Double Prior Network for Multidegradation Remote Sensing Image Super-Resolution

Mengyang Shi, *Student Member, IEEE*, Yesheng Gao, *Member, IEEE*, Lin Chen, *Student Member, IEEE*, and Xingzhao Liu, *Member, IEEE*

***Abstract*—Image super-resolution (SR) is widely used in remote sensing because it can effectively increase image details. Neural networks have shown remarkable performance in recent years, benefitting from their end-to-end training. However, remote sensing images contain a variety of degradation factors. Neural networks lack flexibility in dealing with these complex issues compared with reconstruction-based approaches. Traditional neural network methods cannot take advantage of prior knowledge and lack interpretability. To develop a flexible, accurate, and interpretable algorithm for remote sensing SR, we proposed an effective SR network called YSRNet. It is performed by unfolding a traditional optimization process into a learnable network. Combining conventional reconstruction-based methods and neural networks can significantly improve the algorithm's performance. Since the gradient features of remote sensing images contain valuable information, the total variation constraints and the deep prior constraints are introduced into the objective function for image SR. Furthermore, we propose an enhanced version called YSRNet+, which can apply attention weights to different prior terms and channels. Compared with the YSRNet, the YSRNet+ enables networks to focus more on useful prior information and improve the interpretability of networks. Experiments on three remote sensing datasets are performed to evaluate the algorithm's effectiveness. The experimental results demonstrate that the proposed algorithm performs better than some state-of-the-art neural network algorithms, especially in the scenario of the multidegradation factors.**

***Index Terms*—Double prior, interpretability, multidegradation, remote sensing images, super-resolution (SR), total variation (TV), unfolding.**

I. Introduction

REMOTE sensing images are widely used in target recognition and detection [1], [2], [3], [4], [5], [6], land classification [7], [8], [9], [10], [11], [12], resource exploration [13], etc. However, due to remote sensing images mostly covering a large area of the ground, the image resolution is usually low, degrading real-world applications' performance. Such problem can be improved by two aspects: hardware [14], [15], [16], [17] and algorithms [18], [19], [20], [21]. Considering the sharp increase in cost by improving resolution through hardware, it is more widely used to optimize algorithms for remote sensing images super-resolution (SR).

The original remote sensing images degrade to varying degrees due to natural noise, motion blurring, and hardware limitations, resulting in loss of information [22], [23], [24]. These three degradation factors constitute a task of multidegradation image SR. Image SR aims to reconstruct a high-quality image from one or more low-quality images. SR algorithms need to restore as much detail as possible and present target information. This process is the inversion of image degradation. Degradation models can be represented as

$$y = Ax + n \tag{1}$$

where A denotes the comprehensive effect of the downsampling matrix and the blurring kernel. x and y represent high-quality images and low-quality images, respectively. n is white Gaussian noise.

Remote sensing image SR is divided into single-image SR (SISR) [25], [26], [27] and multi-image SR (MISR) [28], [29], [30]. MISR algorithms utilize the relationships between multiple images to obtain additional information. Neural networks have also been utilized for MISR. DeepSUM is a deep neural network for unregistered multitemporal images SR, which exploits both spatial and temporal correlations to combine multiple images [31]. Liu et al. proposed a novel MISR network called progressive multiscale deformable residual network. It aimed to improve the spatial resolution of sea ice passive microwave images, according to the characteristics of both passive microwave images and sea ice motions [32]. However, compared with MISR methods, SISR algorithms are more widely used. It is a more direct and effective image quality improvement method with no need for other information. SISR can be divided into interpolation-based, reconstruction-based, and learning-based methods. Interpolation-based methods include nearest neighbor interpolation, bilinear interpolation, cubic interpolation, etc. Some scholars proposed improved interpolation-based SR algorithms such as [33], [34], [35].

Reconstruction-based methods aim to study the degradation process from high-resolution (HR) images to low-resolution (LR) images. For instance, Li et al. proposed maximum a posteriori based on a universal Hidden Markov Tree (HMT) model for remote sensing image SR [36]. The HMT theory sets up a prior model to reconstruct SR images from a sequence of warped, blurred, subsampled, and noise-contaminated LR images. And

Manuscript received 22 March 2022; revised 8 August 2022 and 22 November 2022; accepted 22 January 2023. Date of publication 6 February 2023; date of current version 5 April 2023. This work was supported by the National Key R&D Program of China (Grant No. 2021YFA0715400). *(Corresponding author: Yesheng Gao.)*

The authors are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: cjpp944@sjtu.edu.cn; ysgao@sjtu.edu.cn; linchenee@sjtu.edu.cn; Liu,xzhliu@sjtu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3242053

Schultz et al. [37] proposed a Bayesian approach method to reconstruct LR images by maximum a posteriori estimation.

Learning-based methods are mainly based on sparse representation and neural networks. Yang et al.[38] first proposed an image SR method based on sparse representation. This method establishes the relationship between LR and HR images for image SR by learning a redundant dictionary. Then, Zheng et al. [39] first applied the sparse representation to remote sensing image SR. After that, Hou et al. [40] proposed a sparse representation and global union dictionary model. It utilizes nonlocal self-similarity to obtain global constraints and improve the performance of image SR.

In recent years, neural networks have been widely used for image SR. Dong et al. [41] introduced a convolution neural network into image SR in 2015 and achieved good results. Liebe et al. [42] proposed SRCNN network for remote sensing image SR. Furthermore, Dong et al. [43] proposed the FSRCNN based on SRCNN. After that, Lei et al. [44] suggested the LGCnet. The LGCnet exchanges local and global information by cascading shallow and deep feature mappings. Since He et al. [45] proposed residual network, it has been widely used for SR tasks. Kim et al. [46] proposed VDSR algorithm, which solves the problem that deep networks are challenging to train. For remote sensing images, Haut et al. [47] proposed an attention mechanism called RSRCAN, which can improve network performance with a small number of calculations.

Recently, an unfolding method combining traditional optimization algorithms with neural networks has been proposed for image reconstruction [48], [49], [50], [51], [52], [53]. It unfolds the traditional optimization process into a network, and the parameters in the algorithm can be obtained through training without artificial design. Zhang et al. [54] proposed USRNet for multidegradation tasks and obtained good results. However, in [54], it only uses the deep prior information for image SR. Luo et al. proposed DAN network for image SR. The algorithm completes kernel estimating and image restoring through an end-to-end learning process, which improves the accuracy of the algorithm.

Different algorithms have their characteristics. Fast reconstruction is the advantage of interpolation-based SR algorithms while the performance is limited. The reconstruction-based methods and sparse representation can fully use the prior information, but they also have the drawbacks of a large number of calculations. The neural network has a good performance. However, it lacks interpretability and is not flexible enough to utilize the traditional experience and knowledge due to its unique structure [54]. The deep unfolding network has a strong fitting ability of the neural network and is flexible to utilize prior knowledge. Thus, the unfolding method can handle multidegradation tasks well. However, as far as we know, the unfolding method has not been widely used in remote sensing. And some hyperparameters and prior information are not well combined with neural networks.

The number of remote sensing images is limited for remote sensing image SR. The interpretability is essential for remote sensing SR algorithms. Moreover, remote sensing SR contains multidegradation factors, which requires the algorithm to be highly adaptable. To improve the algorithm's flexibility and increase its interpretability, we propose YSRNet for multidegradation remote sensing image SR. Compared with the traditional neural network method for remote sensing image SR, we utilize the variable splitting method to unfold the optimization process into a network, which significantly improves the algorithm's performance while still maintaining its interpretability. In addition, most deep unfolding algorithms have only one prior term. Thus, plenty of the knowledge in traditional optimization algorithms has not yet been utilized. Considering that the gradient domain of remote sensing images contains plenty of details [55], [56], [57], we utilize total variation (TV) priori for image SR. TV feature is essentially the image gradient features along row direction and column direction. For an image, the edge gradient contains more information than the smooth region. Therefore, we extract the gradient information separately, making it easier for the network to obtain and facilitate image reconstruction. And this approach also increases the number of feature maps in the network. The network consists of two interpretable modules, which perform different functions, respectively. TV features are introduced into the network as the prior knowledge. Furthermore, we propose an enhanced version of YSRNet+ by introducing an attention module. Unlike traditional attention modules, this attention module can also assign weights to different prior constraint terms. The attention weights can be interpreted as the penalty factors for additional prior terms in traditional algorithms. It improves the network performance while further increasing the interpretability of the network. It is worth noting that this improvement only requires a small amount of computation. The main contributions of this article are summarized as follows.

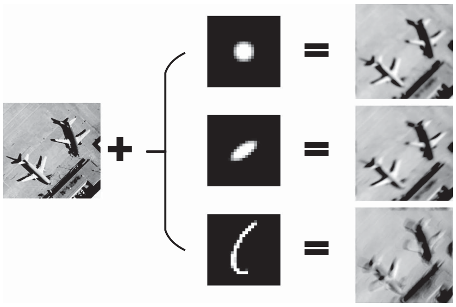


Fig. 1. Effects of different blurring kernels on images. Column 1–3 denote the isotropic Gaussian kernel, the anisotropic Gaussian kernel and the motion blurring kernel, respectively.

1) In this article, we propose a deep unfolding strategy aiming at a new optimization problem, which consists of a reconstruction term, a deep prior term, and a TV prior term. We map the optimization algorithm to a network and build a bridge for TV priori and neural networks.
2) A new network module called Ynet is designed to perform the noise removal task. TV priori are introduced into this

module for feature enhancement, which can help the network extract image information more efficiently. Then, by introducing the attention module, we propose an enhanced version, which has a more reasonable explanation for combining the hyperparameters with neural networks. At the same time, it can make the network pay more attention to essential feature maps.

3) Considering various degradation factors, we implement image SR under comprehensive factors, which makes the task more universal.

The rest of this article is organized as follows. In Section I, we briefly introduce the degradation models and principles of the deep unfolding networks. We propose the remote sensing SR algorithm in Section II and the enhanced version is described in Section III. Section IV shows the experimental results and analysis. Finally, Section V concludes this article.

II. Related Work

A. Degradation Models

In the process of remote sensing image acquisition, the image is usually accompanied by various degradation factors due to the influence of equipment and the environment. This article considers several typical degradation factors, including image blurring, random noise, and image downsampling, to make the SR model realistic. Image blurring is divided into the following three categories: isotropic Gaussian blurring, anisotropic Gaussian blurring, and motion blurring. Motion blurring is caused by the relative motion of the aircraft and ground, and the Gaussian blur is caused by the inaccurate focus of the camera and atmospheric turbulence. Hence, the degradation model can be expressed as: $y = (k \otimes x_h)_{\downarrow S} + n$, where n denotes the white Gaussian noise. The loss function can be expressed as follows:

$$\text{loss} = \arg\min_x \left\| (k \otimes x)_{\downarrow S} - y \right\|_F^2 \tag{2}$$

where x_h denotes the HR image, k denotes the blurring kernels, and S is the downsampling factor, usually being 2, 3, or 4. y denotes the observed LR image. Fig. 1 shows various kinds of blurring kernels, where (a), (b), and (c) denote the isotropic Gaussian kernel, anisotropic Gaussian kernel, and motion blurring kernel, respectively.

The blurring kernel makes the image structure overlap. The noise can bring fake information, and the downsampling can cause the loss of image details. In this article, we perform image SR under these three degradation conditions. This article contains isotropic Gaussian kernels of various sizes, anisotropic Gaussian kernels of multiple directions, and motion blurring kernels of different tracks.

B. Deep Unfolding Networks

Unfolding methods are the combination of reconstruction-based and learning-based algorithms. The principle of deep unfolding is to decompose complex problems into simple subproblems. Different subproblems are solved independently. It uses traditional methods to solve the subproblems such as deblurring and scaling because these problems have closed-form solutions. There is no closed-form solution for the noise removal problem, and the noise removal results of traditional reconstruction-based methods are limited. Thus, we utilize neural networks to complete the noise removal task. The calculation process of the closed-form solution and the denoising networks constitute the unfolding network. Each part of the unfolding network has definite functions and meanings. Different closed-form solutions can solve problems under different conditions. Therefore, networks have great flexibility and interpretability. It is suitable for solving multidegradation issues and problems with high interpretability requirements, such as remote sensing image SR.

III. Proposed Scheme

Based on the degradation model in Section II-A, this unfolding network can be divided into two parts, i.e., the establishment of the optimization model and the design of unfolding networks. Most unfolding methods use only one kind of prior information as the regularization term in the objective function. Equation (3) shows the basic unfolding networks, where $R(x)$ denotes the image degradation process and $\varphi(x)$ denotes the prior information of images, such as sparse priori or deep priori. Only using one kind of prior information cannot fully mine the information of the images. Therefore, we propose reconstructing the target image with more than one kind of prior information in the proposed network. The utilization of prior information essentially adds more input features to neural networks. This method improves the network performance and provides a bridge between the traditional prior term and the neural networks

$$\arg\min_x \|R(x) - y\|_F^2 + \varphi(x). \tag{3}$$

It is worth noting that the article aims to solve nonblind remote sensing image SISR. Blurring kernels can be estimated by other estimation algorithms or hardware devices.

In this section, we first introduce the decomposition of the multidegradation problem and the solving methods of the subproblems. Then, this article shows the design of the deep unfolding network framework. Finally, the algorithm flow chart of unfolding networks based on end-to-end training is presented.

A. Optimization Methods

Among traditional optimization algorithms, TV regularization is a remarkable prior term. It utilizes the gradient information of the target images, which can make the results of the SR reconstruction clearer. Combining with (2), the objective function can be expressed as

$$\arg\min_x \left\| (k \otimes x)_{\downarrow S} - y \right\|_F^2 + \alpha \cdot \varphi(x) + \beta \cdot \psi_{\text{TV}}(x) \tag{4}$$

where the first term denotes the reconstruction term, $\varphi(x)$ denotes the deep prior term, and $\psi_{\text{TV}}(x)$ denotes the TV prior term. And we have $\psi_{\text{TV}}(x) = \sum_1^w (x_{i+1} - x_i) + \sum_1^h (y_{j+1} - y_j)$, where h and w denote the height and width of the image, respectively. α and β are the tradeoff parameters. Since (4) contains implicit terms, we cannot solve it directly. Thus, we can use variable splitting algorithms to solve it such as ADMM [58]. By introducing the auxiliary variables z and m, (4) can be redefined as

$$\begin{aligned}&\arg\min_{x} \left\|(k\otimes x)_{\downarrow S} - y\right\|_F^2 + \alpha\cdot\varphi(z) + \beta\cdot\psi_{\mathrm{TV}}(m)\\ &\text{s.t.}\quad z = x, m = x.\end{aligned} \tag{5}$$

Therefore, Lagrange function can be defined as

$$\begin{aligned}L_{\substack{\alpha,\beta,\mu_1,\\ \mu_2,\mu_3}}(x,z,m) = &\left\|y-(k\otimes x)_{\downarrow S}\right\|_F^2 + \alpha\cdot\varphi(z)\\ &+\beta\cdot\psi(m) + \tfrac{\mu_1}{2}\cdot\|x-z\|_F^2\\ &+\tfrac{\mu_2}{2}\cdot\|z-m\|_F^2 + \tfrac{\mu_3}{2}\cdot\|x-m\|_F^2\end{aligned} \tag{6}$$

where $\|x-z\|_F^2$, $\|z-m\|_F^2$, and $\|x-m\|_F^2$ denote the penalty functions to guarantee the variable z, m, and x being approximate. μ_1, μ_2, and μ_3 are the corresponding penalty parameters, which should be large enough to guarantee the convergence of the algorithm. Here, z and m denote the reconstructed images obtained through different kind of prior knowledge in traditional optimization-based methods. Different priori terms can impose different constraints on the final image. Finally, through the dual rising method, we can make the auxiliary variables close enough to the solution variables to get the final reconstructed images. Such problem can be solved by calculating x-subproblem, z-subproblem, and m-subproblem alternately. Therefore, it can be divided into these three subproblems

$$\begin{aligned}x_{k+1} =\ &\arg\min\left\|y_{k+1}-(k\otimes x_k)_{\downarrow S}\right\|_F^2 + \tfrac{\mu_1}{2}\|x_k - z_k\|_F^2\\ &+\tfrac{\mu_3}{2}\|x_k - m_k\|_F^2\end{aligned} \tag{7}$$

$$\begin{aligned}z_{k+1} =\ &\arg\min \tfrac{\mu_1}{2}\|z_k - x_{k+1}\|_F^2 + \alpha_{k+1}\cdot\varphi(z_k)\\ &+\tfrac{\mu_2}{2}\|z_k - M_k\|_F^2\end{aligned} \tag{8}$$

$$\begin{aligned}m_{k+1} =\ &\arg\min \tfrac{\mu_3}{2}\|m_k - x_{k+1}\|_F^2 + \beta_{k+1}\cdot\psi(m_k)\\ &+\tfrac{\mu_2}{2}\|z_{k+1} - m_k\|_F^2\,.\end{aligned} \tag{9}$$

Equation (6) is decoupled into three subproblems, and the reconstruction items and two regularization terms can be solved separately. Thus, (6) involves three subtasks, i.e., denoising, deblurring, and scaling. Deblurring and scaling is solved by (7), which has a closed-form solution. The solution of (7) is

$$x_k = F^{-1}\left(\frac{2\overline{F(k)}F(y_{\uparrow S})+\mu_1\cdot F(z_{k-1})+\mu_3\cdot F(m_{k-1})}{2\overline{F(k)}F(k)+(\mu_1+\mu_3)}\right). \tag{10}$$

In (10), $F(\cdot)$ and $F^{-1}(\cdot)$ denote the operation of fft and $ifft$. $\uparrow S$ denotes the s-fold upsampler.

B. Unfolding Network Framework

Inspired by the abovementioned deep unfolding methods, we propose a new network to solve (5), called YSRNet. The overall framework of the network follows the process of variable splitting algorithms. Equation (7) can be solved efficiently by (10). In traditional optimization algorithms, z and m need to be calculated by the threshold shrinking method, such as singular value thresholding [59], [60], [61]. Instead, in this article, we use a network instead of threshold shrinking algorithms to achieve a better reconstruction performance.

The framework of YSRNet under N iterations is shown in Fig. 2. The upper part of Fig. 2 mainly shows the calculation process of the modules. y denotes the degraded image. LR images are the input from the left, and SR images x are generated after N iterations. The algorithm flow strictly follows the iterative process of the variable splitting optimization method. The whole iteration is an end-to-end learnable process. Each layer of the iteration process has different parameters, which makes the network have a strong mapping ability. Since the target reconstruction module is accomplished by (7), the network only needs to perform simple tasks by solving (8) and (9). In traditional optimization algorithms, the effect of the prior constraint is mostly used to remove the noise, so the function of this network is also to perform the noise removal function. As shown in (4), $\varphi(x)$ denotes the deep prior term. The second regularization we selected is the TV term, which can extract the gradient information of the images. Traditional TV regularization can be obtained by multiplying the image by the gradient matrix D_x in (11) and then calculate the sum of all values in the ∇_x and ∇_y to get L_1 norm values, which means $\|x\|_{\mathrm{TV}} = \|\nabla x\|_1$. And the optimization objective is to make the sum minimize for denoising tasks. However, because calculating L_1 norm will lose the structure information. So the gradient matrices ∇_x and ∇_y are used as the network's input. And L_1 norm of TV regularization term becomes an implicit TV norm in deep unfolding network, which can be expressed as $\psi_{\mathrm{TV}}(x)$

$$D_x = \begin{pmatrix}1 & 0 & 0 & 0 & 0\\ -1 & 1 & 0 & 0 & 0\\ 0 & -1 & 1 & 0 & 0\\ 0 & 0 & \ddots & \ddots & 0\\ 0 & 0 & 0 & -1 & 1\end{pmatrix} \tag{11}$$

$$\begin{cases}\nabla_x = D_x\otimes x\\ \nabla_y = x\otimes D_x.\end{cases} \tag{12}$$

The gradient of the image along the row and column directions can be obtained by (12). ∇_x and ∇_y denote the gradient features of TV term. Then, the details about the deconvolution module $\mathcal{D}$ and the prior module $\mathcal{P}$ are provided.

1) Deconvolution (**D**) *Module:* $\mathcal{D}$ module is a deconvolution module, which is presented as green blocks in Fig. 2. This module is corresponding to (7). The input image is initialized as the nearest interpolation degraded image. Utilizing (10), x-subproblem can be calculated by the last iteration results of $\mathcal{P}$ module and related parameters, where μ_1 and μ_2 in (10) can be obtained by back propagation method. It is the solution of $\|y-(k\otimes x)_{\downarrow S}\|_F^2$ after variable splitting. The input parameters of this module are variables z, m, convolution kernel k, sampling factor S, weight coefficient μ_1, μ_2, and the last reconstructed image x_{k-1}, as shown in Fig. 3. The output of $\mathcal{D}$ module is the reconstructed image, which is also the input of $\mathcal{P}$ module in the next iteration stage.

2) Prior (**P**) *Module:* $\mathcal{P}$ module is the prior module, which is dedicated to the noise removal of input images. It acts as a

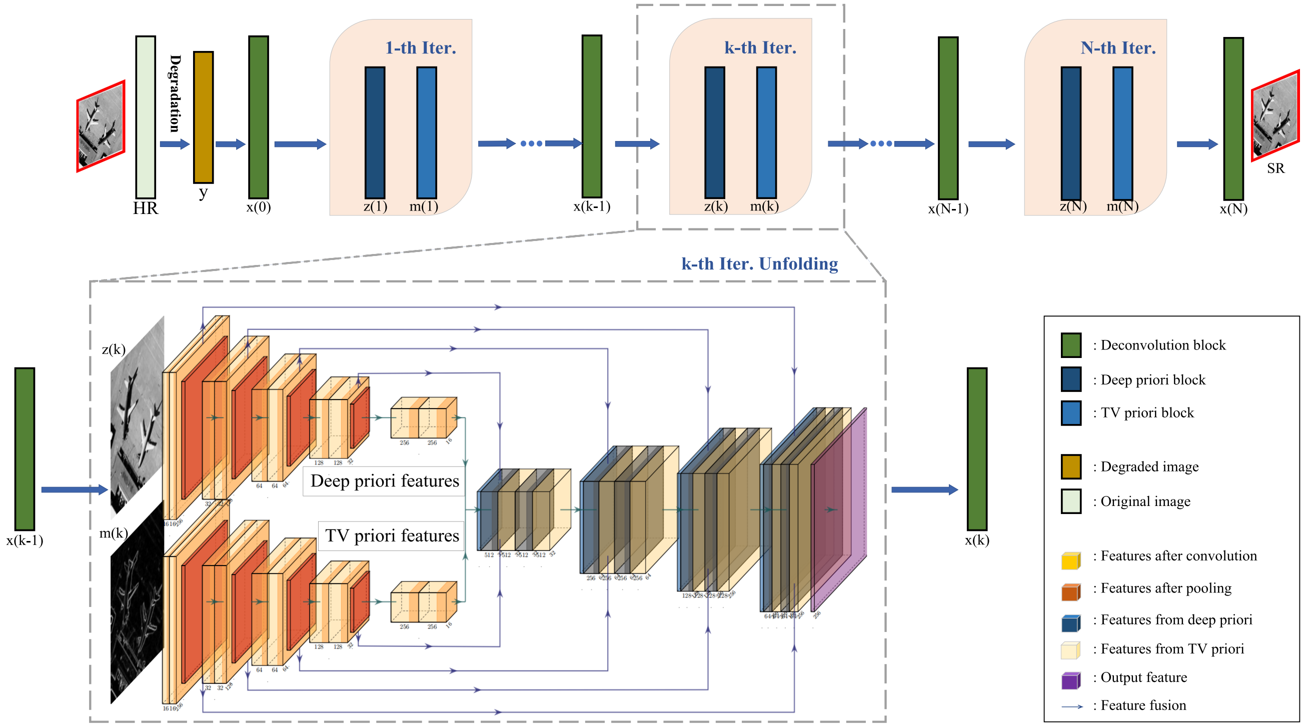


Fig. 2. Illustration of our proposed YSRNet framework. y is the image with degradation. SR denotes the SR image. The green module represents the deconvolution process. The two blue modules represent the noise removal process. Arrows in the network represent data exchange and the network is a fully symmetric structure.

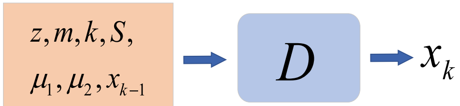


Fig. 3. Inputs of $\mathcal{D}$ module are z, m, k, S, μ_1, μ_2, and x_{k-1}. The output of $\mathcal{D}$ module is the reconstructed image.

denoiser. Inspired by Res-Unet proposed in [62], a new network with multiple feature inputs is proposed in this article, which is called YSRNet. This module corresponds to (8) and (9). It solves z-subproblem and m-subproblem by neural networks. The final results are fused and approximate HR images by end-to-end training. Res-Unet is a combination of res-blocks and U-net. The size of the input images is $b \times n \times s \times s$, where s denotes the height and weight of images, b and n denote the batchsize and channel number, respectively. This network can be divided into two parts. The first part acts as a feature extractor. The second part acts as a data reconstructor. Since the network structure is Y-shaped, we call it Ynet. For each input feature, the feature extractor extracts different scale features of the image to fully mine the intrinsic information from the image. The feature extraction process is accomplished by using several Res-blocks. Each scale transformation is done by a pooling layer, which can make the length and width of the picture to be half of the original and double the number of channels. The design of the Res-block is the same as the structure in [62], which contains two convolution layers and one Relu layer. The structure of the Res-block is shown in (13). Each residual block connects the input to the output using a skip connection. Taking the scale factor equaling to 2 as an example, the five scales for feature extraction process are 256, 128, 64, 32, and 16, respectively. The number of channels is 16, 32, 64, 128, and 256, respectively. After deep prior and TV prior feature extraction, we merge the two parts. Notably, fusion here refers to channel concatenating. Simply adding two parts may lead to a loss of information. Up to now, the feature extraction of the two channels is completed

$$x^{k+1} = x^k + \text{conv}(\text{relu}(\text{conv}(x^k))). \tag{13}$$

The second part is the process of upsampling, which is the inverse process of feature extraction. Image scale changes from 16 to 256, and the channel number changes from 256 to 1. The deconvolution layer accomplishes Upsampling. It is worth noting that during the feature extraction and image reconstruction, we use the skip connection to connect layers of the same scale. This operation can exchange information between different layers to mine deep information of the image. It can also retain shallow features. And this information exchange can bring about the improvement of the network performance [63], [64].

The deep prior features and TV prior features are shown in Figs. 4 and 5, respectively. Deep priori is mainly aimed at the extraction of the overall features. And TV priori is mainly aimed at the extraction of the edge features. TV features can effectively enhance the edge information of the image, which can improve the performance of $\mathcal{P}$ module.

C. End-to-End Training

This network is obtained by end-to-end training. Each iteration contains one $\mathcal{D}$ module and one $\mathcal{P}$ module. N iteration

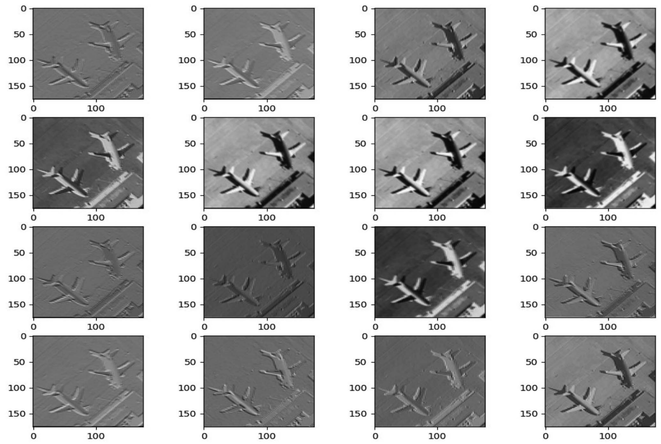


Fig. 4. Deep prior features. Deep prior features are adaptively learned from the network and focus on all the information of the image.

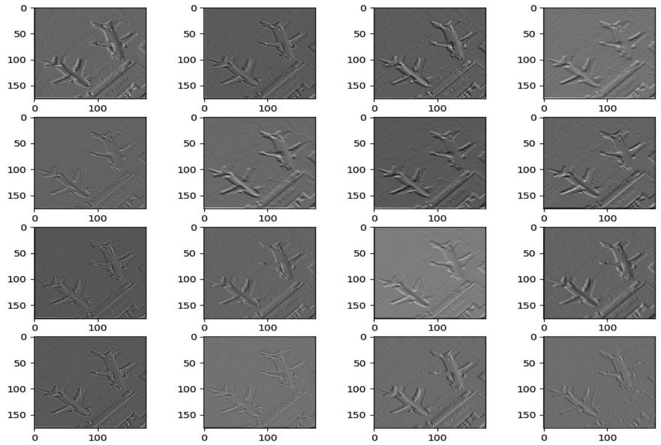


Fig. 5. TV prior features. TV prior features mainly learn the edge information of the image and can provide edge features for image reconstruction.

YSRNet network contains N Ynet blocks. The flows of YSRNet and Ynet are shown in algorithm 1 and algorithm 2, respectively. The original HR images are used to calculate the loss functions. Ultimately, SR images can gradually approximate original HR images.

D. Enhanced Version: YSRNet+

Considering different features and channels in YSRNet have the same weight, an enhanced version (YSRNet+) is proposed. Because each convolution kernel of a neural network also contains weight information to some extent. Compared with YSRNet+, YSRNet can also learn the weights for different prior terms to some extent. However, the performance of YSRNet is limited. Attention mechanism as a weight learning method can make the network learn weight information better. We apply an attention module in YSRNet+. By introducing the attention module, we can apply different weights on different prior terms and channels, and thus, the model will focus more on important features. This module can better explain the effect of α and β in (4). The purpose of α and β is to balance the loss between the reconstruction term and the prior terms, which is also the function of the attention module. Thus, the attention module can further enhance the interpretability of the network. The attention module can extract more valuable features of the image and give it a larger weight, which is helpful for image reconstruction. The Res-block combined with the attention block is shown in Fig. 6. We map the features of different channels into various weights through a simple network so that the network can learn the importance of the features. The Res-blocks are applied in

Algorithm 1: YSRNet.

```
Input: Degraded images $y$, Iteration times N, Epoch
       times t, Network settings
Output: High quality images
1.Data preparation
Data Enhancement(rotating and random clipping).
Data blurring.
Downsampling.
Data adding noise.
while $i < t$ do
    for $j < N$ do
        (Note: Each layer has the same network
          architecture but different parameters.)
        2. $\mathcal{D}$ module
        Calculate $x$ by Eq. (10)
        3. $\mathcal{P}$ module
        Input $x$ and $\psi_{TV}(x)$ to Ynet and get the output.
        j = j+1
    4. Back propagation
    i = i+1
5. Super resolution
Input degraded images and output high quality images.
return High quality images
```

Algorithm 2: Ynet.

```
Input: Noise images $x$, TV priori matrix $TV_x$
Output: Denoised images $X$
1. Feature Extraction Network for x
  x module:
  x $\rightarrow$ Conv(3×3) $\rightarrow$ Downsampling module I $\rightarrow$
  Downsampling module II $\rightarrow$ Downsampling module III
  $\rightarrow$ Downsampling module IV $\rightarrow$ $\mathcal{F}$(x).
2. Feature Extraction Network for $TV_x$
  $TV_x$ module:
  $TV_x$ $\rightarrow$ Conv(3×3) $\rightarrow$ Downsampling module I $\rightarrow$
  Downsampling module II $\rightarrow$ Downsampling module III
  $\rightarrow$ Downsampling module IV $\rightarrow$ $\mathcal{F}(TV_x)$.
3. Channel Concat
  $\mathbb{X}$ = Concat($\mathcal{F}$(x), $\mathcal{F}(TV_x)$).
4. Noise Removal and Images Output
  $\mathbb{X}$ $\rightarrow$ Conv(3×3) $\rightarrow$ Upsampling module I $\rightarrow$
  Upsampling module II $\rightarrow$ Upsampling module III $\rightarrow$
  Upsampling module IV $\rightarrow$ X.
return Denoised images $X$
```

```
Algorithm 3: Attention Module.
  Input: $x$-Feature channels $\mathcal{R}$(x), $TV$-Feature channels
         $\mathcal{R}(TV)$, Channel numbers $n$
  Output: Channels with attention $\mathcal{X}$
  Part I: Deep priori attention module
    $\mathcal{R}$(x) $\rightarrow$ pooling($\mathcal{R}$(x)) $\rightarrow$ full connection($\mathcal{R}$(x)) $\rightarrow$
    $\sum_1^n w_x$.
  Part II: TV attention module
    $\mathcal{R}$(TV) $\rightarrow$ pooling($\mathcal{R}$(TV)) $\rightarrow$ full connection($\mathcal{R}$(TV))
    $\rightarrow \sum_1^n w_{TV}$.
  :Part III Feature Fusion (concatenate feature maps)
    $\mathcal{X} = \varphi(\mathcal{R}(\mathrm{x})) \times \sum_1^n w_x + \varphi(\mathcal{R}(\mathrm{TV})) \times \sum_1^n w_{TV}$.
  return $\mathcal{X}$
```

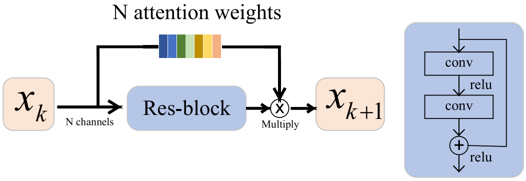


Fig. 6. Res-block with attention block. Blocks of different colors denote weight values for different channels. The networks in blue box denotes a res-block.

each stage of Ynet. Thus, the attention module is also applied in all stages.

The function of attention Res-block module is given in (14), where conv, relu, pooling, and full_con denote convolution layer, activation layer, pooling layer, and full connection layer, respectively. Equation (14) is corresponding to Fig. 6. The second term denotes the attention weight, which consists of a full connection layer and a pooling layer. As shown in algorithm 3, the deep priori and TV priori both have n feature maps in the network. According to (14), we can multiply feature maps by the corresponding weight. $\sum_1^n w_x$ and $\sum_1^n w_{\mathrm{TV}}$ are corresponding to α and β in (4). We concatenate the deep priori feature channels and TV priori feature channels for feature fusion. Then, all the features are used for image reconstruction

$$x^{k+1} = \left(x^k + \mathrm{conv}(\mathrm{relu}(\mathrm{conv}(x^k)))\right) \cdot \left(\mathrm{full\_con}(\mathrm{pooling}(x^k))\right). \tag{14}$$

IV. Experiment

We evaluate the proposed networks' performance using three remote sensing datasets to demonstrate the algorithm's effectiveness.

UC Merced Land-Use dataset is a common aerial dataset containing 21 kinds of scenes and 100 samples of 256 × 256 pixels in each class. The pixel resolution of this dataset is 1 ft. Training can be time-consuming because of too many pictures in this dataset. Therefore, we selected the first ten categories of scenes, each selecting 40 pictures as training samples. Then, for each class, ten pictures are selected as the testing sample to evaluate image reconstruction performance. Ultimately, we use 400 images for training and 50 for testing. For the second aerial dataset, the WHU-RS19 dataset is selected. WHU-RS19 dataset is collected from Google Earth by Wuhan university. It contains 19 categories of physical scenes in the satellite imagery, including airport, beach, bridge, commercial, desert, river, and so on. For the universality of the algorithm, we choose the NWPU-RESISC45 dataset to demonstrate the algorithm's effectiveness. NWPU-RESISC45 dataset is created by Northwestern Polytechnical University, which is a large scale on the scene classes and the total image number. It holds big variations in translation, spatial resolution, viewpoint, object pose, illumination, and so on. For this dataset, we selected the first forty categories of scenes, each selecting 30 images for training and 2 for testing.

Examples of these three datasets are shown in Fig. 7. The experiments are performed under the Pytorch framework, and we train them on NVIDIA Titan RTX GPUs. Adam is selected as the optimizer. For each training sample, the images are cropped to the patches of size 48 × 48 as the input. The learning rate is initialized as 0.0005. The mini-batch size is set to 48. We choose L1 loss for the PSNR performance. It takes about 20 h to obtain the YSRNet model.

In addition to YSRNet and YSRNet+, several advanced SR algorithms are compared, including DPSR [65], IMDN [66], MAN [68], HSENet [69], USRNet [54]. DPSR proposes a deep plug-and-play SR framework to solve the SISR problem. It considers multiple blurring kernels and supports existing deblurring methods for blurring kernel estimation. IMDN proposes a multidistillation module to optimize memory and real-time. MAN is an attention network, which consists of a multiscale large kernel attention structure. HSENet was proposed for optical remote sensing image SR, which is a hybrid-scale self-similarity exploitation network. USRNet presents a learnable unfolding network, which is the first to solve multiple degradation problems using a single end-to-end model. We use FLOPs (floating point operations) to evaluate the computational complexity of the algorithm. The FLOPs of DPSR, IMDN, MAN, HSENet, USRNet, YSRNet, and YSRNet+ are 52.07 G, 7.67 G, 9.26 G, 18.82 G, 9.08 G, 7.13 G, 11.00 G, and 11.17 G, respectively. Due to the introduction of double priors, the algorithm's computational complexity has increased. Moreover, due to the introduction of the attention mechanism, YSRNet+ has only a little higher computational complexity than YSRNet, but it also has performance improvements.

Considering the article's length, we only use PSNR to evaluate the performance of algorithms. PSNR is an image evaluation index, which is defined by MSE between the ground-truth image and the generated SR result. It can be expressed as

$$\mathrm{PSNR} = 20\log_{10}\frac{\mathrm{Max}(I^{\mathrm{HR}})}{\sqrt{\mathrm{MSE}(I^{\mathrm{HR}}, G(I^{\mathrm{LR}}))}} \tag{15}$$

where $\mathrm{Max}(I^{\mathrm{HR}})$ denotes the maximum pixel value in original images. The larger the value of PSNR demonstrates, the better performance.

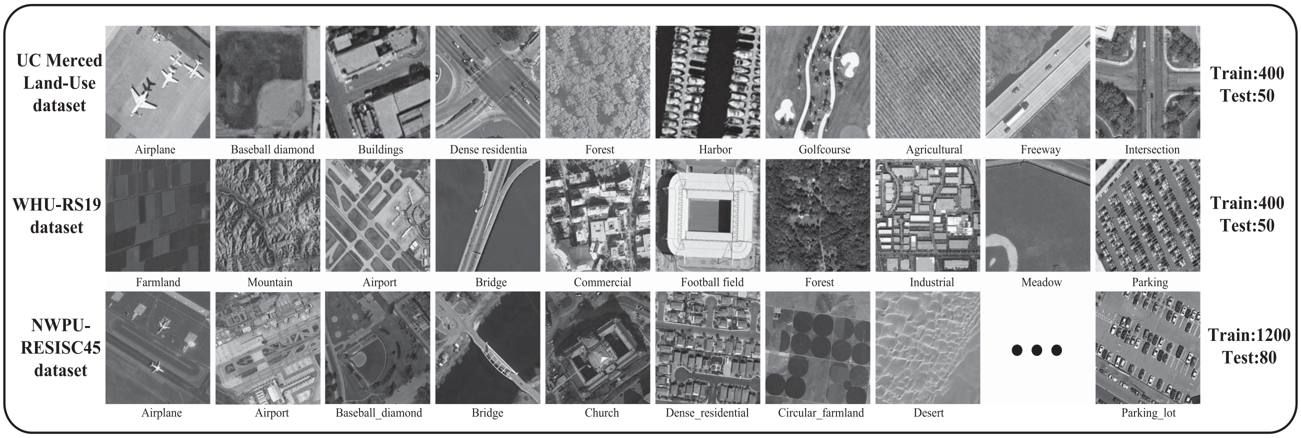


Fig. 7. Illustration of the datasets. We select 10 types of images from UC Merced Land-Use dataset and WHU-RS19 dataset as training samples, respectively. And we randomly select five of each class as testing samples. We select 40 types of images from NWPU-RESISC45 dataset as training samples. And we randomly select two of each class for testing.

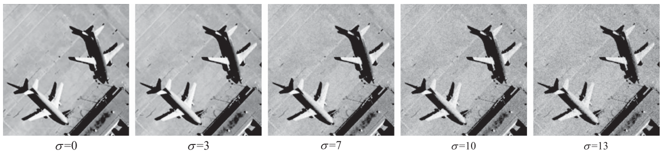


Fig. 8. Degradation images with various noise levels. The noise levels are set to 0, 3, 7, 10, and 13, respectively.

In what follows, we will verify the validity and universality of the proposed algorithms in three remote sensing datasets.

A. Comparison With Different Methods on Dataset 1

In this section, we use the UC Merced Land-Use dataset to verify the validity of the proposed algorithms. As Fig. 7 shows, 400 Gy images are used for training, and 50 images are used for testing. The size of training images is 180 × 180, and the scales of SR are 2, 3, and 4. The noise level is set to $\sigma = 0$, $\sigma = 3$, and $\sigma = 7$. n in (1) is usually assumed to be additive white Gaussian noise. The noise intensity for n is $\sigma/255$. We first generate the noise matrix with the mean value of 0 and the standard deviation of $\sigma/255$. Then, we add the image with the noise matrix. Different standard deviations will produce different intensity noise matrices. Images with different noise levels are shown in Fig. 8. We set various noise levels to verify the algorithm's robustness under different conditions. We generate the training data pairs $\{x_i, y_i\}$ by first extracting the component of randomly cropped image blocks (48×48 pixels for each block). And two data enhancement methods are used to increase training data: flipping and rotating. The number of unfolding layers is 6 ($N = 6$). Ten blurring kernels from [54] are used to verify the algorithm's performance under different degradation conditions. We use 4 isotropic Gaussian blurring kernels, 4 anisotropic Gaussian blurring kernels, and 2 motion blurring kernels. Although complex motion blurring kernels are generally not considered for aerial images, they are also considered in order to verify the effectiveness of YSRNet and YSRNet+ more fully.

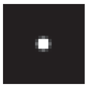
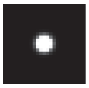
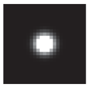
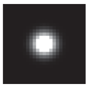
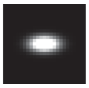
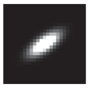
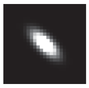
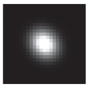
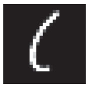
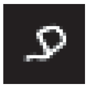
The first eight kernels are Gaussian blurring kernels, and the last two are complex motion blurring kernels. The information entropy of different remote sensing images differs significantly compared with natural images. Simple remote sensing images, such as agricultural, contain less information entropy, while complex images, such as buildings, contain larger information entropy. Therefore, the stability requirements of the algorithm are more stringent. The best results are shown in bold, and the second is underlined. The last two are algorithms proposed in this article, which are also shown in bold.

The PSNR results are presented in Table I . YSRNet and YSRNet+ still stand out under different blurring kernel degradation. We use the average PSNR of 50 testing pictures as the result, which can make results more stable. Traditional neural networks do not have good adaptability to different blurring kernels. They perform well with no blurring or with small blurring kernels.

Fig. 9 shows the SR results of different algorithms with the scale factor being 3. DPSR and IMDN have similar SR performance. But because the lightweight multidistillation network being used in IMDN has fewer network parameters, which can lead to a fast training process. The edge structure of MAN algorithm is not clear enough. USRNet is also a deep unfolding method that can fully use the prior kernels' information. The images of USRNet, YSRNet, and YSRNet+ are clearer than the others; among them, YSRNet+ handles the details best. The PSNR of YSRNet+ is 0.08 dB, 0.18 dB, and 0.22 dB higher than USRNet on average for ×2, ×3, and ×4 times SR, respectively. Except for the first kernel and the last, YSRNet+ can obtain the highest PSNR, which demonstrates the effectiveness of the proposed network architecture.

Compared with USRNet, YSRNet has a larger number of parameters and more time-consuming training due to the processing module for TV prior term. In practice, because testing time is more critical, thus, an appropriate increase in training time is acceptable. Based on this, the attention module introduced by YSRNet+ also slightly increases the training time. The

TABLE I
SCORES OF PSNR FOR DIFFERENT SR ALGORITHMS WITH REMOTE SENSING IMAGES (DATASET 1)

Algorithms	Scale	Noise Level										
DPSR[1] [42]	×2	0	31.14	27.94	26.25	25.10	24.92	24.63	24.95	24.22	20.94	22.44
	×3	0	28.29	27.13	25.98	25.03	24.82	24.55	24.87	24.22	21.06	22.38
	×3	3	**27.47**	26.45	25.49	24.67	24.48	24.24	24.50	23.94	21.14	22.33
	×3	7	26.48	25.66	24.90	24.21	24.04	23.84	24.05	23.58	21.15	22.21
	×4	0	25.98	25.60	25.08	24.46	24.30	24.08	24.29	23.81	20.95	22.20
IMDN[2] [65]	×2	0	31.21	27.99	26.27	25.12	24.94	24.64	24.97	24.23	20.93	22.44
	×3	0	**28.64**	27.32	26.09	25.09	24.88	24.61	24.92	24.25	21.04	22.35
	×3	3	27.35	26.33	25.34	24.49	24.27	24.04	24.33	23.73	20.80	21.98
	×3	7	24.82	24.20	23.54	22.93	22.76	22.59	22.80	22.36	20.01	21.00
	×4	0	**26.20**	25.82	25.25	24.59	24.41	24.18	24.43	23.91	20.97	22.22
MAN[3] [66]	×2	0	30.43	31.39	30.92	29.96	29.59	29.25	29.71	28.82	28.99	28.06
	×3	0	25.62	27.79	28.20	28.25	27.92	27.83	28.00	27.91	25.12	25.32
	×3	3	25.44	27.28	27.30	26.86	26.53	26.48	26.67	26.25	24.62	24.77
	×3	7	25.02	26.39	26.15	25.58	25.30	25.18	25.34	24.99	23.82	23.99
	×4	0	24.04	25.72	26.23	26.38	26.20	26.14	26.21	26.34	23.31	23.99
HSENet[4] [67]	×2	0	30.40	31.89	31.30	30.13	29.55	29.13	29.46	28.74	28.22	27.36
	×3	0	26.17	28.04	28.48	28.65	28.48	27.97	28.05	28.36	23.45	24.12
	×3	3	25.87	27.18	26.95	26.26	26.00	25.69	26.02	25.48	22.88	23.38
	×3	7	25.46	26.28	25.84	25.16	25.06	24.79	25.02	24.46	22.34	22.82
	×4	0	24.41	25.96	26.20	26.35	26.28	25.97	26.13	26.43	22.51	23.17
USRNet[5] [54]	×2	0	31.91	32.50	31.75	30.43	29.83	29.62	30.19	29.12	30.74	30.39
	×3	0	27.40	28.69	28.87	28.73	28.40	28.31	28.61	28.27	26.92	26.56
	×3	3	26.85	27.90	27.39	26.96	26.65	26.45	26.69	26.23	26.03	25.68
	×3	7	25.90	26.73	26.26	25.63	25.46	25.34	25.54	25.05	24.66	24.41
	×4	0	24.78	26.01	26.32	26.43	26.35	26.18	26.28	26.40	24.82	24.83
YSRNet	×2	0	32.02	32.54	31.79	30.32	29.81	29.58	30.09	29.03	30.82	30.42
	×3	0	27.35	28.67	28.98	28.87	28.55	28.47	28.72	28.39	27.21	26.81
	×3	3	27.02	27.45	27.24	26.97	26.62	26.51	26.83	26.21	25.91	25.66
	×3	7	26.13	26.44	26.26	25.73	25.50	25.37	25.58	25.11	24.73	24.55
	×4	0	25.13	26.22	26.50	26.62	26.55	26.38	26.48	26.59	24.83	24.87
YSRNet+	×2	0	**32.11**	**32.61**	**31.83**	**30.48**	**29.90**	**29.66**	**30.20**	29.12	**30.86**	**30.56**
	×3	0	27.58	**28.85**	**29.09**	**28.95**	**28.58**	**28.49**	**28.77**	**28.43**	**27.22**	**26.86**
	×3	3	27.27	**27.98**	**27.68**	**27.05**	**26.70**	**26.65**	**26.94**	**26.30**	**26.20**	**25.80**
	×3	7	**26.53**	**26.81**	**26.38**	**25.79**	**25.54**	**25.43**	**25.67**	**25.16**	**24.88**	**24.61**
	×4	0	25.09	**26.25**	**26.54**	**26.67**	**26.58**	**26.43**	**26.53**	**26.65**	**24.88**	**24.95**

[1] [Online]. Available: https://github.com/cszn/DPSR
[2] [Online]. Available: https://github.com/Zheng222/IMDN
[3] [Online]. Available: https://github.com/icandle/MAN
[4] [Online]. Available: https://github.com/Shaosifan/HSENet
[5] [Online]. Available: https://github.com/cszn/USRNet
The best results are shown in bold and the second best results are underlined.

overall visual effect of USRNet is similar to that of the proposed algorithms, but YSRNet and YSRNet+ do better in detail. At the same time, YSRNet+ is more interpretable.

B. Comparison With Different Algorithms on Dataset 2

In this section, we use another remote sensing dataset to further verify the effectiveness of the algorithms. The training and testing samples are generated from WHU-RS19 dataset. The PSNR results are presented in Table II. YSRNet and YSRNet+ still perform well. In addition, several testing samples are selected for detailed analysis.

Since multiple degradation factors are considered in this article, several specific cases are selected for algorithm comparison. Degradation factors are set as Table III. We choose different scenes, scale factors, noise levels, and kernels to demonstrate the effectiveness of proposed algorithms.

Fig. 10 is the simplest scene showing an industrial building taken by the airplane from WHU-RS19 dataset with the size of 384×384. A part of the image is enlarged to show the SR effect

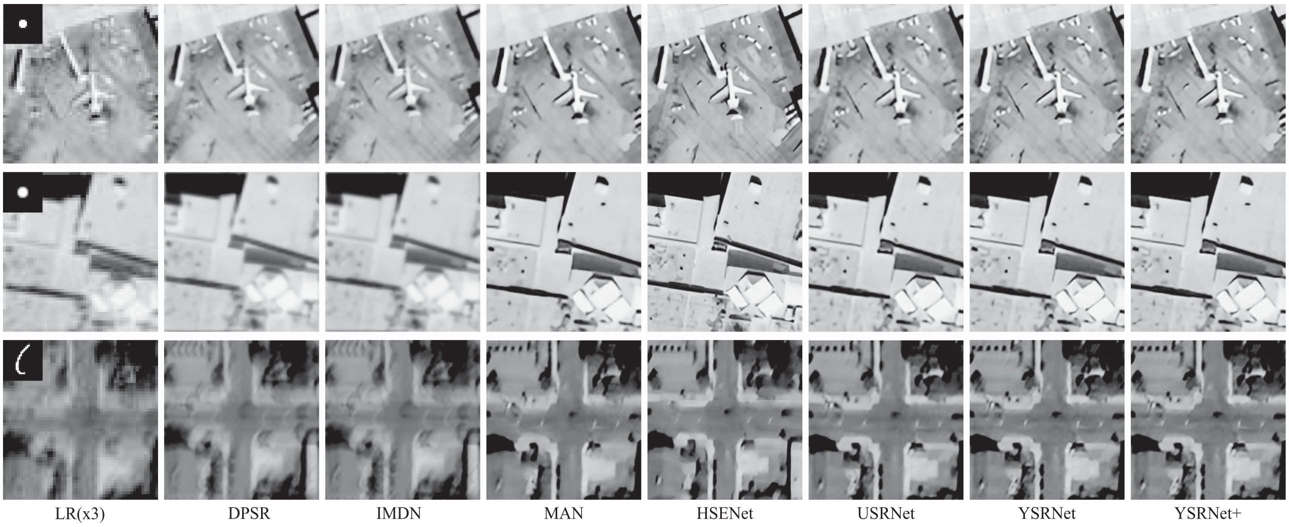


Fig. 9. SR results for different algorithms under different blurring kernels. The blurring kernel is shown on the upper-left corner of x^0. The scale factor is 3, and the noise level of LR image is 0.

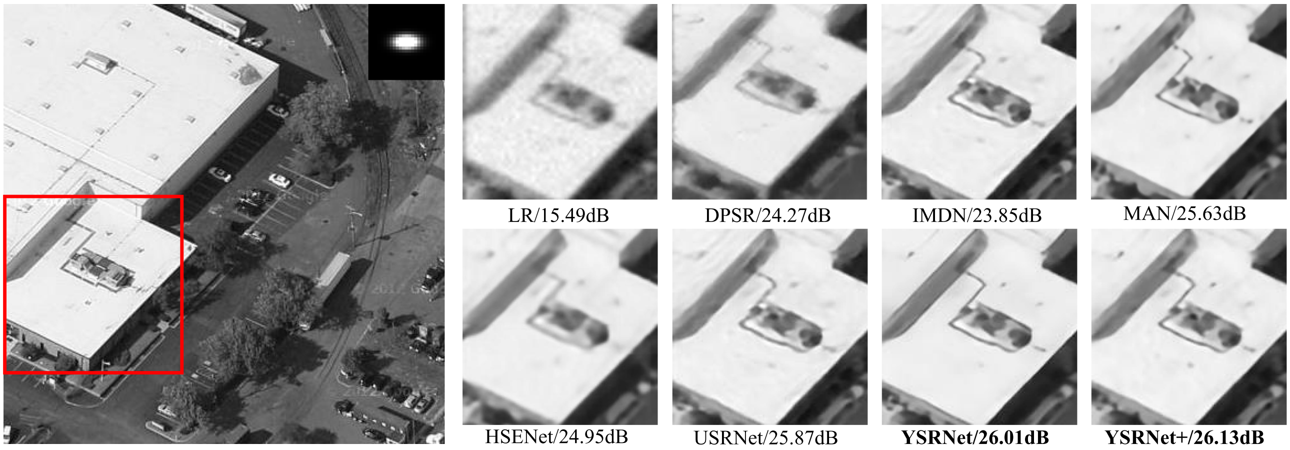


Fig. 10. Reconstruction of buildings. Parts of SR results for different algorithms under the same degradation cases are enlarged. The blurring kernel is shown on the upper-left corner of the whole image. The scale factor is 3, and the noise level of LR image is 5.

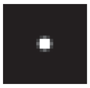
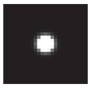
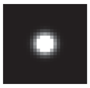
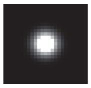
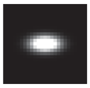
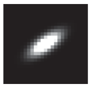
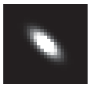
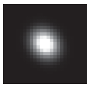
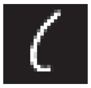
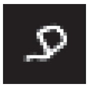
of different algorithms more clearly. The scale factor is 3 with the noise level of 5. The image is blurred by kernel 5, which is an anisotropic Gaussian kernel. LR result is obtained through nearest neighbor interpolation. PSNR of LR image is only 15.49 dB. PSNR is improved to varying degrees by the optimization of different algorithms. The edges of DPSR, IMDN, and HSENet results are somewhat blurred, which roughly restores the shape of the target and provides better noise reduction. MAN lost some details of the image. USRNet restores the information from the original image better, but it does not perform as well as YSRNet and YSRNet+ for detail textures and small target edges. YSRNet has a clearer image and a cleaner noise reduction, but it is still blurry to restore the small objects in the image. YSRNet+ has a better SR performance and can further improve PSNR than YSRNet+.

In Fig. 11, more noise is added ($\sigma = 9$). DPSR and HSENet can roughly restore the HR image. Although IMDN restores the general outline of the object, the noise is not removed cleanly and stripes are produced. Thus, its PSNR is only 0.8 dB higher than LR image. Compared with YSRNet, USRNet does not restore road textures very well. But YSRNet also produces a small amount of interference stripes. YSRNet+ performs best among these SR algorithms.

Fig. 12 is the most complex scene for 3 times SR. Except IMDN, the other methods can do noise reduction well. For freeway scenes, YSRNet and YSRNet+ have the same performance. The PSNR of them are 0.5 dB higher than USRNet.

Figs. 14 and 13 are images of parking lot and dense residentia, respectively. The task of Figs. 14 and 13 is more challenging with the scale factor of 4. For the textures in Fig. 13, YSRNet performs better. Some small cars can also be restored. In Fig. 14, a vehicle-dense area is enlarged. The visual effect of each algorithm is not clear enough due to the difficulty of the task. USRNet, YSRNet, and YSRNet+ behave similarly from visual effects. PSNR of

TABLE II
SCORES OF PSNR FOR DIFFERENT SR ALGORITHMS WITH REMOTE SENSING IMAGES (DATASET 2)

Algorithms	Scale	Noise Level										
DPSR	×2	0	29.49	27.20	25.90	25.01	25.00	24.68	24.99	24.32	21.60	22.80
	×3	0	27.51	26.60	25.71	24.94	24.88	24.59	24.93	24.30	21.70	22.75
	×3	3	**26.87**	26.04	25.27	24.59	24.55	24.27	24.58	24.02	21.76	22.71
	×3	7	26.01	25.34	24.71	24.16	24.12	23.86	24.15	23.67	21.76	22.58
	×4	0	25.96	25.67	25.26	24.77	24.70	24.44	24.72	24.25	21.81	22.86
IMDN	×2	0	29.79	27.36	26.01	25.09	25.08	24.76	25.07	24.39	21.58	22.81
	×3	0	27.71	26.74	25.81	25.03	24.98	24.69	25.01	24.38	21.70	22.78
	×3	3	26.65	25.90	25.13	24.46	24.41	24.15	24.43	23.88	21.40	22.40
	×3	7	24.33	23.89	23.39	22.90	22.86	22.66	22.86	22.46	20.44	21.27
	×4	0	26.16	25.83	25.37	24.85	24.76	24.52	24.81	24.32	21.81	22.88
MAN	×2	0	29.55	30.23	29.85	28.99	**28.76**	28.57	28.75	28.07	28.69	28.12
	×3	0	25.88	27.34	27.47	27.26	27.00	27.05	27.13	26.82	25.58	25.71
	×3	3	25.36	26.45	26.35	25.93	25.70	25.66	25.79	25.48	24.68	24.84
	×3	7	24.98	25.78	25.55	25.12	24.96	24.83	24.93	24.74	24.07	24.20
	×4	0	24.66	25.77	26.11	26.23	26.11	26.09	26.14	26.22	24.31	24.76
HSENet	×2	0	28.97	30.21	29.80	28.73	28.43	28.11	28.20	27.64	27.82	27.36
	×3	0	25.98	27.27	27.59	27.68	**27.54**	27.35	27.33	**27.42**	24.27	24.69
	×3	3	25.74	26.67	26.55	25.99	25.69	25.62	25.86	25.41	23.76	24.05
	×3	7	25.37	25.92	25.60	25.11	25.00	24.79	25.05	24.63	23.21	23.51
	×4	0	24.66	25.85	26.03	26.10	26.08	25.86	25.93	26.10	23.52	23.98
USRNet	×2	0	30.09	30.51	29.98	28.98	28.65	28.54	28.83	27.94	29.43	29.07
	×3	0	26.80	27.56	27.68	27.62	27.42	27.34	27.54	27.29	26.28	26.04
	×3	3	26.29	26.94	26.70	26.23	26.06	26.00	26.16	25.67	25.33	25.06
	×3	7	25.79	26.13	25.75	25.29	25.19	25.02	25.22	24.82	24.45	24.24
	×4	0	24.98	25.86	26.06	26.12	26.04	25.91	26.04	26.09	24.81	24.79
YSRNet	×2	0	30.27	30.70	30.10	29.03	28.71	**28.60**	28.85	**27.97**	**29.69**	**29.38**
	×3	0	26.94	27.65	27.78	27.70	27.51	27.47	27.63	27.38	26.57	26.17
	×3	3	26.59	27.05	26.76	26.28	26.13	26.04	26.19	25.72	25.66	25.34
	×3	7	26.03	26.19	25.79	25.32	25.24	25.04	25.25	24.85	24.51	24.29
	×4	0	25.29	26.08	26.25	26.31	26.23	**26.10**	26.24	26.27	24.83	24.83
YSRNet+	×2	0	**30.35**	**30.73**	**30.13**	**29.04**	28.73	28.57	**28.87**	27.94	29.56	29.35
	×3	0	26.69	**27.70**	**27.82**	**27.73**	27.53	**27.50**	**27.66**	27.38	**26.59**	**26.28**
	×3	3	26.78	**27.12**	**26.84**	**26.35**	**26.19**	**26.11**	**26.25**	**25.77**	**25.78**	**25.45**
	×3	7	**26.20**	**26.24**	**25.84**	**25.37**	**25.30**	**25.15**	**25.28**	**24.89**	**24.60**	**24.42**
	×4	0	25.28	**26.11**	**26.32**	**26.38**	**26.30**	26.08	**26.29**	**26.34**	**24.90**	**24.89**

The best results are shown in bold and the second best results are underlined.

TABLE III
DEGRADATED IMAGES OF FIVE SCENES

Scene	Scale Factor	Noise	Kernel
Buildings	×3	5	5
Bridge	×3	9	3
Freeway	×3	9	6
Dense residentia	×4	3	4
Parking	×4	7	9

The best results are shown in bold and the second best results are underlined.

YSRNet and YSRNet+ are 0.12 dB and 0.17 dB higher than USRNet, respectively.

In summary, all algorithms have the function of noise removing, deblurring and image SR. DPSR and HSENet have a similar performance. These two have problems of detail texture restoring. IMDN has faster computation speed, but its performance decreases with high noise levels. MAN may also lost some details in the process of image reconstruction. USRNet is stable and can handle various degradation factors. YSRNet+ performs best and has better interpretability at the same time.

C. *Universality of the Proposed Algorithms on Dataset 3*

PSNR results of the third dataset are shown in Table IV. Each PSNR is the average of 80 testing samples. The PSNR results demonstrate that our proposed algorithms outperform the other competing methods. YSRNet+ is 0.20 dB, 0.37 dB, and

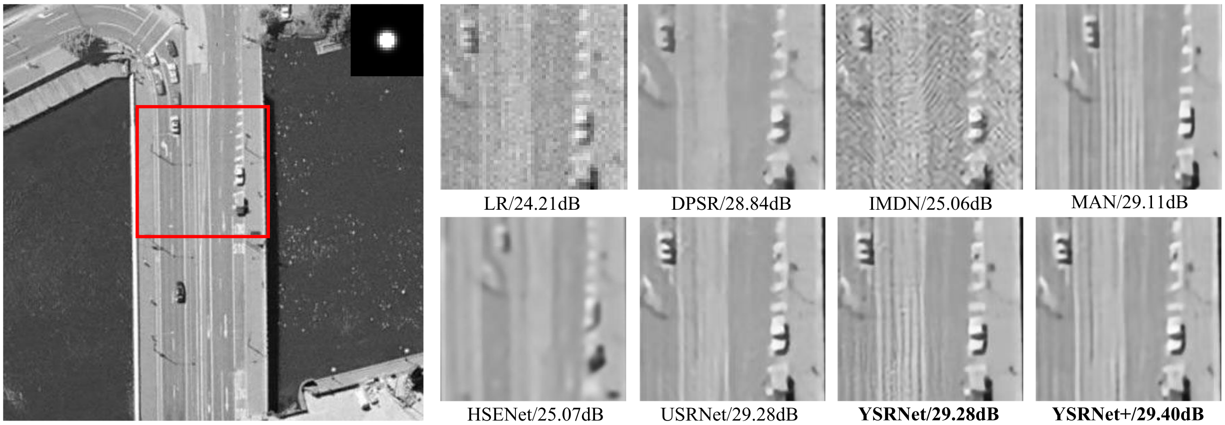


Fig. 11. Reconstruction of bridge. Parts of SR results for different algorithms under the same degradation cases are enlarged. The blurring kernel is shown on the upper-left corner of the whole image. The scale factor is 3, and the noise level of LR image is 9.

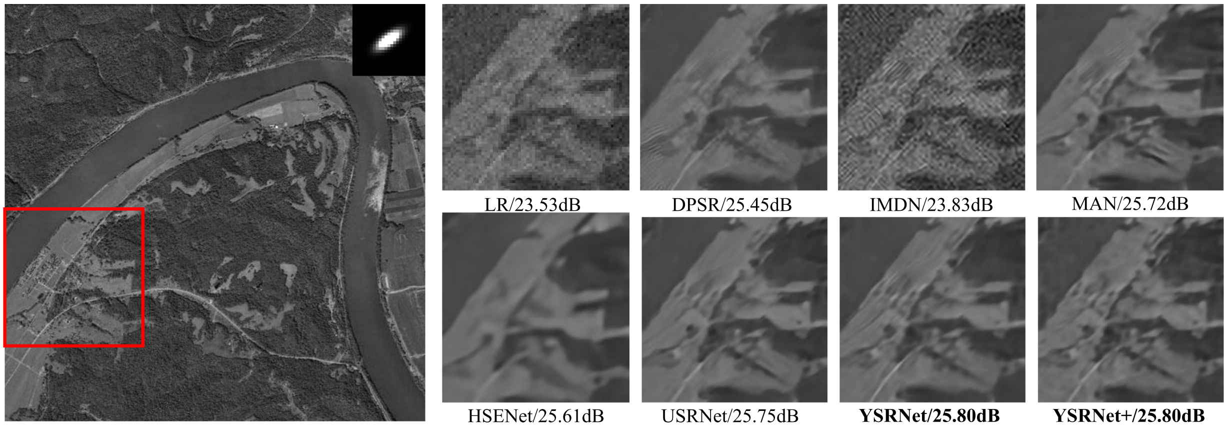


Fig. 12. Reconstruction of freeway. Parts of SR results for different algorithms under the same degradation cases are enlarged. The blurring kernel is shown on the upper-left corner of the whole image. The scale factor is 3, and the noise level of LR image is 9.

0.47 dB higher than USRNet on average at 2x, 3x, and 4x SR, respectively. Traditional neural network methods show a better performance only at the first kernel. The proposed algorithms perform well, especially for complex blurring kernels, which indicates that the proposed algorithms can fully use the image's prior information and can be adapted to different blurring kernels. For the degradation of small blurring kernels, the traditional algorithm performs well. However, as the blurring kernels become more complex, the performance of traditional algorithms degrades dramatically. YSRNet and YSRNet+ perform well in all situations.

The visual effect of different algorithms is shown in Fig. 15. The last three methods are unfolding-based algorithms, which show that the unfolding-based methods have better SR results for complex blurring kernels. DPSR, IMDN, and HSENet cannot remove image blurring very well. MAN may lost some details. USRNet and the proposed methods are suitable for various blurring kernels. Compared to USRNet, the results of YSRNet are better than USRNet in detail due to the introduction of gradient channels. The attention module can further enhance YSRNet.

Since the proposed algorithm is based on the variable splitting algorithm, the algorithm follows the iterative updating process of the optimization algorithm. The number of iterative layers of the algorithm is set to six, so the image reconstruction process is divided into six steps. Each level of iteration brings the iteration results closer to HR pictures. The results of the experiment are shown in Fig. 16. The initialized picture results from the downsampling picture interpolated by nearest neighbors. The downsampling factor is 3. x^k denotes iteration layers. As Fig. 16 shows, the image becomes clearer as the number of iteration times increases, which denotes that the network is similar to traditional optimization algorithms. The experimental results also show that PSNR increases with the number of layers being larger.

In summary, the proposed YSRNet and YSRNet+ algorithms are effective and perform well on standard datasets, especially for complex blurring kernels.

D. Ablation Experiment

The ablation experiments are shown in Table V, where ave. denotes the average value of the results in all SR scales and noise levels. Because the proposed algorithm is an improvement

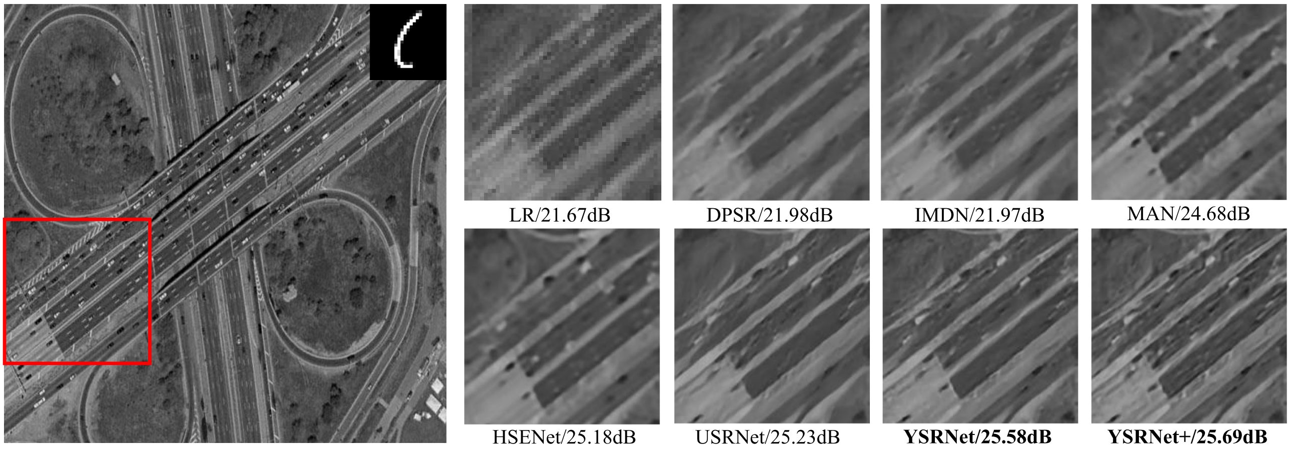


Fig. 13. Reconstruction of dense residentia. Parts of SR results for different algorithms under the same degradation cases are enlarged. The blurring kernel is shown on the upper-left corner of the whole image. The scale factor is 4, and the noise level of LR image is 3.

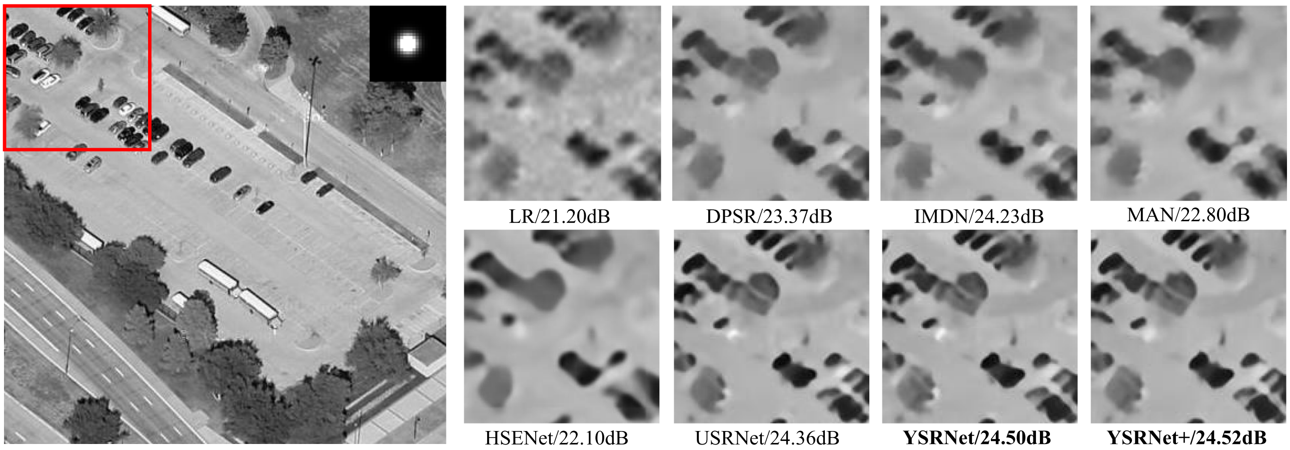


Fig. 14. Reconstruction of parking. Parts of SR results for different algorithms under the same degradation cases are enlarged. The blurring kernel is shown on the upper-left corner of the whole image. The scale factor is 4, and the noise level of LR image is 7.

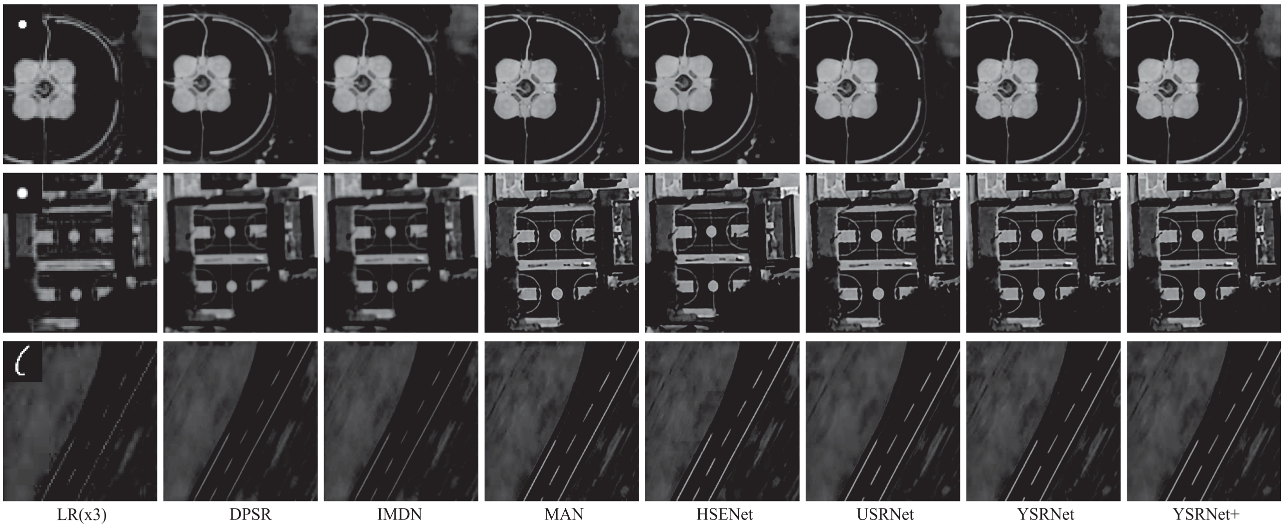
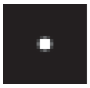
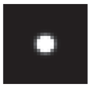
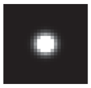
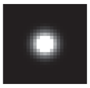
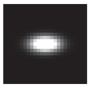
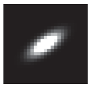
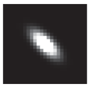
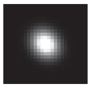
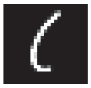
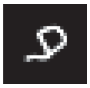


Fig. 15. SR results for different algorithms under different blurring kernels. The blurring kernel is shown on the upper-left corner of x^0. The scale factor is 3, and the noise level of LR image is 0.

TABLE IV
SCORES OF PSNR FOR DIFFERENT SR ALGORITHMS WITH REMOTE SENSING IMAGES (DATASET 3)

Algorithms	Scale	Noise Level										
DPSR	×2	0	31.86	29.11	27.53	26.44	26.35	26.14	26.25	25.57	22.31	23.74
	×3	0	29.76	28.49	27.34	26.38	26.24	26.07	26.16	25.54	22.42	23.71
	×3	3	**28.90**	27.81	26.84	26.01	25.90	25.73	25.82	25.28	22.47	23.68
	×3	7	27.69	26.89	26.16	25.50	25.41	25.26	25.34	24.90	22.50	23.57
	×4	0	27.55	27.16	26.61	25.98	25.83	25.68	25.79	25.33	22.48	23.74
IMDN	×2	0	32.25	29.24	27.61	26.49	26.40	26.20	26.29	25.61	22.33	23.76
	×3	0	**29.89**	28.57	27.41	26.44	26.31	26.13	26.23	25.60	22.42	23.72
	×3	3	28.48	27.53	26.61	25.80	25.69	25.53	25.62	25.08	22.17	23.38
	×3	7	25.65	25.14	24.59	24.06	23.99	23.86	23.93	23.55	21.28	22.26
	×4	0	**28.07**	27.56	26.90	26.21	26.05	25.92	25.99	25.50	22.49	23.80
MAN	×2	0	31.78	33.05	32.32	30.95	30.40	30.11	30.24	29.55	30.46	29.54
	×3	0	28.04	29.70	29.82	29.64	29.38	29.21	29.24	29.08	26.85	26.85
	×3	3	27.83	28.68	28.31	27.70	27.33	27.34	27.39	26.98	26.12	26.05
	×3	7	27.20	27.34	26.95	26.43	26.25	26.14	26.22	25.84	24.95	24.97
	×4	0	25.80	27.51	27.76	27.84	27.75	27.52	27.50	27.78	25.09	25.36
HSENet	×2	0	30.77	32.13	31.67	30.39	29.96	29.61	29.67	29.07	28.88	28.25
	×3	0	26.97	28.62	28.97	29.00	28.72	28.49	28.42	28.51	25.52	25.69
	×3	3	26.78	28.04	27.94	27.44	27.15	27.09	27.12	26.74	25.00	25.05
	×3	7	26.34	27.09	26.76	26.21	26.06	25.94	26.02	25.65	24.22	24.25
	×4	0	25.31	26.86	27.15	27.29	27.30	27.09	26.96	27.32	24.32	24.78
USRNet	×2	0	32.54	33.08	32.25	30.95	30.58	30.35	30.53	29.68	31.80	31.23
	×3	0	28.70	29.69	29.84	29.71	29.43	29.32	29.33	29.17	28.16	27.81
	×3	3	28.10	28.75	28.37	27.72	27.53	27.45	27.52	27.00	26.69	26.40
	×3	7	27.31	27.57	27.04	26.46	26.37	26.26	26.32	25.89	25.43	25.31
	×4	0	26.69	27.75	27.96	28.01	27.93	27.78	27.73	27.95	26.23	26.25
YSRNet	×2	0	32.90	33.34	32.49	31.15	30.77	30.56	30.70	29.81	32.10	31.62
	×3	0	29.12	29.91	29.99	29.80	29.50	29.44	29.41	29.19	28.34	28.03
	×3	3	28.68	28.87	28.42	27.75	27.55	27.48	27.53	27.01	**27.05**	26.64
	×3	7	27.65	27.57	27.04	26.45	26.37	26.28	26.35	25.90	25.44	25.35
	×4	0	26.89	27.82	28.00	28.03	27.92	27.81	27.75	27.96	26.37	26.44
YSRNet+	×2	0	**32.94**	**33.35**	**32.54**	**31.22**	**30.81**	**30.59**	**30.75**	**29.88**	**32.15**	**31.73**
	×3	0	29.09	**29.93**	**30.03**	**29.87**	**29.59**	**29.48**	**29.48**	**29.28**	**28.42**	**28.11**
	×3	3	28.65	**28.89**	**28.43**	**27.77**	**27.57**	**27.50**	**27.55**	**27.05**	27.04	**26.71**
	×3	7	**27.72**	**27.61**	**27.07**	**26.48**	**26.41**	**26.28**	**26.35**	**25.91**	**25.50**	**25.35**
	×4	0	26.94	**27.84**	**28.03**	**28.08**	**27.99**	**27.89**	**27.82**	**28.01**	**26.49**	**26.53**

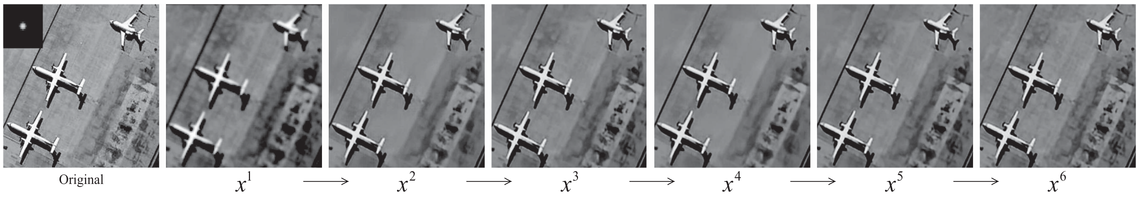


Fig. 16. Initial SR estimation x^0 is the nearest neighbor interpolated version of LR image. The scale factor is 3, and the noise level of LR image is 0. The blurring kernel is shown on the upper-left corner of x^0. x^1–x^6 denote the results in different iterations.

based on USRNet algorithm, we perform the ablation experiments with USRNet. The improvements can be divided into two parts, the attention module, and YSRNet. And YSRNet+ is the combination of the attention module and YSRNet. TV priori denotes the model only using TV prior features for image reconstruction. As shown in Table V, the attention module and YSRNet can improve the algorithm performance to different degrees, verifying the proposed modules' effectiveness. Because TV priori is extracting the edge information of the target images, some image information may be lost. Therefore, the results of image reconstruction only using TV priori are limited. The parameter numbers of base model method, base model with attention method, YSRNet, and YSRNet+ are 592 K, 597 K, 1695 K, and 1707 K, respectively. Therefore, the parameter

TABLE V
ABLATION EXPERIMENT

Method	Scale	Noise	Dataset1	Ave.	Dataset2	Ave.
Deep priori (base model)	×2	0	30.65	27.35	29.20	26.65
	×3	0	28.07		27.16	
	×3	3	26.68		26.04	
	×3	7	25.50		25.19	
	×4	0	25.84		25.67	
TV priori	×2	0	29.79	26.89	28.81	26.53
	×3	0	27.51		27.07	
	×3	3	26.34		26.00	
	×3	7	25.26		25.12	
	×4	0	25.57		25.66	
Attention	×2	0	30.17	27.45	29.23	26.73
	×3	0	28.22		27.23	
	×3	3	26.74		26.18	
	×3	7	25.59		25.22	
	×4	0	25.99		25.78	
YSRNet	×2	0	30.64	27.41	29.33	26.78
	×3	0	28.20		27.28	
	×3	3	26.64		26.18	
	×3	7	25.54		25.25	
	×4	0	26.02		25.84	
YSRNet+	×2	0	30.73	27.52	29.33	26.83
	×3	0	28.28		27.32	
	×3	3	26.86		26.26	
	×3	7	25.68		25.33	
	×4	0	26.06		25.89	

growth of the attention module is worth comparing to the PSNR improvement.

V. CONCLUSION

This article introduces a double prior unfolding strategy for remote sensing image SR. Unlike the previous works, which are unexplainable networks, we propose a double prior unfolding network with interpretability for remote sensing image SR. D module can reconstruct the image and make the image clearer. R module can utilize deep priori and TV priori to remove noise. We combine the traditional optimization algorithm with the neural network and design a network (YSRNet), which has the flexibility of traditional algorithms and the strong fitting ability of neural networks. This unfolding strategy enables the algorithm to better handle multidegeneration SR tasks. In addition, an enhanced version (YSRNet+) is proposed, and an attention module is introduced into the algorithm. Unlike the commonly used attention module, this attention module enables the network to focus on more important features. YSRNet+ enhances the interpretability of the algorithm while improving the performance of the YSRNet. Multiple experiments are performed to evaluate the two proposed networks, from which the following conclusions can be obtained.

1) The proposed YSRNet and YSRNet+ algorithms perform better than state-of-the-art methods. The learning rate of the network should be carefully chosen to avoid the problem of slow or nonconvergence of the network.
2) As the number of iteration layers increases, the model parameters increase significantly, so we need to choose an appropriate number of iteration layers to make a trade-off between the performance and efficiency of the algorithm.
3) When the number of channels in the network is large enough, the performance gained from the attention module will be less noticeable. However, it still works.

Based on the existing network problems, there are the following three aspects in our future work.

1) Using more prior information from traditional methods will be considered, which can make the traditional optimization algorithm better combined with the neural networks.
2) The network framework will be improved to enhance the fitting ability.
3) We will consider the design of lightweight networks to reduce network parameters and training time.

REFERENCES

[1] L. Chen, X. Jiang, Z. Li, X. Liu, and Z. Zhou, "Feature-enhanced speckle reduction via low-rank and space-angle continuity for circular SAR target recognition," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 7734–7752, Nov. 2020.

[2] J. Zhang, M. Xing, and Y. Xie, "FEC: A feature fusion framework for SAR target recognition based on electromagnetic scattering features and deep CNN features," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2174–2187, Mar. 2021.

[3] X. Bai, X. Zhou, F. Zhang, L. Wang, R. Xue, and F. Zhou, "Robust Pol-ISAR target recognition based on ST-MC-DCNN," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9912–9927, Dec. 2019.

[4] C. Cao, Z. Cao, and Z. Cui, "LDGAN: A synthetic aperture radar image generation method for automatic target recognition," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3495–3508, May 2020.

[5] Y. Zhang, Q. Zhang, L. Kang, Y. Luo, and L. Zhang, "End-to-end recognition of similar space cone-cylinder targets based on complex-valued coordinate attention networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5106214, doi: 10.1109/TGRS.2021.3115624.

[6] M. Liu, S. Chen, J. Wu, F. Lu, X. Wang, and M. Xing, "SAR target configuration recognition via two-stage sparse structure representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2220–2232, Apr. 2018.

[7] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[8] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[9] F. Zhang, B. Du, and L. Zhang, "Scene classification via a gradient boosting random convolutional network framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1793–1802, Mar. 2016.

[10] H. Kim and A. Hirose, "Unsupervised fine land classification using quaternion autoencoder-based polarization feature extraction and self-organizing mapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1839–1851, Mar. 2018.

[11] H. Kim and A. Hirose, "Unsupervised hierarchical land classification using self-organizing feature codebook for decimeter-resolution PolSAR," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 1894–1905, Apr. 2019.

[12] K. Wang, J. Chen, A. Kiaghadi, and C. Dawson, "A new algorithm for land-cover classification using PolSAR and InSAR data and its application to surface roughness mapping along the gulf coast," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 4502915, doi: 10.1109/TGRS.2021.3083492.

[13] C. Luo, S. Kumar, and D. N. Mallick, "Achieving excellence in exploration and exploitation by matching appropriate resources," *IEEE Eng. Manag. Rev.*, vol. 46, no. 3, pp. 103–107, Sep. 2018.

[14] W. Xu and Y. Deng, "Multichannel SAR with reflector antenna for high-resolution wide-swath imaging," *IEEE Antennas Wireless Propag. Lett.*, vol. 9, pp. 1123–1126, 2010.

[15] D. Cerutti-Maori, I. Sikaneta, J. Klare, and C. Gierull, "MIMO SAR processing for multichannel high-resolution wide-swath radars," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 5034–5055, Aug. 2014.

[16] W. Xu, J. Hu, P. Huang, W. Tan, and Y. Dong, "Multiaperture antenna architecture design for azimuth uniform sampling in high-resolution wide-swath SAR," *IEEE Antennas Wireless Propag. Lett.*, vol. 19, no. 6, pp. 1042–1046, Jun. 2020.

[17] C. Wang, W. Shi, B. Yuan, and J. Mao, "Pattern-steerable endfire plasma array antenna," *IEEE Trans. Antennas Propag.*, vol. 69, no. 10, pp. 6994–6998, Oct. 2021.

[18] X. Zhu and R. Bamler, "Super-resolution power and robustness of compressive sensing for spectral estimation with application to spaceborne tomographic SAR," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 1, pp. 247–258, Jan. 2012.

[19] Q. Yuan, L. Zhang, and H. Shen, "Multiframe super-resolution employing a spatially weighted total variation model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 379–392, Mar. 2012.

[20] X Zhu and R. Bamler, "Demonstration of super-resolution for tomographic SAR imaging in urban environment," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 8, pp. 3150–3157, Aug. 2012.

[21] M. Veganzones, M. Simões, G. Licciardi, N. Yokoya, J. Bioucas-Dias, and J. Chanussot, "Hyperspectral super-resolution of locally low rank images from complementary multisource data," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 274–288, Jan. 2016.

[22] Y. Wan, W. Wei, and Q. Zheng, "Study on the aerial image degradation caused by satellite swing and its correction," in *Proc. 5th Int. Conf. Comput. Sci. Netw. Technol.*, 2016, pp. 795–799.

[23] F. Palsson, J. Sveinsson, M. Ulfarsson, and J. Benediktsson, "MTF-based deblurring using a wiener filter for CS and MRA pansharpening methods," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 6, pp. 2255–2269, Jun. 2016.

[24] N. Zhang et al., "A multi-degradation aided method for unsupervised remote sensing image super resolution with convolution neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2020, Art. no. 5600814, doi: 10.1109/TGRS.2020.3042460.

[25] Y. Li et al., "Single-image super-resolution for remote sensing images using a deep generative adversarial network with local and global attention mechanisms," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 3000224, doi: 10.1109/TGRS.2021.3093043.

[26] J. Haut, R. Fernandez-Beltran, M. E. Paoletti, J. Plaza, A. Plaza, and F. Pla, "A new deep generative network for unsupervised remote sensing single-image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6792–6810, Nov. 2018.

[27] Z. Pan, W. Ma, J. Guo, and B. Lei, "Super-resolution of single remote sensing image based on residual dense backprojection networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7918–7933, Oct. 2019.

[28] J. Li, R. Cui, Y. Li, B. Li, Q. Du, and C. Ge, "Multitemporal hyperspectral image super-resolution through 3D generative adversarial network," in *Proc. 10th Int. Workshop Anal. Multitemporal Remote Sens. Images*, 2019, pp. 1–4.

[29] S. P. Kim, N. K. Bose, and H. M. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 6, pp. 1013–1027, Jun. 1990.

[30] H. Stark and P. Oskoui, "High-resolution image recovery from image-plane arrays, using convex projections," *JOSA A*, vol. 6, no. 11, pp. 1715–1726, 1989.

[31] A. Bordone Molini, D. Valsesia, G. Fracastoro, and E. Magli, "DeepSUM: Deep neural network for super-resolution of unregistered multitemporal images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3644–3656, May 2020.

[32] X. Liu, T. Feng, X. Shen, and R. Li, "PMDRnet: A progressive multiscale deformable residual network for multi-image super-resolution of AMSR2 arctic sea ICE images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4304118, doi: 10.1109/TGRS.2022.3151623.

[33] L. Xin and M. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[34] T. Blu, P. Thevenaz, and M. Unser, "Linear interpolation revitalized," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 710–719, May 2004.

[35] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.

[36] F. Li, X. Jia, D. Fraser, and A. Lambert, "Super resolution for remote sensing images based on a universal hidden Markov tree model," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1270–1278, Mar. 2010.

[37] R. Schultz and R. Stevenson, "A Bayesian approach to image expansion for improved definition," *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 233–242, May 1994.

[38] Y. Jianchao, J. Wright, T. Huang, and M. Yi, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2008, pp. 1–8.

[39] Z. Zhihui, W. Bo, and S. Kang, "Single remote sensing image super-resolution and denoising via sparse representation," in *Proc. Int. Workshop Multi-Platform/Multi-Sensor Remote Sens. Mapping*, 2011, pp. 1–5.

[40] B. Hou, K. Zhou, and L. Jiao, "Adaptive super-resolution for remote sensing images based on sparse representation with global joint dictionary model," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2312–2327, Apr. 2018.

[41] C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[42] L. Liebel and M. Korner, "Single-image super resolution for multispectral remote sensing data using convolutional neural networks," ISPRS-International Archives of the Photogrammetry, *Remote Sens. Spatial Inf. Sci.*, vol. 41, pp. 883–890, 2016.

[43] C. Dong, C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 391–407.

[44] S. Lei, Z. Shi, and Z. Zou, "Super-resolution for remote sensing images via local–global combined network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1243–1247, Aug. 2017.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2016, pp. 770–778.

[46] J. Kim, J. Lee, and K. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654.

[47] J. Haut, R. Fernandez-Beltran, M. Paoletti, J. Plaza, and A. Plaza, "Remote sensing image superresolution using deep residual channel attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9277–9289, Nov. 2019.

[48] J. Zhang and B. Ghanem, "Ista-net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1828–1837.

[49] O. Solomon et al., "Deep unfolded robust PCA with application to clutter suppression in ultrasound," *IEEE Trans. Med. Imag.*, vol. 39, no. 4, pp. 1051–1063, Apr. 2020.

[50] X. Liu, T. Feng, X. Shen, and R. Li, "PMDRnet: A progressive multiscale deformable residual network for multi-image super-resolution of AMSR2 arctic sea ice images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2022, doi: 10.1109/TGRS.2022.3151623.

[51] I. Marivani, E. Tsiligianni, B. Cornelis, and N. Deligiannis, "Multimodal deep unfolding for guided image super-resolution," *IEEE Trans. Image Process.*, vol. 29, pp. 8443–8456, Jul., 2020, doi: 10.1109/TIP.2020.3014729.

[52] H. Luong, B. Joukovsky, and N. Deligiannis, "Designing interpretable recurrent neural networks for video reconstruction via deep unfolding," *IEEE Trans. Image Process.*, vol. 30, pp. 4099–4113, Mar., 2021, doi: 10.1109/TIP.2021.3069296.

[53] S. Takabe, Y. Yamauchi, and T. Wadayama, "Deep-unfolded sparse CDMA: Multiuser detector and sparse signature design," *IEEE Access*, vol. 9, pp. 40027–40038, 2021.

[54] K. Zhang, L. Van Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2020, pp. 3214–3223.

[55] Y. Zhang, X. Tuo, Y. Huang, and J. Yang, "A TV forward-looking super-resolution imaging method based on TSVD strategy for scanning radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4517–4528, Jul. 2020.

[56] L. Liu, W. Huang, C. Wang, X. Zhang, and B. Liu, "SAR image super-resolution based on tv-regularization using gradient profile prior," in *Proc. CIE Int. Conf. Radar*, 2016, pp. 1–4.

[57] T. Goto, S. Suzuki, S. Hirano, M. Sakurai, and T. Nguyen, "Fast and high quality learning-based super-resolution utilizing TV regularization method," in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 1185–1188.

[58] J. Bai, Y. Wang, and Q. Shi, "Efficient QP-ADMM decoder for binary LDPC codes and its performance analysis," *IEEE Trans. Signal Process.*, vol. 68, pp. 503–518, Dec. 2020, doi: 10.1109/TSP.2020.2964223.

[59] E. Chunikhina, R. Raich, and T. Nguyen, "Performance analysis for matrix completion via iterative hard-thresholded SVD," in *Proc. IEEE Workshop Statist. Signal Process.*, 2014, pp. 392–395.

[60] R. Nadakuditi, "Optshrink: An algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 3002–3018, May 2014.

[61] T. Oh, Y. Matsushita, Y. Tai, and I. Kweon, "Fast randomized singular value thresholding for low-rank optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 376–391, Feb. 2018.

[62] K. Zhang, L. Gool, and R. Timofte, “Deep unfolding network for image super-resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3217–3226.

[63] Z. Zhou, Md Mahfuzur Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet : Redesigning skip connections to exploit multiscale features in image segmentation,” *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020.

[64] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.

[65] K. Zhang, W. Zuo, and L. Zhang, “Deep plug-and-play super-resolution for arbitrary blur kernels,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1671–1681.

[66] Z. Hui, X. Gao, Y. Yang, and X. Wang, “Lightweight image super-resolution with information multi-distillation network,” in *Proc. 27th ACM Int. conf. Multimedia*, 2019, pp. 2024–2032.

[67] K. Zhang, W. Zuo, and L. Zhang, “Learning a single convolutional super-resolution network for multiple degradations,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3262–3271.

[68] Y. Wang, Y. Li, G. Wang, and X. Liu, “Multi-scale attention network for single image super-resolution,” 2022, *arXiv:2209.14145*.

[69] S. Lei and Z. Shi, “Hybrid-scale self-similarity exploitation for remote sensing image super-resolution,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5401410, doi: 10.1109/TGRS.2021.3069889.

Mengyang Shi (Student Member, IEEE) received the B.S. degree in information engineering from Xidian University, Xi’an, China, in 2018. He is currently working toward the Ph.D. degree in radar signal processing and image processing with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China.

His research interests include deep learning, sparse representation, and remote sensing super-resolution.

Yesheng Gao (Member, IEEE) received the B.S. and M.S. degrees in communication and information system from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2005 and 2008, respectively, and the Ph.D. degree in signal and information processing from Shanghai Jiao Tong University, Shanghai, China, in 2013.

He was currently a Research Assistant with Shanghai Jiao Tong University. His research interests include synthetic aperture radar processing, feature extraction, and new concept radar system.

Lin Chen (Student Member, IEEE) received the B.S. degree in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2017. He is currently working toward the Ph.D. degree in image processing and low rank representation with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China.

His research interests include remote sensing, signal processing, numerical optimization, and machine learning.

Xingzhao Liu (Member, IEEE) received the B.S. and M.S. degrees from the Harbin Institute of Technology, Harbin, China, in 1984 and 1992, respectively, and the Ph.D. degree from the University of Tokushima, Tokushima, Japan, in 1995, all in electrical engineering.

From 1984 to 1998, he was an Assistant Professor, an Associate Professor, and a Professor with the Harbin Institute of Technology. Since 1998, he has been a Professor with Shanghai Jiao Tong University, Shanghai, China. His research interests include high-frequency radar and synthetic aperture radar signal processing.