


Article

Deep Memory Connected Neural Network for Optical Remote Sensing Image Restoration

Wenjia Xu ^{1,2}, Guangluan Xu ^{1,*}, Yang Wang ¹, Xian Sun ¹, Daoyu Lin ¹  and Yirong Wu ^{1,2}

¹ Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China; xuwenjia16@mails.ucas.ac.cn (W.X.); primular@163.com (Y.W.); sunxian@mail.ie.ac.cn (X.S.); lindaoyu15@mails.ucas.ac.cn (D.L.); wyr@mail.ie.ac.cn (Y.W.)

² School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: gluanxu@mail.ie.ac.cn

Received: 17 October 2018 ; Accepted: 21 November 2018; Published: 27 November 2018



Abstract: The spatial resolution and clarity of remote sensing images are crucial for many applications such as target detection and image classification. In the last several decades, tremendous image restoration tasks have shown great success in ordinary images. However, since remote sensing images are more complex and more blurry than ordinary images, most of the existing methods are not good enough for remote sensing image restoration. To address such problem, we propose a novel method named deep memory connected network (DMCN) based on the convolutional neural network to reconstruct high-quality images. We build local and global memory connections to combine image detail with global information. To further reduce parameters and ease time consumption, we propose Downsampling Units, shrinking the spatial size of feature maps. We verify its capability on two representative applications, Gaussian image denoising and single image super-resolution (SR). DMCN is tested on three remote sensing datasets with various spatial resolution. Experimental results indicate that our method yields promising improvements and better visual performance over the current state-of-the-art. The PSNR and SSIM improvements over the second best method are up to 0.3 dB.

Keywords: deep memory connected network; remote sensing; image restoration; single image super-resolution; image denoising

1. Introduction

Optical remote-sensing images are widely used in various applications, such as automatic detection, target recognition, object tracking, and image classification. However, the image quality and spatial resolution are still the key limitations affecting remote sensing applications.

The resolution of images indicates the spatial size of the planetary surface covered by one pixel. Thus, it reflects the ability to capture small terrain details. Thanks to the development of the sensor, the most advanced satellites are able to distinguish spatial information within a squared meter [1]. When requiring higher resolution, exceeding the limitation of sensors is a time consuming and costly task. Besides, the optical images will be degraded due to the undesirable environmental conditions (such as clouds and uneven illumination) and the noise produced by the sensor.

As is shown in Figure 1, low-resolution and noise will lead to low image quality and restrain the accuracy of image interpretation [2]. It is necessary to perform image restoration tasks such as super-resolution and denoising, which will improve image quality efficiently.

Image restoration is a classical problem, aiming at recovering latent clean image x from its degraded observation y . We assume that the clean image x is polluted by the function D combined with an additive zero-mean white Gaussian noise N . Thus, the measured image is

$$y = D(x) + N. \quad (1)$$

We desire to design an algorithm that can recover a high-quality image \hat{x} , and making it as close as possible to the original image x . However, there is no unique solution for a given image y , which makes image restoration to be an ill-posed problem.

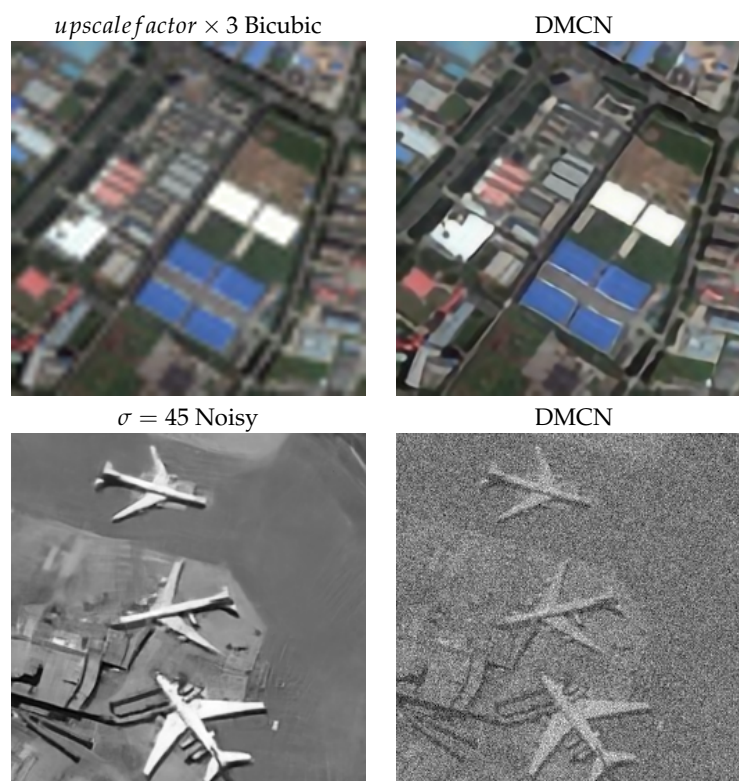


Figure 1. The $\times 3$ super-resolution results of our method (DMCN) compared with $\times 3$ bicubic image. The denoising results of DMCN, compared to a noisy image with Gaussian noise level $\sigma = 45$.

To address this problem, over the last decades, numerous contributions for image restoration are addressed from diverse points of view. These algorithms can be concluded in to neighbor embedding methods [3–5], sparsity-based methods [6,7], and low-rank minimization [8,9]. Some representative algorithms are illustrated in Section 2.1.

Due to the immense popularity of deep learning, convolutional neural networks stand out as a powerful image restoration tool since they provide significantly improved performance. Among them, Super-resolution Convolutional Neural Network (SRCNN) [10] is the first successful attempt that use a three-layer convolutional neural network for super-resolution. Feed-forward Denoising Convolutional Neural Network (DnCNN) [11] is another successful attempt that achieves the-state-of-the-art performance on image denoising, which stacks convolutional layers, Rectified Linear Unit (ReLU) [12] and Batch Normalization (BN) [13] functions. A deep generative network is proposed by [1] to improve the SR process with little external high-resolution (HR) training images.

Although those CNN-based methods achieve excellent results, there are still several weak points could be improved. Firstly, these methods such as SRCNN [10] and DnCNN [11] are very shallow (less than 20 layers). Therefore their receptive fields are small. The network capability is not satisfactory

when reconstructing HR images with extensive information. Some recent methods apply deep network on ordinary images [14–18], but no experiments have been done on remote sensing images. Secondly, these methods ignore the local information produced by the lower layers and sometimes cannot reconstruct image details correctly. Despite achieving better results, those models are time-consuming due to the complex operation or redundancy structure. It is hard to carry out those restoration jobs on real-time platforms.

Particularly, remote sensing images are more complex and blurry than ordinary images. For example, an image from the ImageNet dataset measuring 256×256 pixels may only depict an animal or a building. While an equally sized image in satellite dataset may cover a small town with many buildings and streets (shown in Figure 2). Processing remote sensing images requires stronger ability. However, methods designed for ordinary images such as DnCNN may not meet the special requirement and fails on remote sensing images. We do an experiment to verify this phenomenon and show the result in Section 4.6.



Figure 2. The comparison of remote sensing images and ordinary images. The remote sensing images are randomly selected from NWPU-RESISC45 [19] dataset. The ordinary images are randomly selected from a Set12 dataset.

Motivated by the problem above, we propose a fast and accurate image restoration method by building a deep memory connected network (DMCN). We build a deep convolutional neural network with a large receptive field to recover the latent clean image from the degraded one. The large receptive field provides more context for predicting image details. Inspired by neuron science study that short-term memory can be consolidated to long-term memory by synaptic consolidation after rehearsal. We define the residual information at different stages as short-term memory and the information learned along the pipeline as a long-term memory. To imitate the activating intracellular transduction cascades, we add different residual information to the pipeline, which is called memory connections. Our method used two level memory connections: global and local. These connections can combine local and global information learned by the network, fasten convergence and prevent vanishing gradients problem [20]. In addition, we apply Downsample Units to shrink the spatial size of the feature maps, alleviating the computational burden and accelerating the training process. In details, we use initialization described in [21], adaptive moment estimation (adam) [22] optimizer, batch normalization [19] and parametric rectified linear unit (PReLU) [21] to accelerate training process and achieve better performance. In addition, our model is designed for parallel computation on GPUs, making the restoration tasks more efficient.

The contributions of this work are as follows:

1. We build a deep memory connected network for high-quality remote-sensing image restoration. Our network can handle various image restoration tasks such as super-resolution and Gaussian denoising at the same time. We can also achieve blind Gaussian denoising for unknown noise

- level. By simply changing training datasets, our network can be applicable for super-resolution with different upscale factors.
2. Taking into account the lower layer information, DMCN is elaborately designed with local and global memory connections. With the global connection, DMCN only needs to predict high-frequency residual information instead of predicting the whole image. We use local residual in Basic Blocks to achieve fast error reduction.
 3. DMCN is elaborately designed with Downsample and Upsample Units to build an hourglass structure. With a Downsample Unit, we can shrink the spatial size of the feature map by 2, significantly reducing the memory footprint and time-consumption.
 4. We choose three representative optical remote sensing datasets with different spatial resolutions to train and test the model. Experiments show that our method outperforms the state-of-the-art algorithms in both super-resolution and denoising tasks. Besides, we apply BN and PReLU for faster convergence and relatively high performance.

The remainder of this paper is organized as follows. Section 2 presents an overview of traditional algorithms and deep-learning-based methods on image restoration. Section 3 describes the methodology used by our model. Section 4 verifies the effectiveness of DMCN by performing comparisons with the state-of-the-art image restoration methods. Section 5 concludes the paper with discussions and outlooks.

2. Related Works

2.1. Traditional Algorithms

Image restoration is an old but still hot problem. Over the last decades, numerous approaches based on traditional algorithms are proposed. Neighbor embedding methods find the nearest neighbors in the observed image Y , and reconstruct image X by computing the high-resolution embedding using the appropriate high-resolution features of the K nearest neighbors [3–5]. Xu et al. [23] uses patch grouping-based nonlocal means algorithm to denoise remote sensing images. C. Kwan and J. Zhou [24] propose a patent for image denoising, which provided a useful method for practice.

Sparsity-based methods define a trained dictionary and sparsely represent the latent clean image [6,7]. We assume that the latent clean image is x . Thus, the sparse representation $\hat{\alpha}$ is formulated as:

$$\hat{\alpha} = \arg \min_{\alpha} \|\alpha\| \quad \text{subject to } D\alpha \approx x, \quad (2)$$

where D is a redundant dictionary (matrix). The basic idea here is that every image patch x can be denoted as a linear combination of few columns from dictionary D . Chen et al. [25] presents a nonconvex low rank matrix approximation (NonLRMA) model to decompose the degraded Hyperspectral images. Fan et al. [26] propose an MSI denoising model based on nonlocal multitask sparse learning to fully exploit the nonlocal self-similarity of the MSI on the spatial domain.

Low-rank minimization is another strategy to exploit the underlying low-rank matrix from its degraded observation [8], where weighted nuclear norm minimization (WNNM) [9] problem uses the F -norm to measure the difference between observed data matrix y and latent data matrix x , which can be formulated as

$$\min_x \|y - x\|_2^F + \|x\|_{w,*}, \quad (3)$$

where $\|x\|_{w,*}$ represents the weighted nuclear norm to regularize x .

Pansharpening aims at improving the spatial resolution of multispectral data, which is a special instance of super-resolution. Pansharpening fuse multispectral or hyperspectral image data with panchromatic bands [27]. In [28], the author reviews some advanced methods for multispectral pansharpening. Kwan et al. [29] integrates two newly developed techniques, hybrid color mapping algorithm and Plug-and-Play algorithm, to present a new resolution enhancement method for HS

images. This algorithm is different from former ones because it only requires an HR color image and a low resolution (LR) HS image cube.

2.2. Deep Learning Methods

With the evolution of deep learning techniques, the neural network has shown standout potential. Remote sensing scientists also exploit the advantage of deep learning to tackle various challenges, such as image recognition, object detection, image classification, and image restoration. Remote sensing data also bring new chance and challenge for deep learning. With big and heterogeneous remote sensing data, it is more tangible to study information for image restoration. In this part, we provide a brief overview of some representative deep learning based methods for denoting and super-resolution tasks.

2.2.1. Image Denoising

There have been several methods attempting to handle the denoising problem by neural networks. Vincent et al. [30] first proposes a two-layer neural network (Denoising Autoencoder) that tries to reconstruct the latent image from an observed noisy image. This denoising autoencoder could be formulated as

$$\hat{x} = \sigma(b_2 + W_2\sigma(b_1 + W_1x)), \quad (4)$$

where σ is the sigmoid activation function $\sigma(x) = 1/(1 + e^{-x})$. W_1, W_2 is $d \times d$ weight matrix and b is a bias vector. The weight matrix W of the reverse mapping may optionally be constrained by $W = W^T$, in which case the autoencoder is said to have tied weights.

Jain and Seung [31] propose a simple network with four hidden layers that provide comparable and in some cases superior performance to Markov random field (MRF) methods. In [32], Chen et al. uses a trainable nonlinear reaction diffusion (TNRD) network which extends traditional nonlinear reaction diffusion models by highly parametrized linear filters and highly parametrized influence functions. TNRD can achieve promising performance comparable to BM3D [5]. However, these methods learn the parameter by stage-wise greedy training, involving many handcrafted parameters. Besides, for different noise level, a certain network should be trained. The most famous CNN-based method is DnCNN [11], which achieves the state-of-the-art results. DnCNN is a deep network that utilizes residual learning to estimate the Gaussian noise. Remarkably, a single-blind DnCNN network can achieve denoising tasks with different noise levels.

For image denoising, many new works are proposed recently. Ben et al. [33] utilize a technique for predicting spatially varying kernels that can both align and denoise frames. This method works well on some ordinary image datasets, however, the generalization ability should be discussed. Lefkimmatis et al. [17] design a network for color and grayscale image denoising. This network can be trained for a wide range of noise levels using a single set of parameters. Lehtinen et al. [34] achieves noise removal, denoising synthetic Monte Carlo images, and reconstruction of undersampled MRI scan only based on noisy data, without explicit image priors or likelihood models of the corruption.

Except for the convolutional neural network, some other networks also perform well on denoising tasks. Patrick Putzky and Max Welling [35] propose a network, Recurrent Inference Machines (RIM), based on a recurrent neural network, which allows for an abstraction which removes the need for domain knowledge. The former methods, such as DnCNN [11], often suffer from the lack of paired training data. Chen et al. [18] use a Generative Adversarial Network (GAN) to estimate the distribution of noise and generate noisy data. Then the output of GAN is used for training a CNN for denoising.

2.2.2. Single Image Super Resolution

Network-based super resolution methods are also popular over recent years. Dong et al. propose a three layer network Super-resolution Convolutional Neural Network (SRCNN) [10] to learn an end-to-end mapping between the low/high-resolution images. SRCNN can be viewed as an

extension of sparse-coding-based SR method which demonstrates superior performance to the previous hand-crafted models either in speed and restoration quality. The network SRCNN is represented as follows,

$$\hat{x} = b_3 + W_3 * R(b_2 + W_2 * R(b_1 + W_1 * y)), \quad (5)$$

where W_1, W_2, W_3 represent the weight matrixes. The matrix is of size $c \times f \times f \times n$, where c is the number of the input channel, f is the spatial size of a filter, and n is the number of filters. R represents the Rectified Linear Unit [12] (ReLU, $\text{Max}(0, x)$). Remarkably, $*$ denotes the convolution operation. A convolutional layer is denoted as $\text{Conv}(c, f, n)$.

Then the author adopts smaller filter sizes in Fast SRCNN (FSRCNN) [36] to accelerate SRCNN. FSRCNN introduce a deconvolution layer at the end of the network, the mapping is learned directly from the original low-resolution image (without interpolation) to the high-resolution one. This operation reduces the computational complexity but brings a non-negligible drawback, that a specific model should be trained for a certain upscale factor. Kim et al. [37] use 20 convolutional layers in Very Deep Super-resolution Network (VDSR) to improve SR performance. To accelerate the convergence speed and avoid gradient explosion problem, VDSR adopts residual learning and very high learning rate.

However, those networks are too simple, and the network capability is not satisfactory when reconstructing HR images with extensive information. A deep network can utilize more contextual information in an image and usually achieves better performance than shallow ones.

For image super-resolution, many new works are proposed recently. In [38], the author applies transfer learning to achieve hyperspectral image super-resolution. The network is trained on a natural image dataset, and then fine-tuned on hyperspectral images. Qu et al. [39] propose two encoder-decoder networks to preserve the rich spectral information from the HSI network. This method achieves unsupervised learning for hyperspectral image super-resolution. Assaf Shocher et al. [40] apply “Zero-Shot” in super-resolution, which does not rely on prior training. The image information is extracted by the network, and an image-specific CNN is trained during testing time. Only the information from the input image is used when training.

Except for the convolutional neural network, some other networks also perform well on super-resolution task. Ledig et al. [41] proposes a generative adversarial network Super-resolution Generative Adversarial Network (SRGAN), which achieves photo-realistic natural images for $4 \times$ upscaling factors. In this network, the author uses a perceptual loss function which consists of an adversarial loss and a content loss. Mehdi SM et al. [42] propose an end-to-end trainable frame-recurrent framework for video super-resolution. This method assimilates a large number of previous frames without increased computational demands.

3. Proposed Deep Connected Neural Network

In the following, we will demonstrate the architecture of the proposed DMCN network, including the interior structure and mathematical expressions. Then, the Downsample and Upsample Units and the memory connections will be illustrated in detail. Finally, we study the trade-off between network depth and performance, and introduce some useful training strategies.

3.1. Network Architecture

The overall structure of DMCN is illustrated in Figure 3. DMCN can be decomposed into four parts: the input unit, the Downsample Units, the Upsample Units and the output unit. A network with N convolutional layers can be denoted as follows,

$$F_i(X; W_i, b_i) = \sigma(W_i * F_{i-1}(X) + b_i), \quad i = 1 \sim N \quad (6)$$

As is illustrated above, there are several models for image super-resolution or denoising before DMCN. These methods complete these two tasks separately. However, we find out that there is an

inner connection between SR and denoising. As we can see, the purpose of image restoration tasks is to recover the latent clean image x from the corrupted image y . In denoising problems, the corrupted image y is

$$y = x + v, \quad (7)$$

where v is the additive Gaussian noise.

We observe that when v represents the difference between the ground truth high resolution image and the bicubic upsampling image of the low resolution one, the image restoration model can be converted to a single image super-resolution problem.

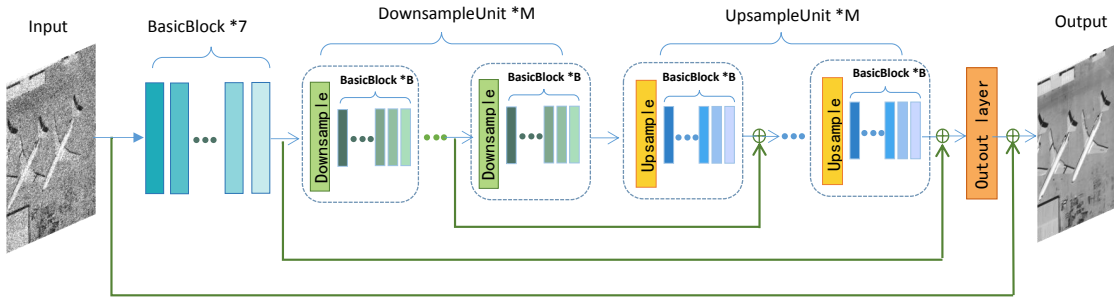


Figure 3. The architecture of DMCN is symmetrical as a whole. The structure of Basic Block and the Upsample Block is shown in Figure 4.

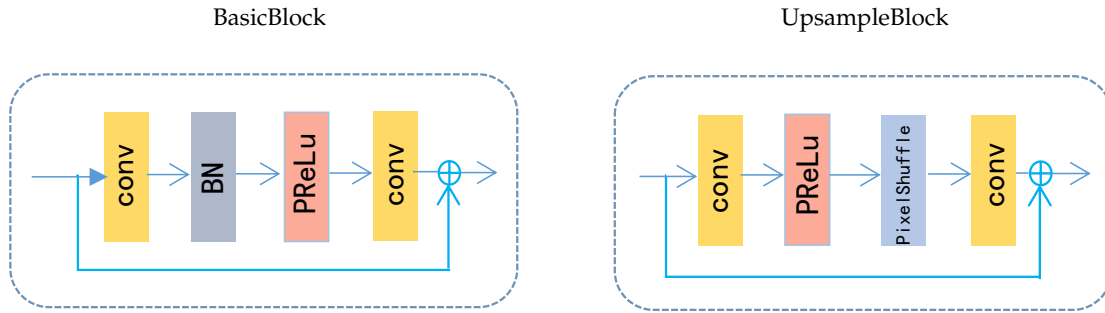


Figure 4. The structure of Basic Block and UpsampleBlock. Conv represents a convolutional layer. BN denotes Batch Normalization. PixelShuffle is the operation mentioned in [43].

3.2. Downsample Unit and Upsample Unit

Before we dive into the Downsample and Upsample Units, let's first have a look at the computational complexity of a convolutional network $Conv(c_i, f_i, n_i)$ with N layers:

$$O_{time} = \sum_{i=1}^N c_i \cdot f_i^2 \cdot n_i \cdot m_i, \quad (8)$$

where m_i is the spatial size of the output feature map, which significantly influences the computational complexity. To ease the computational burden, we propose an hourglass structure with M Downsample Units and Upsample Units to shrink the spatial size of the feature map. With M Downsample Units, the spatial size of feature map is shrunked by $2 * M$. With Upsample Units, the feature map is expanded to the original size. The size of all the feature map looks like an hourglass structure. Hourglass structure is proposed in U-Net [44] to achieve fast training and testing, in which hourglass is achieved by pooling operation. Newell et al. [45] propose a stacked hourglass network for human pose estimation. This network creates a stacked hourglass network by placing several hourglass modules end-to-end.

As is shown in Figure 3, every Downsample Unit contains a Downsample layer, which is a convolutional layer with stride = 2. Thus we can shrink the spatial size of the feature map by 2. There are B Basic Blocks following the Downsample layer, increasing the network depth and extracting useful information. The architecture of Basic Block is shown in Figure 4. Let x and $f_B(x)$ be the input and output of Basic Block, then it can be formulated as

$$f_B(x) = \max(0, BN(W_i x + b_i)), \quad (9)$$

where the function BN denotes Batch Normalization and $\max(0, \cdot)$ denotes the ReLU function.

To rebuild the feature maps, we utilize Upsample Block with $scalefactor = 2$. The structure of Upsample Block is shown in Figure 4, which contains a sub-pixel convolution layer proposed in [43]. Compared to deconvolutional layer, the sub-pixel convolution layer is faster. After every Upsample Block, there are also B Basic Blocks. With this hourglass structure, we significantly reduce computational complexity by 70% while maintaining excellent performance.

3.3. Memory Connection

In convolutional neural networks (CNN), the neurons of lower layers have small receptive field and focus more on local and detailed information. With the increase of network depth, the receptive field gets larger. The neurons of higher layers learn the global information, but high-frequency information gets saturated and degrades rapidly. Thus, merely increasing the network depth may not be a good solution for image restoration. In [20], He et al. propose residual learning to accelerate the training of deep networks, in which the shortcut connections are applied to every few stacked layers.

Inspired by neural science study that human brain will protect previously acquired knowledge in neurons, we novelly propose two kinds of memory connections to combine network output with residual information: local memory connection in Basic Blocks, which is shown in Figure 4 (the blue line), and global memory connection on the pipeline, shown in Figure 3 (the green line). The function of memory connection F_c can be formulated as

$$F_c(H_{in}) = H_{in} + F_{conv}(H_{in}), \quad (10)$$

where H_{in} is the residual information, and F_{conv} denotes the convolutional layers between the connection. Network with memory connections back-propagates gradients to former layers and accelerates the training process.

Instead of learning the mappings from observed degraded image to ground truth image directly, with the global connection, we combine image detail information in lower layers with global information in higher layers. The residual information in Basic Blocks denotes the short-term memory in this network, providing low-level image details for reconstruction.

Global and local memory connections add neither extra parameter nor computational complexity. Besides, it will back-propagate gradient to the bottom layers, accelerating the training process. We perform experiments in Section 4.4 to verify these effects.

3.4. Network Depth

DMCN has two key parameters: M , the number of Downsample Units or Upsample Units; and B , the number of Basic Blocks in every Unit. The input unit contains 7 convolutional layers, extracting the lower layer information of the input image. The output layer contains one convolutional layer. Given different M and B , we can train DMCN in different depths. The depth of DMCN is as follows:

$$depth = 7 + M \times (1 + 2 * B) + M \times (2 + 2 * B) + 1 = 8 + 3 * M + 4 * M * B. \quad (11)$$

It has been pointed out that increasing receptive field size can make use of the context information in a larger image region. Thus, high noise level in denoising and large-scale factor in SR would require

larger receptive field size to capture more context information. However, large receptive field usually require deep network with more parameter and heavier computational complexity. To balance the tradeoff between efficiency and performance, we did experiments in Section 4.2 to set a proper depth for our network.

3.5. Training strategies

Numerous training strategies have been proposed to accelerate convergence and boost performance [13,21,21]. In our work, we apply BN and PReLU for faster and better training performance. The effectiveness of them will be verified in Section 4.5.

Batch Normalization (BN). When training deep neural networks, since the parameters of the previous layers change, the distribution of each layer input also changes. Thus, network will be hard to converge with a large learning rate. This phenomenon is called internal covariate shift, which will slow down the training by requiring lower learning rates. We address this problem by applying normalization for each training batch. Batch Normalization allows larger learning rates and makes the network converges fast with better performance.

Parametric Rectified Linear Unit (PReLU). In the last few years, we have witnessed tremendous improvements in activation functions. Among them, Rectified Linear Unit (ReLU) [12] is one of several keys to the recent success of deep networks than conventional sigmoid-like units. In ReLU, if the input is negative, the output will be zero, which helps to generate a sparse representation.

To further improve the performance of ReLU, [21] propose a Parametric Rectified Linear Unit (PReLU) to improve model fitting with nearly zero extra computational cost and little overfitting risk. PReLU is defined as

$$f(x_i) = \begin{cases} x_i, & \text{if } x_i > 0 \\ a_i x_i, & \text{if } x_i \leq 0 \end{cases} \quad (12)$$

Here x_i is the input of the nonlinear activation f on the i_{th} channel, and a_i is a coefficient controlling the slope of the negative part. Figure 5 shows the comparison of ReLU and PReLU. When a_i become zero, ReLU is a special case of PReLU. In our network, we apply PReLU to improve accuracy at negligible extra parameters.

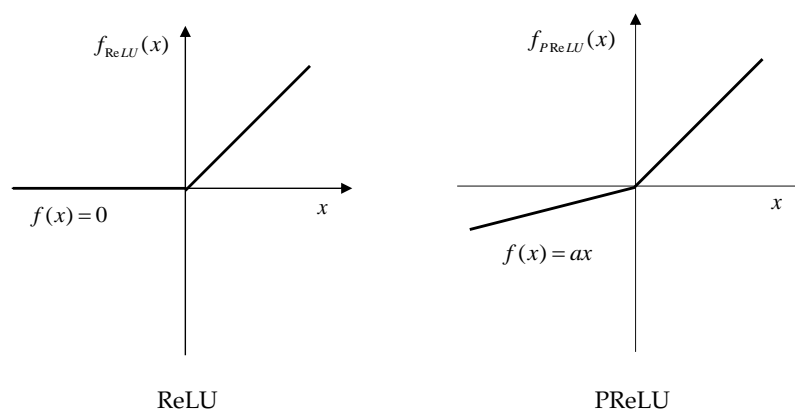


Figure 5. The comparison of ReLU and PReLU.

4. Experiment

In this section, we first introduce three datasets we used. Then we set a proper depth and width for our network. Three ablation experiments are performed to verify the effectiveness of Downsample Units, memory connections, BN and PReLU. Finally, experimental results on gaussian denoising and single image super-resolution are shown.

4.1. Dataset Sets and Environmental Configuration

We choose three datasets with different spatial resolutions to verify the robustness of our proposed method. Some of the training images are listed in Figure 6.

(1) **UCMERCED** [46]: The UC Merced land-use dataset is composed of 21 land-use scene classes with high spatial resolution (0.3 m/pixel) in the RGB color space. Each class consists of 100 aerial images measuring 256×256 pixels. We randomly select 80% of the dataset as training set and the others for testing.

(2) **NWPU-RESISC45** [19]: This dataset is a public benchmark created by Northwestern Polytechnical University (NWPU), which contains images with spatial resolutions varying from 30 m to 0.2 m per pixel. This dataset has 45 scenes with a total number of 31,500 images, 700 per class. The size of each image is 256×256 pixels. We randomly select 4500 images for training and 90 images for testing.

(3) **GaoFen1**: Multispectral images from GaoFen-1 satellite are also applied to our model. The three visible bands of the multispectral image (2 m/pixel) are extracted and stacked into pseudo-RGB image. We select 200 images measuring 512×512 pixels and divide it for training (160 images) and testing (40 images).

Given an input corrupted image Y , we optimize parameters $\Theta = \{W_i, b_i\}$ by minimizing the loss function between the ground truth HR image X and reconstructed image $\hat{X} = F(Y; \Theta)$. The loss function of DMCN is:

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^n |\hat{X}_i - X_i| \quad (13)$$

In this paper, we use the peak-signal-to-noise ratio (PSNR) [dB] and structural similarity index measure (SSIM) as criteria to evaluate the performance of DMCN. All the experiments are conducted on a computer with Intel Core i7, 16 GB of RAM and Nvidia Tesla K40 GPU, 12 GB of RAM.

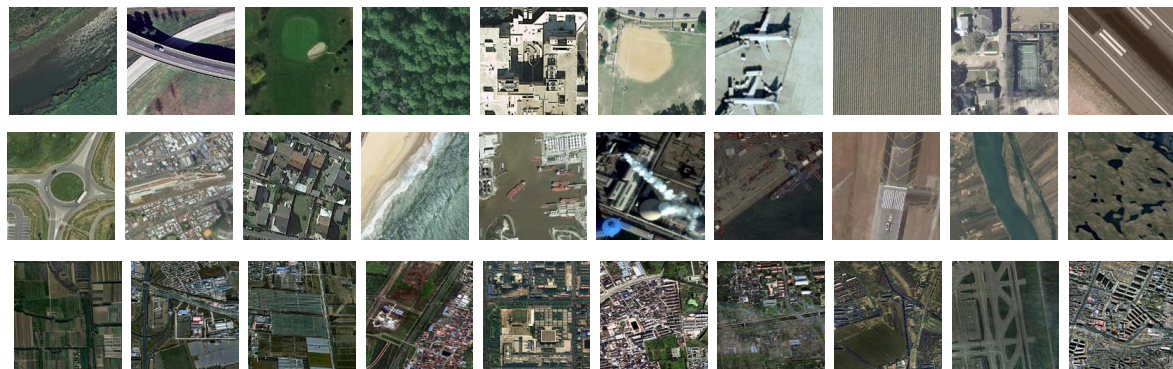


Figure 6. Examples of images in three datasets. UCMERCED dataset (first line) contains 21 land-use image classes such as river, forest, airplane and buildings. NWPU-RESISC45 dataset (second line) contains 45 land-use image classes such as beach, residential, mountain and storage tank. GaoFen1 dataset (third line) contains numerous images of city area.

4.2. Network Depth and Width

There are two parameters in network depth: M , the number of DownsampleUnit and UpsampleUnit; and B , the number of Basic Blocks in every Unit. We build different models with various combination of M and B to study the performance of different network depth.

In Figure 7, network depth ranges from 15 ($M = 1, B = 1$) to 113 ($M = 3, B = 8$). When M is 1, the PSNR grows up with B , because the network depth is growing from 15 layers to 36 layers. The receptive field is growing, making the network capability better. When M is 2, before B reaches 3, the model performance grows up with B since the deeper network has large receptive field, providing

more context to predict image details. After B reaches 4, the network is too deep to converge within 30 epochs, thus the model performance declines respectively. When M is 3, the network depth ranges from 29 layers to 113 layers. The network is too deep to converge. Besides, with three DownsampleUnits, the feature map will be shrunk by 8. A picture measuring 256×256 will be shrunk as a feature map measuring 32×32 , the detail information might be lost, thus the performance decrease.

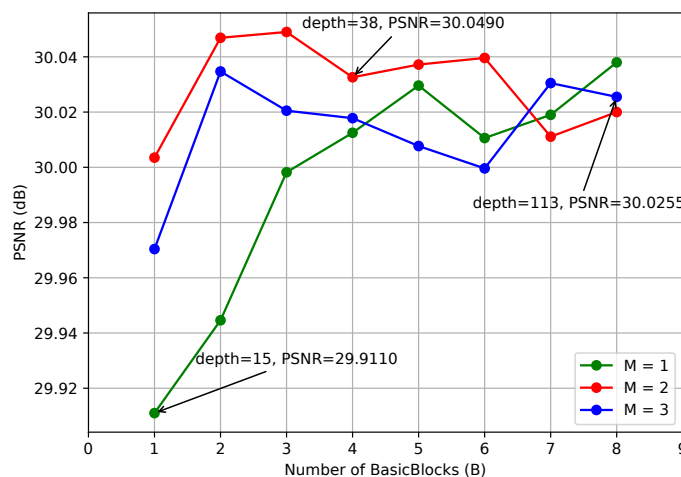


Figure 7. PSNR (dB) results of networks with different combination of M and B . The experiments are performed on denoising tasks on UCMERGED dataset with $\sigma = 25$. M is the number of DownsampleUnits and UpsampleUnits. B is the number of Basic Blocks in every unit.

Considering both the performance and speed, we choose $M = 2$, $B = 3$ ($depth = 38$) as our final model. Compared to DnCNN ($depth = 20$, receptive field = 43) and VDSR ($depth = 17$, receptive field = 37), our model DMCN ($depth = 38$, receptive field = 141) is the deepest network with the largest receptive field.

We also do experiments to verify the width of the network. The network is trained with network width 32, 64, 128, and 256, and the results are shown in Table 1. When network width is 32, the computational complexity is small but the performance is not satisfactory. When network width grows larger than 128, the computational complexity is too large and the model cannot achieve good results in 30 epochs. Thus we choose 64 as the width.

Table 1. PSNR of DMCN with network width 32, 64, 128, and 256. Time means the average time when processing an image measuring 256×256 .

Network Width	32	64	128	256
time(s)	0.005155	0.006224	0.007811	0.008375
PSNR (dB)	29.9917	30.0554	29.9951	29.9360

4.3. Evaluation on Downsample Unit and Upsample Unit

To evaluate the effect of Downsample Unit and Upsample Unit, we perform ablation experiments of super-resolution tasks on UCMERGED dataset. By setting $stride = 1$, the feature map will not be shrunk by Downsample Unit. By setting $upsaclefactor = 1$, the upsample unit is also disabled.

The result is shown in Table 2. Theoretically, network with Downsample Units will reduce computational complexity by 69.05%. In this experiment, without sacrificing performance, Downsample Units reduce the memory footprint of training by 53.4%. And the testing time is reduced by 67.6%. Overall, Downsample Units will significantly improve speed and reduce memory footprint while maintaining satisfactory performance.

Table 2. Evaluate the effect of Downsample Unit and Upsample Unit. Dis_D_U represents network without them. Memory Usage is measured in training process. Time means the average time processing an image measuring 256×256 . The experiment is performed on super-resolution tasks on UCMERGED dataset with upscale factor 2.

Model	Memory (MB)	Time (Sec)	PSNR
Dis_D_U	8265	0.037	34.17
DMCN (ours)	3849	0.012	34.19

4.4. The Effect of Memory Connection

To evaluate the effect of memory connections, we run an ablation study on them in turn and show the results in Figure 8. Network with all the memory connections converges fast and gets the best performance. When we remove global and local memory connections in turn, the results decay. Network without memory connections cannot even converge. In conclusion, memory connections can transfer low layer information to higher layers, making the reconstructed image more detailed. It can also back-propagate gradients, accelerating the convergence.

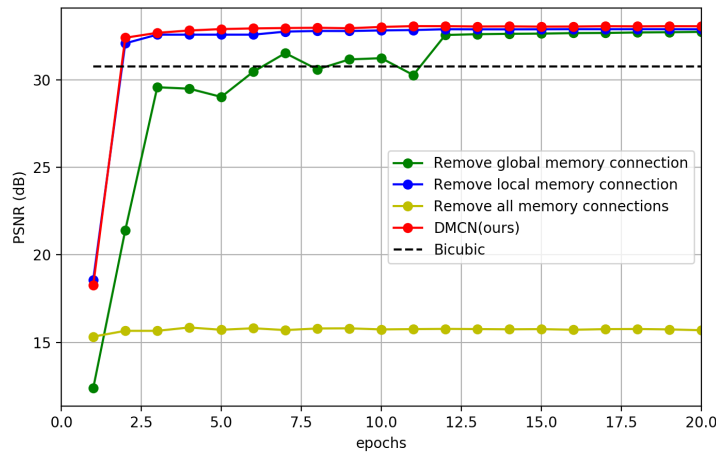


Figure 8. The ablation study of memory connection. The red line is the PSNR of DMCN, while the black line is the PSNR of bicubic. Green line represents network without global memory connections, while blue line represents network without local memory connections. If we remove all the connections, the network cannot converge. The experiment is performed on super-resolution tasks on UCMERGED dataset with upscale factor 2.

4.5. Batch Normalization and PReLU

In order to verify the effect of BN and PReLU, we do ablation study on denoising tasks with UCMERGED dataset. Figure 9 shows the PSNR results of networks with/without PReLU. With PReLU, the network can achieve faster convergence and relatively high performance. Besides, the testing time of network with PReLU is shorter.

As for Batch Normalization, Figure 10 shows the PSNR results of networks with/without BN. The network with BN and PReLU achieves the best PSNR, and we can set a large learning rate for this network. Networks without BN spend less time for testing, but with a large learning rate, it is hard to converge.

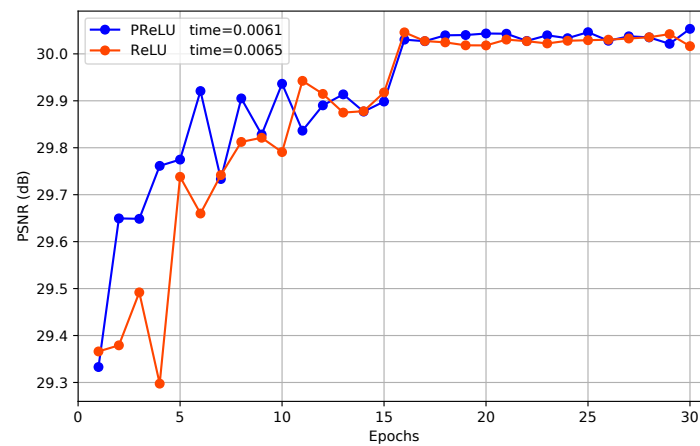


Figure 9. PSNR (dB) results of networks with PReLU (orange line), or without PReLU (blue line). Time means the average time when processing an image measuring 256×256 .

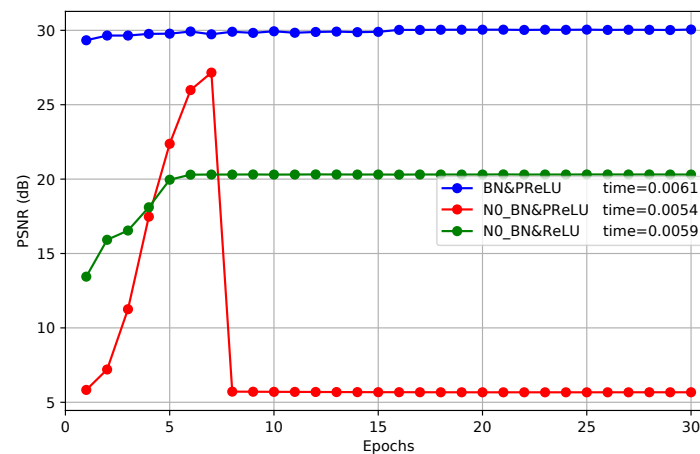


Figure 10. PSNR (dB) results of networks with BN (blue line) or without BN (red line and green line). Time means the average time of processing an image measuring 256×256 .

4.6. Gaussian Denoising

For denoising tasks, we usually assume the latent clean image x is corrupted by additive white Gaussian noise N . Thus the observed image would be formulated as $y = x + N$. In this paper, we consider five noise levels, i.e., $\sigma = 15, 25, 35, 45$ and 55 . Firstly, we train DMCN-S for Gaussian denoising with specific noise level. For one model, we use images with specific noise level to train and test. Then, we extend DMCN-B for blind noise level. We train DMCN-B with images from a wide range of noise levels (e.g., $\sigma = 15, 25, 35, 45$ and 55). Given a test image with unknown noise level, the single DMCN-B model can denoise it.

4.6.1. Training Details

When training DMCN-S model for specific noise level, we follow [11] to split the training data into 40×40 sub-images. Besides, we also train a single DMCN-B for denoise tasks (a single model for arbitrary noise level). The training images are split into 50×50 sub-images. Following previous works, we only denoise for grey images. The learning rate is initially set as $1e-3$ and decayed every 10 epochs by factor 10. We initialize the weights following [21] and use ADAM optimizer [22] by setting $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, weight_decay = 10^{-4}$. We optimize the loss function stated in Equation (13). Our work is compared with several state-of-the-art denoising methods such as non-local similarity based methods BM3D [5], WNNM [9] and CNN based model DnCNN [11].

4.6.2. Quantitative Results

We show the average PSNR and SSIM results of different methods on three datasets in Table 3. DMCN-S and DnCNN-S represent the networks trained for specific noise level σ , while DMCN-B and DnCNN-B is the network trained for blind σ . As can be seen, both DMCN-S and DMCN-B achieves the best PSNR results over other methods. Specifically, the superiority over the second best method DnCNN reaches 0.2 dB when $\sigma = 45$ and 55. DMCN also achieves the best SSIM over other methods, which indicates that our model can reconstruct images with better structure information. It should be noted that even a single DMCN-B model trained for blind noise level outperforms the DnCNN-S model trained for specific noise level.

Table 3. Evaluation of state-of-the-art denoising methods on three remote sensing datasets. We calculate the average PSNR/SSIM for noise level $\sigma = 15 - 55$. The **bold numbers** denote the best performance.

Dataset	\times	Noisy PSNR/SSIM	BM3D [5] PSNR/SSIM	DnCNN-B [11] PSNR/SSIM	DnCNN-S [11] PSNR/SSIM	DMCN-B (Ours) PSNR/SSIM	DMCN-S (Ours) PSNR/SSIM
GF	15	24.75/0.7942	30.03/0.9237	28.85/0.9082	30.57/0.9439	30.55/0.9446	30.60/0.9445
	25	20.49/0.7658	27.33/0.8852	28.01/0.9053	27.99/0.9049	28.09/0.9077	28.10/0.9069
	35	17.79/0.6221	25.58/0.8477	23.07/0.7448	26.37/0.8689	26.52/0.8731	26.52/0.8730
	45	15.86/0.4243	24.08/0.8221	18.79/0.5483	25.19/0.8356	25.39/0.8409	25.36/ 0.8413
	55	14.38/0.3884	22.94/0.7939	16.35/0.4301	24.29/0.8056	24.50/0.8121	24.46/ 0.8123
UC [46]	15	24.68/0.7928	31.80/0.9027	32.17/0.9427	32.30/0.9450	32.18/0.9435	32.38/0.9672
	25	20.32/0.7530	29.37/0.8932	29.88/0.9116	29.94/0.9132	30.01/0.9147	30.07/0.9155
	35	17.52/0.7094	27.81/0.8659	28.42/0.8853	28.45/0.8863	28.61/0.8896	28.64/0.9031
	45	15.50/0.6825	26.49/0.8504	27.33/0.8614	27.36/0.8627	27.54/ 0.8663	27.59/0.8649
	55	13.96/0.6177	25.47/0.8228	26.41/0.8390	26.49/0.8399	26.71/0.8467	26.73/0.8597
NW [19]	15	24.68/0.8059	31.44/0.9339	31.80/0.9332	31.91/ 0.9353	31.84/0.9351	31.98/0.9349
	25	20.33/0.7604	28.99/0.8827	29.49/0.8924	29.56/0.8943	29.60/0.8962	29.64/0.8961
	35	17.55/0.7139	27.50/0.8539	28.07/0.8575	28.13/0.8596	28.23/0.8626	28.23/0.8633
	45	15.57/0.6855	26.28/0.8255	27.07/0.8277	27.12/0.8305	27.22/0.8325	27.24/0.8337
	55	14.06/0.6272	25.35/0.8004	26.27/0.8040	26.33/0.8048	26.47/0.8085	26.49/0.8102

Besides, as is stated in Section 1, remote-sensing images are much more complex than ordinary images. Thus networks designed for ordinary images may not be able to meet the reconstruction needs of remotes sensing images. As can be seen in Table 3, DnCNN-B cannot remove noise for GaoFen1 dataset. The PSNR results are even worse than BM3D. The denoising results is shown in Figure 15, for image “Farmland028” with Gaussian noise $\sigma = 45$, DnCNN-B cannot remove noise successfully.

Figure 11 shows the average improvements of DnCNN-S, DnCNN-B, DMCN-S and DMCN-B over BM3D. Compared to the benchmark BM3D, DMCN-S and DMCN-B have a notable PSNR gain from 0.5 dB to 1.2 dB. Notably, the PSNR gain of DMCN over DnCNN and BM3D rise up with the noise level. When the noise level increase, the difficulty of denoising tasks go up exponentially. Since DMCN is a deep network with large receptive field and higher network capability, it is superior to other methods when dealing with large σ .

Besides, DMCN still have absolute advantages on PSNR when testing on different classes. In Table 4, we list the average PSNR improvements of DMCN-S and DnCNN-S over BM3D. The experiments are performed on seven image classes in NWPU-RESISC45 dataset. We choose images from these classes and show them in Figure 12. It should be noted that DnCNN is inferior to BM3D on image class “industrial”, while DMCN yields the highest PSNR on all the images. Which indicates that DMCN is more robust.

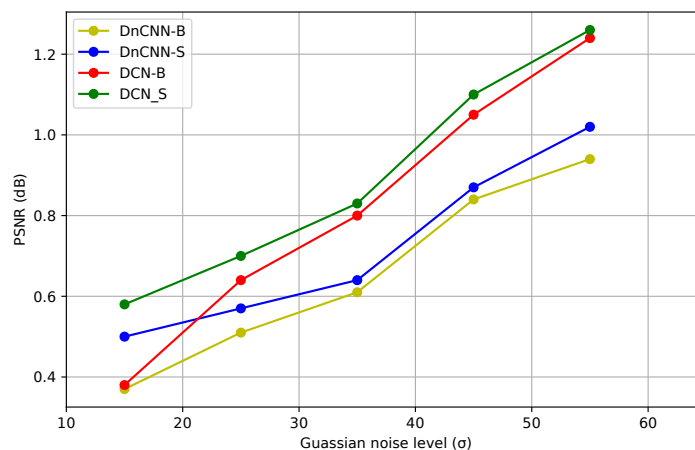


Figure 11. Average PSNR improvements of DnCNN-S, DnCNN-B, DMCN-S and DMCN-B over BM3D. The results are evaluated on the gray UCERCED dataset.

Some image classes with small PSNR difference is highlighted in bold. Remarkably, classes such as “farmland”, “industrial” and “meadow” are dominated by repetitive structures. This phenomenon is consistent with the fact that non-local based methods such as BM3D use repetitive images structures to reconstruct image detail, thus these methods perform better on images with repetitive content and fails on images with irregular textures. While our deep learning methods learn the potential image structure over the whole training dataset with hundreds of images, thus achieving convincing results on all the image calsses.

Table 4. The PSNR results of seven image classes in NWPU-RESISC45 dataset. DnCNN-BM3D represents the PSNR improvement of DnCNN-S over BM3D. DMCN-BM3D represents the PSNR improvement of DMCN-S over BM3D. The **bold numbers** represent Images classes with small PSNR differences.

Image Class	Airplane	Basketball-Court	Farmland	Residential	Industrial	Meadow	Stadium
DMCN-BM3D	0.9248	1.1370	0.5794	1.1951	0.2366	0.2386	1.0050
DnCNN-BM3D	0.7196	0.9766	0.3044	1.0714	−0.0581	0.2031	0.8294

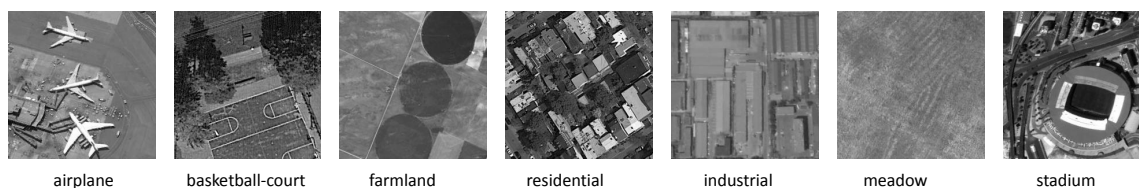


Figure 12. Representative image in seven image classes of Table 4.

4.6.3. Restored Image Quality

To further demonstrate the effectiveness of DMCN, we show the restored image in Figures 13–15. On the whole, it can be seen that BM3D tends to produce images with over-smoothed edges and textures. That’s because non-local based methods are designed to use repetitive image patches to reconstruct details [5]. In contrast, DnCNN is likely to recover sharp edges while ignoring image details. With deep network, DMCN has large receptive field. With local-global connections, the low-layer

information is transferred to higher layers. Thus, DMCN can keep a balance between removing noise and recovering image details. Our results reconstruct sharp edges and image details at the same time.

Particularly, in Figure 13, BM3D ignored some cracks on the runway. DnCNN lost the spot on the upper right corner of the white block area. Only DMCN learns the precise mapping from noisy image to the clean one. In Figure 14, the result of BM3D is over smoothed, thus we cannot clearly tell the outline of the plane. DnCNN loses some useful information such as the engine on the wing. DMCN recover these details and get images that is similar to the ground truth. In Figure 15, although all of the images seem similar, when we zoom in to the white block area, it can be seen that only DMCN produce straight and clear stripes. BM3D cannot distinguish those stripes, and DnCNN produces curved stipes.

4.7. Single Image Super-Resolution

For super-resolution problems, our task is to recover the latent clean image x from the down-scaled image y . Our model is compared with other methods including bicubic interpolation, the classic CNN-based SRCNN [10], LGCNet [47], and VDSR [37] (state-of-the-art).

4.7.1. Training Details

In the training phase, We first down-sample the images with scale factor $\times 2$, $\times 3$, $\times 4$. Then, the ground truth images $\{X_i\}$ are split into 48×48 sub-images with no overlap. We use a mini-batch size of 128 when training. Following previous works, we only consider the luminance channel in YCbCr color space, because humans are more sensitive to luminance changes. We use the initialization scheme described in [21] for all layers. We train our model with ADAM [22] optimizer by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, $weight_decay = 10^{-4}$. The learning rate is initialized as 10^{-3} and decreased every ten epochs by a factor of 10. To augment the training data, we make two operations on them: (1) Flipping: flip images horizontally or vertically with a probability of 0.5. (2) Rotation: randomly rotate images by 90° , 180° , or 270° . Our learning rate is initially set to 5×10^{-4} and decreased every ten epochs by factor 10. We train our model for super-resolution with scale factor $\times 2$, $\times 3$, $\times 4$ respectively.

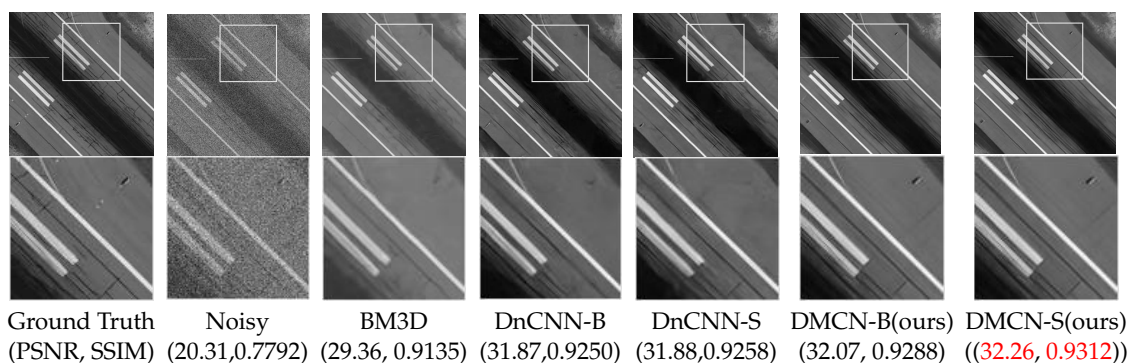


Figure 13. Denoising results of “Runway82” (UCMERCED) with noise level $\sigma = 25$. BM3D ignored some cracks on the runway. DnCNN lost the spot on the upper right corner of the white block area. Only DMCN learns the precise mapping from noisy image to the clean one.

4.7.2. Quantitative Results

Our method is compared with the state-of-the-art methods such as Bicubic [48], neighbour embedding based method NE+NNLS [3], adjusted anchored neighborhood regression A+ [49], and deep learning based methods SRCNN [10], VDSR [37], and LGCNet [47]. We measure PSNR and SSIM on the luminance channel. Table 5 shows the quantitative evaluation results of several methods for $\times 2$, $\times 3$ and $\times 4$ SR. DMCN outperforms all these methods with the highest PSNR and SSIM.

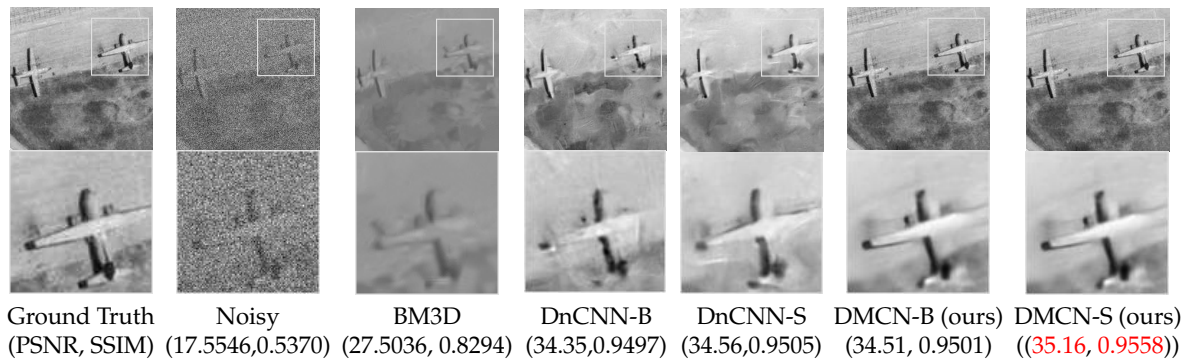


Figure 14. Denoising results of “Airplane079” (NWPU-RESISC45) with noise level $\sigma = 35$. The result of BM3D is smoothed, and the outline of the plane is unclear. DnCNN lost some useful information such as the engine on the wing. DMCN recover these details and gets images similar to the ground truth.

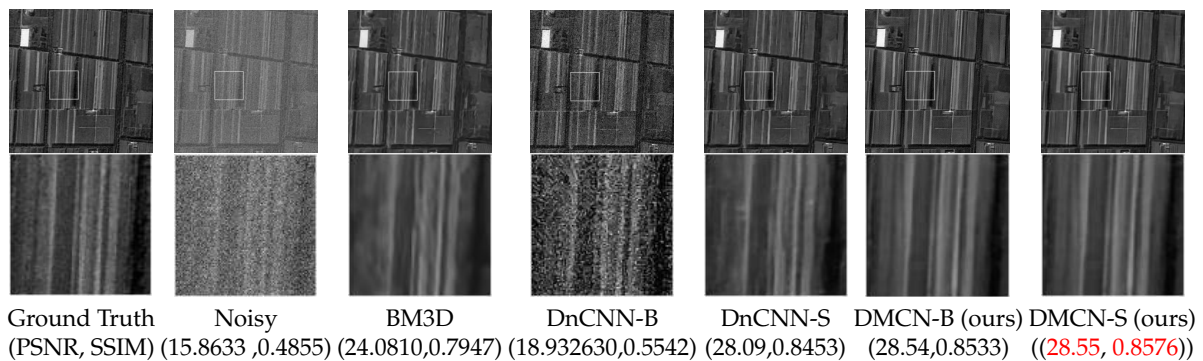


Figure 15. Denoising results of “Farmland028” (GaoFen1) with noise level $\sigma = 45$. In this image, BM3D cannot distinguish those stripes, and DnCNN produces some curved stipes. Only DMCN produces straight and clear stripes.

Table 5. Evaluation of the state-of-the-art SR methods on remote sensing datasets NWPU-RESISC45, UC Merced, and GaoFen1. We calculate the average PSNR/SSIM for scale factor $\times 2$, $\times 3$ and $\times 4$. The **bold number** denotes the best performance.

Dataset	Scale	Bicubic [48] PSNR/SSIM	A + [49] PSNR/SSIM	NE + NNLS [3] PSNR/SSIM	SRCNN [10] PSNR/SSIM	VDSR [37] PSNR/SSIM	LGCNet [47] PSNR/SSIM	DMCN (Ours) PSNR/SSIM
NWPU-RESISC45	$\times 2$	30.77/0.8172	30.86/0.8223	30.94/0.8191	29.37/0.7598	32.77/0.8778	32.76/0.8770	33.07/0.8842
	$\times 3$	27.86/0.6405	27.92/0.6493	27.98/0.6526	27.94/0.6545	29.28/0.7165	29.21/0.7163	29.44/0.7251
	$\times 4$	26.30/0.4970	26.41/0.4996	26.47/0.5057	26.52/0.5252	27.30/0.5549	27.32/0.5633	27.52/0.5858
UC Merced	$\times 2$	31.08/0.8316	31.17/0.8482	31.32/0.8530	31.06/0.8428	33.79/0.8909	33.80/0.8817	34.19/0.8941
	$\times 3$	27.59/0.6557	27.74/0.6763	27.99/0.6898	28.24/0.6998	29.63/0.7359	29.62/0.7350	29.86/0.7454
	$\times 4$	25.72/0.5800	25.91/0.5512	25.98/0.5547	26.07/0.5439	27.31/0.5850	27.29/0.5763	27.57/0.6150
GaoFen1	$\times 2$	26.88/0.8585	26.93/0.8681	27.09/0.8896	26.98/0.8727	29.23/0.9155	29.14/0.9084	29.26/0.9250
	$\times 3$	23.30/0.7263	23.56/0.7276	23.79/0.7261	23.83/0.7264	24.65/0.7631	24.63/0.7602	24.76/0.7658
	$\times 4$	21.48/0.5039	21.60/0.5244	21.74/0.5470	21.78/0.5474	22.31/0.5879	22.23/0.5834	22.38/0.6031

4.7.3. Restored Image Quality

Except for quantitative results, the visual comparisons are shown in Figures 16–18. Viewing the whole picture, we observe that our method accurately reconstruct clear images. When zoom in the yellow window, our method successfully reconstruct the detailed textures, shapes and edges.

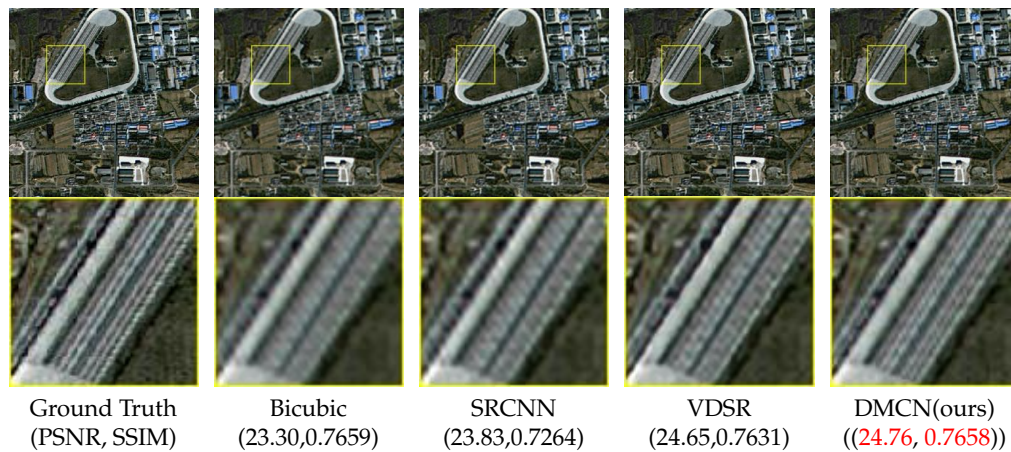


Figure 16. Super-resolution results of “City163” (*GaoFen1*) with scale factor $\times 3$. The stripe in the ground truth is also observed in our result, while it is not distinguished in other results.

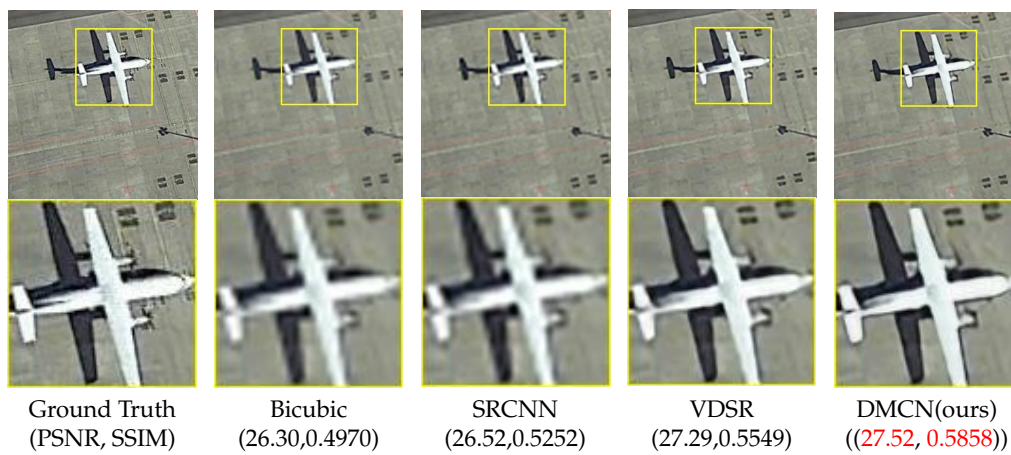


Figure 17. Super-resolution results of “airplane327” (*NWPU-RESISC45*) with scale factor $\times 3$. The airplane in our result has clear edges and more detailed information.

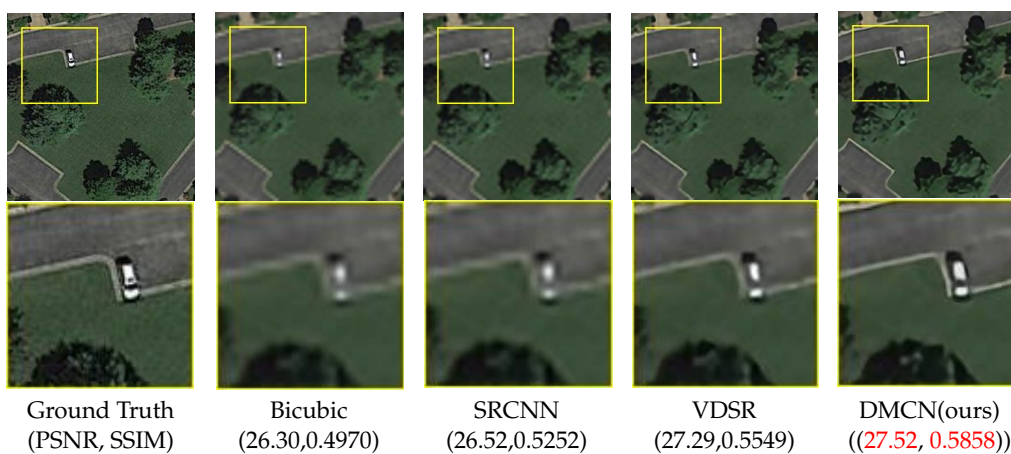


Figure 18. Super-resolution results of “meadow683” (*NWPU-RESISC45*) with scale factor $\times 4$. The outline of the car is distinct in the result of DMCM, while in other works the car is very blurry.

5. Conclusions

In this paper, we have proposed a novel deep memory connected neural network (DMCM) for remote sensing image restoration. DMCM is a deep network with large receptive field as well as

good reconstruction capability. We use memory connections to combine image detail with global information. To further reduce the computational complexity and memory footprint, we propose Downsample Units to shrink the spatial size of feature map. DMCN can achieve high-quality remote sensing image super-resolution and image denoising for specific or blind noise level. Our model is trained and tested on three benchmark datasets with various spatial resolution. Experiments show that DMCN achieves robust results and outperforms the current state-of-the-art by a large margin regarding visual quality and accuracy.

Although the proposed approach results are encouraging as a novel restoration model in remote sensing, the method still has some limitations. Though the network can deal with images measuring 256×256 in 0.006 s on GPUs, when dealing with large quantity of dataset, the network ability is still limited by computational complexity. Besides, when training the network, we need to use corrupted observation as well as the clean signals. However, the clean images are usually unobserved in the real world. Specifically, our future work will focus on the following aspects: (1) further shrinking the network computational complexity and maintain good performance by some operations on network width, depth, and filter size; (2) learning to turn corrupted images into clean images by only looking at observed images.

Author Contributions: W.X. and D.L. conceived and designed the experiments; W.X. performed the experiments; W.X. analyzed the data; G.X., X.S., Y.W. (Yang Wang), and Y.W. (Yirong Wu) contributed materials; W.X. wrote the paper. Y.W. (Yang Wang), and Y.W. (Yirong Wu) supervised the study and reviewed this paper.

Funding: This research received no external funding

Acknowledgments: Our code is available at <https://github.com/wenjiaXu/Optical-RemoteSensing-Image-Resolution>.

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

References

1. Haut, J.M.; Fernandez-Beltran, R.; Paoletti, M.E.; Plaza, J.; Plaza, A.; Pla, F. A New Deep Generative Network for Unsupervised Remote Sensing Single-Image Super-Resolution. *IEEE Trans. Geosci. Remote Sens.* **2018**. [CrossRef]
2. Huang, Z.; Zhang, Y.; Li, Q.; Zhang, T.; Sang, N.; Hong, H. Progressive Dual-Domain Filter for Enhancing and Denoising Optical Remote-Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 759–763. [CrossRef]
3. Chang, H.; Yeung, D.Y.; Xiong, Y. Super-resolution through neighbor embedding. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004; Volume 1, p. I.
4. Freeman, W.T.; Jones, T.R.; Pasztor, E.C. Example-based super-resolution. *IEEE Comput. Gr. Appl.* **2002**, *22*, 56–65. [CrossRef]
5. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. [CrossRef] [PubMed]
6. Elad, M.; Aharon, M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.* **2006**, *15*, 3736–3745. [CrossRef] [PubMed]
7. Dong, W.; Zhang, L.; Shi, G.; Wu, X. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Trans. Image Process.* **2011**, *20*, 1838–1857. [CrossRef] [PubMed]
8. Yair, N.; Michaeli, T. Multi-Scale Weighted Nuclear Norm Image Restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–21 June 2018; pp. 3165–3174.
9. Gu, S.; Zhang, L.; Zuo, W.; Feng, X. Weighted nuclear norm minimization with application to image denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2862–2869.
10. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 184–199.

11. 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.* **2017**, *26*, 3142–3155. [[CrossRef](#)] [[PubMed](#)]
12. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
13. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
14. Haris, M.; Shakhnarovich, G.; Ukita, N. Deep backprojection networks for super-resolution. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Beijing, China, 20 August 2018.
15. Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; Fu, Y. Residual dense network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
16. Lim, B.; Son, S.; Kim, H.; Nah, S.; Lee, K.M. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, HI, USA, 21–26 July 2017; Volume 1, p. 4.
17. Lefkimmiatis, S. Universal Denoising Networks: A Novel CNN Architecture for Image Denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3204–3213.
18. Chen, J.; Chen, J.; Chao, H.; Yang, M. Image Blind Denoising With Generative Adversarial Network Based Noise Modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3155–3164.
19. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [[CrossRef](#)]
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
21. 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, Santiago, Chile, 13–16 December 2015; pp. 1026–1034.
22. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
23. Xu, S.; Zhou, Y.; Xiang, H.; Li, S. Remote Sensing Image Denoising Using Patch Grouping-Based Nonlocal Means Algorithm. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2275–2279. [[CrossRef](#)]
24. Kwan, C.; Zhou, J. Method for Image Denoising. Patent 9,159,121, 13 October 2015.
25. Chen, Y.; Guo, Y.; Wang, Y.; Wang, D.; Peng, C.; He, G. Denoising of hyperspectral images using nonconvex low rank matrix approximation. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5366–5380. [[CrossRef](#)]
26. Fan, Y.R.; Huang, T.Z.; Zhao, X.L.; Deng, L.J.; Fan, S. Multispectral Image Denoising via Nonlocal Multitask Sparse Learning. *Remote Sens.* **2018**, *10*, 116. [[CrossRef](#)]
27. Loncan, L.; Almeida, L.B.; Bioucas-Dias, J.M.; Briottet, X.; Chanussot, J.; Dobigeon, N.; Fabre, S.; Liao, W.; Licciardi, G.A.; Simoes, M.; et al. Hyperspectral pansharpening: A review. *arXiv* **2015**, arXiv:1504.04531.
28. Vivone, G.; Alparone, L.; Chanussot, J.; Dalla Mura, M.; Garzelli, A.; Licciardi, G.A.; Restaino, R.; Wald, L. A critical comparison among pansharpening algorithms. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2565–2586. [[CrossRef](#)]
29. Kwan, C.; Choi, J.; Chan, S.; Zhou, J.; Budavari, B. A super-resolution and fusion approach to enhancing hyperspectral images. *Remote Sens.* **2018**, *10*, 1416. [[CrossRef](#)]
30. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.
31. Jain, V.; Seung, S. Natural image denoising with convolutional networks. In Proceedings of the 21st International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–11 December 2008; pp. 769–776.
32. Chen, Y.; Pock, T. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1256–1272. [[CrossRef](#)] [[PubMed](#)]

33. Mildenhall, B.; Barron, J.T.; Chen, J.; Sharlet, D.; Ng, R.; Carroll, R. Burst Denoising with Kernel Prediction Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2502–2510.
34. Lehtinen, J.; Munkberg, J.; Hasselgren, J.; Laine, S.; Karras, T.; Aittala, M.; Aila, T. Noise2Noise: Learning Image Restoration without Clean Data. *arXiv* **2018**, arXiv:1803.04189.
35. Putzky, P.; Welling, M. Recurrent inference machines for solving inverse problems. *arXiv* **2017**, arXiv:1706.04008.
36. Dong, C.; Loy, C.C.; Tang, X. Accelerating the super-resolution convolutional neural network. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 391–407.
37. Kim, J.; Kwon Lee, J.; Mu Lee, K. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1646–1654.
38. Yuan, Y.; Zheng, X.; Lu, X. Hyperspectral image superresolution by transfer learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 1963–1974. [[CrossRef](#)]
39. Qu, Y.; Qi, H.; Kwan, C. Unsupervised Sparse Dirichlet-Net for Hyperspectral Image Super-Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2511–2520.
40. Shocher, A.; Cohen, N.; Irani, M. Zero-Shot super-resolution using deep internal learning. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Beijing, China, 20 August 2018.
41. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.P.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the CVPR, Honolulu, HI, USA, 22–25 July 2017; Volume 2, p. 4.
42. Sajjadi, M.S.; Vemulapalli, R.; Brown, M. Frame-Recurrent Video Super-Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6626–6634.
43. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
44. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing And Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
45. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 483–499.
46. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference On Advances in Geographic Information Systems, San Jose, CA, UAS, 3–5 November 2010; pp. 270–279.
47. Lei, S.; Shi, Z.; Zou, Z. Super-Resolution for Remote Sensing Images via Local-Global Combined Network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1243–1247. [[CrossRef](#)]
48. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 1153–1160. [[CrossRef](#)]
49. Timofte, R.; De Smet, V.; Van Gool, L. A+: Adjusted anchored neighborhood regression for fast super-resolution. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014; pp. 111–126.

