Deep Gradient Projection Networks for Pan-sharpening

Shuang Xu, Jiangshe Zhang*, Zixiang Zhao, Kai Sun, Junmin Liu, Chunxia Zhang
School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China
{shuangxu,zixiangzhao,}@stu.xjtu.edu.cn, {jszhang,kaisun,junminliu,cxzhang}@mail.xjtu.edu.cn

Abstract

Pan-sharpening is an important technique for remote sensing imaging systems to obtain high resolution multispectral images. Recently, deep learning has become the most popular tool for pan-sharpening. This paper develops a model-based deep pan-sharpening approach. Specifically, two optimization problems regularized by the deep prior are formulated, and they are separately responsible for the generative models for panchromatic images and low resolution multispectral images. Then, the two problems are solved by a gradient projection algorithm, and the iterative steps are generalized into two network blocks. By alternatively stacking the two blocks, a novel network, called gradient projection based pan-sharpening neural network, is constructed. The experimental results on different kinds of satellite datasets demonstrate that the new network outperforms state-of-the-art methods both visually and quantitatively. The codes are available at `https://github.com/xsxjtu/GPPNN`.

1. Introduction

Multispectral images store multiple images corresponding to each band (or say, channel) in an optical spectrum, and they are widely utilized in literature of remote sensing. With the limitation of imaging devices, satellites however often measure the low spatial resolution multispectral (LRMS) images [4, 21, 29]. Compared with the multispectral image, the panchromatic (PAN) image is characterized by the high spatial resolution but only one band. Lots of satellites carry both multispectral and panchromatic sensors to simultaneously measure the complementary images, such as Landsat8, GaoFen2 and QuickBird. To obtain the high resolution multispectral (HRMS) image, a promising way is to fuse the complementary information of the LRMS image and the PAN image. This technique is called as *pan-sharpening* [4, 21].

Pan-sharpening can be cast as a typical image fusion on super-resolution problems. The past decades witnessed the development of pan-sharpening. The classic algorithms including component substitution (CS) [7, 12], multiresolution analysis (MRA) [1, 23] and other techniques. In the era of deep learning, convolutional neural networks have emerged as a significant tool for pan-sharpening [19]. One of the seminal work is the pan-sharpening neural network (PNN) proposed by Masi *et al*. [20]. Borrowing the idea of the first super-resolution network [2], PNN is fed with the concatenation of a PAN image and an upsampled LRMS

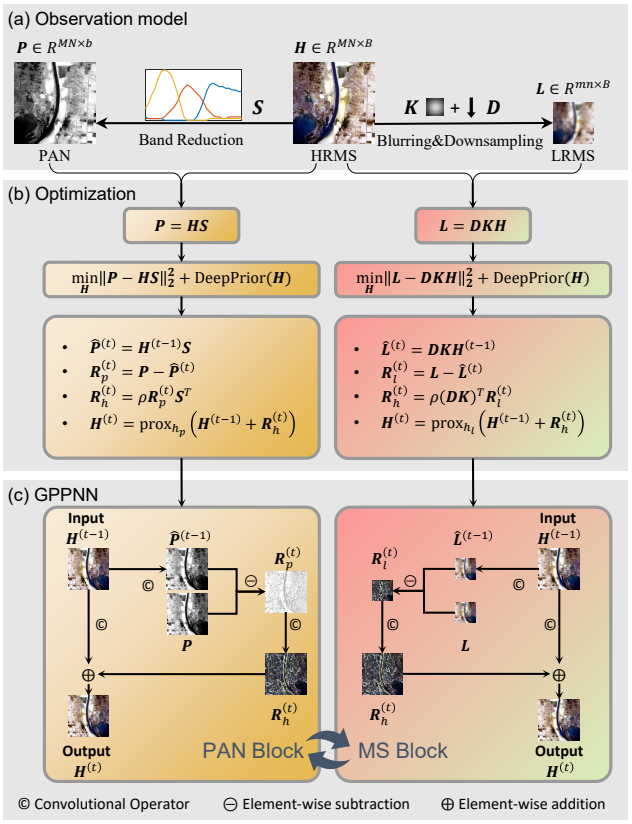


Figure 1. (a) The observation models for LRMS and PAN images. (b) Two formulated optimization problems and iterative steps of the gradient projection algorithm. (c) The main blocks in our proposed GPPNN.

*Corresponding author.

image to regress the HRMS image.

In fact, there are only three convolutional units in PNN, so it is a relatively shallow network. Recently, numerous models have been proposed to improve the PNN. Owing to the triumphs of residual networks [8], several papers utilize the shortcut or residual convolutional units to build deep networks, including MIPSM [16], DRPNN [34] and PanNet [38]. They generally contain 10 or more convolutional units. Besides these networks, to make the best of advantages of neural networks, some researchers build deeper networks. For example, Wang *et al*. employ the densely connected convolutional unit [10] to design a 44-layer network [31] for pan-sharpening.

It is well-known that deepening the layers of networks does not necessarily improve the performance, since it is difficult to train deeper networks and redundant parameters make them easily over-fit. Very recently, the remote sensing community begins to rethink how to make the full use of PAN images' information [9, 24]. It is worthy noting that most the pan-sharpening networks regard the PAN image as a channel of the input. This manner ignores different characteristics between PAN and LRMS images. A growing number of researchers attempt to propose the two-branch networks [18, 40]. In the first stage, the two branches separately extract the features for PAN and LRMS images. In the second stage, the features are fused to reconstruct the HRMS image.

Although convolutional neural networks exhibit promising performance in pan-sharpening, they require a large amount of training samples [22, 33], and they do not account for the observation progress of PAN and LRMS images, i.e., lacking the interpretability. Therefore, there still leaves the room for improvement. The research on model-based deep learning is the trend in image processing field to close the gap between classic models and neural networks, and it is found that model-based deep networks usually outperform the intuitively designed networks [22, 33]. Xie *et al*. present a multispectral and hyperspectral (HS) image fusion network (MHNet) for the *hyperspectral pan-sharpening* task [35]. There is no doubt that MHNet can be naturally adapted to pan-sharpening [36]. Nonetheless, MHNet is designed to describe the low-rank property for hyperspectral images, and our experiments show that MHNet may perform badly in the pan-sharpening scenario.

In this paper, we develop a novel model-based deep network for pan-sharpening. Our contributions are summarized as follows:

Firstly, this paper considers the generative models for PAN and LRMS images. That is, as shown in Fig. 1(a), PAN images are the linear combination of the bands in HRMS images, and LRMS images are generated by blurring and downsampling HRMS images. Combining the observation models and the deep prior, we propose two optimization problems, and they can be effectively solved by the gradient projection method as illustrated in Fig. 1(b).

Secondly, inspired by the idea of algorithm unrolling techniques, the iterative steps are generalized as two neural blocks separately justifying the generative models for PAN and LRMS images. The computational flows in the proposed neural blocks are interpretable. As show in Fig. 1(c), for the MS Block, given a current estimation of the HRMS image, it generates corresponding LRMS image and computes the residual between the generated LRMS image and the real one. This residual then is upsampled and is added into the current estimation to reconstruct the next HRMS image. The PAN block can be interpreted similarly. We build a new network by alternatively stacking the two blocks. In what follows, it calls the gradient projection based pan-sharpening neural network (GPPNN). To the best of our knowledge, it is the first model-driven deep network for pan-sharpening.

Thirdly, the proposed GPPNN is compared with the 13 state-of-the-art (SOTA) and classic pan-sharpening methods. The extensive experiments conducted on three popular satellites (i.e., Landsat8, QuickBird, GF2) demonstrate that our networks outperform other counterparts both quantitatively and visually.

2. Related work

2.1. Classic pan-sharpening methods

The classic pan-sharpening methods mainly consists of CS based algorithms, MRA based algorithms and other algorithms. CS methods assume that the spatial and spectral information of a multispectral image can be decomposed. Therefore, an HRMS image is reconstructed by combining the spatial information of a PAN image and the spectral information of an LRMS image. In the past decades, researchers have designed various decomposition algorithms. For example, intensity-hue-saturation (IHS) fusion [7] employs the IHS transformation, Brovey method [5] uses a multiplicative injection scheme, and Gram-Schmidt (GS) method [12] exploits the Gram-Schmidt orthogonalization procedure. The main drawback of CS methods is that the image contains artifacts if the spectral and spatial information is not appropriately decomposed. The MRA methods apply the multi-scale transformation to PAN images to extract spatial details which then are injected into the upsampled LRMS images. Typical algorithms include high-pass filter (HPF) fusion [26], and Indusion method [11], smoothing filter-based intensity modulation (SFIM) [15]. The performance of the MRA method strongly depends on the multi-scale transformation.

2.2. Deep learning based methods

Recently, convolutional neural networks have been one of the most effective tools for remote sensing. Given a parameterized network, it is fed with an LRMS image and a PAN image to regress an HRMS image, and its parameters (or say, weights) are learned from data in the end-to-end fashion. The first attempt is the PNN with three convolutional units [20]. Recently, thanks to the rapid development of computer vision [8, 10], it is able to train very deep networks. Researchers propose the deep pan-sharpening networks with dozens of layers and the performance has been greatly improved [27, 31, 38, 40]. At the same time, researchers also explore the two-branch networks to separately extract the features from MS and PAN images [18, 40]. Recently, one of the research trends of the pan-sharpening community is to combine the classic methods with deep neural networks to improve the interpretability of the deep learning based methods. For example, inspired by the idea of MRA algorithms, MIPSM [16] designs a spatial detail extraction network for the PAN images and injects the details into the LRMS images. Liu *et al*. propose an adaptive weight network for integrating the advantages of different classic methods [14]. It overcomes the shortcomings of the CS and MRA algorithms, and outperforms some SOTA deep learning based methods.

2.3. Model-driven deep networks

Most of the deep neural networks are designed intuitively. Recently, a growing number of researchers focus on model-based neural networks for image processing tasks [22, 33]. The basic idea of model-driven deep learning is to formulate an observation model or optimization problem by integrating the prior knowledge for a specific task and to translate each iteration of the algorithm step into a layer of deep neural networks [22, 33]. Passing through the stacked layers corresponds to execute the algorithm with a certain number of times. Model-based deep learning builds the bridge between classic models and deep neural networks. This idea has been successfully applied in various tasks, including sparse coding [6], compressive sensing [39], image deblurring [13], image dehazing [37] and image deraining [32]. It is worth mentioning the MHNet, a model-driven network for the hyperspectral pan-sharpening task [35] to super-resolve HS images with the guidance of MS images. It can be naturally adapted to pan-sharpening, but MHNet mainly focuses on the low-rank property for HS images, i.e., its rank r_{HS} is far lower than the number of bands B_{HS}. In practice, there are dozens or hundreds of bands in an HS image, while there are no more than 10 bands in an MS image. So, the low-rank property is not evident for MS images, and MHNet may break down in pan-sharpening task.

3. GPPNN

In this section, we develop a model-driven network for pan-sharpening. For convenience, we summarize the notations in this paper before presentation of the GPPNN. $\boldsymbol{L} \in R^{mn\times B}$ is an LRMS image with a height of m, a width of n and the number of bands of B. $\boldsymbol{H} \in R^{MN\times B}$ is an HRMS image with a height of M, a width of N and the number of bands of B. $\boldsymbol{P} \in R^{MN\times b}$ is a PAN image whose spatial resolution is the same with that of $\boldsymbol{H}$, but there is only one band (i.e., $b = 1$). $r = M/m = N/n$ is the spatial resolution ratio. With abuse of notations, we use their tensor versions in the context of deep learning (namely, $\mathcal{L} \in R^{m\times n\times B}, \mathcal{H} \in R^{M\times N\times B}, \mathcal{P} \in R^{M\times N\times b}$). Notation $\texttt{Conv}(\cdot; c^{\mathrm{in}}, c^{\mathrm{out}})$ is the convolutional operator whose input and output are with c^{in} and c^{out} channels, respectively. In what follows, the function $\texttt{Conv}(\cdot; c^{\mathrm{in}}, c^{\mathrm{mid}}, c^{\mathrm{out}})$ denotes the cascaded convolutional operator, that is,

$$\begin{aligned}&\texttt{Conv}\left(\cdot; c^{\mathrm{in}}, c^{\mathrm{mid}}, c^{\mathrm{out}}\right)\\ =&\texttt{Conv}\left(\texttt{ReLU}\left(\texttt{Conv}\left(\cdot; c^{\mathrm{in}}, c^{\mathrm{mid}}\right)\right); c^{\mathrm{mid}}, c^{\mathrm{out}}\right).\end{aligned} \tag{1}$$

3.1. Model formulation

Our network starts with the observation model for the LRMS, HRMS and PAN images. It is assumed that an LRMS image is obtained by downsampling and blurring an HRMS image, while a PAN image is the result of spectral response for an HRMS image. In formula, we have $\boldsymbol{L} = \boldsymbol{DKH}, \boldsymbol{P} = \boldsymbol{HS}$, where $\boldsymbol{D} \in R^{mn\times MN}$ denotes a downsampling matrix and $\boldsymbol{K}$ is the (low-passing) circular convolution matrix, and $\boldsymbol{S} \in R^{B\times b}$ is the so-called spectral response function. It is well-known that inferring the HRMS image is an ill-posed problem. Hence, it often formulates the following penalized optimization,

$$\min_{\boldsymbol{H}} f(\boldsymbol{L}, \boldsymbol{H}) + g(\boldsymbol{P}, \boldsymbol{H}) + \lambda h(\boldsymbol{H}), \tag{2}$$

where $h(\cdot)$ is the prior term, and $f(\boldsymbol{L}, \boldsymbol{H}) = \|\boldsymbol{L} - \boldsymbol{DKH}\|_2^2/2$ and $g(\boldsymbol{P}, \boldsymbol{H}) = \|\boldsymbol{P} - \boldsymbol{HS}\|_2^2/2$ are data fidelity terms which are responsible to LRMS and PAN images, respectively. In the classic methods, $h(\cdot)$ is usually designed as a hand-craft function, such as the total variation or nuclear norm [17]. However, in the era of deep learning, it is suggested to set $h(\cdot)$ as a deep prior [28, 41]. In other words, it is better to set an implicit prior captured by the neural network parametrization. Additionally, the deep prior is learned from data and can adapt to different tasks and observation models. To make the best of deep prior, instead of the above issue, we consider an LRMS-aware problem and a PAN-aware problem:

$$\min_{\boldsymbol{H}} \frac{1}{2}\|\boldsymbol{L} - \boldsymbol{DKH}\|_2^2 + \lambda h_l(\boldsymbol{H}), \tag{3a}$$

$$\min_{\boldsymbol{H}} \frac{1}{2}\|\boldsymbol{P} - \boldsymbol{HS}\|_2^2 + \lambda h_p(\boldsymbol{H}). \tag{3b}$$

Here, $h_l(\cdot)$ and $h_p(\cdot)$ are two deep priors accounting for the observations of LRMS and PAN images, respectively. The ablation experiment in section 4.4 verifies that Eq. (3) achieves better results than Eq. (2). In the next, we describe how to solve the two problems. Moreover, the solutions are generalized into an LRMS-aware block (MS Block) and a PAN-aware block (PAN block) that can be embedded into neural networks.

3.2. MS Block

We employ the gradient projection method [25] to solve Eq. (3a) and the updating rule is

$$\boldsymbol{H}^{(t)} = \text{prox}_{h_l}\left(\boldsymbol{H}^{(t-1)} - \rho\nabla f(\boldsymbol{H}^{(t-1)})\right), \tag{4}$$

where ρ is the step size, $\text{prox}_{h_l}(\cdot)$ is a proximal operator corresponding to penalty $h_l(\cdot)$ and $\nabla f(\boldsymbol{H}^{(t-1)}) = -(\boldsymbol{DK})^T(\boldsymbol{L} - \boldsymbol{DKH})$ denotes the gradient of the data fidelity term.

Inspired by the principle of model-driven deep learning [22], we generalize Eq. (4) as a network block. To begin with, Eq. (4) is split into four steps as follows,

$$\hat{\boldsymbol{L}}^{(t)} = \boldsymbol{DKH}^{(t-1)}, \tag{5a}$$

$$\boldsymbol{R}_l^{(t)} = \boldsymbol{L} - \hat{\boldsymbol{L}}^{(t)}, \tag{5b}$$

$$\boldsymbol{R}_h^{(t)} = \rho\left(\boldsymbol{DK}\right)^T\boldsymbol{R}_l^{(t)}, \tag{5c}$$

$$\boldsymbol{H}^{(t)} = \text{prox}_{h_l}\left(\boldsymbol{H}^{(t-1)} + \boldsymbol{R}_h^{(t)}\right). \tag{5d}$$

Then, each step is translated with deep learning terminologies. For convenience, we use the tensor versions to represent the variables in the context of deep learning. In Eq. (5a), given a current HRMS image $\mathcal{H}^{(t-1)}$, it generates an LRMS image $\hat{\mathcal{L}}^{(t)}$ by applying a low-passing filter and downsampling. In neural networks, this step is implemented by

$$\hat{\mathcal{L}}^{(t)} = \texttt{Conv}\left(\mathcal{H}^{(t-1)}; B, C, B\right)\downarrow, \tag{6}$$

where downsampling is conducted with a bicubic interpolation $\downarrow$ and the filter $\boldsymbol{K}$ is replaced by a cascaded convolutional operator $\texttt{Conv}(\cdot; B, C, B)$ to obtain more expressive features. C is the number of channels for the feature maps, and we set it to 64 in this paper. B is the number of channels for MS images, and it depends on the input data.

Afterwards, Eq. (5b) computes residuals between the real LRMS image $\mathcal{L}$ and the generated LRMS image $\hat{\mathcal{L}}^{(t)}$, and the translation is trivial as shown in following equation,

$$\mathcal{R}_l^{(t)} = \mathcal{L} - \hat{\mathcal{L}}^{(t)}. \tag{7}$$

In the next, Eq. (5c) obtains the high-resolution residuals. Analogous to Eqs. (5a) and (6), this step is rewritten as

$$\mathcal{R}_h^{(t)} = \rho\texttt{Conv}\left(\mathcal{R}_l^{(t)}; B, C, B\right)\uparrow. \tag{8}$$

Remark that the filters in Eqs. (5a) and (5c) transpose to each other, but we do not force the convolutional kernels in Eqs. (6) and (8) to satisfy this requirement for flexibility. The ablation experiment in section 4.4 shows that it slightly improves GPPNN's performance. At last, Eq. (5d) outputs the HRMS image by taking the residual into account with a proximal operator. As illustrated before, proximal operators regarding the deep prior are modeled by the deep networks [28, 41]. In this manner, the deep prior can be learned implicitly from data. So, we have

$$\mathcal{H}^{(t)} = \texttt{Conv}\left(\mathcal{H}^{(t-1)} + \mathcal{R}_h^{(t)}; B, C, B\right). \tag{9}$$

In what follows, Eqs. (6), (7), (8) and (9) are named as an MS Block. For better understanding, the computational flow for an MS Block is displayed in Fig. 1(c).

3.3. PAN block

In this subsection, we consider the observation model for PAN (i.e., Eq. (3b)). With the gradient projection method, the updating rule is

$$\boldsymbol{H}^{(t)} = \text{prox}_{h_p}\left(\boldsymbol{H}^{(t-1)} - \rho\nabla g(\boldsymbol{H}^{(t-1)})\right), \tag{10}$$

where $\nabla g(\boldsymbol{H}^{(t-1)}) = -(\boldsymbol{P} - \boldsymbol{HS})\boldsymbol{S}^T$. With the similar techniques, it is able to translate Eq. (10) into a block of neural networks. At first, Eq. (10) is split into four steps as follows,

$$\hat{\boldsymbol{P}}^{(t)} = \boldsymbol{H}^{(t-1)}\boldsymbol{S}, \tag{11a}$$

$$\boldsymbol{R}_p^{(t)} = \boldsymbol{P} - \hat{\boldsymbol{P}}^{(t)}, \tag{11b}$$

$$\boldsymbol{R}_h^{(t)} = \rho\boldsymbol{R}_p^{(t)}\boldsymbol{S}^T, \tag{11c}$$

$$\boldsymbol{H}^{(t)} = \text{prox}_{h_p}\left(\boldsymbol{H}^{(t-1)} + \boldsymbol{R}_h^{(t)}\right). \tag{11d}$$

In the context of deep learning, as shown in Fig. 1(c), these steps are rewritten as,

$$\hat{\mathcal{P}}^{(t)} = \texttt{Conv}\left(\mathcal{H}^{(t-1)}; B, C, b\right), \tag{12a}$$

$$\mathcal{R}_p^{(t)} = \mathcal{P} - \hat{\mathcal{P}}^{(t)}, \tag{12b}$$

$$\mathcal{R}_h^{(t)} = \rho\texttt{Conv}\left(\mathcal{R}_p^{(t)}; b, C, B\right), \tag{12c}$$

$$\mathcal{H}^{(t)} = \texttt{Conv}\left(\mathcal{H}^{(t-1)} + \mathcal{R}_h^{(t)}; B, C, B\right). \tag{12d}$$

Here, $b = 1$ is the number of channel for PAN images. Remark that the underlying assumption of Eq. (3b) is that the PAN image is a linear combination of the HRMS image. $\boldsymbol{S}/\boldsymbol{S}^T$ is regarded as a band reduction/expansion operator. With this assumption, convolutional units in Eqs. (12a) and (12c) should be with the kernel size of 1.

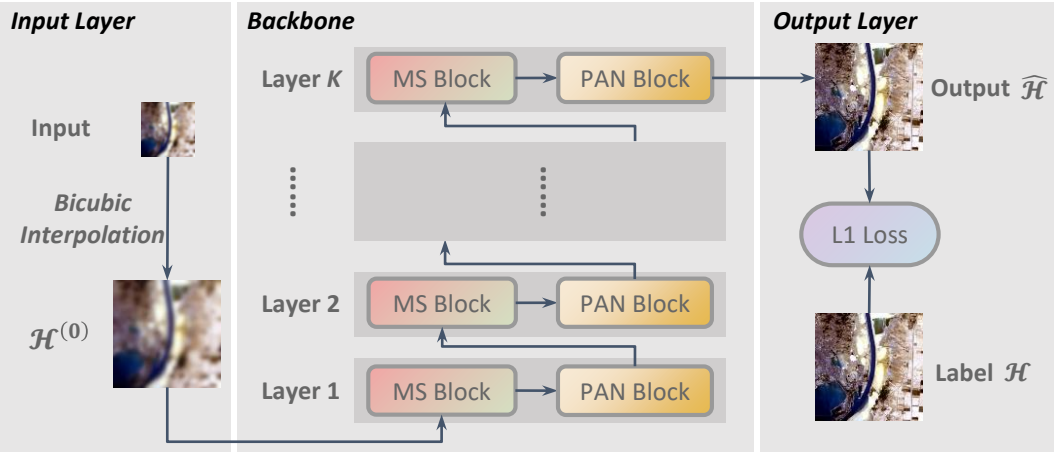


Figure 2. The structure of GPPNN.

3.4. GPPNN

Now, with the MS Block and the PAN block, we are ready to construct the gradient projection based pan-sharpening neural network (GPPNN). The structure of our GPPNN is shown in Fig. 2. The network starts with an input layer, and it requires an initial value of the HRMS image. We initialize $\mathcal{H}^{(0)} \in R^{M\times N\times B}$ by applying the bicubic interpolation to the input LRMS image $\mathcal{L} \in R^{m\times n\times B}$. The network is followed by a backbone subnetwork. There are K layers, each of which consists of an MS Block and a PAN block. In formula, there are

$$\mathcal{H}^{(t+0.5)} = \texttt{MS\_Block}\left(\mathcal{H}^{(t)}, \mathcal{L}\right) \tag{13}$$

and

$$\mathcal{H}^{(t+1)} = \texttt{PAN\_Block}\left(\mathcal{H}^{(t+0.5)}, \mathcal{P}\right). \tag{14}$$

The output of the last layer, denoted by $\hat{\mathcal{H}} \in R^{M\times N\times B}$, is the final reconstructed HRMS.

3.5. Training details

Our GPPNN is supervised by the ℓ_1 loss between $\hat{\mathcal{H}}$ and the ground truth $\mathcal{H}$, $\|\hat{\mathcal{H}} - \mathcal{H}\|_1$. The paired training samples are unavailable in practice. When we construct the training set, the Wald protocol [30] is employed to generate the paired samples. For example, given the multispectral image $\mathcal{H} \in R^{M\times N\times B}$ and the PAN image $\mathcal{P} \in R^{rM\times rN\times b}$, both of them are downsampled with ratio r, and the downsampled versions are denoted by $\mathcal{L} \in R^{M/r\times N/r\times B}$ and $\tilde{\mathcal{P}} \in R^{M\times N\times b}$. In the training set, $\mathcal{L}$ and $\tilde{\mathcal{P}}$ are regarded as the inputs, while $\mathcal{H}$ is the ground-truth.

GPPNN is implemented with Pytorch framework. They are optimized by Adam over 100 epochs with a learning rate of 5×10^{-4} and a batch size of 16. In our experiments, $k_{\text{LR}} = 3$ and $k_{\text{PAN}} = 1$. In section 4.2, we report the performance of GPPNN with different C and K. As a balance, C and K are set to 64 and 8, respectively.

Table 1. The information of datasets. B is the number of bands for multispectral images.

	Landsat8	GaoFen2	QuickBird
B	10	4	4
Resolution-MS	256	256	256
Resolution-PAN	512	1024	1024
# Train/Val/Test	350/50/100	350/50/100	474/103/100

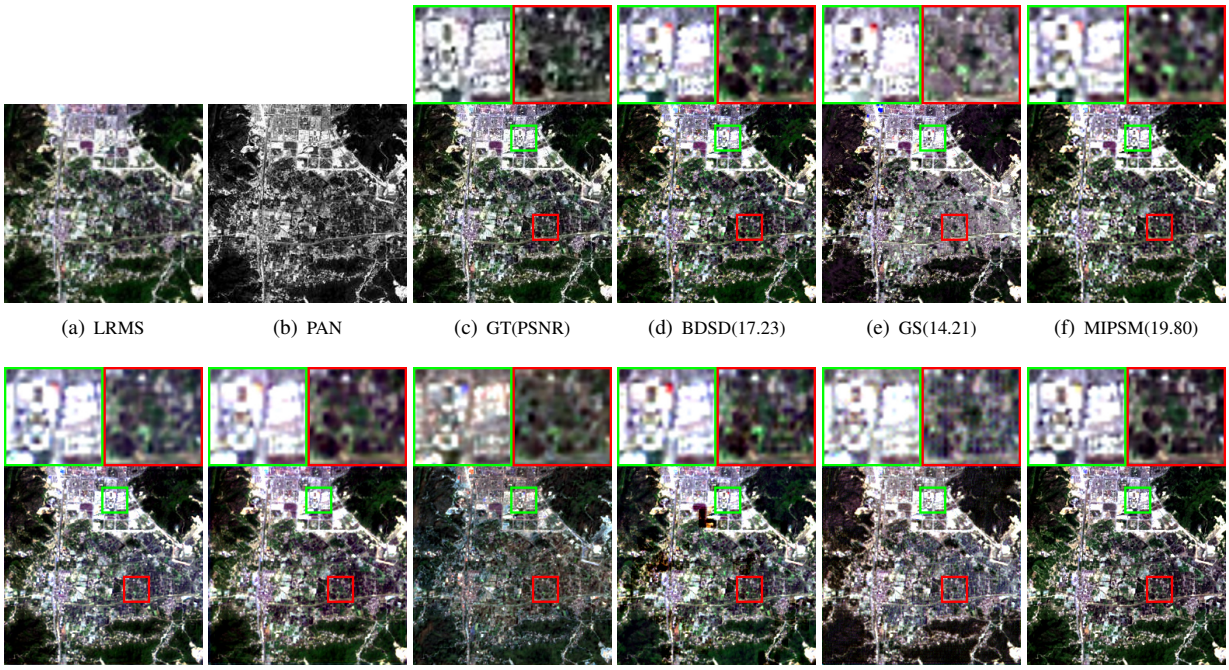
4. Experiments

A series of experiments are carried out to evaluate the performance of GPPNN. SOTA deep learning based methods are selected for comparison, namely, MIPSM [16], DRPNN [34], MSDCNN [40], RSIFNN [27], PanNet [38], and MHNet [35]. Our method is also compared with seven classic methods, including BDSD [3], Brovey [5], GS [12], HPF [26], IHS fusion [7], Indusion [11], SFIM [15]. The experiments are conducted on a computer with an Intel i7-9700K CPU at 3.60GHz and an NVIDIA GeForce RTX 2080ti GPU.

4.1. Datasets and metrics

Remote sensing images acquired by three satellites are used in our experiments, including Landsat8, QuickBird and GaoFen2, the basic information of which is listed in Table 1. For each satellite, we have hundreds of image pairs, and they are divided into three parts for training, validation and test. Note that we determine K and C on the validation dataset. In the training set, the multispectral images are cropped into patches with the size of 32×32, and the corresponding PAN patches are with the size of 64×64 (for Landsat8) or 128×128 (for GaoFen2 and QuickBird). For numerical stability, each patch is normalized by dividing the maximum value to make the pixels range from 0 to 1.

Four popular metrics are used to evaluate the algorithms' performances, including peak signal-to-noise ra-



(a) LRMS (b) PAN (c) GT(PSNR) (d) BDSD(17.23) (e) GS(14.21) (f) MIPSM(19.80)

(g) DRPNN(21.45) (h) MSDCNN(21.94) (i) RSIFNN(17.14) (j) PANNET(20.09) (k) MHNet(19.58) (l) GPPNN(24.40)

Figure 3. Visual inspection on Landsat8 dataset. The caption of each subimage displays the corresponding PSNR value.

Table 2. The PSNR values on validation datasets for GPPNN with different K and C. The best value is highlighted by the **bold**.

Satellites	K layers							C Filters				
	2	4	6	8	10	12	14	8	16	32	64	128
Landsat8	39.0648	39.5878	39.9876	40.0368	**40.1336**	39.9531	40.0509	36.6455	39.6156	39.6702	**40.0368**	39.0841
QuickBird	30.4994	30.4392	30.6370	**30.5636**	30.4803	30.4773	30.5560	30.2962	30.4681	30.4592	30.5636	**30.5979**
GaoFen2	36.7583	36.9740	36.2181	**37.5606**	37.0589	36.7835	36.6840	35.8116	36.9061	36.2810	**37.5606**	36.5873

tio (PSNR), structural similarity (SSIM) and erreur relative globale adimensionnelle de synthese (ERGAS) and spectral angle mapper (SAM). The first three metrics measure the spatial distortion, and the last one measures the spectral distortion. An image is better if its PSNR and SSIM are higher, and ERGAS and SAM are lower.

4.2. The effect of depth and width

The network's depth K and width C play significant roles. Table 2 lists the PSNR values on validation datasets for GPPNN with different K and C. At first, C is fixed to 64, and K is set to 2, 4, $\cdots$, 14. It is shown that more layers do not necessarily increase the PSNR value, and $K = 8$ strikes the balance between performance and the number of weights. The reason may be that it is not easy to train a GPPNN with more layers. Then, we fix K to 8 and set C to 8, 16, 32, 64 and 128. The similar conclusion can be drawn, and the best choice for C is 64. In summary, our GPPNN is configured with $K = 8$ layers and $C = 64$ filters in the next experiments.

4.3. Comparison with SOTA methods

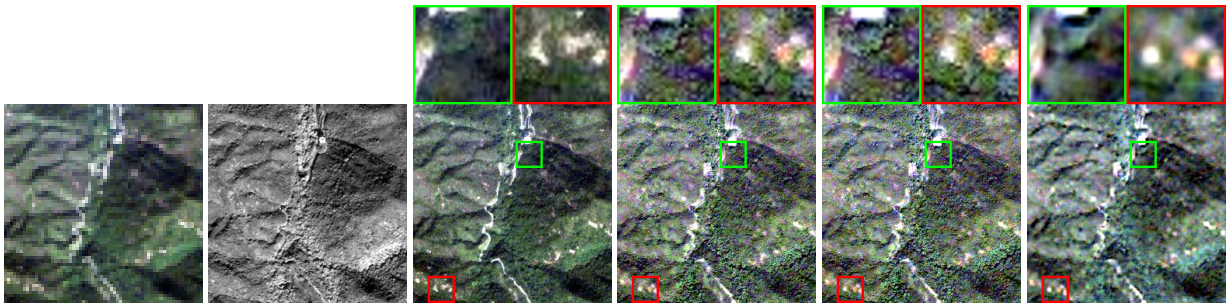
The evaluation metrics on three datasets are reported in Table 3. It is found that GPPNN outperforms other methods regarding all metrics on three satellites. Figs. 3, 4 and 5 show the RGB bands of the three satellites for some representative methods. Our GPPNN is the closest to the ground truth. From the amplified local regions in Fig. 3, it found that BDSD, GS, MIPSM, DRPNN, PANNET suffer from spatial distortion, and GS, MSDCNN, RSIFNN and MHNet suffer from spectral distortion. However, our GPPNN has the smallest spatial and spectral distortions. As for Fig. 4, it is a difficult case. It is shown that the most methods have obvious artifacts or noise, and their images are blurring or spectrally distorted. Our GPPNN is without artifacts, noise or spectral distortion. As shown in Fig. 5, it is observed that compared with other methods, our GPPNN has finer-grained textures and coarser-grained structures.

4.4. Ablation experiments

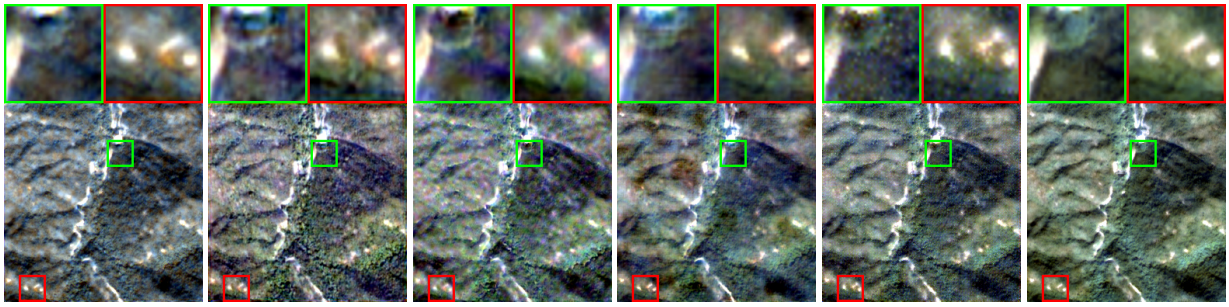
To further investigate the role of some modules in the proposed GPPNN, a series of ablation experiments are car-

Table 3. The four metrics on test datasets. The best and the second best values are highlighted by the **bold** and underline, respectively. The up or down arrow indicates higher or lower metric corresponds to better images.

	Landsat8				QuickBird				GaoFen2			
	PSNR↑	SSIM↑	SAM↓	ERGAS↓	PSNR↑	SSIM↑	SAM↓	ERGAS↓	PSNR↑	SSIM↑	SAM↓	ERGAS↓
BDSD	33.8065	0.9128	0.0255	1.9128	23.5540	0.7156	0.0765	4.8874	30.2114	0.8732	0.0126	2.3963
Brovey	32.4030	0.8533	0.0206	1.9806	25.2744	0.7370	0.0640	4.2085	31.5901	0.9033	0.0110	2.2088
GS	32.0163	0.8687	0.0304	2.2119	26.0305	0.6829	0.0586	3.9498	30.4357	0.8836	0.0101	2.3075
HPF	32.6691	0.8712	0.0250	2.0669	25.9977	0.7378	0.0588	3.9452	30.4812	0.8848	0.0113	2.3311
IHS	32.8772	0.8615	0.0245	2.3128	24.3826	0.6742	0.0647	4.6208	30.4754	0.8639	0.0108	2.3546
Indusion	30.8476	0.8168	0.0359	2.4216	25.7623	0.6377	0.0674	4.2514	30.5359	0.8849	0.0113	2.3457
SFIM	32.7207	0.8714	0.0248	2.0775	24.0351	0.6409	0.0739	4.8282	30.4021	0.8501	0.0129	2.3688
MIPSM	35.4891	0.9389	0.0209	1.5769	27.7323	0.8411	0.0522	3.1550	32.1761	0.9392	0.0104	1.8830
DRPNN	37.3639	0.9613	0.0173	1.3303	31.0415	0.8993	0.0378	2.2250	35.1182	0.9663	0.0098	1.3078
MSDCNN	36.2536	0.9581	0.0176	1.4160	30.1245	0.8728	0.0434	2.5649	33.6715	0.9685	0.0090	1.4720
RSIFNN	37.0782	0.9547	0.0172	1.3273	30.5769	0.8898	0.0405	2.3530	33.0588	0.9588	0.0112	1.5658
PANNET	38.0910	0.9647	0.0152	1.3021	30.9631	0.8988	0.0368	2.2648	34.5774	0.9635	0.0089	1.4750
MHNet	37.0049	0.9566	0.0189	1.3509	31.1557	0.8947	0.0368	2.1931	33.8930	0.9291	0.0176	1.3697
GPPNN	**38.9939**	**0.9727**	**0.0138**	**1.2483**	**31.4973**	**0.9075**	**0.0351**	**2.1058**	**35.9680**	**0.9725**	**0.0084**	**1.2798**



(a) LRMS (b) PAN (c) GT(PSNR) (d) BDSD(16.23) (e) GS(15.80) (f) MIPSM(16.84)



(g) DRPNN(18.85) (h) MSDCNN(18.15) (i) RSIFNN(17.33) (j) PANNET(18.22) (k) MHNet(19.50) (l) GPPNN(21.13)

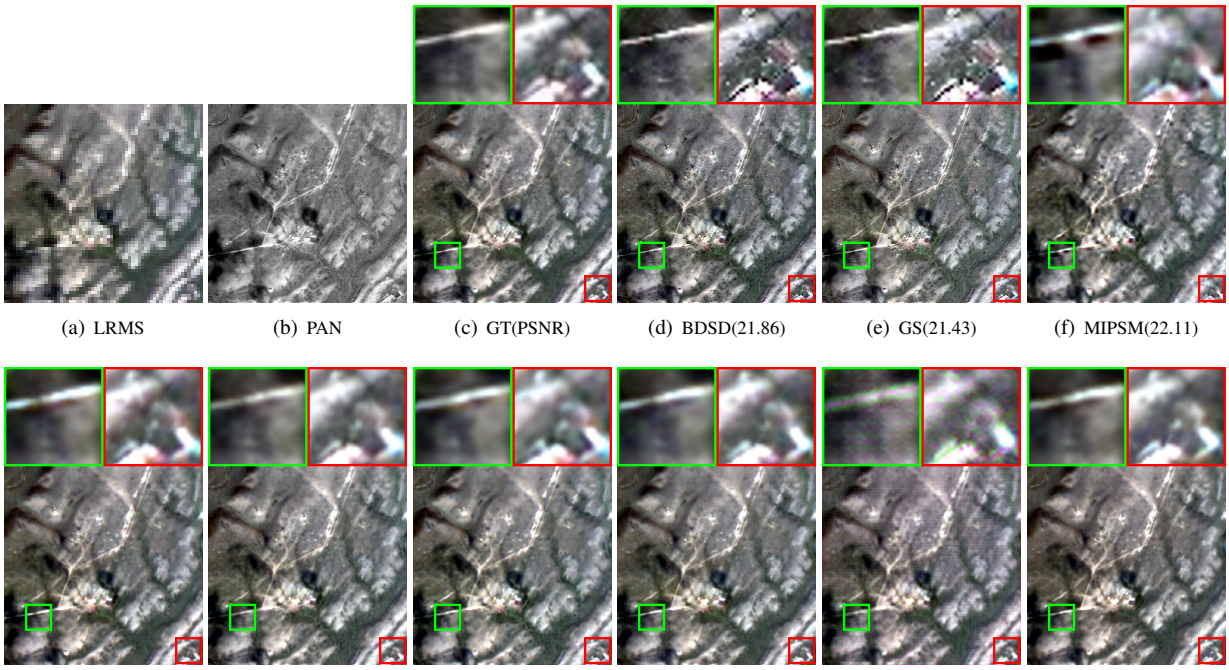
Figure 4. Visual inspection on QuickBird dataset. The caption of each subimage displays the corresponding PSNR value.

ried out. There are 5 different configurations and the results of ablation experiments are shown in Table 4.

(I) The proximal operators make the current HRMS image restricted to deep priors. In the first experiment, we delete proximal modules (namely, the convolutional units in Eqs. (9)&(12d)) to verify the necessity of deep priors. Table 4 shows that deleting proximal modules make all metrics dramatically get worse. Therefore, the deep prior plays a significant role in our network.

(II) In the second experiment, we share the weights of all layers. In other words, the network contains only an MS Block and a PAN block, and the network is repeatedly fed with the current HRMS image K times. The results in Table 4 demonstrate that sharing the weights will weaken our network's performance.

(III) As illustrated in Section 3.1, the original problem Eq. (2) is split into an LRMS-aware subproblem and a PAN-aware subproblem. Now, to verify the rationality, we generalize Eq. (2) as a neural network with the same techniques for GPPNN. We exploit this block to build a neural network corresponding to Eq. (2). From Table 4, we learn that the network for Eq. (2) is worse than GPPNN. It is necessary to



(a) LRMS (b) PAN (c) GT(PSNR) (d) BDSD(21.86) (e) GS(21.43) (f) MIPSM(22.11)

(g) DRPNN(26.41) (h) MSDCNN(26.90) (i) RSIFNN(24.74) (j) PANNET(26.72) (k) MHNet(25.17) (l) GPPNN(28.70)

Figure 5. Visual inspection on GaoFen2 dataset. The caption of each subimage displays the corresponding PSNR value.

Table 4. The results of ablation experiments on the Landsat8 dataset.

Configurations	Proximal Module	Sharing Weights	Block for Eq. (2)	Transpose-ment	PSNR↑	SSIM↑	SAM↓	ERGAS↓
I	×	×	×	×	37.0404	0.9498	0.0180	1.4246
II	✓	✓	×	×	38.1650	0.9669	0.0164	1.2943
III	✓	×	✓	×	38.3213	0.9682	0.0155	1.3215
IV	✓	×	×	✓	38.5487	0.9700	0.0150	1.2746
GPPNN	✓	×	×	×	**38.9939**	**0.9727**	**0.0138**	**1.2483**

consider two deep priors to separately account for the generative models of LRMS and PAN images.

(IV) In the last experiment, the convolutional kernel in Eq. (8)/(12c) is replaced by the kernel in Eq. (6)/(12a) with the rotation of 180° to force them to satisfy the transposing requirement. It is found that, if the two kernels transpose to each other, the metrics will slightly become worse. The reason may be that the model with transposed kernels has fewer degrees of freedom weakening network's performance.

5. Conclusion and Future Work

This paper provides a new paradigm combining deep unrolling and observation models of pan-sharpening. We develop a model-driven pan-sharpening network, GPPNN, by alternatively stacking MS and PAN blocks whose designs are inspired by two optimization problems. Experiments on three satellites show that our network outperforms SOTA methods. And a series of ablation experiments verify the rationality of our network structure.

Remark that each satellite has its unique imaging parameters, including $\boldsymbol{D}$, $\boldsymbol{K}$ and $\boldsymbol{S}$. GPPNN trained on a satellite cannot be generalized to another satellite. Hence, the future work is how to improve the generalization of GPPNN.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102201, and in part by the National Natural Science Foundation of China under Grants 61976174, 61877049.

References

[1] L. Alparone, B. Aiazzi, S. Baronti, and A. Garzelli. Sharpening of very high resolution images with spectral distortion minimization. In *IEEE International Geoscience and Remote Sensing Symposium.*, volume 1, pages 458–460 vol.1, 2003.

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional net-

works. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2):295–307, 2016.

[3] A. Garzelli, F. Nencini, and L. Capobianco. Optimal mmse pan sharpening of very high resolution multispectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 46(1):228–236, 2008.

[4] P. Ghamisi, B. Rasti, N. Yokoya, Q. Wang, B. Hofle, L. Bruzzone, F. Bovolo, M. Chi, K. Anders, R. Gloaguen, P. M. Atkinson, and J. A. Benediktsson. Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 7(1):6–39, 2019.

[5] Alan R Gillespie, Anne B Kahle, and Richard E Walker. Color enhancement of highly correlated images. ii. channel ratio and "chromaticity" transformation techniques. *Remote Sensing of Environment*, 22(3):343 – 365, 1987.

[6] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *ICML, June 21-24, Haifa, Israel*, pages 399–406, 2010.

[7] R. Haydn, G. W. Dalke, J. Henkel, and J. E. Bare. Application of the ihs color transform to the processing of multisensor data and image enhancement. In *Proc. Int. Symp. Remote Sensing Arid and Semi-Arid Lands*, page 599–616, 1982.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR, Las Vegas, NV, USA, June 27-30*, pages 770–778, 2016.

[9] L. He, Y. Rao, J. Li, J. Chanussot, A. Plaza, J. Zhu, and B. Li. Pansharpening via detail injection based convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(4):1188–1204, 2019.

[10] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR, Honolulu, HI, USA, July 21-26*, pages 2261–2269, 2017.

[11] M. M. Khan, J. Chanussot, L. Condat, and A. Montanvert. Indusion: Fusion of multispectral and panchromatic images using the induction scaling technique. *IEEE Geoscience and Remote Sensing Letters*, 5(1):98–102, 2008.

[12] C. A. Laben and B. V. Brower. Process for enhancing the spatial resolution of multispectral imagery using pansharpening. (U.S. Patent 6011875A), January 2000.

[13] Yuelong Li, Mohammad Tofighi, Junyi Geng, Vishal Monga, and Yonina C. Eldar. Efficient and interpretable deep blind image deblurring via algorithm unrolling. *IEEE Trans. Computational Imaging*, 6:666–681, 2020.

[14] Junmin Liu, Yunqiao Feng, Changsheng Zhou, and Chunxia Zhang. Pwnet: An adaptive weight network for the fusion of panchromatic and multispectral images. *Remote. Sens.*, 12(17):2804, 2020.

[15] J. G. Liu. Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details. *International Journal of Remote Sensing*, 21(18):3461–3472, 2000.

[16] Lu Liu, Jun Wang, Erlei Zhang, Bin Li, Xuan Zhu, Yongqin Zhang, and Jinye Peng. Shallow-deep convolutional network and spectral-discrimination-based detail injection for multispectral imagery pan-sharpening. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 13:1772–1783, 2020.

[17] P. Liu, L. Xiao, and T. Li. A variational pan-sharpening method based on spatial fractional-order geometry and spectral–spatial low-rank priors. *IEEE Transactions on Geoscience and Remote Sensing*, 56(3):1788–1802, 2018.

[18] Xiangyu Liu, Qingjie Liu, and Yunhong Wang. Remote sensing image fusion based on two-stream fusion network. *Information Fusion*, 55:1 – 15, 2020.

[19] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Alan Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152:166 – 177, 2019.

[20] Giuseppe Masi, Davide Cozzolino, Luisa Verdoliva, and Giuseppe Scarpa. Pansharpening by convolutional neural networks. *Remote. Sens.*, 8(7):594, 2016.

[21] X. Meng, Y. Xiong, F. Shao, H. Shen, W. Sun, G. Yang, Q. Yuan, r. Fu, and H. Zhang. A large-scale benchmark data set for evaluating pansharpening performance: Overview and implementation. *IEEE Geoscience and Remote Sensing Magazine*, pages 0–0, 2020.

[22] Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *CoRR*, abs/1912.10557, 2019.

[23] Xavier Otazu, María González-Audícana, Octavi Fors, and Jorge Núñez. Introduction of sensor spectral response into image fusion methods. application to wavelet-based methods. *IEEE Trans. Geosci. Remote. Sens.*, 43(10):2376–2385, 2005.

[24] F. Ozcelik, U. Alganci, E. Sertel, and G. Unal. Rethinking cnn-based pansharpening: Guided colorization of panchromatic images via gans. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–16, 2020.

[25] R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[26] R. A. Schowengerdt. Reconstruction of multispatial multispectral image data using spatial frequency content. *Photogramm Eng Remote Sens*, 46(10):1325–1334, 1980.

[27] Z. Shao and J. Cai. Remote sensing image fusion with deep convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(5):1656–1669, 2018.

[28] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. In *CVPR, Salt Lake City, UT, USA, June 18-22*, pages 9446–9454, 2018.

[29] G. Vivone, L. Alparone, J. Chanussot, M. Dalla Mura, A. Garzelli, G. A. Licciardi, R. Restaino, and L. Wald. A critical comparison among pansharpening algorithms. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2565–2586, 2015.

[30] Lucien Wald, Thierry Ranchin, and Marc Mangolini. Fusion of satellite images of different spatial resolution: Assessing the quality of resulting images. *Photogramm. Eng. Remote Sensing*, pages 691–699, 1997.

[31] Dong Wang, Ying Li, Li Ma, Zongwen Bai, and Jonathan Cheung-Wai Chan. Going deeper with densely connected convolutional neural networks for multispectral pansharpening. *Remote. Sens.*, 11(22):2608, 2019.

[32] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR, Seattle, WA, USA, June 13-19*, pages 3100–3109, 2020.

[33] Na Wang and Jian Sun. Model meets deep learning in image inverse problems. *CSIAM Transactions on Applied Mathematics*, 1(3):365–386, 2020.

[34] Yancong Wei, Qiangqiang Yuan, Huanfeng Shen, and Liangpei Zhang. Boosting the accuracy of multispectral image pansharpening by learning a deep residual network. *IEEE Geosci. Remote. Sens. Lett.*, 14(10):1795–1799, 2017.

[35] Qi Xie, Minghao Zhou, Qian Zhao, Deyu Meng, Wangmeng Zuo, and Zongben Xu. Multispectral and hyperspectral image fusion by MS/HS fusion net. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1585–1594, 2019.

[36] Shuang Xu, Ouafa Amira, Junmin Liu, Chun-Xia Zhang, Jiangshe Zhang, and Guanghai Li. HAM-MFN: hyperspectral and multispectral image multiscale fusion network with RAP loss. *IEEE Trans. Geosci. Remote. Sens.*, 58(7):4618–4628, 2020.

[37] Dong Yang and Jian Sun. Proximal dehaze-net: A prior learning-based deep network for single image dehazing. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 729–746, 2018.

[38] Junfeng Yang, Xueyang Fu, Yuwen Hu, Yue Huang, Xinghao Ding, and John W. Paisley. Pannet: A deep network architecture for pan-sharpening. In *ICCV, Venice, Italy, October 22-29*, pages 1753–1761, 2017.

[39] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Admm-csnet: A deep learning approach for image compressive sensing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(3):521–538, 2020.

[40] Q. Yuan, Y. Wei, X. Meng, H. Shen, and L. Zhang. A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3):978–989, 2018.

[41] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In *CVPR, Honolulu, HI, USA, July 21-26*, pages 2808–2817, 2017.