and $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{C_o \times C_i \times K_2 \times K_2}$, as shown in Figure 3. $\boldsymbol{\mathcal{G}}$ focuses on extracting spatial adaptive correlation and shares the kernel between different channels. $\boldsymbol{\mathcal{F}}$ focuses on extracting the inter-channel relationship and shares kernel between different spatial locations. We set $K_1 + K_2 - 1 = K$ to keep the same receptive field. After decomposition, the memory occupation is $\dfrac{H \times W \times K_1 \times K_1 + C_o \times C_i \times K_2 \times K_2}{C_o \times C_i \times H \times W \times K \times K}$ times than undecomposed one and $\dfrac{H \times W \times K_1 \times K_1 + C_o \times C_i \times K_2 \times K_2}{C_o \times C_i \times K \times K}$ times than conventional convolution. Taking $C_o = C_i = 32$, $H = W = 64$ and $K_1 = K_2 = 3$ as an example, the memory occupation is almost 0.00037 times than the undecomposed one and 1.8 times than conventional convolution. The operation can be expressed as

$$\begin{aligned}\mathbf{F}_o^O[p] = &\sum_{q_1 \in \mathcal{N}_1(p)} \boldsymbol{\mathcal{G}}_p[p-q_1] \\ &\left(\sum_{i=1}^{C_i} \sum_{q_2 \in \mathcal{N}_2(p)} \boldsymbol{\mathcal{F}}_{o,i}[p-q_2]\mathbf{F}_i^I[q_2]\right)[q_1],\end{aligned} \tag{4}$$

where $\mathcal{N}_1(p)$ and $\mathcal{N}_2(p)$ are the neighboring pixels of p for $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{F}}$, respectively.

Different from conventional convolution with a fixed weight, the kernel weights of our proposed spatial adaptive convolution is generated by pattern information in the receptive field, as shown in Figure 3. $\boldsymbol{\mathcal{F}}$ is a spatial consistent kernel, and we only need to generate spatial adaptive weights for $\boldsymbol{\mathcal{G}}$. We symbolize the weight generation function as $\boldsymbol{\phi}$, which can be represented as

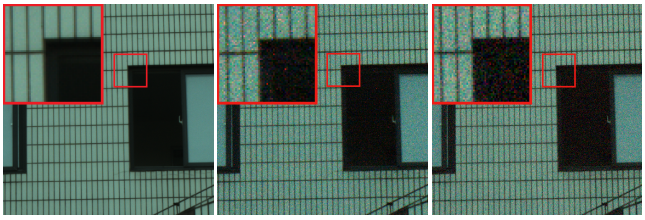
$$\boldsymbol{\mathcal{G}}_p = \boldsymbol{\phi}(\mathbf{P}_{q \in \mathcal{N}(p)}), \tag{5}$$

where $\mathbf{P}$ denotes the pattern information, and q is the index in the receptive field $\mathcal{N}(p)$. Specifically, we utilize a $K \times K$ conventional convolution to extract the pattern information in the combined receptive field of $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$. Then, we feed the extracted information through LeakyReLU activation function and fully-connection layer to generate weights in a K_1^2 vector for each location. Finally, we reshape the vector into $K_1 \times K_1$ as the kernel of $\boldsymbol{\mathcal{G}}$.

Network Architecture

The architecture of SANet is illustrated in Figure 2. The overall structure is based on typical Unet (Ronneberger, Fischer, and Brox 2015) architecture. SANet consists of 4 encoder stages and 4 corresponding decoder stages. At the end of each encoder stage, the feature maps are downsampled to $1/2\times$ scale with a 4×4 kernel size and 2 stride convolution. Before each decoder stage, the feature maps are upsampled to $2\times$ scale with bilinear interpolation. Skip connections pass large-scale low-level feature maps from each encoder stage to its corresponding decoder stage. To ease the training, we introduce residual learning into SANet, including global residual and local residual learning. For global residual learning, we employ a spatial adaptive convolution to initially recovery the RGB image from input mosaic image and is added by the global residual of Unet output. For local residual learning, we utilize residual block as the fundamental block to build encoder and decoder. The residual block is conducted by two $K \times K$ convolutions followed by LeakyReLU activation function and a 1×1 convolution, which learns the local residual.

The spatial adaptive convolution replace the conventional convolution in each residual block. In the spatial adaptive convolution, the wights for each location are generated by the pattern information in the receptive field. To guarantee the receptive field on input mosaic image is the same as that on pattern information, we input the pattern information



Clean data Real noisy data Synthetic noisy data

Figure 4: The difference between real and synthetic noisy data. Real data contains signal-dependent and signal-independent noises (Wei et al. 2020), but synthetic data is only synthesized by signal-independent noise with gaussian distribution.

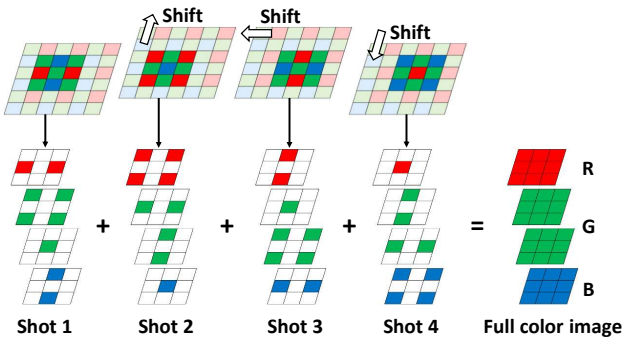


Figure 5: The illustrate of pixel shift technology. In each capturing, the camera sensor is physically moved in horizontal and vertical dimension and takes four shots, and integrating these mosaic images can get a full color image.

to the same architecture except with small feature maps, as shown in the bottom of Figure 2(a).

Generally, the mosaic pattern is represented as a $\{0, 1\}$ binary mask. It only indicates whether this location has information or not. To identify the color information for each location, we replace the $\{0, 1\}$ mask by $\{0, 1, 2, 3\}$ mask, which denote no information, R information, G information and B information, respectively.

The network is trained with paired mosaic and RGB images, and we use L_1 distance between mosaic image and demosaiced RGB images as the loss function, which can be expressed as

$$\mathcal{L}(\theta) = ||\mathbf{X} - f(\mathbf{Y}; \theta)||_1, \tag{6}$$

where f and θ denote SANet and the corresponding parameters, respectively.

Paired Real Image Demosaicing Dataset

It is well-known that the powerful deep learning methods rely on training dataset. The existing datasets for demosaicing network training have several problems. The sRGB datasets (Gharbi et al. 2016; Timofte et al. 2018, 2017) are demosaiced by existing demosaicing algorithm and lie in nonlinear representation, which introduces undesirable artifacts and does not match the linear workspace of demosaicing algorithms, respectively. The linear RGB datasets

Figure 6: Paired real data capture setup.

(Khashabi et al. 2014) downsample the raw mosaic image, which changes the structure of signal. Recently, Qian *et al.* (Qian et al. 2019) capture the full color RGB dataset by advanced pixel shift camera. Nevertheless, all these datasets only contain RGB image, and the mosaic image is synthesized with CFA and gaussian noise. The noise difference between real and synthetic data is shown in Figure 4. There has domain gap between synthetic data and real data, which limits the practice of trained demosaicing algorithms.

To support the research, we employ a pixel shift camera to capture a real paired mosaic and full color RGB images dataset. For each full color RGB image capturing, pixel shift camera physically controls the camera sensor to horizontally or vertically move one pixel four times, and takes one mosaic image at each movement, as shown in Figure 5. After four times capturing, the color information of each pixel can be fully captured. Then, we fix the camera setting and turn the capturing mode from pixel shift to normal to capture a corresponding mosaic image. Besides, according to the work in (Chen et al. 2018), we reduce the exposure time to capture noisy mosaic image. Therefore, we can capture paired real noiseless/noisy mosaic and clean full color RGB images.

To capture the dataset, we employ a Sony A7R4 digital camera with pixel shift technology, as shown in Figure 6. The camera is mounted on sturdy tripods. We adjust camera settings such as aperture, focus and exposure time to maximize the quality of the full color RGB image for each scene. Then, we employ a remote control software to turn the capturing mode from pixel shift to normal. When capturing noisy mosaic image, we further set a shorter exposure time. Finally, we collect the noiseless/noisy mosaic image. Since we capture multiply images for one scene, all scenes in the dataset are static. Our dataset contains 100 indoor and outdoor scenes with 9568×6376 resolution. We will continuously expand our dataset and capture more data being suitable for image demosaicing in future.

The captured paired real demosaicing dataset can support deep learning methods to be more practical under noiseless and noisy cases in the real world.

Experiments

In this section, we first introduce the settings in our experiments, including implementation details and metrics for quantitative evaluation. Then, our method is compared with several state-of-the-art methods on our captured real demosaicing dataset under both noiseless and noisy cases. Finally,

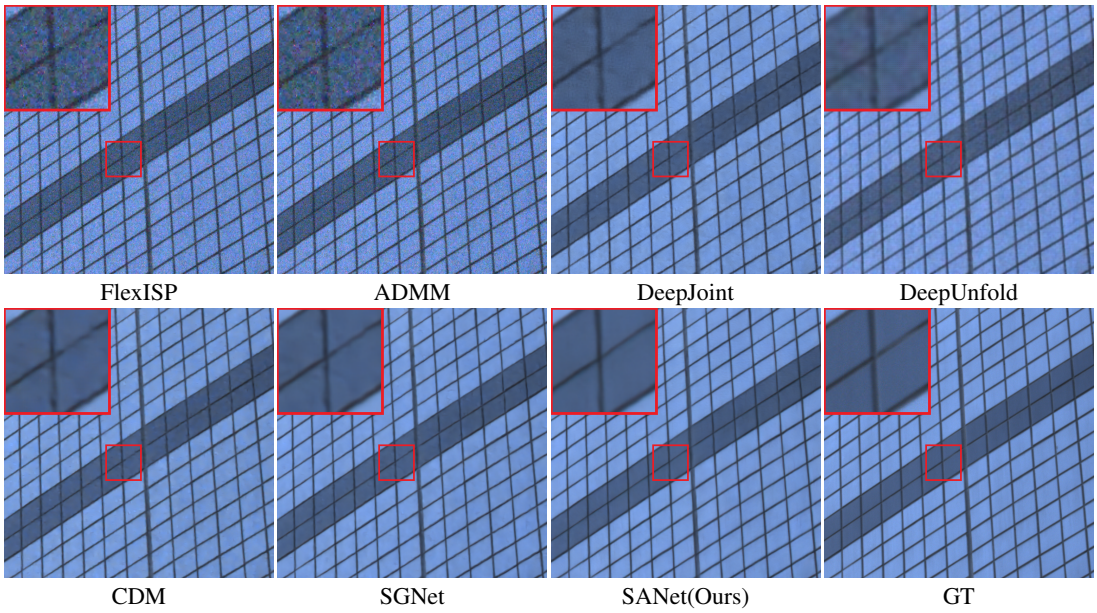


Figure 7: Visual quality comparison on a typical scene for image demosaicing in our real demosaicing dataset. The results recovered by different methods and the groundtruth image are shown from left to right and from top to bottom.

Cases	Metrics	Methods						
		FlexISP	ADMM	DeepJoint	DeepUnfold	CDM	SGNet	SANet(Ours)
Noiseless	PSNR	39.580	39.443	51.089	51.082	51.124	51.771	**52.012**
	SSIM	0.9641	0.9643	0.9923	0.9860	0.9942	0.9947	**0.9951**
Noisy	PSNR	30.246	30.078	41.186	41.503	41.540	42.298	**42.576**
	SSIM	0.9149	0.9120	0.9832	0.9811	0.9822	0.9829	**0.9833**

Table 1: Quantitative results of different methods on our real demosaicing dataset. The best results are highlighted in bold.

Methods	Params(M)	FLOPs(G)
DeepJoint	0.56	9.39
DeepUnfold	0.38	245.60
CDM	0.27	17.44
SGNet	13.62	221.69
Ours	10.78	19.18

Table 2: Efficiency comparison of deep learning methods.

we discuss the effect of different network modules.

Settings

The proposed architecture requires no pre-training and is trained in an end-to-end manner. The kernel size K is set to be 5, and the decomposed kernel size K_1 and K_2 are set to be 3 and 3 for all spatial adaptive convolution, respectively.

In the training stage, we randomly crop overlapped 256×256 spatial regions from image in our paired real demosaicing dataset. Our implementation is based on PyTorch (Paszke et al. 2019). The models are trained with Adam optimizer (Kingma and Ba 2014) ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) for 100 epochs. The initial learning rate and mini-batch size are set to 1×10^{-4} and 1, respectively.

We employ two evaluation metrics to evaluate the performance of all methods, including the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). The larger PSNR and SSIM indicate better performance.

Evaluation on Real Image Demosaicing Dataset

We compare our SANet with six state-of-the-art methods, including two traditional methods, *i.e.*, FlexISP (Heide et al. 2014) and ADMM (Tan et al. 2017a), and four deep learning methods, *i.e.*, DeepJoint (Gharbi et al. 2016), DeepUnfold (Kokkinos and Lefkimmiatis 2018), CDM (Tan et al. 2017b) and SGNet (Liu et al. 2020). We evaluate all methods in both noiseless and noisy cases on our captured real image demosaicing dataset. Note that we do not employ noise map for all deep learning methods, for noiseless data do not need noise map and noisy data do not know the accurate noise level.

Table 1 provides the averaged recovery results of noiseless and noisy cases on our real image demosaicing dataset, to quantitatively compare our SANet with FlexISP, ADMM, DeepJoint, DeepUnfold, CDM and SGNet. The best results are highlighted in bold for each metric. It can be seen that deep learning methods always have better performance than

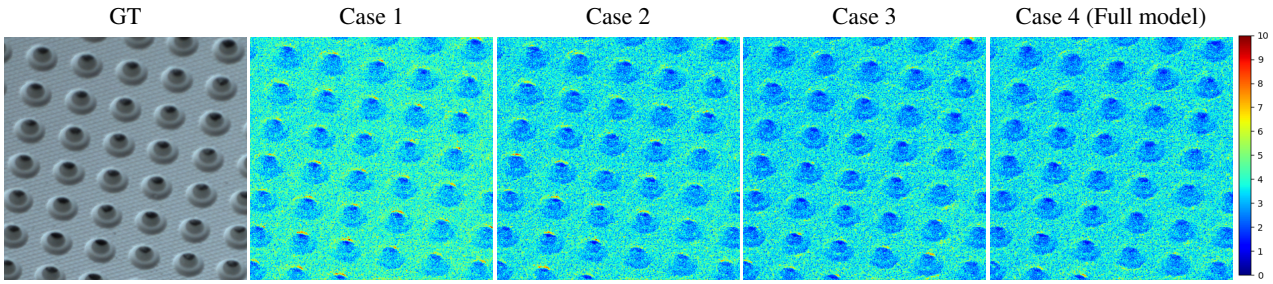


Figure 8: Visual quality comparison of our network with different modules.

traditional methods, which demonstrates the advantage of deeply exploiting the intrinsic characteristic of nature image. Comparing the results with different deep learning methods in the same case, our method outperforms the existing methods in both PSNR and SSIM metrics. This reveals the effectiveness of our spatial adaptively learned nonlinear mapping function.

We further quantitatively evaluate the efficiency of all deep learning methods by parameters and floating point operations (FLOPs) metrics, and the results are provided in Table 2. Note that the FLOPs is calculated by recovering a 256×256 resolution image. It can be seen that the number of parameters of our method and SGNet are larger than that of other methods, which indicates that our method and SGNet have stronger capacity to model the intrinsic characteristic of nature image. The FLOPs of our method is similar to DeepJoint and CDM, and is an order of magnitude smaller than DeepUnfold and SGNet. It reveals the efficiency of our method.

To visualize the experimental results, a representative recovered result for noisy case is shown in Figure 7. The recovered results of FlexISP/ADMM/DeepJoint/DeepUnfold/CDM/SGNet/our methods and ground truth are shown from left to right and from top to bottom. The results of FlexISP and ADMM still contain noise, and indicates the handcrated prior is insufficient for image demosacing in the real world. The recovered result from our SANet is more accurate than the results from compared methods, which demonstrates the effectiveness of our method.

Modules	Cases			
	1	2	3	4
Unet	✓	✓	✓	✓
spatial AdaConv		✓	✓	✓
local residual			✓	✓
global residual				✓
PSNR	50.715	51.473	51.941	**52.012**
SSIM	0.9936	0.9948	0.9950	**0.9951**
Params(M)	10.02	10.43	10.78	10.78
FLOPs(G)	17.48	18.45	19.14	19.18

Table 3: Quantitative results of our network with different modules. The best results are highlighted in bold.

Ablation Study

To investigate the effectiveness and efficiency of spatial adaptive convolution, local residual learning and global residual learning, we conduct an ablation study on our real image demosaicing dataset with noiseless case. The results are provided in Table 3. It can be seen that all modules contribute to the performance improving, which verifies the effectiveness of spatial adaptive convolution, local and global residual learning. Comparing the parameters and FLOPs of our method with different modules, we can see that each module only sightly improve the computational cost. It verifies the efficiency of spatial adaptive convolution, local and global residual learning.

A visual comparison of our method with different modules is provided in Figure 8. The error maps are the average absolution errors between ground truth and recovered results across channels. It can be seen that our method different modules all recover the image well and are similar to the ground truth. Our method with spatial adaptive convolution and residual learning can further improve the recovery accuracy.

Conclusion

In this paper, we propose a novel spatial adaptive network for image demosaicing, which consists of a serial of spatial adaptive convolution considering the pattern information for each location. The proposed method can adaptively learn the nonlinear mapping function for each location in the mosaic image. Besides, we collect a real paired mosaic and full color RGB images dataset by pixel shift camera under both noiseless and noisy cases, which makes the trained network more practical in the real world. Experimental results show that the proposed SANet outperforms current state-of-the-art methods under both comprehensive quantitative metrics and perceptive quality. In the future, we will further consider the content information to generate weights of spatial adaptive convolution and expand our real demosaicing dataset to support image demosaicing in the real world.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants No. 62171038, No. 61827901 and No. 62088101.

References

Bako, S.; Vogels, T.; McWilliams, B.; Meyer, M.; Novák, J.; Harvill, A.; Sen, P.; Derose, T.; and Rousselle, F. 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. on Graphics*, 36(4): 97–1.

Buades, A.; Coll, B.; Morel, J.-M.; and Sbert, C. 2009. Self-similarity driven color demosaicking. *IEEE Trans. Image Processing*, 18(6): 1192–1202.

Chen, C.; Chen, Q.; Xu, J.; and Koltun, V. 2018. Learning to see in the dark. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 3291–3300.

Chen, J.; Wen, S.; and Chan, S.-H. G. 2021. Joint Demosaicking and Denoising in the Wild: The Case of Training Under Ground Truth Uncertainty. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*, 2, 1018–1026.

Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; and Liu, Z. 2020. Dynamic convolution: Attention over convolution kernels. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 11030–11039.

Cok, D. R. 1987. Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. US Patent 4,642,678.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proc. of International Conference on Computer Vision (ICCV)*, 764–773.

Gharbi, M.; Chaurasia, G.; Paris, S.; and Durand, F. 2016. Deep joint demosaicking and denoising. *ACM Trans. on Graphics*, 35(6): 1–12.

Ha, D.; Dai, A.; and Le, Q. V. 2016. Hypernetworks. *Proc. of International Conference on Learning representations(ICLR)*.

Heide, F.; Steinberger, M.; Tsai, Y.-T.; Rouf, M.; Pajak, D.; Reddy, D.; Gallo, O.; Liu, J.; Heidrich, W.; Egiazarian, K.; et al. 2014. Flexisp: A flexible camera image processing framework. *ACM Trans. on Graphics*, 33(6): 1–13.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 7132–7141.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. of International Conference on Computer Vision (ICCV)*, 1501–1510.

Khashabi, D.; Nowozin, S.; Jancsary, J.; and Fitzgibbon, A. W. 2014. Joint demosaicing and denoising via learned nonparametric random fields. *IEEE Trans. Image Processing*, 23(12): 4968–4981.

Kiku, D.; Monno, Y.; Tanaka, M.; and Okutomi, M. 2016. Beyond color difference: Residual interpolation for color image demosaicking. *IEEE Trans. Image Processing*, 25(3): 1288–1300.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kligvasser, I.; Shaham, T. R.; and Michaeli, T. 2018. xunit: Learning a spatial activation function for efficient image restoration. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2433–2442.

Kokkinos, F.; and Lefkimmiatis, S. 2018. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *Proc. of European Conference on Computer Vision (ECCV)*, 303–319.

Laroche, C. 1994. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. *United States Patent, no. 5373322*.

Liu, L.; Jia, X.; Liu, J.; and Tian, Q. 2020. Joint demosaicing and denoising with self guidance. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2240–2249.

Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G.; and Zisserman, A. 2009. Non-local sparse models for image restoration. In *Proc. of International Conference on Computer Vision (ICCV)*, 2272–2279.

Mairal, J.; Elad, M.; and Sapiro, G. 2007. Sparse representation for color image restoration. *IEEE Trans. Image Processing*, 17(1): 53–69.

Malvar, H. S.; He, L.-w.; and Cutler, R. 2004. High-quality linear interpolation for demosaicing of Bayer-patterned color images. In *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, III–485.

Monno, Y.; Kiku, D.; Tanaka, M.; and Okutomi, M. 2017. Adaptive residual interpolation for color and multispectral image demosaicking. *Sensors*, 17(12): 2787.

Pan, J.; Liu, Y.; Sun, D.; Ren, J.; Cheng, M.-M.; Yang, J.; and Tang, J. 2020. Image formation model guided deep image super-resolution. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*, 07, 11807–11814.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Proc. of Conference on Neural Information Processing Systems (NeurIPS)*, 8024–8035.

Pekkucuksen, I.; and Altunbasak, Y. 2010. Gradient based threshold free color filter array interpolation. In *Proc. of International Conference on Image Processing (ICIP)*, 137–140.

Qian, G.; Gu, J.; Ren, J. S.; Dong, C.; Zhao, F.; and Lin, J. 2019. Trinity of pixel enhancement: a joint solution for demosaicking, denoising and super-resolution. *arXiv preprint arXiv:1905.02538*, 6: 8.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proc. of International Conference on Medical image computing and computer-assisted intervention*, 234–241.

Song, D.; Xu, C.; Jia, X.; Chen, Y.; Xu, C.; and Wang, Y. 2020. Efficient residual dense block search for image super-resolution. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*, 07, 12007–12014.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826.

Tan, D. S.; Chen, W.-Y.; and Hua, K.-L. 2018. DeepDemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks. *IEEE Trans. Image Processing*, 27(5): 2408–2419.

Tan, H.; Zeng, X.; Lai, S.; Liu, Y.; and Zhang, M. 2017a. Joint demosaicing and denoising of noisy bayer images with ADMM. In *Proc. of International Conference on Image Processing (ICIP)*, 2951–2955.

Tan, R.; Zhang, K.; Zuo, W.; and Zhang, L. 2017b. Color image demosaicking via deep residual learning. In *Proc. of International Conference Multimedia and Expo (ICME)*, 4, 6.

Timofte, R.; Agustsson, E.; Van Gool, L.; Yang, M.-H.; and Zhang, L. 2017. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proc. of Conference on Computer Vision and Pattern Recognition workshops (CVPRW)*, 114–125.

Timofte, R.; Gu, S.; Wu, J.; and Van Gool, L. 2018. Ntire 2018 challenge on single image super-resolution: Methods and results. In *Proc. of Conference on Computer Vision and Pattern Recognition workshops (CVPRW)*, 852–863.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 7794–7803.

Wei, K.; Fu, Y.; Yang, J.; and Huang, H. 2020. A physics-based noise formation model for extreme low-light raw denoising. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2758–2767.

Yu, G.; Sapiro, G.; and Mallat, S. 2011. Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *IEEE Trans. Image Processing*, 21(5): 2481–2499.

Zhang, L.; and Wu, X. 2005. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Trans. Image Processing*, 14(12): 2167–2178.