NORTHWESTERN POLYTECHNICAL UNIVERSITY

A multi-scale CNN for single image spectral super-resolution

by

Yiqi Yan

A thesis submitted in partial fulfillment for the
Bachelor of Engineering

in the

School of Computer Science
Department of Computer Information and Engineering

May 2018

Declaration of Authorship

I, Yiqi Yan, declare that this thesis titled, 'THESIS TITLE' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

"I can do all things."

Stephen Curry

NORTHWESTERN POLYTECHNICAL UNIVERSITY

Abstract

School of Computer Science
Department of Computer Information and Engineering

Bachelor of Engineering

by Yiqi Yan

Hyperspectral imaging enhances the solution of many visual problems but suffers from low-resolution image data. Due to the trade-off between spectral and spatial resolution, it is hard to directly get high spectral-spatial resolution data. In addition, building a high-resolution hyperspectral imaging system can be really costly. Therefore, computational super-resolution methods mean a lot in practice.

This thesis focuses on one type of super-resolution method, spectral super-resolution. We aim to produce a high-resolution hyperspectral image from a signal RGB observation. Mapping three discrete intensity values to a continuous spectrum is highly under-constrained. Fortunately, the inherent correlation of natural images serves as a nice prior to help solve this problem. In fact, for each candidate pixel, there often exist locally and non-locally similar pixels. In this thesis, we propose a novel multi-scale convolutional neural network to explicitly map the input RGB image into a hyperspectral image. Through symmetrically downsampling and upsampling the intermediate feature maps in a cascading paradigm, the local and non-local image information can be jointly encoded for spectral representation, ultimately improving the spectral reconstruction accuracy.

We do experiments on a large hyperspectral database and prove that our method achieves state-of-the-art performance with regards to both pixel-level accuracy and spectral similarity. What's more, we experimentally show that our method is much more robust in that it is less sensitive to hyper-parameters compared to previous methods.

Keywords: Hyperspectral imaging. Spectral super-resolution. Multi-scale convolutional neural networks.

Acknowledgements

First of all, I'd like to thank my thesis supervisor Professor Wei Wei for his mentorship, support, and infinite patience. My other collaborators, including Dr. Lei Zhang and Mr. Yong Li, also helped a lot over the past few months.

I would also like to thank my friends. Gratitude should be given to Yue Zhang, Minghang Li, Peiyu Liu, and Jianrui Xiao for backing me up when I was in a really bad situation last month, and all other friends' support during my college life should not be forgotten.

I am also grateful to Yixiao Zhang and Hao Ju from University of Electronic Science and Technology of China for their company when I was in a bad mood.

What' more, I would never have the opportunity to have a taste on computer vision and machine learning if I hadn't joined the NPU Soccer Robot Group. I had an unforgettable time with my teammates developing our own robots.

Of course, I would never miss my idol, Stephen Curry. His optimistic attitude towards life and strong will to break stereotypes enlighten my mind all the time.

Finally, thank Yiqi for never ever giving up.

Contents

List of Figures

List of Tables

Symbols

I_h	hyperspectral image
I_{rgb}	RGB image
I_e	the estimated hyperspectral image
H	the height of an image
W	the width of an image
C	number of bands in an hyperspectral image
$\boldsymbol{p}_h \epsilon\ \mathbb{R}^C$	hyperspectral pixel in real hyperspectral images
$\boldsymbol{p}_e \epsilon\ \mathbb{R}^C$	hyperspectral pixel in reconstructed hyperspectral images
$\boldsymbol{p}_l \epsilon\ \mathbb{R}^3$	RGB pixel
$RMSE_1, RMSE_2$	root mean square error[1]
$rRMSE_1, rRMSE_2$	relative root mean square error[1]
SAM	spectral angle mapper
$D_H = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_m\}$	hyperspectral dictionary with m signature atoms
$D_L = \{\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_m\}$	RGB dictionary with m signature atoms

[1] There are two different formulas for RMSE and rRMSE respectively

Chapter 1

Introduction

Hyperspectral imaging encodes the reflectance of the scene from dozens or hundreds of bands with a narrow wavelength interval (*e.g.* 10nm) into a hyperspectral image. Different from conventional images, each pixel in the hyperspectral image contains a continuous spectrum, thus allowing the acquisition of abundant spectral information. Since spectral responses reflect the characteristics of different kinds of materials at each observation point, hyperspectral images have been widely exploited to facilitate various applications in computer vision community, such as visual tracking [2], image segmentation [3], face recognition [4], document analysis [5, 6], scene classification [7, 8], anomaly detection [9, 10], and other general remote sensing tasks [11–14].

The ability to achieve such richness of information, however, comes with an unavoidable cost. There are two main challenges that limit the application of hyperspectral images. The first is the trade-off between spatial and spectral resolution. When shotting a hyperspectral image, a fewer number of photons are captured by each detector due to the narrower width of the spectral bands. In order to maintain a reasonable signal-to-noise ratio (SNR), the instantaneous field of view (IFOV) needs to be increased [15, 16]. This makes it really hard to get "fully high-resolution" image. The second disadvantage is the high cost of hyperspectral devices. This results from the requirement of recording a 3-dimensional data. In order to do this, some scanning operations must be performed spatially or spectrally, and careful elaboration of imaging devices is required. To address these two problems, many computational methods have been proposed, typically known as super-resolution.

This chapter will give a brief introduction of hyperspectral imaging technique, pointing out its pros and cons, followed by a review of existing super-resolution methods. Then we will introduce three public hyperspectral datasets. Finally, we will summarize our contributions.

1.1 Hyperspectral Imaging Techniques

Conventional imaging sensors produce images within several relatively broad wavelength. For example, RGB imaging sensors capture reflectance within three wavelength bands in the range of the visible light spectrum. On the contrary, hyperspectral sensors have the ability to collect data simultaneously in dozens or hundreds of narrow, adjacent spectral bands, as illustrated in Figure 1.1.

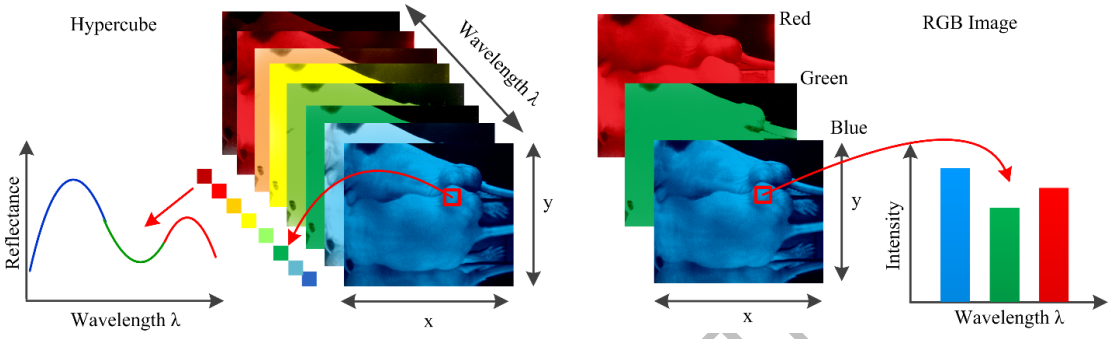


FIGURE 1.1: Illustration of RGB and hyperspectral imaging[2]

The acquired hyperspectral image is a 3-dimensional data cube, with spatial dimension x, y, and spectral dimension λ. In order to sample the hyperspectral cube from a continuous signal space, certain kind of scanning is performed along some specific dimensions. Technically speaking, there are four ways: spatial scanning, spectral scanning, non-scanning (snapshot imaging) and spatio-spectral scanning. Their difference is shown in Figure 1.2.

Spatial scanning In spatial scanning, a slit aperture is moved across a scene (alongside y direction) to capture image sections sequentially. At each scanning point, a 2-dimensional output is produced, representing a full slit spectrum (x, λ). This kind of device is a line-scan system, as each scanning position is a line-shaped area on the (x, y) plane. Therefore, stable mounts are required for "reconstructing" the image. The advantage of this scanning strategy is that it produces high (spatial) resolution, but it also gives rise to relatively high motion artifacts (caused by the scanning operation).

Spectral scanning spectral scanning is somewhat similar to spatial scanning, in that they both produce multiple 2-dimensional outputs to "reconstruct" the whole image. The only difference is that in spectral scanning each output represents a monochromatic spatial map (x, y) of the scene. This is achieved by inserting filters to select " color" (different wavelength bands). Spectral scanning produces high (spectral) resolution, and also results in motion artifacts.

[2] http://feilab.org/Research/Research_HSI.htm

Non-scanning This method is also called snapshot imaging, as it actually needs no scanning operation. All spatial and spectral attributes are captured in one single frame. The most prominent advantage of snapshot imaging is high throughput and quick acquisition. Since no scanning is needed, motion artifacts no longer exist. These benefits come at the cost of high computational efforts, and the manufacturing costs make it a challenge to get high-resolution images.

Spatio-spectral scanning Spatiospectral scanning it a combined version of spatial and spectral scanning. In this case, each 2-dimensional output is a "wavelength-coded" spatial map of the scene, where λ follows $\lambda = \lambda(y)$. This technique takes the advantages of spatial and spectral scanning and reduces their disadvantages to some degree.

No matter which imaging technique is utilized, the contradiction of spatial and spectral resolution always occur in practice. We can easily acquire an RGB image with very high spatial resolution, but this lacks rich spectral information. On the contrary, when we gather rich spectral information in hyperspectral images, the spatial resolution must be reduced. Due to this, reconstructing the reduced dimension by computation methods, typically known as super-resolution, is essential in practice. According to which dimension needs to be restored, there are two categories of super-resolution methods for hyperspectral images: spatial super-resolution and spectral super-resolution. We will discuss this in the next section.

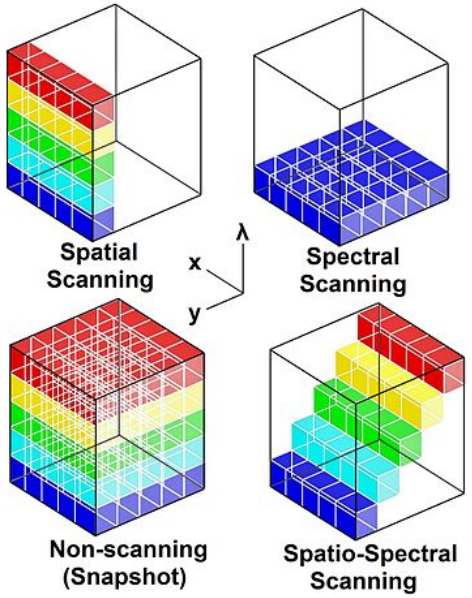


FIGURE 1.2: Four types of scanning methods for hyperspectral imaging

1.2 Existing Super-resolution Methods

1.2.1 Spatial Super-resolution

Fusion based super-resolution This category of methods fuse a high-resolution convention image (*e.g.*, panchromatic image, RGB image, or multispectral image) and a low-resolution hyperspectral image to produce a high-resolution hyperspectral image [17, 18]. Particularly, the fusion of panchromatic and hyperspectral images are known as hyperspectral pansharpening. Generally speaking, there are three classes of pansharpening methods: component substitution (CS), multiresolution analysis (MRA), and Bayesian methods. The **CS approach** relies on the substitution of a component of the hyperspectral image by the panchromatic image. The CS approach includes algorithms such as intensity-hue-saturation [19–21], principal component analysis [22–24] and Gram-Schmidt [25]. The **MRA approach** is based on the injection of spatial details into the hyperspectral data. The spatial details can be extracted through a multiresolution decomposition of the panchromatic image. There are several modalities of MRA: decimated wavelet transform [26], undecimated wavelet transform [27], Laplacian pyramid [28] and nonseparable transforms [29, 30]. The **Bayesian approach** relies on the use of posterior distribution of the full resolution target image given the observed hyperspectral and panchromatic images [31–33]. Hyperspectral pansharpening can be easily extended from panchromatic images to RGB/multispectral images by fusing each band separately using the conventional pansharpening methods and then synthesizing all bands to get high-resolution hyperspectral images [34–37].

Single image super-resolution Fusion based super-resolution methods require the simultaneous acquisition of two well-registered observations, which is always infeasible in practice. In recent years, some methods take efforts to directly increases the spatial resolution of a hyperspectral image. [38] used convolutional neural networks to encode both spatial context and spectral correlation for hyperspectral super-resolution. Furthermore, [39] proposed a three dimensional fully convolutional neural network (3D-FCNN) to better exploit the spectral correlation of neighboring bands, such that spectral distortion when directly applying traditional CNN algorithms in band-wise manners is alleviated. In addition, a sensor-specific mode is designed for the proposed 3D-FCNN such that none of the samples from the target scene are required for training, and fine-tuning by a small number of training samples from the target scene can further improve the performance of such a sensor-specific method. In [40, 41], a spectral difference convolutional neural network (SDCNN) was proposed to enhance spatial resolution. Spatial constraint strategy was utilized to correcting the spatial error while preserving the spectral information.

In [42], the author took the advantage of residual learning for spatial super-resolution. What's more, an extra term that calculates the spectral angle was introduced to the loss function.

1.2.2 Spectral Super-resolution

Compared to spatial super-resolution, relatively rare work has been done on spectral super-resolution. Here we briefly review existing methods.

Early imaging methods Early methods attempted to acquire hyperspectral images using RGB sensors under certain controlled circumstances. For example, [43] took the advantage of active lighting by using spectral filters before the illumination. This is only feasible under laboratory conditions. Similarly, [44, 45] were also limited to capturing RGB images under controlled lighting. [46] proposed an algorithm to combine multiple RGB images of the same scene. The idea was based on different spectral sensitivities of different camera sensors. However, this imaging system was designed using dedicated devices that needed to be carefully deposited.

Statistic based methods This line of research mainly focus on exploiting the inherent statistical distribution of the latent hyperspectral image as priors to guide the super-resolution. Most of these methods involve building overcomplete dictionaries and learning sparse coding coefficients to linearly combine the dictionary atoms. For example, in [47], Arad *et al.* leveraged image priors to build a dictionary using K-SVD [48]. At test time, orthogonal matching pursuit [49] was used to compute a sparse representation of the input RGB image. [50] proposed a new method inspired by A+ [51–53], where sparse coefficients are computed by explicitly solving a sparse least square problem. These methods directly exploit the whole image to build image prior, ignoring local and non-local structure information. What's more, since the image prior is often handcrafted or heuristically designed with shallow structure, these methods fail to generalize well in practice.

Learning based methods These methods directly learn a certain mapping function from the RGB image to a corresponding hyperspectral image. For example, [54] proposed a training based method using a radial basis function network. The input data was pre-processed with a white balancing function to alleviate the influence of different illumination. The total reconstruction accuracy is affected by the performance of this pre-processing stage. Recently, witnessing the great success of deep learning in many

other ill-posed inverse problems such as image denoising [55] and single image super-resolution [56], it is natural to consider using deep networks (especially convolutional neural networks) for spectral super-resolution. In [1], Galliani *et al*. exploited a variant of fully convolutional DenseNets (FC-DenseNets [57]) for spectral super-resolution. However, this method is sensitive to the hyper-parameters and its performance can still be further improved.

1.3 Hyperspectral Image Datasets

Large and high-quality hyperspectral image databases are essential for developing and testing computational methods. CAVE [58] and HARVARD [59] were commonly used in previous publications, while NTIRE2018 [60] is a recently released dataset. Their basic information is shown in Table 1.1.

CAVE CAVE dataset consists of 32 images with a spatial resolution of 512×512 and 31 spectral bands between 400 and 700 nm. The content of CAVE is a collection of diverse objects, including faces, fruits, paint, and textiles.

HARVARD HARVARD contains 50 images of indoor and outdoor scenes, captured using a commercial hyperspectral camera (Nuance FX). The spatial resolution is 1024×1024.

NTIRE2018 This dataset is extended from the ICVL dataset [47]. The ICVL dataset includes 203 images captured using Specim PS Kappa DX4 hyperspectral camera. Each image is of size 1392×1300 in spatial resolution and contains 519 spectral bands in the range of $400 \sim 1000nm$. In experiments, 31 successive bands ranging from $400 \sim 700nm$ with $10nm$ interval are extracted from each image for evaluation. In the NTIRE2018 challenge, this dataset is further extended by supplementing 53 extra images of the same spatial and spectral resolution. As a result, 256 high-resolution hyperspectral images are collected as the training data. In addition, another 5 hyperspectral images are further introduced as the test set. In the NTIRE2018 dataset, the corresponding RGB rendition is also provided for each image. Since all other databases pale in terms of the amounts and resolution of image data, all experiments in this thesis are performed on NTIRE2018.

TABLE 1.1: Basic information about three different hyperspectral image databases

	number of images	size	bands	spectral band
NTIRE2018	256 training + 5 test	1392×1300	31	$400 \sim 700nm$
CAVE	32	512×512	31	$400 \sim 700nm$
HARVARD	50	1024×1024	31	$420 \sim 720nm$

1.4 Our Contribution

In this paper, we aim to perform single image spectral super-resolution. It is challenging to accurately reconstruct a hyperspectral image from a single RGB observation, since mapping three discrete intensity values to a continuous spectrum is a highly ill-posed inverse problem (much information is lost when downsampling the latent spectrum). To address this problem, we propose to learn a complicated non-linear mapping function for spectral resolution with deep convolution neural networks. It has been shown that for a candidate pixel, there often exist abundant locally and no-locally similar pixels (*i.e.* exhibiting similar spectra) in the spatial domain. As a result, the color vectors $\big(r, g, b\big)$ corresponding to those similar pixels can be viewed as a group of downsampled observations of the latent spectra for the candidate pixel. Therefore, accurate spectrum reconstruction requires to explicit considering both the local and non-local information from the input RGB image. To this end, we develop a novel multi-scale convolution neural network. Our method jointly encodes the local and non-local image information through symmetrically downsampling and upsampling the intermediate feature maps in a cascading paradigm, enhancing the spectral reconstruction accuracy. We experimentally show that the proposed method can be easily trained in an end-to-end scheme and beat several state-of-the-art methods on a large hyperspectral image dataset with respect to various evaluation metrics.

Our contributions are twofold:

- We design a novel CNN architecture for spectral reconstruction. Our method is able to encode both local and non-local information simultaneously.
- We perform extensive experiments on a large hyperspectral dataset and prove that our method achieves state-of-the-art performance.

Chapter 2

Background Theory

In this chapter, necessary background knowledge is described. To begin with, we briefly summarize interpolation methods, which will serve as the most primitive baseline. In the rest of this chapter, we will focus on deep learning, especially convolutional neural networks(CNNs). First, basic concepts of CNNs are reviewed. Second, we revisit three classic architectures that are most relevant to this thesis. Then we summarize some commonly used techniques against overfitting, followed by a quick review of hardware and software implementation of deep learning.

2.1 Interpolation

Interpolation is a method of constructing new data points within a discrete set of known data points. If the known data points are some "downsampled signal", then interpolation serves as an upsampling method by reconstructing original data points. Interpolation algorithms often assume that the observed signal is a direct downsampled version of the original signal. This limits its application in more complicated cases. In this thesis, our goal is to perform spectral reconstruction, so we only focus on interpolation for 1-dimensional signals (Figure 2.1).

Nearest neighbor interpolation This method is very straightforward, directly setting the value of an interpolated point to the value of the nearest existing data point. The interpolated signal is "step-sized" (Figure 2.1 left). Nearest neighbor interpolation tends to increase noise and jaggies at boundaries. Clearly, it lacks the ability to recover rich spectral information.

Linear interpolation Given two data points, (x_a, y_a) and (x_b, y_b), the interpolant at (x, y) is given by the following equation.

$$\frac{y - y_a}{x - x_a} = \frac{y_b - y_a}{x_b - x_a} \tag{2.1}$$

This equation states that the slope of the line between (x_a, y_a) and (x, y) is the same as the slope of the line between (x_a, y_a) and (x_b, y_b). In other words, linear interpolation just places each interpolated point on the straight line between two neighboring data points (Figure 2.1 middle).

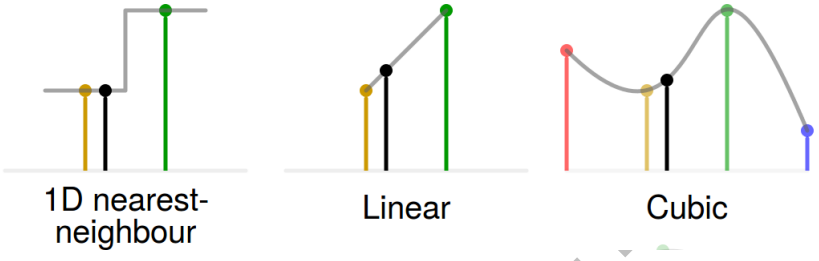


FIGURE 2.1: Comparison of three interpolation methods. Black dots represents the interpolated point. Red/yellow/green/blue dots correspond to known data points.

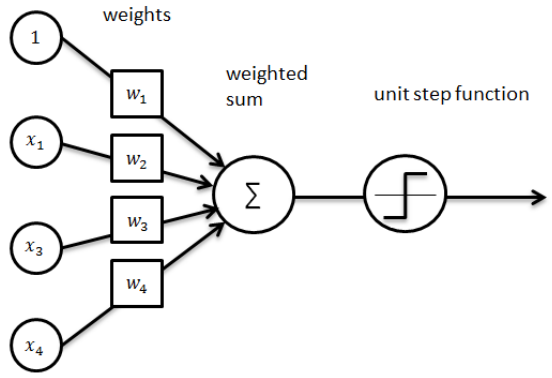
Spline interpolation Given a set of data points, polynomial interpolation tries to find a polynomial function that passes through the points of the dataset. The polynomial is of degree at most n. When $n = 3$, we get cubic interpolation (Figure 2.1 right). There are various methods to find such a polynomial, among which spline interpolation is commonly used. In spline interpolation, a polynomial of relatively low degree is assigned between each pair of data points. In the meantime, the boundaries of polynomials are continuously differentiable. Spline interpolation is often preferred over regular polynomial interpolation because the interpolation error can be made small even when using low degree polynomials for the spline.

2.2 Convolutional Neural Networks

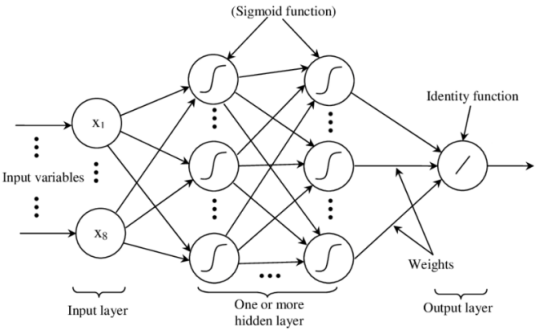
In the late 1950s, Frank Rosenblatt proposed the perceptron algorithm inspired by the mechanism of biological neurons [61]. This algorithm was later extended to multi-layer neural networks (or multi-layer perceptrons, MLPs). Generally speaking, each artificial neuron takes an input, performs a linear transformation followed by a non-linear activation function.

$$y = \sigma\left(\boldsymbol{W}^T \boldsymbol{x} + b\right) \tag{2.2}$$

For the primitive perceptron model (Figure 2.2 a), there is only one such neuron, and the activation function is a unit step function. This activation function states that the neuron should be "activated" according to a specific threshold. In MLPs (Figure 2.2 b), however, each layer may contain more than one neurons, and new kinds of activation functions are exploited (*e.g.* sigmoid function). The training of these models was infeasible until the invention of backpropagation [62] and gradient descent.



(a) Perceptron (with 3 input nodes and 1 bias term)



(b) 3-layer multi-layer perceptrons (MLPs)

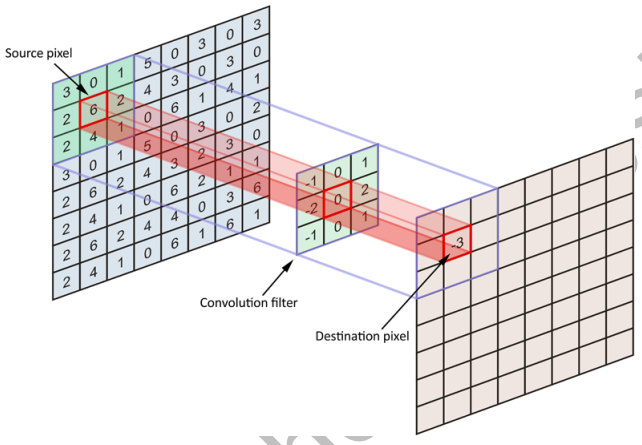
FIGURE 2.2: The perceptron models

Conventional neural network models don't scale well on image data. The image data is multidimensional and partially correlated. On the one hand, MLPs result in an exploding number of parameters when handling high dimensional data. On the other hand, ignoring spatial correlation means losing lots of structural information. In [63], a novel kind of model called convolutional neural networks (CNNs) was proposed to analyze image data. There are two main differences between CNNs and MLPs.

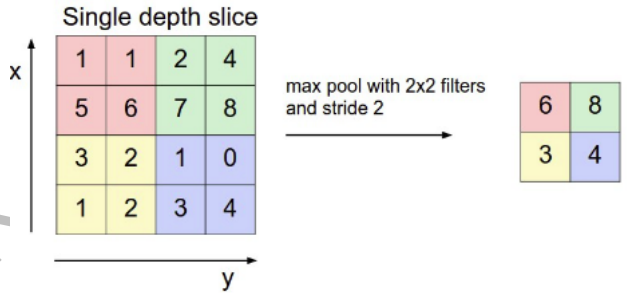
- In CNNs the weights are shared. In each layer, convolutional operations are performed on the input, and the kernels (or filters) are locally connected, *i.e.* different parts of the inputs "share" the same set of parameters. The advantages of weight sharing are obvious. First, it largely reduces the number of parameters, making it possible to build deeper networks. Second, it utilizes the inner correlation of

image data and maintains structural information. An example of convolutional operation is shown in Figure 2.3 a.

- Besides convolutional operations, another important characteristic of CNNs is the incorporation of pooling layers. Its function is to progressively reduce the spatial size of the hidden features and reduce computation in the network. It can also induce a certain degree of rotation and shift invariant. There are two kinds of pooling operations, max-pooling and average-pooling. The former is most commonly used. An example of max-pooling is shown in Figure 2.3 b.



(a) Convolutional Operation



(b) Max-pooling Operation

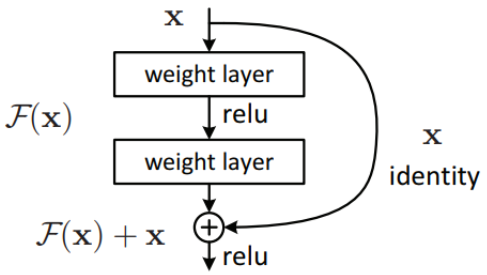
FIGURE 2.3: Basic building blocks of CNNs

Over the past few years, several classic CNN architectures have been proposed, including LeNet [63], AlexNet [64], VGGNet [65], ResNet [66, 67], DenseNet [68], *etc.* The last two are most relevant to the models used in this thesis. In the next section, we will give a brief introduction of ResNet and DenseNet.

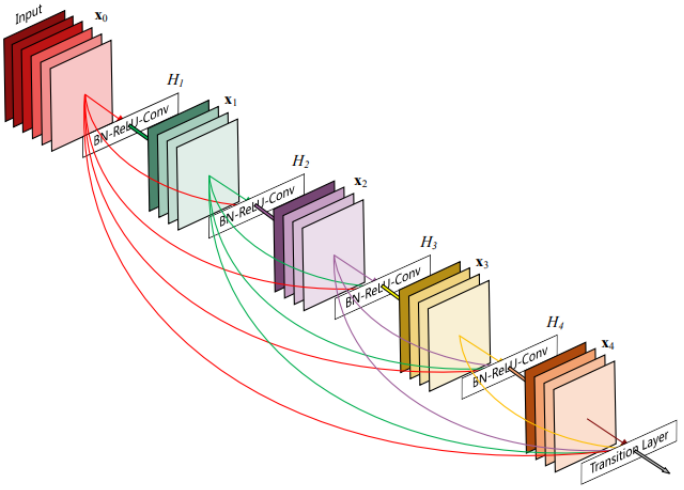
2.3 ResNet and DenseNet Architecture

Deep Residual Network (ResNet) was the winner of ImageNet Classification Challenge in 2015. Instead of directly learning some "mapping functions", the author reformulated the layers as explicitly learning residual functions with reference to the inputs. Inducing these "residual blocks" with skip connections (Figure 2.4 a) makes it easier to optimize much deeper CNNs than before (as deep as hundreds of layers).

Densely connected convolutional networks (DenseNets) use dense connections rather than "sparse" skip layers. In one dense block (Figure 2.4 b), the input of each layer is the combination of the outputs from every early layer. Unlike ResNet, DenseNet utilizes concatenation operations instead of element-wise additions to combine features from different layers. The advantages of DenseNet includes the alleviation of vanishing-gradient, efficient feature reuse, and significant parameter reduction.



(a) Residual block: the building block of ResNet



(b) Dense block: the building block of DenseNet

FIGURE 2.4: Residual block and dense block

2.4 Reducing Overfitting

Overfitting is a critical challenge in all kinds of deep learning methods. This problem mainly results from limited amounts of training data. Besides gathering more data, there are some nice techniques to reduce overfitting and better train deep networks.

Data augmentation For low-level vision tasks such as image denoising and super-resolution, it is common to train a model with sub-images (or "patches") extracted from the original data. In this way, we can get multiple times as many as training samples without gathering new data. In addition, other augmentation methods such as resizing, rotating, and adding noises are often used.

Weight penalty This is also known as regularization. By adding a term in loss function with respect to weights, parameters whose values go beyond a reasonable range are heavily penalized. Weight penalty prefers "diffuse weights", encouraging the network to use all of its inputs a little rather than some of its inputs a lot.

Dropout Overfitting occurs due to too many learnable parameters compared to relatively limited data. Dropout [69] means to randomly deactivate a fraction of neurons when training a deep model. This is somewhat equivalent to adding some noise to each hidden layer's activations.

Batch normalization Batch normalization [70] works by normalizing the output of a previous activation layer before passing it to the next stage. It reduces the amount by what the activation of hidden layers shift around ("covariance shift").

2.5 Hardware and Software Implementation

Nowadays, GPU acceleration with the support of CUDA software makes training deep learning models more and more efficient. What's more, it has been a trend for large group/companies to turn their deep learning frameworks into open source projects. Implementing and validating a new model has been a lot more straightforward. The most popular deep learning frameworks are Tensorflow [71] and PyTorch [72], supported by Google and Facebook respectively.

In this thesis, we use PyTorch for its flexibility to build dynamic computational graphs. As for the hardware platform, we have access to eight GTX 1080 Ti GPUs.

Chapter 3

The Proposed and Comparison Methods

This chapter includes detailed information about the comparision methods in this thesis: the sparse coding method in [47] (Arad *et al.*), A+ [50], and the deep learning method in [1] (Galliani *et al.*). Following this, we describe the proposed multi-scale convolutional neural network.

3.1 Comparison Methods

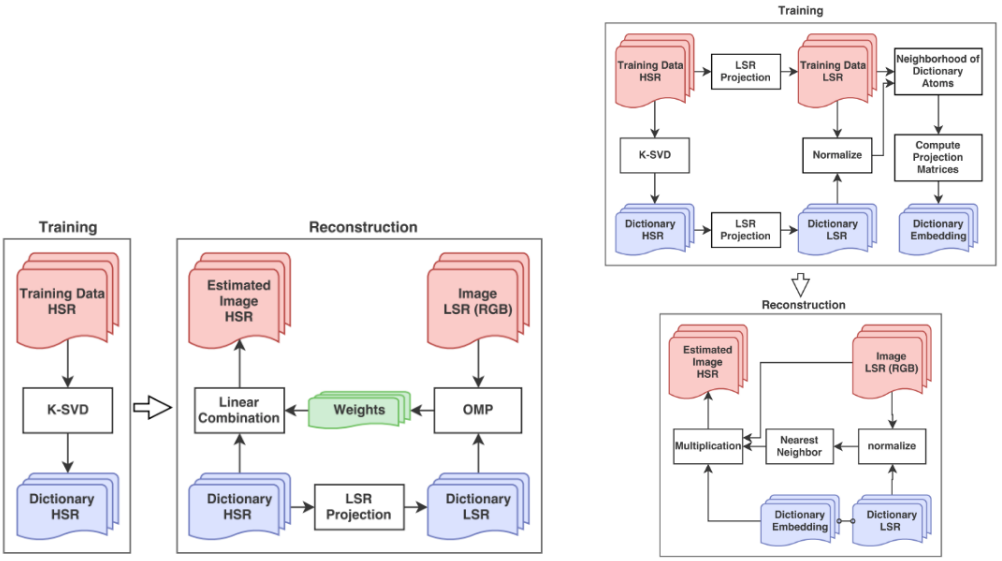
3.1.1 Sparse Coding Based Methods

Arad *et al.* and A+ [47, 50] are both based on dictionary learning and sparse coding. Their diagrams are shown in Figure 3.1.

Arad *et al.* At the training stage, an overcomplete dictionary with m atoms is built from a collection of hyperspectral images (training data) using K-SVD [48]. These atoms lie in the space of high spectral resolution (HSR).

$$D_H = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_m\} \tag{3.1}$$

Since the spectral response function (*i.e.* the projection matrix from the hyperspectral image to the corresponding RGB image) is assumed to be perfectly known, the hyperspectral dictionary can be projected to low spectral resolution (LSR) space.



(a) Arad *et al.* (b) A+

FIGURE 3.1: Diagrams of sparse coding based methods

$$D_L = \{\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_m\} \tag{3.2}$$

When it comes to reconstruction phase, the first step is to linearly decompose each RGB pixel $\boldsymbol{p}_l = (r, g, b)$ over D_L via orthogonal matching pursuit (OMP [49]), *i.e.* to find a weight vector $\boldsymbol{w}$ such that

$$D_L \cdot \boldsymbol{w} = \boldsymbol{p}_l \tag{3.3}$$

Having computed the decomposition coefficients $\boldsymbol{w}$, the corresponding hyperspectral pixel can be reconstructed.

$$\boldsymbol{p}_h = D_H \cdot \boldsymbol{w} \tag{3.4}$$

A+ algorithm A+ [51–53] was originally proposed for single image super-resolution. [50] extends it to spectral super-resolution and keeps the name "A+". Similar to Arad *et al.*, an overcomplete dictionary is built using K-SVD during the training stage. In A+, both the dictionary and the image data are projected to LSR space. For each LSR dictionary atom $\boldsymbol{l}_i$, a sparse coefficient $\boldsymbol{\alpha}$ is then computed by minimizing the least square error of the linear combination of its nearest neighbors ($\boldsymbol{N}_l$) with respect to LSR image data $\boldsymbol{y}_l$.

$$\min_{\alpha} \|\boldsymbol{y}_l - \boldsymbol{N}_l\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_2^2 \tag{3.5}$$

There exists a closed form solution for Equation 3.5.

$$\boldsymbol{\alpha} = \left(\boldsymbol{N}_l^T\boldsymbol{N}_l + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{N}_l^T \cdot \boldsymbol{y}_l \tag{3.6}$$

Similar to LSR space, let $\boldsymbol{N}_h$ donate the nearest neighbors of the hyperspectral atom. Due to the correspondence between HSR and LSR space, the following equation is satisfied.

$$\boldsymbol{y}_h = \boldsymbol{N}_h\boldsymbol{\alpha} \tag{3.7}$$

If we define a projection matrix $\boldsymbol{P}_i$ as follows:

$$\boldsymbol{P}_i = \boldsymbol{N}_h \cdot \left(\boldsymbol{N}_l^T\boldsymbol{N}_l + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{N}_l^T \tag{3.8}$$

then it is easy to tell that $\boldsymbol{P}_i$ is the projection matrix from LSR to HSR data. In fact, combining Equation 3.6, 3.7, 3.8, we can get:

$$\boldsymbol{y}_h = \boldsymbol{P}_i \cdot \boldsymbol{y}_l \tag{3.9}$$

Therefore, after offline computing and storing all the projection matrices, RGB samples can be embedded into hyperspectral space at the reconstruction stage.

3.1.2 Deep Learning Based Methods

Galliani *et al*. [1] utilized a variant of fully convolutional DenseNets (FC-DenseNets [57]) for spectral reconstruction. This network architecture was originally meant for image segmentation. It takes the advantage of DenseNets structure [68]. Figure 3.2 is a brief illustration of the network, and the complete composition is shown in Table 3.2. There are three basic building blocks in the network (Table 3.1).

Dense block (DB) Within each block, each layer creates k feature maps, which are concatenated to the input feature of the layer. One layer within the dense block is a combination of batch normalization, leaky ReLU, 3×3 convolution, and dropout

(Table 3.1 a). The output of the block is the concatenation of the outputs of all layers. In [1], k is set to 16, and each block consists of 4 layers, and thus each block creates an output containing 64 feature maps.

Transition down block (TD) The TD block in the downsampling path reduces the spatial resolution of the feature map. Rather than merely exploiting max-pooling, one TD block actually stacks various other operations before pooling, including batch normalization, leaky ReLU, 3×3 convolution (which conserves the number of feature maps), and dropout (Table 3.1 b).

Transition up block (TU) Galliani *et al*. used pixel shuffle (also known as sub-pixel convolution [73]), to upsample feature maps (Table 3.1 c). This is different from the original FC-DenseNets, which used transposed convolution. Pixel shuffle itself doesn't have learnable parameters, so it helps reduce overfitting. It also alleviates checkboard artifacts commonly caused by transposed convolution.

3.2 Proposed Method

In this section, we give a brief description of the basic components of our method. Following this, the complete network architecture is proposed.

3.2.1 Building Blocks

There are three basic building blocks in our network: double convolution (Double Conv, Table 3.3 a), downsample block (Table 3.3 b), and upsample block (Table 3.3 c).

TABLE 3.1: Basic elements of fully convolutional DenseNets

(a)	(b)	(c)
One Layer in Dense Block	**Transition Down (TD)**	**Transition Down (TD)**
Batch normalization	Batch normalization	Pixel shuffle
Leaky ReLU	Leaky ReLU	
3×3 convolution	1×1 convolution	
Dropout	Dropout	
	2×2 max-pooling	

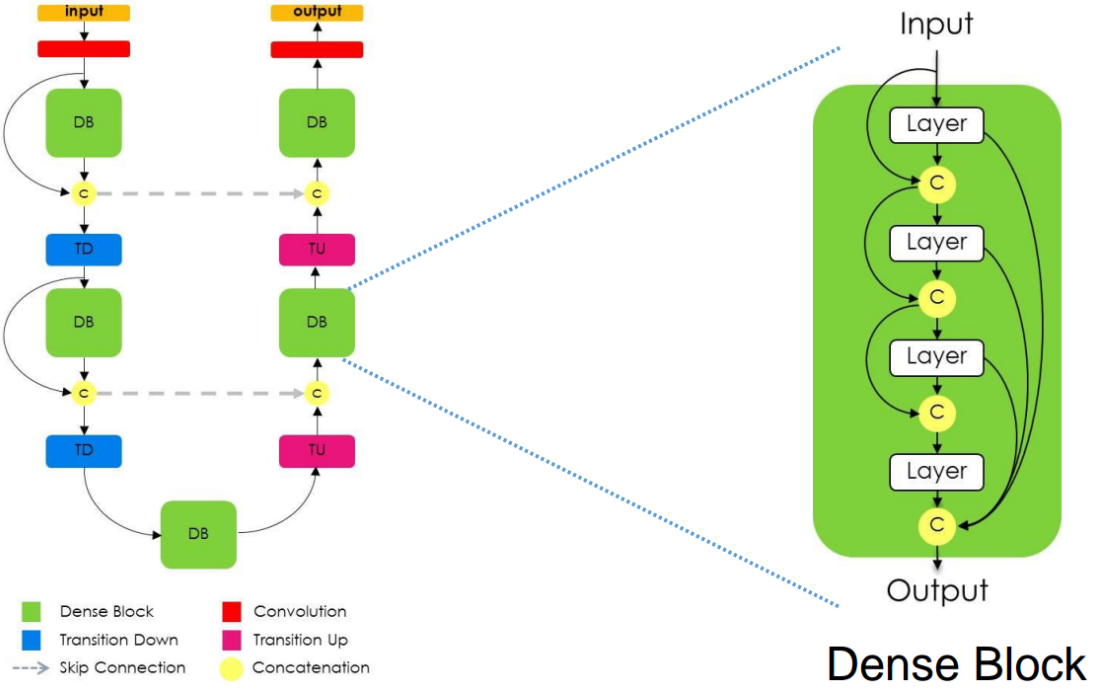


FIGURE 3.2: Illustration of fully convolutional DenseNets

TABLE 3.2: The complete composition of the method in [1]. Concatenation operations are not shown.

	Network components	Number of features
RGB Input	Input	3
	3×3 convolution	64
Downsampling Path	DB + TD	128
	DB + TD	192
	DB + TD	256
	DB + TD	320
	DB + TD	384
Bootleneck	DB	448
Uownsampling Path	TU+DB	400
	TU+DB	326
	TU+DB	272
	TU+DB	208
	TU+DB	144
Hyperspectral Output	3×3 convolution	31
	Output	31

Double Conv block This type of block consists of two 3×3 convolutions. Each of them is followed by batch normalization, leaky ReLU and dropout. We exploit batch normalization and dropout to address overfitting.

Downsample block Downsampling is performed using a regular max-pooling layer. It reduces the spatial size of the feature and enlarges the receptive field of the network.

TABLE 3.3: Basic elements of the proposed method

Double Conv
3×3 convolution
Batch normalization
Leaky ReLU
Dropout
3×3 convolution
Batch normalization
Leaky ReLU
Dropout

Downsample
2×2 max-pooling

Upsample
Pixel shuffle

Upsample block Similar to FC-DenseNets, we use pixel shuffle for feature upsampling to improve overfitting and alleviates checkboard artifacts.

3.2.2 Network Architecture

Figure 3.3 demonstrates the structure of our network. We follow the encoder-decoder pattern. For the **encoder** part, each downsampling step consists of a "Double Conv" with a downsample block. The spatial size is progressively reduced, and the number of features is doubled at each step. The **decoder** is symmetric to the encoder path. Every step in the decoder path consists of an upsampling operation followed by a "Double Conv" block. The spatial size of the features is recovered, while the number of features is halved every step. Finally, a 1×1 convolution maps the output feature to the reconstructed 31-band hyperspectral image. In addition to the feedforward path, skip connections are used to concatenate the corresponding feature maps of the encoder and decoder.

Our method naturally fits the task of spectral reconstruction. The encoder can be interpreted as extracting features from RGB images. During the downsampling process, the progressive increase of receptive field allows the network to "see" larger scale of pixels, and this non-local information is encoded by the increasing features. The decoder represents reconstructing hyperspectral images based on these deep and compact features. The skip connections with concatenations are essential for inducing multi-scale information and yielding better estimation of the spectra.

3.2.3 Discussion

The U-Net architecture [74] proposed for biomedical image segmentation is similar to our network. Here we summerize the main differences of these two networks.

FIGURE 3.3: Diagram of the proposed method. "Conv m" represents convolutional layers with an output of m feature maps. We use 3×3 convolution in green blocks and 1×1 convolution in the red block. Gray arrows represent feature concatenation.

- We use zero padding for convolution to keep the spatial size unchanged. In the original U-Net, feature cropping is required when concatenating features because of the use of unpadded convolution. Our goal is to avoid losing border features.
- We exploit batch normalization and dropout after each convolution to address the overfitting problem.
- We use Leaky ReLU instead of regular ReLU as the non-linear activation function.
- We use pixel shuffle instead of transposed convolution to upsample the intermediate features. This decreases the amounts of learnable parameters and avoids checkboard artifacts.

Chapter 4

Experiments

4.1 Implementation Details

To demonstrate the effectiveness of the proposed method, we compare it with four spectral super-resolution methods, including spline interpolation, Arad *et al*. [47], A+ [50], Galliani *et al*. [1]. [47, 50] are implemented by the codes released by the authors. Since there is no code released for [1], we reimplement it in this study. In the following, we will give the implementation details of each method.

Spline interpolation The interpolation algorithm serves as the most primitive baseline in this study. Specifically, for each RGB pixel $\boldsymbol{p}_l = (r, g, b)$, we use spline interpolation to upsample it and obtain a 31-dimensional spectrum ($\boldsymbol{p}_h$). According to the visible spectrum (Figure 4.1), the r, g, b values of an RGB pixel are assigned to $700nm$, $550nm$, and $450nm$, respectively.

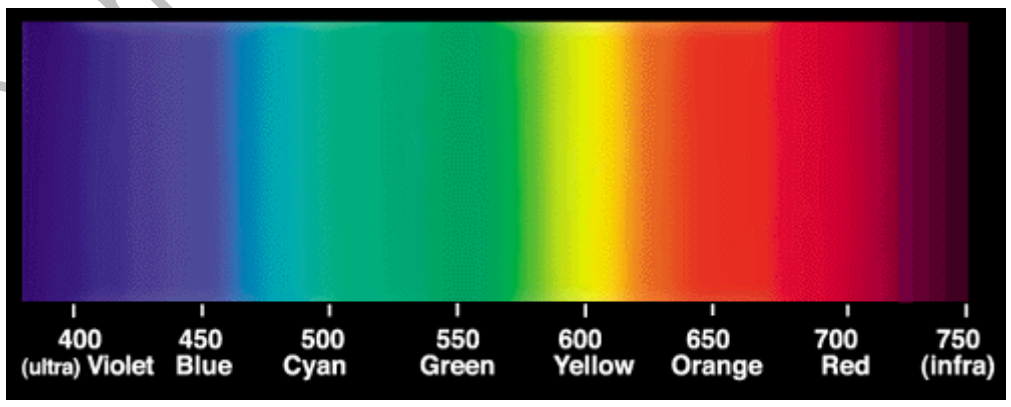


FIGURE 4.1: The spectrum of visible light.[3]

[3] http://www.gamonline.com/catalog/colortheory/visible.php

Arad *et al.* and A+ The low spectral resolution image is assumed to be a directly downsampled version of the corresponding hyperspectral image using some specific linear projection matrix. In [47, 50] this matrix is required to be perfectly known. In our experiments, we fit the projection matrix using training data with conventional linear regression.

Galliani *et al.* and our method We experimentally find the optimal set of hyper-parameters for both methods (summarized in Table 4.1). 50% dropout is applied to Galliani *et al.*, while our method utilizes 20% dropout rate. All the leaky ReLU activation functions are applied with a negative slope of 0.2. We train the networks for 100 epochs using Adam optimizer with 10^{-6} regularization. Weight initialization and learning rate vary for different methods. For Galliani *et al.*, the weights are initialized via HeUniform [75], and the learning rate is set to 2×10^{-3} for the first 50 epochs, decayed to 2×10^{-4} for the next 50 epochs. As for our method, we use HeNormal initialization [75]. The initial learning rate is 5×10^{-5} and is multiplied by 0.93 every 10 epochs. We perform data augmentation by extracting patches of size 64×64 with a stride of 40 pixels from training data. The total amount of training samples is over $267,000$. At the test phase, we directly feed the whole image to the network and get the estimated hyperspectral image in one single forward pass.

TABLE 4.1: Implementation details of deep learning based methods

	Galliani *et al.*	Ours
Dropout rate	0.5	0.2
Slope for leaky ReLU	0.2	0.2
Initial learning rate	2×10^{-3}	5×10^{-5}
Weight penalty	1×10^{-6}	1×10^{-6}
Weight initialization	HeUniform	HeNormal

4.2 Evalutation Metrics

To quantitatively evaluate the performance of the proposed method, we adopt the following two categories of evaluation metrics.

Pixel-level reconstruction error We follow [50] to use absolute and relative root-mean-square error (RMSE and rRMSE) as quantitative measurement for reconstruction accuracy. Let $I_h^{(i)}$ and $I_e^{(i)}$ denote the ith scalar element of the real and estimated hyperspectral images, $\bar{I}_h$ is the average of all elements in I_h, and n is the total number of elements in one hyperspectral image. There are two formulas for RMSE and rRMSE respectively.

$$RMSE_1 = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\left(I_h^{(i)} - I_e^{(i)}\right)^2} \tag{4.1}$$

$$RMSE_2 = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(I_h^{(i)} - I_e^{(i)}\right)^2} \tag{4.2}$$

$$rRMSE_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{\sqrt{\left(I_h^{(i)} - I_e^{(i)}\right)^2}}{I_h^{(i)}} \tag{4.3}$$

$$rRMSE_2 = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{\left(I_h^{(i)} - I_e^{(i)}\right)^2}{\bar{I}_h^2}} \tag{4.4}$$

Spectral similarity Since the key for spectral super-resolution is to reconstruct the spectra, we also use spectral angle mapper (SAM) to evaluate the performance of different methods. SAM calculates the average spectral angle between the spectra of real and estimated hyperspectral images. Let $\boldsymbol{p}_h^{(j)}, \boldsymbol{p}_e^{(j)} \epsilon\, \mathbb{R}^C$ represents the spectra of the jth hyperspectral pixel in real and estimated hyperspectral images (C is the number of bands), and m is the total number of pixels within an image. The SAM value can be computed as follows.

$$SAM = \frac{1}{m}cos^{-1}\left(\sum_{j=1}^{m}\frac{(\boldsymbol{p}_h^{(j)})^T \cdot \boldsymbol{p}_e^{(j)}}{\left\|\boldsymbol{p}_h^{(j)}\right\|_2 \cdot \left\|\boldsymbol{p}_e^{(j)}\right\|_2}\right) \tag{4.5}$$

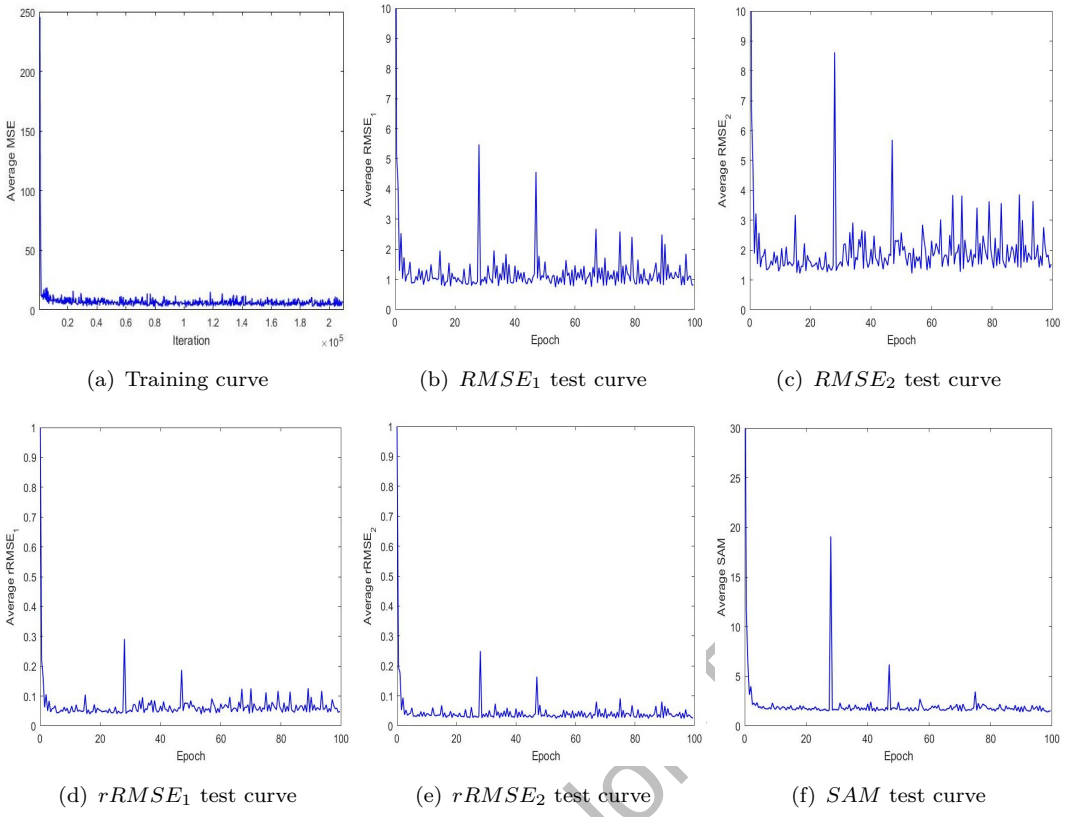
4.3 Experimental Results

4.3.1 Convergence Analysis

We plot the curve of MSE loss on the training set and the curves of five evaluation metrics computed on the test set in Figure 4.2. It can be seen that both the training loss and the value of metrics gradually decrease and ultimately converge with the proceeding of the training. This demonstrates that the proposed multi-scale convolution neural network converges well.

4.3.2 Quantitative Results

Table 4.2 provides the quantitative results of our method and all baseline methods. It can be seen that our model outperforms all competitors with regards to $RMSE_1$ and



(a) Training curve (b) $RMSE_1$ test curve (c) $RMSE_2$ test curve

(d) $rRMSE_1$ test curve (e) $rRMSE_2$ test curve (f) SAM test curve

FIGURE 4.2: Training and test curves.

$rRMSE_1$, and produces comparable results to Galliani *et al*. on $RMSE_2$ and $rRMSE_2$. More importantly, our method surpasses all the others with respect to spectral angle mapper. This clearly proves that our model reconstructs spectra more accurately than other competitors. It is worth pointing out that pixel-level reconstruction error (absolute and relative $RMSE$) is not necessarily positively correlated with spectral angle mapper (SAM). For example, when the pixels of an image are shuffled, $RMSE$ and $rRMSE$ will remain the same, while SAM will change completely. According to the results in Table 4.2, we can find that our finely designed network enhances spectral super-resolution from both aspects, *viz*., yielding better results on both average root-mean-square error and spectral angle similarity.

4.3.3 Visual Results

To further clarify the superiority in reconstruction accuracy. We show the absolute reconstruction error of every test image in Figure 4.5. The error is summarized over all bands of the hyperspectral image. Since A+ outperforms Arad *et al*. in terms of any evaluation metric, we use A+ to represent the sparse coding methods. It can be seen

that our method yields smoother reconstructed images as well as lower reconstruction error than other competitors.

In addition, we randomly choose three test images and plot the real and reconstructed spectra for four pixels in Figure 4.3 to further demonstrate the effectiveness of the proposed method in spectrum reconstruction. It can be seen that only slight difference exists between the reconstructed spectra and the ground truth.

According to these results above, we can conclude that the proposed method is effective in spectral super-resolution and outperforms several state-of-the-art competitors.

TABLE 4.2: Quantitative results on each test image.

			$RMSE_1$			
	BGU_00257	BGU_00259	BGU_00261	BGU_00263	BGU_00265	Average
Interpolation	1.8622	1.7198	2.8419	1.3657	1.9376	1.9454
Arad *et al.*	1.7930	1.4700	1.6592	1.8987	1.2559	1.6154
A+	1.3054	1.3572	1.3659	1.4884	0.9769	1.2988
Galliani *et al.*	0.7330	0.7922	0.8606	0.5786	**0.8276**	0.7584
Our	**0.6172**	**0.6865**	**0.9425**	**0.5049**	0.8375	**0.7177**
			$RMSE_2$			
	BGU_00257	BGU_00259	BGU_00261	BGU_00263	BGU_00265	Average
Interpolation	3.0774	2.9878	4.1453	2.0874	3.9522	3.2500
Arad *et al.*	3.4618	2.3534	2.6236	2.5750	2.0169	2.6061
A+	2.1911	1.9572	1.9364	2.0488	1.3344	1.8936
Galliani *et al.*	1.2381	**1.2077**	**1.2577**	0.8381	**1.6810**	**1.2445**
Ours	**0.9768**	1.3417	1.6035	**0.7396**	1.7879	1.2899
			$rRMSE_1$			
	BGU_00257	BGU_00259	BGU_00261	BGU_00263	BGU_00265	Average
Interpolation	0.0658	0.0518	0.0732	0.0530	0.0612	0.0610
Arad *et al.*	0.0807	0.0627	0.0624	0.0662	0.0560	0.0656
A+	0.0580	0.0589	0.0612	0.0614	0.0457	0.0570
Galliani *et al.*	0.0261	0.0268	0.0254	0.0237	0.0289	0.0262
Ours	**0.0235**	**0.0216**	**0.0230**	**0.0205**	**0.0278**	**0.0233**
			$rRMSE_2$			
	BGU_00257	BGU_00259	BGU_00261	BGU_00263	BGU_00265	Average
Interpolation	0.1058	0.0933	0.1103	0.0759	0.1338	0.1038
Arad *et al.*	0.1172	0.0809	0.0819	0.0685	0.0733	0.0844
A+	0.0580	0.0589	0.0612	0.0614	0.0457	0.0610
Galliani *et al.*	0.0453	**0.0372**	**0.0331**	0.0317	**0.0562**	**0.0407**
Ours	**0.0357**	0.0413	0.0422	**0.0280**	0.0598	0.0414
			SAM (degree)			
	BGU_00257	BGU_00259	BGU_00261	BGU_00263	BGU_00265	Average
Interpolation	3.9620	3.0304	4.2962	3.1900	3.9281	3.6813
Arad *et al.*	4.2667	3.7279	3.4726	3.3912	3.3699	3.6457
A+	3.2952	3.5812	3.2952	3.0256	3.2952	3.2985
Galliani *et al.*	1.4725	1.5013	**1.4802**	1.4844	**1.8229**	1.5523
Ours	**1.3305**	**1.2458**	1.7197	**1.1360**	1.9046	**1.4673**

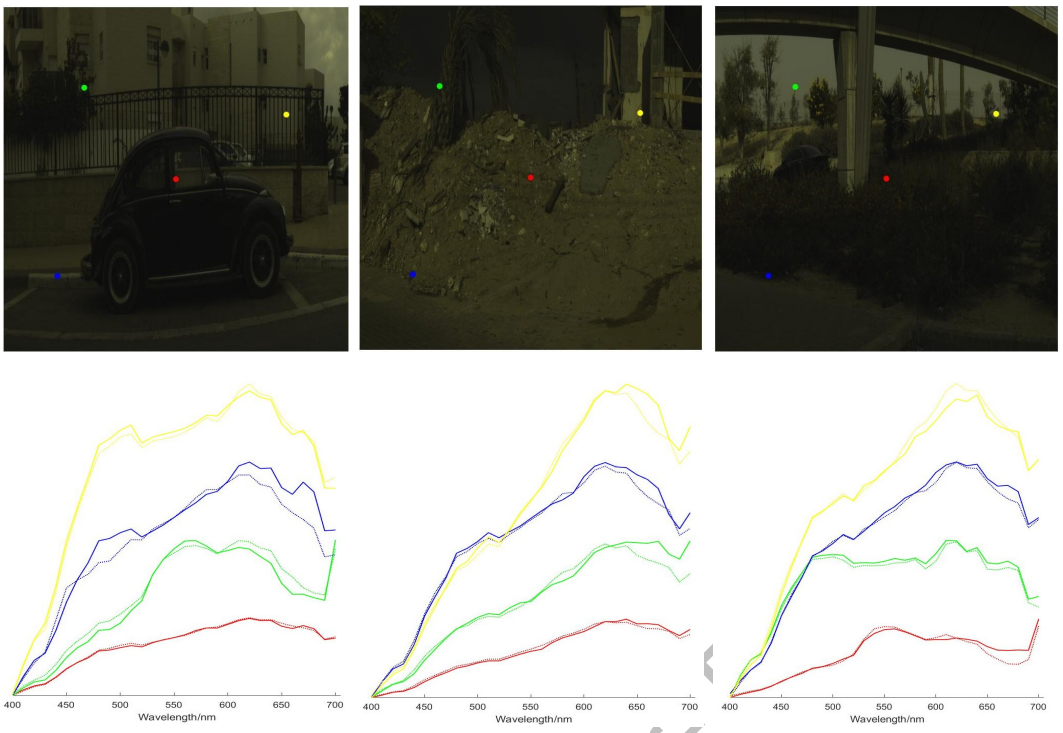


FIGURE 4.3: Sample results of spectral reconstruction by our method. Top line: RGB rendition. Bottom line: groundtruth (solid) and reconstructed (dashed) spectral response of four pixels identified by the dots in RGB images.

4.4 Sensitivity Analysis

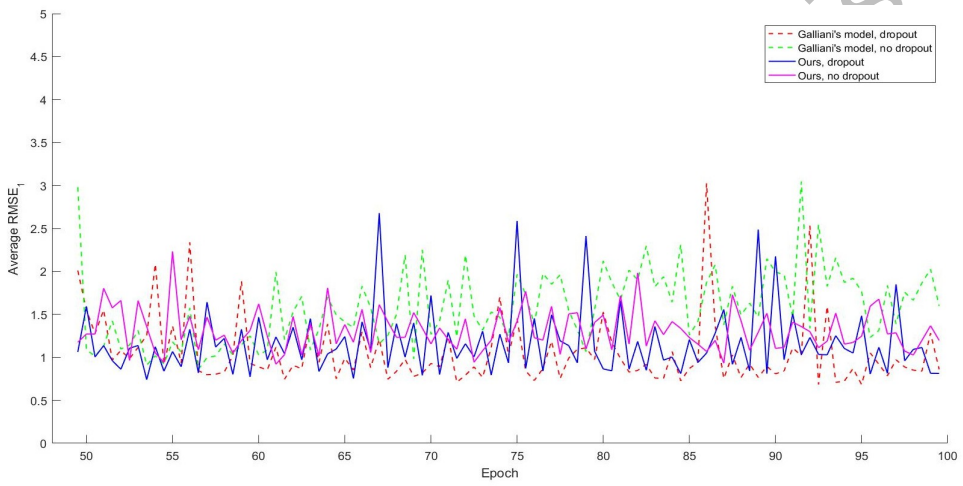
Galliani *et al.* [1] is similar to ours to a degree in that it also follows the encoder-decoder pattern, but our model is more robust and less sensitive to hyper-parameters. In order to prove this, we turn off the dropout (*i.e.* to set the dropout rate to 0) and re-train them. Table 4.3 shows the quantitative results on test data. Although the performance of both models is impaired, our model is much less affected. For Galliani *et al.* model, the pixel-level errors are increased by over 60%, with $rRMSE_1$ incremented by as much as 135.50%. On the contrary, our model is influenced by no more than 50%.

TABLE 4.3: Quantitative comparison of Galliani *et al.* and our network with/without dropout.

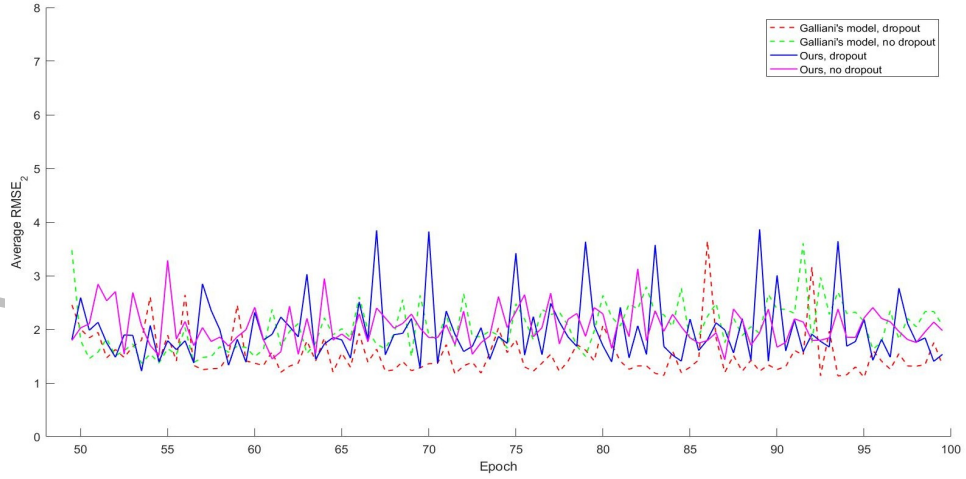
	Galliani *et al.*	Galliani *et al.* (no dropout)	Increment (%)	Ours	Ours (no dropout)	Increment (%)
$RMSE_1$	0.7584	1.6092	112.18	0.7177	1.0662	48.56
$RMSE_2$	1.2445	2.0492	64.66	1.2899	1.8168	40.85
$rRMSE_1$	0.0262	0.0617	135.50	0.0233	0.0320	37.34
$rRMSE_2$	0.0407	0.0673	65.36	0.0414	0.0593	43.24
SAM	1.5523	2.1358	37.59	1.4673	1.6206	10.45

In Figure 4.4 we plot the test curve of all evaluation metrics for Galliani *et al*. and our model. When turning off dropout, it is clear that the test curve of Galliani *et al*. (the green dash line) lies above the other three. While the curves of our model (the blue and magenta lines) lie close to each other.

Reviewing the architecture of these two networks, we find that the most significant difference is that Galliani *et al*. uses dense blocks. Dense blocks encourage a high degree of feature reuse. This helps with high-level vision tasks, where the key is to extract rich semantic information. When it comes to the super-resolution task, we hypothesize that too much feature sharing may lead to unnecessarily repeated computation, thus gives rise to unstable training.

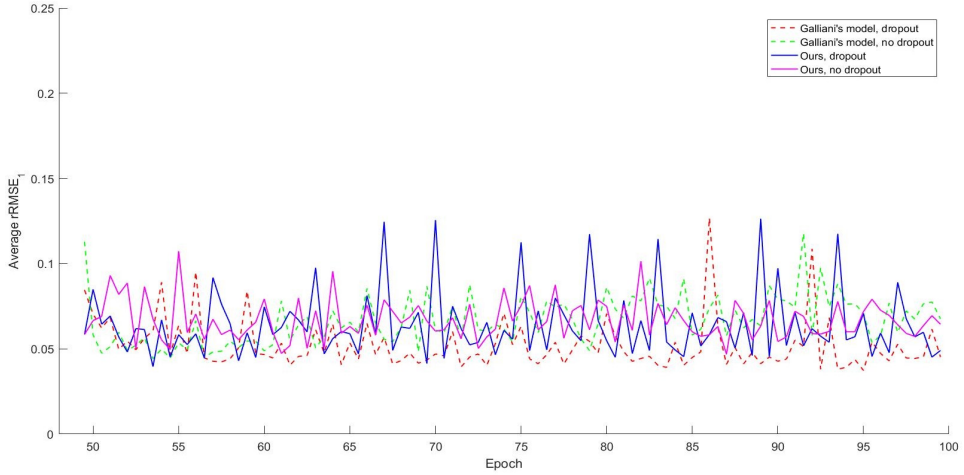


$RMSE_1$

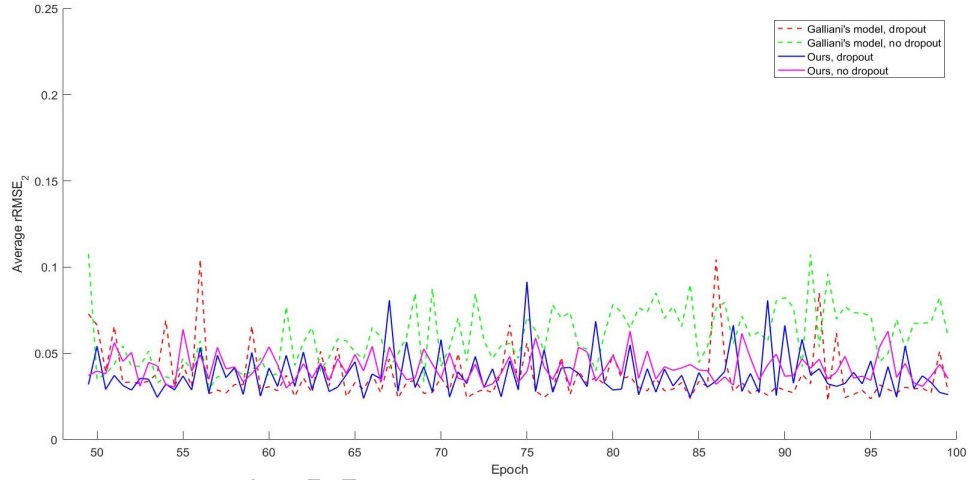


$RMSE_2$

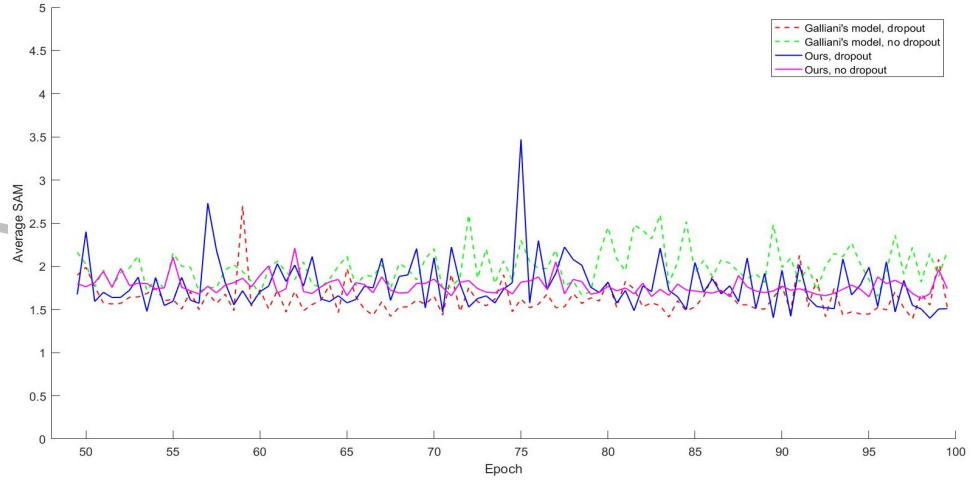
FIGURE 4.4



$rRMSE_1$



$rRMSE_2$



SAM

FIGURE 4.4: Test error for Galliani *et al.* [1] and our network with/without dropout. Only the last 50 epochs are plotted

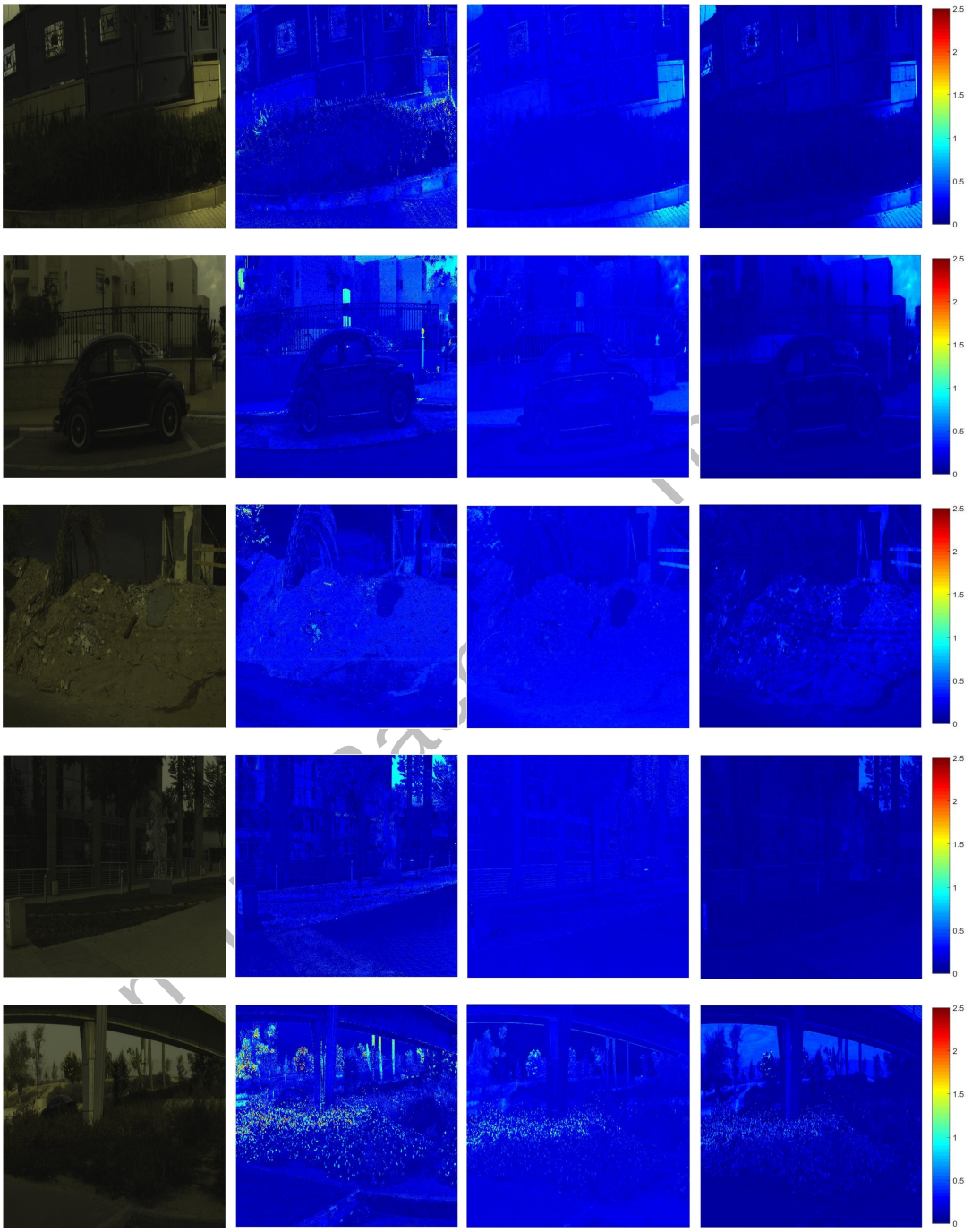


FIGURE 4.5: Visualization of absolute error. From left to right: RGB rendition, A+, Galliani *et al.*, our method

Chapter 5

Conclusion

In this thesis, we review the pros and cons of current hyperspectral imaging techniques, and aim to perform single image spectral super-resolution, *i.e.* to reconstruct the hyperspectral image using one single RGB image.

As for the proposed method, we show that leveraging both the local and non-local information of input images is essential for the accurate spectral reconstruction. Following this idea, we design a novel multi-scale convolutional neural network, which employs a symmetrically cascaded downsampling-upsampling architecture to jointly encodes the local and non-local image information for spectral reconstruction. With extensive experiments on a large hyperspectral images dataset, the proposed method clearly outperforms several state-of-the-art methods in terms of reconstruction accuracy and spectral similarity. What's more, it also guarantees stability and generalizes well.

Bibliography

[1] Silvano Galliani, Charis Lanaras, Dimitrios Marmanis, Emmanuel Baltsavias, and Konrad Schindler. Learned spectral super-resolution. *CoRR*, abs/1703.09470, 2017. URL http://arxiv.org/abs/1703.09470.

[2] Hien Van Nguyen, Amit Banerjee, and Rama Chellappa. Tracking via object reflectance using a hyperspectral video camera. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 44–51. IEEE, 2010.

[3] Yuliya Tarabalka, Jocelyn Chanussot, and Jon Atli Benediktsson. Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recognition*, 43(7):2367–2379, 2010.

[4] Zhihong Pan, Glenn Healey, Manish Prasad, and Bruce Tromberg. Face recognition in hyperspectral images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1552–1560, 2003.

[5] Seon Joo Kim, Fanbo Deng, and Michael S Brown. Visual enhancement of old documents with hyperspectral imaging. *Pattern Recognition*, 44(7):1461–1469, 2011.

[6] R Padoan, Th AG Steemers, M Klein, B Aalderink, and G De Bruin. Quantitative hyperspectral imaging of historical documents: technique and applications. *Art Proceedings*, pages 25–30, 2008.

[7] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. *IEEE Transactions on Geoscience and Remote Sensing*, 2018.

[8] Qin Zou, Lihao Ni, Tong Zhang, and Qian Wang. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, 12(11):2321–2325, 2015.

[9] Xudong Kang, Xiangping Zhang, Shutao Li, Kenli Li, Jun Li, and Jón Atli Benediktsson. Hyperspectral anomaly detection with attribute and edge-preserving filters. *IEEE Transactions on Geoscience and Remote Sensing*, 55(10):5600–5611, 2017.

[10] Chein-I Chang and Shao-Shan Chiang. Anomaly detection and classification for hyperspectral imagery. *IEEE transactions on geoscience and remote sensing*, 40(6): 1314–1325, 2002.

[11] Driss Haboudane, John R Miller, Elizabeth Pattey, Pablo J Zarco-Tejada, and Ian B Strachan. Hyperspectral vegetation indices and novel algorithms for predicting green lai of crop canopies: Modeling and validation in the context of precision agriculture. *Remote sensing of environment*, 90(3):337–352, 2004.

[12] E Keith Hege, Dan O'Connell, William Johnson, Shridhar Basty, and Eustace L Dereniak. Hyperspectral imaging for astronomy and space surviellance. In *Imaging Spectrometry IX*, volume 5159, pages 380–392. International Society for Optics and Photonics, 2004.

[13] Enrica Belluco, Monica Camuffo, Sergio Ferrari, Lorenza Modenese, Sonia Silvestri, Alessandro Marani, and Marco Marani. Mapping salt-marsh vegetation by multispectral and hyperspectral remote sensing. *Remote sensing of environment*, 105(1): 54–67, 2006.

[14] Marcus Borengasser, William S Hungate, and Russell Watkins. *Hyperspectral remote sensing: principles and applications*. CRC press, 2007.

[15] Ozgur Yilmaz, Ozgur Selimoglu, Fethi Turk, and M Sancay Kirik. Snr analysis of a spaceborne hyperspectral imager. In *Recent Advances in Space Technologies (RAST), 2013 6th International Conference on*, pages 601–606. IEEE, 2013.

[16] Valero Laparrcr and Raul Santos-Rodriguez. Spatial/spectral information trade-off in hyperspectral images. In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, pages 1124–1127. IEEE, 2015.

[17] Naoto Yokoya, Claas Grohnfeldt, and Jocelyn Chanussot. Hyperspectral and multispectral data fusion: A comparative review of the recent literature. *IEEE Geoscience and Remote Sensing Magazine*, 5(2):29–56, 2017.

[18] Laetitia Loncan, Luis B de Almeida, José M Bioucas-Dias, Xavier Briottet, Jocelyn Chanussot, Nicolas Dobigeon, Sophie Fabre, Wenzhi Liao, Giorgio A Licciardi, Miguel Simoes, et al. Hyperspectral pansharpening: A review. *IEEE Geoscience and remote sensing magazine*, 3(3):27–46, 2015.

[19] WJOSEPH CARPER, THOMASM LILLESAND, and RALPHW KIEFER. The use of intensity-hue-saturation transformations for merging spot panchromatic and multispectral image data. *Photogrammetric Engineering and remote sensing*, 56(4): 459–467, 1990.

[20] Te-Ming Tu, Shun-Chi Su, Hsuen-Chyun Shyu, and Ping S Huang. A new look at ihs-like image fusion methods. *Information fusion*, 2(3):177–186, 2001.

[21] Pats Chavez, Stuart C Sides, Jeffrey 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*, 57(3):295–303, 1991.

[22] P Kwarteng and A Chavez. Extracting spectral contrast in landsat thematic mapper image data using selective principal component analysis. *Photogramm. Eng. Remote Sens*, 55:339–348, 1989.

[23] Vittala K Shettigara. A generalized component substitution technique for spatial enhancement of multispectral images using a higher resolution data set. *Photogram. Enggineer. Remote Sen.*, 58:561–567, 1992.

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

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

[26] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11 (7):674–693, 1989.

[27] Guy P Nason and Bernard W Silverman. The stationary wavelet transform and some statistical applications. In *Wavelets and statistics*, pages 281–299. Springer, 1995.

[28] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. pages 671–679, 1987.

[29] Minh N Do and Martin Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on image processing*, 14 (12):2091–2106, 2005.

[30] Jean-Luc Starck, Jalal Fadili, and Fionn Murtagh. The undecimated wavelet decomposition and its reconstruction. *IEEE Transactions on Image Processing*, 16 (2):297–309, 2007.

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

[32] Frosti Palsson, Johannes R Sveinsson, and Magnus O Ulfarsson. A new pansharpening algorithm based on total variation. *IEEE Geoscience and Remote Sensing Letters*, 11(1):318–322, 2014.

[33] Xiyan He, Laurent Condat, José M Bioucas-Dias, Jocelyn Chanussot, and Junshi Xia. A new pansharpening method based on spatial and spectral sparsity priors. *IEEE Transactions on Image Processing*, 23(9):4160–4174, 2014.

[34] Claas Grohnfeldt, Xiao Xiang Zhu, and Richard Bamler. Jointly sparse fusion of hyperspectral and multispectral imagery. In *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International*, pages 4090–4093. IEEE, 2013.

[35] Claas Grohnfeldt, Xiao Xiang Zhu, and Richard Bamler. The j-sparsefi-hm hyperspectral resolution enhancement method—now fully automated. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2014 6th Workshop on*, pages 1–4. IEEE, 2014.

[36] Claas Grohnfeldt, Xiao Xiang Zhu, and Richard Bamler. Splitting the hyperspectral-multispectral image fusion problem into weighted pan-sharpening problems-the spectral grouping concept. In *Proceedings of 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing-WHISPERS 2015*, pages 1–4. IEEE Xplore, 2015.

[37] Massimo Selva, Bruno Aiazzi, Francesco Butera, Leandro Chiarantini, and Stefano Baronti. Hyper-sharpening: A first approach on sim-ga data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):3008–3024, 2015.

[38] Shaohui Mei, Xin Yuan, Jingyu Ji, Shuai Wan, Junhui Hou, and Qian Du. Hyperspectral image super-resolution via convolutional neural network. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 4297–4301. IEEE, 2017.

[39] Shaohui Mei, Xin Yuan, Jingyu Ji, Yifan Zhang, Shuai Wan, and Qian Du. Hyperspectral image spatial super-resolution via 3d full convolutional neural network. *Remote Sensing*, 9(11):1139, 2017.

[40] Yunsong Li, Jing Hu, Xi Zhao, Weiying Xie, and JiaoJiao Li. Hyperspectral image super-resolution using deep convolutional neural network. *Neurocomputing*, 266: 29–41, 2017.

[41] Jing Hu, Yunsong Li, and Weiying Xie. Hyperspectral image super-resolution by spectral difference learning and spatial error correction. *IEEE Geoscience and Remote Sensing Letters*, 14(10):1825–1829, 2017.

[42] Chen Wang, Yun Liu, Xiao Bai, Wenzhong Tang, Peng Lei, and Jun Zhou. Deep residual convolutional neural network for hyperspectral image super-resolution. In *International Conference on Image and Graphics*, pages 370–380. Springer, 2017.

[43] Cui Chi, Hyunjin Yoo, and Moshe Ben-Ezra. Multi-spectral imaging by optimized wide band illumination. *International Journal of Computer Vision*, 86(2-3):140, 2010.

[44] Shuai Han, Imari Sato, Takahiro Okabe, and Yoichi Sato. Fast spectral reflectance recovery using dlp projector. In *Asian Conference on Computer Vision*, pages 323–335. Springer, 2010.

[45] Jong-Il Park, Moon-Hyun Lee, Michael D Grossberg, and Shree K Nayar. Multispectral imaging using multiplexed illumination. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[46] Seoung Wug Oh, Michael S Brown, Marc Pollefeys, and Seon Joo Kim. Do it yourself hyperspectral imaging with everyday digital cameras. In *CVPR*, pages 2461–2469, 2016.

[47] Boaz Arad and Ohad Ben-Shahar. Sparse recovery of hyperspectral signal from natural rgb images. In *European Conference on Computer Vision*, pages 19–34. Springer, 2016.

[48] Michal Aharon, Michael Elad, and Alfred Bruckstein. *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[49] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[50] Jonas Aeschbacher, Jiqing Wu, Radu Timofte, D CVL, and ETH ITET. In defense of shallow learned spectral reconstruction from rgb images. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 471–479, 2017.

[51] Radu Timofte, Vincent De, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1920–1927. IEEE, 2013.

[52] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126. Springer, 2014.

[53] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 1865–1873. IEEE, 2016.

[54] Rang MH Nguyen, Dilip K Prasad, and Michael S Brown. Training-based spectral reconstruction from a single rgb image. In *European Conference on Computer Vision*, pages 186–201. Springer, 2014.

[55] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[56] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[57] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1175–1183. IEEE, 2017.

[58] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE transactions on image processing*, 19(9):2241–2253, 2010.

[59] Ayan Chakrabarti and Todd Zickler. Statistics of real-world hyperspectral images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 193–200. IEEE, 2011.

[60] NTIRE 2018 challenge on spectral reconstruction from rgb images. URL http://www.vision.ee.ethz.ch/ntire18/.

[61] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[62] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[63] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[64] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[65] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[68] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[69] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. URL http://arxiv.org/abs/1207.0580.

[70] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

[71] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[72] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[73] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1874–1883, 2016.

[74] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[75] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.