Unsupervised Deep Learning in Remote Sensing with Application to Image Fusion and Denoising

Han Van Nguyen

Dissertation submitted in partial fulfillment of a
Philosophiae Doctor degree in Electrical and Computer Engineering

Supervisors
Professor Magnús Ö. Úlfarsson
Professor Jóhannes R. Sveinsson

Doctoral Committee
Professor Magnús Ö. Úlfarsson
Professor Jóhannes R. Sveinsson
Professor Mauro Dalla Mura

Opponents
Professor Danfeng Hong
Professor Farid Melgani

Faculty of Electrical and Computer Engineering
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, December 2022

Unsupervised Deep Learning in Remote Sensing with Application to Image Fusion and Denoising

Dissertation submitted in partial fulfillment of a *Philosophiae Doctor* degree in Electrical and Computer Engineering

Faculty of Electrical and Computer Engineering
School of Engineering and Natural Sciences
University of Iceland
Hjarðarhagi 2-6
107 Reykjavik
Iceland

Telephone: 525-4000

Bibliographic information:
Han Van Nguyen (2022) *Unsupervised Deep Learning in Remote Sensing with Application to Image Fusion and Denoising*, PhD dissertation, Faculty of Electrical and Computer Engineering, University of Iceland, 108 pp.

ISBN 978-9935-9697-3-6

Printing: Háskólaprent
Reykjavik, Iceland, December 2022

ABSTRACT

Optical remote sensing (RS) uses optical sensors to create images of the Earth's surface. Those imaging sensors are mounted on spaceborne or airborne vehicles and capture visible, near-infrared, and shortwave infrared radiation reflected from the Earth's surface. Optical remote sensing imaging systems usually provide multi-band images, such as hyperspectral images (HSIs) and multispectral images (MSIs), often with band-dependent spatial resolution. However, those images are often corrupted by noise and have low spatial/spectral resolution. This is caused by several reasons, such as atmospheric absorption, sensor imperfection, and a trade-off between spectral and spatial resolutions. Therefore, denoising or sharpening the images is crucial for many RS applications.

This thesis focuses on HSI denoising and RS image fusion. HSI denoising is the problem of recovering the original true image from the noisy HSI. On the other hand, in RS image fusion, one has a set of co-registered images, each acquired at a different frequency band and having a different spatial resolution. The aim is to sharpen the images so they all have a spatial resolution equal to the highest spatial resolution of the input images.

The main objective of this thesis is to propose new HSI denoising and RS image fusion methods using unsupervised deep learning (DL). The proposed unsupervised DL-based methods are inspired by the deep image prior idea, which centers around training a convolutional neural network (CNN) in an unsupervised manner. Moreover, several novel points are proposed, such as sparse and low-rank ideas, the sensors' modulation transfer functions (MTFs) utilization, and the usage of Stein's unbiased risk estimate (SURE). The proposed HSI denoising and RS image fusion methods are summarized below.

- The thesis proposes two HSI denoising methods based on unsupervised CNNs. The first method incorporates the sparse and low-rank property induced by the high spectral and spatial correlation of HSIs to a CNN. Training a CNN for HSI denoising using the sparse and low-rank data significantly reduces computational load and improves the results. The second HSI denoising method derives a SURE-based loss function for training a CNN. Since SURE is an unbiased estimate of the mean-square error (MSE) between the denoised and the reference images and is calculated using only the noisy image, training a CNN with SURE loss avoids overfitting and is unsupervised. Additionally, the SURE-based HSI denoising method can be extended to deal with non-Gaussian noise and to work with low-dimensional HSI data obtained by projecting the original data to a subspace. The SURE-based method improves results and is more feasible in a practical HSI denoising application.

- The thesis proposes a Sentinel-2 (S2) image fusion method using a single unsupervised CNN where the sensors' MTFs are embedded as a network layer. The proposed method uses a single CNN to sharpen both the 20 m and 60 m bands of the S2 image, unlike traditional DL-based methods that usually use separate CNN to sharpen each resolution band. Moreover, since the manufacturer provided the S2 sensors' MTFs, the proposed method employs an MTF-based degradation model as a CNN layer. By doing this, training the CNN is unsupervised, and the fused images are well-preserved in both spectral and spatial domains.
- A general framework for RS image fusion is proposed. In this framework, a loss function based on SURE and a linear operator that maps an LR image to its HR is derived for training a CNN. The loss function used in this method has two main benefits. First, SURE is an unbiased estimate of the MSE between the fused and the reference images and is computed without using the reference image. Thus, the method is unsupervised and avoids overfitting. Second, the linear operator is chosen to give upsampling results, at least better than a simple interpolation method, e.g., bicubic. Therefore, the linear operator improves the overall fusion results. The method is applied for three representative RS image fusion problems, i.e., MSI and HSI fusion, S2 sharpening, and pansharpening, where the back-projection operator is used as a linear operator in the SURE-based loss. Experimental results show that the fusion quality is significantly enhanced by using back-projection and SURE.

ÚTDRÁTTUR

Ljósfræðileg fjarkönnun (RS) notar myndskynjara til að taka myndir af yfirborði jarðar. Þessir myndskynjarar eru festir á gervihnetti eða flugvélar og fanga sýnilega, nær-innrauða og stuttbylgju-innrauða geislun sem endurkastast frá yfirborði jarðar.

Ljósfræðileg fjarkönnunarmyndkerfi eru skilgreind útfrá fjölda tíðnibanda og helstu tegundir mynda eru margrása myndir (e. multispectral images (MSI)), og fjölrásamyndir (e. hyperspectral images (HSI)). Af verkfræðilegum og eðlisfræðilegum ástæðum hafa þessar myndir rýmisupplausn (e. spatial resolution) sem er tíðniháð og einnig innihalda þessar myndir oft suð. Í þessari ritgerð er lögð áhersla á að auka gæði MSI og HSI bæði með því að suðsía þær (e. denoising) og auka rýmisupplausn þeirra með myndsambræðslu (skerping) (e. image fusion).

Þessi ritgerð er þróar nýjar aðferðir sem eru byggðar á því að nota óleiðbeindar djúpnámsaðferðir (e. deep learning) sem byggja á földunarnetum (e. convolution neural networks) til að suðsíða og skerpa MSI og HSI myndir. Til þess að þróa þessar aðferðir eru notaðar hugmyndir frá merkjafræði og tölfræði eins og t.d., notkun á tíðnisvörun myndskynjarana, SURE (e. Stein's unbiased risk estimator), rýr merkjafræði (e. sparse signal processing), og að fjarkönnunarmyndir "lifa" oft í stærðfræðilegu rúmi af miklu lægri vídd en þær eru teknar á.

Í þessari ritgerð eru þróaðar tvær aðferðir til suðsíunnar á fjölrásamyndum (e. hyperspectral images (HSI)) með óileiðbeindum földunarnetum (e. convolution neural networks (CNN)). Fyrri aðferðin nýtir rýra merkjafræði (e. sparse signal processing) og að fjarkönnunarmyndir má oft greina í stærðfræðilegu rúmi af miklu lægri vídd en þær eru teknar á. Þjálfun földunarneta með rýrum gögnum af lágri vídd dregur verulega úr reikniþunga og bætir niðurstöður. Seinni aðferðin leiðir út tapfall (e. loss function) byggt á SURE (e. Stein's unbiased risk estimator) til þjálfunar á földunarnetum. Þar sem sem reikna má SURE útfrá myndum sem innihalda suð og það er óbjagaður metill á meðalferskekkju milli suðsíaðra mynda og viðmiðunarmynda kemst þjálfun tauganeta með SURE tapfalli hjá því að ofmáta gögnin og er óleiðbeind. Einnig má útvíkka þessa SURE miðuðu suðsíunnar aðferð til að vinna á ógaussísku suði og virka með víddafækkuðum fjölrásamyndum. Aðferðin bætir niðurstöður og er fýsilegri í raunverulegum hagnýtingum til suðsíunnar.

Þessi ritgerð þróar myndsambræðsluaðferð (e. image fusion method) fyrir Sentinel-2 (S2) myndir með óleiðbeindu földunarneti þar sem tíðnisvörun myndskynjaranna er innfeld sem lag í netið. Aðferðin notar stakt földunarnet til að skerpa bæði 20 og 60 m bönd S2 mynda, ólíkt mörgum fyrri djúpnámsaðferðum (e. deep learning) sem flestar nota aðskild földunarneta til að skerpa bönd af ólíkri upplausn. Ennfremur nýtar aðferðin mælda tíðnisvörun myndskynjaranna frá framleiðanda þeirra sem innfelt sem lag í földunarnetið til að herma myndbreytingareiginleika þeirra. Þannig má nota óleiðbeinda þjálfun en viðhalda bæði róf- og rúmþáttum sambræddu myndanna.

Víðtækt fyrirkomulag til myndbræðslu fjarskynjunar mynda er sett fram. Innan þessa fyrirkomulags er tapfall byggt á SURE notað ásamt línulegum virkja sem varpar mynd af lágri upplausn í hærri upplausn til að þjálfa földunarnet. Tapfallið hefur tvo sérlega kosti. Í fyrsta lagi er SURE reiknað án viðmiðunarmynda útfrá myndum sem innihalda suð en er óbjagaður metill á meðalferskekkju milli suðsíaðrar myndar og undirlyggjandi viðmiðunarmyndar. Þar af leiðir að aðfeðrðin er óleiðbeind og kemst hjá því að ofmáta gögn. Í öðru lagi er línulegi virkinn valinn til þess að gefa úrtaksþéttingu (e. upsampling) sem er alltént betri en einföld brúun á borð við tvívíða þriðja stigs brúun (e. bicubic interpolation). Línulegi virkinn bætir með því móti heildargæði myndbræðslunnar. Aðferðinni er beitt á þrjú einkennandi verkefni í myndbræðslu fjarskynjunarmynda, myndbræðslu margrása mynda og fjölrásamynda (e. multispectral images (MSI) og hyperspectral images (HSI)), skerpingu S2 mynda og panskerping (e. pansharpening), þar sem afturvarpsvirki (e. back-projection operator) er notaður sem línulegur virki með SURE tapfalli. Niðurstöður tilrauna sýna að gæði myndbræðslu aukast verulega með notkun afturvarps og SURE.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ORIGINAL PUBLICATIONS

Paper I: **H. V. Nguyen**, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral image denoising using SURE-based unsupervised convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 3369–3382, 2020.

Paper II: **H. V. Nguyen**, M. O. Ulfarsson, J. R. Sveinsson, and M. Dalla Mura, "Sentinel-2 sharpening using a single unsupervised convolutional neural network with MTF-based degradation model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 6882–6896, 2021.

Paper III: **H. V. Nguyen**, M. O. Ulfarsson, J. R. Sveinsson, and M. D. Mura, "Deep SURE for unsupervised remote sensing image fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-13, 2022.

Paper IV: **H. V. Nguyen**, M. O. Ulfarsson, and J. R. Sveinsson, "SURE based convolutional neural networks for hyperspectral image denoising," in *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2020, pp. 1784–1787.

Paper V: **H. V. Nguyen**, M. O. Ulfarsson, J. R. Sveinsson, and J. Sigurdsson, "Zero-shot Sentinel-2 sharpening using a symmetric skipped connection convolutional neural network," in *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2020, pp. 613–616.

Paper VI: **H. V. Nguyen**, M. O. Ulfarsson, and J. R. Sveinsson, "Sharpening the 20 m bands of Sentinel-2 image using an unsupervised convolutional neural network," in *2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2021, pp. 2875–2878.

Paper VII: **H. V. Nguyen**, M. O. Ulfarsson, J. Sigurdsson, and J. R. Sveinsson, "Deep sparse and low-rank for hyperspectral image denoising," in *2022 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2022, pp. 1217-1220.

Paper VIII: **H. V. Nguyen**, M. O. Ulfarsson, J. R. Sveinsson, and M. D. Mura, "Hyperspectral super-resolution by unsupervised convolutional neural network and SURE," in *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2022, pp. 903-906.

ABBREVIATIONS

BP Back-projection

CNN Convolutional neural network

DIP Deep image prior

DL Deep learning

DWT Discrete wavelet transform

ERGAS Erreur Relative Globale Adimensionnelle de Synthèse

FFT Fast Fourier transform

HSI Hyperspectral image

HR High resolution

LR Low resolution

MC Monte Carlo

MSI Multispectral image

MSE Mean-square-error

MTF Modulation transfer function

PAN Panchromatic image

PSF Point spread function

PSNR Peak signal-to-noise ratio

RS Remote sensing

SAM Spectral angle mapper

SRE Signal-to-reconstructed error

SRF Spectral response function

SSIM Structure similarity index

SURE Stein's unbiased risk estimate

SVD Singular value decomposition

NOTATIONS

$\mathcal{X}, \mathbf{X}, \mathbf{x}$	Original/reference image in 3-dimensional array, matrix and vector, respectively
$\mathcal{Y}, \mathbf{Y}, \mathbf{x}$	Noisy/low resolution image in 3-dimensional array, matrix and vector, respectively
$\hat{\mathcal{X}}, \hat{\mathbf{X}}, \hat{\mathbf{x}}$	Denoised/fused image in 3 dimensional array, matrix and vector, respectively
$\mathbf{G}$	Guided/high resolution image to be fused
$\mathbf{M}$	Downsampling matrix
$\mathbf{B}$	Blurring matrix
$\mathbf{H}$	Degradation operator
$\mathbf{H}^\dagger$	Back-projection (pseudo inverse of $\mathbf{H}$)
$f_{\boldsymbol{\theta}}(\cdot)$	Network mapping function
$\boldsymbol{\theta}$	Network parameters
$\mathbf{P}$	Linear operator
$\mathbf{R}$	Spectral response matrix
$\mathbf{N}$	Noise matrix
$\mathbf{n}$	Noise vector
$\boldsymbol{\Omega}$	Noise covariance matrix
σ_i	Noise standard deviation of ith band
h, w	Height and width of a noisy/low resolution image
H, W	Height and width of a guided/high resolution image to be fused
d	Number of spectral bands of a noisy/low resolution image
D	Number of spectral bands of a guided/high resolution image to be fused
$N = h \times w$	Number of pixels in a spectral band of a low resolution image
$M = H \times W$	Number of pixels in a spectral band of a high resolution image

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisors, Professor Magnús Örn Úlfarsson and Professor Jóhannes Rúnar Sveinsson, for their guidance, support and encouragement. Specially, their lectures and discussions opened me to the new horizon of knowledge. I am also thankful to my PhD committee member, Professor Mauro Dalla Mura, for his valuable advice.

My sincere thanks go to Professor Danfeng Hong and Professor Farid Melgani who accepted to be the thesis opponents, and they gave me positive comments that improved the quality of my thesis. Also, I would like to thank Professor Lotta María Ellingsen for being the chair of my thesis defence ceremony.

I acknowledge the financial support for my PhD study from the Icelandic Research Fund under Grant 174075-05 and Grant 207233-051, and the University of Iceland Doctoral Fund under Grant 1547-154305.

During my PhD studying in the University of Iceland, I have been with many great friends and colleagues. For their friendship, help and encouragement to me and my family, I would like to give sincere appreciation to Professor Jakob Sigurðsson, Dr. Bin Zhao, Burkni Pálsson, Sveinn Eiríkur Ármannsson, Magnús Magnússon, Li Mengyu, my best friends Ninh Tang and Hung Pham, and all the teammates of the Viet-Ice football team.

Finally, I would like to give special thanks from my heart to my family: My parents, my wife Hang Thi Nguyen, my lovely sons Nam Quoc Nguyen and Phong Hai Nguyen. Their unconditioned, unlimited love and encouragement motivate me to overcome all difficulties.

CHAPTER 1

INTRODUCTION

This chapter introduces optical remote sensing imagery and the problem of remote sensing image restoration. A comprehensive literature review on remote sensing image restoration is presented, focusing on two critical issues, i.e., hyperspectral image denoising and remote sensing image fusion. The contributions and organization, along with a list of publications, are given at the end of this chapter.

1.1 Remote sensing images

Remote sensing (RS) is acquiring the reflected radiation from the Earth's surface using sensors mounted in a spaceborne or airborne system. The radiation source can be an artificial source, such as in a radar system, or the reflection from the sun. The observed reflected radiation is often represented as an image. In this thesis, we are interested in the optical RS image observed in the wavelength range of visible, near-infrared (NIR), and shortwave regions. Those images to be processed are usually stored in the form of digital images, which are characterized by spectral, spatial, and radiometric resolutions [1]. Based on those characteristics, optical RS images are categorized into three main classes, i.e., the hyperspectral image (HSI), the multispectral image (MSI), and the panchromatic image (PAN).

HYPERSPECTRAL IMAGE

Hyperspectral imaging, also known as imaging spectroscopy, is a process of acquisition, analysis, and interpretation of the spectra of an objection which is usually given by a hyperspectral image (HSI) composed of several hundred narrow bands. In RS, HSIs are remotely acquired by using a hyperspectral spectrometer (sensor) carried in an airborne platform. The hyperspectral spectrometer measures the sunlight reflected from the Earth's surface in the visible, NIR, and shortwave regions of the spectrum ($300 - 2500$ nm). An HSI is given in a three-dimensional array of $h \times w \times d$. In a spatial perspective, an HSI is a set of d gray-scale images of $h \times w$ pixels, and in a spectral view, an HSI consists of $h \times w$ spectral vectors of length d. An example of the HSI is the Indian Pines HSI dataset collected by the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) [2] shown in Fig. 1.1. This image contains 145×145 pixels and 220 spectral bands of the wavelength ranging from 400 nm to 2500 nm, with the spatial and spectral resolutions being 20 m and 10 nm, respectively.

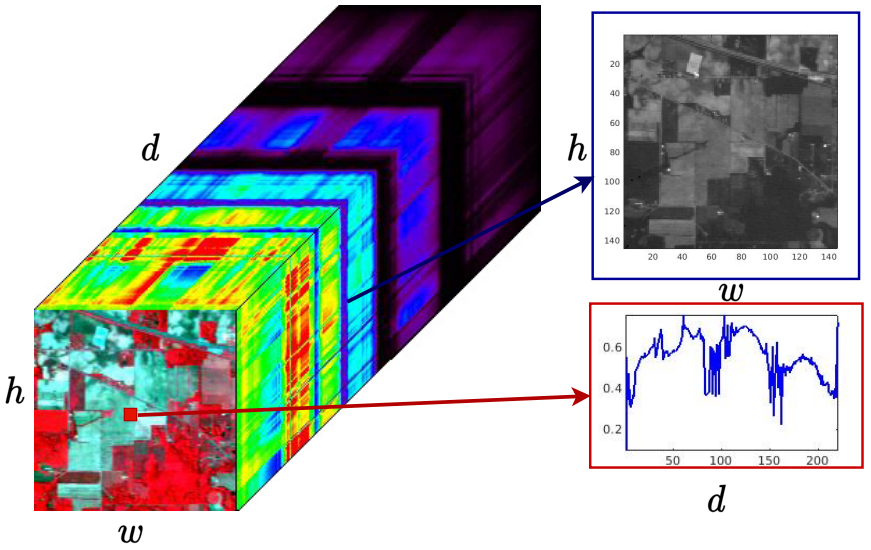


Figure 1.1. An HSI collected by AVIRIS sensor.

Since HSI provides rich spectral information embedded in several hundred bands, it gives more potential to analyze materials with high precision. HSIs are helpful in many applications such as agriculture, environmental monitoring, mineral mining, target detection and classification, and many more [3]–[5].

RS HSIs are corrupted with noise and have a low spatial resolution. This is because there is always a trade-off between spatial and spectral resolution. Since HSI has a very high spectral resolution, the radiation energy (i.e., signal-to-noise ratio: SNR) received within a spectral band is low. To guarantee a sufficient image quality (e.g., SNR), the spatial resolution of HSI must be lower than one of a broadband image (e.g., a PAN). Additionally, the reflected sunlight reaches the HSI sensors via the atmosphere. Atmospheric effects, such as absorption and scattering from water vapor and aerosol, degrade the HSI quality [6]. HSIs also suffer from several kinds of noise, such as Gaussian noise, thermal noise, and stripped noise [7], caused by the sensor imperfection. Therefore, improving the HSI quality has to be concerned for further processing and analysis of HSIs.

Multispectral and Panchromatic images

Nowadays, many satellite imaging systems provide multispectral images (MSIs) and a panchromatic image (PAN) for Earth observation. The MSI and PAN sensors are mounted on the same platform and give the MSI and PAN that display the same scene simultaneously. This means that MSI and PAN are co-registered but have different characteristics. PAN is a single broadband image with a spectrum ranging from visible to NIR wavelength. MSI comprises several narrow bands (but fewer than HSI) with spectra that overlap the PAN spectrum. Since PAN is obtained in a wider range of

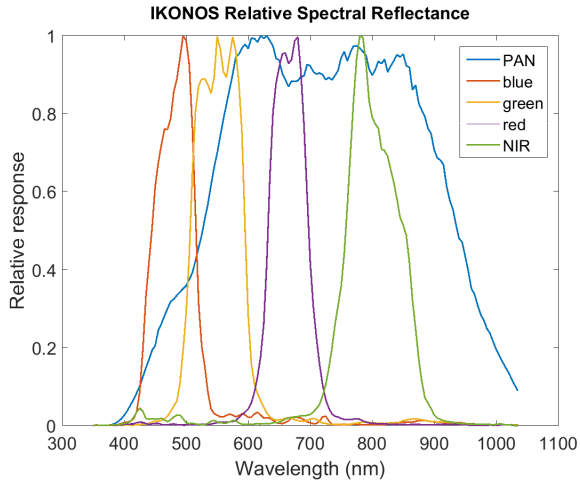


Figure 1.2. IKONOS MSI and PAN spectral response.

wavelengths than MSI, the spatial resolution of PAN is higher than the spatial resolution of MSI. A typical multispectral imaging system, such as IKONOS [8], provides four spectral bands from visible red, green and blue (RGB) to NIR and one PAN. The IKONOS sensor spectral response is shown in Fig. 1.2. The IKONOS PAN has a spatial resolution four times higher than MSI. Other system, e.g., Sentinel 2 (S2), has thirteen spectral bands of three different spatial resolutions of $10, 20$, and 60 m, without a PAN.

MSIs have relatively fine spatial and coarse spectral resolution, they are usually used in vegetation and water mapping [9], [10], crop monitoring [11], object classification [12]. In many applications, the spatial resolution of MSI is enhanced by merging the MSI with the PAN. This process is called pansharpening and often benefits the performance of the applications [13].

1.2 Remote sensing image restoration

As discussed above, RS images are usually corrupted, for example, suffering from noises (Gaussian noise, stripe noise, impulse noise, mixed noise, etc.); missing data (cloud, shadow, sensor malfunction, etc.); and spatial resolution degradation due to equipment limitations, working conditions, limited radiance energy, and generally narrow bandwidth. These phenomena severely degrade the quality of RS images and limit the performance of the subsequent applications, e.g., classification [14], unmixing [15], and target detection [16]. Therefore, restoration is a critical step to improve the quality of RS images.

RS image restoration is recovering the true, unknown image from the observed, degraded image [17]. Usually, the degradation is assumed to be known and used as prior knowledge to reconstruct the image. Depending on the degradation, several RS image restoration problems have been concerned. Denoising [18] is the task of removing noise

from a perturbed image to obtain a clean image. Inpainting [19] is to estimate the true image from the incompleted image. Super-resolution [20] is the problem of recovering the original high (spatial) resolution (HR) image using its low (spatial) resolution (LR) version. RS image fusion [21] is a special case of super-resolution where a part of the original HR image is known and is fused with the LR image in the estimation process. In this thesis, HSI denoising and RS image fusion are addressed.

1.2.1 Hyperspectral image denoising

Many HSI denoising methods have been proposed to improve the quality of HSI applications. Those HSI denoising methods can be roughly divided into three categories: The traditional single band methods, the multiband methods, and the deep learning (DL)-based methods.

Traditional single-band methods

The most straightforward HSI denoising approach is to apply the classical two-dimensional (2-D) image denoising method band-by-band. Examples of those methods are the block-matching and three-dimensional (3-D) filtering (BM3D) algorithm [22], the weighted nuclear norm minimization (WNNM) algorithm [23], and a learning model such as expected patch log-likelihood (EPLL) [24]. Applying the band-wise denoising methods to HSI data is simple, but the results usually suffer from spectral distortion since those methods ignore the spectral information.

Multi-band methods

HSIs are spectrally and spatially correlated. Therefore many HSI denoising methods have exploited the high correlation between all bands of HSI to improve the denoising performance. Those methods usually take into account multiband information in the denoising process.

The first approach of the multiband methods [25]–[28] employ some linear transformations such as the principal component analysis (PCA), Fourier and wavelet transform to obtain the sparsity in the transform domain. Then, the clean image is separated from noise using a thresholding algorithm on the sparse coefficients. The second multiband approach is model-based. In this approach, the denoising problem is formulated as an inverse problem, and the denoised image is the result of solving a penalized regression problem. The cost function to be optimized is the sum of a fidelity term and one or more regularization terms. Here, the fidelity term tends to keep the solution not too far from the observation, and the regularization term is used to constrain the solution space. The regularizer is usually designed based on prior knowledge about the HSI to be restored. The spectral-spatial total variation (TV) was used in [29]–[32] to deal with Gaussian noise and sparse noise. The global and local self-similarity in a band and between bands were utilized in the methods [33]–[35]. Low-rank approximation [36] is another common idea used in HSI denoising. A tensor-based low-rank approximation using the Tucker3 tensor decomposition was proposed in [37] for both HSI denoising and dimensionality reduction. Similarly, a tensor-based low-rank approximation technique using a learning tensor decomposition approach was proposed in [38]. The low-rank

matrix recovery (LRMR) [39] and its extension with an iterative noise adjusted procedure [40] were developed for Gaussian and mixed type of noise removal, respectively. Additionally, the combination of more than one regularizer is often used. For example, the joint of low-rank and sparse representation was proposed in [41]–[44], and the low-rank and TV were used in [45], [46]. The multiband methods often yield good results if the regularizers are chosen properly. However, those model-based methods suffer from a high computational load, making it hard to select the tuning parameters and regularizers since they are data-dependent.

DEEP LEARNING-BASED METHODS

In the last decades, many HSI denoising methods based on DL have been proposed with the rapid development of computational resources and breakthroughs in artificial intelligence, specifically in deep learning (DL). Early methods are inspired by the DL-based denoising methods, which were developed for 2-D images [47]–[49]. Recently developed DL-based HSI denoising methods focused on investigating more sophisticated network structures that were designed to adapt to the HSI domain knowledge. Convolutional neural network (CNN) and residual CNN [50] were used in [51]–[53], to extract the useful spectral-spatial features, which leveraged the denoising results. Three-dimensional (3-D) CNNs have been widely used in HSI denoising since the HSI data are naturally a 3-D array. Examples are the 3-D atrous CNN used in [54], 3-D CNN with attention gates proposed in [55] and 3-D quasi-recurrent CNN used in [56]. Another new trend is to embed the HSI characteristics with the DL models. The paper [57] exploited the HSI sparse representation and incorporated it with a CNN, while the papers [58], [59] used the low-rank and nonlocal similarity with the DL models. Those methods have shown superiority over the ones using plain CNN. The limitation of the DL-based methods mentioned above is that the networks must be trained using a training dataset containing noisy-ground truth pairs (supervised learning). However, the ground truth (i.e., the clean HSI) is hard to obtain in RS due to the technical limitation and the high cost. To overcome this limitation, unsupervised DL-based methods for HSI denoising have been proposed. The main aspect of the unsupervised DL-based methods is to ignore the dependency on the ground truth. Representative unsupervised DL-based methods are the work proposed in [60]–[62], which were based on the deep image prior (DIP) [63]. Other methods in the unsupervised DL approach are the zero-shot denoising method [64] and the hybrid model and DL-based method [65].

1.2.2 Remote sensing image fusion

Because of the technical limitation of the imaging sensors and the trade-off between spectral and spatial resolution, an RS imaging system usually provides high spatial but low spectral resolution and vice versa. This means a PAN has higher spatial resolution but lower spectral resolution than an MSI and an HSI. In practice, many applications require RS images with high spectral and spatial resolution [66]–[68]. This requirement motivates researchers to combine a low spatial resolution but high spectral resolution image with a high spatial resolution but low spectral resolution image to synthesize a new image having both high spatial and spectral resolution. This process is known as

(optical) image fusion [69]. Image fusion is a special case of image super-resolution and is an ill-posed inverse problem. Optical RS has several kinds of image fusion problems depending on the HR and LR images to be fused. Pansharpening [70]–[72] and hyperspectral pansharpening [73]–[75] involve fusing of a PAN with an MSI, and an HSI, respectively. Hypersharpening [76]–[78] is the fusion problem where the spatial source (i.e., HR image) can be composed of several bands. Hypersharpening can be further split into multispectral and hyperspectral (MS-HS) image fusion [79] where the images to be fused are MSI and HSI, and MS-MS image fusion where the fused image sources are both MSIs, e.g., the fusion of all bands of the S2 image [80], and fusion of Landsat 8 and S2 data [81].

Existing optical RS fusion methods can be categorized into three approaches. The first approach consists of classical component substitution (CS) and multiresolution analysis (MRA) methods. The second approach contains model-based methods, and the third approach includes DL-based methods.

CS AND MRA METHODS

The CS [82]–[84] and MRA [85]–[87] methods were originally developed for pansharpening. In those methods, the sharpened images are obtained by estimating the missing details (i.e., the high frequencies) of the LR images and injecting them into the interpolated LR images. The main difference between CS and MRA methods is how the details are estimated. In CS, details are estimated by substituting a component of the MSI in a transformed domain by the PAN. In contrast, the details are obtained in MRA methods using a multiresolution decomposition applied to the PAN. The CS and MRA methods have been extended for hyperspectral pansharpening [88], MS-HS fusion [89], and S2 sharpening [90]. The limitation of the CS and MRA methods is that CS suffers from some spectral distortion, and MRA suffers from some spatial degradation [91].

MODEL-BASED METHODS

In model-based RS image fusion methods, the fusion problem is formulated as an ill-posed inverse problem, and the fused image is the solution to this inverse problem. The common approach to solve this inverse problem is to regularize a regression problem using prior knowledge (i.e., image prior) about the estimated images. One of the most widely used characteristics is the low-rank property which is induced by the high correlation between the bands of the RS image. Low-rank means that it is possible to represent a multi/hyperspectral image in a lower-dimensional space. Low-rank has been exploited in several RS image fusion methods that benefit computation and fusion results. The representative low-rank based methods are [92]–[96] for MS-HS image fusion, [97]–[100] for S2 sharpening, and [101], [102] for pansharpening. The other characteristics of an RS image are spectral-spatial piece-wise smoothness, repetitive patterns across spatial and spectral dimensions, and sparsity in a transform domain. Those characteristics were used in many model-based methods including the TV methods [103], [104], the non-local similarity methods [105], [106] and the methods based on sparse representation [107], [108]. The main drawbacks of those model-based methods are hard to design an image prior, high computation load, and sensitivity to tuning parameters.

DEEP LEARNING-BASED METHODS

Early RS image fusion methods using DL were inspired by the super-resolution DL models developed for computer vision applications. Masi *et al.* [109] extended the three layers CNN originally used in RGB image super-resolution [110] for pansharpening with several maps of nonlinear radiometric indices to boost the results. Yang *et al.* [111] proposed a pansharpening method that used a residual network with high-pass filters to estimate the missing high frequencies of MSI that obtained improvement in fusion results and reduced running time. Similarly, residual CNNs were proposed in [112] and [113] for S2 sharpening and MS-HS image fusion, respectively. Nowadays, more advanced network architectures have been developed for RS image fusion. Examples are 3-D CNN [114]–[116], recurrent structure [117], [118], attention structure [119]–[121], transformer vision [122], [123], and generative adversarial networks (GANs) [124]–[126]. In those DL-based methods, a complex network mapping function was trained by using a big dataset containing observation and ground truth pairs (supervised learning). Although the supervised DL-based methods have demonstrated their efficiency and obtained state-of-the-art results, the performance of those methods depends heavily on the training data. In practice, it is hard to acquire the observation and ground truth image pairs due to the high cost and sensor limitations. Thus, the training datasets are artificially synthesized using a sensor degradation assumption. Training a network with a synthesized dataset may not be correct for real applications [127]. Various unsupervised DL-based methods have been proposed to overcome the limitation of supervised DL-based methods. Most unsupervised DL-based methods [128]–[131] used the DIP [63] that was refined to the RS image. Other unsupervised DL-based methods are based on the spectral and spatial constraint [132], [133] and unmixing model [134], [135].

1.3 Thesis contributions and organization

This thesis addresses image restoration in RS with particular concentration in HSI denoising and RS image fusion. The thesis's main contribution is to propose new methods for HSI denoising and RS image fusion. All the proposed methods in this thesis are unsupervised DL-based and can also be considered as hybrid model-based and DL-based methods. The reason is that those methods require no ground truth to optimize (train) CNN models, and the optimization of the CNN is similar to the model-based methods in which the DIP [63] is used as a regularizer, and the DL optimizer (e.g., Adam [136]) plays the role of the optimization algorithm. Several novel points have been proposed to extend the DIP in the context of RS imagery, such as the HSI sparse and low-rank property, the RS imaging sensor modulation transfer function (MTF), and Stein's unbiased risk estimate (SURE) [137]. Those aspects have shown improvement in both HSI denoising and RS image fusion.

In Chapter 2, two HSI denoising methods are presented. The first method incorporates the sparsity and low-rank to an unsupervised CNN. Since HSIs are spectrally and spatially correlated, using the singular value decomposition (SVD) and the 2-D discrete wavelet transform to obtain the low-rank representation and sparse coefficients

of the HSIs significantly improves the denoising results and reduces the computational complexity. One downside of the unsupervised DL denoising methods based on DIP is that it is prone to overfitting. In DIP, the CNN is optimized by using only the corrupted images. Therefore the output of the CNN (i.e., the denoised image) will be overfitting the noisy image if the training iteration is too long. The second denoising method proposed in this thesis overcomes this limitation by using SURE [137]. Here, SURE is the unbiased estimate of the MSE of the denoised and the reference images and is calculated without using the reference image. By using SURE, training a CNN is unsupervised and avoids overfitting. Additionally, the proposed SURE-based method can be extended to deal with non-Gaussian noise (e.g., Poissonian noise) and with subspace HSI data via the SVD. Various experiments with both simulated and real HSI datasets verify that the proposed unsupervised DL-based HSI denoising methods using sparse, low-rank, and SURE yield good results and outperform the competitive methods.

Chapter 3 details an unsupervised DL-based method for S2 sharpening. The novelty of this method is that it uses a single CNN to sharpen both the 20 m and 60 m bands, and the S2 sensors' MTFs are embedded as a CNN layer. Those ideas are motivated by the facts that S2 bands are highly correlated and the manufacturer provides the S2 sensors' MTFs. By using the MTFs, the sensors' point spread functions (PSFs) are derived and are used for building a degradation model using convolution and downsampling. Several experiments have been done to demonstrate the advantages of implementing a single CNN with the sensors' MTFs used in a convolutional layer. Four real S2 datasets are used to evaluate the performance of the proposed method in both reduced-resolution and full-resolution. The results show that the proposed method gives good sharpened images and better quantitative and qualitative performance than the competitive methods.

In Chapter 4, a general framework for RS image fusion is proposed, which is based on SURE and unsupervised CNN. A new loss function to train a CNN is derived using SURE with a linear operator mapping an LR image to its HR space. Unlike the generalized SURE [138], [139] specified for an orthogonal projection operator, the SURE formula derived here is more straightforward and holds for any linear operator. The advantages of the new SURE loss function are twofold. First, SURE is the unbiased estimate of the MSE of the fused and reference images and is calculated without a reference image. Training a CNN with the SURE loss is unsupervised and avoids overfitting. Second, since the linear mapping operator is a pre-processing step, incorporating it into the SURE loss function improves the fusion results. Three RS image fusion examples, i.e., MS-HS image fusion, S2 sharpening, and pansharpening, are addressed in Chapter 4. In those examples, the back-projection operator [140] is chosen as the linear operator. Experiments show that the proposed method yields superior results in both simulated and real datasets.

The remaining of this thesis is organized as follows. Chapters 2, 3 and 4 present the unsupervised DL methods for HSI denoising, S2 sharpening, and RS image fusion, respectively. Chapter 5 presents some conclusions and future directions, and the Appendix describes the datasets and evaluation metrics used in this thesis.

1.4 Publications

Chapter 2 is based on the following publications:

(a) H. V. Nguyen, M. O. Ulfarsson, J. Sigurdsson, and J. R. Sveinsson, "Deep sparse and low-rank for hyperspectral image denoising," in *2022 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2022, pp. 1217-1220.

(b) H. V. Nguyen, M. O. Ulfarsson, and J. R. Sveinsson, "SURE based convolutional neural networks for hyperspectral image denoising," in *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2020, pp. 1784–1787.

(c) H. V. Nguyen, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral image denoising using SURE-based unsupervised convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 3369–3382, 2020.

Chapter 3 is based on the following publications:

(a) H. V. Nguyen, M. O. Ulfarsson, J. R. Sveinsson, and J. Sigurdsson, "Zero-shot Sentinel-2 sharpening using a symmetric skipped connection convolutional neural network," in *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2020, pp. 613–616.

(b) H. V. Nguyen, M. O. Ulfarsson, and J. R. Sveinsson, "Sharpening the 20 m bands of sentinel-2 image using an unsupervised convolutional neural network," in *2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2021, pp. 2875–2878.

(c) H. V. Nguyen, M. O. Ulfarsson, J. R. Sveinsson, and M. Dalla Mura, "Sentinel-2 sharpening using a single unsupervised convolutional neural network with MTF-based degradation model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 6882–6896, 2021.

Chapter 4 is based on the following publications:

(a) H. V. Nguyen, M. O. Ulfarsson, J. R. Sveinsson, and M. D. Mura, "Hyperspectral super-resolution by unsupervised convolutional neural network and SURE," in *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2022, pp. 903-906.

(b) H. V. Nguyen, M. O. Ulfarsson, J. R. Sveinsson, and M. D. Mura, "Deep SURE for unsupervised remote sensing image fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-13, 2022.

CHAPTER 2

DEEP SPARSE, LOW-RANK AND SURE FOR UNSUPERVISED HYPERSPECTRAL IMAGE DENOISING

Hyperspectral image (HSI) denoising is addressed in this chapter. Two HSI denoising methods are proposed, which both use unsupervised convolutional neural networks.

2.1 Problem formulation and motivation

An HSI is represented as a 3-D array of size $h \times w \times d$ where the first two dimensions represent spatial space and the third dimension represents spectral space. For the ease of mathematical expression, an HSI is reshaped into either a vector by stacking a vectorized band on top of each other, or into a matrix where each column is a vectorized band. The denoising problem is to estimate a clean HSI, $\mathbf{X} \in \mathbb{R}^{N \times d}$, having $N = h \times w$ pixels and d bands, which is related to the observed noisy HSI, $\mathbf{Y} \in \mathbb{R}^{N \times d}$, via the observation model given by

$$\mathbf{Y} = \mathbf{X} + \mathbf{N}, \tag{2.1}$$

where $\mathbf{N} = [\mathbf{n}_i]_{i=1}^{d}$ is an $N \times d$ matrix whose each column contains vectorized additive noise for corresponding band, i.e., $\mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \sigma_i^2 \mathbf{I}),\ i = 1, \ldots, d$, where σ_i is the noise standard deviation for the ith band.

The main motivation for developing new HSI denoising methods is to take advantages of both modern DL and traditional model-based methods. Recent developments in DL show that the structure of a CNN provides an implicit image prior called deep image prior (DIP) [63] which can be used to to solve the ill-posed inverse problem in image reconstruction. The reason behind the success of DIP is that a CNN provides higher impedance to an unstructured image (e.g., noise) than an image. It means that the CNN tends to fit an image faster than noise during its optimization. Early stopping the optimization iterations can yield a good result. In the context of HSI denoising using DIP, the reconstructed HSI $\hat{\mathbf{X}} \in \mathbb{R}^{N \times d}$ is obtained by early stopping optimization using the following loss function [60]

$$\mathcal{L}_{\text{DIP}}(\boldsymbol{\theta}) = \|\mathbf{Y} - f_{\boldsymbol{\theta}}(\mathbf{Y})\|_2^2, \tag{2.2}$$

where $\hat{\mathbf{X}} = f_{\boldsymbol{\theta}}(\mathbf{Y})$ is the output of the CNN with parameters $\boldsymbol{\theta}$. The optimization problem (2.2) is similar to the one in the model-based methods and is unsupervised,

because it does not require any ground truth and the DIP implicitly acts as a regularizer. However, the limitation of DIP is that its performance depends on the CNN structure and is prone to overftting. To maximize the use of DIP, recent works have strengthened the DIP by incorporating with explicit prior [51], designing better CNN structure [128], and other works have focused on reducing overfitting [141].

The work in this thesis falls into the same the above-mentioned directions, i.e., strengthening the DIP and overfitting avoidance, with focusing on HSI denoising. Since HSI is a kind of image having high correlation between bands, it induces the low-rank and sparsity. The first denoising technique exploits the sparse and low-rank by transforming the data (i.e., using singular value decomposition (SVD) and two dimensional discrete wavelet transform (2-D DWT)) and incorporating them with a CNN. The second denoising technique optimizes a CNN using a loss function based on the SURE which is an unbiased estimate of the MSE between the denoised and the ground truth HSI. The interesting point is that the unbiased MSE estimation requires no ground truth, thus the method is unsupervised. More importantly, the method optimizes the (estimated) MSE with respect to the ground truth, it significantly avoids overfitting. Experimental results demonstrate that the methods are efficient and outperform the competitive methods in both simulated and real HSI datasets.

2.2 Deep sparse and low-rank for HSI denoising

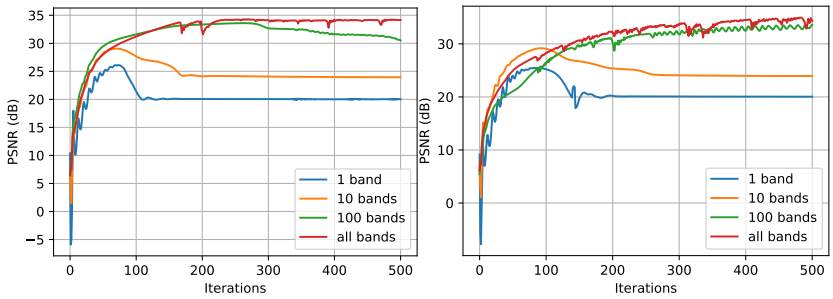
This section presents an HSI denoising method (called DIP-SLR) based on DIP and sparse and low-rank prior. First, with HSI data, DIP is experimentally justified that it implies sparse and low-rank prior. Second, the sparsity and low-rank are incorporated with DIP by transforming the HSI data to leverage the performance of DIP and accelerate the method. Finally, experiments are performed to verify the efficiency of the proposed method. Codes of the proposed DIP-SLR method are available at `https://github.com/hvn2/DIP-SLR`.

2.2.1 Deep sparse and low-rank prior in HSI

The DIP implicitly implies correlation in both spectral and spatial domain for HSIs. This claim is justified experimentally by using the same CNN in [60] with different inputs and outputs for the Pavia University (PU) and Washington DC Mall (DC) datasets (refer to Appendix for datasets and evaluation metrics) as follows:

- One band input and one band output, i.e., $\hat{\mathbf{X}}_{:,1} = f_{\boldsymbol{\theta}}(\mathbf{Y}_{:,1})$
- Ten bands input and ten bands output, i.e., $\hat{\mathbf{X}}_{:,1:10} = f_{\boldsymbol{\theta}}(\mathbf{Y}_{:,1:10})$
- One hundred bands input and one hundred bands output, i.e., $\hat{\mathbf{X}}_{:,1:100} = f_{\boldsymbol{\theta}}(\mathbf{Y}_{:,1:100})$
- All bands input and all bands output, i.e., $\hat{\mathbf{X}} = f_{\boldsymbol{\theta}}(\mathbf{Y})$

The results are shown in Fig. 2.1 in terms of peak-signal-to-noise ratio (PSNR) as a function of iterations for both PU and DC datasets. In the case where only one band



(a) PU *(b)* DC

Figure 2.1. DIP for different number of bands.

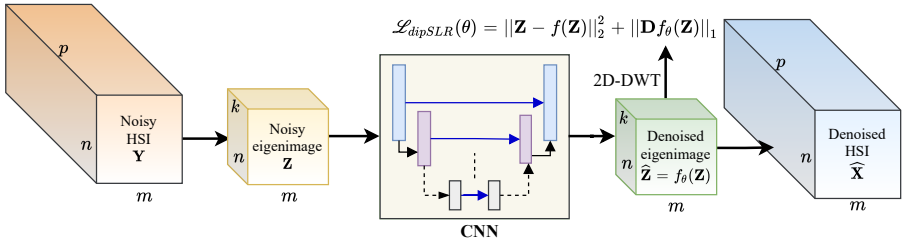


Figure 2.2. DIP-SLR framework.

is used at the network input and output, it means that there is no spectral correlation, the CNN gives the worst results. For example, in Fig. 2.1a, the PSNR curve reaches a peak at 80 iterations and decreases quickly after that indicating overfitting. The results are improved when more bands are used at the network input and output. Therefore, the high correlation in the spectral-spatial domain of HSI which promotes low-rank property significantly improves the DIP performance.

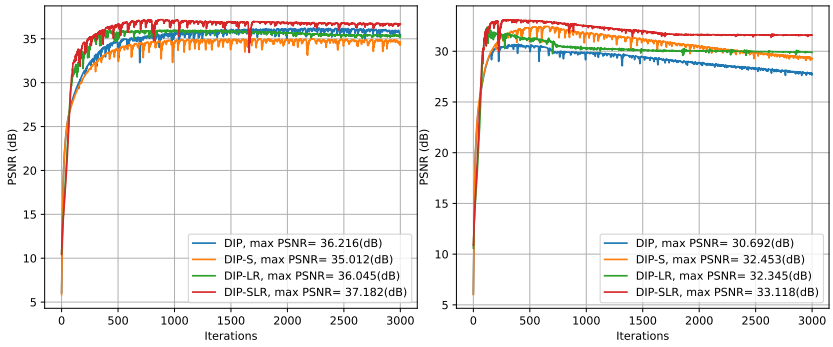
Although the DIP implies low-rank property obtained by using all HSI bands for the CNN, one of its downside is the computational burden. To overcome this downside, an explicit rank reduction method (i.e., SVD) is applied to a noisy observed HSI and obtained

$$\mathbf{Y} \approx \mathbf{Z}\mathbf{E}^T,$$

where $\mathbf{E}$ is a $d \times k$ ($k \ll d$) matrix, and $\mathbf{E}^T\mathbf{E} = \mathbf{I}$. The $N \times k$ matrix called "eigenimage" is computed as $\mathbf{Z} \approx \mathbf{Y}\mathbf{E}$. Replacing the high dimensional HSI, $\mathbf{Y}$, by the low dimensional eigenimage, $\mathbf{Z}$, not only retains the low-rank property but also significantly reduces the running time. In addition, it is observed that the eigenimages have local correlation [43] by which the coefficients are sparse by using a transform (e.g., 2-D DWT). Thus, incorporating the spare prior by using the wavelet transform to the eigenimage, the final loss function to optimize the CNN is given by

$$\mathcal{L}_{\text{DIP-SLR}}(\boldsymbol{\theta}) = \|\mathbf{Z} - f_{\boldsymbol{\theta}}(\mathbf{Z})\|_2^2 + \lambda\|\mathbf{D}f_{\boldsymbol{\theta}}(\mathbf{Z})\|_1, \tag{2.3}$$

where $\mathbf{D}$ is the forward 2-D DWT basis, and λ is a positive scalar controlling the



(a) DC ($\sigma = 0.1$) *(b)* DC ($\sigma = 0.2$)

Figure 2.3. Denoising results for the DC dataset using DIP, DIP-S, DIP-LR, and DIP-SLR shown as PSNR (dB) as a function of iterations.

contribution of sparsity. The denoised HSI is obtained at the output of the CNN, i.e., $\hat{\mathbf{X}} = f_{\boldsymbol{\theta}}(\mathbf{Z})\mathbf{E}$.

General framework of the proposed method is depicted in Fig. 2.2. There, the CNN is an hourglass with skip connection architecture as used in [60], [63].

2.2.2 Experimental results

Datasets and evaluation metrics

Both simulated and real datasets are used to evaluate the proposed method. The simulated datasets are the Pavia University (PU) and Washington DC Mall (DC). The real dataset is the Indian Pines (IP) dataset. Those datasets are described in detail in the Appendix.

To simulate the noisy data, two parts are cropped from the PU and DC datasets resulting in $400 \times 200 \times 103$ and $400 \times 200 \times 191$ data cubes for the PU and DC datasets, respectively. Those data cubes are considered as the clean HSIs. The noisy HSIs are generated by adding Gaussian noise to all bands. Here, we address only the identical independent isotropic noise, i.e., $\sigma_i^2 = \sigma^2,\ i = 1, ..., d$. The clean PU data cube is shown as a false color image in Fig. 2.4a. In the simulated case where the references are available, the following metrics are used: The peak-signal-to-noise-ratio (PSNR) in decibels, the mean structural similarity index (MSSIM) and the spectral angle mapper (SAM) in degrees. The Appendix describes all the metrics in detail. For the real IP dataset, denoising results are evaluated visually due to the lack reference image. For all datasets, the intensity values are normalized between 0 and 1 band-by-band before further processing.

Evaluation of sparse and low-rank with DIP for HSI denoising

Here, the performance of sparse and low-rank prior incorporating with DIP are examined by an ablation study using different configurations. Those configurations are listed below:

- DIP: Only the DIP is used, the loss function to be optimized is (2.2).
- DIP-S: The DIP, and sparse prior are used, the loss function to be optimized is $\mathcal{L}_{\text{DIP-S}}(\boldsymbol{\theta}) = \|\mathbf{Y} - f_{\boldsymbol{\theta}}(\mathbf{Y})\|_2^2 + \eta\|\mathbf{D}f_{\boldsymbol{\theta}}(\mathbf{Y})\|_1$, where the balance parameter η is a positive number.
- DIP-LR: The DIP, and low-rank prior are used, the loss function to be optimized is $\mathcal{L}_{\text{DIP-LR}}(\boldsymbol{\theta}) = \|\mathbf{Z} - f_{\boldsymbol{\theta}}(\mathbf{Z})\|_2^2$.
- DIP-SLR (the proposed method): The DIP, sparse, and low-rank priors are used, the loss function to be optimized is (2.3).

Fig. 2.3 shows the denoising results for the DC datasets with $\sigma = 0.1$ and $\sigma = 0.2$ using DIP, DIP-S, DIP-LR, and DIP-SLR. For DIP-SLR, k and λ are manually chosen as in Table. 2.1; for DIP-S, $\eta = 0.07$ ($\sigma = 0.1$) and $\eta = 0.1$ ($\sigma = 0.2$); and for DIP-LR, $k = 8$ ($\sigma = 0.1$) and $k = 5$ ($\sigma = 0.2$). It is clear that DIP-SLR gives higher PSNR than DIP, DIP-S, and DIP-LR. For higher noise level, DIP and DIP-S overfit quickly since the PSNR curves decrease at higher iterations (see Fig. 2.3b). Overfitting is reduced in DIP-SLR, because it uses low-rank approximation which can be considered as a pre-denoising process.

Comparison with other HSI denoising methods

This section shows the comparison of the proposed method (DIP-SLR) against the HSI denoising methods which are based on sparse, and low-rank priors. The first method is the 3-D Wavelet [26] which is a wavelet shrinkage denoising method using a 3-D undecimated DWT. The second method is the HSI denoising method via noise-adjusted iterative low-rank matrix approximation (NAILRMA) [40]. The parameters of the 3-D Wavelet and NAILRMA methods are set as default values, while the hyperparameters for DIP-SLR (i.e., k and λ) are listed as in Table 2.1.

Table 2.1. Hyperparameter setting for DIP-SLR.

Parameters	DC		PU	
σ	0.1	0.2	0.1	0.2
λ	0.25	0.55	0.2	0.5
k	8	6	4	4

For the simulated datasets, the quantitative and qualitative results are given in Tables 2.2 and 2.3 and Fig. 2.4, respectively. Note that the results for DIP-SLR in Tables 2.2 and 2.3 are obtained based on the highest PSNR during 3000 iterations. In the real applications, there is no criteria for stopping the iterations, and the results have to be observed based on empirical visualization. The results in Table 2.2 and Table 2.3 highlight that DIP-SLR outperforms the competitive methods in all evaluation metrics

for all datasets. Visual judgment of the denoised images for the PU dataset ($\sigma = 0.2$) using all methods pictured in Fig. 2.4 reveals that 3-D Wavelet and NAILRMA cannot completely remove the noise, while DIP-SLR gives good results and looks similar to the reference image.

For the real IP dataset, the same setting for all methods as in the simulated case are used, and the stopping iteration for DIP-SLR is manually chosen as 1000 iterations by monitoring the denoised image visually. The restored images are shown in Fig. 2.5. Apparently, noise is still visible in the 3-D Wavelet results. Both NAILRMA and DIP-SLR successfully remove the noise and give clean denoised images. However, NAILRMA cannot remove the dead pixel noise that exists in bands 105 and 220. Running times for all methods are also given in Fig. 2.5 where the fastest to slowest methods are ranked as: 3-D Wavelet, DIP-SLR, and NAILRMA. Note that 3-D Wavelet and NAILMA were implemented in Matlab using a Linux computer of 16 GB of RAM, 3.2 GHz of Intel Core I7-6900K CPU, and DIP-SLR was implemented using the same computer and was run in GPU mode (12 GB Nvidia TitanX).

Table 2.2. Quantitative results for denoising PU dataset. The uparrow (↑) means that higher values are better, the downarrow (↓) means that the lower values are better. Best results are in bold.

Dataset	Noise std. dev.	Metrics	Noisy	3-D Wavelet	NAILRMA	DIP-SLR
PU	0.1	PSNR (dB)↑	20.00	33.82	34.99	**36.23**
		MSSIM↑	0.324	0.911	0.917	**0.943**
		SAM (°)↓	29.185	5.597	4.672	**3.853**
	0.2	PSNR (dB)↑	13.97	30.18	30.18	**32.86**
		MSSIM↑	0.127	0.813	0.803	**0.889**
		SAM (°)↓	46.732	7.817	7.084	**4.916**

Table 2.3. Quantitative results for denoising the DC dataset. The uparrow (↑) means that higher values are better, the downarrow (↓) means that the lower values are better. Best results are in bold.

Dataset	Noise std. dev.	Metrics	Noisy	3-D Wavelet	NAILRMA	DIP-SLR
DC	0.1	PSNR (dB)↑	20.00	31.47	36.2	**37.02**
		MSSIM↑	0.361	0.888	0.941	**0.965**
		SAM (°)↓	20.969	5.467	2.806	**2.428**
	0.2	PSNR (dB)↑	13.97	28.03	31.22	**33.67**
		MSSIM↑	0.155	0.789	0.862	**0.937**
		SAM (°)↓	37.080	7.877	4.736	**3.498**

(a) Reference *(b)* Noisy *(c)* 3-D Wavelet *(d)* NAILRMA *(e)* DIP-SLR

Figure 2.4. Denoising results for the PU dataset ($\sigma = 0.2$). The bottom parts (200×200 pixels) are shown in false color images using bands 57, 27, and 17.

2.3 Deep SURE-based unsupervised HSI denoising

One of the limitation of the DIP is overfitting, because optimization of the loss function (2.2) until convergence results in an approximation of the input (noisy image) at the output. To overcome this problem, a new loss function which is derived by SURE is proposed. In HSI denoising, SURE is an unbiased estimate of the MSE between the estimated and the ground truth images, and is computed by using only the noisy image. Therefore, optimization a CNN with the SURE-based loss function is not only unsupervised but also avoids overfitting. Moreover, the proposed denoising method based on SURE and unsupervised CNN is capable to remove the non-Gaussian noise, and works with reduce-rank data to reduce the running time. In the following, the proposed method is called SURE-CNN for short. Codes of the SURE-CNN method are available at `https://github.com/hvn2/HSI_Denoising_SURE_CNN`.

2.3.1 The SURE-based unsupervised CNN HSI denoising method

THE SURE LOSS FUNCTION

For the ease of mathematical formulation, in this section, images are represented as vectors. The model (2.1) is rewritten in vector form as

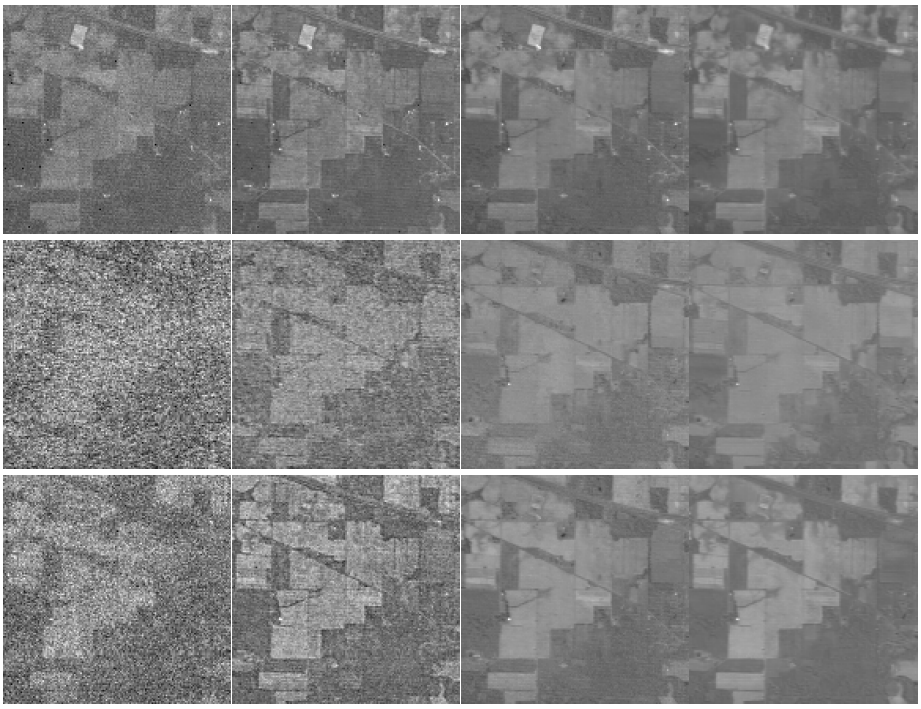
$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \tag{2.4}$$

where $\mathbf{x} \in \mathbb{R}^{Nd \times 1}$ and $\mathbf{y} \in \mathbb{R}^{Nd \times 1}$ are clean and noisy HSIs, respectively. The additive noise is represented by $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Omega})$, where $\mathbf{\Omega}$ is a block diagonal matrix with the ith diagonal element is $\sigma_i^2 \mathbf{I}$.

As discussing above, a DIP denoising method using the loss function (2.2) is prone to overfitting. To avoid overfitting, a loss function should involve with the ground truth such as,

$$R = \mathbb{E}\|\mathbf{x} - f_{\boldsymbol{\theta}}(\mathbf{y})\|_2^2, \tag{2.5}$$

where $\mathbb{E}$ is the expectation. However, $\mathbf{x}$ is not observed and therefore R is not computable. To make use of (2.5), R is exchanged to its unbiased estimate, $\hat{R}$, using



(a) Observation *(b)* 3-D Wavelet (**3.4** s) *(c)* NAILRMA (68.7 s) *(d)* DIP-SLR (27.3 s)

Figure 2.5. Denoising results for the IP dataset. Top to bottom rows are the observed noisy and denoised bands 2, 105 and 220. The numbers given in brackets are the running time in seconds. Smallest running time is bold.

SURE [137], given by

$$\hat{R} = \mathbb{E}\big[\|\mathbf{y} - f_{\boldsymbol{\theta}}(\mathbf{y})\|_2^2 + 2\mathrm{tr}\big(\boldsymbol{\Omega}\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{y})}{\partial \mathbf{y}}\big) - \mathrm{tr}(\boldsymbol{\Omega})\big], \tag{2.6}$$

where $\mathrm{tr}(\cdot)$ is the trace of a square matrix. For any variable $\mathbf{z}$, the unbiased estimate of $\mathbb{E}[\mathbf{z}]$ is $\mathbf{z}$. Therefore, the SURE-based loss function is obtained by dropping the expectation in (2.6), given by

$$\mathcal{L}_{\mathrm{SURE}}(\boldsymbol{\theta}) = \|\mathbf{y} - f_{\boldsymbol{\theta}}(\mathbf{y})\|_2^2 + 2\mathrm{tr}\big(\boldsymbol{\Omega}\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{y})}{\partial \mathbf{y}}\big) - \mathrm{tr}(\boldsymbol{\Omega}). \tag{2.7}$$

Training a CNN with the loss function (2.7) not only avoids overfitting but also is unsupervised since it does not require any ground truth.

Computation of the network divergence (i.e., the trace of the CNN output Jacobian matrix $\mathrm{tr}\big(\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{y})}{\partial \mathbf{y}}\big)$) is hard, because the CNN is a non-linear function and there is not available a close form for the Jacobian matrix. Recent DL frameworks, such as Tensorflow and Pytorch, are able to compute the exact value of the network divergence,

but it is very inefficient in terms of memory and running time [142]. This limits the SURE loss function in practice, since the network divergence is needed to compute for every training iteration. A solution for fast computing the network divergence is to use the Monte-Carlo SURE (MC SURE) approximation [143] as follows,

$$\text{tr}\big(\mathbf{\Omega}\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{y})}{\partial \mathbf{y}}\big) \approx \mathbf{b}^T\mathbf{\Omega}\frac{f_{\boldsymbol{\theta}}(\mathbf{y}+\beta\mathbf{b}) - f_{\boldsymbol{\theta}}(\mathbf{y})}{\beta},$$

where $\mathbf{b}$ is a vector drawn from Gaussian distribution with zero mean and unit variance, and β is a small number. Integrating the MC SURE approximation of the network divergence to (2.7), the final SURE-based loss function is

$$\mathcal{L}_{\text{SURE}}(\boldsymbol{\theta}) = \|\mathbf{y} - f_{\boldsymbol{\theta}}(\mathbf{y})\|_2^2 + 2\mathbf{b}^T\mathbf{\Omega}\frac{f_{\boldsymbol{\theta}}(\mathbf{y}+\beta\mathbf{b}) - f_{\boldsymbol{\theta}}(\mathbf{y})}{\beta} - \text{tr}(\mathbf{\Omega}). \tag{2.8}$$

In a real application, the noise standard deviation σ_i is unknown. It is estimated using the HySime algorithm [144] or the median absolute deviation estimator in the highest subband (HH) of the wavelet transform of each band as in [145]

$$\widehat{\sigma}_i = \frac{\text{median}\big(\left|\mathbf{W}_{(i)}^{HH}\right|\big)}{0.6745}, i = 1, \ldots, p \tag{2.9}$$

Extension of SURE for non-Gaussian noise removal and for subspace-based denoising

Theoretically, the assumption under SURE is Gaussian distribution. However, SURE-CNN can be extended to deal with non-Gaussian noise such as Poissonian noise. For Poissonian noise, the noisy data are approximated to Gaussian one [43] by applied the Anscombe transform [146]. Then, the approximated Gaussian noisy data are denoised using the SURE-CNN method and the results are inverse-Anscombe transformed to obtain the final denoised HSI. Algorithm 1 below details the steps of SURE-CNN applied for Poissonian noise removal.

Algorithm 1: Poissionian noise removal using SURE-CNN.

1. **Input:** Additive Poissionian noisy HSI $\mathbf{Y}$.
2. Approximate Gaussian noise from Poissionian noise by applying Anscombe transform, $\tilde{\mathbf{Y}} = 2\sqrt{\mathbf{Y} + \frac{3}{8}}$.
3. Denoise $\tilde{\mathbf{Y}}$ using SURE-CNN, $\tilde{\mathbf{Z}} = f_{\boldsymbol{\theta}}(\tilde{\mathbf{Y}})$.
4. **Output:** Inverse Anscombe transform to obtaine denoised image, $\widehat{\mathbf{X}} = \big(\frac{\tilde{\mathbf{Z}}}{2}\big)^2 - \frac{3}{8}$.

The SURE-CNN HSI denoising method works with big volumetric data, and it may run slowly. To accelerate the method, the noisy HSI is reduced its dimension by

utilizing a dimensionality reduction technique, e.g., the SVD. For a noisy HSI in matrix form, $\mathbf{Y}$, we can write (by using SVD)

$$\mathbf{Y} \approx \mathbf{Z}\mathbf{E}^T,$$

where $\mathbf{E} \in \mathbb{R}^{d\times k}(k \ll d)$ is a matrix containing the orthonormal basis that spans $\mathbf{Y}$ (i.e., $\mathbf{E}^T\mathbf{E} = \mathbf{I}_k$). The matrix $\mathbf{Z}$ (called eigenimage) are the coefficients of $\mathbf{Y}$ represented in $\mathbf{E}$, and are computed as $\mathbf{Z} \approx \mathbf{Y}\mathbf{E}$. Since $\mathbf{Z}$ is the result of a linear transformation of $\mathbf{Y}$, it also follows a Gaussian distribution with zero mean and covariance of $\mathbf{E}\mathbf{\Omega}\mathbf{E}^T$. Then, the SURE-CNN denoising method is applied to the eigenimage. Note that the eigenimage has a low (spectral) dimension of $k \ll d$, therefore the running time is significantly reduced. Finally, the the denoised eigenimage is transformed back to the original HSI space to obtain the noise removal HSI. The above-mentioned procedure is call as the subspace SURE-CNN and is described in Algorithm 2.

Algorithm 2: The subspace SURE-CNN HSI denoising algorithm.

1. **Input:** Noisy HSI $\mathbf{Y}$, dimension of subspace k.
2. Find a subspace base $\mathbf{E}$ using SVD, compute noisy eigenimage, $\mathbf{Z} = \mathbf{Y}\mathbf{E}$.
3. Denoise noisy eigenimages using SURE-CNN, $\widehat{\mathbf{Z}} = f_\theta(\mathbf{Z})$.
4. **Output:** Transform denoised eigenimage to denoised HSI, $\widehat{\mathbf{X}} = \widehat{\mathbf{Z}}\mathbf{E}^T$.

2.3.2 CNN architecture

Inspired by the success of the skip-connection CNN [60], [63] which was proved to provide strong DIP, the CNN used with SURE for HSI denoising is also a kind of skip-connection CNN. Fig 2.6 shows the network structure. It consists of three main parts: The encoder part, the decoder part, and the skip connection part. The encoder part is composed of K blocks of convolutional and LeakyReLU (*conv-relu*) layers are stacked consecutively. The decoder part is also composed of K block of *conv-relu* layers and one convolutional layer at the output. In each block of the encoder and decoder, a convolutional layer has 128 filters, a filter size of 3, and a stride of 1. The last convolutional layer in decoder has d filters, the filter size and strides of 1, and and is followed by the Sigmoid activation function. The encoder extracts feature maps of the input image and merges those feature maps with the decoder via the skip connections. Each skip connection is a convolutional layer with F filters, and the filter size, stride, and activation function are the same as the those in the convolutional layer of the encoder and decoder. Denoised HSI image is obtained at the output of the last layer in the decoder. The proposed network is different from the networks in [60], [63] since the downsampling, upsampling and the batchnormalization layers are not used. Instead, the spatial size of the HSI is kept constant throughout the network. Although using the downsampling and upsampling layers narrow down the network scale and reduce running time but they may lose information, and the batchnormalization layers do not

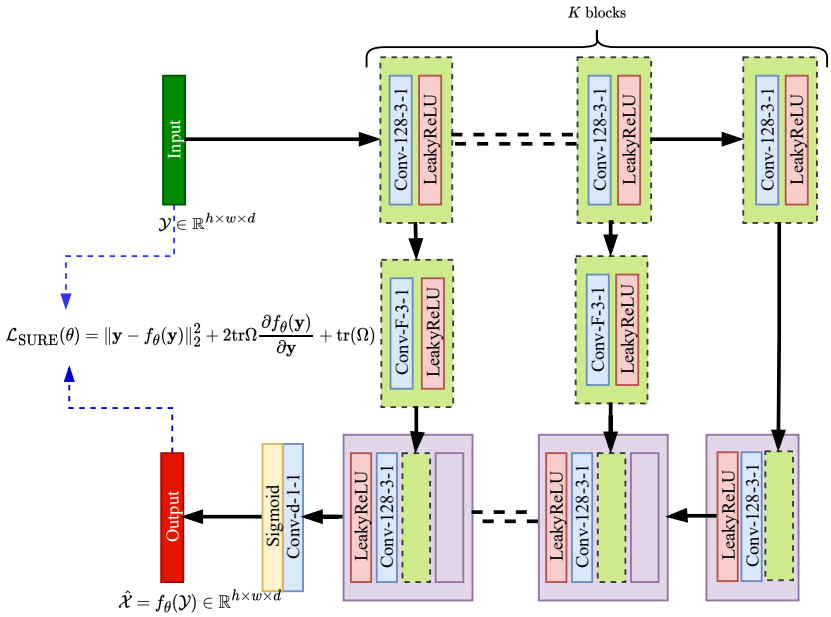


Figure 2.6. SURE-CNN network structure. Conv-128-3-1 represents a convolutional layer with 128 filters of kernel size 3 and stride 1. The number of conv-relu blocks is $K = 5$, and the number of filters in a skip connection layer is $F = 5$.

improve the result. The network is optimized with the SURE loss as in (2.8) using the Adam [136] optimizer with a learning rate of 0.001, and is implemented in Tensorflow 2.0.

The network depth, i.e., the numbers of *conv-relu* blocks, K, and the numbers of filters, F, in the skip connection layers, directly influences to the DIP and the results. To find a network depth which best suits to the HSI denoising problem, an experiment with the PU dataset has been conducted. In this experiment, the values of PSNR during training are monitored with respect to different values of K and F. Fig. 2.7a and Fig. 2.7b show the PSNR as a function of training iterations for $K \in \{3,\ 5,\ 7\}$ and $F = 5$, and for $F \in \{5,\ 25,\ 64\}$ and $K = 5$, respectively. Table 2.4 gives the PSNR for the networks using various values of K and F, for different cases of noise. By addressing those parameters, the network with five *conv-relu* blocks ($K = 5$) and five filters ($F = 5$) in the skip connection is chosen, since it gives the best trade-off between PSNR and the network complexity.

2.3.3 Experimental results

Datasets and evaluation metrics

The datasets used in the experiments are both simulated and real datasets. The simulated datasets are the PU and DC datasets, and the real datasets are the IP and the

Table 2.4. The SURE-CNN performance given by PSNR in dB for different conv-relu blocks K and number of filters F in the skip connection layers for various cases of noise. The results are the average values over 10 runs.

PSNR (dB)		Case 1: $\sigma = 0.3$	Case 2: $\sigma = 1$, $\eta = 20$	Case 3: $\sigma \sim \mathcal{U}(0.1, 0.2)$
$F = 5$	$K = 3$	30.90	37.23	34.94
	$K = 5$	31.45	37.34	35.29
	$K = 7$	31.50	37.03	35.37
$K = 5$	$F = 5$	31.45	37.34	35.29
	$F = 25$	30.90	36.89	35.05
	$F = 64$	30.67	36.93	34.78

Urban (UB) datasets. Those datasets are detailed in the Appendix. The DC and PU datasets are assumed to be noise-free HSIs. Small parts of $400 \times 200 \times 191$ for the DC dataset and $400 \times 200 \times 103$ for the PU dataset, respectively, are used to simulate the noisy HSIs by adding noise. The intensity values are normalized band-by-band between 0 and 1 before adding the noise. The noisy simulated datasets are created as follows:

1. Case 1: Isotropic Gaussian noise with zero mean and standard deviation σ is added to each band, i.e., $\sigma_i^2 = \sigma^2$. We consider the following values of standard deviation, $\sigma \in \{0.05,\ 0.1,\ 0.2,\ 0.3\}$.

2. Case 2: Band-wise Gaussian noise with zero mean and variance that varies according to a bell shape is added to each band. The variance varies according to

$$\sigma_i^2 = \sigma^2 \frac{e^{-\frac{(i-d/2)^2}{2\eta^2}}}{\sum_{j=1}^{d} e^{-\frac{(i-d/2)^2}{2\eta^2}}}, \tag{2.10}$$

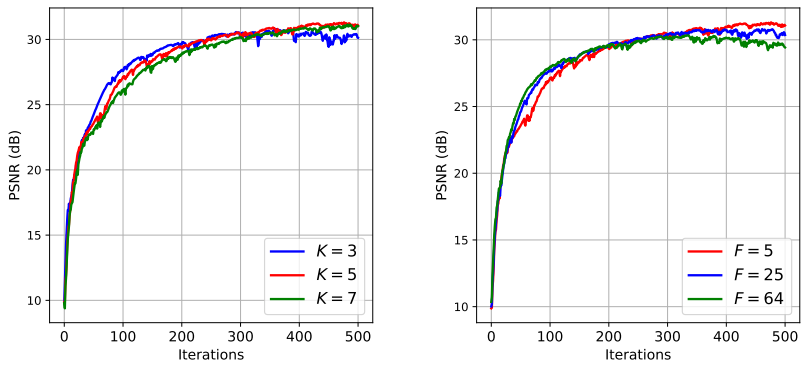
   where $\sigma = 1$ and $\eta = 20$ are two numbers that control the noise intensity and the bell width, respectively.

3. Case 3: A zero mean Gaussian noise is added to each band. The standard deviation for each band is drawn from a uniform distribution between 0.1 and 0.2 ($\sigma_i \sim \mathcal{U}(0.1, 0.2)$).

The real datasets (IP and UB) are contaminated by various types of noise, such as Gaussian noise, Poissonian noise, stripped noise, and missing pixels noise.

To evaluate the denoising performance in the simulated case where the references are available, the same metrics used in Section 2.2.2 (the PSNR, MSSIM and SAM) are used. In the real datasets case, only visually inspection is concerned since there is no ground truth.

PARAMETERS VALIDATION



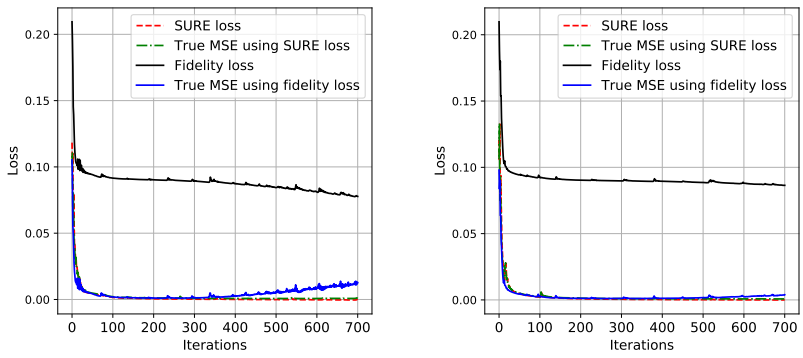
(a) $K \in \{3,\ 5,\ 7\}$, and $F = 5$ *(b)* $F \in \{\ 5,\ 25,\ 64\}$, and $K = 5$

Figure 2.7. PSNR as a function of iterations by SURE-CNN for different numbers conv-relu blocks K, and number of filters F in the skip connection layers. The results are the average values over 10 runs.

This section validates that SURE is an unbiased estimate of the MSE and training the CNN with SURE loss function avoids overfitting. Fig. 2.8 shows the training losses and the true MSEs as functions of training iterations for both PU and DC datasets in denoising Case 1 with $\sigma = 0.3$. In these experiments, the CNN structure as in Fig. 2.6 is used. The training losses are the SURE and the DIP (fidelity) losses, and the true MSEs are computed between the denoised and ground truth HSIs. It can be seen that the SURE loss approximates almost perfectly the true MSE while the DIP loss fails to do so. Furthermore, the network that uses DIP loss tends to overfit at 300 iterations (for PU) and 400 iterations (for DC), because the training loss decreases while the true MSE starts increasing. Therefore, training the network with DIP loss needs a validation scheme, such as early stopping, to reach an optimal point. However, in a real application it is hard to choose an optimal stopping point due to the lack of ground truth. On the other hand, the network using SURE loss does not overfit, thus it does not require early stopping. This makes the SURE-CNN method more feasible in practice.

Next, the SURE loss with noise standard deviation σ_i estimated by the wavelet (2.9) and the HySime algorithms are evaluated. Fig. 2.9a and Fig. 2.9c depict the true σ_i in (2.10) and their estimated values by wavelet and HySime algorithms for the PU and DC datasets, respectively. Fig. 2.9b and Fig. 2.9d show the SURE loss using σ_i and $\widehat{\sigma}_i$ with the CNN sketched in Fig. 2.6, for denoising Case 2 of the PU, and DC datasets, respectively. It is observed from these figures that both wavelet and HySime give a relatively good estimate of the true σ_i, and there is almost no difference in the SURE loss between using true σ_i and estimated $\widehat{\sigma}_i$.

Furthermore, the performance of different CNN architectures using SURE loss is assessed. Fig. 2.10 shows the SURE loss and the true MSE during training using the modified UNet [63] and the ResNet [112] for denoising of the PU and DC datasets in Case 1 with $\sigma = 0.3$. The numbers of input and output channels for both the UNet and ResNet are changed to fit with HSI data. For the UNet, the number of filters in the skip



(a) The PU dataset *(b)* The DC dataset

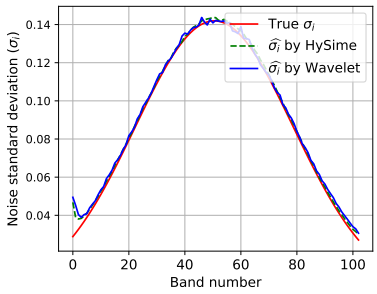
Figure 2.8. The true MSE, SURE and DIP (fidelity) loss for denoising Case 1 with $\sigma = 0.3$.

connections is 4. The number of residual blocks in the ResNet is 5. It is clearly verified that the SURE loss works well for both UNet and ResNet structures. The PSNR, for each dataset and network, shows that the UNet gives approximately 2 dB higher than the ResNet for the same number of training iterations.

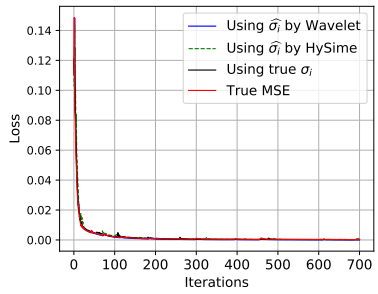
Performance Evaluation using Simulated Datasets

Here, with the simulated DC and PU datasets, the performance of SURE-CNN and the competitive methods are presented by using quantitative metrics and visualization of the denoised images. SURE-CNN is compared to the following competitive methods: Two HSI denoising method based on tensor decomposition that are low-rank tensor approximation (LRTA) [37], and tensor learning dictionary (TDL) [38], an HSI denoising method using the first-order spectral roughness penalty in the wavelet domain (FORPDN) [28], the nonlocal transform-domain filter for volumetric data (BM4D) [27], an HSI denoising method via noise-adjusted iterative low-rank matrix approximation (NAILRMA) [40], an HSI restoration technique using sparse and low-rank model (HyRes) [42], and the fast HSI denoising and inpainting based on low-rank and sparse representation (FastHyDe) [43]. All the competitive methods use the parameters recommended in the corresponding papers. The proposed SURE-CNN method uses the CNN architecture as shown in Fig 2.6 ($K = 5$ and $F = 5$), and the noise standard deviation $\widehat{\sigma}_i$ is estimated by the wavelet algorithm (2.9).

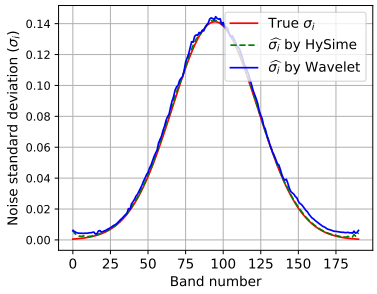
The denoising results are given in Table 2.5 and Table 2.6 in terms of PSNR, MSSIM, and SAM for the DC and PU datasets, respectively. The PSNR and MSSIM of SURE-CNN are higher than other methods in most cases. For high noise levels, such as in Case 1 with $\sigma \in \{0.3,\ 0.2,\ 0.1\}$, and in Case 3, SURE-CNN outperforms all the competitive methods. The margins are nearly 0.4 dB of PSNR, and a few percents of MSSIM in comparison with the second best method, FastHyDe. However, FastHyDe gives the best PSNR in Case 1 with $\sigma = 0.05$ for both DC and PU datasets; and HyRes gives the best PSNR and MSSIM in Case 2 for the DC dataset.



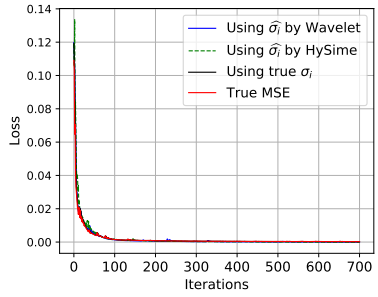
(a) σ_i and $\widehat{\sigma}_i$ for the PU dataset



(b) The SURE loss using σ_i and $\widehat{\sigma}_i$ for the PU dataset



(c) σ_i and $\widehat{\sigma}_i$ for the DC dataset



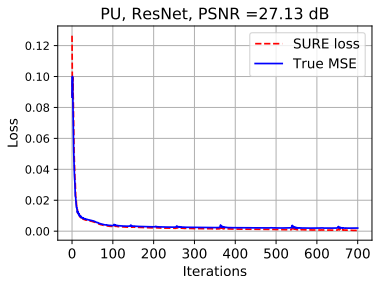
(d) The SURE loss using σ_i and $\widehat{\sigma}_i$ for the DC dataset

Figure 2.9. The true and estimated noise standard deviations, and the SURE loss using true and estimated noise standard deviation compared with the true MSE for denoising Case 2 with $\sigma = 1, \eta = 20$ *with the PU and DC dataset.*

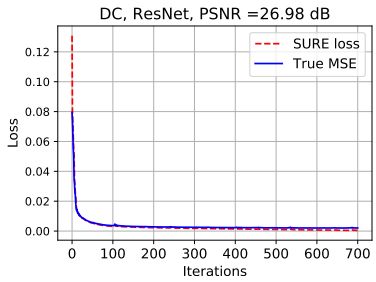
The denoising results for Case 1 ($\sigma = 0.3$), Case 2 and Case 3 are pictured to demonstrate the visual quality. Figs. 2.11, 2.12, and 2.13 show the denoised results of a sub-scene of band 60 of the DC dataset, while Figs. 2.14, 2.15, and 2.16 show the denoised results of a sub-scene of band 60 of the PU dataset. LRTA, TDL, FORPDN and NAIRLMA can not completely remove the noise as there is still noise in the denoised images. BM4D gives over-smooth denoised images. HyRes, FastHyDe, and SURE-CNN work well in filtering out the noise in all cases and provide clean denoised images. But, for denoising PU dataset Case 1 (Fig. 2.11), and Case 2 (Fig. 2.12), HyRes is slightly worse than FastHyDe and SURE-CNN.

Performance Evaluation using Real Datasets

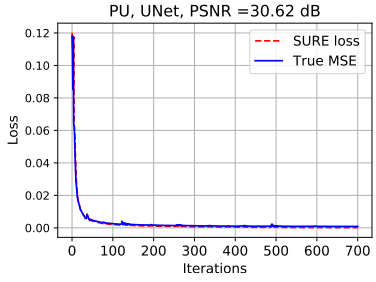
The results by means of visual perception of the denoised images for the real IP and UB datasets are shown here. To obtain the results, the proposed method and the competitive methods are run with the parameters set as in the simulated case. SURE-CNN are run 500 iterations for the IP dataset and 150 iterations for the UB dataset. Fig. 2.17 shows the denoising results for the IP dataset. Since the dominant noise in the IP dataset is Gaussian noise, all the denoising methods significantly remove the



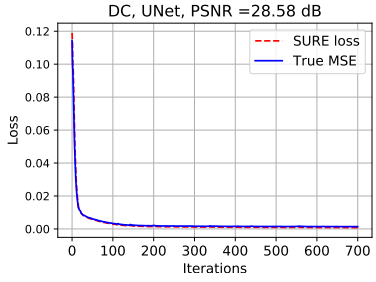
(a) ResNet, PU dataset



(b) ResNet, DC dataset



(c) Unet, PU dataset



(d) Unet, DC dataset

Figure 2.10. The SURE loss and true MSE of the ResNet and UNet for denoising Case 1 with $\sigma = 0.3$.

noise and improve the visual quality of the denoised images. However, FORPDN produces artifacts. The denoised images by BM4D are over-smooth. For the strongly contaminated noise band such as band 104, only SURE-CNN can recover the image from noise while all the other methods fail to remove the noise.

The denoised UB dataset is shown in Fig. 2.18. It is harder to remove the noise in the UB dataset because it contains mixed types of Gaussian and stripped noise. All the competitive methods can filter out the Gaussian noise, but they are unable to remove the stripped noise. It is easy to recognize that SURE-CNN can separate the images from all types of noise. Specifically, for very strongly noise affected bands such as bands 144 and 208, SURE-CNN yields very clean denoised images, in contrast to the competitive methods. It is also noticed that, for bands 108 and 139, which have low noise levels, FORPDN and NAILRMA also work well.

Performance evaluation of Subspace SURE-CNN and SURE-CNN for Poissionian noise removal

As discussed above, the SURE-CNN method applying to the reduced dimensional data can benefit the running time and may improve the result since the dimensionality reduction is a pre-denoising technique. To validate the claim, experiments are conducted using the simulated PU and DC datasets in Case 1 with $\sigma = 0.3$. The subspace is found by using SVD. The subspace SURE-CNN performance is evaluated in term of PSNR

Table 2.5. Denoising for the DC dataset using different methods. The evaluated metrics are PSNR in dB, MSSIM, and SAM in degrees. The results are the average values over 10 runs. The standard deviations for PSNR, MSSIM, and SAM in each method are less than 0.1 *dB,* 0.0008*, and* 0.03 *degrees, respectively. Best results are in bold.*

DC	Noise level	Metric	Noisy	LRTA	TDL	FORPDN	BM4D	NAILRMA	HyRes	FastHyDe	SURE-CNN
Case 1	$\sigma = 0.05$	PSNR↑	26.02	37.65	40.43	35.27	36.54	40.73	40.49	**41.73**	41.32
		MSSIM↑	0.628	0.962	0.981	0.946	0.955	0.977	0.979	0.983	**0.987**
		SAM↓	10.910	2.626	1.781	2.919	3.516	1.765	1.744	**1.506**	1.555
	$\sigma = 0.1$	PSNR↑	20.00	34.13	35.24	32.39	32.49	36.20	36.62	37.75	**37.89**
		MSSIM↑	0.3617	0.934	0.952	0.898	0.899	0.942	0.949	0.962	**0.973**
		SAM↓	20.977	3.595	3.005	4.521	4.881	2.816	2.645	**2.266**	**2.266**
	$\sigma = 0.2$	PSNR↑	13.98	30.18	30.41	29.39	28.78	31.23	32.15	33.98	**34.15**
		MSSIM↑	0.1549	0.867	0.870	0.818	0.792	0.862	0.888	0.924	**0.941**
		SAM↓	37.072	5.228	5.174	6.834	6.819	4.725	4.208	**3.321**	3.390
	$\sigma = 0.3$	PSNR↑	10.46	27.85	28.33	27.48	26.73	28.23	29.90	31.84	**32.07**
		MSSIM↑	0.081	0.799	0.808	0.740	0.698	0.793	0.834	0.892	**0.911**
		SAM↓	48.258	6.581	6.490	8.653	8.232	6.448	5.332	**4.281**	4.439
Case 2	$\sigma = 1, \eta = 20$	PSNR↑	22.81	24.809	24.59	34.19	34.06	38.60	**40.59**	39.75	39.86
		MSSIM↑	0.743	0.785	0.790	0.956	0.955	0.985	**0.990**	0.985	0.985
		SAM↓	15.515	12.392	12.663	3.829	3.956	2.074	**1.685**	1.876	1.808
Case 3	$\sigma \sim \mathcal{U}(0.1, 0.2)$	PSNR↑	16.52	24.87	29.49	30.59	30.37	33.52	34.57	35.64	**35.87**
		MSSIM↑	0.2475	0.594	0.803	0.853	0.844	0.906	0.925	0.942	**0.959**
		SAM↓	29.517	11.940	6.731	5.787	5.890	3.705	3.264	2.819	**2.816**

in decibels and running time in seconds. For the denoising Case 1 with $\sigma = 0.3$, the denoising process is run over 500 iterations, since the loss does not decrease more. The hardware is a Linux computer equipped with an Nvidia TitanX GPU of 12 GB memory. Fig. 2.19 shows the PSNR and running time of SURE-CNN in different subspace dimensions, and the full-rank dimension (i.e., do not use subspace). It is shown that the subspace SURE-CNN not only reduces the running time but also improves the PSNR over the full-rank HSI, such as $k_{sub} = 5$ gives PNSR $= 32.39$ dB for the PU dataset, and $k_{sub} = 10$ gives PSNR $= 32.65$ dB for the DC dataset. The results are pictured in Fig. 2.20 for bands 60 of both datasets.

Results for denoising Poissionian noise are shown in Fig. 2.21 using the PU dataset. SURE-CNN is compared against FastHyDe [43]. Both methods give good results and FastHyDe is better than SURE-CNN for lower noise case, but SURE-CNN outperforms FastHyDe in the higher noise case.

2.3.4 Further discussions

Further discussions on the comparison performance between SURE-CNN and the DL-based methods and the running time are presented in this section.

Comparison of SURE-CNN and the DL-based methods

The first method to be compared is a supervised DL denoising method, HSI-SDeCNN, proposed in [53]. The second method is an unsupervised DL denoising method, DIP-HSI, proposed in [60]. HSI-SDeCNN used a CNN which was trained

Table 2.6. Denoising for the PU dataset using different methods. The evaluated metrics are PSNR in dB, MSSIM, and SAM in degrees. The results are the average values over 10 runs. The standard deviations for PSNR, MSSIM, and SAM in each method are less than 0.06 *dB,* 0.0006*, and* 0.01 *degrees, respectively. Best results are in bold.*

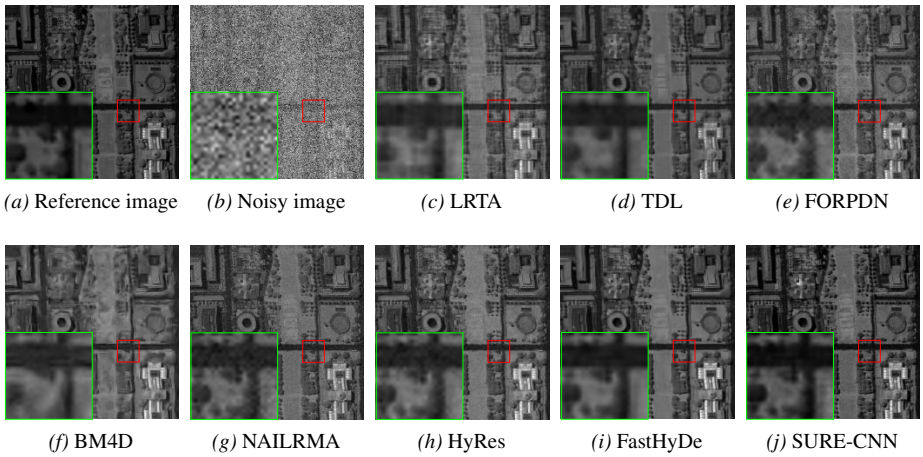
PU	Noise level	Metric	Noisy	LRTA	TDL	FORPDN	BM4D	NAILRMA	HyRes	FastHyDe	SURE-CNN
Case 1	$\sigma = 0.05$	PSNR↑	26.02	35.67	39.24	37.84	38.01	38.46	38.75	**39.62**	39.17
		MSSIM↑	0.593	0.922	0.967	0.955	0.957	0.959	0.963	**0.970**	0.969
		SAM↓	16.584	5.172	3.287	3.934	3.938	3.752	3.541	3.185	**3.149**
	$\sigma = 0.1$	PSNR↑	20.00	32.43	35.46	34.26	34.35	34.65	35.24	36.75	**37.12**
		MSSIM↑	0.320	0.863	0.935	0.904	0.914	0.910	0.925	0.950	**0.955**
		SAM↓	29.683	6.462	4.559	5.536	5.495	5.336	4.845	4.053	**3.643**
	$\sigma = 0.2$	PSNR↑	13.98	28.81	31.68	30.33	30.39	30.20	31.67	33.62	**33.91**
		MSSIM↑	0.125	0.741	0.868	0.795	0.815	0.800	0.852	0.913	**0.918**
		SAM↓	47.376	7.947	5.952	7.915	7.892	7.689	6.571	5.287	**4.695**
	$\sigma = 0.3$	PSNR↑	10.46	26.53	29.38	27.86	28.25	27.47	29.69	31.75	**31.91**
		MSSIM↑	0.062	0.629	0.799	0.689	0.723	0.694	0.790	**0.878**	**0.878**
		SAM↓	57.886	9.401	6.937	9.670	9.545	9.501	7.520	6.334	**5.769**
Case 2	$\sigma = 1, \eta = 20$	PSNR↑	20.13	25.67	26.81	35.06	34.83	35.49	36.32	37.19	**37.66**
		MSSIM↑	0.403	0.612	0.699	0.920	0.921	0.926	0.940	0.952	**0.957**
		SAM↓	29.304	16.609	14.081	5.326	5.235	4.875	4.284	3.916	**3.593**
Case 3	$\sigma \sim \mathcal{U}(0.1, 0.2)$	PSNR↑	16.63	25.36	28.38	31.91	32.13	32.40	33.54	34.98	**35.67**
		MSSIM↑	0.212	0.573	0.740	0.855	0.867	0.866	0.899	0.932	**0.941**
		SAM↓	39.816	16.371	11.497	6.840	7.272	6.448	5.766	4.755	**4.269**

using small noisy-clean image patches extracted from the synthesis DC dataset. As shown in [53], HSI-SDeCNN concerned the noisy simulated dataset only as in Case 1 with various values of σ. The experimental results presented here are carried out by using a pre-trained network for HSI-SDeCNN, a 2-D CNN for DIP-HSI (default setting), and the setting as used in the simulated cases for SURE-CNN. To be fair, all the methods are tested on the simulated dataset that was used in [53]. It is a part of the DC dataset which has 200×200 pixels of 191 bands. It is shown in false color using bands 57, 27, and 17 in the first column of Fig. 2.22.

Table 2.7, and Figs. 2.22 shows the denoising results using SURE-CNN and competitive DL-based methods. SURE-CNN outperforms HSI-SDeCNN and DIP-HSI in terms of PSNR and MSSIM. Also, the denoised images from SURE-CNN are cleaner than the denoised images obtained by HSI-SDeCNN and DIP-HSI.

RUNNING TIME

Running times for all methods for denoising the IP and UB datasets are given in Table 2.8. There, SURE-CNN implemented using Tensorflow 2.0 GPU was run on a Linux computer with eight cores Intel CPU 3.2 GHz, 64 GB of RAM and the Nvidia Titan X GPU of 12 GB memory. The remaining denoising methods are implemented using Matlab R2019b, and were run on the same computer. The fastest method is FastHyDe with less than 1 second for denoising the IP and UB datasets, while BM4D is the slowest method. The average running time of SURE-CNN is in the middle between the slow group methods (BM4D, and NAILRMA) and the fast group methods (FastHyDe, HyRes, and FORPDN). SURE-CNN running time is 89.87 seconds for the



(a) Reference image *(b)* Noisy image *(c)* LRTA *(d)* TDL *(e)* FORPDN

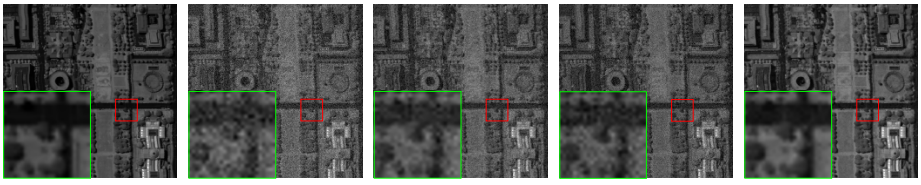
(f) BM4D *(g)* NAILRMA *(h)* HyRes *(i)* FastHyDe *(j)* SURE-CNN

Figure 2.11. Denoising for the DC dataset band 60, Case 1 with $\sigma = 0.3$, using different methods. The green square is a zoomed-in area shown in the red square.

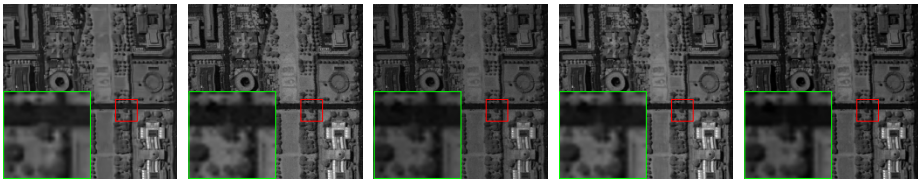
IP dataset and 126.42 seconds for the UB dataset.

2.4 Conclusions

Two new HSI denoising methods are proposed, which both are unsupervised DL-based methods. The first method (DIP-SLR) explicitly exploits the sparse and low-rank characteristic of HSI and incorporated those characteristic with the DIP, where the results and the computational efficiency are improved. The second method (SURE-CNN) uses the SURE-based loss function to train a CNN which overcomes the overfitting problem. SURE-CNN can be also extended to work with non-Gaussian noise and with the low-rank data to leverage the results and reduce the running time. However, there are still some open questions such as the exact computation of the network divergence in SURE-CNN, the automatic parameter selection for DIP-SLR and denoising of other noise kinds (e.g., stripped noise, salt and pepper noise), that will be concerned in the future work.

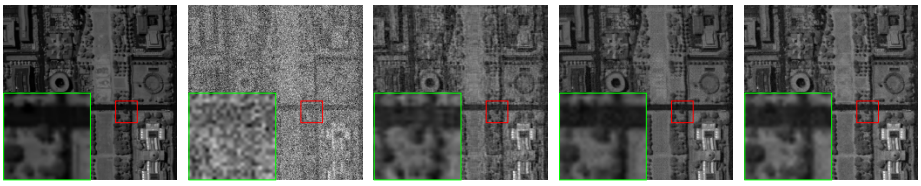


(a) Reference image *(b)* Noisy image *(c)* LRTA *(d)* TDL *(e)* FORPDN

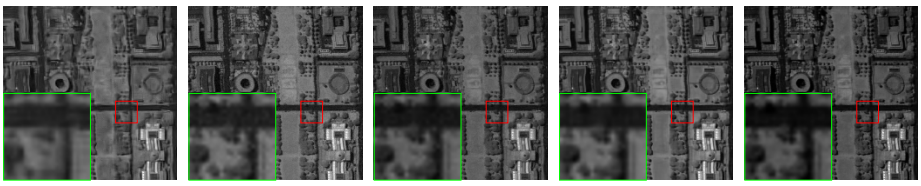


(f) BM4D *(g)* NAILRMA *(h)* HyRes *(i)* FastHyDe *(j)* SURE-CNN

Figure 2.12. Denoising for the DC dataset band 60, Case 2 with $\sigma = 1, \eta = 20$, using different methods. The green square is a zoomed-in area shown in the red square.

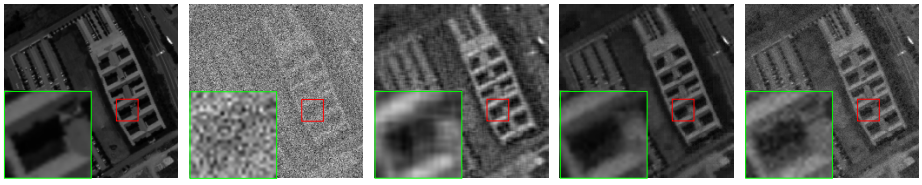


(a) Reference image *(b)* Noisy image *(c)* LRTA *(d)* TDL *(e)* FORPDN



(f) BM4D *(g)* NAILRMA *(h)* HyRes *(i)* FastHyDe *(j)* SURE-CNN

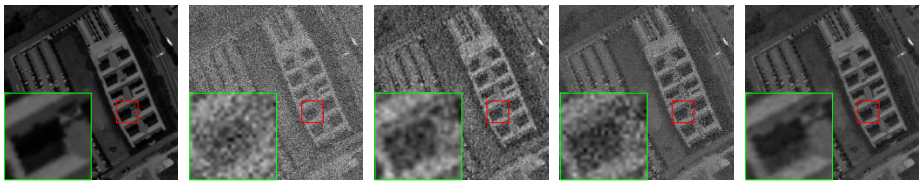
Figure 2.13. Denoising for the DC dataset band 60, Case 3 with $\sigma \sim \mathcal{U}(0.1, 0.2)$, using different methods. The green square is a zoomed-in area shown in the red square.



(a) Reference image *(b)* Noisy image *(c)* LRTA *(d)* TDL *(e)* FORPDN

(f) BM4D *(g)* NAILRMA *(h)* HyRes *(i)* FastHyDe *(j)* SURE-CNN

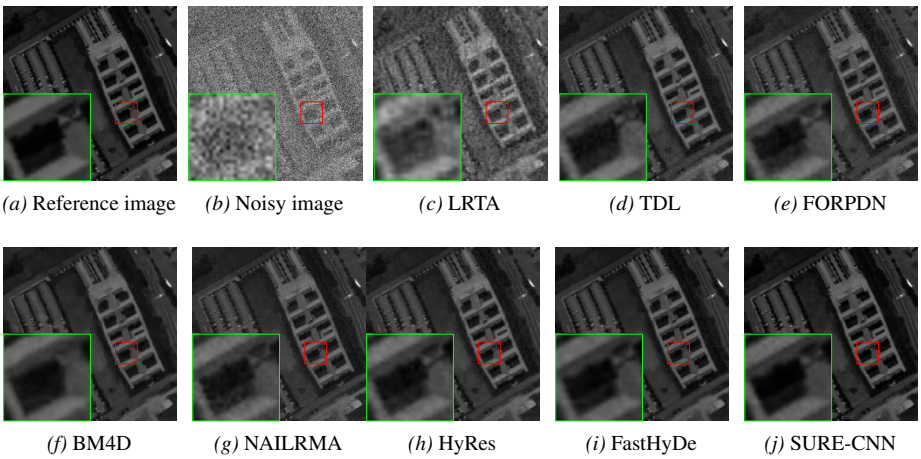
Figure 2.14. Denoising for the PU dataset band 60, Case 1 with $\sigma = 0.3$, using different methods. The green square is a zoomed-in area shown in the red square.



(a) Reference image *(b)* Noisy image *(c)* LRTA *(d)* TDL *(e)* FORPDN

(f) BM4D *(g)* NAILRMA *(h)* HyRes *(i)* FastHyDe *(j)* SURE-CNN

Figure 2.15. Denoising for the PU dataset band 60, Case 2 with $\sigma = 1, \eta = 20$, using different methods. The green square is a zoomed-in area shown in the red square.



(a) Reference image *(b)* Noisy image *(c)* LRTA *(d)* TDL *(e)* FORPDN

(f) BM4D *(g)* NAILRMA *(h)* HyRes *(i)* FastHyDe *(j)* SURE-CNN

Figure 2.16. Denoising for the PU dataset band 60, Case 3 with $\sigma = \mathcal{U}(0.1, 0.2)$, using different methods. The green square is a zoomed-in area shown in the red square.

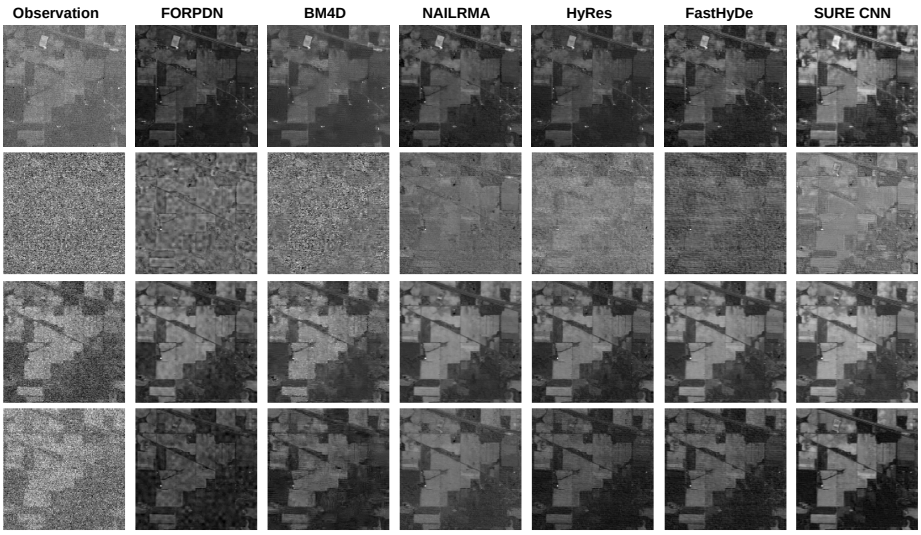


Figure 2.17. Denoising for the IP dataset using different methods, from top to bottom: bands 2, 104, 149, and 219.

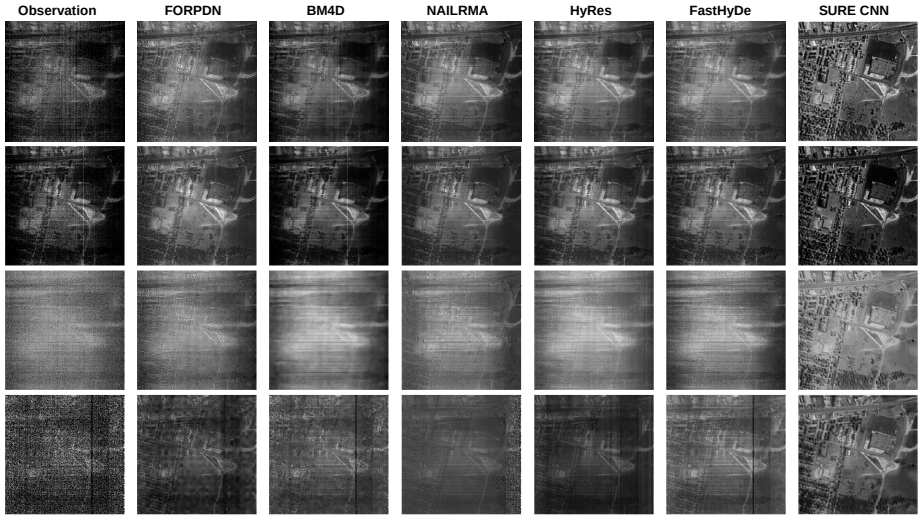
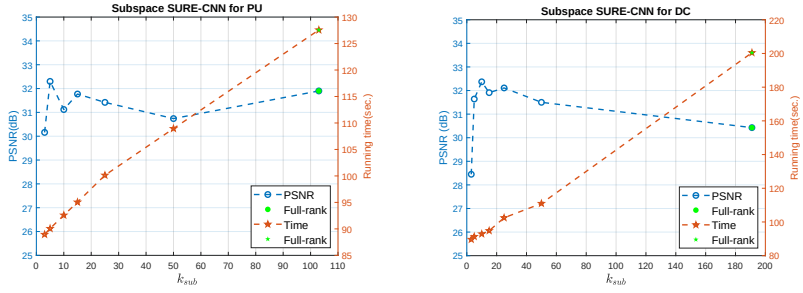
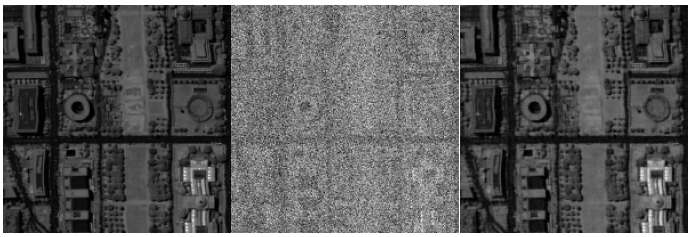


Figure 2.18. Denoising for the UB dataset using different methods, from top to bottom: bands 108, 139, 144, and 208.

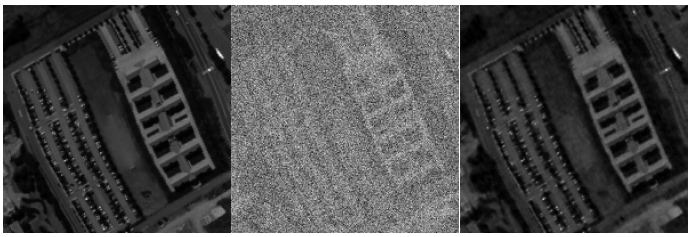


(a) PU dataset *(b)* DC dataset

Figure 2.19. Subspace SURE-CNN for denoising in Case 1 with $\sigma = 0.3$. The green markers show SURE-CNN in full-rank (no subspace).

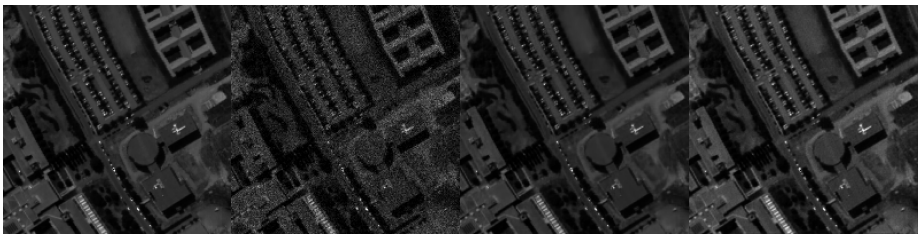


(a) Reference image (DC) *(b)* Noisy image (DC) *(c)* Subspace SURE-CNN ($k_{sub} = 10$)



(d) Reference image (PU) *(e)* Noisy image (PU) *(f)* Subspace SURE-CNN ($k_{sub} = 5$)

Figure 2.20. Denoising results using Subspace SURE-CNN for the DC and PU datasets, band 60.



(a) Reference image (PU) *(b)* Noisy image, PSNR=26.48 dB *(c)* FastHyDe, PSNR=39.94 dB *(d)* SURE-CNN, PSNR=39.14

(e) Noisy image, PSNR=21.47 dB *(f)* FastHyDe, PSNR=35.19 dB *(g)* SURE-CNN, PSNR=36.06

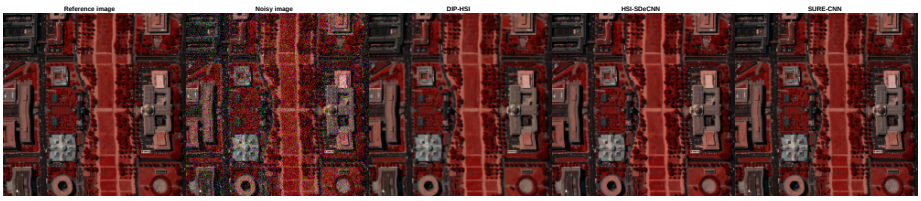
Figure 2.21. Poissionian denoising results using SURE-CNN for PU dataset, band 60.

Table 2.7. Denoising results given by PSNR in dB and MSSIM using DL-based methods. The results are the average values over 10 runs. The standard deviations for PSRN and MSSIM in each method are less than 0.06 *dB and* 0.0005*, respectively. Best results are in bold.*

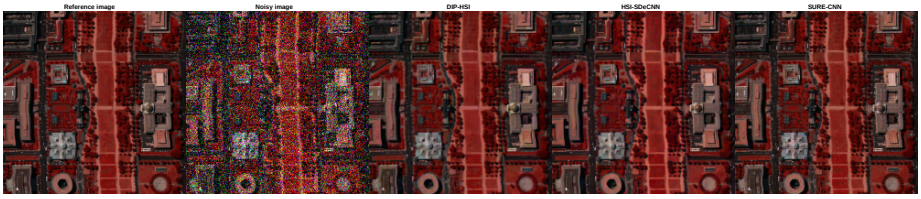
Noise level	Metric	Noisy	HSI-SDeCNN	DIP-HSI	SURE-CNN
$\sigma = 25/255$	PSNR (dB)↑	20.17	33.18	34.88	**36.07**
	MSSIM↑	0.491	0.963	0.949	**0.975**
$\sigma = 50/255$	PSNR (dB)↑	14.15	31.03	31.20	**33.23**
	MSSIM↑	0.238	0.918	0.924	**0.953**
$\sigma = 100/255$	PSNR (dB)↑	8.13	27.65	27.71	**28.98**
	MSSIM↑	0.080	0.852	0.825	**0.888**

Table 2.8. Average running time (in seconds) over 5 runs for different denoising methods. The standard deviations of SURE-CNN are 1.54 and 2.30 seconds for the IP and UB datasets, respectively. Best results are in bold.

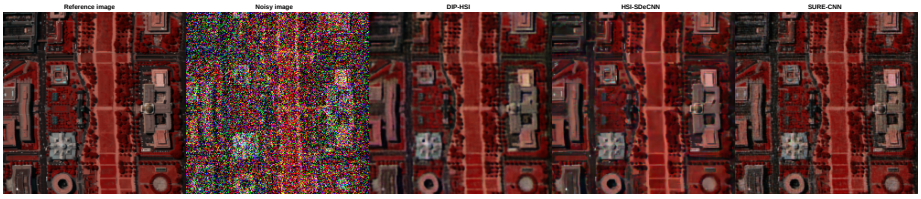
Dataset	FORPDN	BM4D	NAIL-RMA	HyRes	FastHyDe	SURE-CNN
IP	2.08	281.21	94.09	1.19	**0.15**	89.87
UB	5.79	1237.7	362.44	3.99	**0.79**	126.42



(a) Reference image *(b)* Noisy image *(c)* DIP-HSI *(d)* HSI-SDeCNN *(e)* SURE-CNN



(f) Reference image *(g)* Noisy image *(h)* DIP-HSI *(i)* HSI-SDeCNN *(j)* SURE-CNN



(k) Reference image *(l)* Noisy image *(m)* DIP-HSI *(n)* HSI-SDeCNN *(o)* SURE-CNN

Figure 2.22. Denoising results for isotropic Gaussian noisy data (a part of the DC dataset) with $\sigma = 25/255$ (a-e), $\sigma = 50/255$ (f-j) and $\sigma = 100/255$ (k-o) using DL-based methods.

CHAPTER 3

SENTINEL 2 IMAGE FUSION USING UNSUPERVISED CONVOLUTIONAL NEURAL NETWORK

The recently launched Sentinel-2 (S2) constellation provides multi-resolution MSI at 10 m, 20 m and 60 m, which is useful in many remote sensing applications. To maximize the use of the S2 spectral and spatial resolution, this chapter introduces a fusion method to obtain all S2 bands at their maximum resolution (10 m). The S2 fusion proposed here uses a single unsupervised CNN for sharpening both the 20 m and 60 m bands simultaneously, and also takes into account the degradation model which is based on the S2 sensors' modulation transfer functions (MTFs). Hereafter in this chapter, the proposed method is called S2SUCNN for short. Codes of the S2SUCNN method are available at `https://github.com/hvn2/S2S_UCNN`.

3.1 Problem formulation and motivation

S2 is a constellation of two polar-orbiting satellites S2A and S2B placed in the same sun-synchronous orbit and phased at 180° to each other. The twin satellites cover almost all the Earth's land surface with a revisit time of five days. The S2 MSI has 13 spectral bands. Four bands at visible and NIR wavelength region have a spatial resolution of 10 m. Six bands at shortwave infrared spectrum have a spatial resolution of 20 m. Three bands have 60 m resolution, which are used for atmospheric correction. S2 sharpening (fusion) is categorized as a hypersharpening technique and also can be considered as a special case of the pansharpening problem where the PAN is the 10 m bands and the LR images are the 20 m and the 60 m bands. However, S2 sharpening is more difficult than pansharpening since the spectrum of the 10 m bands only partially overlaps the spectrum of the 20 m and 60 m bands, and the LR bands have more than one resolution ratios (i.e., the resolution ratios are 2 and 6 for the 20 m and the 60 m bands, respectively), while in pansharpening the PAN usually has spectrum fully overlapping the spectrum of the MSI, and there is only one resolution ratio between the PAN and MSI.

S2 sharpening is the problem of estimating the HR image (10 m) of the 20 m and 60 m bands subject to the observation model behind the LR and (unknown) HR images, which is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon}, \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^{M\times 1}$ and $\mathbf{y} \in \mathbb{R}^{N\times 1}$ ($M = r^2N$, where r is the HR and LR ratio) are the

original HR and LR images, respectively. The degradation operator $\mathbf{H} \in \mathbb{R}^{M \times N}$ is a block diagonal matrix, i.e.,

$$\mathbf{H} = \text{bdiag}(\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_L)$$

where $L = 12$ is the number of bands (band 10 is ignored since it does not contain the Earth's surface information). For the ith band, the degradation operator $\mathbf{H}_i$ is composed of blurring and downsampling operators and is written as $\mathbf{H}_i = \mathbf{M}_i\mathbf{B}_i$, where $\mathbf{B}_i$ is a block-circulant-circulant-block (BCCB) matrix that represents the blurring, and $\mathbf{M}_i$ is a downsampling matrix by a factor of r_i, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Omega})$ is the additive noise. Note that the 10 m bands are assumed to be a part of the HR bands, and degradation matrices applied to the 10 m bands are identity.

Early S2 sharpening methods [90], [147] are based on pansharpening algorithms where the PAN is selected or synthesized from the 10 m bands. Modern approaches for S2 sharpening are model-based and DL-based methods. The model-based methods [97]–[100] define explicitly image priors used to seek a solution for (3.1) via an optimization algorithm, while the DL-based methods [148]–[150] learn a network that maps the LR image to its HR counterpart by a training process with a huge dataset containing input-ground truth pairs. Both model-based and DL-based methods have pros and cons, for example, the model-based methods are unsupervised but they require hand-crafted priors and the performance heavily depends on several tuning parameters; the DL-based methods are fast and usually parameter-free, but they require the ground truth, which is hard to obtain, in the training phase. The above-mentioned pros and cons are the main motivation to develop a hybrid method that inherits the advantage and overcomes the disadvantage of both model-based and DL-based methods. In fact, the proposed method relies on the DIP which is discussed in Section 2.1 to optimize a CNN in a similar way to the model-based methods where the DIP replaces the pre-defined prior and the DL training mechanism (e.g., back-propagation algorithms) is used as an optimization algorithm. The proposed method has several novel aspects. Firstly, it is an unsupervised DL-based method, since there is no ground truth required. Secondly, unlike the traditional DL-based methods [148]–[150] that train each CNN for sharpening the 20 m and 60 m bands, separately, the proposed method uses a single CNN for both 20 m and 60 m. This setting exploits the high correlation between all the S2 bands and the multitask learning to enhance the sharpening results. Finally, since the MTFs of the S2 sensors are available [151], the proposed method implements a MTF-based degradation as a network layer, which is as identical as the sensor's observation model. By doing this, the spectral information of the sharpened images is preserved. Experiments with both simulated and real datasets show that the proposed method yields good results and outperforms the competitive methods in both quantitative and quality evaluation.

3.2 Unsupervised CNN with MTF degradation model for S2 sharpening

The proposed unsupervised CNN S2 sharpening method in this section is based on the Wald's protocol [152] stating that the image obtained by spatially decimating the fused

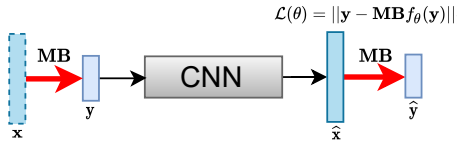


Figure 3.1. S2SUCNN framework. The red arrow represents an MTF-based degradation.

image to the resolution of the LR image should be as identical as possible to the LR image. Note that the filter kernels used in the decimated process should ideally match the PSFs of the MSI sensors. In the context of S2 sharpening, the Wald's protocol means that the estimated 20 m and 60 m bands at 10 m resolution whenever they are degraded to the 20 m and 60 m resolution should be identical to the observed 20 m and 60 m bands, respectively. Since the S2 sensors' MTFs are available [151], the PSFs can be computed and the degradation model is constructed. Assuming that the observed LR bands are obtained by applying the MTF-based degradation model to the original HR bands, and the estimated HR bands are obtained by a CNN. The Wald's protocol motivates the proposed method to minimize the difference between the observed LR bands and the decimated version of the estimated HR bands. This idea is depicted in Fig. 3.1. Here, the CNN output is the sharpened bands, i.e., $\hat{\mathbf{x}} = f_{\boldsymbol{\theta}}(\mathbf{y})$, and the MTF-based degradation, $\mathbf{MB}$, is implemented as a CNN layer. The loss function to be optimized involves only the observed image. Thus, the method is unsupervised. In the following subsections, the CNN structure and the implementation are detailed.

3.2.1 Network structure and optimization

The CNN structure sketched in Fig. 3.2 is used to demonstrate the proposed method. In Fig. 3.2, the CNN explicitly works with an image represented in multi-dimensional array, the 10 m, 20 m, 60 m bands and the fused image are denoted as $\mathcal{Y}_{10}, \mathcal{Y}_{20}$, and $\mathcal{Y}_{60}$, and $\widehat{\mathcal{X}}$, respectively. The network is comprised of two subnetworks. The first subnetwork (called the 60 m subnetwork) sharpens the 60 m bands to the 20 m resolution, and the second subnetwork (call 20 m subnetwork) takes the output of the 60 m subnetwork, and fuses with the 10 m bands to produce all bands at 10 m resolution. In the 60 m subnetwork, the features of the 60 m bands are extracted by using four *conv+cn+relu* blocks where each block consists of a 2-D convolutional layer with 128 filters of size (3×3) and a stride of 1, followed by a channel normalization layer and a LeakyReLU layer. Those 60 m bands features are fused with the 20 m bands features which are obtained by an autoencoder-like network with skip connections to produce the 60 m features map at the 20 m resolution. The skip-connection network is used because it gives good DIP and converges fast [153]. The 20 m subnetwork is similar to the 60 m subnetwork with the main components are four *conv+cn+relu* blocks and the autoencoder-like network with skip connections. There, the skip-connection autoencoder-like network extracts the 10 m bands features and the *conv+cn+relu* blocks extract the features of the output of the 60 m subnetwork. Those features are merged to result in the sharpened bands at 10 m resolution via a *conv+sigmoid* block composing of

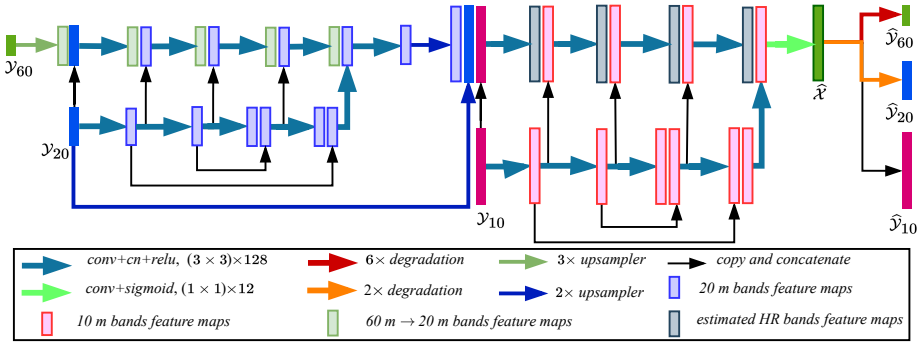


Figure 3.2. S2SUCNN network structure. In the bottom box, conv+cn+relu, (3×3)×128 is a block of a 2-D convolutional layer with 128 filters of size (3×3), followed by a channel normalization layer and ReLU activation layer. conv+sigmoid, (1×1)×12 is a block of 2-D convolutional layer with 12 filters of size (1×1), followed by a Sigmoid activation layer. $k\times$ degradation ($k = 2, 6$) and $k\times$ upsampler ($k = 2, 3$) are the MTF-based degradation layer and the bilinear upsampling layer, by a factor of k, respectively.

a 2-D convolutional layer with 12 filters of size (1×1) and a Sigmoid layer. In practice, the interested targets are only the 20 m and 60 m bands. But the 10 m bands are also included in the CNN output, since it is benefited by the high correlation between all S2 bands and the multitask learning [154], [155]. Finally, the estimated 60 m and 20 bands are degraded using the MTF-based degradation layers with factors of six and two, respectively. The degradation of the estimated image is required to compute the loss function, given by

$$\mathcal{L}(\boldsymbol{\theta}) = \|\mathbf{y} - \mathbf{H} f_{\boldsymbol{\theta}}(\mathbf{y})\|_1 = \|\mathbf{y} - \mathbf{MB} f_{\boldsymbol{\theta}}(\mathbf{y})\|_1. \tag{3.2}$$

Note that the degradation operator for the 10 m bands is identity, and the ℓ_1 is used for the loss function since it converges faster and outperforms the ℓ_2 one [148], [150]. The network is implemented by using Tensorflow 2.1, and is optimized by using the Adam optimizer [136] with a learning rate of 0.001

3.2.2 MTF-based degradation model

As it was mentioned in (3.1) and depicted in Fig. 3.1, the observed LR image, $\mathbf{y}$, is assumed to be a degraded version of the (unknown) original HR image, $\mathbf{x}$, by applying the degradation operator, $\mathbf{H}$. In S2 imagery, it is hard to estimate the correct degradation operator. It is common practice to assume the degradation model to be an MTF-filtering and downsampling process [97]–[100], where the information to construct the MTF-filters are the S2 PSFs [151]. We would like to minimize the difference between the observed LR image, $\mathbf{y}$, and the decimated version of sharpened image, $\mathbf{MB} f_{\boldsymbol{\theta}}(\mathbf{y})$, by optimizing the loss function (3.2). Therefore, it is naturally to incorporate the degradation operator as a part of the network. The degradation operator used in the network should be as identical as possible to the degradation operator in the observation

model (3.1), as it was stated in the Wald's protocol. Therefore, the degradation operator here is created as a depth-wise convolutional layer with fixed Gaussian filter kernels taken from the S2 MTFs, followed by downsampling with the ratio between the HR and LR images. The procedure to implement the MTF-based degradation layer is described as following steps,

- Step 1: Create Gaussian kernels for the filters with standard deviations [97]

$$\text{sd}_i = r_i \sqrt{\frac{-2\ln(\text{MTF}_i)}{\pi^2}}, i = 1, 2, ..., 12,$$

  where $r_i = 1, 2, 6$ for different resolutions and MTF_i is the MTFs given in Table 3.1 [151].

- Step 2: Create depth-wise convolutional layers for 20 m and 60 m bands using fixed kernels $\text{sd}_i, i \in \{5, 6, 7, 8a, 11, 12\}$ and $\text{sd}_i, i \in \{1, 9\}$ given in Step 1, respectively. The kernel size is 7×7.

- Step 3: Apply the filters created in Step 2 to the sharpened bands and downsample the results by factors of 2 and 6 for 20 m and 60 m resolution bands, respectively.

Table 3.1. MTF values at the Nyquist frequency for the S2 bands.

Bands	B1	B2	B3	B4	B5	B6
MTF	0.32	0.26	0.28	0.24	0.38	0.34
Bands	B7	B8	B8a	B9	B11	B12
MTF	0.34	0.26	0.33	0.26	0.22	0.23

3.3 Experimental results

Datasets and Evaluation Metrics

The datasets used to evaluate the proposed method are the real S2 data which are publicly available on the ESA/Copernicus portal. Four datasets called Australia, Iceland, USA and Vietnam, where each dataset is a S2 image captured a scene in Australia, Iceland, the USA and Vietnam, respectively. The Australia, Iceland, USA and Vietnam datasets are used for the 20 m bands sharpening experiments. One dataset called USA-60, which is a S2 image captured a scene in the USA, is used for the 60 m bands sharpening experiment. Those datasets are described in detail in the Appendix.

The sharpening results are evaluated in reduced-resolution where references are available and in full-resolution where the references are missed. For reduced-resolution evaluation, the following metrics are used, the signal-to-reconstruction error (SRE) and the mean SRE (MSRE) in decibels, the Erreur Relative Globale Adimensionnelle de

Syntheses (ERGAS), the spectral angle mapper (SAM) in degrees, and the structural similarity (SSIM) index. The definition and description of those quantitative metrics are given in the Appendix.

3.3.1 Verification of the MTF-based degradation model and multitask learning

As discussed above, the proposed method employs the MTF-based degradation model as a network layer, and exploits the multitask learning advantage by adding the 10 m bands at the output according with the 20 m and the 60 m bands. To verify the effectiveness of this setting, four networks with or without the MTF-based degradation layer and the 10 m bands at the network output are constructed. Those networks have the same structure as described in Fig. 3.2 but different degradation layers and outputs. The experiments are conducted in reduced-resolution for both 60 m and 20 m bands sharpening where the observed data are decimated to a lower resolution to synthesize the LR bands (i.e., 20m $\rightarrow$ 40 m and 60 m $\rightarrow$ 360 m), and the observed data can be used as references. The experiments with each network and reduced-resolution data are described as below:

- Network 1: Uses the MTF-based degradation layer. The reduced-resolution data are synthesized by the MTF-based degradation model. The output includes all bands. This configuration indicates that the degradation layer implemented in the network and the degradation model assumed in (3.1) are identical with the MTFs taken into account, and the multitask learning is utilized.

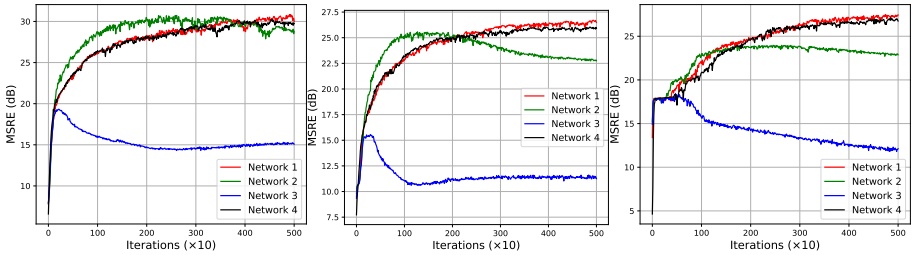
- Network 2: Uses the bicubic-based degradation layer. The reduced-resolution data are synthesized by the bicubic-based degradation model. The bicubic-based degradation model and layer are the function $tf.image.resize(\cdot)$ of Tensorflow 2.1. The output includes all bands. This configuration indicates that the degradation layer implemented in the network degradation model assumed in (3.1) are identical but the MTFs are not used, and the multitask learning is utilized.

- Network 3: Uses the MTF-based degradation layer. The reduced-resolution data are synthesized by the bicubic-based degradation model. The output includes all bands. This configuration indicates that the degradation layer implemented in the network and the degradation model assumed in (3.1) are not matched, and the multitask learning is utilized.

- Network 4: Uses the MTF-based degradation layer. The reduced-resolution data are synthesized by the MTF-based degradation model. The output includes only 20 m bands (for 20 m bands sharpening), or 60 m bands (for 60 m bands sharpening). This configuration indicates that the degradation layer implemented in the network and the degradation model assumed in (3.1) are identical with the MTFs taken into account, and the multitask learning is not utilized.

Fig. 3.3 shows the results for 20 m bands sharpening using Iceland and Vietnam datasets, and for 60 m bands sharpening using USA-60 dataset. The graphs plot MSRE as a function of optimized iterations. Clearly, the agreement between the degradation used in the observation model and in the network significantly improves the fusion results.

It can be seen that Network 3 gives worse results and overfits quickly, while Network 1, Network 2, and Network 4 give significantly higher MSRE than Network 3 for both 20 m and 60 m bands sharpening. Besides, the MTF-based degradation model is more appropriate than the bicubic-based one. Fig 3.3 indicates that Network 2 not only produces lower MSRE than Network 1 but also begins overfitting after 2000 iterations, especially for the images having more details such as in Vietnam, and USA-60 datasets. The reason is likely that the bicubic-based degradation smooths out the image details more than the MTF-based degradation does. Finally, a single network for both 20 m and 60 m bands sharpening (Network 1) performs slightly better than the separate networks (Network 4). The results verify the effectiveness of multitask learning obtained by the high correlation of all S2 band.



(a) Iceland dataset *(b)* Vietnam dataset *(c)* USA-60 dataset

Figure 3.3. The effectiveness of MTF-based degradation layer and multitask learning. The graphs show MSRE (dB) as a function of optimization iterations for four networks using reduced-resolution data. The results are average values over 5 runs. (a) and (b) 20 m bands sharpening, (c) 60 m bands sharpening.

3.3.2 Reduced-resolution evaluation

In S2 sharpening, the HR image (reference) is unknown. Therefore, it is unable to evaluate the methods quantitatively using the criteria such as SRE, ERGAS, SAM, and SSIM in original (full) resolution. However, one widely adopted solution is to evaluate the performance in reduced-resolution [71], [156]. To make this solution practical, the observed LR image must be reduced the resolution spatially r times, which is equal to the resolution ratio between the HR and LR images, by using the decimated filters that match the sensors' MTFs. As a result, the observed LR image can be used as a reference. It is obvious that the evaluation in reduced-resolution may not perfectly indicate the quality of the methods when they perform in full-resolution. However, it gives reasonable information to compare the methods.

20 M BANDS SHARPENING

To evaluate the 20 m bands sharpening using the reduced-resolution data, all observed S2 bands are filtered by using the MTF-based low-pass filters and downsampled by a factor of two. This process results in the data at a lower resolution (i.e., 10 m $\rightarrow$

20 m, 20 m $\rightarrow$ 40 m, and 60 m $\rightarrow$ 120 m), and the 20 m bands are the references. The proposed method is compared against both model-based and DL-based methods. The model-based methods are ATPRK [147], SupReME [97], S2Sharp [99], and SSSS [100]. And, the DL-based methods are one supervised DL-based method using a skip connection CNN (SSC-CNN) [157] and the S2SUCNN network without the 10 m at the output (called S2SUCNN+, i.e., Network 4 described in Section 3.3.1). To find the optimal parameters for the model-based methods, a tuning pipeline suggested in [80] is applied. The SSC-CNN network is trained 300 epochs by using small patches in another lower reduced-resolution (i.e., in training, the network maps 80 m $\rightarrow$ 40 m) and tested on this reduced-resolution (i.e., in testing, the network maps 40 m $\rightarrow$ 20 m). The network structure and parameters used for S2SUCNN+ and S2SUCNN are described in Fig. 3.1, and the MSRE is monitored during training and the model giving highest MSRE is chosen.

Tables 3.2-3.5 show the quantitative results in terms of each band SRE, MSRE, SAM, ERGAS, and mean SSIM (MSSIM) for Australia, Iceland, USA, and Vietnam datasets, respectively. The best results are highlighted in bold. The DL-based methods significantly outperform the model-based methods in all metrics and all datasets. The gaps between two groups for all datasets are approximately $4 - 8$ (dB), $0.6 - 3.1$, $0.5 - 1.1$ ($^\circ$), and $0.05 - 0.12$ of MSRE, ERGAS, SAM, and SSIM, respectively. The results are not surprising since the model-based methods heavily depend on the regularizers and hyperparameters which are data-dependent. While the DL-based methods implicitly learn them from data or by the network structure. Among the model-based methods, SSSS generally gives better results than the other three methods for all datasets. SupReME and S2Sharp results are quite similar, while ATPRK is considerably worse than SupReME, S2Sharp and SSSS. For the DL-based methods, S2SUCNN outperforms S2SUCNN+ and SSC-CNN in terms of MSRE and SAM for all datasets. But, SSC-CNN gives better structural similarity of the sharpened images for Australia and USA datasets, which is indicated by the MSSIM for SSC-CNN lower than ones for S2SUCNN+ and S2SUCNN with a margin of nearly $0.005 - 0.01$.

Table 3.2. Australia dataset: Reduced-scale resolution performance for 20 m bands sharpening. The columns B5 to B12 are SRE of each band from band 5 to band 12. SRE is given in decibels and SAM is given in degrees. Best results are in bold.

Dataset	Methods	B5↑	B6↑	B7↑	B8a↑	B11↑	B12↑	MSRE↑	ERGAS↓	SAM↓	MSSIM↑
Australia	ATPRK	29.31	23.82	23.05	21.58	20.88	20.70	23.22	4.570	1.847	0.818
	SupReMe	30.43	27.66	27.31	26.69	22.42	21.24	25.96	3.617	1.696	0.883
	S2Sharp	29.35	29.00	29.13	28.69	21.40	19.95	26.25	3.917	1.891	0.882
	SSSS	29.81	28.49	28.26	27.43	22.60	18.08	25.78	4.329	1.835	0.886
	SSC-CNN	**36.77**	32.94	32.61	32.07	29.68	27.89	31.99	1.716	0.784	**0.944**
	S2SUCNN+	34.60	32.04	32.42	32.13	29.85	27.40	31.41	1.786	0.905	0.928
	S2SUCNN	36.42	**34.05**	**34.02**	**34.20**	**30.64**	**27.93**	**32.87**	**1.593**	**0.782**	0.939

The qualitative results for all methods are pictured in Figs. 3.4-3.7 for small parts of bands 7, 8a, 11 and 12 of the Australia, Iceland, USA and Vietnam datasets, respectively. Overall, all methods give good quality sharpened images. However, ATPRK and SSSS

Table 3.3. Iceland dataset: Reduced-scale resolution performance for 20 m bands sharpening. The columns B5 to B12 are SRE of each band from band 5 to band 12. SRE is given in decibels and SAM is given in degrees. Best results are in bold.

Dataset	Methods	B5↑	B6↑	B7↑	B8a↑	B11↑	B12↑	MSRE↑	ERGAS↓	SAM↓	MSSIM↑
Iceland	ATPRK	26.98	21.81	21.03	20.61	24.50	24.37	23.22	3.988	1.805	0.854
	SupReMe	28.82	26.01	25.58	24.85	25.38	26.32	26.16	2.708	1.693	0.899
	S2Sharp	29.76	26.22	25.72	25.31	25.90	25.99	26.48	2.623	1.669	0.906
	SSSS	28.58	28.09	27.47	27.42	27.71	25.87	27.52	2.284	1.532	0.928
	SSC-CNN	31.94	30.03	29.59	29.56	31.53	30.52	30.53	1.624	1.166	0.961
	S2SUCNN+	31.91	30.02	29.43	29.36	31.39	29.59	30.28	1.673	1.200	0.950
	S2SUCNN	**33.70**	**30.83**	**31.06**	**31.17**	**32.68**	**30.66**	**31.68**	**1.427**	**1.093**	**0.962**

Table 3.4. USA dataset: Reduced-scale resolution performance for 20 m bands sharpening. The columns B5 to B12 are SRE of each band from band 5 to band 12. SRE is given in decibels and SAM is given in degrees. Best results are in bold.

Dataset	Methods	B5↑	B6↑	B7↑	B8a↑	B11↑	B12↑	MSRE↑	ERGAS↓	SAM↓	MSSIM↑
USA	ATPRK	23.29	29.53	28.42	28.43	25.18	20.64	25.92	3.152	1.228	0.819
	SupReMe	26.03	28.98	31.57	31.57	25.45	23.02	27.77	2.508	1.209	0.854
	S2Sharp	26.95	28.29	32.10	32.24	25.42	23.89	28.15	2.366	1.211	0.866
	SSSS	26.28	31.34	32.26	32.30	26.95	24.86	29.00	2.146	1.112	0.879
	SSC-CNN	30.64	35.36	35.88	36.11	34.69	30.90	33.93	1.156	0.609	**0.942**
	S2SUCNN+	30.12	35.62	35.84	36.40	34.47	30.96	33.90	1.174	0.645	0.925
	S2SUCNN	**31.79**	**36.13**	**36.94**	**37.77**	**35.13**	**32.51**	**35.04**	**1.0078**	**0.579**	0.935

introduce some slight artifacts to the reconstructed images. The ATPRK images look more pixelated and blurrier as can be seen clearly by zooming in the image (shown the river) for the Australia dataset and the image (shown the hill at the upper right corner) for the Iceland dataset. SSSS creates grid-like noise in the image (at the upper right corner) for the USA dataset. The images estimated by SupReME and S2Sharp are roughly identical, and they are slightly blurrier than the images estimated by the DL-based methods. All DL-based methods produce high quality super-resolved images which look almost indistinguishable with the reference. S2SUCNN is slightly better than S2SUCNN+ and SSC-CNN. The superiority of S2SUCNN over the competitive methods is indicated in Fig. 3.8 that shows residual images of bands 7, 8a, 11, 12 in logarithm scale (i.e., $\log\big(1 + \left|\mathbf{X} - \widehat{\mathbf{X}}\right|\big)$) for Australia, Iceland, USA, and Vietnam datasets, respectively. It is clear that the residual structures of the images for S2SUCNN are less pronounced than the ones for the competitive methods.

60 M BANDS SHARPENING

Here, the evaluation of the 60 m sharpening results using the reduced-resolution data is performed. Similar to the 20 m bands, all the observed S2 bands are filtered by the MTF-based low-pass filter and downsampled by a factor of six, to create the simulated

Table 3.5. Vietnam dataset: Reduced-scale resolution performance for 20 m bands sharpening. The columns B5 to B12 are SRE of each band from band 5 to band 12. SRE is given in decibels and SAM is given in degrees. Best results are in bold.

Dataset	Methods	B5↑	B6↑	B7↑	B8a↑	B11↑	B12↑	MSRE↑	ERGAS↓	SAM↓	MSSIM↑
Vietnam	ATPRK	19.84	21.14	21.37	21.40	19.65	16.89	20.05	5.486	2.697	0.822
	SupReMe	23.03	23.09	23.46	23.90	22.47	16.35	22.05	4.766	2.735	0.858
	S2Sharp	23.48	24.16	24.55	24.74	21.93	18.15	22.83	4.197	2.770	0.891
	SSSS	22.87	23.86	24.31	24.63	24.62	20.88	23.53	3.623	2.385	0.912
	SSC-CNN	26.17	26.93	27.30	27.60	26.96	22.49	26.24	2.723	1.770	0.951
	S2SUCNN+	27.00	26.91	27.25	27.46	27.06	22.17	26.31	2.7315	1.799	0.948
	S2SUCNN	**28.06**	**28.17**	**28.39**	**28.41**	**27.48**	**22.70**	**27.20**	**2.499**	**1.745**	**0.958**

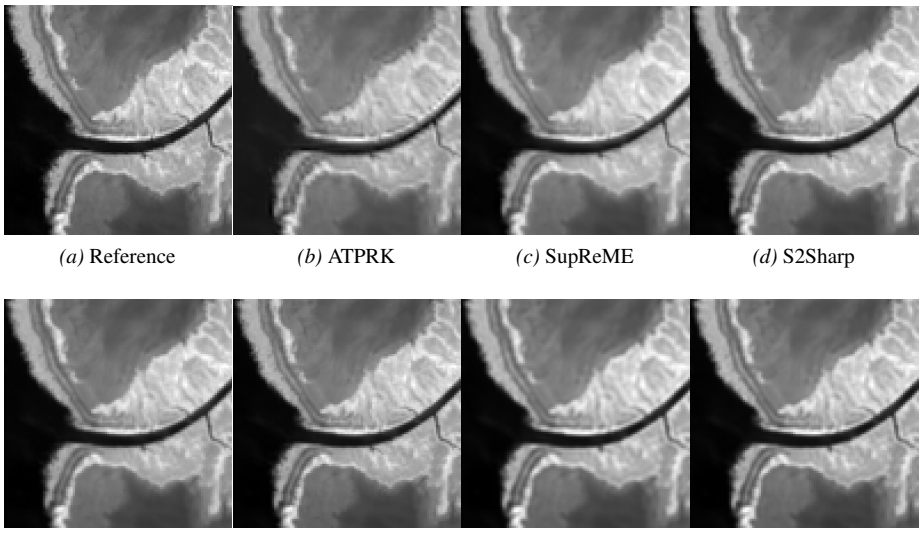
data. The proposed S2SUCNN method is compared against SupReME, S2Sharp, SSSS, and S2SUCNN+. The same tuning parameters technique applied for the 20 m bands sharpening is used for all methods.

The sharpening results for the USA-60 dataset are given in Table 3.6 and Figs. 3.9-3.11. Once again, S2SUCNN+ and S2SUCNN strongly outperform the model-based methods in all numerical metrics. For example, S2SUCNN gives 27.84 (dB) of MSRE which is about 7 (dB) higher than the model-based methods. In terms of ERGAS, SAM, and SSIM, S2SUCNN+ and S2SUCNN outperform the best model-based method, SSSS, with the gaps of 0.9, 1.3 ($^\circ$), and 0.3, respectively. S2SUCNN+ and S2SUCNN give similar results on MSE, ERGAS, and SAM. However, S2SUCNN performs considerably better than S2SUCNN+ on the structural similarity of the reconstructed images, since the MSSIM achieved by S2SUCNN (0.725) is higher than one by S2SUCNN+ (0.687). Surprisingly, S2Sharp is considerably worse than SupReME in all cases, although SupReME is a special case of S2Sharp with fixed subspace. The reason is more likely that the parameters of S2Sharp are not optimal for those datasets.

Table 3.6. Reduced-scale resolution performance for 60 m bands sharpening. The columns B1 and B12 are SRE of band 1 and band 9. SRE is given in decibels and SAM is given in degrees. Best results are in bold.

Method	B1↑	B9↑	MSRE↑	ERGAS↓	SAM↓	MSSIM↑
SupReME	20.93	21.18	21.06	1.489	1.519	0.525
S2Sharp	19.63	24.92	22.28	1.410	1.920	0.580
SSSS	19.25	20.82	20.03	1.687	2.266	0.513
S2SUCNN+	28.46	26.03	27.24	0.745	1.017	0.687
S2SUCNN	**29.24**	**26.44**	**27.84**	**0.700**	**0.970**	**0.725**

Fig. 3.9 displays parts of the estimated 60 m bands (band 1 and 9) for the USA-60 dataset. Apparently, SSSS fails to estimate the 60 m bands, since the images for bands 1 and 9 are significantly noisy. SupReME and S2Sharp also generate strong artifacts to the reconstructed images that look blurry and unnatural. The images obtained by S2SUCNN+ and S2SUCNN are also blurry, since the six times super-resolution for 60



(a) Reference *(b)* ATPRK *(c)* SupReME *(d)* S2Sharp

(e) SSSS *(f)* SSC-CNN *(g)* S2SUCNN+ *(h)* S2SUCNN

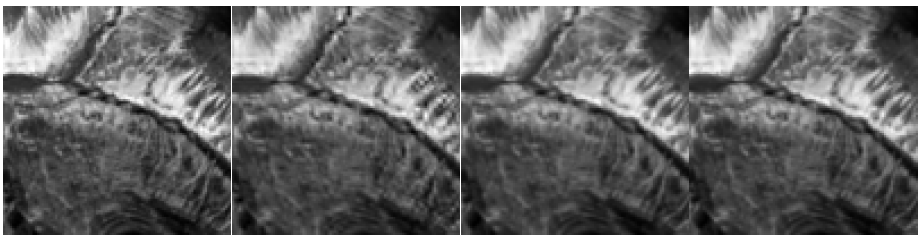
Figure 3.4. Reduced-resolution 20 m bands sharpening results for Australia dataset, the gray scale image shows a part of band 7.

m bands sharpening is naturally a hard problem. S2SUCNN gives a sharpened image that is visually sharper and closer to the reference image than the other methods. The better quality of the image obtained by S2SUCNN is verified in Fig. 3.10, where the residual structures for bands 1 and 9 are less than the other methods.

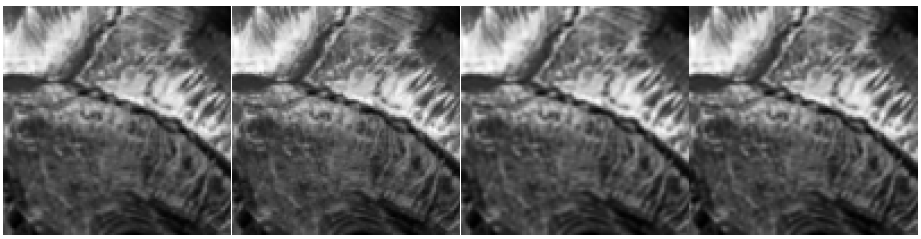
3.3.3 Full-resolution evaluation

This section presents the sharpening results in full-resolution for the Australia, Iceland, USA, and Vietnam datasets. The sharpened images obtained by ATPRK, SupReME, S2Sharp, SSSS, and S2SUCNN are the 20 m and 60 m bands estimated at 10 m resolution. The parameters for all methods are set the same as in reduced-resolution evaluation. In the 10 m resolution, there are no ground truth for the 20 m and 60 m bands. Therefore, the sharpened images are only assessed visually. In addition, the observed LR bands are upsampled to the 10 m resolution using the bicubic interpolation, and are used as a baseline.

Fig. 3.11 and Fig.3.12 show the results for 20 m bands sharpening, where the images are shown in false-color images using bands 12, 8a, and 5 as the R, G, and B channels. The images estimated by all five advanced methods are clearly sharper than ones interpolated by the simple bicubic method. However, the ATPRK images look more pixelated in all datasets. S2Sharp and SSSS also generate artifacts and a grid-like effect that can be seen in the zoomed-in areas of Australia, USA, and Vietnam datasets. The SupReME images are slightly blurrier than S2SUCNN images. We further assess the spectral information by inspecting the colors of the sharpened images and the



(a) Reference *(b)* ATPRK *(c)* SupReME *(d)* S2Sharp



(e) SSSS *(f)* SSC-CNN *(g)* S2SUCNN+ *(h)* S2SUCNN

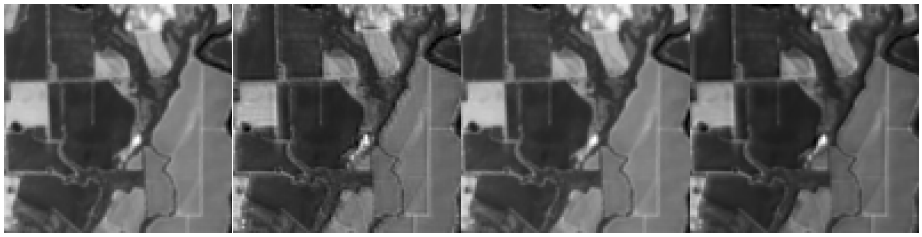
Figure 3.5. Reduced-resolution 20 m bands sharpening results for Iceland dataset, the gray scale image shows a part of band 8a.

bicubic interpolated images. SSSS and S2Sharp cause stronger spectral distortion in the Vietnam and USA datasets than other methods. S2SUCNN gives sharpened images which have both best spatial and spectral quality among the methods.

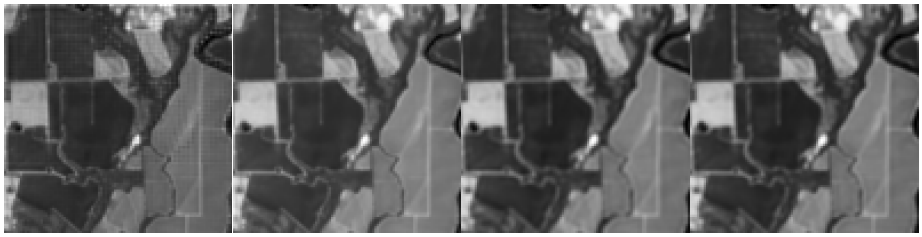
Fig. 3.13 and Fig. 3.14 display the sharpened 60 m bands shown in false-color images using bands 1, 9, and 1 as the R, G, and B channels. The quality of the sharpened images is remarkably improved more than the interpolated images by bicubic. Spatial details of the images are clearly recovered. However, SupReME and S2Sharp cause spectral bias on Australia and USA datasets, since the color of recovered images differs far from the color of the bicubic images. SSSS images are surprisingly sharp, but SSSS also creates the grid-like noise in the sharpened images for all datasets and a spectral distortion in the Vietnam dataset. The images obtained by S2SUCNN have spatial resolution quality as comparable as the competitive methods, while the spectral information is considerably better than the other methods.

Running Time

The running time in seconds for all methods is measured in the full-resolution experiments with the Australia, Iceland, USA and Vietnam datasets. S2SUCNN was implemented using Keras framework of Tensorflow 2.1 and was run in GPU mode of a Linux computer equipped 3.2 GHz Intel core I7 CPU, 12 GB of memory Nvidia Titan X GPU, and 64 GB of RAM. The running iterations are empirically chosen as 1000 iterations based on the visualization of the reconstructed images, for all four datasets. The competitive methods were implemented in Matlab R2020 and were run in CPU mode using the same computer. It should be noticed that SSSS handles a large



(a) Reference *(b)* ATPRK *(c)* SupReME *(d)* S2Sharp



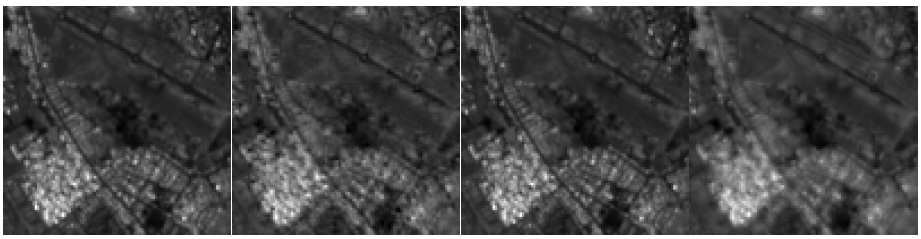
(e) SSSS *(f)* SSC-CNN *(g)* S2SUCNN+ *(h)* S2SUCNN

Figure 3.6. Reduced-resolution 20 m bands sharpening results for USA dataset, the gray scale image shows a part of band 11.

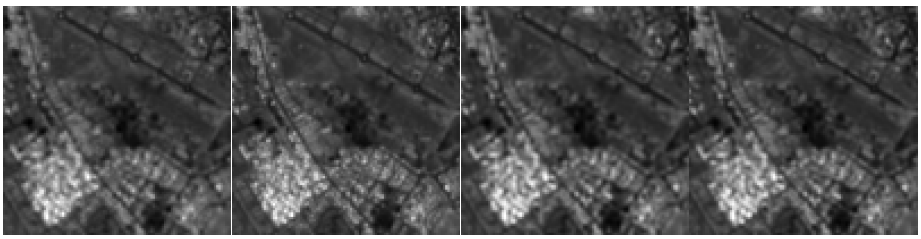
image by dividing it to smaller overlapping subimages to accelerate the algorithm. The smaller subimages are the faster the algorithm runs. In this experiment, SSSS used two subimages. The running times for all methods are given in Table 3.7. The fastest method is ATPRK which has computing time of approximately 9 seconds for all datasets. The S2SUCNN method is slower than SupReME and S2Sharp, but is faster than SSSS. The S2SUCNN running time is from 310.13 to 315.55 seconds, while the SSSS running time is from 1271.31 to 1291.66 seconds, for all datasets.

Table 3.7. Running time (in seconds) for full-resolution sharpening of all methods. Best results are in bold.

Method/Dataset	Australia	Iceland	USA	Vietnam
ATPRK	**8.70**	**8.60**	**8.61**	**8.81**
SupReME	11.95	13.56	12.42	11.96
S2Sharp	21.16	19.52	23.36	23.75
SSSS	1276.63	1271.31	1291.66	1274.06
S2SUCNN	314.77	315.55	312.16	310.13



(a) Reference *(b)* ATPRK *(c)* SupReME *(d)* S2Sharp

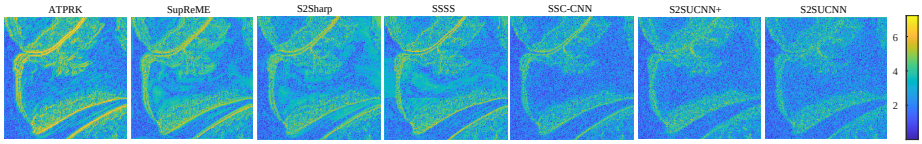


(e) SSSS *(f)* SSC-CNN *(g)* S2SUCNN+ *(h)* S2SUCNN

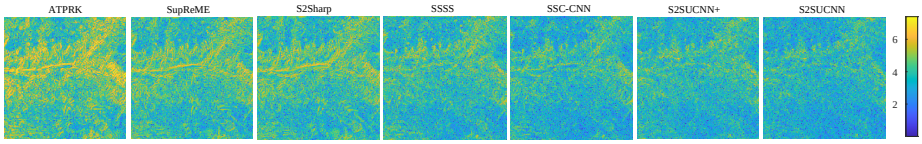
Figure 3.7. Reduced-resolution 20 m bands sharpening results for Vietnam dataset, the gray scale image shows a part of band 12.

3.4 Conclusions

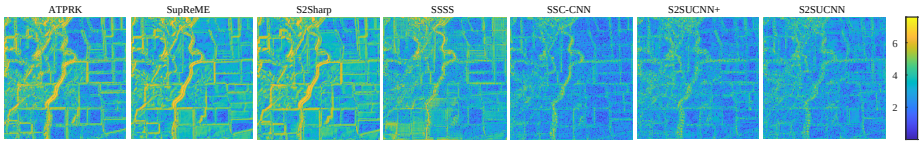
A new method called S2SUCNN has been proposed to address the S2 sharpening problem where the 20 m and 60 m are fused with the 10 m bands resulting in all bands at 10 m resolution. The innovation of the proposed method is that it is an unsupervised DL-based method that works directly on full-resolution of S2 data. Moreover, S2SUCNN uses only a single CNN for sharpening both the 20 m and the 60 m bands, and investigates the MTF-based degradation model as a network layer. Those aspects have been verified to improve the results since it takes advantage of multitask learning and the high correlation of all S2 bands. However, the proposed method requires an early stopping to obtain good results, which is manually chosen. Therefore, to automatically determine when to stop the iteration is a question that will be concerned in the future work.



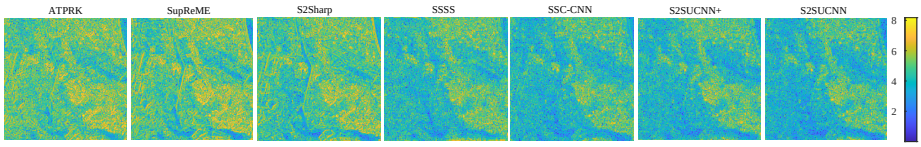
(a) Australia dataset, band 7



(b) Iceland dataset, band 8a

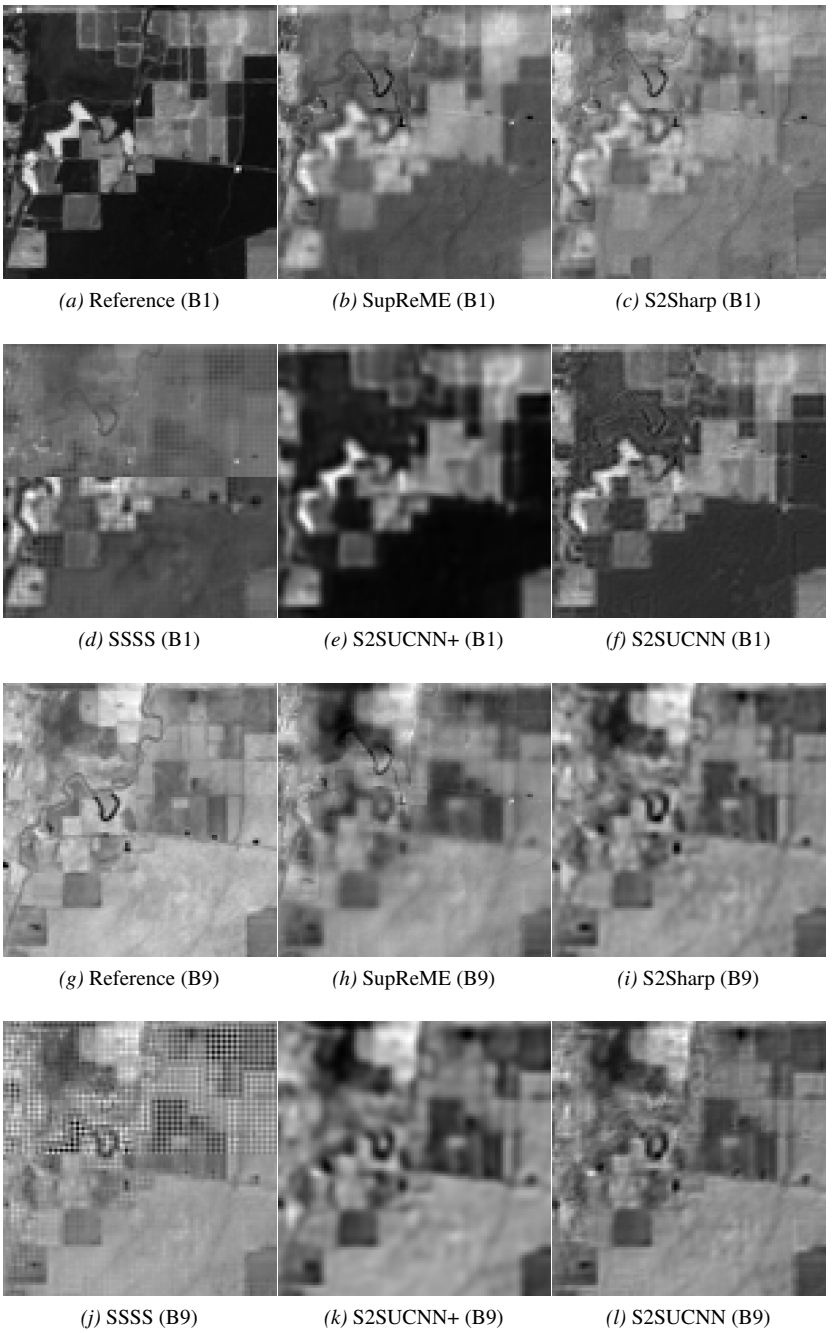


(c) USA dataset, band 11



(d) Vietnam dataset, band 12

Figure 3.8. Reduced-resolution 20 m bands sharpening results for four datasets. The images are residual structure shown in logarithm scale, log(1+$\left|\mathbf{X}-\widehat{\mathbf{X}}\right|$).



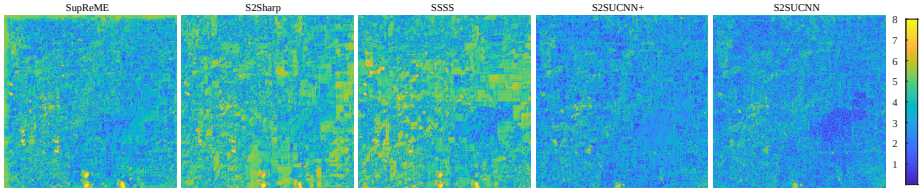
(a) Reference (B1) *(b)* SupReME (B1) *(c)* S2Sharp (B1)

(d) SSSS (B1) *(e)* S2SUCNN+ (B1) *(f)* S2SUCNN (B1)

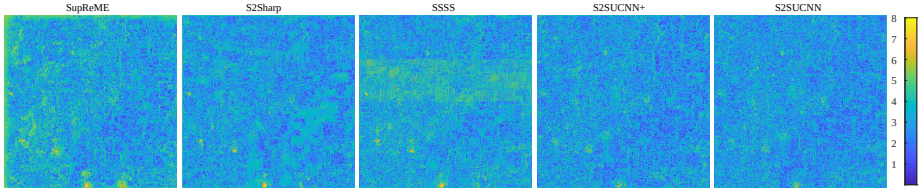
(g) Reference (B9) *(h)* SupReME (B9) *(i)* S2Sharp (B9)

(j) SSSS (B9) *(k)* S2SUCNN+ (B9) *(l)* S2SUCNN (B9)

Figure 3.9. Reduced-resolution 60 m bands sharpening results for USA-60 dataset, the gray scale images shown parts of bands 1 and 9.

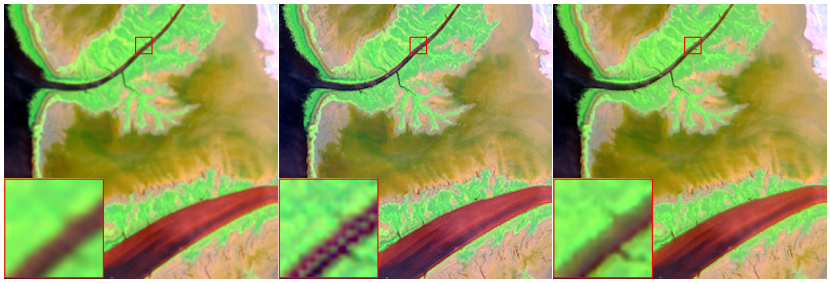


(a) USA-60 dataset, band 1

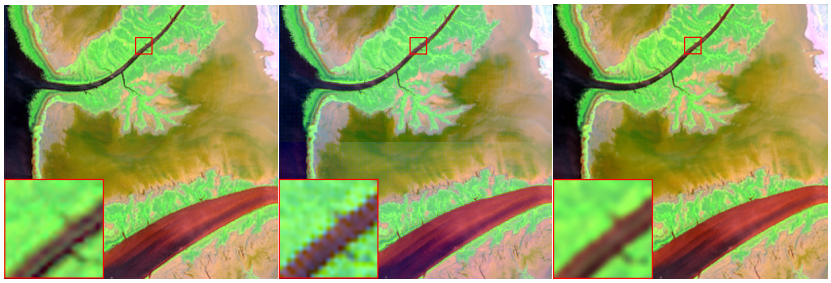


(b) USA-60 dataset, band 9

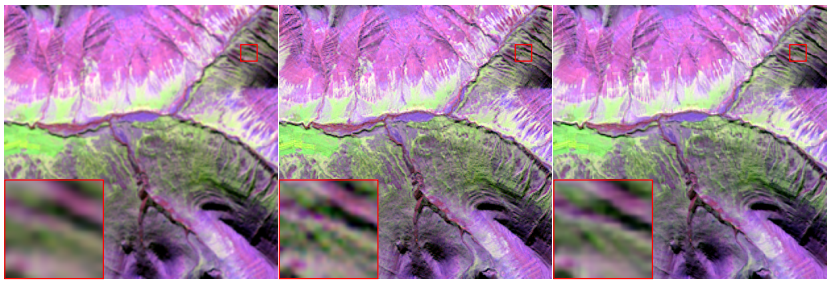
Figure 3.10. Reduced-resolution 60 m bands sharpening results for USA-60 dataset. The images are residual structure shown in logarithm scale, log(1+$\left|\mathbf{X}-\widehat{\mathbf{X}}\right|$).



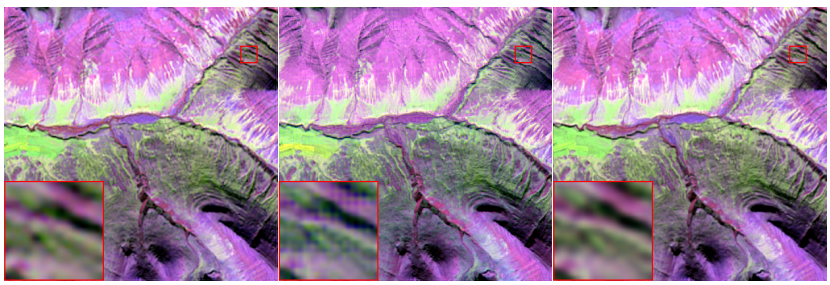
(a) bicubic *(b)* ATPRK *(c)* SupReME



(d) S2Sharp *(e)* SSSS *(f)* S2SUCNN

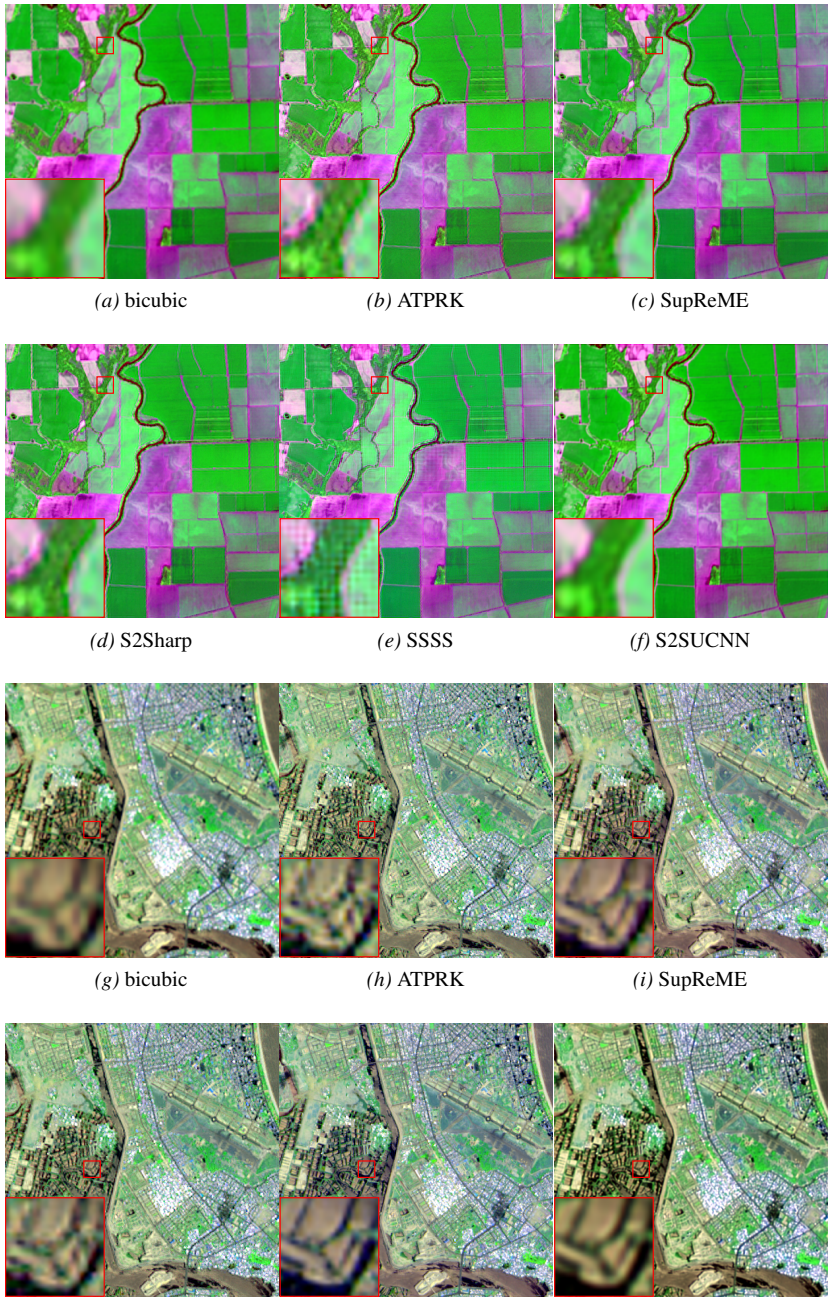


(g) bicubic *(h)* ATPRK *(i)* SupReME



(j) S2Sharp *(k)* SSSS *(l)* S2SUCNN

Figure 3.11. Full-resolution 20 m bands sharpening results. The images (410×410 *pixels) are shown in false-color images using bands 12, 8a, and 5 as R, G, and B channels. (a-f) Australia dataset, (g-l) Iceland dataset.*



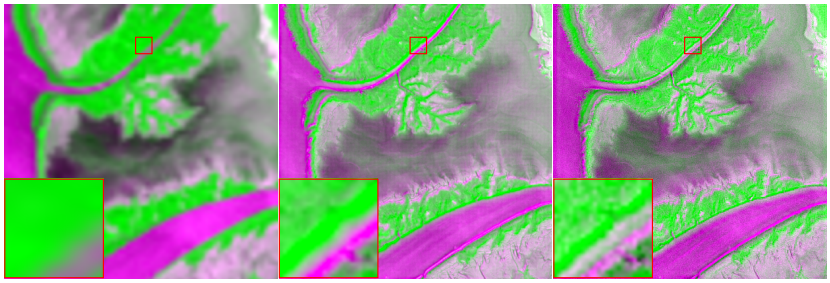
(a) bicubic *(b)* ATPRK *(c)* SupReME

(d) S2Sharp *(e)* SSSS *(f)* S2SUCNN

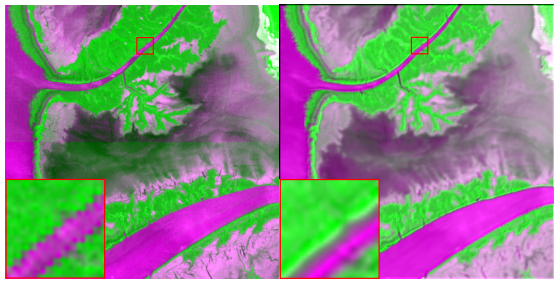
(g) bicubic *(h)* ATPRK *(i)* SupReME

(j) S2Sharp *(k)* SSSS *(l)* S2SUCNN

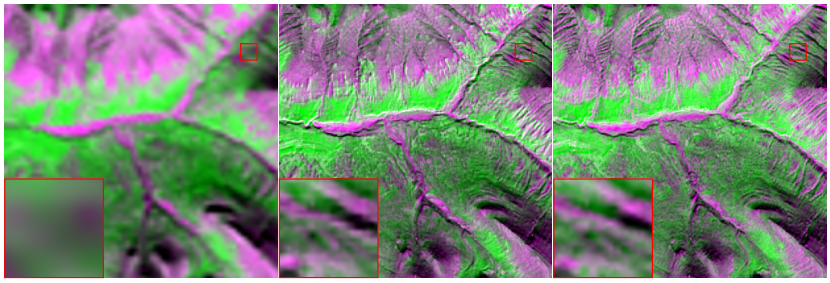
Figure 3.12. Full-resolution 20 m bands sharpening results. The images (410×410 *pixels) are shown in false-color images using bands 12, 8a, and 5 as R, G, and B channels. (a-f) USA dataset, (g-l) Vietnam dataset.*



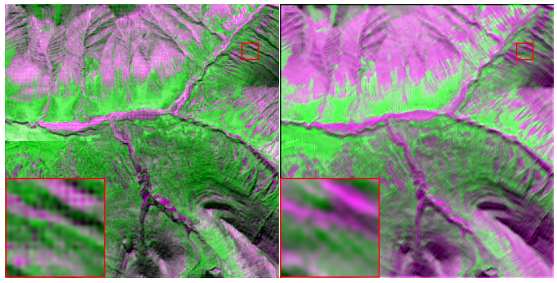
(a) bicubic *(b)* SupReME *(c)* S2Sharp



(d) SSSS *(e)* S2SUCNN

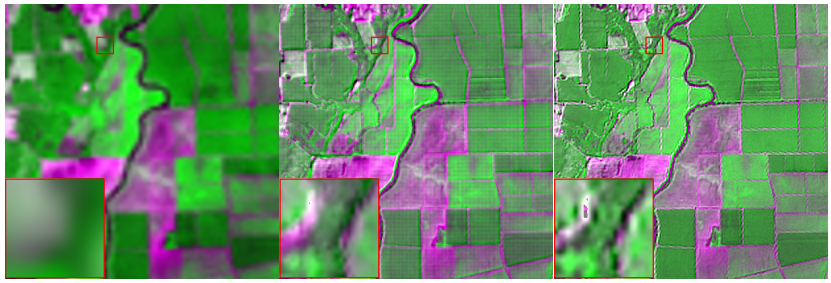


(f) bicubic *(g)* SupReME *(h)* S2Sharp

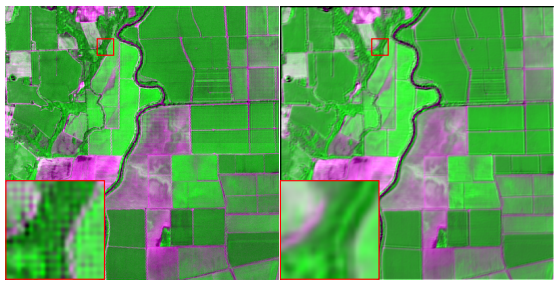


(i) SSSS *(j)* S2SUCNN

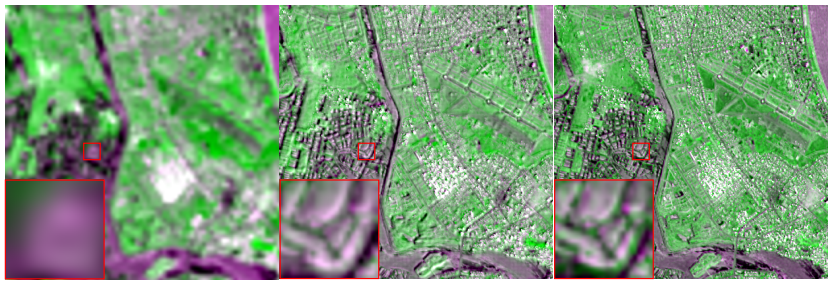
Figure 3.13. Full-resolution 60 m bands sharpening results. The images (410×410 *pixels) are shown in false-color images using bands 12, 8a, and 5 as R, G, and B channels. (a-e) Australia dataset, (f-j) Iceland dataset.*



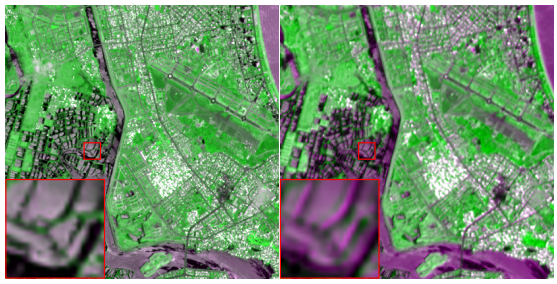
(a) bicubic *(b)* SupReME *(c)* S2Sharp



(d) SSSS *(e)* S2SUCNN



(f) bicubic *(g)* SupReME *(h)* S2Sharp



(i) SSSS *(j)* S2SUCNN

Figure 3.14. Full-resolution 60 m bands sharpening results. The images (410×410 *pixels) are shown in false-color images using bands 12, 8a, and 5 as R, G, and B channels. (a-e) USA dataset, (f-j) Vietnam dataset.*

CHAPTER 4

DEEP SURE FOR UNSUPERVISED REMOTE SENSING IMAGE FUSION

Recently, several RS image fusion methods [60], [128]–[130], [134] based on the DIP [63] have been proposed. Those methods bridge the gap between the model-based and the DL-based methods. However, the main limitation of those methods is that under the noisy scenario they suffer from overfitting problem, and their performance heavily depends on the CNN structure. To overcome this limitation, a fusion method based on SURE [137], [158] is proposed and described in details in this chapter. The novel point of the proposed SURE-based method is to derive a loss function using SURE and a linear operator that maps a LR image to its HR space. Optimizing a CNN with this SURE loss function significantly avoids overfitting. Three representative fusion problems (e.g., MS-HS fusion, S2 sharpening, and pansharpening) are addressed to demonstrate the proposed method, where the back-projection (BP) operator is chosen as an example of the linear operator in the SURE loss function. Experimental results verify that the proposed method yields high quality fused images and outperforms the competitive methods. Hereafter in this chapter the proposed, method is call SURE for short. Codes of the proposed SURE method are available at `https://github.com/hvn2/Deep-SURE-Fusion`.

4.1 Problem formulation and motivation

In RS, many researchers treated image fusion as separate problems, such as pansharpening [71], hypersharpening [77], and hyperspectral pansharpening [75]. There, MS-HS fusion and S2 sharpening are considered as hypersharpening which is defined as a fusion technique where the spatial source (i.e, the HR image to be fused) is a set of bands. However, image fusion in RS can be formulated as an unique problem that is to estimate the missing spatial details of the LR image, from the HR image to be fused (also called the guided image). Mathematically, the fusion problem is expressed as both spatial and spectral degradation as below

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon}, \tag{4.1}$$

$$\mathbf{G} = \mathbf{X}\mathbf{R} + \mathbf{N}_g. \tag{4.2}$$

Here, the spatial degradation (4.1) is the same as (3.1), where the degradation operator $\mathbf{H}$ includes the blurring ($\mathbf{B}$) and downsampling ($\mathbf{M}$) operators (i.e., $\mathbf{H} = \mathbf{M}\mathbf{B}$). The

spectral degradation (4.2) is characterized by the spectral response function (SRF), and is represented as a matrix $\mathbf{R}$. The spectral degradation is the process that decimates d bands of the original HR image $\mathbf{X} \in \mathbb{R}^{M \times d}$ to the HR image to be fused, $\mathbf{G} \in \mathbb{R}^{M \times D} (D < d)$. The noise matrix, $\mathbf{N}_g = [\mathbf{n}_{ig}]_{i=1}^{D}$, is Gaussian noise, i.e, $\mathbf{n}_{ig} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Omega}_{ig}), i = 1, \ldots, D$. The information for constructing the blurring matrix, $\mathbf{B}$, (using the sensors' PSFs) and the SRF matrix, $\mathbf{R}$, is either known (e.g., given by the manufacturer) or unknown. In the latter case, $\mathbf{B}$ and $\mathbf{R}$ can be estimated by using the observation data [93], [94]. Here, $\mathbf{B}$ and $\mathbf{R}$ are assumed to be known.

Recently, the RS image fusion methods using the DIP have been proposed. The earliest method [60] simply extended the vanilla DIP [63] for hyperspectral data. Other works have improved the DIP results by adding explicit constraints and designing better CNN structure. For example, the spectral degraded constraint was used in [128], [159], and the unmixing constraint and the spectral-spatial attention mechanism [95] were added to the DIP in [129], [134], [135]. However, those methods have not addressed the fusion of noisy data, which is usually dominant in the real HSIs (e.g., the IP and UB datasets). Under the high noise scenario, those DIP-based methods are susceptible to overfitting problem in which the fused image tends to fit the observed noisy data. To overcome the overfitting problem, the proposed method employs a loss function based on SURE incorporating with a linear operator. Here, the linear operator maps a LR image to its HR version, and is assumed that the HR image obtained by this operator is at least better than one obtained by a simple interpolation method (e.g., bicubic). Since SURE is the unbiased estimate of the MSE, and is computed without using ground truth, training a CNN with the SURE-based loss function avoids overfitting and results in the fused image that fits the ground truth. Additionally, the linear operator improves results because it can be considered a pre-processing operator that assists the fusion algorithm. Main contributions of the proposed method are summarized as follows:

- A new loss function based on SURE is derived to optimize a CNN without ground truth. The proposed method is an unsupervised DL-based and overcomes the overfitting problem in many DIP-based image fusion methods.
- The loss function includes SURE and a linear operator which is a pre-processing operator mapping an LR to its HR space. As an example, the BP [140] is chosen as the linear operator, because the BP is constructed using the sensors' PSFs, and is computed very fast. Experimental results show that the SURE loss function with BP gives better performance than one with a simple upsampling operator.
- The SURE with BP loss function is derived in a more straightforward way than the GSURE [138], [160], and is specified for RS image fusion rather than RGB image super-resolution using GSURE. Experimental results for three fusion representatives (MS-HS fusion, S2 sharpening, and pansharpening) verify that the method is effective and outperforms the competed methods.

4.2 The SURE loss function for unsupervised DL-based RS image fusion

4.2.1 Deep image prior and back-projection

The proposed method is inspired by the DIP [63] which was discussed in Section 2.1. Using DIP, the fused image is obtained at the output of a CNN optimized by using the following loss function

$$\mathcal{L}_{\text{DIP}}(\boldsymbol{\theta}) = \|\mathbf{y} - \mathbf{H}f_{\boldsymbol{\theta}}(\mathbf{z})\|_2^2, \tag{4.3}$$

where $\mathbf{z}$ is the CNN input, which is formed by concatenating the LR image, $\mathbf{y}$, and the guided image, $\mathbf{g}$, and $f_{\boldsymbol{\theta}}(\cdot)$ is the CNN output. In DIP, optimization of the loss function tends to make the degraded version of the fused image close to the LR image. One idea to leverage the DIP performance is the back-projection DIP (BP-DIP) [161] which optimizes a CNN using the loss function

$$\mathcal{L}_{\text{BP-DIP}}(\boldsymbol{\theta}) = \|\mathbf{P}(\mathbf{y} - \mathbf{H}f_{\boldsymbol{\theta}}(\mathbf{z}))\|_2^2, \tag{4.4}$$

where

$$\mathbf{P} = \mathbf{H}^{\dagger} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}. \tag{4.5}$$

Optimization of the loss function in BP-DIP tends to keep the BP versions of the LR and the degraded version of the fused images close. BP-DIP outperformed DIP in the low noise fusion case [140], [161]. However, the main issues of both DIP and BP-DIP are that their performances depend on the CNN structure, and overfitting in the fusion problem with noisy data. To make the DIP and BP-DIP efficient, one should carefully design the CNN structure, and use an early stopping schedule to avoid overfitting. However, in a real application, it is hard to choose a proper stopping point, because there is no criterion to make a stopping decision.

4.2.2 Deep SURE for RS image fusion

As discussed above, DIP and BP-DIP have overfitting problem, since optimizing (4.3) and (4.4) for long iterations leads to results that replicated the LR images. To avoid overfitting, the loss function should ideally involve the ground truth $\mathbf{x}$, such as

$$R = \mathbb{E}\|\mathbf{P}(\mathbf{H}\mathbf{x} - \mathbf{H}f_{\boldsymbol{\theta}}(\mathbf{z}))\|_2^2, \tag{4.6}$$

where $\mathbb{E}$ is the expectation operator, $\mathbf{P}$ is a linear operator that maps a LR image to its HR space.

Unfortunately, R is unable to compute since $\mathbf{x}$ is unknown. To overcome this problem, R is exchanged to its unbiased estimate which is based on SURE. The SURE-based unbiased estimate of (4.6) is derived as following with the main goal is to eliminate the dependency of $\mathbf{x}$ from (4.6).

Adding and subtracting $\mathbf{u} = \mathbf{P}\mathbf{y}$ in (4.6), we have

$$\begin{aligned} R &= \mathbb{E}\|(\mathbf{u} - \mathbf{P}\mathbf{H}\mathbf{x}) - (\mathbf{u} - \mathbf{P}\mathbf{H}f_{\boldsymbol{\theta}}(\mathbf{z}))\|_2^2 \\ &= \mathbb{E}\|\mathbf{n} - \mathbf{e}\|_2^2 \\ &= \mathbb{E}\|\mathbf{e}\|_2^2 + \mathbb{E}\|\mathbf{n}\|_2^2 - 2\mathbb{E}[\mathbf{e}^T\mathbf{n}]. \end{aligned} \tag{4.7}$$

where $\mathbf{n} = \mathbf{u} - \mathbf{PHx} = \mathbf{P}(\mathbf{y} - \mathbf{Hx})$ and $\mathbf{e} = \mathbf{u} - \mathbf{PH}f_{\boldsymbol{\theta}}(\mathbf{z})$. Since, from (4.1) $\mathbf{y}$ is a Gaussian random variable with mean $\mathbf{Hx}$ and covariance $\boldsymbol{\Omega}$, $\mathbf{u}$ is also a Gaussian random variable with mean $\boldsymbol{\mu}_P = \mathbf{PHx}$ and covariance $\boldsymbol{\Omega}_P = \mathbf{P}\boldsymbol{\Omega}\mathbf{P}^T$. We also have that $\|\mathbf{e}\|_2^2$ is an unbiased estimate of $\mathbb{E}\|\mathbf{e}\|_2^2$, and

$$\mathbb{E}\|\mathbf{n}\|_2^2 = \mathbb{E}[\mathbf{n}^T\mathbf{n}] = \text{tr}(\mathbb{E}[\mathbf{n}\mathbf{n}^T]) = \text{tr}(\boldsymbol{\Omega}_P),$$

where $\text{tr}(\cdot)$ is the matrix trace. Thus, the difficulty is raised from the term $\mathbb{E}\big[\mathbf{e}^T\mathbf{n}\big]$. By using the Stein's lemma [137] we get

$$\begin{aligned}
\mathbb{E}\big[\mathbf{e}^T\mathbf{n}\big] &= \mathbb{E}\big[\mathbf{e}^T\mathbf{P}(\mathbf{y} - \mathbf{Hx})\big] \\
&= \mathbb{E}\big[\text{tr}\big(\boldsymbol{\Omega}\frac{\partial\mathbf{P}^T\mathbf{e}}{\partial\mathbf{y}}\big)\big] \\
&= \mathbb{E}\big[\text{tr}\big(\boldsymbol{\Omega}\frac{\partial\mathbf{P}^T\mathbf{e}}{\partial\mathbf{u}}\frac{\partial\mathbf{u}}{\partial\mathbf{y}}\big)\big] \\
&= \mathbb{E}\big[\text{tr}\big(\boldsymbol{\Omega}\frac{\partial\mathbf{P}^T\mathbf{e}}{\partial\mathbf{u}}\mathbf{P}\big)\big] \\
&= \mathbb{E}\big[\text{tr}\big(\boldsymbol{\Omega}_P\frac{\partial\mathbf{e}}{\partial\mathbf{u}}\big)\big] \\
&= \mathbb{E}\big[\text{tr}\big(\boldsymbol{\Omega}_P\frac{\partial(\mathbf{u} - \mathbf{PH}f_{\boldsymbol{\Theta}}(\mathbf{z}))}{\partial\mathbf{u}}\big)\big] \\
&= \text{tr}(\boldsymbol{\Omega}_P) - \mathbb{E}\big[\text{tr}\big(\boldsymbol{\Omega}_P\mathbf{PH}\frac{\partial f_{\boldsymbol{\Theta}}(\mathbf{z})}{\partial\mathbf{u}}\big)\big].
\end{aligned}$$

Substituting this in (4.7) and use the fact that, for any random variable $\mathbf{z}$, $\mathbf{z}$ is an unbiased estimator of $\mathbb{E}[\mathbf{z}]$, an unbiased risk estimate for R is given by

$$\hat{R} = \|\mathbf{P}(\mathbf{y} - \mathbf{H}f_{\boldsymbol{\theta}}(\mathbf{z}))\|_2^2 + 2\text{tr}\big(\boldsymbol{\Omega}_P\mathbf{PH}\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{z})}{\partial\mathbf{u}}\big) - \text{tr}(\boldsymbol{\Omega}_P).$$

To train (optimize) a CNN using $\hat{R}$ as a loss function, the term $\text{tr}(\boldsymbol{\Omega}_P)$ which does not depend on the network parameters can be ignored. The SURE loss function to train a CNN is written as

$$\mathcal{L}_{\text{SURE}}(\boldsymbol{\theta}) = \|\mathbf{P}(\mathbf{y} - \mathbf{H}f_{\boldsymbol{\theta}}(\mathbf{z}))\|_2^2 + 2\text{tr}\big(\boldsymbol{\Omega}_P\mathbf{PH}\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{z})}{\partial\mathbf{u}}\big) \tag{4.8}$$

The SURE loss function (4.8) is meaningful. Firstly, it not only overcomes the overfitting problem but also allows one to train a CNN in an unsupervised manner, because no ground truth is required. Secondly, the trace term in (4.8) can be viewed as correction for overfitting and this can clearly been seen, for example, in Fig. (4.4) below. Finally, another insight in the trace term is that it arises from the term $\mathbb{E}[\mathbf{e}^T\mathbf{n}]$. In the supervised setting $\mathbf{e}$ is a function of training data, and $\mathbf{n}$ is a function of independent validation data. Due to independence $\mathbb{E}[\mathbf{e}^T\mathbf{n}] = \mathbb{E}[\mathbf{e}^T]\mathbb{E}[\mathbf{n}] = 0$. Therefore, in the supervised setting the trace term is zero indicating no overfitting.

The trace term in (4.8) does not have a close form expression because the CNN function is highly non-linear and has too many parameters. It is possible to compute the trace term by using the automatic differentiation of the deep learning framework [142],

e.g., Tensorflow and PyTorch. However, the exact computation of the trace term using DL automatic differentiation package consumes a lot of memory and running time since it builds a computational graph for every single pixel. This problem is a barrier of SURE because the trace term needs to be computed in every training iteration. To overcome this problem the MC SURE [143] can be used to approximate the trace term owing to its simplicity and fast execution. The MC SURE computation of the trace term in (4.8) is given by

$$\mathrm{tr}\big(\mathbf{\Omega}_P\mathbf{P}\mathbf{H}\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{z})}{\partial \mathbf{u}}\big) \approx \mathbf{b}^T\mathbf{\Omega}_P\mathbf{P}\mathbf{H}\frac{f_{\boldsymbol{\theta}}(\mathbf{z}+\beta\mathbf{b}) - f_{\boldsymbol{\theta}}(\mathbf{z})}{\beta}, \tag{4.9}$$

where $\mathbf{b}$ is a vector drawn from a zero-mean, unit variance Gaussian distribution, and β is a small number.

4.2.3 Computation of the BP

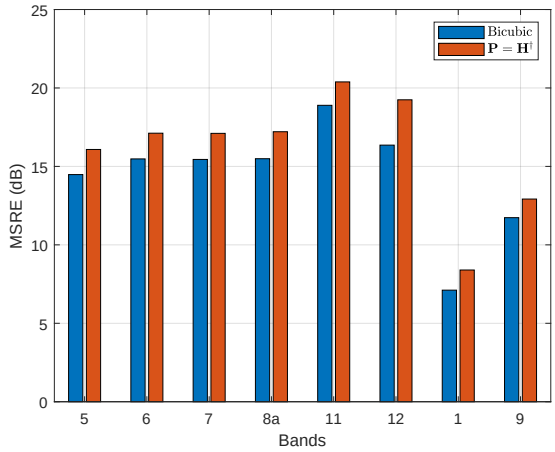


Figure 4.1. Upsampling the LR bands (SNR $= 30$ dB) of the S2 simulated APEX dataset using bicubic and BP. The results are given in MSRE in decibels between the upsampled and reference images.

We are interested in using BP given in (4.5) as a linear operator $\mathbf{P}$. The reason is that the BP, $\mathbf{P} = \mathbf{H}^\dagger$, applied to a LR image gives better result than a simple interpolation method, such as bicubic. This is shown in Fig. 4.1 where the LR bands of the simulated S2 APEX dataset (see Section 4.4.1 for the simulation of APEX dataset) are upsampled by using bicubic and BP. Clearly, BP gives higher MSRE than bicubic. More detail analysis of different kinds of linear operator $\mathbf{P}$ with SURE is given in Sections 4.3.2 and 4.4.2 below. Additionally, computation of BP is very fast by using the poly-phase decomposition and the fast Fourier transform (FFT) described as following.

The computation of BP (4.5) involves computing a transpose term, $\mathbf{H}^T$, and an inverse term of the blurring and downsampling matrices, $(\mathbf{H}\mathbf{H}^T)^{-1}$, consequently. The first term can be interpreted as upsampling by inserting zeroes between samples ($\mathbf{M}^T$)

and filtering ($\mathbf{B}^T$) with the blurring kernel $\overline{\mathcal{F}(\mathbf{h})}$, where $\mathbf{h} = [\mathbf{h}^{(i)}]_{i=1}^d$ is the PSF vector of $\mathbf{B}$ and is constructed by stacking the ith band PSF vector, $\mathbf{h}^{(i)}$, on top of each other. The notation $\mathcal{F}(\cdot)$ means FFT and $\overline{\mathcal{F}(\cdot)}$ is its conjugation. The second term can be interpreted as the inversion of sequence operators of upsampling ($\mathbf{M}^T$), filtering ($\mathbf{B}\mathbf{B}^T$) and downsampling ($\mathbf{M}$). These sequence operators can be efficiently computed using the polyphase decomposition of the filter $\mathbf{B}\mathbf{B}^T$ [162]. Note that $\mathbf{H}\mathbf{H}^T$ is ill-conditioned (i.e., having many singular values close to zero). It is needed to regularize zero singular values by a small number α, which is manually chosen as a constant of 10^{-3} for all bands. The computation for $\mathbf{P} = \mathbf{H}^\dagger$ applied to an LR image $\mathbf{Y}$ is summarized in Algorithm 3.

With $\mathbf{P} = \mathbf{H}^\dagger$, the SURE loss function (4.8) becomes

$$\mathcal{L}_{\text{SURE}}(\boldsymbol{\Theta}) = \mathcal{L}_{\text{BP-DIP}}(\boldsymbol{\Theta}) + 2\text{tr}\big(\boldsymbol{\Omega}_{\boldsymbol{P}} \frac{\partial f_{\boldsymbol{\Theta}}(\mathbf{z})}{\partial \mathbf{u}}\big), \tag{4.10}$$

since $\boldsymbol{\Omega}_P \mathbf{H}^\dagger \mathbf{H} = \boldsymbol{\Omega}_P$. Although the objectives of (4.10) and the GSURE [138], [160] are similar, the derivation of (4.10) is simpler than GSURE. Remember that for this case ($\mathbf{P} = \mathbf{H}^\dagger$) the MC SURE computation (4.9) has to be modified accordingly. The MC SURE approximation of the trace term in (4.10) involves $\boldsymbol{\Omega}_P = \mathbf{H}^\dagger \boldsymbol{\Omega} (\mathbf{H}^\dagger)^T$ which can be computed using the poly-phase decomposition technique above. In practice, the noise covariance matrix $\boldsymbol{\Omega}$ is unknown. However, it can be estimated using the HySure [144] algorithm or the median absolute deviation of the details sub-band (HH) using 2-D wavelet decomposition [163].

Algorithm 3: Fast computation for linear inverse filter $\mathbf{P} = \mathbf{H}^\dagger$ applied to LR image $\mathbf{Y}$

Input: LR image $\mathbf{Y} = [\mathbf{y}_i]_{i=1}^d$,
blurring kernel $\mathbf{h} = [\mathbf{h}^{(i)}]_{i=1}^d$,
up/downsampling factor $\mathbf{r} = [r_i]_{i=1}^d$,
regularized value $\boldsymbol{\alpha} = [\alpha_i]_{i=1}^d = 10^{-3}$
Output: Filtered image $\mathbf{X}_F = [\mathbf{x}_{iF}]_{i=1}^d$

`/* Repeating for all bands */`

for *i=1:d* **do**
- $\tilde{\mathbf{h}}^{(i)} = \mathcal{F}^{-1}\{|\mathcal{F}(\mathbf{h}^{(i)})|^2\}$
- $\mathbf{h}_0^{(i)} =_{(\downarrow r_i)} \tilde{\mathbf{h}}^{(i)}$ `// Downsampling by` r_i
- $\mathbf{y}_{iF} = \mathcal{F}^{-1}\{\frac{\mathcal{F}(\mathbf{y}_i)}{\mathcal{F}(\mathbf{h}_0^{(i)}) + \alpha_i}\}$
- $\mathbf{x}_{iUP} = \mathbf{M}_i^T \mathbf{y}_{iF}$ `// Upsampling by inserting zeros between samples`
- $\mathbf{x}_{iF} = \mathcal{F}^{-1}\{\mathcal{F}(\mathbf{x}_{iUP})\overline{\mathcal{F}(\mathbf{h}^{(i)})}\}$

end

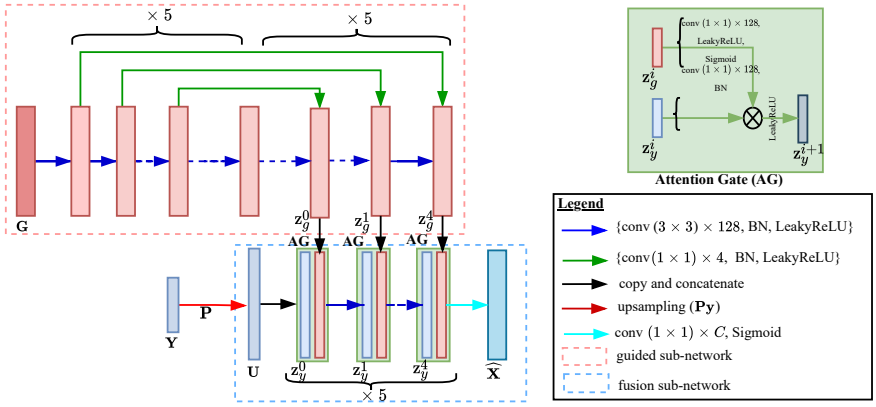


Figure 4.2. Fusion CNN architecture. The rectangles represent the images or latent feature maps at the corresponding layers, and $\times 5$ means 5 times repetition.

4.2.4 Network structure

As one of the method inspired by the DIP approach, the proposed method using the loss function (4.8) works with a wide range of CNNs [63], [153]. In this section, a CNN with skip connections [164] and attention gates (AG) [95], [128], [134] is used to demonstrate the proposed method. The network architecture is depicted in Fig. 4.2, where it is composed of a guided sub-network and a fusion sub-network. The guided image $\mathbf{G}$ is propagated through the guided sub-network, which is an autoencoder-like network with skip connections to extract multilevel guided feature maps. The guided feature maps merge with the feature maps at the fusion sub-network using several AGs. Since the autoencoder-like network uses skip connections, the fusion sub-network requires only the guided feature maps at the decoder to be fused in every AG. Denoting the guided feature maps and the fusion feature maps at layer ith as $\mathbf{z}_g^{(i)}$ and $\mathbf{z}_y^{(i)}$, respectively, the output of each AG is

$$\mathbf{z}_y^{(i+1)} = \text{LeakyReLU}\big(F_g(\mathbf{z}_g^{(i)}) \odot F_y(\mathbf{z}_y^{(i)})\big),$$

where $\odot$ represents an element-wise multiplication. $F_g(\cdot)$ is a function of 128 convolution filters with filter size of 1 (conv$(1 \times 1) \times 128$), followed by a LeakyReLU and a Sigmoid layer. $F_y(\cdot)$ is a function of conv$(1 \times 1) \times 128$ followed by a batchnormalization (BN) layer. Note that AG is a multiplicative transformation that scales the guided feature maps with different weights. Therefore, the guided feature maps are better aligned with the fusion feature maps yielding a better fusion result [95], [128].

The network is implemented in Pytorch using Adam optimizer [136] with a learning rate of 0.01.

4.3 MS-HS fusion: Experimental results

The first example to demonstrate the proposed method is MS-HS fusion. In MS-HS fusion, the guided image is the MSI, and the LR image is the LR HSI. The ideal objective function to be optimized is

$$R_{\text{MH}} = \mathbb{E}\|\mathbf{P}(\mathbf{H}\mathbf{x} - \mathbf{H}f_{\boldsymbol{\theta}}(\mathbf{z}))\|_2^2 + \lambda\mathbb{E}\|(\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z}) - \mathbf{X})\mathbf{R}\|_F^2,$$

where $\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z})$ is the CNN output presented in matrix form, and λ is a non-negative number controlling the trade-off between two terms. However, $\mathbf{X}$ is unknown, and R_{MH} is not computable. By using SURE in (4.10), R_{MH} is exchanged to its unbiased estimate, as follows,

$$\mathcal{L}_{\text{SURE(MH)}}(\boldsymbol{\theta}) = \hat{R}_{\text{MH}} = \hat{R}_{\text{H}} + \lambda\hat{R}_{\text{M}}, \tag{4.11}$$

where

$$\hat{R}_{\text{H}} = \|\mathbf{P}(\mathbf{y} - \mathbf{H}f_{\boldsymbol{\theta}}(\mathbf{z}))\|_2^2 + 2\mathbf{b}^T\boldsymbol{\Omega}_P\frac{f_{\boldsymbol{\theta}}(\mathbf{z}+\beta\mathbf{b}) - f_{\boldsymbol{\theta}}(\mathbf{z})}{\beta},$$

and

$$\hat{R}_{\text{M}} = \|\mathbf{g} - \text{vec}[\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z})\mathbf{R}]\|_2^2 + \mathbf{b}^T\boldsymbol{\Omega}_g\text{vec}\big(\frac{\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z}+\beta\mathbf{b}) - \mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z})}{\beta}\mathbf{R}\big).$$

Here $\mathbf{g}$ is the guided image represented as a vector and $\boldsymbol{\Omega}_g$ is a block-diagonal matrix where each element is $\boldsymbol{\Omega}_{ig}$.

4.3.1 Datasets and evaluation metrics

(a) PU: MSI *(b)* PU: LR HSI *(c)* DC: MSI *(d)* DC: LR HSI

Figure 4.3. The PU and DC datasets (high noise). The MSIs are shown in false color images using bands 2, 1, and 3 as the Red (R), Green (G), and Blue (B) channels. The LR images are show in natural color images using a HSI to RGB color rendering method [165].

The PU and DC datasets (referring to Appendix for the dataset description) are used as the synthesized datasets to evaluated the fusion methods. The simulated data are generated using a procedure described in [93]. First, the data are pre-processed by removing some noisy bands, cropping parts of the original data, and normalizing the intensity values between 0 and 1, resulting in $200 \times 200 \times 93$ and $200 \times 200 \times 191$

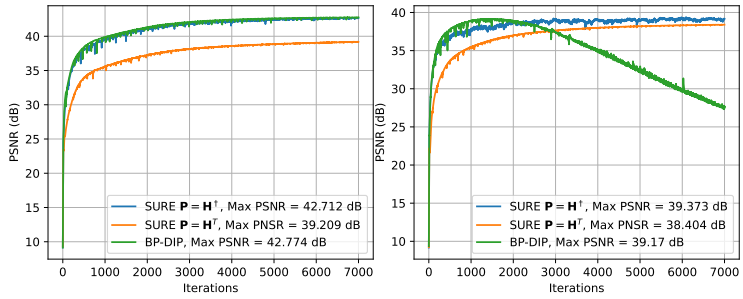
data cubes for the PU and DC datasets, respectively. Those data cubes are used as the ground truth (original HR HSI). Then, the LR HSI is generated by applying the spatial degradation (4.1) to the HR HSI. The spatial degradation is a process of filtering with a Gaussian low pass filter having a standard deviation $\sigma = 2.0$ and support 5×5 and followed by downsampling with a factor of 4, band by band. The MSI is generated by applying the spectral degradation (4.2) to the HR HSI where the spectral response matrix $\mathbf{R}$ is constructed usinng the IKONOS SRF [93]. One low level noise and one high level noise are added to the data, which is described as follows:

- Low noise: Isotropic Gaussian noise is added to LR HSI and MSI such that the signal-to-noise ratio (SNR) is SNR $= 30$ dB and SNR $= 40$ dB, respectively.
- High noise: Band-dependent Gaussian noise is added to LR HSI, $\mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \sigma_i^2 \mathbf{I})$, where $i = 1, ..., d$, and σ_i is randomly chosen between 0 and 0.1 resulting in SNR $= 11.56$ dB and SNR $= 13.75$ dB for the PU, and DC datasets, respectively. For the MSI, the additive noise is isotropic Gaussian noise with SNR $= 40$ dB.

The noisy MSI and LR HSI of the PU and DC datasets are shown in Fig. 4.3 for the high noise case.

The numerical metrics used to evaluate fusion performance are the PSNR in decibels, ERGAS, and SAM in degrees, which are computed with respect to the original HR HSIs without noise added. Those evaluation metrics are detailed in the Appendix.

4.3.2 Validation of SURE for MS-HS fusion



(a) Low noise *(b)* High noise

Figure 4.4. PSNR curves in training for MS-HS image fusion using different loss functions for the PU dataset.

This section verifies that optimizing a CNN with the SURE loss function (4.11) prevents overfitting. Additionally, two kinds of the linear operator $\mathbf{P}$, i.e., BP operator ($\mathbf{P} = \mathbf{H}^\dagger$) and an upsampling filtering operator ($\mathbf{P} = \mathbf{H}^T$), are analysed with the SURE loss function (4.11). Fig. 4.4 shows the experimental results for the PU dataset where the same CNN as in Fig. 4.2 is trained by using the BP-DIP loss function (4.4), the SURE loss functions (4.11) with $\mathbf{P} = \mathbf{H}^T$ and with $\mathbf{P} = \mathbf{H}^\dagger$, respectively. The

hyperparameter λ is manually chosen as $\lambda = 0.1$ for both SURE and BP-DIP loss functions.

BP-DIP significantly overfits in the high noise scenario, as can be seen in Fig. 4.4b where the PSNR curve reaches its peak at 1200 iterations and falls down quickly after that. To obtain good results for BP-DIP, training should stop at the point giving highest PSNR. Due to the lack of ground truth in real applications, the PSNR is not computable, and there is no criterion to stop training in BP-DIP. In contrast, the PSNR curves of both SURE loss functions rise and saturate at 4000 iterations, clearly shows that overfitting is avoided. Therefore, the proposed SURE-based method is feasible for real applications because it does not depend on an early stopping schedule to avoid overfitting. In a low noise case (Fig. 4.4a), the performance of SURE and BP-DIP losses are similar. This is not surprising since the BP-DIP and SURE losses are different up to a value (the trace term) which depends on the noise variance and is very small in the low noise case. Ideally, BP-DIP and SURE loss functions are exactly the same in the noiseless case.

Fig. 4.4 also reveals that the performance of the SURE loss with $\mathbf{P} = \mathbf{H}^{\dagger}$ is better than the SURE loss with $\mathbf{P} = \mathbf{H}^{T}$ (the average PSNR is about 2.0 dB higher) under the low noise scenario, while they give roughly the same highest PSNR in the high noise case. The reason might be that the BP is degraded when the noise level increases causing $\mathbf{H}$ to be more ill-conditioned [140], [161].

4.3.3 Performance evaluation of SURE for MS-HS fusion

Table 4.1. MS-HS image fusion results for the PU dataset in terms of PSNR (dB), ERGAS, and SAM (°). Best results are in bold.

Low noise: SNR (MSI) $= 40$ dB, SNR (HSI) $= 30$ dB					
	CNMF	HySure	SupRes PALM	GDD	SURE
PSNR (dB)↑	36.32	41.43	34.24	39.83	**42.11**
ERGAS↓	2.456	1.309	3.085	1.563	**1.23**
SAM (°)↓	2.836	2.14	3.241	2.578	**1.927**
High noise: SNR (MSI) $= 40$ dB, SNR (HSI) $= 11.56$ dB					
PSNR (dB)↑	32.39	34.81	32.33	36.31	**39.3**
ERGAS↓	3.853	3.209	3.915	2.37	**1.725**
SAM (°)↓	5.368	4.049	4.773	4.436	**2.879**

The proposed SURE method is compared to both the model-based and unsupervised DL-based methods, which are described as follows. The first two model-based methods are CNMF[1] [92] and SupResPALM[2] [94] which rely on a matrix factorization and

[1] http://naotoyokoya.com/Download.html
[2] https://github.com/lanha/SupResPALM

Table 4.2. MS-HS image fusion results for the DC dataset in terms of PSNR (dB), ERGAS, and SAM (°). Best results are in bold.

Low noise: SNR (MSI) $= 40$ dB, SNR (HSI) $= 30$ dB					
	CNMF	HySure	SupRes PALM	GDD	SURE
PSNR (dB)↑	37.37	35.76	34.51	38.062	**39.71**
ERGAS↓	1.682	1.734	2.842	1.56	**1.325**
SAM (°)↓	3.677	4.077	4.263	3.396	**2.979**
High noise: SNR (MSI) $= 40$ dB, SNR (HSI) $= 13.75$ dB					
PSNR (dB)↑	34.62	33.96	33.19	35.67	**37.10**
ERGAS↓	2.312	2.577	3.157	2.199	**1.749**
SAM (°)↓	4.974	5.767	5.791	4.720	**4.113**

unmixing constraints, the third model-based method, called HySure[3] [93], uses a subspace decomposition and TV regularization. The fourth method is GDD[4] [128] which is an unsupervised CNN using the DIP loss. The performance of HySure and SupRePALM strongly depends on the subspace dimension (k) and the number of endmembers (p). The optimal parameters for those methods are found by using a grid-search based on PSNR in the searching spaces of $k \in \{3, 5, 7, 10, 15\}$ and $p \in \{3, 4, 5, 6, 7, 10, 15\}$. For low noise, the best parameters of HySure are $k = 10$ and $k = 15$ and the best parameters for SupResPALM are $p = 15$ and $p = 15$ for the PU and DC datasets, respectively. For high noise, the best parameters of HySure are $k = 3$ and $k = 7$ and the best parameters for SupResPALM are $p = 5$ and $p = 10$ for the PU and DC datasets, respectively. On the other hand, CNMF is a parameter-free method since it utilizes an automatic algorithm to select tuning parameters. The balance parameters in GDD and SURE are manually set to $\lambda = 0.1$ for all datasets and noise levels. Since there is no criterion to stop training GDD, the stopping point is chosen as 3000 iterations for both datasets, and the fused image is the exponential average of the CNN output with a weight of 0.99 [63]. The same strategy used to obtain the fused image in GDD is also applied to SURE for a fair comparison. One important point is that the competitive model-based methods are allowed to use the ground truth to find the best parameters, while the SURE method does not need the ground truth. Thus, the SURE method is more practical in real life applications than the competitive model-based methods.

Tables 4.1 and 4.2 show the results of all methods for the PU and DC datasets, respectively. the proposed SURE-based method gives better results than the competitive methods in all metrics for both datasets. In general, the unsupervised DL-based methods perform better than the model-based methods. However, it is noticeable that HySure works comparably as well as SURE and better than the GDD method in terms of PSNR

[3] `https://github.com/alfaiate/HySure`
[4] `https://github.com/tuezato/guided-deep-decoder`

for the PU dataset under the low noise scenario. Figs. 4.5-4.6 show the fused images obtained by all methods in the high noise case for the PU and DC datasets, respectively. For the PU datasets, although the numerical result of SURE is better than GDD, the fused images yielded by GDD and SURE are not noticeably different and closer to the reference than the other methods. HySure yields a blurry image for the DC dataset, while the SupResPALM image seems to be spectrally distorted. The difference in the results between CNMF, GDD, and the proposed SURE-based method is not visually pronounced and looks closer to the reference than the other methods. Further inspection of the root mean-square-error (RMSE)-based residual images (i.e., the RMSE computed along the third dimension of an HSI) of the estimated HR HSIs with respect to the original HR HSIs tells us that the proposed method yields less residual structure than the competitive methods. In contrast, SupResPALM yields the highest residual structure among all methods.

4.4 MS-MS fusion: Experimental results

Here, the proposed SURE-based method is applied for the MS-MS fusion, where the representative examples are S2 sharpening and pansharpening.

The S2 satellite imaging system provides 13 multispectral multiresolution bands. A S2 MSI comprises of four bands at 10 m (spatial) resolution, six at 20 m resolution, and three at 60 m resolution. In this example, one 60 m band (band 10) is ignored since it does not contain the Earth's surface information. The S2 sharpening problem is to estimate the 20 m and 60 m bands at 10 m resolution by fusing them with the 10 m bands. In S2 sharpening, there is no explicit relationship between the guided image (the 10 m bands) and the estimated LR image (the 20 m and 60 m bands). The guided image is assumed as a part of the fused image, so that the linear mapping operator $\mathbf{P}$ applied to the guided image part is identity. Although we are interested in estimating the LR images at their maximum resolution, adding the guided image to the fused image improves the fusing results since it exploits the correlation between all the S2 bands [130]. To apply the proposed SURE method for the S2 sharpening, the CNN in Fig. 4.2 is used and is optimized with the SURE loss function (4.8).

4.4.1 Datasets and evaluation metrics

The experiments are performed using both simulated and real S2 datasets. The simulated dataset is based on the Airborne Prism Experiment (APEX) HSI dataset, which is described in the Appendix. The APEX MSI is obtained by applying the S2 spectral response to the APEX HSI resulting in an MSI having 12 bands of 198×198 pixels per band and a resolution of 2 m. The MSI is then used as the reference. The 20 m and 60 m bands are simulated by filtering the reference with the PSFs taken from the S2 sensors' MTFs [151] followed by a downsampling operator with a factor of 2 and 6, respectively. Isotropic Gaussian noise is added to the 20 m and 60 m bands, such that the images have SNR $= 30$ dB and SNR $= 40$ dB for all bands.

The real S2 dataset is a level 1C product and can be publicly downloaded from

the ESA/Copernicus portal. This real S2 dataset (called Vietnam) shows a part of a sub-urban area in Khanh Hoa province, Vietnam, and was acquired in April 2019. The spatial size of the 10 m bands of the Vietnam dataset is 420×420 pixels.

For performance evaluation using the APEX dataset where the reference is available, the following evaluation metrics are used, the MSRE in decibels, the SAM in degrees, and the MSSIM. All the metrics are computed with respect to the original HR bands without noise added. Those metrics are given in detail in Appendix. For the Vietnam dataset, due to the lack of reference images the results are assessed visually.

4.4.2 Validation of SURE for S2 Sharpening

As was done with MS-HS fusion, this section verifies that using SURE loss ovoids overfitting. Also the SURE loss performance with different linear operators $\mathbf{P}$ is analysed. For the experiments presented here, the same CNN as in Fig. 4.2 is optimized with the BP-DIP loss function (4.4), the SURE loss functions (4.8) with $\mathbf{P} = \mathbf{H}^{\dagger}$ and with $\mathbf{P} = \mathbf{H}^{T}$, respectively.

Fig. 4.7 shows the MSRE curves in training for three loss functions mentioned above using the APEX dataset with SNR $= 40$ dB and SNR $= 30$ dB. The same trend as in the MS-HS image fusion is observed. There, training a CNN with the BP-DIP loss function suffers an overfitting problem, specifically in the higher noise case (Fig. 4.7b). In contrast, training a CNN with the SURE loss functions ignore overfitting. The performance of the SURE loss with $\mathbf{P} = \mathbf{H}^{\dagger}$ is significantly better than one with $\mathbf{P} = \mathbf{H}^{T}$ when the noise level decreases.

4.4.3 Performance evaluation of SURE for S2 Sharpening

Table 4.3. S2 sharpening results (60 m and 20 m bands) for the APEX dataset in terms of MSRE (dB), SAM (°) and MSSIM. Best results are in bold.

	SNR=40 dB				
	SupReME	S2Sharp	SSSS	BP-DIP	SURE
MSRE (dB)↑	22.86	26.89	25.43	**27.58**	27.53
SAM (°)↓	3.855	4.250	4.914	**2.421**	2.533
MSSIM↑	0.921	0.937	0.911	**0.955**	**0.955**
Time (s)↓	**3.17**	11.60	182.84	277.01	392.54
	SNR=30 dB				
MSRE (dB)↑	22.34	23.13	20.6	23.72	**25.85**
SAM (°)↓	4.499	5.637	9.033	3.908	**2.893**
MSSIM↑	0.905	0.896	0.836	0.906	**0.940**
Time (s)↓	**3.28**	10.01	180.15	273.06	390.16

Experimental results of SURE for S2 sharpening using the simulated APEX dataset and the real Vietnam dataset are presented here. Moreover, the proposed SURE-

based method is compared against the recently published S2 sharpening methods. The competitive methods are three model-based methods (SupReME[5] [97], S2Sharp[6] [99], and SSSS[7] [100]) and one unsupervised DL-based method that is our proposed method trained with the BP-DIP loss function (4.4). SupReME and S2Sharp exploit the low-rank property of the S2 data, where the fused images are the solutions to the penalized least square problems in a reduced-rank domain with a roughness regularizer. The difference between SupReME and S2Sharp is that while SupReME assumes the parameters (e.g., the dimension of subspace and the regularization weights) are fixed, S2Sharp automatically estimates those parameters using a Bayesian optimization method. SSSS also relies on the low-rank property of the S2 data, but it solves the reduced-rank least square problem with the self-similarity regularizer. It is experimentally observed that the main parameters affecting the results of SupReME and SSSS are the dimension of subspace k and the regularization weight λ. Using a grid-search for the APEX dataset, the best parameters (based on highest MSRE) of SupReME and SSSS for both SNR $= 30$ dB and SNR $= 40$ dB was found to be $k = 8$, $\lambda = 0.01032$, and $k = 8$, $\lambda = 0.01$, respectively. Similar to the MS-HS fusion, there is no criterion stop optimizing the BP-DIP method, the training iteration is chosen manually at 3000 iterations, and the result is the running average of the CNN output with an exponential weight is 0.99 [63]. For a fair comparison, the same schedule to obtain a result in BP-DIP is also applied in the SURE method.

The quantitative results are shown in Table 4.3 for the APEX dataset. In the low noise scenario of SNR $= 40$ dB, BP-DIP and SURE yield similar results as was discussed in Section 4.4.2. However, in a higher noise scenario of SNR $= 30$ dB, SURE is considerably better than BP-DIP since BP-DIP overfits at 3000 iterations. Both SURE and BP-DIP outperform the model-based methods in all metrics and noise cases. For example, SURE give results which are roughly 1.5 and 2.7 dB in terms of MSRE higher than the best model-based method, S2Sharp, for SNR $= 40$ dB and SNR $= 30$ dB, respectively. The images sharpened by BP-DIP and SURE have remarkably less spectral distortion than the model-based methods, with 1.2 to 6.2 degrees lower in terms of SAM for all noise cases. Fig. 4.8 and Fig. 4.9 show the sharpened bands 11 (20 m band) and 9 (60 m band), and their residual images (in logarithm scale) of the APEX dataset obtained by all methods and the reference image. Zooming in the reconstructed images of band 11 reveals that SupReME generates some grid-like artifacts, and the images generated by SSSS and S2Sharp are slightly noisy (see the top left corner). The images of BP-DIP and SURE are similar and very close to the reference image. Band 9 has high contrast, and it is very hard to differentiate between the sharpened images visually. However, the residual images of both bands 11 and 9 indicate that SURE and BP-DIP yield less residual structure than the competitive methods.

Computational complexity of all methods is assessed in term of running time in seconds, and is reported in Table 4.3. The proposed SURE-based method was implemented in Pytorch 1.8 GPU, and was run on a Linux computer with a Linux computer with eight cores Intel CPU 3.2 GHz, 64 GB of RAM, and 12 GB memory of GPU (Nvidia Titan X). The competitive methods were implemented using Matlab

[5] https://github.com/lanha/SupReME
[6] https://github.com/mou12/S2Sharp
[7] https://sites.google.com/view/chiahsianglin/software

R2019b, and were run on the same computer. The fastest to slowest methods are ranked as SupReME, S2Sharp, SSSS, BP-DIP, and SURE.

The experimental results for the real S2 dataset (Vietnam) are shown in Fig. 4.10 and Fig. 4.11. In this experiment, the parameters for all methods are set as the same as in the simulated case. Noise is estimated using the median absolute deviation of the details sub-band (HH) using 2-D wavelet decomposition [163]. Sharpened images of the 20 m bands are shown in Fig. 4.10. SupReME produces some grid-like artifacts, the remaining methods yield well-looking images, and the difference between those images is not visually pronounced. For the 60 m bands shown in Fig. 4.11, the SupReME image has some artifacts, and the SSSS image is noisy and blurry. S2Sharp, BP-DIP, and the proposed methods give good sharpened images, which are sharper than SupReME and SSSS images. The results of BP-DIP and SURE are very similar, that is because the noise might be small for this dataset. Noting that BP-DIP and SURE seem to generate spectral distortion in the 60 m bands sharpened images since the color of those images deviates from the observed bands.

4.4.4 Pansharpening

In pansharpening, the guided image is a PAN, and the LR image is an MSI. Since the PAN is a linear combination of the bands of an HR MSI, pansharpening can be considered a special case of the MS-HS image fusion where the PAN and LR MSI play the roles of the MSI and LR HSI, respectively. The proposed SURE method is applied to pansharpening by optimizing a CNN with the SURE loss function (4.11), where the LR HSI and MSI are replaced by the LR MSI and the PAN, respectively.

The pansharpening algorithms are evaluated using simulated Pleiades dataset [71]. The Pleiades dataset was collected by an aerial platform, showing a scene of Toulouse (France) in 2006. It has four MS bands (HR MSI) with 60 cm resolution and lacks a PAN. The LR MSI is simulated by band-wise filtering the HR MSI using a Gaussian low-pass filter with a standard deviation of 2.0 and kernel support of 15 pixels and is downsampled with a factor of 4 [71]. The simulated PAN is the average of all bands of the HR MSI. The proposed method is compared against the representative CS, MRA, and supervised DL-based methods, i.e., BDSD [82], MTF-GLP [87][8], and PNN[9] [109]. The BDSD and MTF-GLP methods are fine tuned by modifying the MTF-based filter with the Gaussian filter described above (i.e., a standard deviation of 2 and kernel support of 15 pixels). For the PNN method, the pre-trained network provided by the authors is used.

Experimental results are shown in Fig. 4.12. The SURE result is significantly better than the competitive methods in quantitative metrics and the pansharpened images' quality. SURE yields 2.76° and 1.74 in terms of SAM and ERGAS which are roughly 2.3° and 1.2 lower than the second best method, BDSD. The competitive methods suffer some degrees of spectral and spatial distortion where the resulted images look blurry, and the color deviates far from the reference image. It is not surprising that PNN performs poorly, although it is a supervised DL-based method, since the pre-trained

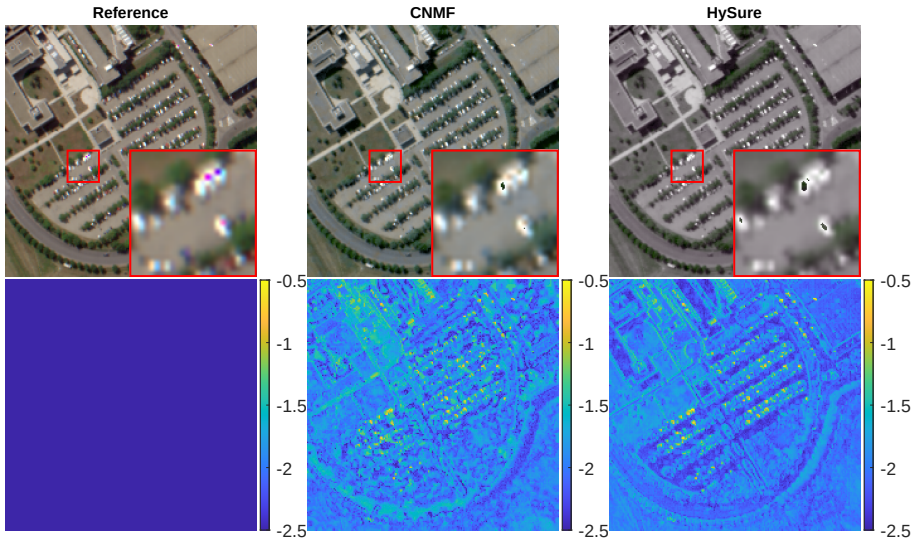
[8] Codes for BDSD and MTF-GLP are available at https://openremotesensing.net/knowledgebase/a-critical-comparison-among-pansharpening-algorithms/

[9] https://github.com/sergiovitale/pansharpening-cnn-matlab-version

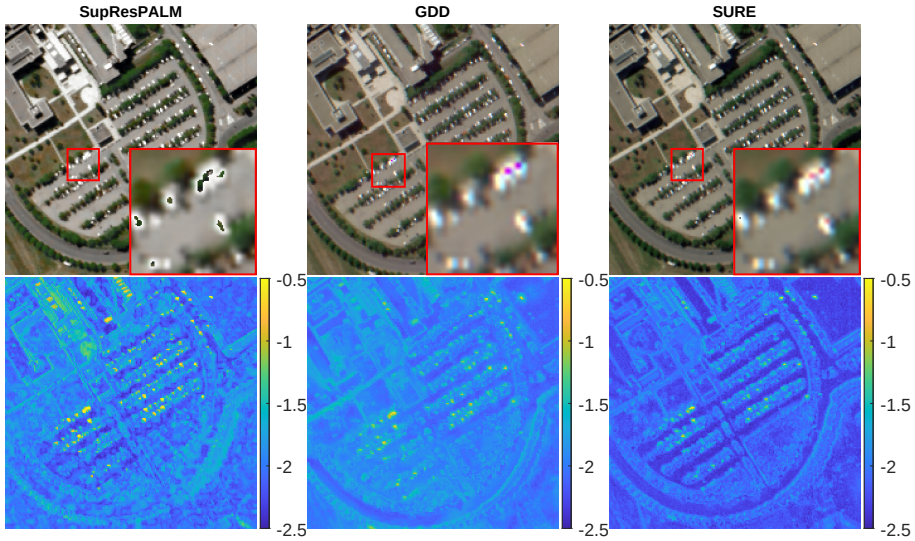
CNN of PNN was trained using the bicubic degradation model that does not match the degradation model assumed in this section.

4.5 Conclusions

A general RS image fusion problem has been addressed by using a novel method based on SURE. The main innovation of the proposed method is that it incorporates a linear mapping operator and SURE to prevent overfitting and improve the results. Also, the proposed method is a hybrid model and DL-based method, which overcomes the difficulty in the hand-crafted designed image prior for the model-based methods and the dependency of the ground truth for training in supervised DL-based method. Experimental results for three representative RS image fusion problem using SURE and BP verify that the proposed method yields significant improvement results over competitive methods. However, there are still open questions that need to be addressed in the future works, for example, different kinds of the pre-processing operators $\mathbf{P}$ (e.g., a non-linear operator such as a neural network mapping function), the more efficient way to compute the trace of the derivative of the network with respect to its input, and applying the SURE for the other MSE-related loss function (e.g., ERGAS) is an interesting idea.

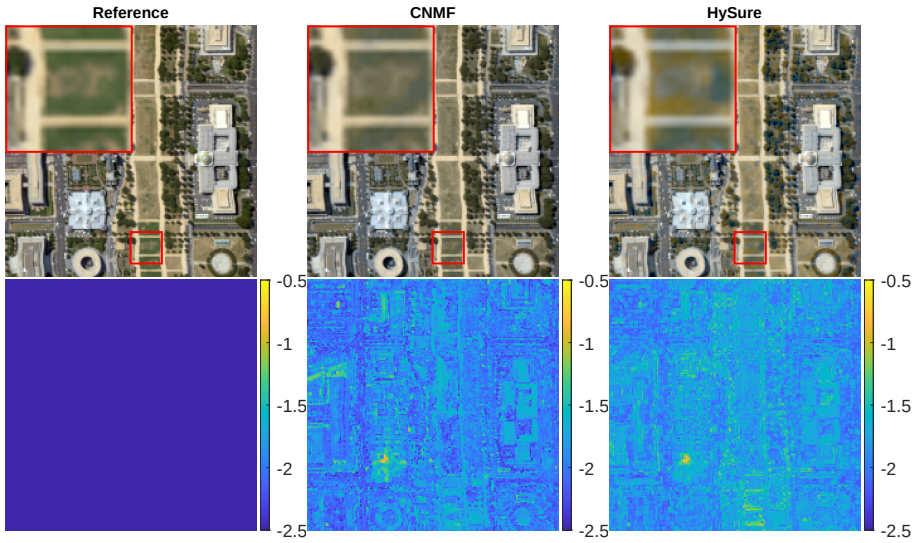


(a) Reference *(b)* CNFM *(c)* HySure

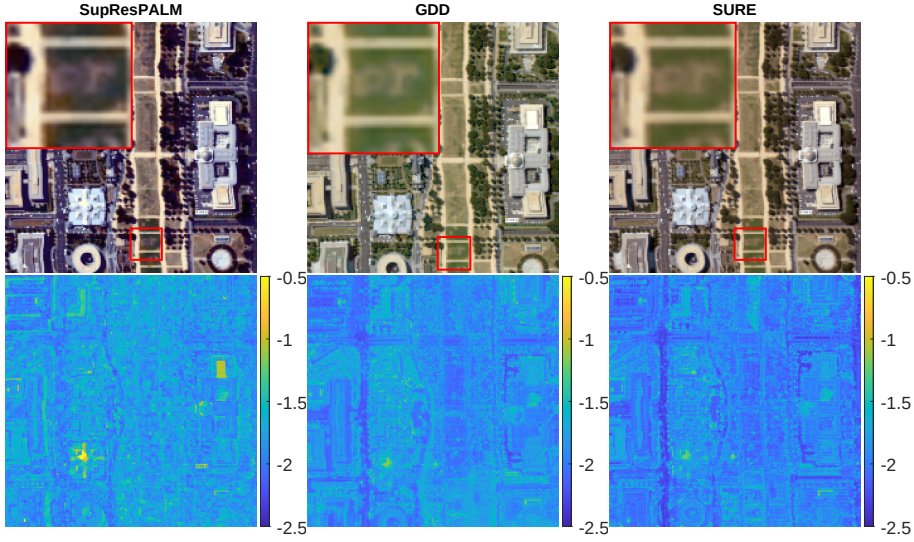


(d) SupResPALM *(e)* GDD *(f)* SURE (Ours)

Figure 4.5. Fusion results for the PU dataset (high noise). The first row are the images shown in natural color using an HSI to RGB color rendering method [165]. The images shown in big red rectangles are the 4 times zooming in of the images shown in small red rectangles. The second row are the RMSE-based residual images (shown in logarithm scale) with respect to the reference.

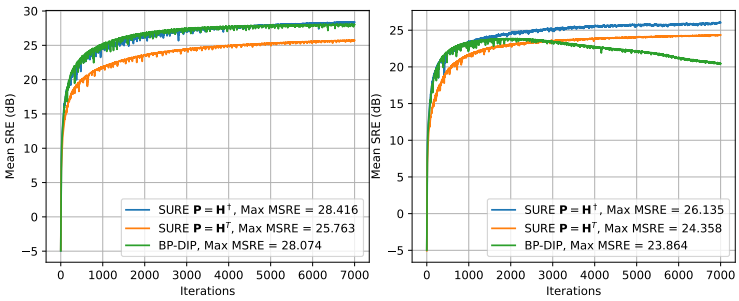


(a) Reference *(b)* CNFM *(c)* HySure



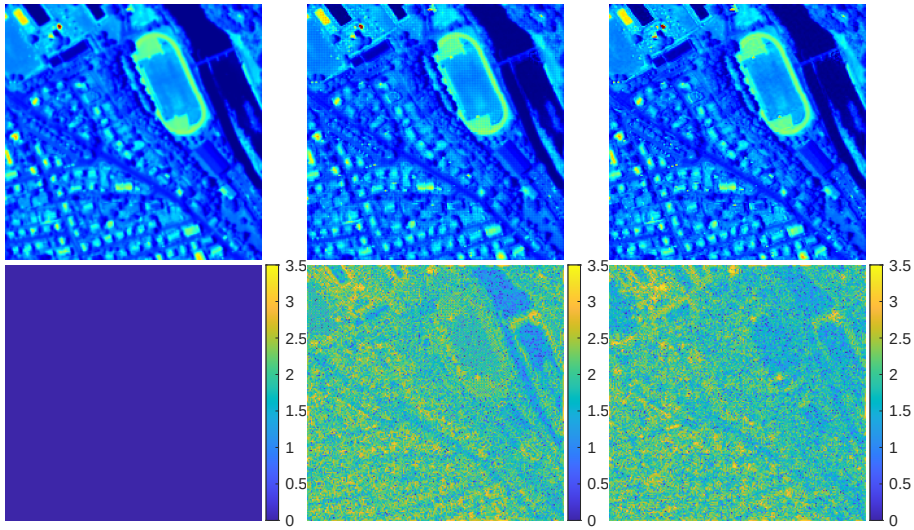
(d) SupResPALM *(e)* GDD *(f)* SURE (Ours)

Figure 4.6. Fusion results for the DC dataset (high noise). The first row are the images shown in natural color using an HSI to RGB color rendering method [165]. The images shown in big red rectangles are the 4 times zooming in of the images shown in small red rectangles. The second row are the RMSE-based residual images (shown in logarithm scale) with respect to the reference.

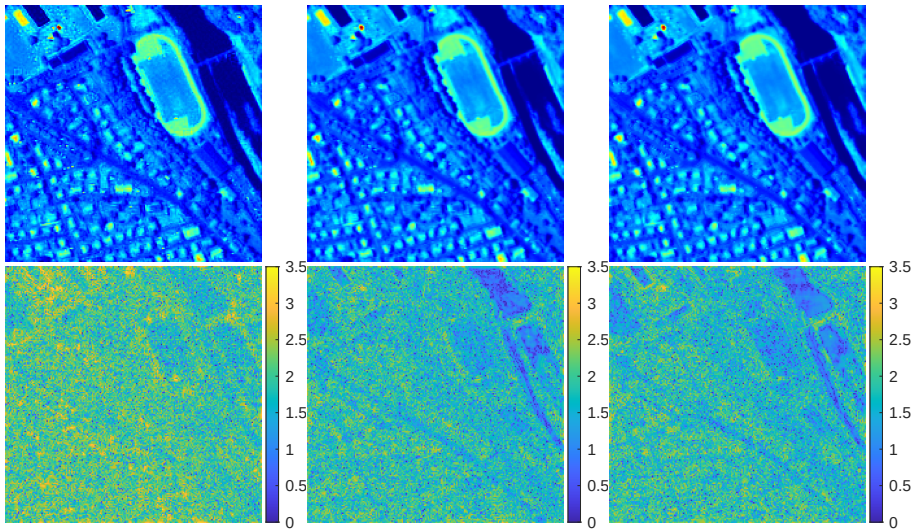


(a) SNR $= 40$ dB *(b)* SNR $= 30$ dB

Figure 4.7. MSRE curves in training for the APEX dataset using different loss functions.

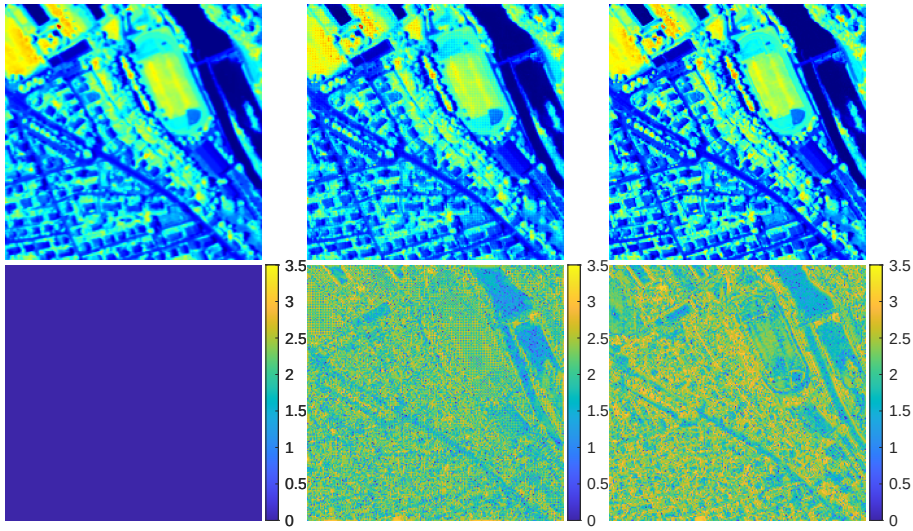


(a) Reference *(b)* SupReME *(c)* S2Sharp

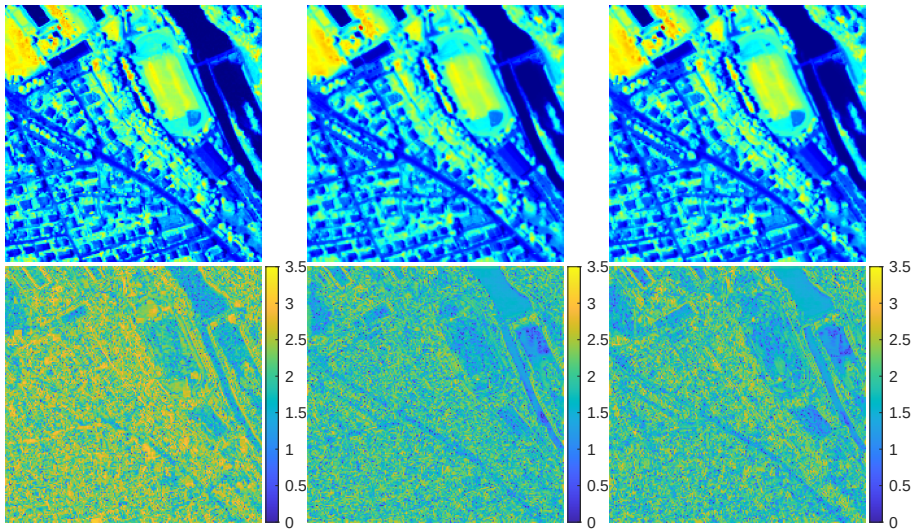


(d) SSSS *(e)* BP-DIP *(f)* SURE (Ours)

Figure 4.8. Image fusion results for the APEX dataset (SNR $= 40$ dB). Top row are the images for band 11 (20 m) and bottom row are the respective residual images shown in logarithm scale.

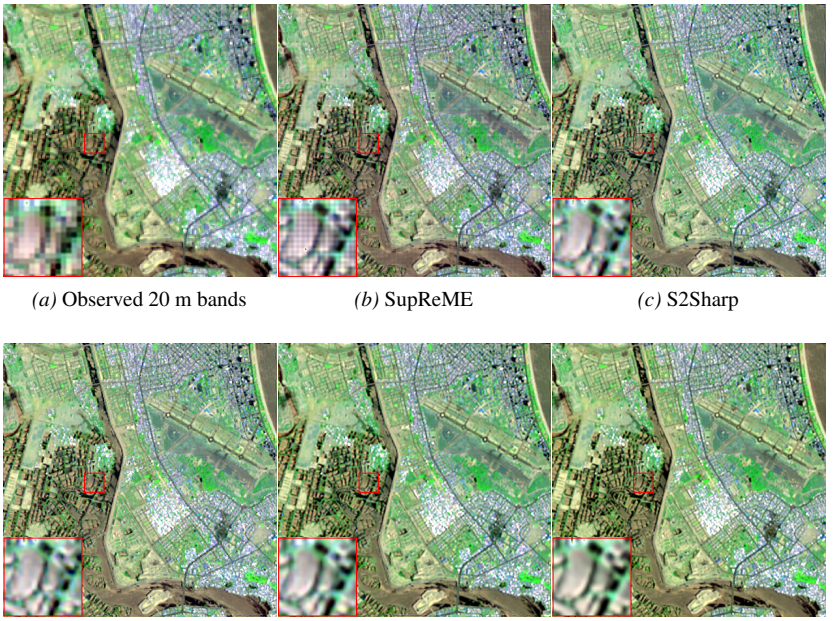


(a) Reference *(b)* SupReME *(c)* S2Sharp



(d) SSSS *(e)* BP-DIP *(f)* SURE (Ours)

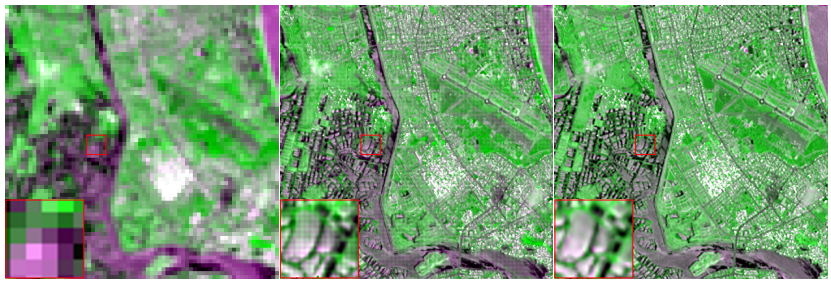
Figure 4.9. Image fusion results for the APEX dataset (SNR $= 40$ dB). Top row are the images for band 9 (60 m) and bottom row are the respective residual images shown in logarithm scale.



(a) Observed 20 m bands *(b)* SupReME *(c)* S2Sharp

(d) SSSS *(e)* BP-DIP *(f)* SURE

Figure 4.10. Image fusion results of the 20 m bands for the Vietnam dataset. The images are shown in false color images using bands 12, 8a, and 5 as the R, G, and B channels. The images shown in big red rectangles are the 4 times zooming in of the images shown in small red rectangles.



(a) Observed 60 m bands *(b)* SupReME *(c)* S2Sharp

(d) SSSS *(e)* BP-DIP *(f)* SURE

Figure 4.11. Image fusion results of the 60 m bands for the Vietnam dataset. The images are shown in false color images using bands 1, 9, and 1 as the R, G, and B channels. The images shown in big red rectangles are the 4 times zooming in of the images shown in small red rectangles.

(a) Reference *(b)* LR MSI *(c)* PAN *(d)* BDSD $(5.08°, 2.93)$

(e) MTF-GLP $(5.45°, 4.58)$ *(f)* PNN $(5.59°, 4.99)$ *(g)* SURE (**2.76°**, **1.74**)

Figure 4.12. Pansharpening results for the Pleiades dataset. The reference, LR-MSI, and the pansharpened images of 256×256 *pixels are shown in natural color using bands 1, 2, and 3 as the R, G, B channel. The PAN is shown in gray scale. The number presented in brackets are the SAM in degrees and ERGAS where the best results are in bold.*

CHAPTER 5

CONCLUSIONS

This chapter gives the conclusions of the thesis and future directions.

The main topics of this thesis are HSI denoising and RS image fusion which are the special cases of RS image restoration. Several HSI denoising and RS image fusion methods have been proposed. All the proposed methods are unsupervised DL-based. The core idea is based on the DIP theory stating that the structure of a CNN provides a good image prior to reconstructing an image. This allows one to train a CNN for RS image restoration similarly to the model-based methods without any ground truth. In this thesis, several techniques are added to the DIP in the proposed methods to improve the results of HSI denoising and RS image fusion. The main remarks and conclusions are listed below

- RS images have high correlation in both spectral and spatial domains. This property allows us to represent an RS image in a low-rank subspace and very few sparse coefficients using a transform. The DIP-SLR HSI denoising method proposed in this thesis exploited the sparse and low-rank properties using the SVD and 2-D wavelet transform. Then a CNN was trained in the transform domain to obtain better performance in denoising results and computational time. Additionally, it was also shown that the high correlation of RS image benefited the results of the SURE-based HSI denoising method and the S2 sharpening with the MTF-based degradation model method proposed in this thesis.

- SURE was used to derive the loss functions for unsupervised training CNNs in both HSI denoising and RS image fusion. In the SURE-based HSI denoising method, SURE was applied to estimate the unbiased value of the MSE between the denoised and reference images. In the SURE-based RS image fusion method, SURE, accompanied by a linear mapping operator, was applied to estimate the unbiased value of the linear transform of the MSE between the fused and reference images. In both cases, the unbiased estimate MSEs were calculated without using any information about the reference images. By training CNNs with the SURE loss functions, overfitting was avoided, and the methods were unsupervised.

- The SURE-based methods could work with non-Gaussian noise, such as Poissonian noise, although the theoretical assumption behind SURE is Gaussian noise. To deal with the non-Gaussian, the data were first approximated to Gaussian via a transform (e.g., the Anscombe transform) and then applied the SURE methods. Moreover, the SURE-based HSI denoising method could work with subspace

data. In some cases, using the SURE-based method in subspace improved the results and significantly reduced the running time.

- SURE has been used for parameter selection in many model-based methods where only a few parameters need to be selected. Applying SURE in the DL context, the proposed HSI denoising and RS image fusion methods showed that SURE was successfully used to select the optimal values for the network weights and bias of DL models.
- In the proposed SURE-based RS image fusion method, an unsupervised CNN assisted by a pre-processing operator, such as a linear mapping operator that maps an LR image to its HR one, improved the results. As an example where the degradation model, i.e., the sensors' PSFs and downsampling factor, was known, a BP operator was constructed and was used as the linear operator. Experiments showed that the BP gave a significant improvement in fusion results for MS-HS fusion, S2 sharpening, and pansharpening.
- The sensors' MTFs were an important factor in modeling the degradation in S2 sharpening. It was shown that the correct implementation of the MTF-based degradation model that matched the observation model yielded better S2 fusion results in both reduced and full resolution.
- For a multi-factor super-resolution problem, such as S2 sharpening, the study in this thesis showed that using a single CNN for the super-resolution of all bands was better than using separate CNNs for the super-resolution of each band.

However, the RS image restoration problem addressed in this thesis can be extended in many directions. Some ideas and recommendations that will be addressed in future works are listed as follows

- The network divergence in the SURE loss functions was computed using the MC-SURE in this thesis. Another approach to computing the network divergence is available by using the automatic differentiation packages of the DL framework. However, those approaches are limited in accuracy and computational time. Therefore, a more efficient technique to compute the network divergence will be developed in future work.
- The BP operator was chosen as the linear operator in the SURE-based RS image fusion method. However, the other linear mapping operator will be analysed carefully to find the optimal operator for this task.
- Applying SURE to MSE-related metrics (e.g., ERGAS) to construct the loss function for training a CNN is another exciting idea that will be done in future work.
- The HSI denoising method in this thesis focused on only Gaussian noise. Several kinds of noises, such as impulse noise, dead pixel noise, and mixed noise, will be addressed in a future study.
- Another direction is to use advanced DL techniques (e.g., generative adversarial network (GAN)) and architectures (e.g., transformer) for RS imagery.

APPENDIX

This appendix describes the datasets, and defines the evaluation metrics used in the thesis.

A.1 Datasets

A.1.1 Hyperspectral image datasets

THE INDIAN PINES DATASET

The Indian Pines[10] dataset is an HSI that was gathered by the Airborne Visible InfraRed Imaging Spectrometer (AVIRIS) sensor over the Indian Pines test site in North-western Indiana. It consists of 145×145 pixels and 224 spectral reflectance bands in the wavelength range $400 - 2500$ nm. The Indian Pines dataset contains mostly Gaussian noise and is used as a real dataset to evaluate the denoising algorithms in this thesis. Fig. A.1 shows an RGB image of the Indian Pines dataset where the RGB image is created using the method [165].

Figure A.1. The Indian Pines dataset shown as an RGB image using the HS to RGB image rendering method [165].

[10] http://lesun.weebly.com/hyperspectral-data-set.html

THE URBAN DATASET

Urban[11] is an HSI collected by Hyperspectral Digital Image Collection Experiment (HYDICE). The image has 307×307 pixels and 210 spectral bands ranging from 400 nm to 2500 nm. The spatial and spectral resolutions are 2 m, and 10 nm, respectively. Gaussian, death pixels, and stripped noise are dominant in the Urband dataset, and this dataset is used as a real dataset to evaluate the denoising algorithms in this thesis. Fig. A.2 shows an RGB image of the Urban dataset where the RGB image is created using the method [165].

Figure A.2. The Urban dataset shown as an RGB image using the HS to RGB image rendering method [165].

[11] http://lesun.weebly.com/hyperspectral-data-set.html

WASHINGTON DC MALL DATASET

The Washington DC Mall HSI dataset[12] was collected by the Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor. The image has 191 bands of 1208×307 pixels per band. The spatial and spectral resolution is 2 m and 10 nm, respectively. This dataset is relatively clean, and it is used as ground truth to simulate the noisy and low resolution data in this thesis. Fig. A.3 shows an RGB image of the Washington DC Mall dataset where the RGB image is created using the method [165].

Figure A.3. The Washington DC Mall dataset shown as an RGB image using the HS to RGB image rendering method [165].

[12] https://engineering.purdue.edu/ biehl/MultiSpec/hyperspectral.html

THE PAVIA UNIVERSITY DATASET

The Pavia University dataset [13] is hyperspectral dataset acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor showing a scene of Pavia, northern of Italy. The wavelength covered by the sensor is in the range of $430 - 860$ nm. The spatial and spectral dimensions for the Pavia University are 610×610 and 103, respectively. In this thesis, the first few noisy bands of the Pavia University dataset are removed and the rest is used to simulate the noisy and low resolution data. Fig. A.4 shows an RGB image of the Pavia University dataset where the RGB image is created using the method [165].

Figure A.4. The Pavia University dataset shown as an RGB image using the HS to RGB image rendering method [165].

A.1.2 Multispectral image datasets

APEX DATASET

APEX[14] is an open science dataset provided by the Airborne Prism Experiment

[13] https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes
[14] https://apex-esa.org/en/data/free-data-cubes

(APEX). APEX is a hyperspectral image acquired by the APEX imaging spectrometer, which covers the wavelength from 413 nm to 2412 nm. The APEX hyperspectral image was obtained by the sensors mounted on a flight at 4600 m altittude. The image has a spatial resolution of 1.8 m and shows a scene in the vicinity of Baden, Switzerland, in 2011. In this thesis, the APEX hyperspectral image is used to simulate the S2 image. Fig. A.5 shows the APEX dataset as an RGB image.

Figure A.5. The APEX dataset is shown as an RGB image.

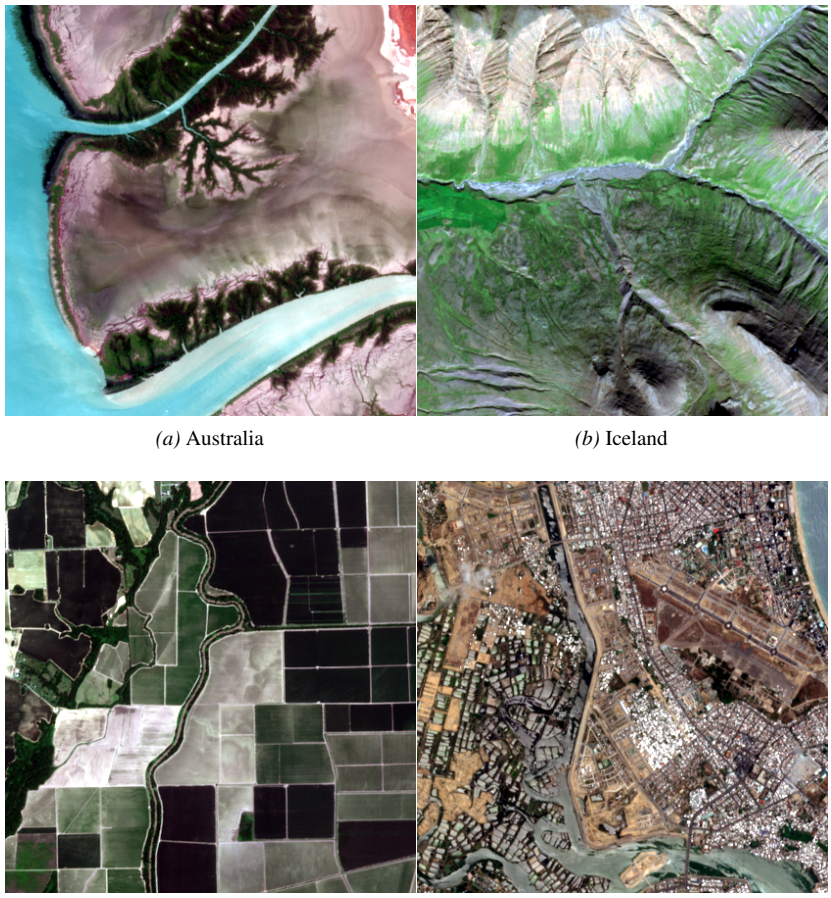
THE REAL SENTINEL 2 DATASETS

The real S2 data which are publicly available on the ESA/Copernicus portal[15]. The data are Level 1C product obtained at top of the atmosphere (TOA) by the Sentinel-2A constellation. The S2 constellations provide data in terms of tiles, where each tile covers an area of approximately 110×110 km^2. It means that the 10 m bands having a spatial resolution of approximately $11,000 \times 11,000$ pixels. In this thesis, each dataset is a part of an S2 tile. For 20 m bands sharpening evaluation, four datasets called Australia, Iceland, USA, and Vietnam are used. The datasets have the spatial resolution of 410×410 pixels (4.1×4.1 km^2) for the 10 m bands. They are described as follows:

- Australia dataset covers a coastal area in the Keep River National Park, northern Australia. It is acquired on September 2020.
- Iceland dataset covers a mountain area in Varmahlid, northern Iceland. It is acquired on August 2020.
- Vietnam dataset covers a sub-urban area in Khanh Hoa province, central coast of Vietnam. It is acquired on April 2019.
- USA dataset cover an agriculture area of Mississippi River Delta, in Mississippi state of the USA. It is acquired on June 2020.

The Australia, Iceland, USA, and Vietnam datasets are shown in RGB images for the 10 m bands as in Fig. A.6. Another real S2 dataset used for 60 m sharpening evaluation is called USA-60 which has the spatial resolution of $1,800 \times 1,800$ pixels (18×18 km^2) for the 10 m bands. This dataset is obtained at the same tile of the USA dataset mentioned above, and is shown as an RGB image in Fig. A.7.

[15] https://scihub.copernicus.eu/dhus/#/home



(a) Australia *(b)* Iceland

(c) USA *(d)* Vietnam

Figure A.6. The real S2 datasets used for 20 m bands sharpening in this thesis.

A.2 Evaluation metrics

PSNR

The peak signal-to-noise-ratio (PSNR) in decibels is the highest signal-to-noise ratio of a reconstructed image with respect to the reference. The higher value of PSNR implies that the better image is reconstructed.

$$\text{PSNR (dB)} = 10\log_{10}\left(\frac{\max^2(\mathbf{x})}{\text{MSE}(\mathbf{x},\widehat{\mathbf{x}})}\right),$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the reference and fused image, respectively. The maximum value of the reference is represented by $\max(\mathbf{x})$ and $\text{MSE}(\mathbf{x},\widehat{\mathbf{x}})$ is the mean square error

Figure A.7. The USA-2 dataset shown as an RGB image.

between the reconstructed image and the reference.

SRE

The signal-to-reconstructed error (SRE) in decibels for the ith band of an image is computed as

$$\text{SRE (dB)} = 10\log_{10}\left(\frac{\|\mathbf{x}_i\|_2^2}{\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2}\right),$$

where $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ are the reference and reconstructed image at band ith, respectively. The mean SRE (MSRE) in decibels is the average value of SRE calculate over all bands of an image. The higher value of SRE and MSRE imply that the better image is reconstructed

ERGAS

Erreur Relative Globale Adimensionnelle de Synthèse (ERGAS) [166] computes the amount of spectral/spatial distortion of the reconstructed image with respect to the reference. Smaller value of ERGAS is desired for a good reconstructed image.

$$\text{ERGAS} = \frac{100}{r}\sqrt{\frac{1}{d}\sum_{i=1}^{d}\frac{\text{MSE}(\mathbf{x}_i, \hat{\mathbf{x}}_i)}{\mu_{\mathbf{x}_i}^2}},$$

where d is the number of spectral bands, r is the resolution ratio between the HR and LR images, and $\mu^2_{\mathbf{x}_i}$ is the mean value of the ith band of the reference image.

SAM

The spectral angle mapper (SAM) [167] in degrees indicates the spectral similarity of two vector as an angle. The value of SAM for an entire image is given by the average of SAM for each pixel, and is computed as below

$$\text{SAM}(^\circ) = \frac{1}{N}\sum_{p=1}^{N} \arccos\left(\frac{\mathbf{x}^T_{(p)}\hat{\mathbf{x}}_{(p)}}{\|\mathbf{x}_{(p)}\|\|\hat{\mathbf{x}}_{(p)}\|}\right)\frac{180}{\pi},$$

where $\mathbf{x}(p)$ and$\hat{\mathbf{x}_{(p)}}$ are two vectors containing data of reference and reconstructed image at pixel p, respectively. The smaller values of SAM indicates better reconstructed image. The optimal value for SAM is 0 meaning that there is no spectral distortion.

SSIM

The mean structural similarity index (MSSIM) is the band-wise mean of SSIM. The SSIM of ith band is calculated as [168]

$$\text{SSIM}_{i,j} = \frac{(2\mu_{x_{i,j}}\mu_{\widehat{x}_{i,j}} + c_1)(2\sigma_{x_{i,j}\widehat{x}_{i,j}} + c_2)}{(\mu^2_{x_{i,j}} + \mu^2_{\widehat{x}_{i,j}} + c_1)(\sigma^2_{x_{i,j}} + \sigma^2_{\widehat{x}_{i,j}} + c_2)},$$

where $\mu_{x_{i,j}}, \mu_{\widehat{x}_{i,j}}, \sigma_{x_{i,j}}$, and $\sigma_{\widehat{x}_{i,j}}$ denotes mean and standard deviation for the reference and reconstructed image in a local window centered at pixel j, respectively. $\sigma_{x_{i,j}\widehat{x}_{i,j}}$ is the cross-covariance between two images. $c_1 = (K_1 D)^2$ and $c_2 = (K_2 D)^2$, where $K_1 = 0.01, K_2 = 0.3$, and D is the dynamical range of the image. The optimal value of SSIM is 1, and the closer to 1 of SSIM value indicates the better reconstructed image.

REFERENCES

[1] J. A. Richards and J. Richards, *Remote sensing digital image analysis*. Springer, 1999, vol. 3.

[2] R. O. Green, M. L. Eastwood, C. M. Sarture, *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sensing of Environment*, vol. 65, no. 3, pp. 227–248, 1998.

[3] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 29–43, Jan. 2002.

[4] B. Zhang, D. Wu, L. Zhang, Q. Jiao, and Q. Li, "Application of hyperspectral remote sensing for environment monitoring in mining areas," *Environmental Earth Sciences*, vol. 65, no. 3, pp. 649–658, 2012.

[5] B. Lu, P. D. Dao, J. Liu, Y. He, and J. Shang, "Recent advances of hyperspectral imaging technology and applications in agriculture," *Remote Sensing*, vol. 12, no. 16, pp. 2659–2699, 2020.

[6] B.-C. Gao, C. Davis, and A. Goetz, "A review of atmospheric correction techniques for hyperspectral remote sensing of land surfaces and ocean color," in *2006 IEEE International Symposium on Geoscience and Remote Sensing*, 2006, pp. 1979–1981.

[7] Q. Zhang, Q. Yuan, J. Li, X. Liu, H. Shen, and L. Zhang, "Hybrid noise removal in hyperspectral imagery with a spatial–spectral gradient network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 7317–7329, 2019.

[8] M. González-Audıcana, X. Otazu, O. Fors, and J. Alvarez-Mozos, "A low computational-cost method to fuse IKONOS images using the spectral response function of its sensors," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 6, pp. 1683–1691, 2006.

[9] Y. Xie, Z. Sha, and M. Yu, "Remote sensing imagery in vegetation mapping: A review," *Journal of Plant Ecology*, vol. 1, no. 1, pp. 9–23, 2008.

[10] X. Wang and W. Yang, "Water quality monitoring and evaluation using remote sensing techniques in China: A systematic review," *Ecosystem Health and Sustainability*, vol. 5, no. 1, pp. 47–56, 2019.

[11] A. Bouvet, T. Le Toan, and N. Lam-Dao, "Monitoring of the rice cropping system in the Mekong Delta using ENVISAT/ASAR dual polarization data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 2, pp. 517–526, 2009.

[12] F. Palsson, J. R. Sveinsson, J. A. Benediktsson, and H. Aanaes, "Classification of pansharpened urban satellite images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 1, pp. 281–297, 2012.

[13] C. Padwick, M. Deskevich, F. Pacifici, and S. Smallwood, "WorldView-2 pansharpening," in *Proceedings of the ASPRS 2010 Annual Conference, San Diego, CA, USA*, vol. 2630, 2010, pp. 1–14.

[14] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804–3814, 2008.

[15] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, 2002.

[16] N. M. Nasrabadi, "Hyperspectral target detection: An overview of current and future challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 34–44, 2013.

[17] R. C. Gonzalez, *Digital image processing*. Pearson Education India, 2009.

[18] B. Rasti, Y. Chang, E. Dalsasso, L. Denis, and P. Ghamisi, "Image restoration for remote sensing: Overview and toolbox," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 2, pp. 201–230, 2022.

[19] H. Shen and L. Zhang, "A MAP-based algorithm for destriping and inpainting of remotely sensed images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1492–1502, 2008.

[20] T. Akgun, Y. Altunbasak, and R. M. Mersereau, "Super-resolution reconstruction of hyperspectral images," *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1860–1875, 2005.

[21] H. Ghassemian, "A review of remote sensing image fusion methods," *Information Fusion*, vol. 32, pp. 75–89, 2016.

[22] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[23] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 2862–2869.

[24] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 479–486.

[25] I. Atkinson, F. Kamalabadi, and D. L. Jones, "Wavelet-based hyperspectral image estimation," in *2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IGARSS)*, vol. 2, Jul. 2003, 743–745 vol.2.

[26] B. Rasti, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Hyperspectral image denoising using 3D wavelets," in *2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2012, pp. 1349–1352.

[27] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 119–133, 2013.

[28] B. Rasti, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Hyperspectral image denoising using first order spectral roughness penalty in wavelet domain," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2458–2467, 2014.

[29] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral–spatial adaptive total variation model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 3660–3677, 2012.

[30] H. Zhang, "Hyperspectral image denoising with cubic total variation model," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 7, pp. 95–98, 2012.

[31] Y. Chang, L. Yan, H. Fang, and C. Luo, "Anisotropic spectral-spatial total variation model for multispectral remote sensing image destriping," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1852–1866, 2015.

[32] H. K. Aggarwal and A. Majumdar, "Hyperspectral image denoising using spatio-spectral total variation," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 442–446, 2016.

[33] T. Lu, S. Li, L. Fang, Y. Ma, and J. A. Benediktsson, "Spectral–spatial adaptive sparse representation for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 373–385, 2015.

[34] J. Li, Q. Yuan, H. Shen, and L. Zhang, "Noise removal from hyperspectral image with joint spectral–spatial distributed sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 9, pp. 5425–5439, 2016.

[35] X. Bai, F. Xu, L. Zhou, Y. Xing, L. Bai, and J. Zhou, "Nonlocal similarity based nonnegative tucker decomposition for hyperspectral image denoising," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 3, pp. 701–712, 2018.

[36] J. M. Nascimento and J. M. Bioucas-Dias, "Hyperspectral signal subspace estimation," in *2007 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2007, pp. 3225–3228.

[37] N. Renard, S. Bourennane, and J. Blanc-Talon, "Denoising and dimensionality reduction using multilinear tools for hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 2, pp. 138–142, 2008.

[38] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, and B. Zhang, "Decomposable nonlocal tensor dictionary learning for multispectral image denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2949–2956.

[39] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 4729–4743, 2013.

[40] W. He, H. Zhang, L. Zhang, and H. Shen, "Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 3050–3061, 2015.

[41] Y. Zhao and J. Yang, "Hyperspectral image denoising via sparsity and low rank," in *2013 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2013, pp. 1091–1094.

[42] B. Rasti, M. O. Ulfarsson, and P. Ghamisi, "Automatic hyperspectral image restoration using sparse and low-rank modeling," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2335–2339, 2017.

[43] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 3, pp. 730–742, 2018.

[44] G. Ma, T.-Z. Huang, J. Huang, and C.-C. Zheng, "Local low-rank and sparse representation for hyperspectral image denoising," *IEEE Access*, vol. 7, pp. 79 850–79 865, 2019.

[45] L. Sun, T. Zhan, Z. Wu, and B. Jeon, "A novel 3D anisotropic total variation regularized low rank method for hyperspectral image mixed denoising," *ISPRS International Journal of Geo-Information*, vol. 7, no. 10, p. 412, 2018.

[46] H. Fan, C. Li, Y. Guo, G. Kuang, and J. Ma, "Spatial–spectral total variation regularized low-rank tensor decomposition for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 6196–6213, 2018.

[47] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[48] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[49] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, "Unprocessing images for learned raw denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 036–11 045.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[51] Q. Yuan, Q. Zhang, J. Li, H. Shen, and L. Zhang, "Hyperspectral image denoising employing a spatialspectral deep residual convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1205–1218, 2018.

[52] Y. Chang, L. Yan, H. Fang, S. Zhong, and W. Liao, "HSI-DeNet: Hyperspectral image restoration via convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 667–682, 2018.

[53] A. Maffei, J. M. Haut, M. E. Paoletti, J. Plaza, L. Bruzzone, and A. Plaza, "A single model CNN for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–14, 2019.

[54] W. Liu and J. Lee, "A 3-D atrous convolution neural network for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5701–5715, 2019.

[55] Q. Shi, X. Tang, T. Yang, R. Liu, and L. Zhang, "Hyperspectral image denoising using a 3-D attention denoising network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 12, pp. 10 348–10 363, 2021.

[56] K. Wei, Y. Fu, and H. Huang, "3-D quasi-recurrent neural network for hyperspectral image denoising," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 363–375, 2020.

[57] T. Bodrito, A. Zouaoui, J. Chanussot, and J. Mairal, "A trainable spectral-spatial sparse coding model for hyperspectral image restoration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5430–5442, 2021.

[58] H. Zhang, H. Chen, G. Yang, and L. Zhang, "LR-Net: Low-rank spatial-spectral network for hyperspectral image denoising," *IEEE Transactions on Image Processing*, vol. 30, pp. 8743–8758, 2021.

[59] F. Xiong, J. Zhou, Q. Zhao, J. Lu, and Y. Qian, "MAC-Net: Model-aided nonlocal neural network for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[60] O. Sidorov and J. Yngve Hardeberg, "Deep hyperspectral prior: Single-image denoising, inpainting, super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 3844–3851.

[61] Y.-S. Luo, X.-L. Zhao, T.-X. Jiang, Y.-B. Zheng, and Y. Chang, "Hyperspectral mixed noise removal via spatial-spectral constrained unsupervised deep image prior," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9435–9449, 2021.

[62] Y.-C. Miao, X.-L. Zhao, X. Fu, J.-L. Wang, and Y.-B. Zheng, "Hyperspectral denoising using unsupervised disentangled spatiospectral deep priors," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[63] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[64] R. Imamura, T. Itasaka, and M. Okuda, "Zero-shot hyperspectral image denoising with separable image prior," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 1416–1420.

[65] G. Fu, F. Xiong, S. Tao, J. Lu, J. Zhou, and Y. Qian, "Learning a model-based deep hyperspectral denoiser from a single noisy hyperspectral image," in *2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021, pp. 4131–4134.

[66] G. Simone, A. Farina, F. C. Morabito, S. B. Serpico, and L. Bruzzone, "Image fusion techniques for remote sensing applications," *Information fusion*, vol. 3, no. 1, pp. 3–15, 2002.

[67] N. Joshi, M. Baumann, A. Ehammer, *et al.*, "A review of the application of optical and radar remote sensing data fusion to land use mapping and monitoring," *Remote Sensing*, vol. 8, no. 1, p. 70, 2016.

[68] C. Pohl and J. Van Genderen, *Remote sensing image fusion: A practical guide*. CRC Press, 2016.

[69] L. Alparone, B. Aiazzi, S. Baronti, and A. Garzelli, *Remote Sensing Image Fusion*. CRC Press, 2015.

[70] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, M. Selva, *et al.*, "Twenty-five years of pansharpening: A critical review and new developments," *Signal and Image Processing for Remote Sensing, 2nd Edition*, no. Cap. 27, pp. 533–548, 2012.

[71] G. Vivone, L. Alparone, J. Chanussot, *et al.*, "A critical comparison among pansharpening algorithms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2565–2586, 2014.

[72] G. Vivone, M. D. Mura, A. Garzelli, *et al.*, "A new benchmark based on recent advances in multispectral pansharpening: Revisiting pansharpening with classical and emerging pansharpening methods," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, pp. 53–91, 2020.

[73] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, and M. Selva, "Pansharpening of hyperspectral images: A critical analysis of requirements and assessment on simulated prisma data," in *Image and Signal Processing for Remote Sensing XIX*, SPIE, vol. 8892, 2013, pp. 889 203–889 207.

[74] L. Loncan, L. B. Almeida, J. Bioucas-Dias, *et al.*, "Comparison of nine hyperspectral pansharpening methods," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 1–4.

[75] L. Loncan, L. B. De Almeida, Bioucas-Dias, *et al.*, "Hyperspectral pansharpening: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 3, no. 3, pp. 27–46, 2015.

[76] M. Selva, B. Aiazzi, F. Butera, L. Chiarantini, and S. Baronti, "Hyper-sharpening of hyperspectral data: A first approach," in *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2014, pp. 1–4.

[77] ——, "Hyper-sharpening: A first approach on sim-ga data," *IEEE Journal of selected topics in applied earth observations and remote sensing*, vol. 8, no. 6, pp. 3008–3024, 2015.

[78] X. Lu, J. Zhang, X. Yu, W. Tang, T. Li, and Y. Zhang, "Hyper-sharpening based on spectral modulation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 5, pp. 1534–1548, 2019.

[79] N. Yokoya, C. Grohnfeldt, and J. Chanussot, "Hyperspectral and multispectral data fusion: A comparative review of the recent literature," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 2, pp. 29–56, 2017.

[80] S. E. Armannsson, M. O. Ulfarsson, J. Sigurdsson, H. V. Nguyen, and J. R. Sveinsson, "A comparison of optimized Sentinel-2 super-resolution methods using Wald's protocol and Bayesian optimization," *Remote Sensing*, vol. 13, no. 11, pp. 2192–2213, 2021.

[81] Q. Wang, G. A. Blackburn, A. O. Onojeghuo, *et al.*, "Fusion of Landsat 8 OLI and Sentinel-2 MSI data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3885–3899, 2017.

[82] A. Garzelli, F. Nencini, and L. Capobianco, "Optimal MMSE pan sharpening of very high resolution multispectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 1, pp. 228–236, 2007.

[83] T.-M. Tu, P. S. Huang, C.-L. Hung, and C.-P. Chang, "A fast intensity-hue-saturation fusion technique with spectral adjustment for IKONOS imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 4, pp. 309–312, 2004.

[84] B. Aiazzi, S. Baronti, and M. Selva, "Improving component substitution pan-sharpening through multivariate regression of MS + Pan data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3230–3239, 2007.

[85] P. Chavez, S. C. Sides, J. A. Anderson, *et al.*, "Comparison of three different methods to merge multiresolution and multispectral data-Landsat TM and SPOT panchromatic," *Photogrammetric Engineering and Remote Sensing*, vol. 57, no. 3, pp. 295–303, 1991.

[86] T. Ranchin and L. Wald, "Fusion of high spatial and spectral resolution images: The ARSIS concept and its implementation," *Photogrammetric Engineering and Remote Sensing*, vol. 66, no. 1, pp. 49–61, 2000.

[87] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, and M. Selva, "MTF-tailored multiscale fusion of high-resolution ms and pan imagery," *Photogrammetric Engineering & Remote Sensing*, vol. 72, no. 5, pp. 591–596, 2006.

[88] G. Vivone, R. Restaino, G. Licciardi, M. Dalla Mura, and J. Chanussot, "Multiresolution analysis and component substitution techniques for hyperspectral pansharpening," in *2014 IEEE Geoscience and Remote Sensing Symposium (IGARSS)*, 2014, pp. 2649–2652.

[89] Z. Chen, H. Pu, B. Wang, and G.-M. Jiang, "Fusion of hyperspectral and multispectral images: A novel framework based on generalization of pan-sharpening methods," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 8, pp. 1418–1422, 2014.

[90] H. Park, J. Choi, N. Park, and S. Choi, "Sharpening the VNIR and SWIR bands of Sentinel-2A imagery through modified selected and synthesized band schemes," *Remote Sensing*, vol. 9, no. 10, pp. 1080–1100, 2017.

[91] Q. Du, N. H. Younan, R. King, and V. P. Shah, "On the performance evaluation of pan-sharpening techniques," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 518–522, 2007.

[92] N. Yokoya, T. Yairi, and A. Iwasaki, "Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 528–537, 2011.

[93] M. Simoes, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "A convex formulation for hyperspectral image superresolution via subspace-based regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3373–3388, 2014.

[94] C. Lanaras, E. Baltsavias, and K. Schindler, "Hyperspectral super-resolution with spectral unmixing constraints," *Remote Sensing*, vol. 9, no. 11, pp. 1196–1220, 2017.

[95] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.

[96] J. Xue, Y.-Q. Zhao, Y. Bu, W. Liao, J. C.-W. Chan, and W. Philips, "Spatial-spectral structured sparse low-rank representation for hyperspectral image super-resolution," *IEEE Transactions on Image Processing*, vol. 30, pp. 3084–3097, 2021.

[97] C. Lanaras, J. Bioucas-Dias, E. Baltsavias, and K. Schindler, "Super-resolution of multispectral multiresolution images from a single sensor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 20–28.

[98] C. Paris, J. Bioucas-Dias, and L. Bruzzone, "A novel sharpening approach for superresolving multiresolution optical images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1545–1560, 2018.

[99] M. O. Ulfarsson, F. Palsson, M. Dalla Mura, and J. R. Sveinsson, "Sentinel-2 sharpening using a reduced-rank method," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6408–6420, 2019.

[100] C.-H. Lin and J. M. Bioucas-Dias, "An explicit and scene-adapted definition of convex self-similarity prior with application to unsupervised Sentinel-2 super-resolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3352–3365, 2019.

[101] K. Rong, L. Jiao, S. Wang, and F. Liu, "Pansharpening based on low-rank and sparse decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 12, pp. 4793–4805, 2014.

[102] F. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, "Model-based reduced-rank pansharpening," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 4, pp. 656–660, 2020.

[103] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "A new pansharpening algorithm based on total variation," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 318–322, 2013.

[104] T. Xu, T.-Z. Huang, L.-J. Deng, X.-L. Zhao, and J. Huang, "Hyperspectral image superresolution using unidirectional total variation with Tucker decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4381–4398, 2020.

[105] J. Duran, A. Buades, B. Coll, and C. Sbert, "A nonlocal variational model for pansharpening image fusion," *SIAM Journal on Imaging Sciences*, vol. 7, no. 2, pp. 761–796, 2014.

[106] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Nonlocal patch tensor sparse representation for hyperspectral image super-resolution," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 3034–3047, 2019.

[107] N. Akhtar, F. Shafait, and A. Mian, "Sparse spatio-spectral representation for hyperspectral image super-resolution," in *European Conference on Computer Vision*, Springer, 2014, pp. 63–78.

[108] M. R. Vicinanza, R. Restaino, G. Vivone, M. Dalla Mura, and J. Chanussot, "A pansharpening method based on the sparse representation of injected details," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 1, pp. 180–184, 2014.

[109] G. Masi, D. Cozzolino, L. Verdoliva, and G. Scarpa, "Pansharpening by convolutional neural networks," *Remote Sensing*, vol. 8, no. 7, pp. 594–616, 2016.

[110] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[111] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding, and J. Paisley, "PanNet: a deep network architecture for pan-sharpening," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5449–5457.

[112] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "Sentinel-2 image fusion using a deep residual network," *Remote Sensing*, vol. 10, no. 8, pp. 1290–1312, 2018.

[113] R. Dian, S. Li, A. Guo, and L. Fang, "Deep hyperspectral image sharpening," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5345–5355, 2018.

[114] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "Multispectral and hyperspectral image fusion using a 3-D-convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 639–643, 2017.

[115] L. Wang, T. Bi, and Y. Shi, "A frequency-separated 3D-CNN for hyperspectral image super-resolution," *IEEE Access*, vol. 8, pp. 86 367–86 379, 2020.

[116] Y. Fu, Z. Liang, and S. You, "Bidirectional 3D quasi-recurrent neural network for hyperspectral image super-resolution," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2674–2688, 2021.

[117] L. Chen, Z. Lai, G. Vivone, G. Jeon, J. Chanussot, and X. Yang, "ArbRPN: A bidirectional recurrent pansharpening network for multispectral images with arbitrary numbers of bands," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2021.

[118] R. Lu, B. Chen, Z. Cheng, and P. Wang, "RAFnet: Recurrent attention fusion network of hyperspectral and multispectral images," *Signal Processing*, vol. 177, p. 107 737, 2020.

[119] X. Zhong, Y. Qian, H. Liu, *et al.*, "Attention_FPNet: two-branch remote sensing image pansharpening network based on attention feature fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 11 879–11 891, 2021.

[120] J.-F. Hu, T.-Z. Huang, L.-J. Deng, T.-X. Jiang, G. Vivone, and J. Chanussot, "Hyperspectral image super-resolution via deep spatiospectral attention convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[121] S. T. Seydi, M. Amani, and A. Ghorbanian, "A dual attention convolutional neural network for crop classification using time-series Sentinel-2 imagery," *Remote Sensing*, vol. 14, no. 3, pp. 498–522, 2022.

[122] X. Meng, N. Wang, F. Shao, and S. Li, "Vision transformer for pansharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[123] J.-F. Hu, T.-Z. Huang, L.-J. Deng, H.-X. Dou, D. Hong, and G. Vivone, "Fusformer: A transformer-based fusion network for hyperspectral image super-resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[124] Q. Huang, W. Li, T. Hu, and R. Tao, "Hyperspectral image super-resolution using generative adversarial network and residual learning," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3012–3016.

[125] Q. Liu, H. Zhou, Q. Xu, X. Liu, and Y. Wang, "PSGAN: A generative adversarial network for remote sensing image pan-sharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 12, pp. 10 227–10 242, 2020.

[126] L. Salgueiro Romero, J. Marcello, and V. Vilaplana, "Super-resolution of Sentinel-2 imagery using generative adversarial networks," *Remote Sensing*, vol. 12, no. 15, pp. 2424–2451, 2020.

[127] S. A. Hussein, T. Tirer, and R. Giryes, "Correction filter for single image super-resolution: Robustifying off-the-shelf deep super-resolvers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1428–1437.

[128] T. Uezato, D. Hong, N. Yokoya, and W. He, "Guided deep decoder: Unsupervised image pair fusion," in *European Conference on Computer Vision*, Springer, 2020, pp. 87–102.

[129] K. Zheng, L. Gao, W. Liao, *et al.*, "Coupled convolutional neural network with adaptive response function learning for unsupervised hyperspectral super resolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 2487–2502, 2020.

[130] H. V. Nguyen, M. O. Ulfarsson, J. R. Sveinsson, and M. Dalla Mura, "Sentinel-2 sharpening using a single unsupervised convolutional neural network with MTF-based degradation model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 6882–6896, 2021.

[131] W. G. C. Bandara, J. M. J. Valanarasu, and V. M. Patel, "Hyperspectral pansharpening based on improved deep image prior and residual reconstruction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[132] W. Chen, X. Zheng, and X. Lu, "Hyperspectral image super-resolution with self-supervised spectral-spatial residual network," *Remote Sensing*, vol. 13, no. 7, pp. 1260–1282, 2021.

[133] M. Ciotola, S. Vitale, A. Mazza, G. Poggi, and G. Scarpa, "Pansharpening by convolutional neural networks in the full resolution framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.

[134] J. Yao, D. Hong, J. Chanussot, D. Meng, X. Zhu, and Z. Xu, "Cross-attention in coupled unmixing nets for unsupervised hyperspectral super-resolution," in *European Conference on Computer Vision*, Springer, 2020, pp. 208–224.

[135] J. Liu, Z. Wu, L. Xiao, and X.-J. Wu, "Model inspired autoencoder for unsupervised hyperspectral image super-resolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[136] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015. [Online]. Available: `http://arxiv.org/abs/1412.6980`.

[137] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The Annals of Statistics*, pp. 1135–1151, 1981.

[138] Y. C. Eldar, "Generalized SURE for exponential families: Applications to regularization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, 2008.

[139] R. Giryes, M. Elad, and Y. C. Eldar, "The projected gsure for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, pp. 407–422, 2011.

[140] T. Tirer and R. Giryes, "Back-projection based fidelity term for ill-posed linear inverse problems," *IEEE Transactions on Image Processing*, vol. 29, pp. 6164–6179, 2020.

[141] C. A. Metzler, A. Mousavi, R. Heckel, and R. G. Baraniuk, "Unsupervised learning with Stein's unbiased risk estimator," in *Proceeding of the International Biomedical and Astronomical Signal Processing Frontiers Workshop*, Feb. 2019, pp. 67–79.

[142] S. Soltanayev, R. Giryes, S. Y. Chun, and Y. C. Eldar, "On divergence approximations for unsupervised training of deep denoisers based on stein's unbiased risk estimator," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3592–3596.

[143] S. Ramani, T. Blu, and M. Unser, "Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1540–1554, 2008.

[144] J. M. Bioucas-Dias and J. M. Nascimento, "Hyperspectral subspace identification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 8, pp. 2435–2445, 2008.

[145] S. Mallat, *A wavelet tour of signal processing*. Academic Press, 1998, pp. 447–448.

[146] F. J. Anscombe, "The transformation of poisson, binomial and negative-binomial data," *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948.

[147] Q. Wang, W. Shi, Z. Li, and P. M. Atkinson, "Fusion of Sentinel-2 images," *Remote Sensing of Environment*, vol. 187, pp. 241–252, 2016.

[148] C. Lanaras, J. Bioucas-Dias, S. Galliani, E. Baltsavias, and K. Schindler, "Super-resolution of Sentinel-2 images: Learning a globally applicable deep neural network," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 305–319, 2018.

[149] M. Gargiulo, A. Mazza, R. Gaetano, G. Ruello, and G. Scarpa, "Fast super-resolution of 20 m Sentinel-2 bands using convolutional neural networks," *Remote Sensing*, vol. 11, no. 22, pp. 2635–2653, 2019.

[150] J. Wu, Z. He, and J. Hu, "Sentinel-2 sharpening via parallel residual network," *Remote Sensing*, vol. 12, no. 2, pp. 279–299, 2020.

[151] S. Clerk and M. team. "S2 MPC - data quality report." (2020), [Online]. Available: `https://sentinel.esa.int/documents/247904/685211/Sentinel-2_L1C_Data_Quality_Report`.

[152] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images," *Photogrammetric Engineering and Remote Sensing*, vol. 63, pp. 691–699, 1997.

[153] H. V. Nguyen, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral image denoising using SURE-based unsupervised convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 3369–3382, 2021.

[154] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[155] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.

[156] F. Palsson, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Quantitative quality evaluation of pansharpened imagery: Consistency versus synthesis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1247–1259, 2015.

[157] H. V. Nguyen, M. O. Ulfarsson, J. R. Sveinsson, and J. Sigurdsson, "Zero-shot Sentinel-2 sharpening using a symmetric skipped connection convolutional neural network," in *Proceedings of the IEEE 2020 International Geoscience and Remote Sensing Symposium*, 2020, pp. 617–620.

[158] V. Solo, "A sure-fired way to choose smoothing parameters in ill-conditioned inverse problems," in *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 3, 1996, pp. 89–92.

[159] Z. Liu, Y. Zheng, and X.-H. Han, "Unsupervised multispectral and hyperspectral image fusion with deep spatial and spectral priors," in *Proceedings of the Asian Conference on Computer Vision*, 2020. [Online]. Available: `https://openaccess.thecvf.com/content/ACCV2020W/MLCSA/html/Liu_Unsupervised_Multispectral_and_Hyperspectral_Image_Fusion_with_Deep_Spatial_and_ACCVW_2020_paper.html`.

[160] S. Abu-Hussein, T. Tirer, S. Y. Chun, Y. C. Eldar, and R. Giryes, "Image restoration by deep projected GSURE," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 3602–3611.

[161] J. Zukerman, T. Tirer, and R. Giryes, "BP-DIP: A backprojection based deep image prior," in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 675–679.

[162] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, 2016.

[163] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[164] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, 2015, pp. 234–241.

[165] M. Magnusson, J. Sigurdsson, S. E. Armansson, M. O. Ulfarsson, H. Deborah, and J. R. Sveinsson, "Creating RGB images from hyperspectral images using a color matching function," in *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2020, pp. 2045–2048.

[166] L. Wald, "Quality of high resolution synthesised images: Is there a simple criterion?" In *Proceeding of the third conference "Fusion of the Earth data: Merging points measurements, raster maps, and remotely sensed images*, 2000, pp. 99–103.

[167] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, “Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm,” *Summaries of the Third Annual JPL Airborne Geoscience Workshop*, vol. 1, pp. 147–149, 1992.

[168] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, ISSN: 1941-0042.