The Visual Computer (2023) 39:6571–6587
https://doi.org/10.1007/s00371-022-02749-y

ORIGINAL ARTICLE

Recursive lightweight convolutional neural networks that make noisy images purer and purer

Jiayi Shi[1] · Taiyong Li[1] · Jiaxuan Xu[1]

Accepted: 21 November 2022 / Published online: 27 December 2022

Abstract

Convolutional neural network (CNN) has shown its superpower in image denoising in recent years. However, most CNN models suffer from a large number of model parameters and the effect of image denoising still needs to be improved. To cope with these issues, we propose a recursive lightweight CNN approach that can make the noisy images purer and purer, namely PPNets, in this paper. The PPNets mainly consist of four parts: separable convolution–batch normalization–ReLU (SCBR) blocks to extract coarse features, bottlenecks with skip connection to integrate coarse features and refined features to enhance expression ability of model, noise proposal network with an attention mechanism to predict noise level and recursive strategies to stack the denoising model to make the noisy images purer and purer. Since SCBR uses depthwise convolution and pointwise convolution to replace traditional convolution operations, the proposed PPNets have fewer weight parameters. We conduct extensive experiments on two gray image datasets and three color image datasets. The experimental results demonstrate that the PPNets are significantly superior to the traditional models in denoising effectiveness. At the same time, the PPNets outperform the compared state-of-the-art CNN models in terms of both denoising effectiveness and the number of model parameters.

Keywords Image denoising · Convolutional neural networks · Lightweight networks · Recursive networks · Depthwise separable convolution · Pointwise convolution

1 Introduction

In real-world applications, a clean image x is easily contaminated by noise v during production, transmission and/or storage and it will result in a noisy image y. This procedure can be simply formulated as $y = x + v$, where v generally denotes additive Gaussian noise with a standard deviation σ. Image denoising aims at removing v from y to restore the clean image x. It is a typical inverse problem in computer vision [1]. Many researchers have devoted themselves to solving it with different techniques.

Image denoising techniques can be roughly divided into two groups: traditional models and deep learning models. Most traditional models usually apply handcrafted filters to spatial domain and/or transform domain of noisy images [2]. In spatial domain denoising, filters are directly applied to the original pixel matrix. In this way, the filters can be classified as either local filters or non-local filters according to whether they only filter the pixels within a certain distance or the pixels in the entire image. Some pioneering local filters include Gaussian filter, weighted median filter, rank filter, Weiner filter, bilateral filter and so on [3–5]. The non-local mean (NLM) filter and its variants are popular non-local filters for image denoising [6–8]. In contrast, transform domain denoising targets the nonzero coefficients transformed from the original image. Discrete cosine transform (DCT), fast Fourier transform (FFT) and discrete wavelet transform (DWT) are among the most widely used transformation methods [2,9]. Another denoising model is based on sparse representation and dictionary learning, whose basic idea is that a signal (image patch) can be represented by a linear combination of basis from a redundant dictionary and most of the weights of the basis in the combinations are zeros [10–12]. For example, Dabov et al. used an enhanced sparse representation in transform domain for image denoising, and the proposed block-matching and 3D filtering (BM3D) achieved good

✉ Taiyong Li
litaiyong@gmail.com

[1] School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu 611130, China

denoising performance [13]. Zhang and Desrosiers combined gradient histogram with sparse representation to denoise while preserving the texture [14]. Liu et al. presented a group sparsity mixture model to first learn the prior of the image patch group and then applied it to the whole image for denoising [15]. Besides, denoising with total variation (TV) has shown its promising performance [16–18].

Although the traditional models have made significant progress in image denoising, most of them have at least two defects: subjectivity of filter design and complex optimization algorithms. The first defect makes it challenging to design universal filters for different types of noisy images. In other words, well-designed filters on an image set may perform poorly on another set. The second defect makes image denoising very time-consuming, limiting applications in practical denoising problems [1,19].

In recent years, deep learning techniques, especially convolutional neural networks (CNN), have achieved promising results in computer vision tasks including denoising. It is due to the powerful representation capabilities of CNN. The pioneering work of using CNN for image denoising was proposed by Zhang et al., which combined residual learning and batch normalization to handle blind Gaussian noise [1]. To speed up denoising and enhance the flexibility to handle spatially variant noise, Zhang et al. presented a fast and flexible CNN model (FFDNet) that had been proven to be effective and efficient for image denoising [20]. Another approach to fast image denoising is attention-guided scaling [21]. To improve denoising performance, Tian et al. introduced batch renormalization and attention mechanism into CNN models [19]. Several image denoising models are derived from the well-known U-Nets [22–25]. Among them, Jia et al. put forward a cascading U-Nets architecture with multi-scale dense processing to focus on the connections between U-Nets for image denoising [25]. A multistage progressive denoising network that decomposed the denoising task into some subtasks to progressively remove noise has been proven effective for image denoising [26]. One challenging problem of the CNN-based image restoration model is that the performance might degrade with increasing depth. To solve this problem, scholars have tried different ideas like deeply recursive CNN, skip connection, residual learning and dense connection [27–29]. The above CNN-based work mainly used real-valued CNN for denoising. Very recent studies showed that complex-valued CNN could also achieve competitive denoising effects [30,31]. Despite CNN's success in the field of image denoising, it still faces the following issues: (1) The denoising effect still has some room for improvement; (2) vanishing gradients may appear especially when the depth of CNN model grows; (3) it lacks fusion of features from different layers; and (4) the massive amount of model parameters may result in difficulty of applications in mobile devices.

Inspired by the above analysis, this paper aims to propose a novel CNN-based denoising network that can make a noisy image purer and purer, namely PPNets. The key concepts of PPNets include SCBR, bottleneck, noise proposal network (NPN), block and recursive model. Specifically, SCBR refers to as separable Conv-BN-ReLU (CBR), in which depthwise convolution and pointwise convolution are used to replace standard convolution to reduce the number of model parameters. Bottleneck stacks three SCBR components and introduces skip connection to avoid gradient vanishing. NPN denotes a noise proposal network, which can predict noise levels and noise values. Block represents a complete denoising model composed of CBR, several bottlenecks, skip connection, long–short connection and NPN. A recursive model means that the block can be recursively applied for denoising. The more the blocks applied, the purer the images will be produced. With the concepts, the proposed PPNets achieve good denoising effects with lightweight denoising models. The novelty of the proposed PPNets lies in (1) dividing a complex denoising task into several recursive simple subtasks, and (2) a noise proposal network with an attention mechanism is proposed to improve the denoising performance. Both design strategies make the proposed PPNets effective and lightweight.

The PPNets have the following key contributions:

(1) SCBR with depthwise convolution and pointwise convolution is introduced into denoising. It has strong representation learning ability and produces fewer model parameters to result in lightweight denoising models. Three SCBRs are stacked as a bottleneck.
(2) Skip connection is used to avoid gradient vanishing and fuse features from different layers. Long–short connection enhances the effect of shallow layers' features on deep layers. NPN is proposed to predict noise levels and noise values.
(3) A block containing a CBR, several bottlenecks and NPN is a complete denoising model. And they can be recursively applied to make noisy images purer and purer.
(4) Experiments reveal that the proposed PPNets significantly outperform the state-of-the-art compared models in terms of both denoising effect and model parameters.

The rest of the paper is organized as follows: Sect. 2 introduces related work of CNN-based image denoising. The proposed PPNets are presented in detail in Sect. 3. The experimental settings and results are analyzed and discussed in Sect. 4. Finally, we conclude the paper and provide recommendations for future work in Sect. 5.

2 Related work

2.1 Convolutional neural networks for image denoising

CNNs have shown great power in the community of computer vision, such as image restoration, image segmentation, image classification and object detection. Burger et al. [32] applied multilayer perceptron (MLP) to image denoising and achieved comparable results as BM3D. Jain and Seung [33] introduced a convolutional neural network to denoise image and the results show that CNN can achieve similar or better performance than the Markov random field model. The authors in [1] proposed a deep convolutional neural network model (DnCNN) that combines residual learning, batch normalization and ReLU to remove the noises from contaminated images. Generally, There are two ways to boost the denoising performance of neural networks. First, it is shown that the depth of network is of vital importance with the success of AlexNet [34], GoogleNet [35], ResNet [36] and so on. Therefore, adding the width and/or depth of the neural networks can help to obtain better results. Tai et al. proposed a very deep persistent memory network (MemNet) to explicitly fuse the shallow features and deep features [37]. Tian et al. [38] combined two networks to increase the width of the network. Second, enlarging the receptive fields is also a working methods. Chen et al. integrated dilated convolution and several convolutional networks in a feed-forward way, where dilated convolution helps to capture more useful information [39]. An attention-guided denoising convolutional neural network (ADNet) was introduced and it could be expressed as a feed-forward neural network by introducing dilated convolution and attention mechanism [19]. Some other works also demonstrate the effectiveness of the attention mechanism on denoising [40,41]. An image restoration CNN (IRCNN) used multiple dilated convolution with different dilation factors to recover the corrupted image [42]. Besides, a prior-guided dynamic tunable network was proposed to handle handling much more complicated real noise existed in images [43]. These approaches have demonstrated the effectiveness of CNN for image denoising.

2.2 Separable convolution

Due to the huge amount of parameters, deep CNNs may suffer from high computation costs. A direct way to cope with this issue is to reduce the amounts of CNN model parameters to improve the efficiency of the model. The separable convolution was presented for this purpose [44]. MobileNets are such examples that used separable convolutions to reduce the model complexity while still maintaining comparable results [45]. In addition to this, the separable convolutions can significantly improve the quality of semantic segmentation [46,47], image recognition [48,49], change detection [50], image super-resolution [51], etc.

Generally, separable convolution consists of two steps: depthwise convolution that applies a single filter to the each channel of input and pointwise convolution that uses 1 × 1 convolution to integrate the outputs of depthwise convolution.

Traditional standard convolution operation takes an $h_{\text{in}} \times w_{\text{in}} \times m$ feature map F_{in} as input and outputs an $h_{\text{out}} \times w_{\text{out}} \times n$ feature map F_{out}, where h_{in} and w_{in} are the height and width of F_{in}, m is the number of input channels, h_{out} and w_{out} are the height and width of the output feature map F_{out}, and n is the number of output channels. The computational cost (cc_1) and parameter size (ps_1) of standard convolution with kernel size of k can be expressed as Eqs. (1) and (2), respectively:

$$cc_1 = k \cdot k \cdot m \cdot n \cdot h_{\text{out}} \cdot w_{\text{out}}, \tag{1}$$

$$ps_1 = k \cdot k \cdot m \cdot n. \tag{2}$$

It can be seen that the computational cost and the parameter size mainly depend on m, n, h_{out} and w_{out}. As a comparison, the computational cost (cc_2) and parameter size (ps_2) of separable convolution with kernel size of k can be expressed as Eqs. (3) and (4), respectively:

$$cc_2 = k \cdot k \cdot m \cdot h_{\text{out}} \cdot w_{\text{out}} + m \cdot n \cdot \ h_{\text{out}} \cdot w_{\text{out}}, \tag{3}$$

$$ps_2 = k \cdot k \cdot m + m \cdot n. \tag{4}$$

From Eqs. (1) to (4), it can be easily found that both the computational cost and the parameter size of separable convolution are much less than those of standard convolution. Therefore, separable convolution has the potential to improve the efficiency of denoising.

2.3 Skip connection

Increasing the depth of CNN can help to mine more features for image denoising. However, with the growth of depth, the CNNs appear to performance degradation and may suffer from gradient vanishing [52]. To address these issues, He et al. introduced skip connection to allow the model to learn information from shallow layers, and the results show that the residual network is easier to train and performance can be improved even if the depth increases [36]. Further, Mao et al. proposed very deep residual encoder–decoder networks (RED-Net) that composed of skip connection between convolution and deconvolution operations to remove the noises/corruptions [53]. Tian et al. introduced residual learning and skip connection to obtain clean images [38]. These studies show that skip connection enables model to learn different level features and thus is helpful for image denoising.

3 Proposed PPNets

In this section, we propose the denoising CNN model, namely PPNets, in detail. Generally, a CNN model for image denoising includes two aspects:

(1) Model performance. In order to improve the effect of model, we use SCBR as basic components of network. SCBR is inspired by Convolution–BN–ReLU combination while standard convolutions are replaced with separable convolutions, which significantly reduces the model complexity with little performance degradation. The skip connection is adopted to transmit information from shallow features to deep features and reactivate "dead" neurons caused by ReLU. Moreover, a novel noise proposal network (NPN) is proposed to detect and quantify noise from refined extracted features.

(2) Model learning. We use residual learning and recursion to boost model performance, and combine them with batch normalization to accelerate model convergence.

3.1 Flowchart of PPNets

The proposed PPNets are built on one or more Blocks. A block is a complete denoising unit of the proposed PPNets, in which a series of strategies will be designed or be introduced to improve denoising performance. The details of the block and its corresponding components will be described in the following subsections in this section.

Several blocks can also be stacked for recursive denoising models. Suppose the maximal number of blocks is 3, we name the PPNets with 1, 2 or 3 Blocks PPNet-1, PPNet-2 or PPNet-3, respectively. Figure 1 details the PPNets based on Blocks.

It can be seen that the contaminated image was denoised recursively by the stacked blocks. Specifically, PPNet-3 stacks blocks three times to obtain the final model architecture. And when it starts denoising, PPNet-3's Block-2 will take the Block-1's output (roughly denoised image) as input, the Block-2 will output a cleaner image to the Block-3, and so on. It is obviously that the output of the Block-1 is cleaner than raw input, the Block-2's output is much cleaner than the Block-1's output, and the output of Block-3 is the final denoising result, which achieves the best denoising performance. Note that all the PPNets of the final model do not share parameters with each other as they will handle different noise level input.

3.2 SCBR: separable convolution–batch normalization–ReLU

It is well known that the convolution–batch normalization–ReLU (CBR) combination has shown great power in image processing, such as image classification, object detection and semantic segmentation. Typically, standard convolutions can be split into depthwise convolution and pointwise convolution, i.e., separable convolutions, which significantly reduce the model computational cost and memory consumption without noticeable performance degradation. Inspired by these, we choose Separable Conv-BN-ReLU (SCBR) as PPNet's basic component and set all the depthwise convolution filter size to 3 × 3 and pointwise convolution filter size to 1 × 1. An SCBR is illustrated in Fig. 2.

The channel's numbers of feature maps are closely related to the performance of model. Generally speaking, the larger the channel's numbers, the better the effectiveness of the model. However, we simply set 64 as the number of channels of feature maps in consideration of model performance and model complexity. Specifically, each feature map before depthwise convolution and standard convolution of the intermediate layers are padded zeros to ensure the output feature map has the same size of input image. The description of SCBR can be shown in Eq. (5):

$$O_{\mathrm{SCBR}} = f_{\mathrm{relu}}(f_{\mathrm{bn}}(f_{\mathrm{pconv}}(f_{\mathrm{dconv}}(f_{\mathrm{pad}}(I_{\mathrm{in}}))))), \tag{5}$$

where O_{SCBR} and I_{in} denote the output and input of this process, respectively, and f_{pad}, f_{dconv}, f_{pconv}, f_{bn} and f_{relu} represent zero padding, depthwise convolution, pointwise convolution, batch normalization and ReLU activation, respectively.

3.3 Bottleneck

As we all know, deep neural networks always suffer from gradients vanishing/exploding, performance degradation problems and so on. ResNet proposes skip connection and residual learning to solve these problems [36]. Skip connection directly enables both information transmission from shallow layers to deep layers in network forward propagation and gradients transferring from deep layers to shallow layers in network backward propagation. Meanwhile, skip connection can alleviate neurons "death" problem caused by ReLU function to some extent. The bottleneck stacks 3 SCBRs linearly to extract more expressive features, and integrates shallow features with deep features via skip connection to obtain robust features. The structure of bottleneck is illustrated in Fig. 3, where the long arrow denotes the skip connection.

The procedure of a bottleneck can be formulated as Eq. (6):

$$O_{\mathrm{bottleneck}} = f_{\mathrm{SCBR}}(f_{\mathrm{SCBR}}(f_{\mathrm{SCBR}}(I_{\mathrm{in}}))) + I_{\mathrm{in}}, \tag{6}$$

where $O_{\mathrm{bottleneck}}$ and I_{in} denote the output and input of this procedure, respectively, f_{SCBR} denotes the function of SCBR in Sect. 3.2 and "+" denotes the addition operation for elementwise.

3.4 Block, the complete structure of PPNet

3.4.1 Structure

For input noisy images, the denoising block in this paper is composed of one CBR, five bottlenecks and one noise proposal network (NPN), as shown in Fig. 4. In this figure, the NPN (will be described in detail in Sect. 3.4.2) is identified by a red dashed box. If we consider one CBR or SCBR as 1 layer, then each Bottleneck has 3 layers and one Block has 18 layers in total (1 CBR, 15 SCBRs from the five Bottlenecks and 2 layers in NPN), which can effectively balance the large receptive field size and the complexity of the model. Again, we use another skip connection in block (so-called long–short connection, shown as the second-longest arrow in Fig. 4) to combine the coarse features with refined features to enhance the model expression ability. Since the NPN predicts the noise values in the noisy images, we use the noisy images minus the predicted noise values and can obtain the denoised images, i.e., the "outputs" in the Block.

According to the functions of the components in the Block, it can be further divided into three parts: extract layer, refine layer and proposal layer.

The extract layer uses CBR to extract coarse features. Assuming that I_{in} and O_{ext} represent the input noisy image and the output of extract layer, respectively, the extract layer can be formulated as Eq. (7):

$$O_{\text{ext}} = f_{\text{ext}}(I_{\text{in}}), \tag{7}$$

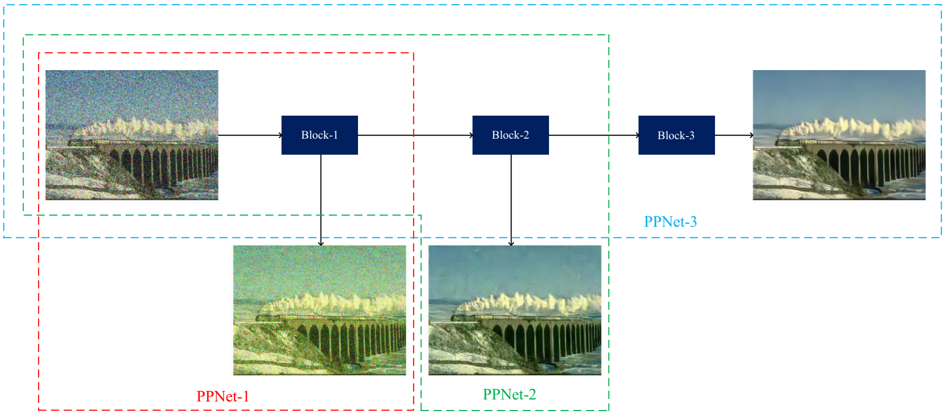
where f_{ext} denotes the function of extract layer.

The following refine layer consists of five Bottlenecks. It takes the coarse features as input and makes full use of skip connection to integrate coarse features and refined features to enhance model expression ability. The refine layer is shown as Eq. (8):

$$O_{\text{ref}} = f_{\text{ref}}(O_{\text{ext}}) + O_{\text{ext}}, \tag{8}$$

where f_{ref} denotes the function of refine layer, O_{ref} denotes the output of refine layer and "+" denotes elementwise addition operation.

Fig. 1 Proposed PPNets (PPNet-1, PPNet-2 and PPNet-3). PPNet-1 takes a raw noisy image as input and outputs a relatively cleaner image; PPNet-2 takes the previous output as input and outputs a much cleaner image; and PPNet-3 receives a very low noise level image and further removes noises



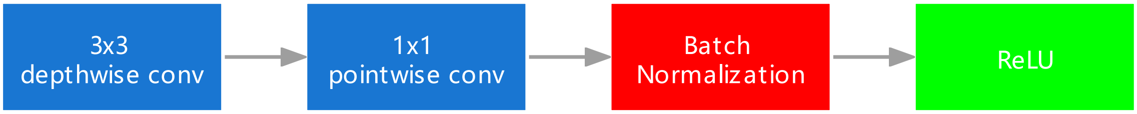


Fig. 2 SCBR: Separable convolution–batch normalization–ReLU. Separable conv denotes splitting standard convolution into depthwise convolution and pointwise convolution

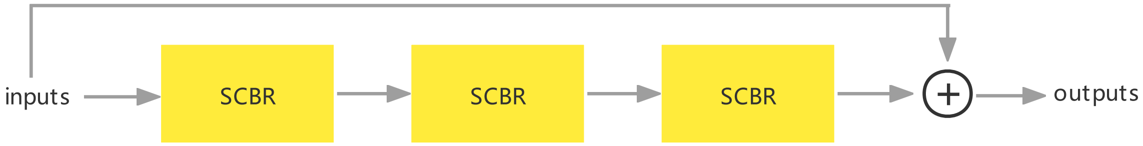


Fig. 3 Structure of a bottleneck, which consists of three stacked SCBR with skip connection

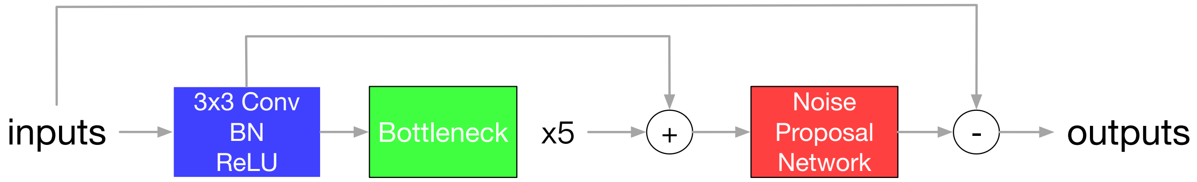


Fig. 4 Structure of a block

It is noted that complex background from the given image or video might be more easier to hide features, which increases the difficulty of extracting key features in the training process. To address this issue, the proposal layer uses NPN to detect the position of noises and the level of noises simultaneously, as formulated by Eq. (9):

$$\begin{cases} O_{\text{mer}} = Conv(O_{\text{ref}}) \\ O_{\text{pro}} = f_{\text{nrp}}(O_{\text{mer}}) * f_{\text{nlp}}(O_{\text{mer}}), \\ \quad\;\; = f_{\text{pro}}(O_{\text{ref}}) \end{cases} \tag{9}$$

where O_{pro} stands for the output of proposal layer.

Finally, we adopt residual learning to generate the model prediction. This process can be formulated as Eq. (10):

$$\begin{aligned} O_{res} &= I_{\text{in}} - O_{\text{pro}} \\ &= I_{\text{in}} - f_{\text{pro}}(f_{\text{ext}}(I_{\text{in}}) + f_{\text{ref}}(f_{\text{ext}}(I_{\text{in}}))) \\ &= f_{\text{ppnet}}(I_{\text{in}}). \end{aligned} \tag{10}$$

3.4.2 Noise proposal network

Noise proposal network (NPN) takes highly integrated coarse and refined features as input and outputs the noise value at each position of input noisy image. To predict the noise values, we use a small fully convolutional network to process the feature map generated by the refine layer. Specifically, NPN applies standard 3 × 3 convolution on the result of refine layer to merge the different expressive ability's features and output a lower-dimensional feature map (1-d for gray images and 3-d for color images). Then this feature map is fed into two sibling standard 1 × 1 convolutional layers (one for rough noise values prediction and the other for noise level prediction). In a word, this network is simply implemented with a 3 × 3 convolutional layer followed by two 1 × 1 convolutional layers, as shown in Fig. 5.

Noise rough prediction. It is difficult to predict the exact noise values directly, especially for the input image with only a little noise or the very end stage of recursive learning in Sect. 3.5. In these cases, the noise values of most regions in the input image will be zero and convolutional layers are hard to predict them. Therefore, we first generate a set of coarse noise values and then the coarse noise values will be handled by noise level prediction.

Noise level prediction. At the same time, a noise level representation feature map will be generated by another 1 × 1 convolution on the shared feature map with the activate function tanh. It is worth mentioning that although both Sigmoid and Softmax can be adopted as activate functions for noise level prediction, Sigmoid function always suffers from saturation problem and Softmax function tends to predict the similar result for the same class. Therefore, we simply add tanh function to convert the features into nonlinearity in the proposed PPNet.

Noise value generation. After obtaining the output of the noise rough prediction and the noise level prediction, we can generate the final prediction of the noise value by Eq. (11):

$$O_{\text{npn}} = O_{\text{nrp}} * O_{\text{nlp}}, \tag{11}$$

where O_{nrp} and O_{nlp} are the result of noise rough prediction and noise level prediction, respectively, O_{npn} denotes the ultimate result of noise proposal networks and $*$ denotes the elementwise multiply operation.

3.5 Recursive model

Nowadays, denoising models are becoming more and more complex in order to obtain better results. However, investigating the denoising results, we can see that whether the model is complex or not, the denoising performance on lower noise level images is usually better than on higher noise level images. In other words, removing noises from relatively cleaner images is much easier for denoising models. An intuition is that dividing the denoising process into several steps may help to improve the denoising performance. Instead of obtaining noise removal images directly, we change our goal to predict relatively clean images. By applying this procedure recursively, the denoised image in the previous step can be taken as input and continue to be denoised. Finally, we can obtain better denoising performance as the input is much cleaner than raw noisy image except for the first step. Followed by this intuition, we simply stack the same network architecture several times while every network owns different parameters. Assume f_{ppnet} stands for the denoising function mentioned in Eq. (10) and f_{ppnetx} represents the xth time of

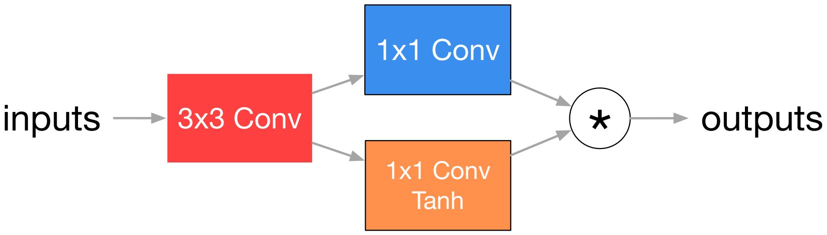


Fig. 5 Noise proposal network

denoising, the formulation of recursive denoising model can be described as Algorithm 1.

Algorithm 1 Recursive denoising

```
Input: raw noisy image $I_{\text{in}}$, target noise level $\sigma$, recursive round $N$ and
    plain denoising method $f_{\text{ppnet}}$.
Output: denoised image $O_{res}$
function RD($I_{\text{in}}, \sigma, N$)
    $O_{res} \leftarrow I_{\text{in}}$
    for $i = 1 \rightarrow N$ do
        $O_{res} = f^{i}_{\text{ppnet}}(O_{res})$
        if noise level of $O_{res} == \sigma$ then
            Break
        end if
    end for
    return $O_{res}$
end function
```

Suppose I_{in} is an image with 80% noises, and the ideal goal of denoising is to obtain an image with 0% noises. It is obviously difficult for models to remove the noises from 80% to a very low level straightly. Hence, our proposed recursive model performs denoising task indirectly. That is, f^{1}_{ppnet} takes 80% noises image as input and may output an image with 60% noises, and f^{2}_{ppnet} takes 60% noises image as input and may output an image with 30% noises, and so on. At last, f^{n}_{ppnet} may take an image with 10% noises as input and remove 10% noises from this image and then can obtain a clean image. In this example, a one-step denoising task that removes 80% noises directly is divided into several subtasks of denoising and each removes some-level noises only.

Due to the simplicity of a single block, we can stack several blocks linearly to compose the final model without increasing too much complexity.

3.6 Loss function

Discriminative denoising models such as DnCNN [1] and ADNet [19] adopt residual learning to train the model. That is, given a noisy input $y = x + n$, their training objective is residual mapping n. Following this idea, given the training dataset $\{x^{i}, y^{i}_{t}\}^{D}_{i=1}$, where x^{i} and y^{i}_{t} denote the $i-$th input noisy image and the corresponding clean image, respectively, the objective of PPNet is to optimize the model parameters θ to predict the residual image $n = y - x$ as accurately as possible. Specifically, the mean squared error (MSE) is adopted as the loss function to train the denoising model. This procedure can be described as Algorithm 2.

Algorithm 2 Non-recursive loss

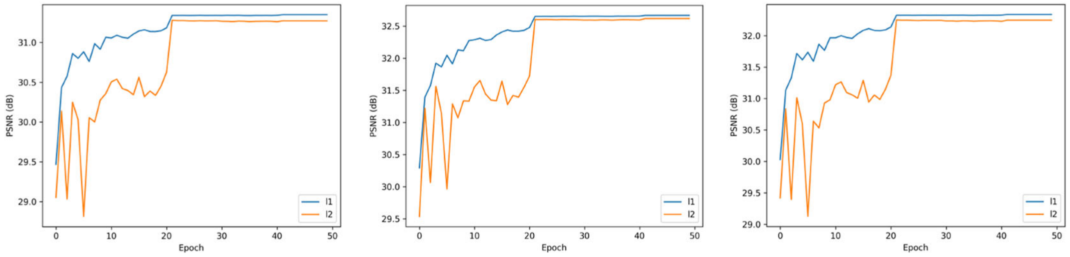
```
Input: input training dataset $\{x, y_t\} = \{x^{i}, y^{i}_{t}\}^{D}_{i=1}$
Output: $l_1(\theta)$
function STANDARDLOSS($x, y_t$)
    loss $\leftarrow$ 0
    for i do = 1 $\rightarrow$ D
        $loss = loss + \left\| f^{all}_{\text{ppnet}}(x^{i}) - (x^{i} - y^{i}_{t}) \right\|^{2}$
    end for
    $l_1(\theta) \leftarrow \frac{loss}{2D}$
    return $l_1(\theta)$
end function
```

As PPNets are recursive models, we can add supervised learning at every recursion to reform the loss function. Formally, all outputs of intermediate PPNets are simultaneously supervised by the MSE during training, and it can be formulated as Algorithm 3.

Algorithm 3 Recursive loss

```
Input: input training dataset $\{x, y_t\} = \{x^{i}, y^{i}_{t}\}^{D}_{i=1}$, recursive round $N$
Output: $l_2(\theta)$
function RECURSIVELOSS($x, y_t, N$)
    loss $\leftarrow$ 0
    for i do = 1 $\rightarrow$ D
        $I_{in} \leftarrow x^{i}$
        $loss_j \leftarrow 0$
        for j do = 1 $\rightarrow$ N
            $I_{out} = f^{j}_{ppnet}(I_{in})$
            $loss_j = loss_j + \left\| I_{out} - (x^{i} - y^{i}_{t}) \right\|^{2}$
            $I_{in} \leftarrow I_{in} - I_{out}$
        end for
        $loss \leftarrow loss + \frac{loss_j}{N}$
    end for
    $l_2(\theta) \leftarrow \frac{loss}{2D}$
    return $l_2(\theta)$
end function
```

We train $l_1(\theta)$ (non-recursive loss) and $l_2(\theta)$ (recursive loss) loss with images of noise level of 25 with PPNet-3. The images are from three publicly popular datasets for image denoising, i.e., CBSD68 [54], Kodak24 [55] and McMas-



(a) (b) (c)

Fig. 6 Average peak signal-to-noise ratio (PSNR) of PPNet-3 trained by $l_1(\theta)$ (non-recursive loss) and $l_2(\theta)$ (recursive loss) on **a** CBSD68, **b** Kodak24 and **c** McMaster with noise level of 25 at every epoch

ter [56]. The results of the corresponding PSNR regarding training epochs are shown in Fig. 6.

It is obvious that the model achieves better performance with non-recursive loss, so we use $l_1(\theta)$ as the loss function. Note that with the recursion time increasing, the network depth becomes deeper and deeper and hence non-recursive loss may suffer from gradients vanishing. However, the skip connection is integrated into the PPNets, which alleviates the effect of gradients vanishing to some extent. The possible reason for the poor performance of the recursive loss is the training instability brought by the multi-supervised loss, which makes the model hard to converge.

4 Experimental setup and results

4.1 Datasets

(1) Training datasets: For Gaussian denoising, it is known that different positions of an image carry different information. Following the idea of ADNet [19], we split the Berkeley Segmentation Dataset [57] and the Waterloo Exploration Database [58] into patches of size 50 × 50, and obtain 1348480 patches in total for training.

(2) Test datasets: we use two datasets, i.e., Set12 and BSD68 [54] to evaluate the model denoising performance on gray images, and another three datasets, i.e., CBSD68 (the colorful version of BSD68) [54], Kodak24 [55] and McMaster [56] to evaluate the model denoising performance on color images. Note that these five datasets consist of 12, 68, 68, 24 and 18 pictures, respectively, and all of them are widely used to test denoising performance.

4.2 Training setup

To balance the model complexity and the ability to capture more information, we simply set the depth of one Block to 18. The parameters of PPNets are initialized by He's method [59], and the BN's parameter values are set as the default of PyTorch, i.e., $\epsilon = 1e$-5, $\beta_1 = 0.99$ and $\beta_2 = 0.99$. We adopt Adam with learning rate of $1e$-3 and a mini-batch size of 128 to train the PPNets for 50 epochs. Specially, the learning rate decreased by a factor of 10 and 100 at the 20th epoch and the 40th epoch, respectively.

We use PyTorch 1.6.0 and Python 3.6 to train the proposed PPNets. All the experiments are conducted on Ubuntu 18.04 from a server with two Intel(R) Xeon(R) Gold 6240 CPUs at 2.60GHz, 256G RAM and two 2080Ti GPUs. Besides, the Nvidia CUDA of 10.2 and cuDNN of 7.6.5 are used to accelerate the training and test speed on GPUs. The source code of the PPNets can be accessed at https://github.com/e1evenn/ppnet/.

4.3 Quantitative and qualitative analysis

In this subsection, we will analyze and compare the PPNets with the competitive denoising models quantitatively and qualitatively. Specifically, we will evaluate the denoising models with PSNR, structural similarity index measure (SSIM), model complexity and visual effect.

4.3.1 PSNR

First, we report the average PSNR on two gray image sets: Set12 and BSD68 [54]. Six traditional denoising models (BM3D [13], WNNM [60], EPLL [61], MLP [32], CSF [62] and TNRD [63]) and five deep learning models (DnCNN [1], IRCNN [42], FFDNet [20], ECNDNet [64] and ADNet [19]) are selected as compared denoising models. They are all state-of-the-art denoising models and have great impacts in the community of image denoising. For each denoising model and each test image set, three noise levels ($\sigma = 15$, 25 and 50), are considered. The results are listed in Table 1, where the best result and the second best result of each test image set and noise level are shown in bold and italic, respectively.

From Table 1, we have the following findings: (1) In general, the deep learning denoising models are clearly superior to the traditional ones. The worst PSNR by deep learning

Table 1 Average PSNR (dB) on Set12 and BSD68 (gray images)

Method	Set12			BSD68		
	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50
BM3D [13]	32.37	29.97	26.72	31.07	28.57	25.62
WNNM [60]	30.70	30.26	27.05	31.37	28.83	25.87
EPLL [61]	32.14	29.69	26.47	31.21	28.68	25.67
MLP [32]	–	30.03	26.78	–	28.96	26.03
CSF [62]	32.32	29.84	–	31.24	28.74	–
TNRD [63]	32.50	30.06	26.81	31.42	28.92	25.97
DnCNN [1]	32.86	30.43	27.18	31.72	29.23	26.23
IRCNN [42]	32.77	30.38	27.14	31.63	29.15	26.19
FFDNet [20]	32.77	30.44	27.32	31.64	29.19	*26.29*
ECNDNet [64]	32.81	30.39	27.15	31.71	29.22	26.23
ADNet [19]	*32.98*	*30.58*	27.37	**31.74**	*29.25*	*26.29*
PPNet-1	32.88	30.45	27.24	31.66	29.15	26.19
PPNet-2	32.97	*30.58*	*27.43*	*31.73*	29.23	*26.29*
PPNet-3	**33.00**	**30.63**	**27.48**	**31.74**	**29.26**	**26.32**

models are better than the best PSNR by traditional models. For example, when we investigate the results on Set12 with $\sigma = 25$, we can see that the worst PSNR achieved by deep learning model is 30.38 with IRCNN [42], while the best PSNR by traditional models is 30.26 with WNNM [60]. The former is clearly better than the latter. (2) As for the deep learning models except for the proposed PPNets, ADNet [19] achieves the best results in most cases, indicating that the attention mechanism may improve the denoising effectiveness. (3) PPNets exhibit promising denoising results. Specifically, PPNet-3 achieves the best results in all cases, followed by PPNet-2 that obtains the second best results in 4 out of 6 cases. PPNet-1 achieves results that are comparable to the DnCNN [1], IRCNN [42], FFDNet [20] and ECNDNet [64].

Then, we list the denoising results on the three color image sets (BSD68 [54], Kodak24 [55] and McMaster [56]) with five noise levels ($\sigma = 15$, 25, 35, 50 and 70) in Table 2. The PPNets denoising models are compared with one traditional model (CBM3D, BM3D for color images [13]) and four deep learning models (FFDNet [20], DnCNN [1], IRCNN [42] and ADNet [19]). From this table, we have the following similar findings as those from gray test image sets: (1) CBM3D performs the worst in all cases when compared with the deep learning models. Specifically, the PSNR values obtained by CBM3D [13] are much lower than those by the deep learning models. (2) When ADNet is compared with FFDNet [20], DnCNN [1] and IRCNN [42], it achieves the best results in all cases. It may owe to the attention mechanism used by ADNet [19]. (3) PPNet-3 is superior to any other denoising models in all cases, while PPNet-2 achieves the second best results in 9 out of 15 cases. PPNet-1 achieves comparable results to FFDNet [20], DnCNN [1] and IRCNN [42]. All these indicate that the proposed PPNets are very effective for denoising.

4.3.2 SSIM

Besides, we report the average SSIM values of different methods on five image sets: Set12, BSD68, CBSD68, Kodak24 and McMaster, where the first two are gray image sets and the rest are color image sets. Note that the noise level is set to be 25. As given in Table 3, PPNet-3 outperforms the compared methods in all gray image sets and achieves the second best results in all color image sets. Besides, PPNet-2 also achieves good results with gray images. Although ADNet obtains the best SSIM values in all the three color image sets, it performs poorly in gray image sets.

4.3.3 Model complexity

We further compare the complexity of the deep learning denoising models for grayscale images using the number of model parameters, the number of floating-point operations (FLOPs) and the sizes of the generated denoising model, as shown in Table 4. Note that the FLOPs are with the patch size of 50 × 50.

From this table, we have the following findings: (1) For the DnCNN [1], FFDNet [20], ECNDNet [64] and ADNet [19], they have close complexity in both parameter numbers ($\approx$ 0.5M) and model sizes ($\approx$ 2000KB) while FFDNet has less FLOPs than the other three models. (2) PPNet-1 has the lowest complexity among all the deep learning models in terms of the three indicators; and its complexity is only about $1/7 \sim 1/6$ of DnCNN [1], FFDNet [20], ECNDNet [64] and ADNet [19]. (3) As the number of the blocks in the PPNets increases, the complexity of the PPNets continues to grow. Specifically, once a block is appended to the PPNets, the number of the parameters, FLOPs and model sizes of the PPNets increase approximately by 0.07M, 0.19G and 290KB, respectively. (4) Although the PPNet-3 achieves the best PSNR with both grayscale images and color images, its complexity is far lower than that of DnCNN [1], FFDNet [20], ECNDNet [64] and ADNet [19]. Note that the complexity of IRCNN [42] is slightly lower than that of PPNet-3, but is significantly higher than that of PPNet-1 and PPNet-2. All the PPNets perform better than IRCNN [42] with PSNR, as shown in Tables 1 and 2. All these findings demonstrate that the proposed lightweight PPNets have lower complexity than the compared deep learning denoising models.

Table 2 Average PSNR (dB) on CBSD68, Kodak24 and McMaster (color images)

Dataset	Method	σ=15	σ=25	σ=35	σ=50	σ=75
CBSD68	CBM3D [13]	33.52	30.71	28.89	27.38	25.74
	DnCNN [1]	33.98	*31.31*	29.65	28.01	–
	IRCNN [42]	33.86	31.16	29.50	27.86	–
	FFDNet [20]	33.80	31.18	29.57	27.96	26.24
	ADNet [19]	*33.99*	*31.31*	*29.66*	28.04	26.33
	PPNet-1	33.88	31.17	29.51	27.87	26.16
	PPNet-2	33.98	31.24	29.64	*28.05*	*26.35*
	PPNet-3	**34.06**	**31.35**	**29.70**	**28.10**	**26.43**
Kodak24	CBM3D [13]	34.28	31.68	29.90	28.46	26.82
	DnCNN [1]	34.73	32.23	30.64	29.02	–
	IRCNN [42]	34.56	32.03	30.43	28.81	–
	FFDNet [20]	34.55	32.11	30.56	28.99	27.25
	ADNet [19]	*34.76*	*32.26*	*30.68*	29.10	27.40
	PPNet-1	34.64	32.09	30.51	28.90	27.19
	PPNet-2	34.75	32.23	30.67	*29.15*	*27.45*
	PPNet-3	**34.86**	**32.34**	**30.78**	**29.21**	**27.57**
McMaster	CBM3D [13]	34.06	31.66	29.92	28.51	26.79
	DnCNN [1]	34.80	32.47	30.91	29.21	–
	IRCNN [42]	34.58	32.18	30.59	28.91	–
	FFDNet [20]	34.47	32.25	30.76	29.14	27.29
	ADNet [19]	34.93	32.56	31.00	29.36	27.53
	PPNet-1	34.78	32.36	30.78	29.09	27.30
	PPNet-2	*34.95*	*32.57*	*31.05*	*29.46*	*27.62*
	PPNet-3	**35.04**	**32.67**	**31.13**	**29.54**	**27.76**

Table 3 Average SSIM on Set12, BSD68, CBSD68, Kodak24 and McMaster with noise level 25

Method	Set12	BSD68	CBSD68	Kodak24	McMaster
BM3D [13]	0.8529	0.8055	–	–	–
DnCNN [1]	0.8623	0.8259	–	–	–
IRCNN [42]	0.8613	0.8242	0.8780	0.8741	0.8443
FFDNet [20]	0.8637	*0.8267*	0.8748	0.8723	0.8444
ADNet [19]	0.8625	0.8263	**0.8793**	**0.8777**	**0.8497**
PPNet-1	0.8611	0.8214	0.8748	0.8722	0.8454
PPNet-2	*0.8650*	**0.8268**	0.8745	0.8724	0.8466
PPNet-3	**0.8656**	**0.8268**	*0.8788*	*0.8771*	*0.8493*

Table 4 Complexity of models on grayscale images

Method	Parameters (M)	FLOPs (G)	Model sizes (KB)
DnCNN [1]	0.56	1.39	2176
IRCNN [42]	0.19	0.46	726
FFDNet [20]	0.49	*0.3*	1896
ECNDNet [64]	0.52	1.3	2028
ADNet [19]	0.52	1.3	2028
PPNet-1	**0.07**	**0.19**	**290**
PPNet-2	*0.15*	0.37	*580*
PPNet-3	0.22	0.56	870

Table 5 Average running time on images of size 128 × 128, 256 × 256 and 512 × 512 with noise level 25 of running the schemes 10 times (in milliseconds)

Image Size	DnCNN [1]	ADNet [19]	IRCNN [42]	FFDNet [20]	PPNet-3	PPNet-2	PPNet-1
128*128	43.18	**2.26**	135.83	7.78	7.81	5.23	*2.72*
256*256	53.80	**2.34**	532.21	17.00	16.43	7.62	*2.83*
512*512	474.02	**2.50**	2165.35	54.99	59.52	29.97	*2.98*
128*128*3	N/A	**2.36**	59.08	7.26	8.04	5.41	*2.85*
256*256*3	N/A	**2.45**	535.00	20.34	18.08	8.25	*3.05*
512*512*3	N/A	**2.75**	2177.81	85.05	67.29	30.98	*3.43*

4.3.4 Running time

We report the running time of the proposed PPNets and the compared approaches with three different sizes on both grayscale and color images in Table 5.

It can be seen that with the increase in image sizes, the running time of all methods increase. Among the approaches, ADNet ranks the first while the proposed PPNet-1 ranks the second in all cases, but the gaps between them are very narrow. The main reason that PPNet-1 takes a little more running time is that we use group convolutions to implement depthwise convolutional operations. PPNet-3 and FFDNet have comparable running time but the former has significant denoising results. IRCNN's running time is so poor that it is more than 300 times that of PPNet-1.

4.3.5 Ablation study

In this subsection, we conduct ablation studies to validate the effectiveness of the components (skip connection in Bottleneck (SCB), long–short connection (LSC) and noise proposal network (NPN)) of the proposed PPNets. For simplicity, the experiments are conducted with the BSD68 datasets on noise level of 25 using the PPNet-1 and the PPNet-3.

Skip connection in Bottleneck. As we mentioned in Sect. 3.3, Bottleneck is composed of three SCBRs, and skip connection is introduced to combine features from shallow layers and deeper layers. Specifically, ReLU can map all negative values to zero and keep all positive values unchanged, which is called unilateral inhibition, and due to this characteristic, it makes model sparse and avoids gradient vanishing problems. However, negative value's neurons will "dead" after ReLU as their values are directly changed to 0 and no gradients will transmit on them in back propagation. Thus, skip connection can pass through the "dead" neurons' features to the deeper layers, which makes them reactivate.

Long–short connection. It is known that shallow layers' features have weak effect on deep layers with the enhancement of network growth. To solve this problem, we combine the local and global features to strengthen the expression ability of model. Specifically, we add the output of the first layer and the output of the last bottleneck (the second-longest arrow in Fig. 4) to complement the effect of deep layer, the so-called long–short connection.

Table 6 PSNR (dB) results of different methods from BSD68 on noise level of 25 (PPNet-1)

ID	SCB	LSC	NPN	PSNR (dB)
1	×	×	×	29.1381
2	✓	×	×	29.1414
3	✓	✓	×	29.1430
4 (PPNet-1)	✓	✓	✓	29.1468

Table 7 PSNR (dB) results of different methods from BSD68 on noise level of 25 (PPNet-3)

ID	SCB	LSC	NPN	PSNR (dB)
1	×	×	×	29.2468
2	✓	×	×	29.2490
3	✓	✓	×	29.2519
4 (PPNet-3)	✓	✓	✓	29.2607

Noise proposal network. It is very important to extract useful features from complex information. Thus, we propose a noise proposal network (NPN) to detect the rough noise values and noise levels simultaneously. The NPN takes the long–short connection's output features as input to extract the latent noise from the complex background. Besides, the noise level prediction procedure can be regarded as self-attention.

We use PPNet-1 and PPNet-3 to demonstrate the effect of each component of PPNets with BSD68 on noise level of 25, and the results are shown in Tables 6 and 7, respectively, where the first row model (ID=1) is the corresponding PPNet without all the aforementioned components (SCB, LSC and NPN), the second row model (ID=2) adds SCB on the model with ID=1, the third row model (ID=3) adds LSC to the model with ID=2, and finally, when NPN is appended to the model with ID=3, it obtains the corresponding PPNet-1 or PPNet-3 with ID=4.

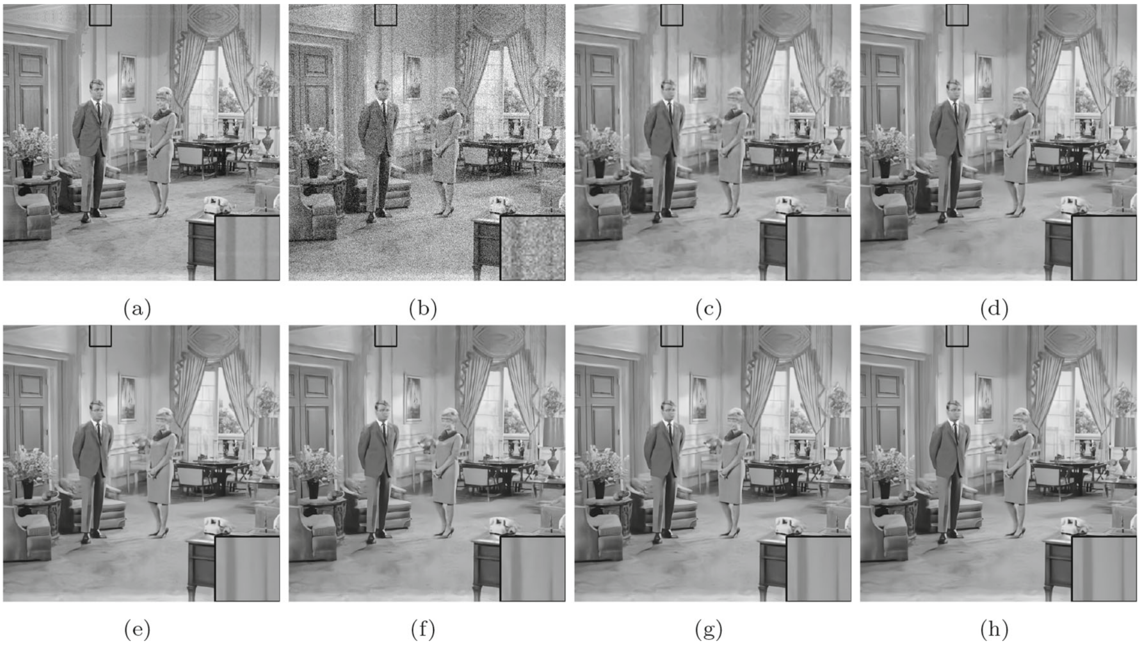


Fig. 7 Denoised images by different methods on an image (Couple) from Set12 with σ=25. **a** Original image, **b** noisy image (20.41 dB), **c** BM3D (29.7 dB) [13], **d** IRCNN (30.05 dB) [42], **e** FFDNet (30.14 dB) [20] **f** DnCNN (30.07 dB) [1], **g** ADNet (30.23 dB) [19], **h** PPNet-3 (30.28 dB)

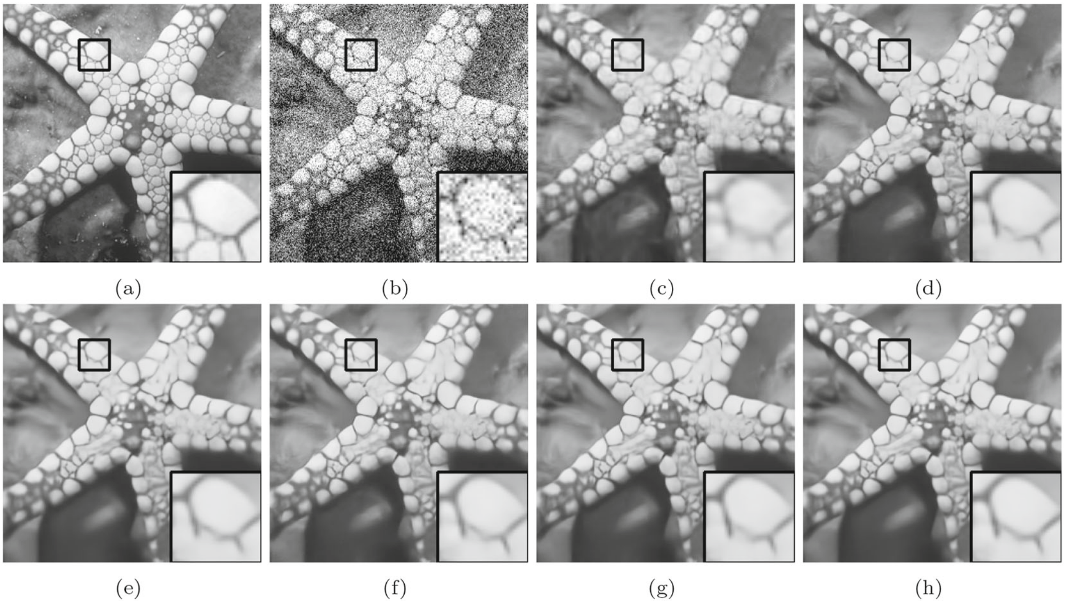
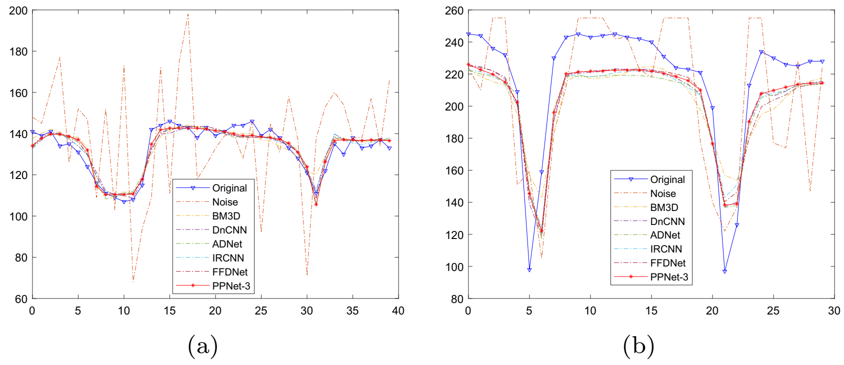


Fig. 8 Denoised images by different methods on an image (Starfish) from Set12 with σ=50. (a) Original image, (b) noisy image (14.98 dB), (c) BM3D (24.56 dB) [13], (d) IRCNN (24.99 dB) [42], (e) FFDNet (25.04 dB) [20], (f) DnCNN (25.1 dB) [1], (g) ADNet (25.04 dB) [19], (h) PPNet-3 (25.13 dB)



(a)

(b)

Fig. 9 Line profiles of rectangle areas of **a** Couple and **b** Starfish in Set12

When we look at these two tables, we can find that: (1) When none of the components (SCB, LSC and NPN) is used, both the PPNet-1 and the PPNet-3 obtain the worst results. (2) When one component is added to the existing models, the new model can improve the denoising effect. (3) When all the components are used, both the PPNet-1 and the PPNet-3 achieve the best results among all the corresponding denoising models. All these findings verify that each component contributes to the proposed denoising models and is capable of removing noises from complex background images.

4.4 Visual effect

In this subsection, we use some images from the experimental image datasets to compare the visual effect of the proposed model and the competitive models. For simplicity, we only use the proposed PPNet-3 for comparison. To be specific, the rectangle area with a black borderline is enlarged to emphasize the denoising effect. The more clearer the area is, the more robust the model is.

Figures 7 and 8 show the denoising results of two grayscale images, namely Couple and Starfish, by the proposed PPNet-3 and BM3D [13], IRCNN [42], FFDNet [20], DnCNN [1] and ADNet [19]. From these two figures, we can see that the classic denoising method BM3D performs the worst among all the denoising models on the two grayscale images. The denoised images by BM3D clearly exhibit noise in some positions of the images. The deep learning denoising approaches can improve the denoising effect so significantly that we cannot see any noise directly from the denoised images by these approaches. Among the deep learning models, the IRCNN and the FFDNet perform slightly worse than the other deep learning models. The proposed PPNet-3 achieves better or comparable visual effect to ADNet, FFDNet and IRCNN, although the PSNR by the PPNet-3 is superior to that of ADNet, FFDNet and IRCNN. These findings can be confirmed by investigating the line profiles of the rectangle areas of Couple and Starfish, where the denoised images by the proposed PPNet-3 are mostly close to the original images, as shown in Fig. 9.

As for color images, Figs. 10 and 11 display the denoising visual effects of an image (kodim07.png) from Kodak24 and an image (13.tif) from McMaster, respectively, by the proposed PPNet-3 and three deep learning denoising models (ADNet, FFDNet and IRCNN). From these two figures, the PPNet-3 and the other three deep learning models perform quite well. Although the difference in the visual effect among all the images is not very obvious, the details of the rectangle areas by the PPNet-3 are much purer than those of the compared models. This can also be found quantitatively from the corresponding line profile of each channel shown in Fig. 12, where the distances from the denoised images by the PPNet-3 to the original images are clearly much shorter than those from the denoised images by the compared models with many points.

All these demonstrate that the PPNet-3 can achieve promising visual effects for both grayscale and color images.

4.5 Discussion

The experimental results in Sect. 4 demonstrate the effectiveness and efficiency of proposed method. We speculate these outstanding performance might due to the task decomposition and growth depth of network.

4.5.1 Task decomposition

To our knowledge, the denoising performance on higher noise level images is usually worse than on lower noise level images. Most approaches conduct denoising directly on raw input noisy image and output the denoising result end to end. In order to obtain better denoising results, we simply divide straightforward denoising process into several recursive steps, and the input of every step is cleaner and cleaner than the raw input noisy image, which significantly reduces the complexity and difficulty of the original denoising task.

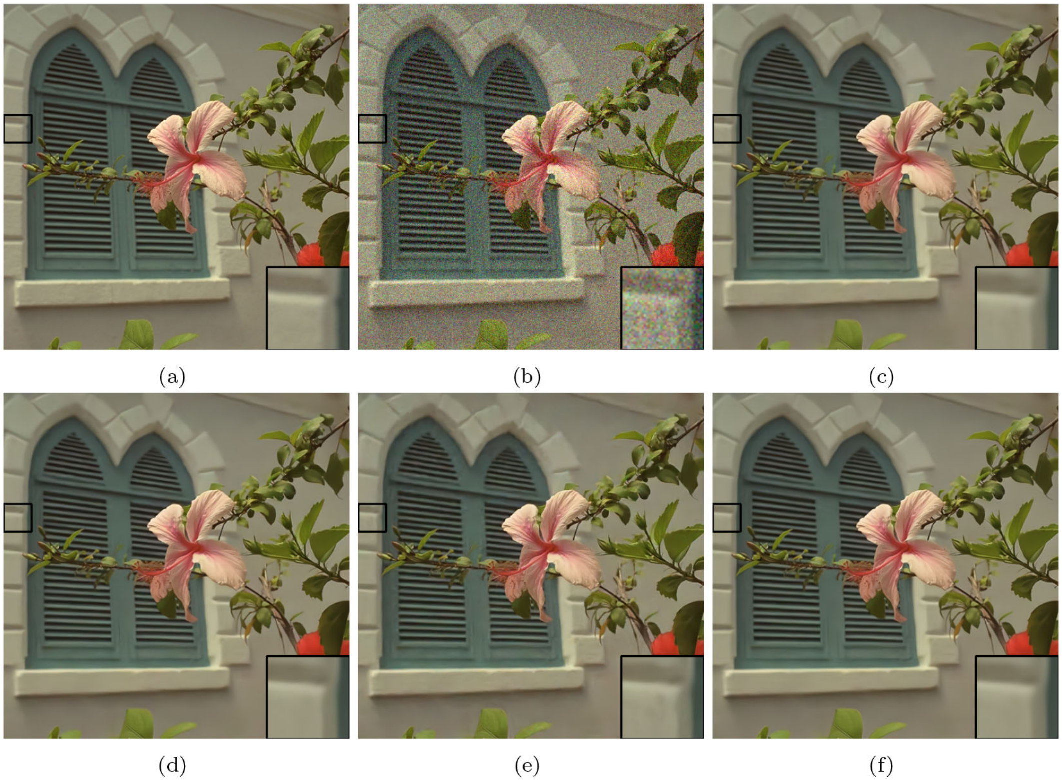


Fig. 10 Denoised images by different methods on an image (kodim07.png) from Kodak24 with σ=25. **a** Original image, **b** noisy image (20.43 dB), **c** ADNet (34.25 dB) [19], **d** FFDNet (34.07 dB) [20], **e** IRCNN (33.83 dB) [42], **f** PPNet-3 (34.36 dB)

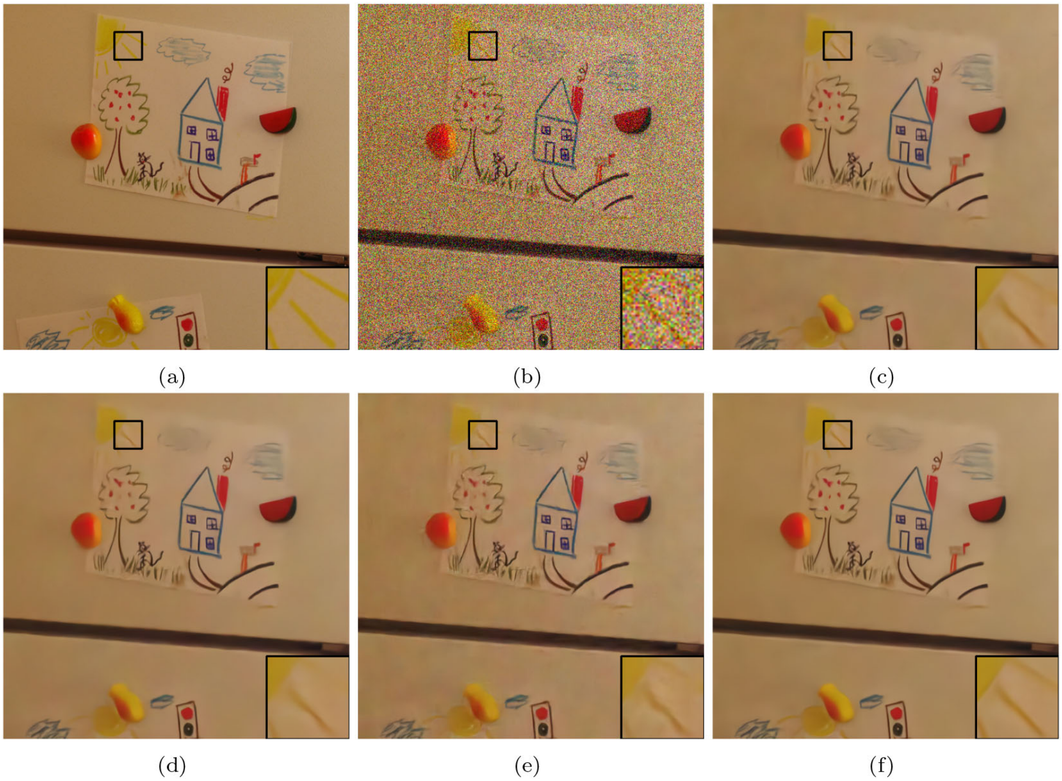


Fig. 11 Denoised images by different methods on an image (13.tif) from McMaster with σ=50. **a** Original image, **b** noisy image (14.57 dB), **c** ADNet (32.73 dB) [19], **d** FFDNet (32.48 dB) [20], **e** IRCNN (32.21 dB) [42], **f** PPNet-3 (33.04 dB)

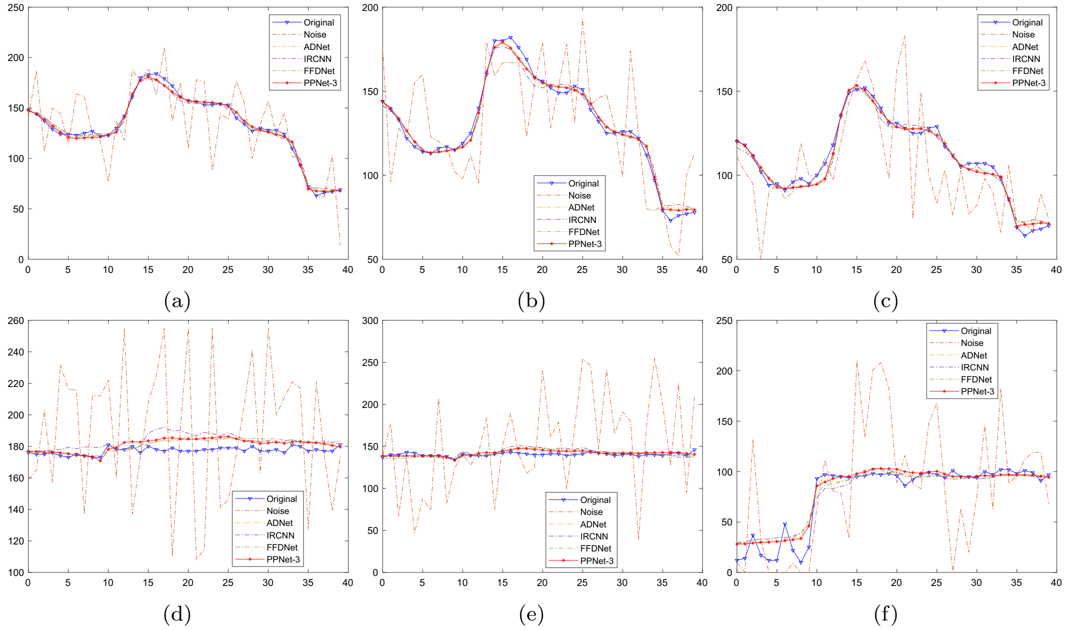


Fig. 12 Line profiles of two color images. **a**, **b** and **c**: Line profiles of the R, G and B channel of 13.tif from Kodak24, respectively. **d**, **e** and **f**: Line profiles of the R, G and B channel of kodim07.png from McMaster, respectively

4.5.2 Growth depth of network

It is known that deep neural networks usually lead to better performance. By stacking the same network architecture several times, PPNets decompose the denoising task and increase the network depth simultaneously. Although deep neural networks may suffer from gradient vanishing and parameter size explosion, PPNets introduce skip connection and separable convolutions to avoid these problems.

The advantages mentioned above make the proposed PPNets lightweight and effective denoising methods.

5 Conclusion

Image denoising is a primary task in computer vision and pattern recognition. In recent years, deep learning-based approaches have shown their power in improving denoising performance. To further enhance the effectiveness of deep learning models and reduce their complexity, this paper proposes recursive lightweight CNNs that can make the noisy images purer and purer for image denoising, the so-called PPNets. Several components contribute to the PPNets. The SCBR with depthwise convolutions and pointwise convolutions can significantly reduce the complexity of the models while keeping the representative ability of convolutions. The skip connection can avoid gradient vanishing problems. The LSC uses local and global features to enhance the representative ability of the models. The NPN integrates the attention mechanism to detect rough noise and noise levels simultaneously. These components construct one Block, and one or more blocks can be stacked to obtain resultant clean images. The extensive experiments demonstrate that the proposed lightweight PPNets are promising for image denoising in terms of the evaluation indicators. In the future, we will apply the PPNets to other tasks of image restoration, such as super-resolution, dehazing and deraining.

Acknowledgements This work was supported by the Ministry of Education of Humanities and Social Science Project (Grant no. 19YJAZH047) and the Scientific Research Fund of Sichuan Provincial Education Department (Grant no. 17ZB0433).

Data Availability All data generated or analyzed during this study are included in this published article.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian Denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**(7), 3142–3155 (2017)
2. Goyal, B., Dogra, A., Agrawal, S., Sohi, B.S., Sharma, A.: Image denoising review: from classical to state-of-the-art approaches. Inf. Fusion **55**, 220–244 (2020)
3. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 839–846. IEEE (1998)

4. Benesty, J., Chen, J., Huang, Y.: Study of the widely linear wiener filter for noise reduction. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 205–208. IEEE (2010)
5. Zhang, M., Gunturk, B.K.: Multiresolution bilateral filtering for image denoising. IEEE Trans. Image Process. **17**(12), 2324–2333 (2008)
6. Buades, A., Coll, B., Morel, J.-M.: A review of image denoising algorithms, with a new one. Multiscale Model. Simul. **4**(2), 490–530 (2005)
7. Wang, S., Xia, Y., Liu, Q., Luo, J., Zhu, Y., Feng, D.D.: Gabor feature based nonlocal means filter for textured image denoising. J. Vis. Commun. Image Represent. **23**(7), 1008–1018 (2012)
8. Zhang, X., Feng, X., Wang, W.: Two-direction nonlocal model for image denoising. IEEE Trans. Image Process. **22**(1), 408–412 (2012)
9. Tian, C., Zheng, M., Zuo, W., Zhang, B., Zhang, Y., Zhang, D.: Multi-stage image denoising with the wavelet transform. Pattern Recognit. **134**, 109050 (2022)
10. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. **15**(12), 3736–3745 (2006)
11. Zhuang Zhang, X., Chen, L.L., Li, Y., Deng, Y.: A sparse representation denoising algorithm for visible and infrared image based on orthogonal matching pursuit. SIViP **14**(4), 737–745 (2020)
12. Mahdaoui, A.E., Ouahabi, A., Moulay, M.S.: Image denoising using a compressive sensing approach based on regularization constraints. Sensors **22**(6), 2199 (2022)
13. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
14. Zhang, M., Desrosiers, C.: Image denoising based on sparse representation and gradient histogram. IET Image Process **11**(1), 54–63 (2017)
15. Liu, H., Li, L., Jiangbo, L., Tan, S.: Group sparsity mixture model and its application on image denoising. IEEE Trans. Image Process. **31**, 5677–5690 (2022)
16. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imag. Vis. **20**(1), 89–97 (2004)
17. Fang, F., Li, F., Zeng, T.: Single image dehazing and denoising: a fast variational approach. SIAM J. Imag. Sci. **7**(2), 969–996 (2014)
18. Islam, M.R., Xu, C., Han, Y., Ashfaq, R.A.R.: A novel weighted variational model for image denoising. Int. J. Pattern Recognit. Artif. Intel. **31**(12), 17540222 (2017)
19. Tian, C., Yong, X., Li, Z., Zuo, W., Fei, L., Liu, H.: Attention-guided CNN for image denoising. Neural Netw. **124**, 117–129 (2020)
20. Zhang, K., Zuo, W., Zhang, L.: FFDNet: toward a fast and flexible solution for CNN-based image denoising. IEEE Trans. Image Process. **27**(9), 4608–4622 (2018)
21. Zhang, Y., Li, K., Li, K., Sun, G., Kong, Y., Fu, Y.: Accurate and fast image denoising via attention guided scaling. IEEE Trans. Image Process. **30**, 6255–6265 (2021)
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 234–241. Springer (2015)
23. Qiao, S., Yang, J., Zhang, T., Zhao, C.: Layered input GradiNet for image denoising. Knowl.-Based Syst. **254**, 109587 (2022)
24. Zhang, J., Cao, L., Wang, T., Wenlong, F., Shen, W.: NHNet: a nonlocal hierarchical network for image denoising. IET Image Proc. **16**(9), 2446–2456 (2022)
25. Jia, F., Wong, W.H., Zeng, T.: DDUNet: Dense dense U-net with applications in image denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 354–364. IEEE (2021)
26. Bai, Yu., Liu, M., Yao, C., Lin, C., Zhao, Y.: Multi-stage progressive network for image denoising. Neurocomputing, MSPNet (2022)
27. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 3147–3155. IEEE (2017)
28. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 2472–2481. IEEE (2018)
29. Jia, X., Chai, H., Guo, Y., Huang, Y., Zhao, B.: Multiscale parallel feature extraction convolution neural network for image denoising. J. Electron. Imaging **27**(6), 063031 (2018)
30. Quan, Y., Chen, Y., Shao, Y., Teng, H., Yong, X., Ji, H.: Image denoising using complex-valued deep CNN. Pattern Recogn. **111**, 107639 (2021)
31. Rawat, S., Rana, K.P.S., Kumar, V.: A novel complex-valued convolutional neural network for medical image denoising. Biomed. Signal Process. Control **69**, 102859 (2021)
32. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with BM3D? In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 2392–2399. IEEE (2012)
33. Jain, V., Seung, S.: Natural image denoising with convolutional networks. In Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS), NIPS'08, pp. 769–776, Red Hook, NY, USA, Curran Associates Inc (2008)
34. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **25**, 1097–1105 (2012)
35. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 1–9. IEEE (2015)
36. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 770–778. IEEE (2016)
37. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: a persistent memory network for image restoration. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 4549–4557. IEEE (2017)
38. Tian, C., Yong, X., Zuo, W.: Image denoising using deep CNN with batch renormalization. Neural Netw. **121**, 461–473 (2020)
39. Chen, C., Xiong, Z., Tian, X., Wu, F.: Deep boosting for image denoising. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–18. Springer (2018)
40. Zuo, Z., Chen, X., Xu, H., Li, J., Liao, W., Yang, Z.X., Wang, S.: IDEA-Net: adaptive dual self-attention network for single image denoising. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 739–748. IEEE (2022)
41. Jia, F., Ma, L., Yang, Y., Zeng, T.: Pixel-attention CNN with color correlation loss for color image denoising. IEEE Signal Process. Lett. **28**, 1600–1604 (2021)
42. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep CNN denoiser prior for image restoration. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 3929–3938. IEEE (2017)
43. Huang, J., Zhao, Z., Ren, C., Teng, Q., He, X.: A prior-guided deep network for real image denoising and its applications. Knowl.-Based Syst. **255**, 109776 (2022)
44. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 1251–1258. IEEE (2017)
45. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient con-

volutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

46. Li, G., Jiang, S., Yun, I., Kim, J., Kim, J.: Depth-wise asymmetric bottleneck with point-wise aggregation decoder for real-time semantic segmentation in urban scenes. IEEE Access **8**, 27495–27506 (2020)
47. Wang, J., Xiong, H., Wang, H., Nian, X.: ADSCNet: asymmetric depthwise separable convolution for semantic segmentation in real-time. Appl. Intell. **50**(4), 1045–1056 (2020)
48. Xuan, L., Un, K.-F., Lam, C.-S., Martins, R.P.: An FPGA-based energy-efficient reconfigurable depthwise separable convolution accelerator for image recognition. IEEE Trans. Circuits Syst. II Express Briefs **69**(10), 4003–4007 (2022)
49. Li, W., Chen, H., Liu, Q., Liu, H., Wang, Y., Gui, G.: Attention mechanism and depthwise separable convolution aided 3DCNN for hyperspectral remote sensing image classification. Remote Sens. **14**(9), 2215 (2022)
50. Li, X., Zhengshun, D., Huang, Y., Tan, Z.: A deep translation (GAN) based change detection network for optical and SAR remote sensing images. ISPRS J. Photogramm. Remote. Sens. **179**, 14–34 (2021)
51. Muhammad, W., Aramvith, S., Onoye, T.: Multi-scale xception based depthwise separable convolution for single image super-resolution. PLoS ONE **16**(8), e0249278 (2021)
52. Liu, G., Dang, M., Liu, J., Xiang, R., Tian, Y., Luo, N.: True wide convolutional neural network for image denoising. Inf. Sci. **610**, 171–184 (2022)
53. Mao, X., Shen, C., Yang, Y.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. Adv. Neural. Inf. Process. Syst. **29**, 2802–2810 (2016)
54. Roth, S., Black, M.J.: Fields of experts: a framework for learning image priors. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 860–867. IEEE (2005)
55. Franzen, R.: Kodak lossless true color image suite. http://r0k.us/graphics/kodak (1999)
56. Zhang, L., Xiaolin, W., Buades, A., Li, X.: Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. J. Electron. Imaging **20**(20), 023016 (2011)
57. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 416–423. IEEE (2001)
58. Ma, K., Duanmu, Z., Qingbo, W., Wang, Z., Yong, H., Li, H., Zhang, L.: Waterloo exploration database: new challenges for image quality assessment models. IEEE Trans. Image Process. **26**(2), 1004–1016 (2016)
59. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034. IEEE (2015)
60. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 2862–2869. IEEE (2014)
61. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 479–486. IEEE (2011)
62. Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), pp. 2774–2781. IEEE (2014)
63. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1256–1272 (2016)
64. Tian, C., Yong, X., Fei, L., Wang, J., Wen, J., Luo, N.: Enhanced CNN for image denoising. CAAI Trans. Intell. Technol. **4**(1), 17–23 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jiayi Shi is pursuing a master degree at Southwestern University of Finance and Economics. His research interests include image processing and machine learning.

Taiyong Li is a full professor at Southwestern University of Finance and Economics. He received his BS, MS and PhD degrees in computer science from Sichuan University in 2001, 2004 and 2009, respectively. He is the author of more than 50 journal papers. His current research interests include image processing, machine learning, and artificial intelligence.

Jiaxuan Xu is pursuing a master degree at Southwestern University of Finance and Economics. His current research interests include image processing and machine learning.