PanFlowNet: A Flow-Based Deep Network for Pan-sharpening

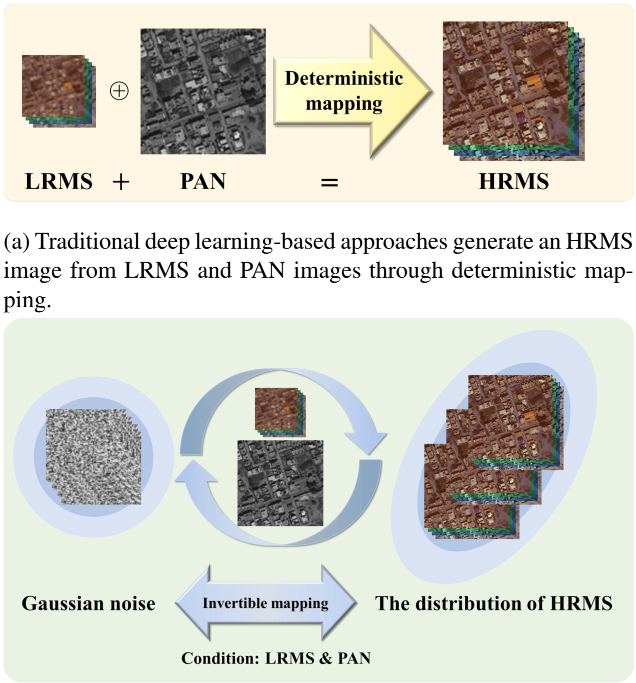
Gang Yang[1], Xiangyong Cao[2]*, Wenzhe Xiao[2], Man Zhou[3], Aiping Liu[1]*; Xun Chen[1], Deyu Meng[2,4]
[1] University of Science and Technology of China;
[2] Xi'an Jiaotong University
[3] Nanyang Technological University
[4] Macao Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macao

Abstract

*Pan-sharpening aims to generate a high-resolution multispectral (HRMS) image by integrating the spectral information of a low-resolution multispectral (LRMS) image with the texture details of a high-resolution panchromatic (PAN) image. It essentially inherits the ill-posed nature of the super-resolution (SR) task that diverse HRMS images can degrade into an LRMS image. However, existing deep learning-based methods recover only one HRMS image from the LRMS image and PAN image using a deterministic mapping, thus ignoring the diversity of the HRMS image. In this paper, to alleviate this ill-posed issue, we propose a flow-based pan-sharpening network (**PanFlowNet**) to directly learn the **conditional distribution** of HRMS image given LRMS image and PAN image instead of learning a deterministic mapping. Specifically, we first transform this unknown conditional distribution into a given Gaussian distribution by an invertible network, and the conditional distribution can thus be explicitly defined. Then, we design an invertible Conditional Affine Coupling Block (CACB) and further build the architecture of PanFlowNet by stacking a series of CACBs. Finally, the PanFlowNet is trained by maximizing the log-likelihood of the conditional distribution given a training set and can then be used to predict diverse HRMS images. The experimental results verify that the proposed PanFlowNet can generate various HRMS images given an LRMS image and a PAN image. Additionally, the experimental results on different kinds of satellite datasets also demonstrate the superiority of our PanFlowNet compared with other state-of-the-art methods both visually and quantitatively. Code is available at Github.*

1. Introduction

With the rapid development of satellite sensors, remote sensing images have become widely used in various applications, such as environmental monitoring [4], classification [8], and target detection [17, 56]. Satellites capture multispectral (MS) and panchromatic (PAN) images simultaneously with complementary information for each modality that PAN images have a high spatial solution [20] and MS images contain rich spectral information [42]. MS sensors reduce the spatial resolution while ensuring spectral richness for MS images [55]. To obtain an MS image with both high spectral and spatial resolution, the pan-sharpening technique that aims to fuse the MS and PAN images has attracted a large amount of attention.



(a) Traditional deep learning-based approaches generate an HRMS image from LRMS and PAN images through deterministic mapping.

(b) Our proposed PanFlowNet can learn the conditional distribution of HRMS images, and thus it can generate diverse HRMSs from LRMS and PAN images as well as noise.

Figure 1: Comparison between traditional deep learning-based methods and our proposed PanFlowNet.

[*] corresponding author

The past decades have witnessed the explosive growth of research works in the pan-sharpening field. In terms of the quality of the generated fusion results, the focuses have been mainly on model-based [42, 44, 35] and deep learning (DL)-based [13, 19, 45, 47, 52, 58, 59] methods. Model-based methods usually optimize a mathematical model that preserves spectral and spatial information, and most of them follow the assumption that the PAN image (or its gradient) can be modelled as a linear combination among all bands (or their gradients) of high-resolution multispectral (HRMS) images. However, they are highly dependent on the assumptions about the relationship between HRMS and PAN images [3]. Unfortunately, previous work did not accurately establish this relationship, which limits the further improvement of pan-sharpening. Besides, the model-based methods are challenging in optimization, limiting their practical applications.

In the era of deep learning, convolutional neural networks (CNN) have emerged as a significant tool for pan-sharpening. CNN-based methods train the network by minimizing the distance between the fused result and the HRMS reference image. Because of the strong nonlinear fitting ability of neural networks, this kind of method always achieves excellent performance. However, pan-sharpening is essentially an ill-posed problem since a given LRMS image can be degraded from infinitely many compatible HRMS images. This poses severe challenges when designing DL-based pan-sharpening approaches. Although existing CNN-based methods can obtain excellent results, they only learn a deterministic mapping from LRMS and PAN images to HRMS images, as shown in Fig. 1a, and thus the ill-posed issue is not well addressed.

To solve the above issues and generate more diverse realistic images, in this paper, we propose a novel neural architecture for pan-sharpening, called **PanFlowNet**, which directly learns the conditional distribution of the HRMS image given the input LRMS image and PAN image instead of learning a deterministic mapping. Specifically, we first transform a sample of the unknown conditional distribution into a sample of a given Gaussian distribution using an invertible network. Thus the ***conditional distribution*** can be explicitly defined by the product of the Gaussian distribution and the determinant of the Jacobian matrix. To build the invertible network, we first design an invertible Conditional Affine Coupling Block (CACB) and then stack a series of CACBs to construct the network architecture of PanFlowNet. Finally, our PanFlowNet can be trained by minimizing the negative log-likelihood of the conditional distribution on a training set. Once the training is finished, we can generate diverse HRMSs by inputting the LRMS and PAN images as well as different noise samples of the given Gaussian distribution into the PanFlowNet, as shown in Fig. 1b.

In summary, the contributions of our work are as follows:

- We propose a flow-based deep network (i.e., PanFlowNet) for pan-sharpening. This network can accurately learn the conditional distribution of HRMS images given the corresponding LRMS and PAN images. To the best of our knowledge, this is the first attempt to learn an explicit distribution by employing the generative flow model for the pan-sharpening task.

- The proposed PanFlowNet can generate diverse HRMSs given the LRMS and PAN images as well as the Gaussian noise sample and thus can alleviate the ill-posed issue to some extent. Besides, the generated HRMS images are diverse since each HRMS image focuses on a different detailed part of the ground truth.

- We extend the vanilla flow model to a probabilistic multi-conditional flow model to adapt to the multi-conditionality of the pan-sharpening task. Extensive experiments over different satellite datasets demonstrate that our method can outperform existing state-of-the-art approaches both visually and quantitatively.

2. Related work

2.1. Classic pan-sharpening methods

The traditional methods of pan-sharpening can be classified into three main categories: component substitution (CS)- [9, 40, 21], multi-resolution analysis (MRA)- [36, 44, 39, 33], and variational optimization (VO)- [18, 43, 12, 3, 10] based methods. The common CS methods [9, 40, 21] project the original MS image into a transform domain and then replace the separated spatial components with PAN images. The typical MRA methods [39, 33] inject the spatial details extracted by the multiresolution decomposition techniques from PAN images into the up-sampled MS images. The VO methods [3, 10] are concerned because of the fine fusion effects on pan-sharpening. In addition, some hybrid methods take advantage of multiple methods to complement each other [57, 27]. Most model-based methods assume that the PAN image (or its gradient) can be modelled as a linear combination among all bands (or their gradients) of the HRMS image. However, this reduces the intensity fidelity of the HRMS image since various sensors mounted on satellites have extremely diverse response characteristics to objects [55].

2.2. Deep learning based pan-sharpening methods

Due to the highly nonlinear fitting capacity of the convolutional neural network, PNN [34] models the relationship between PAN, LRMS, and HRMS images using three convolutional layers, achieving a significant improvement compared with other classical methods. Inspired by PNN, a large number of CNN-based pan-sharpening studies [7, 50, 49] have emerged recently. For instance, PANNet [52] utilizes ResNet's residual learning module, MSDCNN [54] adds multi-scale modules based on residual connection, SRPPNN [5] refers to the design idea of SRCNN [16], and Wang *et al*. [46] adopted U-shaped network. Moreover, WSDFNet [24] propagates shallow features scaled by adaptive skip weightier, and Ma *et al*. [32] proposes an unsupervised framework based on GAN. Additionally, some model-driven CNN models with clear physical meaning emerge, such as MHNet [48], Proximal PanNet [6], PanCSC-Net [7], MADUN [60], GPPNN [51]. Although all these DL-based pansharpening approaches achieve excellent performance, they all only learn a deterministic mapping from the LRMS and PAN image to the HRMS image, thus ignoring the ill-posed issue of the pansharpening task.

2.3. Flow-based methods

Flow-based generative models have shown an excellent ability to explicitly learn the probability density function of data. A sequence of invertible transformations generally constructs them to map a base distribution to a complex one [37, 23, 38, 26, 53]. Several unconditional generative flow models have emerged that extend the early flow models to multiscale architectures with split couplings that allow for efficient inference and sampling. For example, Dinh *et al*. [14] proposes to stack non-linear additive coupling and other transformation layers as the flow model NICE. Inspired by NICE, Dinh *et al*. [15] propose RealNVP, which upgrades additive coupling to affine coupling without loss of invertibility and achieves better performance. After that, Kingma *et al*. [25] propose 1×1 convolution to replace the fixed permutation layer in RealNVP and succeed in synthesizing realistic-looking images. Likewise, various conditional flow models have appeared aiming at conditional image synthesis [41, 2]. Lugmayr *et al*. proposed SRFlow [31] to generate diverse high-resolution images conditioned on low-resolution ones. Abdal *et al*. [1] sampled latent vectors based on given attributes and fed the vectors to the StyleGANgenerator to synthesize high-quality images. Compared with the aforementioned methods, especially SRFlow [31], our approach has several differences. Firstly, our approach is specifically proposed for the pan-sharpening task. Secondly, unlike the flow models for traditional image inverse problems that have no extra guided image information, e.g., SRFlow, the flow model of pansharpening requires embedding the guided PAN image information and thus poses an issue of how to inject the detailed texture information of PAN image into the network.

3. Proposed method

In this section, we first introduce our proposed probabilistic flow model for pan-sharpening in detail. Then, we design a network to implement this model.

3.1. Probabilistic flow model for pan-sharpening

The goal of pan-sharpening is to recover the high-resolution multispectral (HRMS) image $\mathbf{H} \in \mathbb{R}^{H\times W\times B}$ from a given low-resolution multispectral (LRMS) image $\mathbf{L} \in \mathbb{R}^{h\times w\times B}$ under the guidance of a high resolution panchromatic (PAN) image $\mathbf{P}$, where $h = H/s, w = W/s$, and s is a resolution factor.

As aforementioned, the pan-sharpening task is essentially an ill-posed problem since the LRMS image may be degraded from an infinite amount of HRMSs. However, most current deep learning-based pan-sharpening approaches learn a deterministic mapping $f : (\mathbf{L}, \mathbf{P}) \longmapsto \mathbf{H}$, which receives the LRMS image $\mathbf{L}$ and the PAN image $\mathbf{P}$ as input and outputs only one possible HRMS image $\mathbf{H}$. To alleviate the ill-posed issue, this work aims to learn the conditional distribution of the HRMS image $\mathbf{H}$, i.e., $P_{\mathbf{H}|\mathbf{L},\mathbf{P}}(\mathbf{H}|\mathbf{L}, \mathbf{P}; \boldsymbol{\theta})$, given the LRMS image $\mathbf{L}$ and the PAN image $\mathbf{P}$, which is a more difficult task since the conditional distribution model can generate infinite possible HRMS images, instead of just predicting a single HRMS image output by a deterministic mapping. Next, we will propose a probabilistic flow method to learn the conditional distribution $P_{\mathbf{H}|\mathbf{L},\mathbf{P}}(\mathbf{H}|\mathbf{L}, \mathbf{P}; \boldsymbol{\theta})$ given an LRMS-PAN-HRMS training set $\mathcal{D} = \{(\mathbf{L}_j, \mathbf{P}_j, \mathbf{H}_j)\}_{j=1}^{m}$.

Since the conditional distribution $P_{\mathbf{H}|\mathbf{L},\mathbf{P}}(\mathbf{H}|\mathbf{L}, \mathbf{P}; \boldsymbol{\theta})$ is unknown, we thus resort to the probabilistic flow model, which uses an invertible function $f_{\boldsymbol{\theta}}(\cdot)$ to parametrize the conditional distribution. In this conditional setting, $f_{\boldsymbol{\theta}}(\cdot)$ can map a LRMS-PAN-HRMS image pair to a latent variable $\mathbf{z}$, namely

$$\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{H}; \mathbf{L}, \mathbf{P}), \tag{1}$$

where $f_{\boldsymbol{\theta}}(\mathbf{H}; \mathbf{L}, \mathbf{P})$ is required to be invertible for the first argument $\mathbf{H}$ given the LRMS image $\mathbf{L}$ and the PAN image $\mathbf{P}$. Therefore, the HRMS image $\mathbf{H}$ can then be exactly obtained from the latent variable $\mathbf{z}$ as

$$\mathbf{H} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{z}; \mathbf{L}, \mathbf{P}),\ \ \mathbf{z} \sim P_{\mathbf{z}}(\cdot), \tag{2}$$

where $\mathbf{z}$ is a sample of the latent variable distribution $P_{\mathbf{z}}(\cdot)$. For simplicity, the latent variable $\mathbf{z}$ is always assumed to be a simple Gaussian distribution $\mathbf{z} \sim P_{\mathbf{z}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}_s)$, where $\mathbf{I}_s$ is the identity matrix, s is the dimension of $\mathbf{z}$, and s equals to $H \times W \times B$ in order to guarantee $f_{\boldsymbol{\theta}}$ to be invertible. In this setting, the probability density function $P_{\mathbf{H}|\mathbf{L},\mathbf{P}}(\mathbf{H}|\mathbf{L}, \mathbf{P}; \boldsymbol{\theta})$ can then be accurately defined by using the change-of-variables formula, namely

$$P_{\mathbf{H}|\mathbf{L},\mathbf{P}}(\mathbf{H}|\mathbf{L},\mathbf{P};\boldsymbol{\theta}) = P_{\mathbf{z}}(f_{\boldsymbol{\theta}}(\mathbf{H};\mathbf{L},\mathbf{P})) \left|\det\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{H};\mathbf{L},\mathbf{P})}{\partial \mathbf{H}}\right|, \tag{3}$$

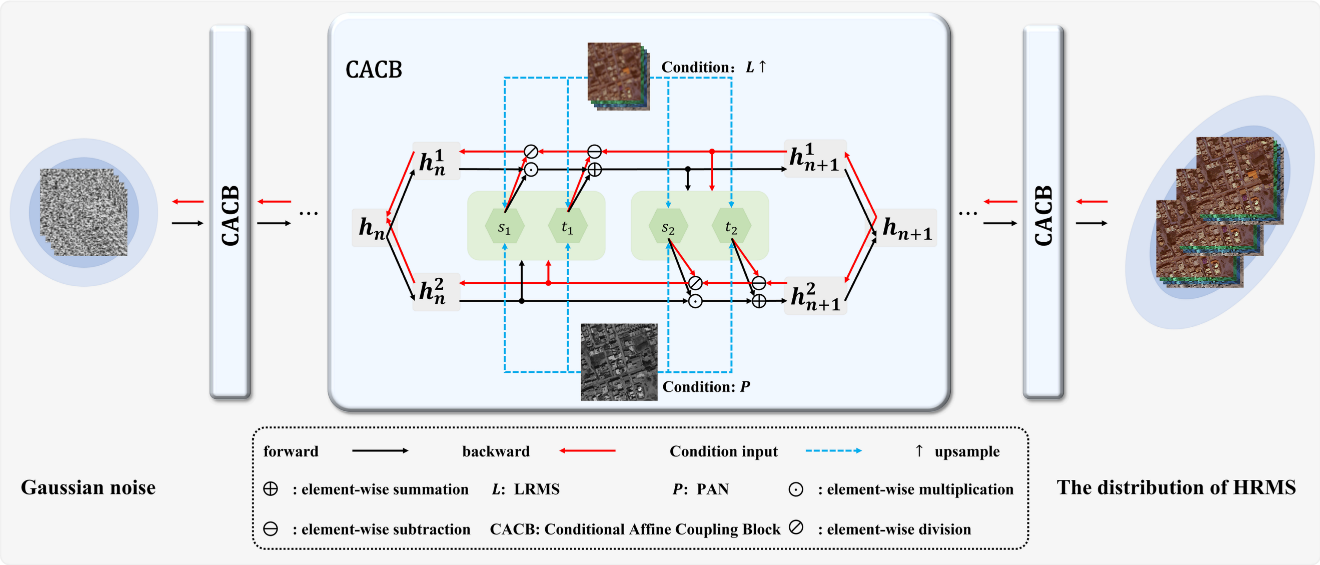


Figure 2: The network architecture of our PanFlowNet consists of a series of invertible Conditional Affine Coupling Blocks (CACBs). The PanFlowNet can directly learn the distribution of HRMS images from Gaussian noise conditioned on LRMS and PAN images.

where $\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{H};\mathbf{L},\mathbf{P})}{\partial \mathbf{H}}$ is the Jacobian matrix of function $f_{\boldsymbol{\theta}}(\cdot)$ at $\mathbf{H}$ and $\det(\cdot)$ is the determinant function. In this work, we choose $f_{\boldsymbol{\theta}}(\cdot)$ such that its determinant of the Jacobian is easily computed. Specifically, we utilize an inverse neural network (INN) to implement $f_{\boldsymbol{\theta}}(\cdot)$, and the detailed design of the INN is presented in the next section. Based on Eq. (3), we can learn the parameter of $f_{\boldsymbol{\theta}}$ by minimizing the negative log-likelihood (NLL) of training pair $(\mathbf{L}, \mathbf{P}, \mathbf{H})$ as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta};\mathbf{L},\mathbf{P},\mathbf{H}) &= -\log P_{\mathbf{H}|\mathbf{L},\mathbf{P}}(\mathbf{H}|\mathbf{L},\mathbf{P};\boldsymbol{\theta})\\ &= -\log P_{\mathbf{z}}(f_{\boldsymbol{\theta}}(\mathbf{H};\mathbf{L},\mathbf{P})) - \log\left|\det\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{H};\mathbf{L},\mathbf{P})}{\partial \mathbf{H}}\right|.\end{aligned} \tag{4}$$

Further, to guarantee the second term in Eq. (4) tractable, we decompose $f_{\boldsymbol{\theta}}$ into a sequence of N invertible layers, i.e., $f_{\boldsymbol{\theta}} = f_{\boldsymbol{\theta}}^N f_{\boldsymbol{\theta}}^{N-1}\cdots f_{\boldsymbol{\theta}}^1$, where $f_{\boldsymbol{\theta}}^n$ is the n_{th} invertible layer, which receives feature $\mathbf{h}_n$ of the preivous layer as input and generates $\mathbf{h}_{n+1}$ as output, i.e., $\mathbf{h}^{n+1} = f_{\boldsymbol{\theta}}^n(\mathbf{h}^n;\mathbf{L},\mathbf{P})$. Based on Eq. (1), we can easily know that $\mathbf{h}^1 = \mathbf{H}$ and $\mathbf{h}^{N+1} = \mathbf{z}$. Additionally, we encode the information of LRMS image $\mathbf{L}$ and PAN image $\mathbf{P}$ into each invertible layer $f_{\boldsymbol{\theta}}^n$ by regarding them as conditional input, which can compensate for detail and structure information to each layer.

By utilizing the chain rule and the multiplicative property of the determinant, we can compute the NLL objective in Eq. (4) as follows:

$$\begin{aligned}&\mathcal{L}(\boldsymbol{\theta};\mathbf{L},\mathbf{P},\mathbf{H})\\ &= -\log P_{\mathbf{z}}(f_{\boldsymbol{\theta}}(\mathbf{H};\mathbf{L},\mathbf{P})) - \sum_{n=0}^{N-1}\log\left|\det\frac{\partial f_{\boldsymbol{\theta}}^n(\mathbf{h}^n;\mathbf{L},\mathbf{P})}{\partial \mathbf{h}^n}\right|.\end{aligned} \tag{5}$$

Therefore, the final objective function on the training set $\mathcal{D}$ is defined as $\mathcal{L}_{final}(\boldsymbol{\theta}) = \sum_{j=1}^m \mathcal{L}(\boldsymbol{\theta};\mathbf{L}_j,\mathbf{P}_j,\mathbf{H}_j)$. To ensure each layer invertible and fast computation of the log-determinant of the Jacobian $\frac{\partial f_{\boldsymbol{\theta}}^n}{\partial \mathbf{h}^n}$, we need to carefully design the network architecture of each layer. This will be discussed in the next section.

Once the optimal parameter θ^* of the invertible network f_θ is learned, we can sample an HRMS from $P_{\mathbf{H}|\mathbf{L},\mathbf{P}}(\mathbf{H}|\mathbf{L},\mathbf{P};\boldsymbol{\theta}_*)$ as follows:

$$\mathbf{H} = f_{\boldsymbol{\theta}_*}^{-1}(\mathbf{z};\mathbf{L},\mathbf{P}),\ \ \mathbf{z}\sim\mathcal{N}(\mathbf{z}|\mathbf{0},\mathbf{I}_s). \tag{6}$$

Since we can sample infinite HRMS images, a vital issue is posed that which sample we should choose in practical application. In this paper, we propose a ***maximum probability criterion***, i.e., the HRMS image corresponding to the maximum probability of $P_{\mathbf{H}|\mathbf{L},\mathbf{P}}(\mathbf{H}|\mathbf{L},\mathbf{P};\boldsymbol{\theta}_*)$ calculated by Eq.(2) is selected. The proposed criterion is used in all the experiments.

3.2. Network architecture

3.2.1 Overall network architecture

The model must span a variety of possible HRMS images instead of just predicting a single HRMS output. Our intention is to learn the parameters $\boldsymbol{\theta}$ of the distribution in a data-driven manner, given a training set.

In this section, we follow the design of the Probabilistic Flow Model to produce an invertible neural network (INN) with Conditional Affine Coupling Blocks (CACBs). Specifically, we build **PanFlowNet** by stacking a series of invertible layers. As shown in Fig. 2, PanFlowNet consists of several flow blocks, and each flow block is composed of a reversible CACB. A CACB corresponds to one step of the transformation process $f_{\boldsymbol{\theta}}^n$. For each flow layer, the transformation process is considered as follows.

Let $\mathbf{h}_n$ be a latent feature variable in a series of invertible transformations. The goal of flow layer $f_{\boldsymbol{\theta}}^n$ is to generate $\mathbf{h}_{n+1}$ with the guidance of LRMS and PAN images as

$$\mathbf{h}_{n+1} \;=\; f_{\boldsymbol{\theta}}^n(\mathbf{h}_n; \mathbf{L}, \mathbf{P}). \tag{7}$$

The inverse process of flow layer $g_{\boldsymbol{\theta}}^n$ takes $\mathbf{h}_{n+1}$ as input and uses LRMS and PAN as conditions to generate $\mathbf{h}_n$, and the process can be expressed as follows:

$$\mathbf{h}_n \;=\; g_{\boldsymbol{\theta}}^n(\mathbf{h}_{n+1}; \mathbf{L}, \mathbf{P}). \tag{8}$$

We follow the design of Eq. 7 and Eq. 8 to provide a CACB as $f_{\boldsymbol{\theta}}^n$ and the reverse as $g_{\boldsymbol{\theta}}^n$. In the next subsection, we will present the structural design of the CACB.

3.2.2 Conditional affine coupling block

The network architecture of the invertible layer requires careful design in order to ensure well-conditioned reversibility and a tractable Jacobian determinant. This challenge was first addressed in [14, 15] and has recently inspired significant interest. Our method is an extension of the affine coupling block architecture established in [15]. As shown in Fig. 2, each flow block is a reversible block consisting of two complementary affine coupling layers, which splits its input $\mathbf{h}_n$ into two parts, i.e., $\mathbf{h}_n$=[$\mathbf{h}_n^1$, $\mathbf{h}_n^2$], and applies affine transformations with coefficients $exp(s_i)$ and $t_i, i = 1, 2$ to them. Specifically, the affine transformation is defined as follows:

$$\mathbf{h}_{n+1}^1 = \mathbf{h}_n^1 \odot \exp\left(s_1\left(\mathbf{h}_n^2\right)\right) + t_1\left(\mathbf{h}_n^2\right), \tag{9}$$
$$\mathbf{h}_{n+1}^2 = \mathbf{h}_n^2 \odot \exp\left(s_2\left(\mathbf{h}_{n+1}^1\right)\right) + t_2\left(\mathbf{h}_{n+1}^1\right), \tag{10}$$

where $\odot$ is the element-wise multiplication. This affine transformation has a triangular Jacobian matrix, and thus its determinant is easy to calculate. Additionally, the output $[\mathbf{h}_{n+1}^1, \mathbf{h}_{n+1}^2]$ are concatenated again and then passed to the next coupling block. The internal functions $s_i(\cdot)$ and $t_i(\cdot)$ can be represented by arbitrary neural networks and are only evaluated in the forward direction. This affine transformation in Eq. (9) and Eq. (10) are easily to invertible, namely

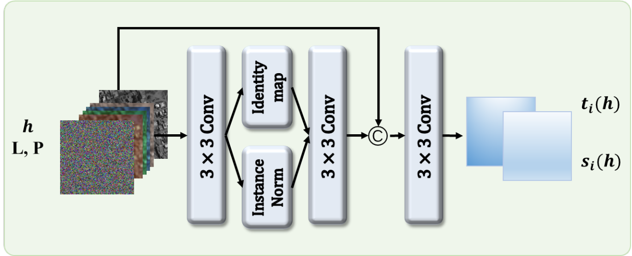


Figure 3: HIN Block is used to implement $s_i(\cdot)$ and $t_i(\cdot)$

$$\mathbf{h}_n^2 \;=\; \left(\mathbf{h}_{n+1}^2 - t_2\left(\mathbf{h}_{n+1}^1\right)\right) \oslash \exp\left(s_2\left(\mathbf{h}_{n+1}^1\right)\right), \tag{11}$$
$$\mathbf{h}_n^1 \;=\; \left(\mathbf{h}_{n+1}^1 - t_1\left(\mathbf{h}_n^2\right)\right) \oslash \exp\left(s_1\left(\mathbf{h}_n^2\right)\right), \tag{12}$$

where $\oslash$ is the element-wise division. As shown in [15], the logarithm of the Jacobian determinant for such a coupling block is simply the sum of $s_1(\cdot)$ and $s_2(\cdot)$ over image dimensions. In the conditional setting that $\mathbf{L}$ and $\mathbf{P}$ are regarded as conditional input of $s_i(\cdot)$ and $t_i(\cdot)$, $i = 1, 2$, the affine transformation is still invertible since its invertibility is not related with the sub-networks $s_j(\cdot)$ and $t_j(\cdot)$. Thus, we generate the input for $s(\cdot)$ and $t(\cdot)$ by concatenating the condition data $\mathbf{L}$ and $\mathbf{P}$ with the latent feature $\mathbf{h}$, which will not affect the invertibility of this affine transformation. Fig. 3 shows the conditional affine coupling layer, which is an ***extension*** of the affine coupling layer presented above. In our implementation, $s_i(\cdot)$ and $t_i(\cdot)$ in the $f_{\boldsymbol{\theta}}^n$ are implemented by HIN Block [11].

4. Experiments

4.1. Datasets and evaluation metrics

In this section, we conduct several experiments to verify the effectiveness of our proposed PanFlowNet on three satellite image datasets, i.e., WorldView II, WorldView III, and GaoFen2. For each dataset, we have hundreds of image pairs, and the MS images are cropped into patches with the size of 32 × 32, and the size of corresponding PAN images is 128 × 128. Each patch is normalized into 0 to 1.

Four assessment metrics are used to evaluate the performance, including peak signal-to-noise ratio (PSNR), Structural similarity (SSIM), Erreur Relative Globale Adimensionnelle de Synthese (ERGAS), and Spectral angle mapper (SAM). The first three metrics measure spatial distortion, and the fourth measures spectral distortion.

4.2. Implementation details

We implement our PanFlowNet in the PyTorch framework. As the paired training samples are not available, we construct the paired training datasets using the Wald protocol [45]. To increase training efficiency, we first pre-train

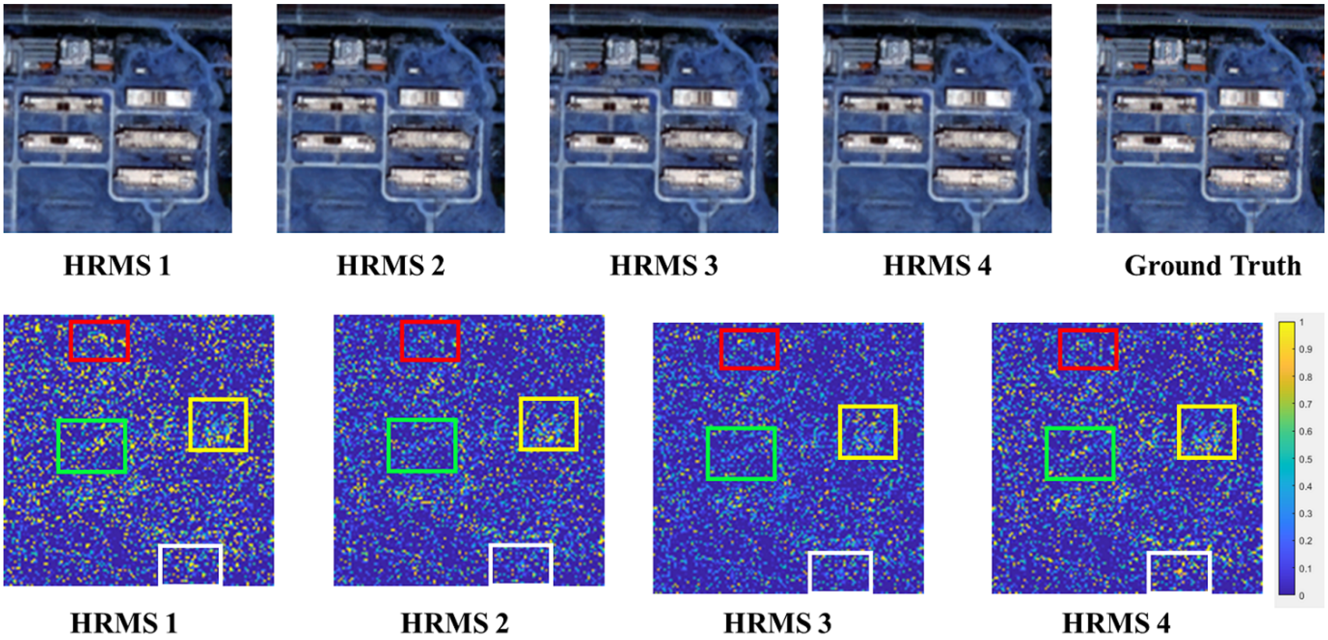


Figure 4: The visualization results are used to validate the effectiveness of our proposed PanFlowNet. The first row visualizes different HRMS images generated from different noises and gives LRMS and PAN images on the WorldView-II dataset. The second row visually shows the differences in the detailed parts that each HRMS image focuses on ground truth.

Table 1: Experimental results of all the competing methods on the three benchmark datasets. The best and the second best values are highlighted in **bold** and underline, respectively.

Methods	Params	WorldView II				WorldView III				GaoFen2			
		PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
SFIM	-	34.1297	0.8975	0.0439	2.3449	21.8212	0.5457	0.1208	8.9730	36.9060	0.8882	0.0318	1.7398
Brovey	-	35.8646	0.9216	0.0403	1.8238	22.5060	0.5466	0.1159	8.2331	37.7974	0.9026	0.0218	1.3720
GS	-	35.6376	0.9176	0.0423	1.8774	22.5608	0.5470	0.1217	8.2433	37.2260	0.9034	0.0309	1.6736
IHS	-	32.1601	**0.9812**	0.0461	2.0278	22.5579	0.5354	0.1266	8.3616	38.1754	0.9100	0.0243	1.5336
GFPCA	-	34.5581	0.9038	0.0488	2.1411	22.3344	0.4826	0.1294	8.3964	37.9443	0.9204	0.0314	1.5604
PNN	0.0689	40.7550	0.9624	0.0259	1.0646	29.9418	0.9121	0.0824	3.3206	43.1208	0.9704	0.0172	0.8528
PANNET	0.0688	40.8176	0.9626	0.0257	1.0557	29.6840	0.9072	0.0851	3.4263	43.0659	0.9685	0.0178	0.8577
MSDCNN	0.2390	41.3355	0.9664	0.0242	0.9940	30.3038	0.9184	0.0782	3.1884	45.6874	0.9827	0.0135	0.6389
SRPPNN	1.7114	41.4538	0.9679	0.0233	0.9899	30.4346	0.9202	0.0770	3.1553	47.1998	0.9877	0.0106	0.5586
GPPNN	0.1198	41.1622	0.9684	0.0244	1.0315	30.1785	0.9175	0.0776	3.2593	44.2145	0.9815	0.0137	0.7361
Ours	0.0873	**41.8584**	0.9712	**0.0224**	**0.9335**	**30.4873**	**0.9221**	**0.0751**	**3.1142**	**47.2533**	**0.9884**	**0.0103**	**0.5512**

Table 2: PSNR values of PanFlowNet with different noises.

Noise	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
noise 1	41.8561	0.971218	0.0223989	0.933770
noise 2	41.8581	0.971229	0.0223936	0.933516
noise 3	41.8579	0.971224	0.0223922	0.933545
noise 4	41.8563	0.971216	0.0223946	0.933642
noise 5	41.8583	0.971228	0.0223937	0.933529
noise 6	41.8552	0.971204	0.0224002	0.933823

our model using an L1 loss for 1000 epochs and then train the whole network using only the loss for 100 epochs. In the training stage, we employ ADAM optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ to update the network parameters for 1000 epochs with a batch size of 4. The learning rate is initialized with $5e-5$ and is decayed by multiplying 0.5 for every 200 epochs. In the inference stage, we randomly select a Gaussian noise. All the experiments are conducted on NVIDIA GeForce GTX 3080Ti GPU.

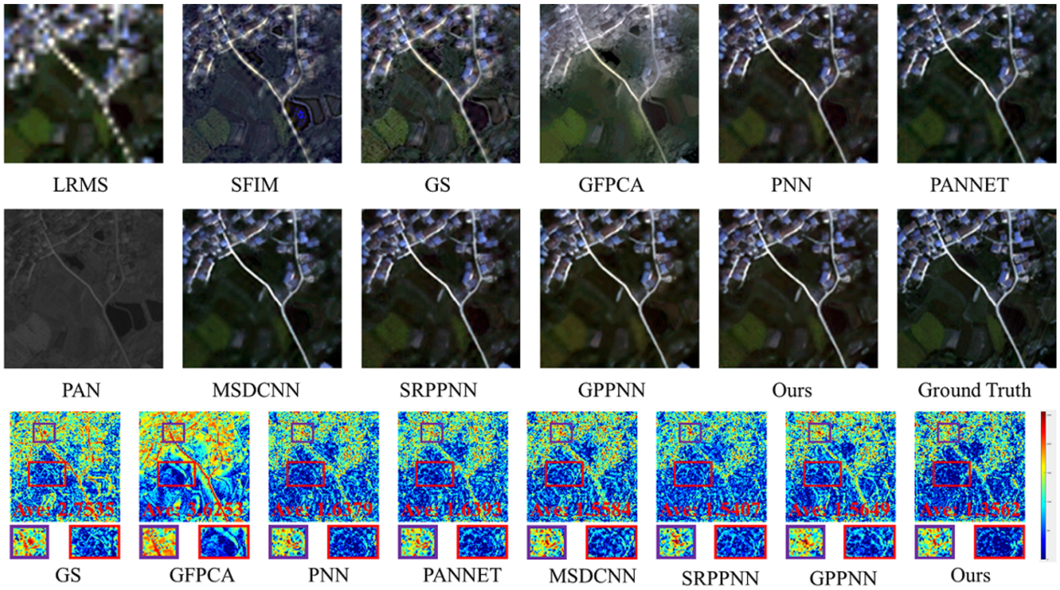


Figure 5: Visual comparison of all the competing methods on WorldViewII. The last row visualizes the error maps and average errors between the pan-sharpening results and the ground truth.

4.3. Effectiveness verification

To verify the effectiveness of the flow-based modelling methodology in our proposed PanFlowNet, we sample different HRMS images from the distribution of HRMS once the PanFlowNet has been trained. Specifically, we use different noise samples to generate the HRMS, and the quantitative experimental results are shown in Table 2. From Table 2, it can be seen that there exist some differences between the restored HRMS. These differences mainly attribute to the different noises. Additionally, we also present the qualitative results in Fig. 4, from which we can observe that the generated HRMS images from different noises are very similar, but there still exist some differences in their fine details, which indicates that the generated different HRMS images will focus on different detailed parts of the ground-truth.

4.4. Comparison with the state-of-the-arts

In this section, to verify the effectiveness of our proposed PanFlowNet, we compare PanFlowNet with ten competitive methods, including five classical methods (i.e., SFIM [30], Brovey [21], GS [28], IHS [22], and GFPCA [29]) and five DL-based methods (i.e., PNN [34], PANNET [52], MSDCNN [54], SRPPNN [5], and GPPNN [51]), which our method is conducted by randomly selected Gaussian noise for inference.

Parameter numbers vs. model performance. The comparison results between parameter number and model performance are shown in Table 1, from which it can be seen that our network is able to achieve a good trade-off and perform best with comparably fewer parameters compared to other deep learning-based methods.

Table 3: Non-reference metrics on full-resolution dataset.

	PNN	PANNET	MSDCNN	SRPPNN	GPPNN	Ours
$D_\lambda \downarrow$	0.0746	0.0737	0.734	0.0767	0.0782	**0.0665**
$D_s \downarrow$	0.1164	0.1224	0.1151	0.1162	0.1253	**0.1113**
QNR ↑	0.8191	0.8143	0.8251	0.8173	0.8073	**0.8257**

Evaluation on full-resolution scene. In order to compare the generalization of methods, we further perform experiments on an additional real-world full-resolution dataset of 200 samples obtained by the GaoFen2 satellite for evaluation. Due to the lack of ground-truth MS images in real-world full-resolution scenes, we measure the model's performance using commonly used three non-reference metrics: the spectral distortion index D_λ, the spatial distortion index D_s, and the quality without reference QNR. The quantitative comparisons between representative CNN-based methods and our method are shown in Table 3. The lower D_λ, D_s and the higher QNR correspond to the better image quality where the best results are remarked by red bold. From Table 3, our methods surpass other competitive Pan-sharpening methods in all the indexes, which shows its generalization ability.

Quantitative results. The comparison results of 10 benchmark methods over three satellite datasets are reported in Table 1, where the best and the second best values are highlighted in red bold and blue underline, respectively. As can

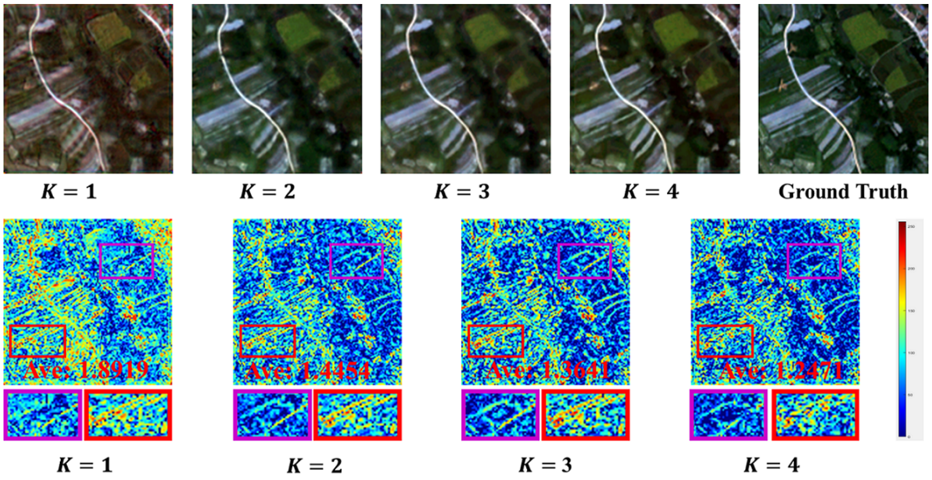


Figure 6: Intermediate visual results of different numbers of CACB in our PanFlowNet on WorldViewII. The last row visualizes the error maps and average errors between the pan-sharpening results and the ground truth.

Table 4: PSNR values of PanFlowNet with different number of stages on WorldViewII. The best and the second best values are highlighted in **bold** and underline, respectively.

Stages (K)	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
1	38.2469	0.9471	0.0344	1.4294
2	40.7152	0.9639	0.0255	0.9935
3	41.2664	0.9674	0.0236	0.9935
4	**41.8584**	**0.9712**	**0.0224**	**0.9335**

be seen clearly, our proposed method achieves the best overall results over other competing methods over all the satellite datasets. This confirms, to a certain extent, the effectiveness and flexibility of our method.

Qualitative results. We also show the visual results of one image from the WorldView II dataset in Fig. 5 to evaluate the effectiveness of our method. From the first two rows of Fig. 5, we can see that the visual result of our PanFlowNet is obviously better than other competing methods. To make the visual advantage clear, the images in the last row are the error maps and the average errors between the output pan-sharpened results and the ground truth. Compared with other competing methods, our PanFlowNet has the minimum spatial and spectral distortions. As for the error maps, it is noted that our proposed method has the smallest average error compared to other comparison methods while being the closest to ground truth. The state-of-the-art performance of our method demonstrates the effectiveness of the proposed PanFlowNet.

4.5. Ablation study

We conduct ablation studies to further validate the effectiveness of our model under different configurations, including different numbers of CACB in the model, different condition settings, and different parameter sharing settings.

The number of CACB. To explore the impact of the number of CACB (i.e., K) in our PanFlowNet, we conduct the experiment with a varying number of parameters K. Table 4 shows the results of different K from 1 to 4. It can be seen that the PSNR performance increases as the number of stages increases. For easy observation, we also visualize obtained HRMS images for different numbers of CACB in Fig. 6, from which we can see that the visual results and the error maps are the best when the number of CACB is 4. Thus we choose $K = 4$ in all the experiments.

Influence of different condition. In the pan-sharpening task, two conditions exist for our method ($\mathbf{L}$ and $\mathbf{P}$), and here, we employ an ablation study to explore the effectiveness of these two conditions. As shown in Tab. 5, we can see that the best performance improvement is achieved when both conditions are available, while the absence of any condition leads to the worst performance due to the fact that it becomes a purely generative task.

Parameter sharing. We evaluate the scenario where the parameters are not shared when $K = 4$. In other words, the parameters of different CABAs in PanFlowNet are no longer shared. From Table 5, we can see that the performance of the model will be improved to some extent without parameter sharing, but it is not a good choice for reducing the model complexity. In our experiment, we still adopt the parameter sharing technique.

4.6. Limitation

Our work has several limitations. The generated HRMS images are diverse, and different HRMS images with different properties are sampled. For this diversity, our method

Table 5: The results of different configurations on WorldViewII. The best and the second best values are highlighted in **bold** and underline, respectively. (PS: Parameters Sharing)

Configuration	L	P	PS	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
I	✗	✗	✓	31.3136	0.9033	0.0840	3.2813
II	✓	✗	✓	36.1760	0.9058	0.0315	1.6287
III	✗	✓	✓	40.8503	0.9647	0.0253	1.0539
IV	✓	✓	✗	**42.0865**	**0.9719**	**0.0215**	**0.9062**
PanFlowNet(Ours)	✓	✓	✓	41.8584	0.9712	0.0224	0.9335

has weak controllability for such different properties of HRMS images and cannot readily generate the HRMS images that match our desired properties, such as generating HRMS images with higher SSIM. In future work, we will try to add controllable elements to control the generated HRMS images so that they can satisfy our demand.

5. Conclusion

In this paper, we proposed a novel neural network architecture used for pan-sharpening called PanFlowNet. Specifically, we introduce a flow-based deep network for pan-sharpening, and this network is capable of accurately learning the distribution of realistic HRMS images condition on the LRMS and PAN images. To the best of our knowledge, this is the first attempt to employ generative methods to learn the distribution of HRMS samples for the pan-sharpening task. The trained network can generate diverse HRMSs by inputting different noises. Extensive experimental results demonstrate the effectiveness and superiority of the proposed network.

6. Acknowledgments

This work was supported by the National Key R&D Program of China (2022YFA1004100), the National Natural Science Foundation of China (NSFC) under Grant 62272375, 61922075 and 82272070, the CAS Project for Young Scientists in Basic Research (Grant No. YSBR-067) and the Macao Science and Technology Development Fund under Grant 061/2020/A2. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM TOG*, 40(3):1–21, 2021.

[2] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019.

[3] Coloma Ballester, Vicent Caselles, Laura Igual, Joan Verdera, and Bernard Rougé. A variational model for p+ xs image fusion. *IJCV*, 69(1):43–58, 2006.

[4] Thomas Blaschke, Stefan Lang, Eric Lorup, Josef Strobl, and Peter Zeil. Object-oriented image processing in an integrated gis/remote sensing environment and perspectives for environmental applications. *Environmental information for planning, politics and the public*, 2:555–570, 2000.

[5] Jiajun Cai and Bo Huang. Super-resolution-guided progressive pansharpening based on a deep convolutional neural network. *IEEE TGRS*, 2020.

[6] Xiangyong Cao, Yang Chen, and Wenfei Cao. Proximal pannet: A model-based deep network for pansharpening. volume 36, pages 176–184, 2022.

[7] Xiangyong Cao, Xueyang Fu, Danfeng Hong, Zongben Xu, and Deyu Meng. Pancsc-net: A model-driven deep unfolding method for pansharpening. *IEEE TGRS*, pages 1–13, 2021.

[8] Xiangyong Cao, Feng Zhou, Lin Xu, Deyu Meng, Zongben Xu, and John Paisley. Hyperspectral image classification with markov random fields and a convolutional neural network. *IEEE TIP*, 27(5):2354–2367, 2018.

[9] Wjoseph Carper, Thomasm Lillesand, and Ralphw Kiefer. The use of intensity-hue-saturation transformations for merging spot panchromatic and multispectral image data. *Photogrammetric Engineering and remote sensing*, 56(4):459–467, 1990.

[10] Chen Chen, Yeqing Li, Wei Liu, and Junzhou Huang. Sirf: Simultaneous satellite image registration and fusion in a unified framework. *IEEE TIP*, 24(11):4213–4224, 2015.

[11] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *CVPR*, pages 182–192, 2021.

[12] Liang-Jian Deng, Gemine Vivone, Weihong Guo, Mauro Dalla Mura, and Jocelyn Chanussot. A variational pansharpening approach based on reproducible kernel hilbert space and heaviside function. *IEEE TIP*, 27(9):4330–4344, 2018.

[13] Liang-Jian Deng, Gemine Vivone, Cheng Jin, and Jocelyn Chanussot. Detail injection-based deep convolutional neural networks for pansharpening. *IEEE TGRS*, 59(8):6995–7010, 2020.

[14] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[16] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2015.

[17] Vinicius Ferraris, Nicolas Dobigeon, Qi Wei, and Marie Chabert. Robust fusion of multiband images with different spatial and spectral resolutions for change detection. *IEEE Transactions on Computational Imaging*, 3(2):175–186, 2017.

[18] Xueyang Fu, Zihuang Lin, Yue Huang, and Xinghao Ding. A variational pan-sharpening with local gradient constraints. In *CVPR*, pages 10265–10274, 2019.

[19] Xueyang Fu, Wu Wang, Yue Huang, Xinghao Ding, and John Paisley. Deep multiscale detail networks for multiband spectral image sharpening. *IEEE TNNLS*, 32(5):2090–2104, 2020.

[20] Hassan Ghassemian. A review of remote sensing image fusion methods. *Information Fusion*, 32:75–89, 2016.

[21] 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.

[22] R Haydn. Application of the ihs color transform to the processing of multisensor data and image enhancement. In *Proc. of the International Symposium on Remote Sensing of Arid and Semi-Arid Lands, Cairo, Egypt, 1982*, 1982.

[23] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *ICML*, pages 2722–2730. PMLR, 2019.

[24] Zi-Rong Jin, Tian-Jing Zhang, Cheng Jin, and Liang-Jian Deng. Weighted shallow-deep feature fusion network for pansharpening. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 2632–2635. IEEE, 2021.

[25] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[26] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE TPAMI*, 43(11):3964–3979, 2020.

[27] Chiman Kwan, Joon Hee Choi, Stanley H Chan, Jin Zhou, and Bence Budavari. A super-resolution and fusion approach to enhancing hyperspectral images. *Remote Sensing*, 10(9):1416, 2018.

[28] Craig A Laben and Bernard V Brower. Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening, Jan. 4 2000. US Patent 6,011,875.

[29] Wenzhi Liao, Xin Huang, Frieke Van Coillie, Guy Thoonen, Aleksandra Pižurica, Paul Scheunders, and Wilfried Philips. Two-stage fusion of thermal hyperspectral and visible rgb image by pca and guided filter. In *2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. Ieee, 2015.

[30] JG 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.

[31] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, pages 715–732. Springer, 2020.

[32] Jiayi Ma, Wei Yu, Chen Chen, Pengwei Liang, Xiaojie Guo, and Junjun Jiang. Pan-gan: An unsupervised pan-sharpening method for remote sensing image fusion. *Information Fusion*, 62:110–120, 2020.

[33] SG Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE TPAMI*, 11(7):674–693, 1989.

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

[35] Xiangchao Meng, Huanfeng Shen, Huifang Li, Liangpei Zhang, and Randi Fu. Review of the pansharpening methods for remote sensing images based on the idea of meta-analysis: Practical discussion and challenges. *Information Fusion*, 46:102–113, 2019.

[36] Jorge Nunez, Xavier Otazu, Octavi Fors, Albert Prades, Vicenc Pala, and Roman Arbiol. Multiresolution-based image fusion with additive wavelet decomposition. *IEEE TGRS*, 37(3):1204–1211, 1999.

[37] George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64, 2021.

[38] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, pages 1530–1538. PMLR, 2015.

[39] Robert A Schowengerdt. Reconstruction of multispatial, multispectral image data using spatial frequency content. *Photogrammetric Engineering and Remote Sensing*, 46(10):1325–1334, 1980.

[40] Vijay P Shah, Nicolas H Younan, and Roger L King. An efficient pan-sharpening method via a combined adaptive pca approach and contourlets. *IEEE TGRS*, 46(5):1323–1335, 2008.

[41] Haoliang Sun, Ronak Mehta, Hao H Zhou, Zhichun Huang, Sterling C Johnson, Vivek Prabhakaran, and Vikas Singh. Dual-glow: Conditional flow-based generative model for modality transfer. In *ICCV*, pages 10611–10620, 2019.

[42] Claire Thomas, Thierry Ranchin, Lucien Wald, and Jocelyn Chanussot. Synthesis of multispectral images to high spatial resolution: A critical review of fusion methods based on remote sensing physics. *IEEE TGRS*, 46(5):1301–1312, 2008.

[43] Xin Tian, Yuerong Chen, Changcai Yang, and Jiayi Ma. Variational pansharpening by exploiting cartoon-texture similarities. *IEEE TGRS*, pages 1–16, 2021.

[44] Gemine Vivone, Luciano Alparone, Jocelyn Chanussot, Mauro Dalla Mura, Andrea Garzelli, Giorgio A Licciardi, Rocco Restaino, and Lucien Wald. A critical comparison among pansharpening algorithms. *IEEE TGRS*, 53(5):2565–2586, 2014.

[45] Lucien Wald, Thierry Ranchin, and Marc Mangolini. Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images. *Photogrammetric engineering and remote sensing*, 63(6):691–699, 1997.

[46] Yudong Wang, Liang-Jian Deng, Tian-Jing Zhang, and Xiao Wu. Ssconv: Explicit spectral-to-spatial convolution for pansharpening. In *ACMMM 2021*, pages 4472–4480, 2021.

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

[48] Qi Xie, Minghao Zhou, Qian Zhao, Deyu Meng, Wangmeng Zuo, and Zongben Xu. Multispectral and hyperspectral image fusion by ms/hs fusion net. In *CVPR*, pages 1585–1594, 2019.

[49] Han Xu, Jiayi Ma, Zhenfeng Shao, Hao Zhang, Junjun Jiang, and Xiaojie Guo. Sdpnet: A deep network for pansharpening with enhanced information representation. *IEEE TGRS*, 59(5):4120–4134, 2020.

[50] Han Xu, Jiayi Ma, Zhenfeng Shao, Hao Zhang, Junjun Jiang, and Xiaojie Guo. Sdpnet: A deep network for pansharpening with enhanced information representation. *IEEE TGRS*, 59(5):4120–4134, 2021.

[51] Shuang Xu, Jiangshe Zhang, Zixiang Zhao, Kai Sun, Junmin Liu, and Chunxia Zhang. Deep gradient projection networks for pan-sharpening. In *CVPR*, pages 1366–1375, June 2021.

[52] Junfeng Yang, Xueyang Fu, Yuwen Hu, Yue Huang, Xinghao Ding, and John Paisley. Pannet: A deep network architecture for pan-sharpening. In *ICCV*, pages 5449–5457, 2017.

[53] Mingde Yao, Dongliang He, Xin Li, Zhihong Pan, and Zhiwei Xiong. Bidirectional translation between uhd-hdr and hd-sdr videos. *IEEE Transactions on Multimedia*, 2023.

[54] Qiangqiang Yuan, Yancong Wei, Xiangchao Meng, Huanfeng Shen, and Liangpei Zhang. A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening. *IEEE J-STARS*, 11(3):978–989, 2018.

[55] Hao Zhang and Jiayi Ma. Gtp-pnet: A residual learning network based on gradient transformation prior for pansharpening. *ISPRS Journal of Photogrammetry and Remote Sensing*, 172:223–239, 2021.

[56] Tian-Jiang Zhang, Liang-Jian Deng, Ting-Zhu Huang, Jocelyn Chanussot, and Gemine Vivone. A triple-double convolutional neural network for panchromatic sharpening. *IEEE TNNLS*, 2022.

[57] Zi-Yao Zhang, Ting-Zhu Huang, Liang-Jian Deng, Jie Huang, and Hong-Xia Dou. Pan-sharpening via rog-based filtering. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 2790–2793. IEEE, 2019.

[58] Man Zhou, Jie Huang, Yanchi Fang, Xueyang Fu, and Aiping Liu. Pan-sharpening with customized transformer and invertible neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3553–3561, 2022.

[59] Man Zhou, Keyu Yan, Jie Huang, Zihe Yang, Xueyang Fu, and Feng Zhao. Mutual information-driven pan-sharpening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1798–1808, 2022.

[60] Man Zhou, Keyu Yan, Jinshan Pan, Wenqi Ren, Qi Xie, and Xiangyong Cao. Memory-augmented deep unfolding network for guided image super-resolution. *IJCV*, 2022.