Deep Spatial Adaptive Network for Real Image Demosaicing

Tao Zhang,1 Ying Fu,1* Cheng Li2

1Beijing Institute of Technology
2Huawei 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.

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, zipper 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
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 (Ron-neberger, 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. 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 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 recover full color image from noisy mosaic image. Heide 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-
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 recover full color RGB image \( \mathbf{X} \in \mathbb{R}^{1 \times H \times W} \) 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

\[
\mathbf{Y} = \mathcal{M}(\mathbf{X}) + \mathbf{N},
\]

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 \( \mathbf{K} \in \mathbb{R}^{C_o \times C_I \times K \times K} \), where each kernel \( \mathbf{K}^o \in \mathbb{R}^{C_I \times K \times K} \) (\( o = 1, 2, \ldots, C_o \)) consist of \( C_I \) convolution filters \( \mathbf{K}^o_i \in \mathbb{R}^{K \times K} \) (\( i = 1, 2, \ldots, C_I \)). 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

\[
\mathbf{F}^O(p) = \sum_{i=1}^{C_I} \sum_{q \in \mathcal{N}(p)} \mathbf{K}^o_i[p-q] \mathbf{F}^I[q],
\]

where \( \mathcal{N}(p) \) denotes the neighborhood of \( p \).
The operation can be expressed as
\[
F_o[p] = \sum_{q_1 \in N_1(p)} G_{p} \left[ p - q_1 \right] + \left( \sum_{i=1}^{C_1} \sum_{q_2 \in N_2(p)} F_{o,i} \left[ p - q_2 \right] F_{i}^{q_2} \left[ q_2 \right] \right) \left[ q_1 \right],
\]
where \( N_1(p) \) and \( N_2(p) \) are the neighboring pixels of \( p \) for \( G \) and \( F \), 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. \( F \) is a spatial consistent kernel, and we only need to generate spatial adaptive weights for \( G \). We symbolize the weight generation function as \( \phi \), which can be represented as
\[
G_{p} = \phi(P_{q\in N(p)}),
\]
where \( P \) denotes the pattern information, and \( q \) is the index in the receptive field \( N(p) \). Specifically, we utilize a \( K \times K \) conventional convolution to extract the pattern information in the combined receptive field of \( F \) and \( G \). Then, we feed the extracted information through LeakyReLU activation function and fully-connection layer to generate weights in a \( K^2 \times 3 \) vector for each location. Finally, we reshape the vector into \( K_1 \times K_1 \) as the kernel of \( G_o \).

**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 \( 1/2 \) scale with a \( 4 \times 4 \) 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 recover 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 \( 1 \times 1 \) convolution, which learns the local residual.

The spatial adaptive convolution replace the conventional convolution in each residual block. In the spatial adaptive convolution, the weights 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
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

\[
L(\theta) = ||X - f(Y; \theta)||_1, \tag{6}
\]

where \(f\) and \(\theta\) 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 (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 \(9568 \times 6376\) 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,
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 $256 \times 256$ 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 $1 \times 10^{-4}$ 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 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 traditional methods.

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

<table>
<thead>
<tr>
<th>Cases</th>
<th>Metrics</th>
<th>FlexISP</th>
<th>ADMM</th>
<th>DeepJoint</th>
<th>DeepUnfold</th>
<th>CDM</th>
<th>SGNet</th>
<th>SANet(Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>39.580</td>
<td>39.443</td>
<td>51.089</td>
<td>51.082</td>
<td>51.124</td>
<td>51.771</td>
<td><strong>52.012</strong></td>
</tr>
<tr>
<td>Noiseless</td>
<td>SSIM</td>
<td>0.9641</td>
<td>0.9643</td>
<td>0.9923</td>
<td>0.9860</td>
<td>0.9942</td>
<td>0.9947</td>
<td><strong>0.9951</strong></td>
</tr>
<tr>
<td></td>
<td>PSNR</td>
<td>30.246</td>
<td>30.078</td>
<td>41.186</td>
<td>41.503</td>
<td>41.540</td>
<td>42.298</td>
<td><strong>42.576</strong></td>
</tr>
<tr>
<td>Noisy</td>
<td>SSIM</td>
<td>0.9149</td>
<td>0.9120</td>
<td>0.9832</td>
<td>0.9811</td>
<td>0.9822</td>
<td>0.9829</td>
<td><strong>0.9833</strong></td>
</tr>
</tbody>
</table>

### Table 2: Efficiency comparison of deep learning methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Params(M)</th>
<th>FLOPs(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepJoint</td>
<td>0.56</td>
<td>9.39</td>
</tr>
<tr>
<td>DeepUnfold</td>
<td>0.38</td>
<td>245.60</td>
</tr>
<tr>
<td>CDM</td>
<td>0.27</td>
<td>17.44</td>
</tr>
<tr>
<td>SGNet</td>
<td>13.62</td>
<td>221.69</td>
</tr>
<tr>
<td>Ours</td>
<td>10.78</td>
<td>19.18</td>
</tr>
</tbody>
</table>
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 $256 \times 256$ 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 handcrafted prior is insufficient for image demosaicing 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.

<table>
<thead>
<tr>
<th>Modules</th>
<th>Cases</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unet</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>spatial AdaConv</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>local residual</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>global residual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>PSNR</td>
<td></td>
<td>50.715</td>
<td>51.473</td>
<td>51.941</td>
<td>52.012</td>
</tr>
<tr>
<td>SSIM</td>
<td></td>
<td>0.9936</td>
<td>0.9948</td>
<td>0.9950</td>
<td>0.9951</td>
</tr>
<tr>
<td>Params(M)</td>
<td></td>
<td>10.02</td>
<td>10.43</td>
<td>10.78</td>
<td>10.78</td>
</tr>
<tr>
<td>FLOPs(G)</td>
<td></td>
<td>17.48</td>
<td>18.45</td>
<td>19.14</td>
<td>19.18</td>
</tr>
</tbody>
</table>

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 slightly 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 absolute 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 perceptual 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


