Deep Feature Learning for Spectral-Spatial Classification of Hyperspectral Remote Sensing Images

Author

Alam, Fahim Irfan

Published

2019-07-08

Thesis Type

Thesis (PhD Doctorate)

School

School of Info & Comm Tech

DOI

https://doi.org/10.25904/1912/1943

Deep Feature Learning for Spectral-Spatial Classification of Hyperspectral Remote Sensing Images

Fahim Irfan Alam

B.Sc. (University of Chittagong, Bangladesh)

M.Sc. (St. Francis Xavier University, Canada)

School of Information & Communication Technology

Griffith University

Australia

Submitted in fulfillment of the requirements of the degree of

Doctor of Philosophy

January 2019

Abstract

The recent advances in aerial- and satellite-based hyperspectral imaging sensor technologies have led to an increased availability of Earth's images with high spatial and spectral resolution, which opened the door to a large range of important applications. Hyperspectral imaging records detailed spectrum of the received light in each spatial position in the image, in which each pixel contains a highly detailed representation of the reflectance of the materials present on the ground, and a better characterization in terms of geometrical details. Since different substances exhibit different spectral signatures, the abundance of informative content conveyed in the hyperspectral images permits an improved characterization of different land coverage. Therefore, hyperspectral imaging emerged as a well-suited technology for accurate image classification in remote sensing. In spite of that, a significantly increased complexity of the analysis introduces a series of challenges that need to be addressed on a serious note. In order to fully exploit the potential offered by these sensors, there is a need to develop accurate and effective models for spectral-spatial analysis of the recorded data.

This thesis aims at presenting novel strategies for the analysis and classification of hyperspectral remote sensing images, placing the focal point on the investigation on deep networks for the extraction and integration of spectral and spatial information. Deep learning has demonstrated cutting-edge performances in computer vision, particularly in object recognition and classification. It has also been successfully adopted in hyperspectral remote sensing domain as well. However, it is a very challenging task to fully utilize the massive potential of deep models in hyperspectral remote sensing applications since the number of training samples is limited which limits the representation capability of a deep model. Furthermore, the existing architectures of deep models need to be further investigated and modified accordingly to better complement the joint use of spectral

and spatial contents of hyperspectral images. In this thesis, we propose three different deep learning-based models to effectively represent spectral-spatial characteristics of hyperspectral data in the interest of classification of remote sensing images.

Our first proposed model focuses on integrating CRF and CNN into an end-to-end learning framework for classifying images. Our main contribution in this model is the introduction of a deep CRF in which the CRF parameters are computed using CNN and further optimized by adopting piecewise training. Furthermore, we address the problem of overfitting by employing data augmentation techniques and increased the size of the training samples for training deep networks. Our proposed 3DCNN-CRF model can be trained to fully exploit the usefulness of CRF in the context of classification by integrating it completely inside of a deep model.

Considering that the separation of constituent materials and their abundances provide detailed analysis of the data, our second algorithm investigates the potential of using unmixing results in deep models to classify images. We extend an existing region based structure preserving non-negative matrix factorization method to estimate groups of spectral bands with the goal to capture subtle spectral-spatial distribution from the image. We subsequently use these important unmixing results as input to generate superpixels, which are further represented by kernel density estimated probability distribution function. Finally, these abundance information-guided superpixels are directly supplied into a deep model in which the inference is implicitly formulated as a recurrent neural network to perform the eventual classification.

Finally, we perform a detailed investigation on the possibilities of adopting generative adversarial models into hyperspectral image classification. We present a GAN-based spectral-spatial method that primarily focuses on significantly improving the multiclass classification ability of the discriminator of GAN models. In this context, we propose to adopt the triplet constraint property and extend it to build a useful feature embedding for remote sensing images for use in classification. Furthermore, our proposed Triplet-3D-GAN model also includes feedback from discriminator's intermediate features to improve the quality of the generator's sample generation process.

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

____

Fahim Irfan Alam

Acknowledgements

Completion of this thesis would never have been possible without the guidance of my supervisors, constant support from my friends, and unconditional love from my family. Too often we underestimate the power of kindness, an honest compliment or the smallest act of caring, all of which have the potential to turn a life around. Therefore, this page is entirely dedicated to the ones whose support greatly contributed in completing this tedious yet beautiful journey of my life.

First, I would like to express my sincere gratitude to my principal supervisor Dr. Jun Zhou for his excellent guidance, scholarly inputs and consistent encouragement I received throughout the research work. A mentor with such an amicable and positive disposition, Dr. Jun has always made himself available to clarify my confusions and I consider it as a great opportunity to do my doctoral programme under his tremendous direction. Thank you Dr. Jun for your continuous assistance and insightful comments in writing several research papers.

I would like to express my deepest appreciation to my associate supervisor Dr. Alan Wee-Chung Liew for his brilliant suggestions in improving my work during the submissions of every manuscript. I am also indebted to Prof. Yongsheng Gao, Prof. Jocelyn Chanussot and Dr. Xiuping Jia for their extensive comments and suggestions regarding my manuscripts. I would like to extend my appreciation to Prof. Kewen Wang and Prof. Abdul Sattar for their encouragements.

My thanks and appreciations also go to all of my colleagues from the Computer Vision and Image Processing Lab, including Mr. Ali Zia, Mr. Jing Wang, Ms. Yanyang Gu, Ms. Fereshteh Nayyeri, Mr. He Chen, Mr. Yue Li, Ms. Lena Zhang, Mr. Gilbert Eaton, Mr. Muhammad Daud Abdullah Asif, Dr. Litao Yu, Dr. Jie Liang, Dr. Lei Tong and Dr. Alex Yu. I am immensely grateful to Ms. Suhad Lateef Al-Khafaji for giving me amazing moral support and helping me in so many occasions.

I would like to express my sincere appreciation to the administrative staffs of the school of information and communication technology and higher degree research of the Griffith University for their assistance. I am particularly grateful to Ms. Mel Gilbert for patiently listening to all my inquiries and responding immediately.

I am also grateful my friends Dr. Md Masbaul Alam Polash, Ms. Sabrina Islam, Mr. Emon Kumar Dey, and Ms. Joy Galaige. I would like to express my heartfelt gratitude to my best friend Ms. Farah Jahan for cheering me up, listening to my frustrations and supporting me through the good times and bad. I am also thankful to Ms. Nazia Majadi for her amazing friendship. I am grateful to Dr. Mohammad Khairul Islam for his kindness and genuine selfless acts in many occasions.

Furthermore, my greatest appreciation goes to my parents, especially to my father for trusting me and giving me wings to chase after my dreams. I can never appreciate you enough for all the sacrifices you went through for me. I love you and I will always be on the path that you taught me. A big thank of mine goes to my dearest siblings for loving me unconditionally and fulfilling all the responsibilities to take care of the family just so I can concentrate on my studies.

Finally, thanks to Almighty Allah for keeping me safe and healthy. Life has been difficult yet unbelievably beautiful because Almighty never left my side and continued being extremely gracious to me.

A few words for my readers: "Whatever the mind can dream and conceive, it can achieve". Don't be frightened of making mistakes and downgrade your dreams to just fit the reality. Go after them with utmost conviction, place your disappointments under your feet and use them as stepping stone to rise above them. When success happens, remain humble and stretch your hands for help to someone else who may be desperately needing them.

Contents

List of Figures

List of Tables

Abbreviations

AOTF	**A**cousto **O**ptical **T**unable **F**ilter
ANC	**A**bundance **N**onegtive **C**onstraint
ASC	**A**bundance **S**um to one **C**onstraint
AVIRIS	**A**irborne **V**isible **I**nfra**R**ed **I**maging **S**pectrometer
CD	**C**ontrastive **D**ivergence
CRF	**C**onditional **R**andom **F**ield
CNN	**C**onvolutional **N**eural **N**etwork
DBN	**D**eep **B**elief **N**etwork
DL	**D**eep **L**earning
DWT	**D**iscrete **W**avelet **T**ransform
EEP	**E**xtinction **M**orphological **P**rofile
EMP	**E**xtended **M**orphological **P**rofile
FCLS	**F**ully **C**onstrained **L**east **S**quares
GAN	**G**enerative **A**dversarial **N**etwork
GCK	**G**eneralized **C**omposite **K**ernel
HSI	**H**yper**S**pectral **I**mage
HYDICE	**HY**perspectral **D**igital **I**magery **C**ollection **E**xperiment
ICA	**I**ndependent **C**omponent **A**nalysis
KD	**K**ernel **D**ensity
LBP	**L**ocal **B**inary **P**attern
LDA	**L**inear **D**iscriminant **A**nalysis
LDP	**L**ocal **D**erivative **P**attern
LMM	**L**inear **M**ixing **M**odel
LR	**L**ogistic **R**egression
LSTM	**L**ong **T**erm **S**hort **M**emory

PCA	**P**rinciple **C**omponent **A**nalysis
PDF	**P**robability **D**ensity **F**unction
MLL	**M**ulti-**L**evel **L**ogistic
MLR	**M**ultinomial **L**ogistic **R**egression
MPM	**M**aximum a **P**osteriori **M**arginal
MRF	**M**arkov **R**andom **F**ield
MTMF	**M**ixture- **T**uned **M**atched **F**iltering
NMF	**N**onnegative **M**atrix **F**actorization
NMM	**N**onlinear **M**ixing **M**odel
RBM	**R**estricted **B**oltzmann **M**achine
ReLU	**R**ectified **L**inear **U**nit
RF	**R**andom **F**orest
RoF	**Ro**tation **F**orest
RGB	**R**ed **G**reen and **B**lue
RNN	**R**ecurrent **N**eural **N**etwork
SAE	**S**tacked **A**uto **E**ncoder
SLIC	**S**imple **L**inear **I**terative **C**lustering
SeLU	**S**caled **E**xponential **L**inear **U**nit
SGD	**S**tochastic **G**radient **D**escent
SVM	**S**upport **V**ector **M**achine
USGS	**U**nited **S**tates **G**eological **S**urvey
VCA	**V**ertex **C**omponent **A**nalysis
WHED	**W**ATERS**HED**
WGAN	**W**asserstein **G**enerative **A**dversarial **N**etwork
WGAN-GP	**W**asserstein **G**enerative **A**dversarial **N**etwork- **G**radient **P**enalty

Symbols

$\mathbb{R}$	Set of real numbers
B	Number of spectral bands
$\mathbb{B}$	Set of spectral bands
N	Number of pixels of hyperspectral image
$\mathbf{H} \in \mathbb{R}^{B \times N}$	Hyperspectral image
$\mathbf{b}_i \in \mathbb{R}^{B}$	Spectral signature at i^{th} pixel
K	Number of endmembers
$\mathbf{M} \in \mathbb{R}^{B \times K}$	Endmember matrix
$\mathbb{M}$	Set of voxels or nodes in the CRF graph
$\mathbf{m}_i \in \mathbb{R}^{H}$	Spectral signature of i^{th} endmember
$\mathbf{A} \in \mathbb{R}^{K}$	Abundance matrix
$\mathbf{a}_i \in \mathbb{R}^{K}$	Vector of abundance of i^{th} pixel
$\mathbf{E} \in \mathbb{R}^{B \times K}$	Additive noise in the linear mixing model
$\mathbb{E}$	Set of edges between the nodes in the CRF graph
Q	Number of homogeneous regions
J	Number of spectral band groups
Y	Number of classes
$\mathbb{Y}$	Set of classes
ϕ	Unary potential function
ψ	Pairwise potential function
φ	Label compatibility function
μ	Mean of superpixel's spectral features
ϵ	Light intensity parameter
ε	CNN learning rate
γ	SVM parameter - kernel width

Chapter 1

Introduction

In this chapter, we present the motivations of our work, followed by an overview of our contributions. We conclude this chapter with an outline of the structure of the thesis.

1.1 Motivations

Human vision has the ability to perceive information from three types of color photoreceptors which are sensitive to three different spectra in the visible light, corresponding to red, blue and green. The cross-referencing of these three colors has enabled us to distinguish a number of color signals and eventually satisfy the requirement to fulfill the basic needs of daily lives. However, human vision is still restricted to visible light and a limited spectral resolution.

Hyperspectral imaging systems use sensors that typically operate in the range of visible through to the infrared wavelengths and can simultaneously capture information from hundreds of narrow spectral channels from the surface of the earth covering an area. Each pixel in a hyperspectral image (HSI) is represented by a vector in which each element is a measurement corresponding to a specific wavelength. The elaborated spectral information increases the possibility of accurately discriminating materials of interest in a scene with an increased classification accuracy. Also, thanks to the advances in hyperspectral technology, finer spatial resolution of the new sensors immensely contributes to the analysis of small spatial structures in images.

In the past several decades, HSI systems have gained considerable attention from the researchers in a wide array of applications. The high discriminative ability of HSI to identify and distinguish different materials on the surface has contributed the field of remote sensing in analysing several applications on the Earth's surface such as earth monitoring, land cover classification, agriculture, and mining [3–6]. Supporting a wide range of research in computer vision and pattern recognition community, hyperspectral imaging also covers broad topics related to predicting the categories of targets such as object/scene classification [7], saliency detection [1], and image reconstruction [8] etc. Also, any material that depends on chemical gradients for functionality can also be compliant to study by techniques that combine chemical and spatial properties. As a result, HSI is increasingly being used in (i) applied chemistry for detecting toxic chemical substances [9], (ii) medicine for disease diagnosis [10, 11] and image-guided surgeries [12], (iii) pharmacy for monitoring powder potency inside a feed frame for tablet manufacturing [13] and (iv) biology as fluorescence correlation tool for studying concentrations of molecules in living cells [14].

The earth observation domain entails several interesting research issues, ranging from hardware technology of the sensors to a higher level of data analysis for the image understanding. The remote sensing image classification, which is the process of identifying meaningful objects of interests with common properties, has emerged as one of the major challenges in the community. This process outputs a thematic map, in which every pixel is represented by a given label, describing the objects within appropriate classes. The rich spectral and spatial information in the HSI provides a strong foundation for achieving high accuracy in the identification of different materials of interests. However, it introduces a number of challenges that need to be addressed in order to reduce the impact of misclassification of pixels during the classification process.

Firstly, the curse of high dimensionality of the data, typically represented by the spectral dimension, results in an expensive computation and limit the exploitation of the traditional classification approaches. Generally, the supervised classification of HSIs suffer from the effect of overfitting as the number of training samples are relatively small in compared to the high spectral dimensions. As a result, it significantly affects the generalization capability of the classifiers. Since the number of training samples are extremely limited, inclusion of additional features after a certain point leads to a decrease in both classification accuracy and generalization ability of the classifiers.

Secondly, most HSIs are captured by sensors placed on satellites, airborne, or unmanned aerial vehicles. Due to the long distances between the sensors and the targets, HSIs normally do not have high spatial resolution, resulting in mixed responses of various types of ground objects. Hence, each signal is the result of a combination of spectra of several pure materials composing the area defined by the pixel projected onto the ground. The mixed estimations of several pure materials impose serious challenges in classification problems. Interestingly, retrieving these pure spectra, usually referred to as a collection of spectral signatures, or *endmembers*, and their corresponding proportions, i.e., *abundances*, at each pixel [15] brings useful sub-pixel information. These estimations can serve as important cues for classification and if combined in a classification model, can facilitate the process of extracting more structured spectral-spatial information.

To address the above-mentioned limitations, it is thus necessary to develop new techniques to exploit the underlying spatial and spectral information in HSI. Spatial correlation statistics measure and analyze the degree of dependencies among pixels in HSI. The spatial homogeneity or heterogeneity contributes in estimating degree of spatial correlation among the pixels across the earth surface. Hence, an integration of both spectral and spatial information can maximize the exploitation of the information residing in the HSI and eventually improve the classification performance.

In the hyperspectral classification problem, a number of features are designed to extract discriminative information from HSI, which can enhance the classification performance. All these methods show that feature extraction and fusion lead to improved classification performance over processing on raw spectral features. These hand-crafted features are extensively used in many traditional approaches to exploit the spectral, textural and geometrical attributes of the HSIs. However, these features depend mostly on domain-specific information and sometimes, cannot properly address the underlying relationships in the data. As a result, it is difficult for these features to achieve an optimal balance between discriminability and robustness.

Recently, deep learning algorithms have gained significant attention from the researchers in the geoscience and remote sensing community and have achieved excellent performance compared to those of traditional learning algorithms. A deep model learns the representative and discriminative features in a hierarchical manner to model high level abstraction of data [16]. Among various deep learning models, a convolutional neural network (CNN) has been widely used for pixel-level labeling problems. With this model, a good representation of spectral-spatial features can be learned, which allows performing an end-to-end classification task [16]. Unfortunately, the possibility of reaching local minima during CNN training and the presence of noise in the input images may produce isolated regions in the classification map. Moreover, since this is very challenging to collect large number of ground truth data in remote sensing applications, CNNs suffer from the problem of "overfitting" and may not perform well on test data.

In this regard, several probabilistic graphical models such as the markov random field (MRF) and conditional random field (CRF) have been introduced in the deep models to explicitly model the contextual information between regions [17–20]. However, in most cases, the integration of such graphical models is entirely disconnected from the CNN and do not completely utilize the incredible potential of integration in an unified framework [21]. It will be interesting to explore the possibilities of combining the graphical models in a deep network that is capable of end-to-end learning.

The existing deep models generally receive input from the HSIs in terms of the whole spectral cube. However, hyperspectral features obtained from smaller sized band groups have several advantages. Firstly, smaller band groups provide better local spectral-spatial estimation of the underlying data. Secondly, different band groups capture spectral information in different ranges of wavelengths which provide more material based information to the classifier. Finally, several band groups provide ample information to the deep model which can benefit the deep network for classification in terms of both sufficient amount of training samples and useful spectral-spatial information as input.

Generative adversarial network (GAN) has recently been adopted for the classification of remote sensing images [22, 23]. Mostly, GANs have been used in order to separate real sample from fake ones. Unfortunately, little attention have been given in order to design the GAN architecture in a way to handle multi-class classification problems. In most cases, the last layer of the discriminator network is modified to include additional classifier for multi-class classification [24]. It is, therefore, not clear how GANs are contributing in the class-specific discrimination which leaves a massive space for improvement.

1.2 Contributions of the Thesis

This thesis proposes three novel methods to learn deep features for HSI classification and showcase their abilities in different scenarios. We present an integration of deep models with traditional probabilistic graphical models in an end-to-end learning framework in two different contexts: (1) explicitly formulate CRF as a deep model to integrate the advantages of both models in modeling the spatial relationships in the data and (2) providing structure-based unmixing results representing spectral-spatial estimation of the underlying endmembers as input. We also adopt GANs for remote sensing and include explicit components to enhance the multi-class classification ability of the discriminator.

The contributions of the thesis are listed as follows:

1. We first propose a method to classify HSIs by considering both spectral and spatial information via a combined framework consisting of CNNs and CRF in an unified end-to-end learning procedure. We use multiple spectral band groups to learn deep features using CNNs, and then formulate an optimized deep CRF with CNN-based unary and pairwise potential functions to effectively extract the semantic correlations between patches consisting of 3-D data cubes. Furthermore, we introduce a deep deconvolution network that improves the final classification performance. We also introduce a new data set and experiment our proposed method on it along with several widely used benchmark data sets to evaluate the effectiveness of our method.

2. Secondly, we present a CNN based classification model by providing unmixing results as input during the training of the model. We extend an existing unmixing method to estimate the individual spectral responses from different materials in different groups of wavelengths. The estimations of the materials are used as important features to generate superpixels in which we introduce kernel density (KD)-estimated probability density function (PDF) to describe the spectral distribution of the superpixels and update the cluster centers accordingly. These abundance information-guided superpixels are provided as input to train a CRF-CNN integrated deep model in which the inference is implicitly formulated as a recurrent neural network (RNN). Instead of raw data, our proposed model receives

significant spectral-spatial information in the data to produce better and powerful features so as to achieve improved classification performance.

3. Thirdly, we demonstrate the potential of GAN-based models in developing an effective spectral-spatial method for HSI classification. We present a novel GAN model, primarily focusing on improving the multi-class classification ability of the discriminator. In this context, we propose to adopt the triplet constraint property and extend it to build a powerful spectral-spatial feature embedding for remote sensing images for use in classification. To further improve the quality of the generated fake samples, our model receives feedback from discriminator's intermediate features, thus enabling it to use those samples as augmented data.

1.3 Outline of the Thesis

This thesis is organized in six chapters.

In chapter 2, we introduce the background of the thesis including the basic knowledge of HSI in the context of remote sensing technology. We then present an overview on HSI feature extraction methods and introduce the widely used classifiers in the contexts of both supervised and unsupervised classification. We further elaborate on the importance of developing spectral-spatial classification approaches and review the related state-of-the-art methods. In the end, we discuss the existing deep network architectures and present a detailed overview on the recent deep model-based classification methods.

In chapter 3, we present an efficient CRF-CNN based deep learning algorithm for classifying HSI data. We first present our proposed 3D-CNN that we apply in a range of more effective spectral-spatial representative band groups to extract initial features. We then propose an optimized deep CRF model and present a detailed parameter calculation and inference procedure, along with the prediction refinement stage. We also present a data augmentation algorithm to increase the size of training samples for training the CNNs followed by presenting elaborated experiments.

In chapter 4, we propose an integrated method which combines unmixing results into a deep model in order to classify hyperspectral data. We first present an unmixing algorithm that we extend to estimate different materials from different group of wavelengths. Next, we present abundance information-guided superpixel extraction algorithm in which we further introduce KD-estimated PDF to describe the suuperpixels. We later present our proposed deep model which is formulated as a recurrent neural network after receiving unmixing results as important cues for classification.

In chapter 5, we present a GAN-based model for HSI classification. After discussing the background on GAN models, we introduce our proposed formulation of triplet constraint construction and the selection process of the triplets. Then we present the network architectures, consisting of the generator and the discriminator. Finally we present detailed experimental results and performed several stages of evaluation to support the potential of our proposed model.

In chapter 6, we summarize the thesis and discuss future work.

Chapter 2

Literature Review

This chapter provides a basic overview of the field of remote sensing, focusing on the HSIs, the data acquisition technologies and the challenges related to their analysis in solving complex tasks such as classification. A comprehensive review on hyperspectral feature extraction, followed by discussion on the recent spectral-spatial methods specifically designed to classify remotely sensed data are also presented. Furthermore, we provide an extensive review on the existing deep network architectures and their corresponding state-of-the-art techniques that are able to learn expressive, high-level contextual features for classification task. Finally, we introduce the datasets used for experimental purposes on HSI classification in this thesis.

2.1 Overview on Remote Sensing

Remote sensing is the process of data acquisition on the environment, geology, atmosphere, and different attributes of the earth by positioning satellite- or aircraft-based sensors. Targeting an object or a scene covering an area under investigation, the sensors collect and transmit data from different parts of the electromagnetic spectrum perceiving a portion of the electromagnetic radiation reflected from the earth's surface in a range of wavelengths.

The main distinction of remote sensing systems is based on the types of the source of energy, considering that the electromagnetic radiation is the primary carrier of information. Remote sensing systems, which rely on naturally occurring energy provided by the sun; either reflected or absorbed and then re-emitted from the earth's surface, are called passive sensors [25]. While this visible radiation is only available when the sun is illuminating the earth, the emitted energy, i.e., the far infrared can be perceived anytime of the day, as long as the amount of energy is large enough to be recorded. Most passive systems used in remote sensing applications operate in the visible, near infrared, medium infrared, far infrared, and microwave portions of the electromagnetic spectrum. Active sensors [25], on the other hand, rely on their own sources of radiation to illuminate objects so that the energy reflected and returned to the sensor may be measured. Those operate in the microwave and radio wavelength regions of the electromagnetic spectrum. Examples of largely used active sensors are the RAdio Detection And Ranging (RADAR) [26] and Light Detection And Ranging (LiDAR) [27].

The basic foundation of the existing sensors is to obtain information about the reflected radiation along the pathway as the satellite- or aircraft-based sensors orbit the Earth. Those information obtained by the sensors can be described in terms of radiometric (spectral), geometric (spatial) and temporal resolution. Radiometric resolution is the number of bands that a sensor captures spectral information, whereas spatial resolution refers to the smallest amount of area on the Earth's surface for which a sensor can record spectral information. The frequency with which a sensor revisits the same part of the Earth's surface is measured by the temporal resolution. Due to the advent of sensor technologies, hyperspectral imaging has become an emerging technology in remote sensing for increasing knowledge and understanding of the Earth's surface.

2.2 Introduction to Hyperspectral Images

Before introducing HSIs, we now present a brief review on the traditional images such as gray scale and color images. Each pixel in a gray scale image represents the intensity of the light received by the sensors over a range of wavelengths. In case of red, green and blue (RGB) images, each pixel is represented by three intensity maps, each of which correspond to the intensity of particular regions of the visible spectrum of three colors, shown in Fig. 2.1.

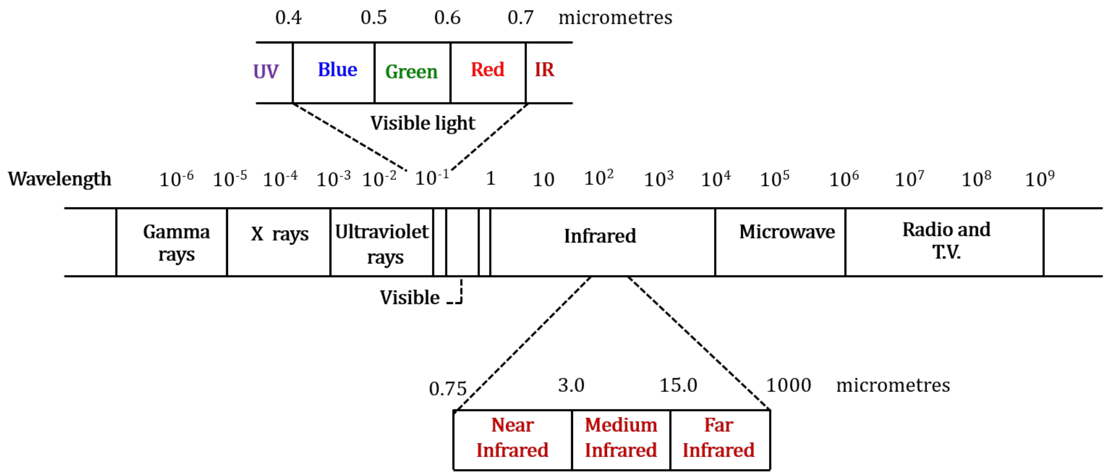


FIGURE 2.1: The electromagnetic spectrum.

Compared to gray scale and RGB images, HSIs contain tens or hundreds of narrow spectral bands, individually containing the light intensity for that wavelength. The spectral range is wider than traditional images covering wavelengths of approximately 380nm to 1100nm (as shown in Fig. 2.1). Due to its wider spectral range, HSIs are able to provide rich information on the spectral and spatial distributions of the objects [28].

Hyperspectral imaging systems started contributing immensely in remote sensing after the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [29] was developed in the 1980's at the Jet Propulsion Laboratory (JPL) for Earth remote sensing. Several other examples of hyperspectral airborne imaging systems are Reflective Optics System Imaging Spectrometer (ROSIS) [30], Hyperspectral Digital Imagery Collection Experiment (HYDICE) [31], Airborne Real-time Cueing Hyperspectral Enhanced Reconnaissance (ARCHER) [32] etc. Examples of sensors operating in space are Hyperion (USA, 2000) [33], Advanced Responsive Tactically Effective Military Imaging Spectrometer (ARTEMIS, USA, 2009) [34]. The hyperspectral sensors typically cover a range of 0.4 to 2.5 μm using 115 to 512 spectral channels, with a spatial resolution varying from 0.75 to 20 m/pixel for airborne sensors and from 5 to 506 m/pixel for satellite sensors [28].

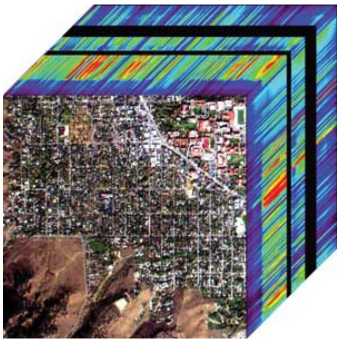


Figure 2.2: Hyperspectral image cube.

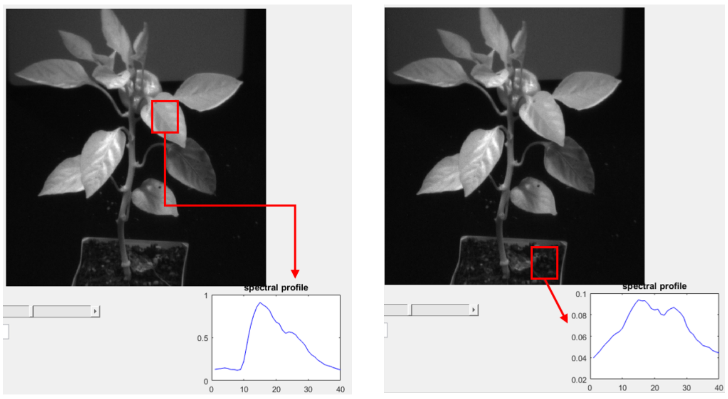


Figure 2.3: Spectral responses from different materials in a HSI

As illustrated in the Fig. 2.2, every pixel of the image can be represented as a high-dimensional vector with the spectral information added as a third dimension of values to the two-dimensional spatial image, generating a three-dimensional data cube, sometimes referred to as an image cube [35]. Different materials exhibit different spectral information as illustrated in Fig. 2.3. It shows an example containing different materials, such as plant, soil, brick etc. We can see the differences between the spectral responses from two different materials: leaves and soil. Due to the wider range of spectral responses, hyperspectral imaging has been acknowledged as an emerging and well-suited technology for various remote sensing applications, such as:

- Agriculture: Hyperspectral imaging has become a popular research tool for monitoring quality parameters and improving grading and classification of major elements of agriculture materials [36].

- Mineralogy: The application of hyperspectral imaging for the automatic identification of minerals from satellite and airborne images, and the relative presence of valuable minerals, has been the subject of interesting researches in recent decades [37].
- Monitoring environment: With the advent of hyperspectral imaging technologies, it has become easier to monitor the environmental changes on the earth's surface caused by natural calamities and human activities [38].
- Surveillance: Recent advances in hyperspectral imaging acquisition and processing have contributed significantly in developing models for detecting hidden objects in the domain of military surveillance and providing useful security services [39].

2.3 Hyperspectral Data Acquisition

Hyperspectral sensors capture a scene as a collection of images, each of which represents a narrow wavelength range of the electromagnetic spectrum. Considering a three-dimensional space (x, y, λ), where x and y are spatial coordinates and λ is the spectral coordinate, each pixel in those images is the integral of the radiance in a cube. The minimum value obtained by the integral represents the radiometric resolution and the spatial resolution is represented by the size of a cube in the plane (x, y). The spectral resolution is the minimum bandwidth on which the measured radiation is integrated. For remote sensing, image spectrometer devices for data acquisition can be categorized from two aspects: (i) underlying architecture of the devices and (ii) filtering techniques.

2.3.1 Architecture of the Devices

The architecture of the devices may depend on the acquisition approaches of spatial information by using techniques including whisk broom, push broom, and framing or 'staring arrays'. In whisk broom scanners [40, 41], the spectral image with zero spatial dimension is processed by dispersing the spectrum on a linear detector array to obtain information from one pixel at a time in both the along-track and cross-track directions perpendicular to the flight path. In push broom scanners [40, 41], a two-dimensional detector array is used which is arranged perpendicular to the flight direction of the aircraft. The detector advances with the carrier's motion, collecting image one line at a time, with all the pixels in a line being simultaneously obtained. The dispersion occurs across the slit of the spectrometer, producing the spectral dimension and one spatial dimension which has to be scanned to complete the data cube. Although limited to the varying sensitivity of the individual detectors, a push broom scanner can capture stronger signal in compared to the whisk broom scanners.

A framing or 'staring array' [42] acquisition device consists of an array of light-sensing pixels at the focal plane of a lens and uses a two-dimensional field of view (FOV), which is kept stationary on the object. The focal plane arrays operate by detecting photons at particular wavelengths and then emitting an electrical charge which is then digitized to construct an image of the object.

2.3.2 Filtering Techniques

During acquision of hyperspectral images, spectral scanning acquires single images for each different wavelength sequentially, while the object is kept stationary under the camera. The spectral filtering can be provided by (i) a number of discrete filters in a filter wheel [43] or (ii) tunable filters [44]. The advantage of spectral scanning is that they are able to choose spectral bands and have a direct representation of the spatial dimensions of the scene. A liquid crystal tunable filter (LCTF) [45] with large aperture uses electronically controlled liquid crystal elements to transmit a selectable wavelength in the visible and near infrared (VNIR) band and exclude others. A significant characteristic of this spectrometer is that it has a strong ability of becoming accustomed to environmental changes. Acousto optical tunable filter (AOTF) [46] camera is another example of a spectral scanning camera that is used to dynamically select a specific wavelength from a broadband or multi-line laser source. As the applied radio frequencies are varied, the transmitted wavelength changes, tuning the wavelength of the beam image in tens of microseconds or less. In Fig. 2.4, we show a Brimrose hyperspectral AOTF camera in the spectral laboratory at the Griffith University.

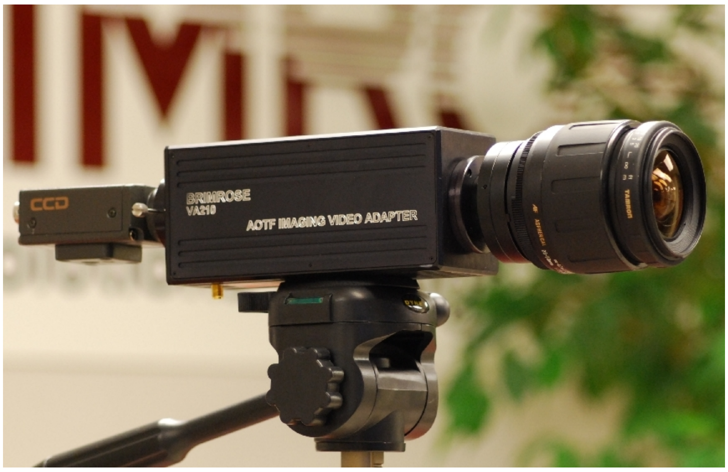


FIGURE 2.4: AOTF hyperspectral imaging system [1].

A Fabry–Pérot interferometer (FPI) [47] consists of a transparent plate with two parallel reflecting surfaces which is used to prevent the rear surfaces from producing interference fringes. Some optical spectrum analysers use this interferometer to determine the wavelength of light with great precision by using different free spectral ranges.

2.4 Mixing Model

Remote sensing images generally suffer from lower resolution due to the long distance between the sensor and the targets. As a result, the responses of the materials are often mixed together. Linear mixing model (LMM) and nonlinear mixing model (NMM) are two main types of mixing models. We now present a brief explanation of each of those:

2.4.1 Linear Mixing Model

LMM is a widely adopted mixing model assuming that there is no interference between the spectral signatures of the materials before the lights are received by the sensor [15, 48]. Illustrated in Fig. 2.5, assuming $\mathbf{h}$ is a pixel in an HSI data, we can see that pixel $\mathbf{h}$ is mixed by $\mathbf{m}_1$, $\mathbf{m}_2$ and $\mathbf{m}_3$ materials with the proportions of α_1, α_2 and α_3.

The spectrum is not altered by a single material as it has no interaction with. Specifically, each pixel is considered to be composed of several materials and hence, the spectral

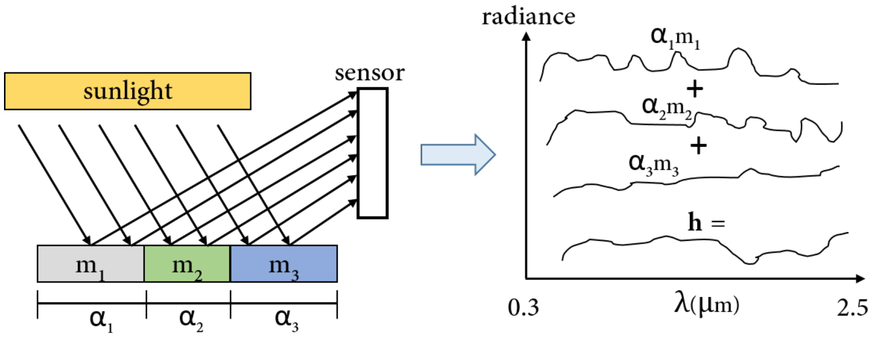


FIGURE 2.5: Linear mixing [2].

signature is represented by a linear mixture of spectral signatures of the materials, which we call endmembers and its corresponding fractional abundances [15, 49].

Let B be the number of bands in an image and K be the number of endmembers. A pixel $\mathbf{h}$ in a hyperspectral image, is represented by a $B \times 1$ column vector containing the data in different bands. Let $\mathbf{M}$ be a $B \times K$ matrix $(\mathbf{m}_1, \ldots, \mathbf{m}_j, \ldots, \mathbf{m}_K)$, where $\mathbf{m}_j$ is a $B \times 1$ column vector representing the spectral signature of the j^{th} endmember. Therefore, $\mathbf{h}$ can be approximated by a linear combination of endmembers

$$\mathbf{h} = \mathbf{M}\mathbf{a} + \mathbf{e} \tag{2.1}$$

where $\mathbf{a}$ is a $K \times 1$ column vector for endmember abundances, and $\mathbf{e}$ is the additive Gaussian white noise. A standard assumption related to the LMM defined here is that the noise follows a Gaussian distribution with a zero-mean and co-variance matrix [50]. It is important to mention that the statistical model makes an assumption that the noise variances are the same in all bands which is extensively used in the literature [51, 52]. The additive noise is generally used to evaluate the effectiveness of the model in extracting endmembers in an environment corrupted by Gaussian noise, in which case there are no pure signatures [51].

Upon extending the model from pixel-level to the whole image, the linear model for the N number of pixels in the image $\mathbf{H}$ becomes:

$$\mathbf{H} = \mathbf{M}\mathbf{A} + \mathbf{E} \tag{2.2}$$

where matrices $\mathbf{H} \in \mathbb{R}_{+}^{B\times N}$, $\mathbf{A} \in \mathbb{R}_{+}^{K\times N}$, and $\mathbf{E} \in \mathbb{R}^{B\times K}$ represent the HSI, the abundance matrix, and the additive noise respectively.

Abundance nonnegative constraint (ANC) and abundance sum to one constraint (ASC) are two constraints imposed on the abundance matrix in order to represent a physically realizable scene. ANC is defined as:

$$a_j \geq 0, j = 1, .., N \tag{2.3}$$

where a_j is the abundance vector of the j^{th} pixel. This constraint indicates that none of the spectral signatures constituting the pixel should be negative. Similarly, ASC can be defined as:

$$\sum_{i=1}^{K} a_j^i = 1 \tag{2.4}$$

which means that the individual abundances a_j for all the K endmembers within the j^{th} pixel should sum to 1.

2.4.2 Nonlinear Mixing Model

In the nonlinear mixing model (NMM), the incident light is scattered multiple times from multiple materials before reaching the sensor. As a result, the produced spectral signature is a non-linear mixture of the material signatures. Non-linear interactions may occur in (1) microscopic scale and (2) macroscopic scale. The microscopic scale or intimate mixture occurs when two materials are homogeneously mixed [53]. In this case, the mixture is produced by photons emitted from one material being absorbed by other materials, which may emit more photons. Macroscopic level or multilayer configuration occurs when the light scattered by a given material reflects off other materials before reaching the sensor. This phenomenon can be observed in forest areas where there may be many interactions between the ground and the canopy.

2.5 Hyperspectral Image Feature Exraction

A critical pre-processing step required to design an effective classifier for HSIs is to identify and characterize the features representing the data [54]. The performance of a classifier largely depends on the quality of the input features and how sensitive are those features to the target classes. However, the issue of additional computational load occurs due to the high dimensional data if we choose to use the complete set of spectral bands, where each band represents a dimension of the input data. Also, it is a common problem with HSI data to observe a decreasing generalization performance of the model if the ratio of the number of available training samples to the number of features is low. As a result, the classification performance is high on the training samples but is significantly low on the testing samples, causing a phenomenon called the Hughes effect or the curse of dimensionality.

To address this issue, the high dimensionality of data can be reduced by selecting a subset of the original features, needed to effectively describe the properties of the HSI data. Kuo and Landgrebe [55] proposed a data processing chain, in which the feature extraction stage is significantly improved after the feature selection stage selects the best subset of features from the entire spectral channels based on a selection criterion. A number of techniques integrating the feature selection and feature extraction steps have been proposed. We have categorized the techniques into (i) knowledge-based (ii) supervised and (iii) unsupervised approaches. We will now present these feature extraction methods for HSI data.

2.5.1 Knowledge-based Feature Extraction

Based on the characteristics of the spectral information, references to a number of arithmetic operations have been found in the literature such as normalized differential vegetation index (NDVI), normalized differential water index (NDWI) and a modified soil adjusted vegetation index (MSAVI2). These arithmetic operations are performed on a set of relevant bands in order to enhance the signal. Originally designed for multispectral images, these features have been extended to adopt for hyperspectral data as well to exploit absorption features. In this context, the genetic programming-spectral vegetation index [56] and the cellulose absorption index [57] have been derived from hyperspectral data. For NDVI[58], since the index is calculated through a normalization procedure, the NDVI values (between 0 and 1) show a sensitive response to green vegetation. NDWI [59] is designed to represent and enhance the presence of open water features by using reflected near-infrared radiation and visible green light and eliminating the presence of soil and vegetation features. NDVI is very sensitive to certain factors such as brightness of soil background and vegetation canopies. To address this issue, MSAVI2 [60] was proposed in which a soil conditioning index was introduced by considering the effect of the solar incidence angle variation and changes in the underlying physical structure of the soil.

2.5.2 Supervised Feature Extraction

For classification task, it is very important to discriminate between the classes of interests in the image. Unfortunately in hyperspectral remote sensing, the classes are often similar and complex in terms of their characteristics, which makes the discrimination task difficult. Therefore, automatic feature selection becomes important based on some criterion using the training data. In this context, a number of parametric supervised approaches have been proposed based on class modelling using training data.

The parametric supervised methods combine adjacent correlated original bands to produce new discriminative features, preserving the spectral information. Few distance functions have been proposed as the selection criterion in order to group the bands, such as B-dis and J-M distance functions. With B-dis function [61], a new feature can be obtained by applying a weighted sum of the bands in each group. J-M distance [62], on the other hand, takes the average of the contiguous groups of bands to produce new features. These distance functions are used to identify a subset of features that best accommodate the class data variation and produce the separation capability provided by each band, leading to a quantitative band selection [63].

Along this direction of obtaining features, linear discriminant analysis (LDA) has been proposed as parametric feature extraction technique, based on the mean vector and covariance matrix of each class. LDA projects the original high-dimensional data onto a low-dimensional space, where the classes are separated by maximizing the ratio of between-class scatter matrix to within-class scatter matrix, referred to as the Raleigh quotient [64]. LDA has been extensively used in remote sensing applications focusing classification and feature reduction. LDA, however, does not work well in situations where the number of features is higher than the number of training samples. It is observed that obtaining solutions in these cases requires the scatter matrices to be non-singular.

Non-parametric feature extraction methods have been proposed to overcome the limitations of LDA. Bandos et al. [65] proposed a regularization method to address the issue of the small ratio between the number of available training samples and the number of spectral features. In this context, Kuo and Landgrebe [55] inferred that the scatter matrices, regularization methods and eigenvalue decomposition are the essential components for solving such ill-posed problems. Fukunaga et al. [66] provided a significant finding through a non-parametric between-class scatter matrix that the contribution of each sample in extracting features is different. According to their observation, the data points close to the boundary are of more importance than those which are far from the boundary, resulting in distinct weights given to each sample.

Cosine-based non-parametric feature extraction (CNFE) [67] technique has been proposed in order to measure similarity by using cosine distance instead of the euclidean distance. A double nearest proportion (DNP) [68] feature extraction method was developed based on a double nearest proportion structure to construct new scatter matrices. The use of a DNP is particularly effective in cases when overlapping occurs during the separability between the boundaries of class distributions. Kernel functions can also be useful in increasing class separability by extending linear models to non-linear models. Some examples include generalized discriminant analysis (GDA) [69] and kernel local Fisher discriminant analysis (KLFDA) [70].

Yuntao *et al.* [71] proposed a feature extraction method based on structured sparse logistic regression and 3-D discrete wavelet transform (3D-DWT) texture features that decomposes an HSI cube at different scales, frequencies and orientations in order to capture geometrical and statistical spectral-spatial structures. Jie *et al.* [72] proposed a novel 3D high-order texture pattern descriptor, based on the local derivative pattern (LDP) for hyperspectral face recognition.

2.5.3 Unsupervised Feature Extraction

Supervised methods primarily combine groups of contiguous bands of the original HSI. However, data transformation methods can also be applied to map the original high-dimensional space to low-dimensional subspace. The principal component analysis (PCA) is such data transformation method, based on the fact that neighboring bands are highly correlated and often convey almost the same information. During the process of transformation, the optimum linear combination of the original bands accounting for the variation of pixel values in the image is identified [37]. The PCA examines band dependency by employing the statistical properties of the spectral channels.

Independent component analysis (ICA), in contrast to PCA, not only recovers independent signals from overlapping signals but also makes the signals as independent as possible by reducing higher-order statistical dependence. After applying ICA, each source is automatically extracted from the observation of linear combination of these sources [73]. ICA has also been adopted in hyperspectral unmixing [74] due to its low computation time and its ability to perform without prior information.

Although PCA and ICA have been very effective in compressing information, HSIs do not always coincide with such orthogonal projections. In this context, projection pursuit (PP) [75] techniques find a set of interesting projections such that those projections are deviated from the Gaussian distribution assumptions. During this process, the projection that maximizes a projection index based on the information divergence of the estimated probability distribution is searched. After that they reduce the dimensions by projecting the data onto the subspace orthogonal to the previous projections.

Non-linear feature extraction methods have excellent ability to better represent complex non-linear data which are well suited for HSI data. In this context, manifold learning which uses isometric mapping (ISOMAP), have widely been used for HSI data analysis since it can approximately maintain the local structure of the original space [76]. Instead of selecting random cluster centers, selecting points focusing the boundaries of the clusters has been adopted in L-ISOMAP approach [77].

Kernel functions can also be adopted in unsupervised feature extraction methods. Input data from the original data space can be mapped to a feature space where inner products in the feature space can be determined by a kernel function with an explicit non-linear mapping. In this context, Kernel-based PCA (KPCA) [78] and kernel-based ICA (KICA) [79] were introduced accordingly.

Considering both spectral and spatial features, Al-khafaji *et al.* [80] proposed a method to extract spectral and geometric transformation invariant features. The method, named spectral-spatial scale invariant feature transform (SS-SIFT), consists of keypoint detection and descriptor construction steps. Jie *et al.* [1] proposed a material-based salient object detection method in which they exploited an unmixing approach to estimate the spatial distribution of different materials, followed by the construction of a conspicuity map based on the global spatial variance of spectral responses.

2.6 Hyperspectral Image Classification

Hyperspectral imaging research focusing on image understanding has long attracted the attention because the analysis results, such as the classification, are the basis for many different applications and domains as mentioned earlier. Similarly, the earth observation domain demands abundant open research issues to overcome, particularly on the higher level data analysis algorithms for the remote sensing image understanding. As a result, remote sensing image classification emerges as one of the most important challenges which refers to the process of identifying materials of interest on the ground of the area of interest with similar properties that are grouped into “classes”.

HSIs contain information on hundreds of continuous narrow spectral wavelengths with very fine spectral resolution in each HSI pixel. The detailed spectral information provided by HSI satellite- or aircraft-based sensors from the surface of the Earth increases the possibility of accurately discriminating materials of interest with a high classification accuracy. In addition, thanks to the advances in HSI technology, the improved spatial resolution of the recently invented sensors facilitate the analysis of small spatial structures in images. For example, the pixels within a neighboring relationship in similar regions have a high possibility of belonging to the same class. By doing this, the 'labelling uncertainty' that happens when only spectral information is considered, can be addressed by taking spatial information into account. Hence, the relation between the spectral channels and the underlying spatial structure within the image can be effectively exploited, offering better potential to discriminate more detailed classes and provide broader applications for hyperspectral feature extraction [1, 72, 80], segmentation [81] and classification [82, 83].

Broadly speaking, classification techniques for remote sensing images can be categorized into two: supervised and unsupervised classifiers which are briefly described as follows:

- Supervised classifiers: These types of methods classify input data by considering the spectral information into the classification procedure, by using a set of representative samples for each class, referred to as training samples. These are usually obtained by labeling the pixels of an image based on some field measurements. For a hyperspectral data cube with B-bands and Y classes, which can be represented as a set of N pixel vectors $X = \{X_i \in \mathbb{R}^B, i = 1, 2, \ldots, N\}$, a supervised classifier classifies the original data into a set of classes $\mathbb{C} = \{y_1, y_2, \ldots, y_Y\}$

- Unsupervised classifiers: With unsupervised classifiers, a remote sensing image is divided into a number of classes without the help of training data or prior knowledge of the study area. The grouping can be done based on an arbitrary number of initial 'cluster centres', which may be user-specified or may be randomly chosen. During the classification process, each pixel is assigned with one of the cluster centers based on some similarity criterion. We now briefly discuss two widely used unsupervised classifiers:

  1. K-means: One of the most widely used clustering techniques, this approach starts with a random initial partition of the data samples into candidate

clusters and then adopts an iterative updating methods to partition the data samples into K clusters in order to minimize the within class distance.

2. Iterative Self-Organizing Data Analysis (ISODATA): The ISODATA algorithm is similar to the K-means algorithm with the distinct difference that ISODATA allows for different number of clusters while the K-means assumes that the number of clusters is known a priori.

Supervised classification techniques play a key role in the analysis of HSIs. A wide variety of classifiers have been proposed in different applications such as land-cover mapping, crop monitoring, urban development and forest application etc. Random Forest (RF), an ensemble method for classification [84], has drawn increasing interest in HSI classification [85–87]. Here, several classifiers are trained and their individual results are then combined through a voting process. Each decision tree is provided with the input vector and provides a unit vote for a particular class and the forest chooses the class that has the most votes. The advantage of this method is that it provides an unbiased estimate of the test set error as trees are added to the forest and therefore, it can avoid overfitting even if the feature dimension is high. Furthermore, in each split of the procedure, it only uses some of the variables, instead of using all. The algorithm is insensitive to noise in the training labels, fast to implement, and can deal with large-scale datasets.

Support vector machine (SVM) is yet another widely used supervised classification model [88]. It aims at tracing maximum margin hyperplanes in the space where the data samples are mapped in order to separate the samples belonging to different classes. The advantage of SVMs over the previous statistical learning methods is that it introduces the concept of geometrical margin that involves only a few training samples at the boundaries (support vectors). This makes it very suitable to address the issue of limited training samples in remote sensing applications. Originally introduced for solving linear classification problems, SVMs can be generalized to non-linear decision functions by employing kernel-based SVM. The Gaussian radial basis function (RBF) is widely used in remote sensing [89].

K-nearest neighbor (KNN) is a non-parametric method widely used for the task of classification in pattern recognition. The main idea behind this classifier is to determine the category of a data according to the classification of the nearest K neighbors. The tradition KNN model has been effectively extended in a spectral-spatial collaborative manner for remote sensing image classification [67].

Next, we present a brief description of the spectral classification methods and the motivation for why spectral and spatial classifiers have been consistently gaining a great deal of attention from different researchers.

2.6.1 Spectral Feature-based Classification

Each pixel in a hyperspectral data cube corresponds to the reflected radiation of the specific region of the earth surface and has multiple values across the spectral bands. Vectors of different pixels belonging to the similar material with high probability may also have similar values. As a result, each band may reveal distinctive features of the materials of interest representing the class and hence, the original hyperspectral bands can be essentially considered as good candidate features.

However, classifying remotely sensed data by considering the complete set of spectral bands remains a challenge because the high dimensionality causes expensive computation during the analysis and eventually limiting the exploitation of the classification approaches. Particularly in the context of supervised classification, the problem of limited training samples affects the generalization capability of the classifier. While keeping the number of available samples constant, adding additional features also results in a decrease in both the classification accuracy and the generalization of the classifier. This is known as Hughes phenomenon [90]. To address this, finding a subspace that consists of the minimum number of attributes [64] required to describe the hyperspectral data has been a significant topic of research recently. A non-parametric weighted feature extraction method [55] was proposed where the feature selection process finds the best subset of features based on an adopted selection criteria. Later, the feature extraction stage generates a small number of features by data transformation based on a criteria for the optimum subspace.

Recent advances in the spectrometers technologies result in an increased spatial resolution of the collected scene. As a result, the geometrical details of the scene are high and eventually leads to the presence of objects which are made of several spatially correlated pixels. This results in an increase of the intra-class variability [91] which affects the classification accuracy when only the spectral information are considered. As the limitation of using only spectral features is identified, the need for including spatial information in the feature extraction stage has been addressed.

2.6.2 Incorporating Spatial Features

Spatial correlation statistics measure and analyze the degree of dependencies among pixels in an HSI. These statistics may reflect the relationships in a neighborhood, the distances between neighbors, the effect of the shared boundaries to determine whether they fall into a specific class etc. Several methods have been proposed [92–94] that address spatial neighborhood operations. Along with this, different kinds of texture features e.g. contrast, correlation and entropy can be generated from a gray-level co-occurrence matrix (GLCM) [95]. The spatial homogeneity or heterogeneity contributes in estimating degree of spatial correlation among the pixels across the earth surface. Hence, an integration of both spectral and contextual information can maximize the exploitation of the information residing in the HSI and can contribute significantly in the classification performance.

Spectral and spatial information can be used separately in which the spatial information is perceived in advance by the use of spatial filters. Benediktsson et al. [96] used morphological profiles (MPs) using erosion and dilation operations to enhance spatial structures that are present in the images with a series of variable sized windows sliding on the image channels. Additionally, the new values created by the filter from those channels with opening and closing method are formed into extended morphological profiles (EMPs), which are used as spatial features. Another popular way of extracting spatial features is to use attribute filters (AFs) which include mean, variance, area or circumference of the window and finally construct attribute profiles (APs) [97] as spatial features. Spatial information can also be exploited as a post-processing step to improve the initial pixelwise classification result, e.g., via mean shift [98] or MRF [99] [100, 101].

Data fusion can also be performed to combine spectral and spatial information, along with local cross-information by using composite kernels [102]. Focusing on SVMs, as kernel machines, this method is able to perform well with high-dimensional input space. A Bayesian framework that integrates spectral and spatial information together to perform supervised classification was proposed in [81]. Gormus *et. al* [103] proposed to extract intrinsic mode functions for each spectral band by applying 2D empirical mode decomposition in the spatial domain. Considering the three-dimensional structures of HSI data, combining spectral and spatial information together should result in a considerable number of discriminative features that can improve the classification performance significantly.

2.7 Deep Learning Models

Remote sensing images are reflections of the land surface and have the ability to record multiple-scale information within an area. Pixel-based, object-based, or structure-based features can be extracted from the land cover data to describe important properties of the surface. The traditional approaches exploit those features with which information-extraction models can be constructed in the form of spectral, textural and geometrical attributes of the image. However, HSIs have underlying relationships in the data and it is difficult to optimally fuse these features to effectively classify the data. Hand-crafted features are mostly designed on the basis of domain-specific knowledge and it is also not practical to address the need of considering all of the underlying details by the use of pre-designed features. Hence, it is nearly impossible for those features to achieve an optimal balance between discriminability and robustness.

Recently, Deep Learning (DL) algorithms have been introduced into hyperspectral remote sensing applications and have achieved outstanding performance compared with those of traditional learning algorithms. A deep model learns the representative and discriminative features in a hierarchical manner to model high level abstraction of data.

Considering low-level features at the low-levels, DL can represent and organize multiple levels of information to express complex relationships between data [16]. We now present the basic architectures of the existing DL models.

2.7.1 Convolutional Neural Networks

The convolutional neural network (CNN) has established itself as a leading model in deep learning community as researchers have successfully adopted it in a wide range of applications in image processing, including image classification [104], object detection [16], super resolution restoration [105] etc. CNNs are a class of deep learning models that integrate feature extraction, feature combination and classification with a single neural network that is trained end to end from raw pixel values to classifier outputs. CNNs consist of multiple convolutional, pooling and fully connected layers, with nonlinear activation functions applied at the end of each layer.

During a convolution operation, small regions of the input maps are convolved with learnable kernels and are subsequently transferred through the activation functions to construct the output feature maps. One major advantage of a CNN is that it allows the use of shared weights in convolutional layers, within same feature maps, which reduces the number of parameters significantly. A convolutional layer is followed by a translation invariant pooling layer which is used to reduce the dimensionality of the feature maps. Average pooling and max pooling - these two types of pooling operations are most commonly used. There are few other pooling operations available such as spatial pyramid pooling [104], stochastic pooling [106] and def-pooling [107].

The output maps obtained from the last convolutional and pooling layer are flattened into one-dimensional vector which is the input to the first fully connected layer. The output generated by the final fully connected layer is considered to be the learnt feature which is later used to compute the loss function with regard to the ground truth labels of the input data. Based on the value computed by the loss function, the error is propagated back into the network and the weights are updated in response to the gradients. In this way, a higher-level representation of the raw input image is formed and can be used to train a classifier, e.g., a softmax classifier, to perform classification [108]. Shelhamer *et. al* [109] proposed a novel convolutional network in which the fully connected layer is replaced with a deconvolutional layer. The basic working principle of a CNN is graphically represented in Fig. 2.6.

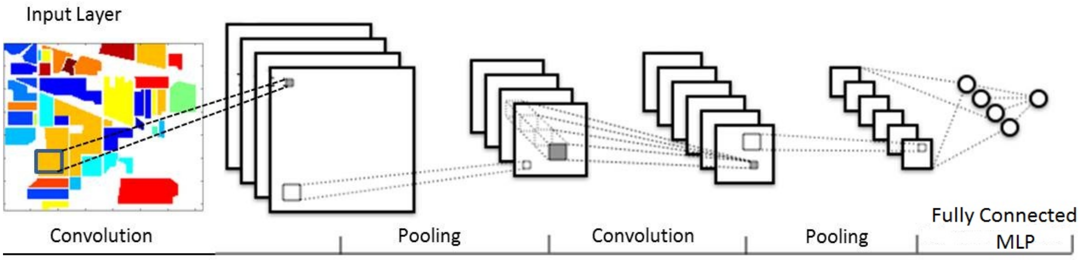


FIGURE 2.6: A basic CNN architecture.

2.7.2 Stacked Autoencoder

A stacked autoencoder (SAE) is a deep architecture consisting of multiple layers of sparse autoencoders (AEs) in which the outputs of each layer is connected to the inputs of the next layer. An AE learns a representation of the input data by encoding, usually for reducing dimensions and then reconstruct into a representation from the compressed data that closely matches the original data. An AE consists of a visible layer of d inputs, one hidden layer of h units with an activation function f. During the training of the network, f first transforms the input vector $x \in \mathbb{R}^d$ into a hidden representation $y \in \mathbb{R}^h$ and y is then mapped back to a reconstructed $z \in \mathbb{R}^d$ in the output layer containing the same number of nodes as the input layer, and with the purpose of reconstructing its own inputs instead of predicting the target value given inputs. The procedure can be formally written as:

$$y = f(W_y x + b_y)$$

$$z = f(W_z y + b_z)$$

where W_y, W_z represent the weights of the input-to-hidden layer and hidden-to-output layer respectively and b_y, b_z are the the biases of hidden and output units respectively. The loss function $L(\theta)$ measures the reconstruction $\mathbf{z}$ with respect to the input $\mathbf{x}$:

$$L(\theta) = \frac{1}{2N} \sum_{n=1}^{N} ||z^{(n)} - x^{(n)}||_2^2$$

where N is the number of training samples, $\theta = (W, b_y, b_z)$ is the set of parameters for minimizing the difference between the reconstructed output and the original input over the entire training set $X = [x^{(1)}, x^{(2)}, \ldots, x^{(n)}, \ldots, x^{(N)}]$. Stochastic gradient descent algorithm [110] can be used to efficiently implement this. Fig. 2.7 shows the structure of a basic AE:

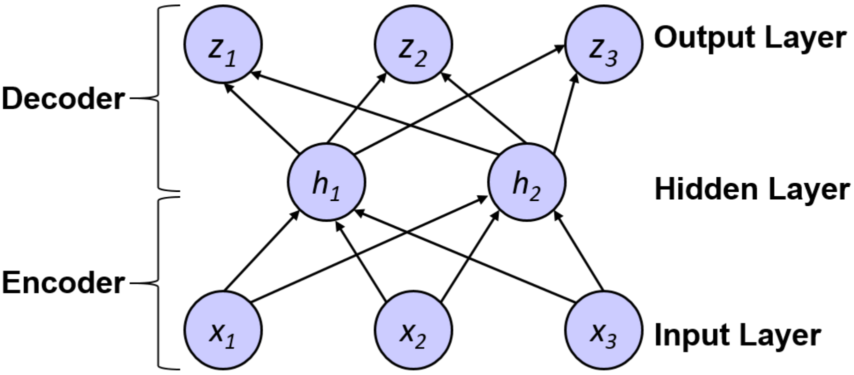


Figure 2.7: Structure of an autoencoder.

There are three well-known variants of AEs, namely denoising AE, sparse AE and variational AE. A denoising AE [111] typically forces the model in capturing the structure of the input distribution and has the ability to recover the correct input from a noisy version. Sparse AE [112] is aimed at minimizing the reconstruction error by imposing a sparsity constraint. It can be achieved by additional terms in the loss function during training by comparing the probability distribution of the hidden unit activations with some low desired value. Variational AEs have the distinct property of allowing easy random sampling and interpolation. It does this by making the encoder to output two vectors: a vector "means" and another of "standard deviations". The mean vector generally controls where the encoding of an input should be centered around and the standard deviation controls how much from the mean the encoding can vary. In this way, the decoder is able to learn that all nearby points to the distribution in the latent space refer to a sample of that class.

The structure of SAEs is to stack n hidden layers by an unsupervised layer-wise learning algorithm and then fine-tuned by a supervised method. It has mainly three steps:

1. Train the first AE by the original input data and generate the learned feature vector.

2. The feature vector of the previous layer is used as the input for the next layer and this step is repeated until the training achieves convergence.

3. After all the hidden layers containing individual AEs are trained, backpropagation algorithm may be used to update the weights with respect to the training set.

A basic representation of an SAE is illustrated in Fig. 2.8. As indicated in the figure, SAE consists of multiple layers of AEs, each of which is a type of neural network used for efficient encodings.

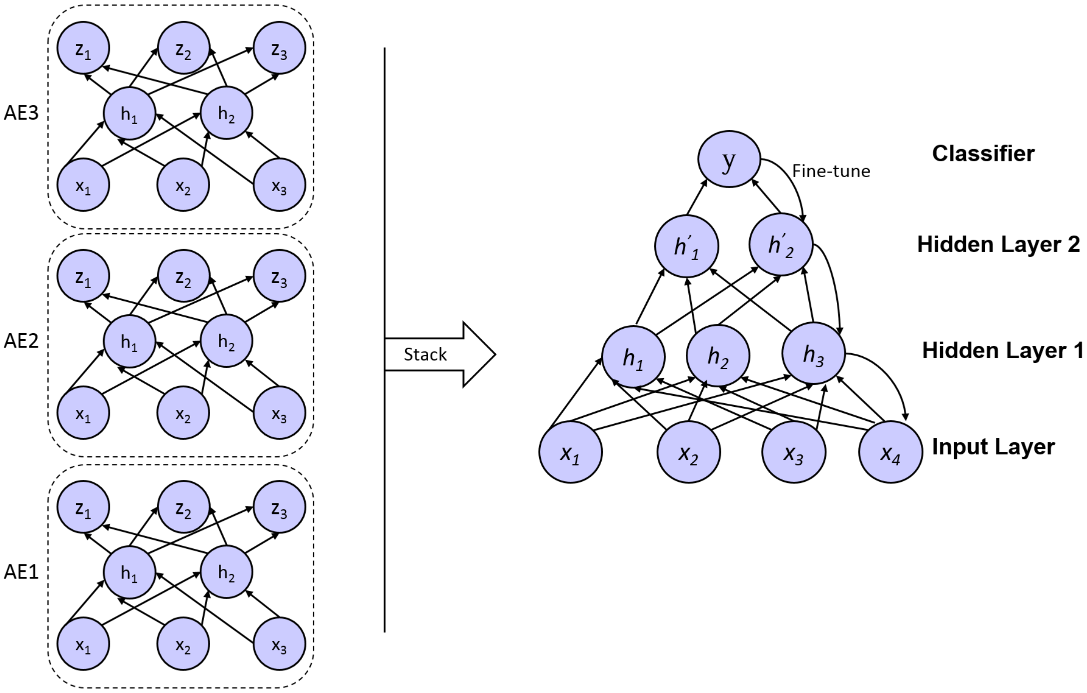


Figure 2.8: Structure of a stacked autoencoder.

2.7.3 Deep Belief Betworks

A deep belief network (DBN) is another deep model structure, capable of extracting high-level, invariant features of HSI data, which can contribute to an improved classification. A DBN model is built with a hierarchically organized series of restricted boltzmann machines (RBMs).

RBM is regarded as a layer-wise training model in forming a DBN. It has a two-layer structure with the lower layer represents a particular type of MRF with "visible" units $v = \{0,1\}^D$ and the higher layer represents the "hidden" units $h = \{0,1\}^F$. The typical structure of an RBM is depicted in the Fig. 2.9

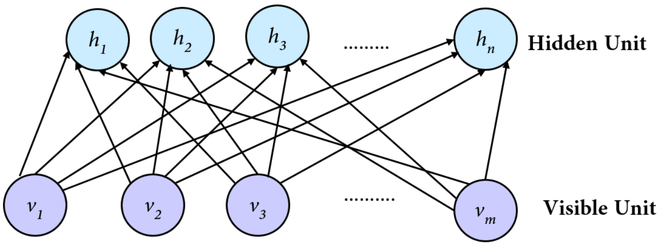


Figure 2.9: A typical RBM

An energy function representing the joint distribution can be given by [113]

$$E(v,h;\theta) = -\sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} a_j h_j - \sum_{i=1}^{D}\sum_{j=1}^{F} w_i v_i h_j$$

$$= -b^T v - a^T h - v^T W h \tag{2.5}$$

where the model parameters $\theta = \{b_i, a_j, w_{ij}\}, w_{ij}$ are the weights between visible and hidden units, b_i and a_j are the respective biases of visible and hidden units. The joint distribution over the visible and hidden units is defined by

$$P(v,h;\theta) = \frac{1}{Z(\theta)} exp(-E(v,h;\theta)) \tag{2.6}$$

$$Z(\theta) = \sum_v \sum_h E(v,h;\theta) \tag{2.7}$$

where $Z(\theta)$ is the normalizing constant, which is the sum of the numerator of all values of v and h. The energy function supplies a probability to every input vector and the the probability can be raised by adjusting θ as given in Eq. 2.5. The conditional distributions of h and v are given by the following logistic functions

$$P(h_j = 1|v) = g(\sum_{i=1}^{D} W_{ij} v_i + a_j) \tag{2.8}$$

$$P(v_i = 1|h) = g(\sum_{j=1}^{F} W_{ij}h_j + b_i) \tag{2.9}$$

where

$$g(x) = \frac{1}{1 + exp(-x)} \tag{2.10}$$

By using the probability computed from 2.9, it is possible to reconstruct the input data by setting each v_i to 1. After updating the hidden units, the reconstructed features can be obtained. Contrastive divergence (CD)[114] can be used to learn the corresponding weights. If the model is able to correctly recover the data, the hidden unit is then believed to have captured sufficient content from the input.

Multiple RBMs are arranged as a layer-by-layer learning process to finally construct the DBN. After the first RBM is trained, the learnt features can be subsequently used as an input to the second RBM. The process is repeated until the output from the last RBM are assumed to be the final deep features of the input data. The fine-tuning of the network can be done by adding a logistic regression layer at the end and using the standard back-propagation algorithm. A conventional DBN to classify an HSI is illustrated in Fig. 2.10.

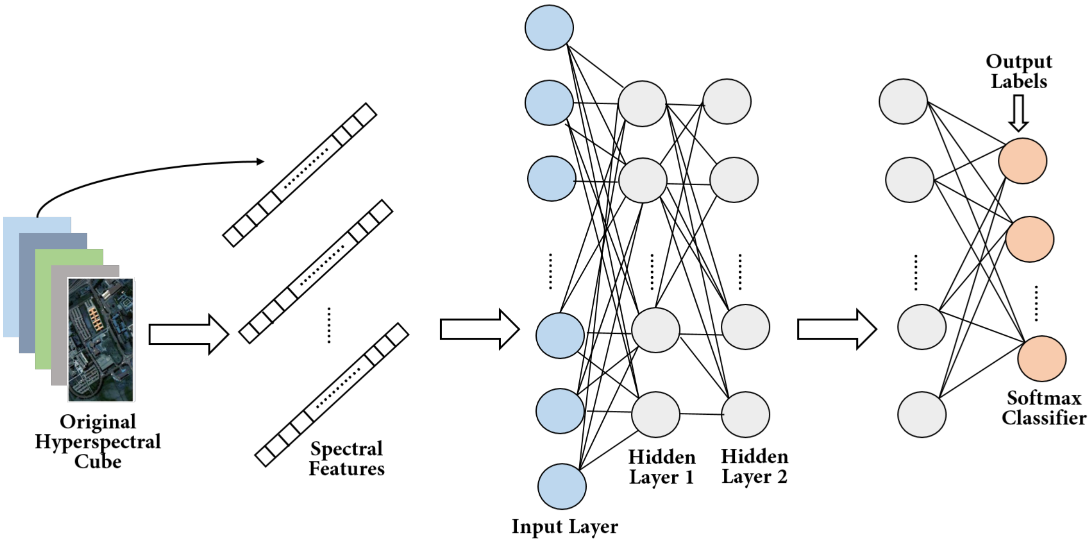


FIGURE 2.10: Training of a typical DBN to classify hyperspectral data

2.7.4 Generative Adversarial Network

The traditional deep learning models have shown striking success in building discriminative models, used for mapping high-dimensional input to a class label. However, similar successes were not observed for deep generative models, mainly due to the difficult inference and learning of the probabilistic models that require computations from the conditional posterior over hidden layers. As a result, exact sampling from these distributions is intractable. To overcome these difficulties, Goodfellow *et al.* [115] proposed a generative adversarial network (GAN), in which a discriminative model learns to determine whether a data sample belongs to a real distribution or a noise distribution. To impose challenges for the discriminative model, the generative model produces samples from random noises. Both models are designed as multilayer perceptrons and are trained using backpropagation algorithm.

GANs have been extended to the context of semi-supervised learning to design the discriminator in producing class labels. Springenberg et al. [116] proposed CatGAN in which the objective function was modified to consider the mutual information between observed samples and their predicted class distributions. In [117], the features learned by the discriminator are reused in classifiers. Odena *et al.* [118] proposed SGAN in which the generative model and the classifier are learned simultaneously and improves classification performance on restricted datasets with no generative component.

2.8 Deep Learning For Remote Sensing Image Classification

Following the success of DL in natural language processing, DL-based methods [119–122] have been introduced in hyperspectral remote sensing image classification and achieved significantly good results. These methods can be organized into three categories: spectral information, spatial information and spectral-spatial information respectively. We now present a review of each of these categories below:

2.8.1 Spectral Feature Classification

DL models can automatically extract high-level and abstract features from raw input data which are more effective than hand-crafted features. In the existing DL-based spectral feature classification methods, the raw spectral vectors are provided as input and the model generates deep spectral features which are later used for classification. Traditional CNNs generally take two-dimensional images as input and generate feature maps accordingly. Hu *et. al* [123] proposed a 1D-CNN model to perform HSI classification. The model contains one convolutional, one pooling and two fully connected layers. Mei et. al [124] proposed another 1D-CNN model but instead of using pooling layer, the model uses dropout and batch normalization.

Similar to the architecture of 1D-CNN, other deep models including SAE and DBN also have the ability to take raw spectral vectors as input and generate deep spectral features which are used for classification. Chen *et al.* [120] proposed a SAE-based model in which the relationship between the input layer and reconstruction layer was exploited in a way that the weights between the hidden and output layers are the transposition of those of input to hidden layer. A distance prior was introduced during the fine tuning of the SAE in [125], to give more effective guidance in extracting features in case of insufficient labeled samples. Zhong *et. al* [126] introduced diversity promoting priors in terms of diversity promoting conditions into the objective function optimization during the pre-training and fine tuning of DBN. Fig. 2.11 illustrates a general framework of DL-based models for spectral feature classification.

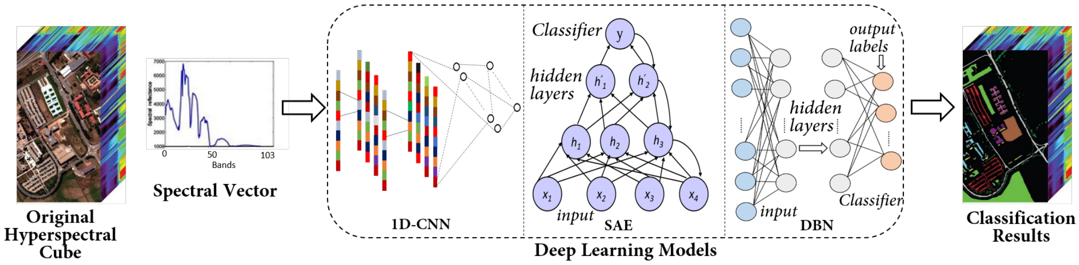


FIGURE 2.11: General framework of deep learning for spectral feature classification

2.8.2 Spatial Feature Classification

Since HSIs contain spectral and spatial contents simultaneously, considering only spectral information sometimes fail to extract effective features and consequently, fails to produce an improved classification accuracy. Therefore, it is better to consider spatial information along with spectral features in order to effectively classify remote sensing images.

In the literature containing DL-based spatial feature classification methods, spectral vectors within a neighborhood region of a given pixel are provided as inputs and spatial features of that pixel is extracted accordingly. In this regard, 2D-CNN models were introduced which can learn spectral features within neighborhoods in raw 2D images. However, due to the large number of spectral bands, the data in a neighborhood region suffer from high dimensionality. To address this issue, dimensionality reduction methods were applied in several DL models. Makantasis *et. al* [127] proposed a randomized PCA to reduce the HSI dimensions to 10 bands/principal components first and then applied a 2D-CNN to extract deep features from the reduced number of HSI bands. A 5×5 spatial neighborhood was used in this work. A trade-off between the window size and the number of PCs was observed in [128] where a 42×42 window size was used but only three PCs were used.

Additional 2D-CNN-based models were introduced in which multiscale spatial features were captured. Zhou et. al [129] used Laplacian pyramid transformation to capture multiscale data from the compressed HSI and then fed into an independent 2D-CNN to extract final features. Attribute profiles were introduced during the additional preprocessing of the HSI image with PCA in [130] before 2D-CNN was used for classification. Following this approach of using PCA before employing 2D-CNN, a further post-processing of the results with sparse coding was done in [131] for classification.

Similar strategies were executed for SAE and DBN as well for spatial feature classification. After employing PCA to reduce the dimensions, SAE is included to perform the classification. During the pre-training, the weights of the hidden-to-output layer are restricted to the transposition of the weights of the input-to-hidden layers [132]. Similar approach was adopted for DBN structures as well [133]. An illustration of DL-based spatial feature extraction methods are given in Fig. 2.12.

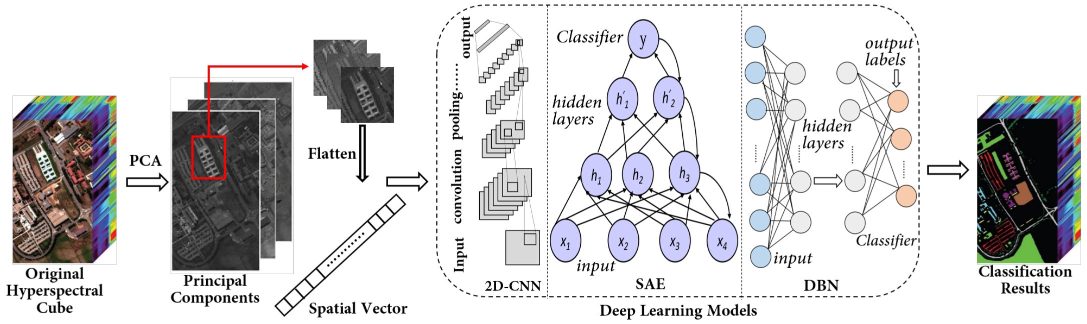


FIGURE 2.12: General framework of deep learning for spatial feature classification

2.8.3 Spectral-Spatial Feature Classification

In order to take full advantages of the powerful structure of HSI, it is important to combine both spectral and spatial information to classify data. There are two ways to do this: (1) extract spectral and spatial features separately and then, combine the features (1D-CNN, 2D-CNN, SAE and DBN are such few examples). and, (2) extract deep spectral-spatial features from 3D-cubes directly.

In [122], 1D-CNN and 2D-CNN were used to extract spectral and spatial featuers respectively and later their individual outputs are combined together to be fed into a softmax classifier. A framework [128] combining SAE, used for spectral features extraction and a 2D-CNN, used for spatial feature extraction was used, in which the spatial features were learned by spatial pyramid pooling.

The main challenge experienced during the use of 2D-CNN is the additional dimension of the 3D structure of HSI image. To address this issue, 2D-CNN architecture can be either extended to 3D-CNN or rearrang the 3D HSI cube to 2D image. Chen *et. al* [119] introduced a 3D-CNN to learn deep spectral-spatial features by taking cubes of 27×27 spatial size as inputs while a similar 3D-CNN architecture [122] uses input cubes of 5×5. Lee *et. al* [134] proposed a model that convolves the 3D subcubes with 3×3-sized and 1×1-sized convolution kernels, and then reconstructs the new 3D data using the convolved outputs jointly.

Spectral-spatial models containing SAE and DBN generally extract spectral and spatial features separately and then combine those to produce spectral-spatial features. In such cases [120, 121], spatial information are flattened to 1D-vector, as both SAE and DBN can process only 1D input. A spatially updated deep AE was used in [135] for spectral-spatial feature extraction in which a sample similarity regularization procedure combined with a collaborative representation-based classification was employed. An unsupervised convolutional sparse AE with an window-in-window selection strategy was adopted in [136] to extract spectral-spatial features. An illustration underlying these approaches is shown in Fig. 2.13.

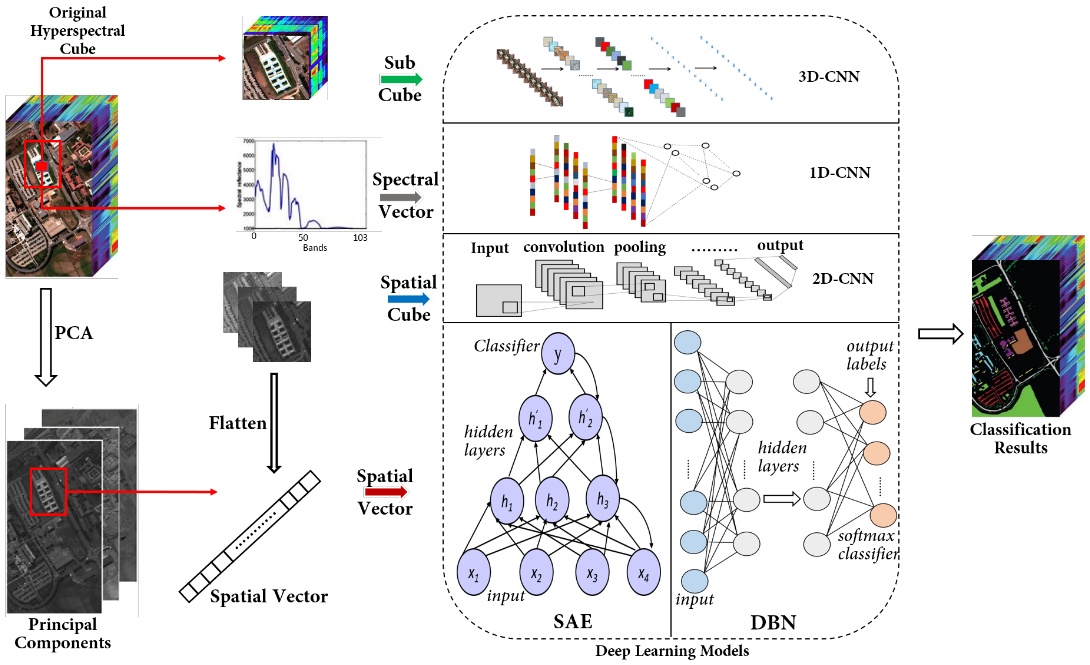


FIGURE 2.13: General framework of deep learning for spectral-spatial feature classification

Recurrent neural network (RNN) is yet another powerful DL-based model, capable of processing sequential data. Initially used for speech data, this model has been adopted in hyperspectral remote sensing image classification. Mou *et al.* [137] proposed a classification method in which an RNN is used to capture the sequential property of an HSI pixel vector. In this method, they used parametric rectified unit in order to avoid the risk of divergence during training. Wu *et al.* [138] proposed a convolutional RNN in which recurrent layers were added after convolutional layers in order to extract middle-level and locally invariant features.

2.8.4 GANs for Remote Sensing Image Classification

Although the architecture of a traditional GAN is very promising, yet very little research has been done on adopting GANs in remote sensing. To handle the extremely time consuming process of labeling huge amount of remote sensing data, GANs can be adopted because the required quantities of training data may be provided by the generator during the training. In this regard, Lin *et al.* [22] proposed a multiple-layer feature-matching generative adversarial networks (MARTA GANs) to learn a representation using only unlabeled data. To fit the complex properties of remote sensing data, they introduced a fusion layer to merge the mid-level and global features. In [7], the authors introduced the scaled exponential linear units (SeLU) instead of ReLU and batch normalization. By adding the SeLU, high-quality and large-sized samples were generated for remote sensing images. He *et al.* [23] proposed a semi-supervised learning model in which three-dimensional bilateral filter (3DBF) was adopted to extract the spectral-spatial features from the HSI data. The GANs were subsequently trained on those spectral-spatial features by adding samples from the generator to the features and increasing the dimension of the classification output.

2.9 Hyperspectral Image Datasets

AVIRIS is a unique optical sensor that captures spectral radiance in 224 contiguous spectral channels with wavelengths from 400 to 2500 nanometers using a high-altitude aircraft. The main objective of the AVIRIS project is primarily focused on understanding and analysing processes related to earth's surface and identifying environmental changes. ROSIS, another advanced sensor, is particularly designed to measure ocean parameters including water biomass detection and water quality measurements and also over land surfaces. Earth Observing-1 (EO-1), first satellite in NASA's New Millennium Program Earth Observing series, has a unique feature that carries an experimental hyperspectral imagery (Hyperion) that can capture high resolution images in 220 contiguous spectral channels of the earth surface.

During our experiments, we used three widely used hyperspectral datasets captured by the sensors mentioned above, in order to evaluate the effectiveness of our proposed method. These are: Indian Pines, Salinas and Pavia University. We now briefly introduce the details regarding the datasets.

2.9.1 Indian Pines

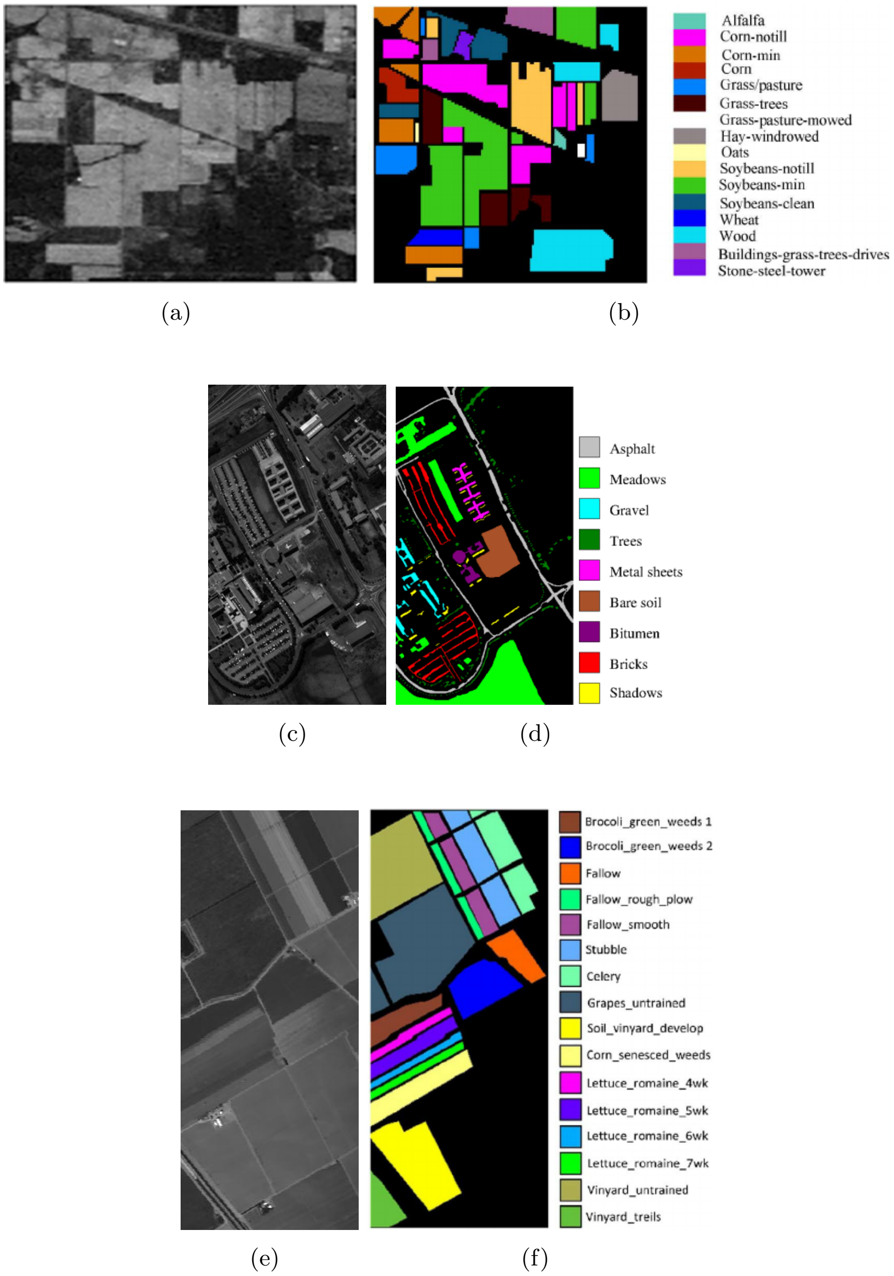
Indian Pines data, acquired by the AVIRIS, consists of 145×145 pixels and 220 spectral bands in the wavelength range of 0.4-2.5 μm with a spatial resolution of 20 meter/pixel [139]. This scene covers a test site in North-western Indiana, covering two-thirds agriculture, and one-third forest. The original 224 number of spectral bands was reduced to 200 by discarding water absorption bands. Sixteen different classes on land-cover were considered in the ground truth.

2.9.2 Pavia University

Pavia University dataset, collected by ROSIS-3, consists of 610×340 pixels and 115 spectral bands and has a high spatial resolution of 1.3 meter/pixels [139]. The number of bands was reduced to 103 by removing noisy bands. The scene, captured during a flight campaign over Pavia, nothern Italy, has nine different classes on land-cover.

2.9.3 Salinas

This scene was collected by AVIRIS covering Salinas Valley, California and entails a high resolution of 3.7 meter/pixel. The scene consists of 512×217 pixels and has 224 contiguous spectral bands. 20 water absorption bands are removed and the rest of the bands include vegetables, bare soils, and vineyard fields, comprising 16 classes [140]. Fig. 2.14 shows randomly chosen bands and the ground truth of of the images for Indian Pines, Pavia University and Salinas datasets respectively.



(a) (b)

(c) (d)

(e) (f)

FIGURE 2.14: (a) Sample band (b) ground truth for Indian Pines (first row), Pavia University (second row) and Salinas (third row).

Chapter 3

Conditional Random Field and Deep Feature Learning for Hyperspectral Image Classification

In this chapter, we propose a method to classify hyperspectral images by considering both spectral and spatial information via a combined framework consisting of CNN and CRF. We use multiple spectral band groups to learn deep features using CNN, and then formulate deep CRF with CNN-based unary and pairwise potential functions to effectively extract the semantic correlations between patches consisting of three-dimensional data cubes. Furthermore, we introduce a deep deconvolution network that improves the final classification performance.

3.1 Introduction

In Section 2.5, we introduced widely used remote sensing classification methods and discussed the importance of including spectral and spatial information in building a more meaningful representation of our hyperspectral data. Spectral-spatial classification methods can be divided into two categories. The first category uses the spectral and spatial information separately in which the spatial information is perceived in advance by the use of spatial filters [141]. After that, these spatial features are added to the spectral data at each pixel. Then dimensionality reduction methods can be used before the final classification. The spatial information can also be used to improve the initial pixelwise classification result as a post-processing step, e.g., via mean shift [98] or Markov random field [99], which is a very common strategy in image classification and segmentation [100, 101]. The second category combines spectral and spatial information for classification and segmentation. Li *et al.* proposed to integrates the spectral and spatial information in a Bayesian framework, and then use either supervised [81] or semi-supervised algorithm [142] to perform an additional step that improves the initial classification results. Yuan *et al.* [143] combined spectral and texture information where linear filters were used to supply enhanced spatial patterns. Since hyperspectral data are normally represented in three-dimensional cubes, the second category of methods can result in a large number of features containing discriminative information which are effective for better classification performance.

Recent advances in “deep” architectures such as CNN have contributed immensely in classifying spectral-spatial features [121, 144]. In some CNN-based hyperspectral data classification methods [145], the spatial features are obtained by a 2D-CNN model which exploits the first few principal component bands of the original hyperspectral data. Yu *et al.* [146] proposed a CNN architecture which uses a convolutional kernel to extract spectral features along the spectral dimension only. To obtain features in the spatial domain, they used normalization layers and a global average pooling layer. On the other hand, 3D-CNN can learn the signal changes in both spatial and spectral dimensions of local spectral images. Therefore, it can extract significant discriminative information for classification and exploit powerful structural characteristics for hyperspectral data. Recently, this model was adopted by Chen *et al.* [119] for feature extraction and classification of hyperspectral images based on three-dimensional data across all the bands, which combines both spectral and spatial information. Similar works have been proposed to extract spectral-spatial features from pixel or pixel-pairs using deep CNN [145–147].

Because CNN can effectively discover spatial structures among the neighboring patches of the input data, the resulting classification maps generally appear smoother in spite of not modeling the neighborhood dependencies directly. However, the possibility of reaching local minima during training of CNN and the presence of noise in the input images may create holes or isolated regions in the classification map. Compared with other machine learning methods, CNN is limited by the absence of shape and edge constraints. As a result, the final classification map appears rough on edges. Moreover, in hyperspectral remote sensing images, cloud shadows and topography cause variations in contrast, which often generates incorrect classes in images. The presence of cloud also may hide regions or decreases the contrast of regions. Due to these reasons, CNN sometimes recognizes only parts of the regions properly [148].

In these circumstances, a further refinement stage produces an improved classification output. To this end, combining probabilistic graphical models such as MRF and Conditional Random Field (CRF) with CNN brings significant improvements by explicitly modelling the contextual information between regions. CRF is traditionally used to perform image segmentation after an initial coarse pixel-level class label has been generated. The goal is to make pixels in a local neighborhood having the same class label. From this perspective, the outcome of CRF can be considered as an improved classification map. For these reasons, there has been a recent trend on exploring the integration of CNN and CRF methods [17–20]. For example, Liu *et. al* [21] used CRF to improve the segmentation outputs as a post-processing step. However, the CRF is entirely disconnected from the training of the deep network. Instead of using a disconnected approach, Chen *et al.* [149] proposed a fully connected Gaussian CRF model where the respective unary potentials were supplied by a CNN. Since CRF can directly model spatial structures, if it can be formulated in a deep modeling approach, the trained model will integrate the advantages of both CNN and CRF in modeling the spatial relationships in the data. Based on this combination, CRFs can better fine-tune model features by using the incredible power of CNNs.

To improve the upsampled low-resolution prediction by a traditional CNN, Zheng *et al.* [150] formulated a dense CRF with Gaussian pairwise potentials as a Recurrent

Neural Network (RNN). We argue that this stage of refinement process can be further improved by applying more advanced refinement methods, such as training a deconvolution network [109]. Deconvolution network [151] was employed to visualize activated features in a trained CNN and update network architecture for performance enhancement. Hence, it is plausible to use it in order to further improve the final classification performance.

In computer vision applications, it is usually common to train a deep network with large amount of samples. This, however, is a very challenging task in hyperspectral remote sensing applications since the number of training samples is limited. Without abundant training samples, a deep network faces the problem of "overfitting" which means the representation capability may not be sufficient to perform well on test data. It is therefore very important to increase the size of the training samples in order to handle this overfitting issue.

In our method, we treat hyperspectral images as spectral groups consisting of the image spanning over a few spectral bands instead of all the bands across the spectra as in [119]. Such smaller-sized yet a large number of spectral groups will be able to provide more accurate local spectral-spatial structure description of the data. Our framework 3DCNN-CRF, as shown in Fig. 3.1, starts with generating a feature map obtained by applying a 3D-CNN over the spectral groups. We then introduce 3DCNN-based deep CRF in our framework by using the output of the 3D-CNN. We calculate the unary and pairwise potentials of the CRF by extending a CNN-based deep CRF architecture [152] to cope with both spectral and spatial information along the entire spectral channels. Then a mean-field inference algorithm [153] was used to generate a classification map. Finally, a deconvolution network was adopted to improve the final classification performance accordingly.

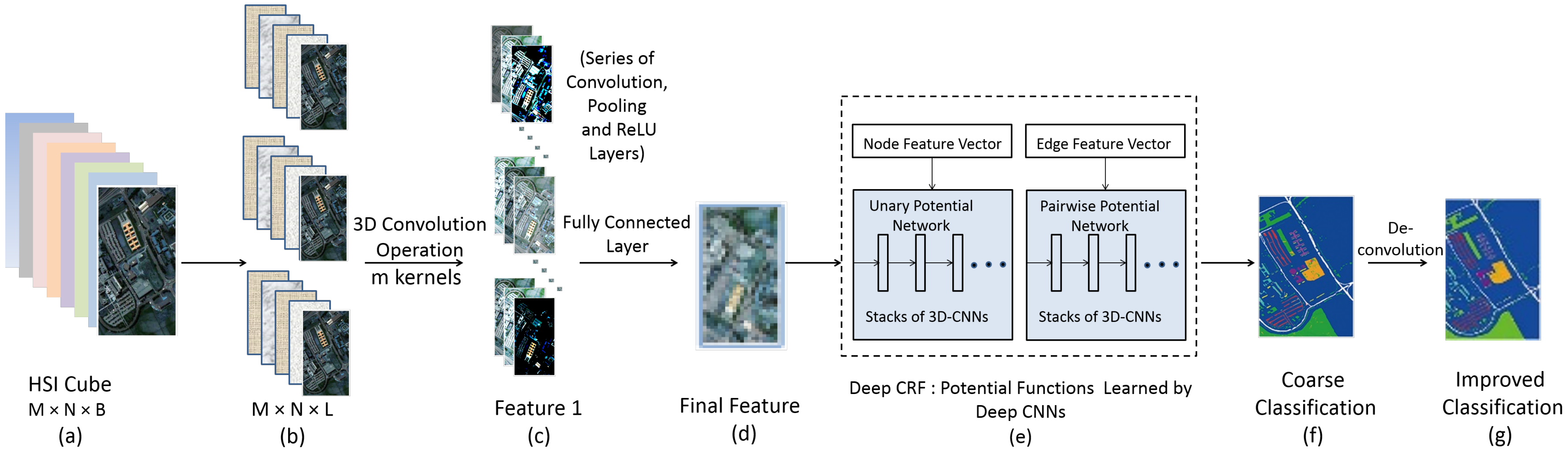


Figure 3.1: Proposed architecture of 3DCNN-CRF. (a) original hyperspectral cube with B bands. (b) different band groups consisting of L bands ($L \ll B$) each. (c) resulting feature maps after first convolution. (d) resulting final spectral-spatial features produced by 3D-CNN. (e) 3D-CNN-based Deep CRF (f) coarse classification map after applying 3D-CNN-based deep CRF. (g) improved classification map after applying deconvolution.

The main contributions of this chapter are as follows:

- 3D-CNN is performed on spectral groups containing a small number of bands, which results in a more effective spectral-spatial structure representation of the hyperspectral data.
- CNN-based general pairwise potential functions in CRFs are extended to explicitly model the spatial relations of local neighborhood along the spectral dimension that results in an end-to-end training scheme from input image to classification output.
- 3DCNN-CRF also learns a deep deconvolution network during CRF pairwise potential training that improved the final classification performance to a considerable extent.
- The size of the training set is augmented by generating virtual samples from real ones in different band groups, which produces different yet useful estimation of spectral-spatial characteristics of the new samples.
- A new hyperspectral dataset is created and the image regions containing relevant classes are manually labelled. 3DCNN-CRF is evaluated on this dataset and compared with alternative approaches accordingly.

3.2 3D-CNN for Hyperspectral Feature Representation

In this section, we explain the first stage of our method for obtaining effective spectral-spatial structure representation of the hyperspectral data using 3D-CNN. In many deep learning based methods [119, 147], the generated spectral-spatial features can be used to obtain a classification map. In our method, however, we further provide the features as input to CRF in order to produce improved classification results. When 3D kernels are used as the convolution operations on the spectral groups, a common practice is to convolve the 3D kernels with a spatial neighborhood along the entire spectral channels [119]. However, in our method, the original image, which has B bands, is divided spectrally into several images consisting of neighboring L bands where $L \ll B$. 3D convolution filters are applied to each of these different band group images. These groups of bands provide more detailed local spatial-spectral information so as to let different wavelength ranges make distinct contribution to the final classification outcome. Repeated convolution operations produce multiple feature maps along the spectral dimension. Let (x, y) define a location in the spatial dimension and z be the band index in the spectral dimension. The value at a position (x, y, z) on the c^{th} feature map is given by [122]:

$$val_{lj_{(c)}}^{xyz} = f(\sum_{i=1}^{m} \sum_{p=0}^{P_l-1} \sum_{q=0}^{Q_l-1} \sum_{r=0}^{R_l-1} k_{lij}^{pqr} val_{(l-1)ij}^{(x+p)(y+q)(z+r)} + b_{lj}) \tag{3.1}$$

where l is the current layer that is being considered, m is the number of feature maps in the $(l-1)$-th layer (previous layer), j is the current kernel number, i is the current feature map in the $(l-1)$-th layer connected to the feature map of the l-th layer, k_{lij}^{pqr} is the (p, q, r)-th value of the kernel connected to the i-th feature map in the previous layer. P_l and Q_l are the height and the width of the kernel respectively, and R_l is the size of the kernel along the spectral dimension.

val_{lij}^{xyz} is calculated by convolving a feature map of the previous layer with a kernel of the current layer. In this process, the number of feature maps in the previous layer will be multiplied by the number of kernels in the current layer which will produce as many feature maps as the output of the l-th convolution layer. Therefore, 3D convolution can preserve the spectral information of the input data.

After the convolution operations, the intermediate feature maps pass through the pooling layers and the activation functions. Finally, the feature maps consisting of the data cubes are flattened into a feature vector and feed into a fully-connected layer which extracts the final learned deep spectral-spatial features. The entire network is trained using the standard back propagation algorithm.

After the 3D-CNN training, the learned parameters θ_λ contain information distinct to each band group along the spectral channel $\lambda \in \mathbb{B}$. Such representation is very useful for deep learning framework as the model will be able to receive as much information as required to identify interesting patterns within the data. The procedure of obtaining spectral-spatial features by 3D-CNN is summarized in Algorithm 1.

Algorithm 1: 3D-CNN Training Algorithm

```
Input: T Input Samples $\{X_1, X_2, \ldots, X_T\}$, $Y$ target labels in
$\mathbb{Y} = \{y_1, y_2, \ldots, y_Y\}$, number of BP epochs $R$
for Each subcube $M \times N \times L$ in $\lambda$ do
    while epoch $r : 1 \rightarrow R$ do
        while training sample $i : 1 \rightarrow T$ do
            Perform convolution operations using Eq. (1) to generate intermediate
            feature maps.
            Compute Soft-max activation $a = \frac{exp(o)}{\sum_k exp(o_k)}$; where $o$ is the output of the
            final layer of the network and first input to softmax classifier
            Compute error $T = \mathbf{y}_i$-$\mathbf{a}$
            Back-propagate error to compute gradient $\frac{\delta T}{\delta o_j}$
            Update network parameter $\theta_\lambda$ using gradient descent $\Delta w_{ij} = -\varepsilon \frac{\delta T}{\delta w_{ij}}$ where
            $\varepsilon$ is the learning rate
        end while
    end while
end for
Output: Trained CNN parameters $\theta_\lambda$
```

The resulting 3D features map is used to formulate our proposed 3DCNN-CRF as explained in Section 3.3.

3.2.1 Addition of Virtual Samples

In many occasions, substantial number of weights in a CNN introduces local minima in the loss function and eventually restricts the classification performance. To overcome this issue, large amount of samples can be used to update weights during the training procedure. Unfortunately, the process of obtaining samples, which normally requires manual labelling, is time consuming and expensive. In remote sensing applications, the number of available training samples is usually limited. This imposes a great deal of challenges to the adoption of a deep network model.

To address this issue, the size of the training samples can be augmented by virtually sampling from the existing labelled samples. Remote sensing scenes generally cover large areas where pixels belonging to same class in different locations fall under different radiations. We can simulate this lighting condition in order to generate virtual samples by multiplying with a random factor and also adding a gaussian noise. A virtual sample $y_{(\lambda)}$ is therefore generated from two real samples of the same class represented by $x_{i_{(\lambda)}}$

and $x_{j_{(\lambda)}}$ along the spectral channel $\lambda \in \mathbb{B}$ by

$$y_{(\lambda)} = \epsilon_i x_{i_{(\lambda)}} + (1 - \epsilon_i) x_{j_{(\lambda)}} + \beta \tag{3.2}$$

where ϵ indicates the effects of light intensity, which vary under different atmospheric conditions and β is the added random Gaussian noise, which may result from the interactions of adjacent pixels.

$y_{(\lambda)}$ is assigned with the same class label as $\epsilon_i x_{i_{(\lambda)}}$ since the hyperspectral characteristics of the new fused virtual sample shall be sitting between $x_{i_{(\lambda)}}$ and $x_{j_{(\lambda)}}$ which belong to the same class. Moreover, we generate virtual samples within band groups and hence, they give us multiple spectral information of the same sample from different wavelengths so as to further augmenting the training data. We insert these new samples into separate images by replacing the real samples which are used for the fusion. The original image containing the limited number of real samples and the augmented images containing the virtual samples obtained from different wavelengths are used for training the CNN.

We further augment the training samples by transforming the sample pixels using rotation and flipping operations. In this approach, each pixel and its 5 × 5 neighbours are considered as one sample. Therefore, within each band group, the size of each sample is $5 \times 5 \times L$ where L is the number of bands. We rotated (90°, 180°, 270°) the samples and also flipped the four samples (the original one and its three variations) to produce additional transformed images. This leads to 7 combinations of images within each band group which significantly increase the amount of available data. From these transformed images, we select limited number of samples for training. Therefore, the smaller number of real samples and the augmented virtual samples, generated by sample fusion and the smaller number of augmented virtual samples generated by transformation operations, are used together to train the 3D-CNN. During the training, part of the real and virtual samples are used for learning the weights of the neurons and the rest are used for validating and updating the architecture. We report the total number of training and testing samples for each class used on the datasets in Section 3.5.6.

3.3 Constructing Deep CRFs

In many cases [119, 147], CNN can effectively discover spatial structures among the neighboring patches of the input data, which generates smooth classification maps in spite of not modeling the neighborhood dependencies directly. However, it still encounters several problems such as

- There are holes or disconnected regions in the classification map obtained by a CNN due to the occurrence of reaching local minima during the training.
- CNN is generally limited by the absence of shape and edge constraints. The final classification result may appear rough on edges of some regions or objects.
- In hyperspectral remote sensing, cloud shadows and topography cause variations in the spectral responses and influence the contrast of regions, which generates incorrect classes in images. As a result, the CNN sometimes recognizes only parts of the regions.

To resolve these critical issues in such classification methods, we, therefore, propose an end-to-end modelling approach by integrating deep CRF with the spectral-spatial features obtained from the first stage in order to utilize the properties of both CNN and CRF to better characterize the spatial contextual dependencies between the regions. We believe that such end-to-end learning approach is very suitable for hyperspectral image analysis as the integrated models will make full use of the spatial relationships among spectral band groups to perform the final classification. This is the motivation of our model 3DCNN-CRF.

In this section, we briefly explain the working principles of this deep CRF employed in our framework. The deep CRF model was motivated by the work of Lin *et al.* [152] which works on color or grayscale images. We have significantly extended this model to cope with the spectral dimension of the data. During the training, CRF makes full use of the spatial contextual information in the process which is very relevant and useful in hyperspectral applications.

In this chapter, our proposed 3DCNN-CRF will further analyze the output obtained by the 3D-CNN. It is important to note that the output provided by the 3D-CNN is in the form of 3D feature maps whose individual location is defined by spatial co-ordinates along the spectral domain. We define these spectral-spatial locations as *voxels*. 3DCNN-CRF is capable of modelling these voxel-neighborhoods, therefore making it ideal for processing hyperspectral data. The parameters of the deep CRF used in our method were trained by stacks of CNNs applied on the initial feature map. However, instead of using group of bands, the 3D-CNNs used in the deep CRF consider the entire spectral channels together as the input to the network since the initial feature map is already a powerful representation of local spectral-spatial features.

The nodes in the CRF graph correspond to each voxel in the feature map along the B bands. The labels of the voxels are given by $l \in \mathbb{C}$. Later, edges are formed between the nodes which construct the pairwise connections between neighboring voxels in the CRF graph by connecting one node to all other neighboring nodes. The CRF model can be expressed as

$$P(l|v_{(d,\lambda)};\theta_\lambda) = \frac{1}{Z(v_{(d,\lambda)})} exp(-\mho(l, v_{(d,\lambda)};\theta_\lambda)) \tag{3.3}$$

where the network parameters θ along different wavelengths λ shall be learned. $\mho(l, v_{(d,\lambda)};\theta_\lambda)$ is the energy function that models how compatible the input voxel v is. v is defined by spatial co-ordinates $d = \{x, y\}$ along the spectral domain λ and is with a particular predicted label l. $Z(v_{(d,\lambda)}) = \sum exp[-\mho(l, v_{(d,\lambda)};\theta_\lambda)]$ is the partition function. In order to combine more useful contextual information, we should model the relationships between the voxels in the CRF graph. Therefore, the energy function can be expressed as

$$E(l, v_{(d,\lambda)};\theta_\lambda) = \sum_{\substack{p\in\mathbb{M}\\ \lambda\in\mathbb{B}}} \phi(l_p, v_p;\theta_\lambda) + \sum_{\substack{(p,q)\in\mathbb{E}\\ \lambda\in\mathbb{B}}} \psi(l_p, l_q, v_p, v_q;\theta_\lambda) \tag{3.4}$$

where $\mathbb{M}$ is the set of voxels or nodes, $\mathbb{B}$ is the set of spectral bands and $\mathbb{E}$ is the set of edges between the nodes in the CRF graph. Here ϕ is a unary potential function calculated for individual voxels, and ψ is a pairwise potential function determined based on the compatibility among adjacent voxels. In our method, each pixel with spatial coordinates $(x \pm 1, y, \lambda)$, $(x, y \pm 1, \lambda)$ or $(x, y, \lambda \pm 1)$ is connected to the pixel at (x, y, λ)

along the spectral dimension λ instead of connecting all other nodes in order to reduce the computational complexity. These 6-connected neighboring pixels are connected along one of the primary axes.

3.3.1 Unary Potential Functions

In our proposed 3DCNN-CRF, we apply stack of 3D-CNNs and generate feature maps and a fully connected layer to produce the final output of the unary potential at each individual voxel along λ. The stack of 3D-CNNs is applied on the node feature vectors, obtained from the initial feature map, to calculate the unary potential for each individual voxel representing nodes in the CRF graph.

The unary potential function ϕ is computed as follows:

$$\phi(l_p, v_p; \theta_\lambda) = -logP(l_p|v_p; \theta_\lambda) \tag{3.5}$$

During the deep CRF training, the network parameters θ_λ are adjusted in the stack of 3D-CNNs along the entire spectral channels as they no longer are partitioned into groups of bands.

3.3.2 Pairwise Potential Functions

The pairwise potential functions are calculated by considering the compatibility between the pair of voxels for all possible combinations (in our case, four adjacent voxels). As the first 3D-CNN applied to the original image gives us the feature vector for individual voxels in the feature map, the edge features can be formed by concatenating the feature vectors of two adjacent voxels [154]. Stack of 3D-CNNs are then applied on the edge feature vectors, which eventually gives us the pairwise potential output. The pairwise potential function is expressed as follows:

$$\psi(l_p, l_q, v_p, v_q; \theta_\lambda) = \varphi(v_p, v_q)\delta_{p,q,l_p,l_q}(f_p, f_q; \theta_\lambda) \tag{3.6}$$

Here $\varphi()$ is the label compatibility function which encodes the possibility of the voxel pair (v_p, v_q) being labeled as (l_p, l_q) by taking the possible combinations of pairs. δ_{p,q,l_p,l_q} is the output value of the 3D-CNNs applied to the pair of nodes that are described by the corresponding feature vectors f_p,f_q previously obtained by the initial 3D-CNN. θ_λ contains the 3D-CNN parameters to be learned for the pairwise potential function along the whole spectral channels λ.

Parameter estimation in CRFs can be performed by maximizing the log likelihood of a training input-output pair (v, l) as defined in Eq. (3.3) and Eq. (3.4). However, exact maximum-likelihood training for undirected graphical models is intractable as the computation involves marginal distribution calculation of the model. This is even more complex for conditional training when we are required to predict certain l given observed input voxel v. This eventually leads the decision of optimizing $P(l|v)$ instead of $p(l, v)$. Therefore, an efficient CRF training is desirable in order to reduce the computational complexities. An efficient CRF training method is described in the following Section 3.3.3.

3.3.3 Piecewise CRF Training

For the proposed CNN based CRF, the objective function for the CRF can be defined as

$$\nabla(\theta) = \sum_{\substack{p \in \mathbb{M} \\ (p,q) \in \mathbb{E} \times \mathbb{E} \\ \lambda \in \mathbb{B}}} \phi(l_p, v_p)\psi(l_p, l_q, v_p, v_q) - Z(v; \theta_\lambda) \tag{3.7}$$

Although such maximization of log-likelihood of (v, l) improves performance, the conditional training is expensive because the calculation of the log partition function $Z(v)$ depends on the model parameters as well as on the input voxels along the spectral channels. Therefore, estimating CRF parameters must include approximating the partition function for each iteration during the training phase in the stochastic gradient descent (SGD) method. This gets more complicated when a large number of iterative steps are required for SGD during the 3D-CNN training.

In order to efficiently train a large model, we can divide the entire model into pieces and then independently train those pieces. Later, we can combine the learned weights from those pieces and use it for testing purposes. This idea, known as piecewise training, was discussed in [155].

A proposition was defined and proved in [155] about the piecewise estimator that maximizes a lower bound on the true likelihood. It says:

$$Z(\mathbf{I}) \leq \sum_{e} Z(\mathbf{I}|_e) \tag{3.8}$$

Here, $\mathbf{I}|_e$ is the vector $\mathbf{I}$ with zeros in the entries that do not correspond to the edge e. Therefore, the piecewise objective function for CRF can be defined for a training input-output pair (v, l) as:

$$\nabla(\theta) = \sum_{\substack{p \in \mathbb{M} \\ (p,q) \in \mathbb{E} \times \mathbb{E} \\ \lambda \in \mathbb{B}}} \phi(l_p, v_p) \psi(l_p, l_q, v_p, v_q) - \sum_{(p,q) \in \mathbb{E} \times \mathbb{E}} Z(v; \theta_\lambda) \tag{3.9}$$

According to the proposition in Eq. (3.8), for each v, the bound needs to be applied separately which removes the requirement of marginal inference for gradient calculation. This idea can be incorporated into CRF training with 3D-CNN potentials. We can formulate $P(l|v)$ as a number of independent likelihoods on both unary and pairwise potentials

$$P(l|v; \theta_\lambda) = \prod_{\phi} \prod_{p \in \mathbb{M}} P_\phi(l_p|v; \theta) \prod_{\psi} \prod_{(p,q) \in \mathbb{E} \times \mathbb{E}} P_\psi(l_p, l_q|v; \theta_\lambda) \tag{3.10}$$

Both $P_\phi(l_p|v; \theta_\lambda)$ and $P_\psi(l_p, l_q|v; \theta_\lambda)$ can be calculated from unary and pairwise potentials respectively.

$$P_\phi(l_p|v; \lambda) = \frac{exp[\phi(l_p, v; \theta_\lambda)]}{\sum_{l_p} exp[\phi(l_p, v; \theta_\lambda)]} \tag{3.11}$$

$$P_\psi(l_p, l_q|v; \theta_\lambda) = \frac{exp[\psi(l_p, l_q, v; \theta_\lambda)]}{\sum_{l_p, l_q} exp[\psi(l_p, l_q, v; \theta_\lambda)]} \tag{3.12}$$

Therefore, it is not required to compute the partition function anymore and we only need to calculate the log likelihood of P_ϕ and P_ψ. As a result, the gradient calculation can be performed without partition function, thus saving expensive inference. After the

CRF training, we perform an inference on our model, for which a mean-field inference algorithm is adopted.

3.3.4 Mean-field Inference

In practice, due to large number of parameters contained in the CRF energy function for both unary and pairwise potentials, the exact minimization of CRF energy is nearly impossible. To this end, the mean-field approximation algorithm [153] is used to calculate the CRF distribution for maximum posterior marginal inference.

We use two Gaussian kernels that operate in the feature space defined by the intensity of voxel v at coordinates $d = \{x, y\}$ and the spectral band λ. We use those two-kernel potentials [153] defined in terms of the feature vectors f_p and f_q for two voxels v_p and v_q. The first term of this potential expresses the size and shape of the voxel-neighborhoods to encourage the homogeneous labels. The degree of this similarity is controlled by a parameter θ_α. This term is defined by

$$k^{(1)}(f_p, f_q) = w^{(1)} exp \left(- \sum_{d \in \{x,y\}} \frac{|v_{p,d} - v_{q,d}|^2}{2\theta^2_{\alpha,d}} \right) \tag{3.13}$$

This kernel is defined by two diagonal covariance matrices (one for each axis) whose elements are the parameters $\theta_{\alpha,d}$.

The second term of the potential is similar. Only an additional parameter ξ is used for interpreting how strong the homogeneous appearances of the voxels are in an area defined by spatial co-ordinates d across the spectral channels λ. It is defined by

$$\begin{aligned} k^{(2)}(f_p, f_q) = w^{(2)} exp(- \sum_{d \in \{x,y\}} \frac{|v_{p,d} - v_{q,d}|^2}{2\theta^2_{\alpha,d}} \\ - \sum_{\lambda \in \mathbb{B}} \frac{|v_{p,\lambda} - v_{q,\lambda}|^2}{2\theta^2_{\xi,\lambda}}) \end{aligned} \tag{3.14}$$

where $|v_{p,d} - v_{q,d}|$ is the spatial distance between voxels p and q and $|v_{p,\lambda} - v_{q,\lambda}|$ is their difference across the spectral domain. The influence of the unary and pairwise terms can be adjusted with their weights $w^{(1)}$ and $w^{(2)}$.

The first step of this iterative inference algorithm [153] is initialization in which a softmax function over the unary potential across all the labels for individual voxels is performed. The second step is message passing which applies convolutions with the two Gaussian kernels defined above on the current estimation of the prediction of the voxels. This reflects how strongly two voxels v_p, v_q are related to each other. By using back propagation, we calculate error derivatives on the filter responses. The next step is to take the weighted sum of the filter outputs for each label of the voxels. When each label is considered, it can be reformulated as the usual convolution of filter with input voxels and the output labels. Next, a compatibility transform step is performed followed by adding the original unary potential for each individual voxel obtained from the initial 3D-CNN. Finally, the normalization step of the iteration can be expressed as another softmax operation which gives us the final labels in the classification map. Algorithm 2 summarizes the important stages of our deep CRF approach.

Algorithm 2: Deep CRF

```
Input: 3D feature map obtained from Algorithm 1, $|\mathbb{M}|$ voxels in
$\{v_1, v_2, \ldots, v_{|\mathbb{M}|}\}$
for Each $v$ in $\mathbb{M}$ do
    Add $v$ in CRF graph
    for Each $v_i$,$v_j$ do
        if $v_i$ is adjacent to $v_j$ then
            Connect edge between $v_i$ and $v_j$ in CRF graph
        end if
    end for
end for
for Each $v$ in $\mathbb{M}$ do
    Compute unary potential function $\phi$ using Eq. (3.5)
end for
for Each $v_p$,$v_q$ in $\mathbb{E}$ do
    Compute pairwise potential function $\psi$ using Eq. (3.6)
end for
Train CRF using Eqs. (3.10), (3.11) and (3.12)
Compute two-kernel potentials using Eqs. (3.13) and (3.14)
Execute Mean-Field inference algorithm
Output: Classified Labels
```

Our proposed 3DCNN-CRF produces an improved classification map, hence, integrate the different 3D-CNN structures in the feature extraction and CRF steps into an end-to-end modelling approach. However, it suffers from low-resolution representation of inaccurate object boundaries due to repeated use of pooling layers during CNN training.

To overcome this problem, we further employ deconvolution network during the CRF pairwise potential computation and produce an improved output in the final stage. We do not include a deconvolution network in the first 3D-CNN because we propose to generate spectral-spatial features only from this stage and use these later to construct an end-to-end training scheme from input image to classification output. We present the basic formulation of deconvolution in Section 3.4.

3.4 Prediction Refinement

To obtain a high-resolution classification map from the mean-field inference, we add a deconvolution network [149] into our framework. This network is composed of deconvolution, unpooling, and rectified linear unit (ReLU) layers [151].

Unpooling

Pooling improves the classification performance by filtering noisy activations in the lower layers and retaining activations in the upper layers only. It can abstract activations in a receptive field with a single value. Unfortunately, spatial information within a receptive field is lost during pooling. As a result, accurate localization is not always possible. To overcome this problem, unpooling layers are used in the deconvolution network, which does the exact reverse operation of the pooling layers. During the CRF pairwise training, unpooling operation produces a finer resolution of an object by reconstructing the original size of the input data and thus restoring the detailed structures of the object of interest. Generally, unpooling operation keeps track of the locations of maximum activations which were selected during the pooling operation. This information can be very useful in placing the activations back to their original pooled location.

Deconvolution

The unpooling operation produces a large activation map which is not regular in nature. Although deconvolution operation is similar to convolution operations, it actually assigns a single input with multiple outputs unlike convolution operation which connects multiple inputs within a filter window or patch to a single activation value [109, 149]. This operation produces a much denser activation map compared to the sparse activation map obtained earlier. The filters used during deconvolution operation help in strengthening the activations that are close to the target classes and also suppressing the noisy activation from the regions containing different classes. As a result, different layers of the deconvolution network can help in reconstructing shapes in different levels. The filters used in lower layers may help in reconstructing the overall shape of an object while the higher layer filters can help in more class-specific details of an object. Therefore, more improved and accurate classification outcome can be obtained by the use of deconvolution network.

In our proposed 3DCNN-CRF, we incorporate deconvolution into the 3D-CNN training only during the deep CRF training. This is because we want to produce a final dense classification map with class-specific information in it instead of simply applying it on low-resolution activation map as a separate step. The integration of deconvolution during the pairwise potential calculation of the deep CRF training particularly helps in producing refined segments and hence, improves the classification accuracy to a large extent.

3.5 Experimental Results

In this section, we present the experimental results on real-world hyperspectral remote sensing images. Then we analyse the performance of the proposed method in comparison with several alternatives.

3.5.1 Hyperspectral Image Datasets

In the experiments, we used two widely used hyperspectral datasets, i.e., Indian Pines and Pavia University, in order to evaluate the effectiveness of our proposed method.

A New Dataset

For better evaluation of our proposed method, we created a new dataset by collecting AVIRIS images from the USGS database[1]. The details on the construction of this dataset are described in Section 3.5.2.

We separated our experiments on 3DCNN-CRF into two stages, 3D-CNN feature extraction and improved classification by deep CRF. For both tasks, we compared our results with state-of-the-art methods to evaluate the usefulness of our proposed algorithm. The details of our experiments will be presented later.

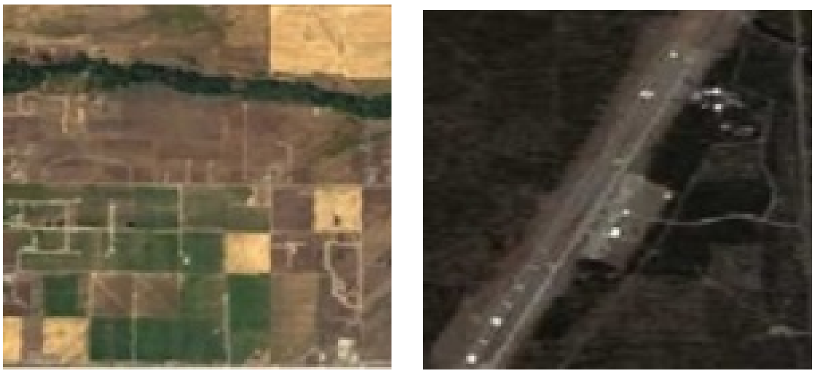


Figure 3.2: Image instances from the new dataset Griffith-USGS

3.5.2 Construction of the new dataset

In the official AVIRIS website[2], we downloaded remote sensing data located in the region of north America spanning over the United States of America, Canada and Mexico using

[1] https://earthexplorer.usgs.gov/

[2] https://aviris.jpl.nasa.gov/alt_locator/

a data acquisition tool. The AVIRIS sensor collects data that can be used for characterization of the Earth's surface and atmosphere from geometrically coherent spectroradiometric measurements. With proper calibration and correction for atmospheric effects, the measurements can be converted to ground reflectance data which then can be used for quantitative characterization of surface features.

We downloaded 19 unlabeled scenes to build the training and testing sets for deep learning. We cropped those scenes into a number of individual portions to build 150 training images and 50 testing images. As we captured scenes from multiple locations, the spatial resolutions of the scenes used in this dataset range from 2.4 to 18 meters. Each image consists of approximately 145×145 pixels and 200 spectral bands. Fig. 3.2 shows two instances from the training set of our new dataset *Griffith-USGS*.

3.5.3 Pre-processing

After collecting the AVIRIS image data, the following step was to undertake some preprocessing tasks in order to convert images into a suitable form for proper use. Hyperspectral sensors should be spectrally and radiometrically calibrated before data analysis. NASA/JPL has already processed the AVIRIS data to remove geometric and radiometric errors associated with the motion of the aircraft during data collection. However, the data should be further corrected for atmospheric effects and converted to surface reflectance. To do the conversion, we used a tool "FLAASH" [156] provided by ENVI which is a model-based radiative transfer program to convert radiance data to reflectance. Developed by Spectral Sciences, Inc., FLAASH uses MODTRAN4 radiation transfer code to correct images.

3.5.4 Manual Labelling

After obtaining the reflectance data, the next step was to create the training and testing datasets accordingly. As our method relies on a supervised training approach, it was important to construct a labeled set in order to fit into our proposed framework. For this purpose, we performed a pixelwise manual labeling on the images. To increase the size of the training set, we cropped smaller portions from the original image. We made sure that the cropped portion should contain instances of at least few classes that we want to classify. We created a training set containing six classes, including road, water, building, grass, tree and soil. We navigated to the exact corresponding locations of the high-resolution color images in Googe Earth to determine the accurate classes of the pixels and labeled our hyperspectral band images accordingly.

The widely used Pavia University or Indian Pines are too small to show the advantages of deep learning based methods. They are not challenging either as their latest classification accuracies are over 99%. The purpose of collecting a new dataset for our method is to introduce more challenges into the data and demonstrate the usefulness of our method. Although the number of classes that we identified is relatively small, the geographical locations had considerable impact on the entire dataset as the atmospheric effects varied significantly which produces different surface features. Those scenes varied in terms of resolution and contrast, and hence introduced more challenges in producing a refined classification map.

3.5.5 Design of the CNNs

As mentioned before, we used spectral-group based 3D-CNN to generate effective spectral-spatial features and train deep CRF for an improved classification outcome. However, to demonstrate the efficacy of such spectral-spatial features in obtaining better classification output, we generated an initial classification map from these features. Later, the final classification map is generated after integrating deep CRF which uses the feature maps rather than the classification map as the input. In this section, we elaborate the design of all the CNNs used in various stages of our framework.

For each CNNs used in our 3DCNN-CRF, we used 32 $5 \times 5 \times 5$ convolution kernels. Depending on the datasets and the two stages of our method, we adopted four to seven convolution layers and two to four pooling layers with 2×2 pooling kernel in each layer. The analysis on the selection of convolution layers is provided later. ReLU layers were used for all datasets as well. All layers were trained using backpropagation/SGD. The architecture of the CNNs used in our method is explained in Table 3.1.

Table 3.1: Architecture of the CNNs

Dataset	3D-CNN Feature Extraction		Deep CRF	
	Layer	Pooling	Layer	Pooling
Indian Pines	1	2 × 2	1	2 × 2
	2	2 × 2	2	No
	3	No	3	No
	4	No	4	2 × 2
	5	2 × 2		
	6	No		
	7	2 × 2		
Pavia University	1	2 × 2	1	2 × 2
	2	2 × 2	2	No
	3	No	3	2 × 2
	4	No		
	5	2 × 2		
	6	No		
Griffith-USGS	1	2 × 2	1	2 × 2
	2	2 × 2	2	No
	3	No	3	No
	4	No		
	5	2 × 2		
	6	No		

The size of the mini-batch was set to 100. For the logistic regression, the learning rate was set to 0.003 for Indian Pines, 0.01 for Pavia University and 0.005 for our new dataset. The number of epochs was 700 in 3D-CNN feature extraction and 500 in deep CRF for Indian Pines, 600 in 3D-CNN feature extraction and 500 in deep CRF for Pavia University and for Griffith-USGS. The weights were randomly initialized and gradually trained using the back propagation algorithm. Each convolution kernel extracted distinct features from the input that convey meaningful structural information about the data. Different kernels used in the convolution layers are able to extract different features on the way to form a powerful representation.

3.5.6 Results and Comparisons

Effective spectral-spatial structure representation of the hyperspectral data provides useful input for classification. These initial inputs guide the subsequent process of constructing deep CRF in improving the classification output. It is, in fact, a common practice to use spectral-spatial features to perform an initial classification which is later used in an additional step [81, 157] to improve the classification output. Hence, we generated an initial classification map in order to demonstrate the effectiveness of spectral band group-based spectral-spatial features in producing better classification. Furthermore, we intended to validate the usefulness of our proposed 3DCNN-CRF model by improving from the initial classification.

To begin with, we evaluated the effectiveness of the first part of our framework which includes the execution of CNN over the image that eventually results in the spectral-spatial structure followed by an initial classification step. For this purpose, we compared our method with an SVM-based classification algorithm [102] which itself is divided into two parts: (1) SVM with composite kernel (SVM-CK) and (2) SVM with generalized composite kernel (SVM-GCK). We compared our results with SVM-GCK as it outperformed its counterpart SVM-CK.

We also compared our method with a spatial-spectral-based method (MPM-LBP-AL) [139]. In this method, active learning (AL) and loopy belief propagation (LBP) algorithms were used to learn spectral and spatial information simultaneously. Then the marginal probability distribution were exploited, which used the whole information in the hyperspectral data. We made comparisons with another supervised method (MLRsubMLL) [81] that integrated spectral and spatial information into a Bayesian framework. In this method, a multinomial logistic regression (MLR) algorithm was used to learn the posterior probability distributions from the spectral information. Moreover, a subspace projection method was used to characterize noisy and mixed pixels. Later, spatial information was added using a multilevel logistic MRF prior. Along with this, we also reported the performance of a recent work developed by Chen *et al.* [119] who proposed classification methods based on 1-D, 2-D and 3-D CNNs. To fit into our method, we simply compared with their 1-D CNN (1D-CNN-LR) and 3-D CNN (3D-CNN-LR) approaches which used logistic regression (LR) to classify pixels.

We chose limited samples for training since we wanted to simulate the real-world cases where the size of labelled data is small. For our experiments, we chose 3 samples per class in the extreme case and continued investigating on different numbers of training samples per class, from 5 to 15. To improve the classification performance and to avoid overfitting

Table 3.2: Number of Training and Testing Samples Distribution for Each Class on Indian Pines Dataset

Class	Training Samples		Real Testing Samples
	Real Samples	Virtual Samples	
Alfalfa	15	33	31
Corn-notill	15	1234	1411
Corn-mintill	15	844	813
Corn	15	238	220
Grass-pasture	15	371	466
Grass-trees	15	675	713
Grass-pasture-mowed	15	18	13
Hay-windrowed	15	402	461
Oats	15	51	5
Soybean-notill	15	991	955
Soybean-mintill	15	1921	2438
Soybean-clean	15	611	576
Wheat	15	193	188
Woods	15	1033	1248
Buildings-Grass-Trees-Drives	15	301	369
Stone-Steel-Towers	15	73	76
Total	9229		9983

problem, we increased the size of the training samples by augmentation discussed in Section 3.2.1. We used those limited real samples for augmenting the training set and the rest of the real samples were included in the testing set.

The results that we report in this chapter are based on a training set in which we performed a 10-fold cross validation to select 15 real samples and 50% of the total number of augmented samples for each class. The reason we did not consider all augmented samples in each iteration of cross validation because samples with good/bad representations may affect the classification performance to a significant extent. We report the total number of training and testing samples for each class used on the three datasets in Tables 3.2, 3.3 and 3.4. Also, during the CNN training, we used 90% of the total number of training samples, consisting of limited number of real samples and augmented samples, to learn the weights and biases of the neurons and the remaining 10% to validate and further update the design of the architecture.

For performance evaluation, we calculated the overall accuracy (OA) and average accuracy (AA) with the corresponding standard deviations. We repeated our experiments for ten times over the randomly split training and testing data. Furthermore, we assessed

Table 3.3: Number of Training and Testing Samples Distribution for Each Class on Pavia University Dataset

Class	Training Samples		Real Testing Samples
	Real Samples	Virtual Samples	
Asphalt	15	5011	6614
Meadows	15	14311	18632
Gravel	15	1904	2082
Trees	15	3055	3047
Painted metal sheets	15	998	1328
Bare Soil	15	4598	5012
Bitumen	15	1072	1313
Self-Blocking Bricks	15	3440	3665
Shadows	15	970	930
Total	35494		52606

Table 3.4: Number of Training and Testing Samples Distribution for Each Class on Griffith-USGS Dataset

Class	Training Samples		Real Testing Samples
	Real Samples	Virtual Samples	
Road	15	2699	1719
Water	15	3594	1559
Building	15	2512	1466
Grass	15	3936	1902
Tree	15	2897	1757
Soil	15	2543	1401
Total	18271		9804

the statistical significance of our results by applying binomial test in which the assessment was done by computing the p-value from the paired t-test. We set the confidence interval to 95% which declares statistical significance at $p < .05$ level.

Table 3.5 reports the pixelwise CNN-based classification results on Indian Pines, Pavia University and Griffith-USGS datasets with 15 real samples per class and augmented samples for training. The results show that our method achieved similar accuracy as 1D-CNN-LR [119]. Both methods outperformed other pixel-wise classification methods and were statistically significant in most cases. Therefore, we can conclude that the CNN-based approaches can effectively improve the classification accuracy.

As described before, we proposed to perform CNN-based classification on spectral groups instead of pixel-wise classification. Table 3.6 reports 3-D CNN-based classification results on Indian Pines, Pavia University and Griffith-USGS datasets with 15 samples per class

Table 3.5: Classification accuracies on different datasets (pixelwise). A '*' denotes that the best average accuracy (shown in bold) is significantly better that the accuracy achieved by the corresponding method according to a statistical paired t-test for comparing classifiers

Dataset		SVM-GCK [102]	MPM-LBP-AL [139]	MLR*sub*MLL [81]	1D-CNN-LR [119]	Proposed Method
Indian	OA (%)	87.53 ± 2.30	90.07 ± 1.76	85.06 ± 1.92	**92.93 ± 1.44**	92.59 ± 0.55
Pines	AA (%)	88.97 ± 0.54*	90.01 ± 0.77	86.00 ± 1.09*	**93.05 ± 2.14**	92.96 ± 1.01
Pavia	OA (%)	89.39 ± 2.19	84.70 ± 1.22	87.97 ± 1.54	**92.35 ± 1.08**	92.06 ±1.36
University	AA (%)	91.98 ± 1.23	85.97 ± 0.07*	89.31 ± 0.77*	93.17 ± 1.26	**93.97 ± 0.30**
Griffith-	OA (%)	67.33 ± 2.71	68.69 ± 0.91	68.05 ± 0.19	75.07 ± 1.23	**75.97 ± 0.19**
USGS	AA (%)	70.45 ± 1.49*	69.33 ± 1.01*	69.02 ± 0.77*	75.98± 1.30*	**76.42 ± 0.83**

Table 3.6: Classification accuracies on different Datasets (spectral groups). A '*' denotes that the best average accuracy (shown in bold) is significantly better than the accuracy achieved by the corresponding method according to a statistical paired t-test for comparing classifiers.

Dataset		SVM-GCK [102]	MPM-LBP-AL [139]	MLR*sub*MLL [81]	3D-CNN-LR [119]	Proposed Method
Indian	OA (%)	90.70 ± 1.35	92.20 ± 1.82	90.66 ± 0.20	97.88 ± 0.48	**98.29 ± 0.33**
Pines	AA (%)	90.83 ± 0.32*	92.18 ± 1.21*	89.91 ± 2.30*	99.18 ± 0.06	**99.20 ± 0.09**
Pavia	OA (%)	96.14 ± 2.19	87.25 ± 1.26	93.91 ± 1.44	98.60 ± 0.07	**99.12 ± 0.41**
University	AA (%)	96.05 ± 0.11	89.09 ± 0.08*	92.00 ± 1.04*	99.53 ± 0.05	**99.69 ± 0.03**
Griffith-	OA (%)	73.97 ± 1.21	63.19 ± 1.99	68.88 ± 1.45	77.71 ± 0.87	**83.05 ± 1.19**
USGS	AA (%)	74.97 ± 0.46*	65.02 ± 0.97*	69.95 ± 1.45*	78.95 ± 0.37*	**84.98 ± 0.86**

Table 3.7: Classification accuracies on different datasets.

Dataset		Without Deconvolution	Deconvolution in Unary CRF	Deconvolution in Pairwise CRF
Indian Pines	OA (%)	98.38 ± 0.37	99.04 ± 0.03	**99.15 ± 0.16**
	AA (%)	99.29 ± 0.24	99.35 ± 0.10	**99.41 ± 0.04**
Pavia University	OA (%)	99.32 ± 0.03	99.23 ± 0.13	**99.63 ± 0.07**
	AA (%)	99.70 ± 0.05	99.53 ± 0.06	**99.79 ± 0.03**
Griffith-USGS	OA (%)	84.91 ± 1.02	86.00 ± 0.29	**88.92 ± 0.17**
	AA (%)	84.55 ± 1.36*	85.05 ± 1.05*	**89.13 ± 0.74**

and augmented samples for training. By keeping the CNN parameters the same as in the pixel-based classification experiments, it is evident from the results in Tables 3.5 and 3.6 that the classification accuracy can be significantly improved with spectral-group based representation. The reason is that 3D operation better characterizes the spatial and structural properties of the hyperspectral data. Both our method and 3D-CNN-LR [119] outperform the rest of the methods, showing the power of deep neural

Table 3.8: Improved classification accuracies on different Datasets. A '*' denotes that the best average accuracy (shown in bold) is significantly better than the accuracy achieved by the corresponding method according to a statistical paired t-test for comparing classifiers.

Dataset		MPM-LBP-AL [139]	MLRsubMLL [81]	WHED [158]	3D-CNN-LR [119]	Proposed Method
Indian Pines	OA (%)	92.91 ± 1.24	91.85 ± 0.83	90.15 ± 1.95	98.25 ± 0.78	**99.15 ± 0.16**
	AA (%)	92.35 ± 1.90*	91.95 ± 0.74*	90.85 ± 2.05*	99.27 ± 0.12	**99.41 ± 0.04**
Pavia University	OA (%)	92.19 ± 0.50	94.77 ± 1.09	87.85 ± 1.75	98.80 ± 0.28	**99.63 ± 0.07**
	AA (%)	93.85 ± 0.16*	95.35 ± 0.71*	86.50 ± 2.56*	99.60 ± 0.07	**99.79 ± 0.03**
Griffith-USGS	OA (%)	69.89 ± 1.47	74.29 ± 0.66	70.20 ± 2.33	82.53 ± 0.69	**88.92 ± 0.17**
	AA (%)	70.19 ± 2.47*	75.06 ± 1.20*	71.85 ± 2.50*	83.04 ± 0.91*	**89.13 ± 0.74**

networks. Our method significantly outperforms 3D-CNN-LR [119] on Griffith-USGS dataset, which proves the usefulness of the proposed paradigm.

Since it is a common practice to compare the accuracy of an initial classification and improved classification by an additonal step in hyperspectral remote sensing [81, 139], we evaluated the effectiveness of the later stages of our framework with MLR*sub*MLL [81] and WHED [158] which included explicit segmentation stages as additional steps. Similarly, we further report the improved accuracy of 3D-CNN-LR [119] since this method included L2 regularization and dropout in the training process to improve the initial coarse classification results. We also report the final accuracies obtained by MPM-LBP-AL [139] in which active learning was used in the later stages of their algorithm to improve the accuracy previously obtained by estimating the marginal inference for the whole image. We tested the usefulness of deconvolution by (1) running the method without using deconvolution at all, (2) using deconvolution during CRF unary potential calculation and (3) using deconvolution during CRF pairwise potential calculation stage. It is important to note that we prefer to include deconvolution into pairwise potential calculation stage as this step plays a major role in constructing accurate segments by connecting regions that actually belong to the same segment. Therefore, we applied deconvolution in the deep pairwise potential calculation stage rather than using it in other stages.

Table 3.8 reports the improved classification accuracies on three datasets respectively. The results show that our proposed 3DCNN-CRF outperforms the methods MLR*sub*MLL [81], MPM-LBP-AL [139], 3D-CNN-LR [119] and WHED [158]. The integrated 3DCNN-based pairwise potentials defined on both spatial and spectral dimensions significantly improved the coarse-level prediction rather than doing local smoothness. During our experiments, we observed that the classification map produced by the initial CNN was too coarse for the Griffith-USGS dataset since we collected images from different scenes. Those scenes varied significantly in terms of resolution and contrast, and hence introduced more challenges in producing an improved classification map. After integrating CRF potentials, an approximately 7% increase in accuracy was observed (Table IV), leading to significant advantages over the baseline methods. Deconvolution network is capable of improving of the final output, particularly when it is used during the pairwise potential calculation, by effectively improving the accuracy on connecting regions that belong to the same segment. The idea of integrating deconvolution into pairwise potential computation was supported by the results this option outperforms the other two versions where deconvolution was not used at all and was used in calculating unary potentials.

Fig. 3.3 illustrates the classification results from the two stages of our method on the Indian Pines, Pavia University and Griffith-USGS datasets respectively. The first and the second columns are the ground truth and the initial classification map on each dataset generated by 3D-CNN, respectively. The third column contains binary error maps obtained by comparing the classification results with the ground truth. The white pixels indicate the parts of the image that are incorrectly classified. The fourth column shows the final improved classification outcome using deep CRF and deconvolution, whose error maps are presented in the last column. The differences between the binary maps represented in columns (c) and (e) show that the number of incorrectly classified pixels are significantly decreased after introducing the deep CRF. This suggests the usefulness of 3DCNN-CRF for classifying hyperspectral images.

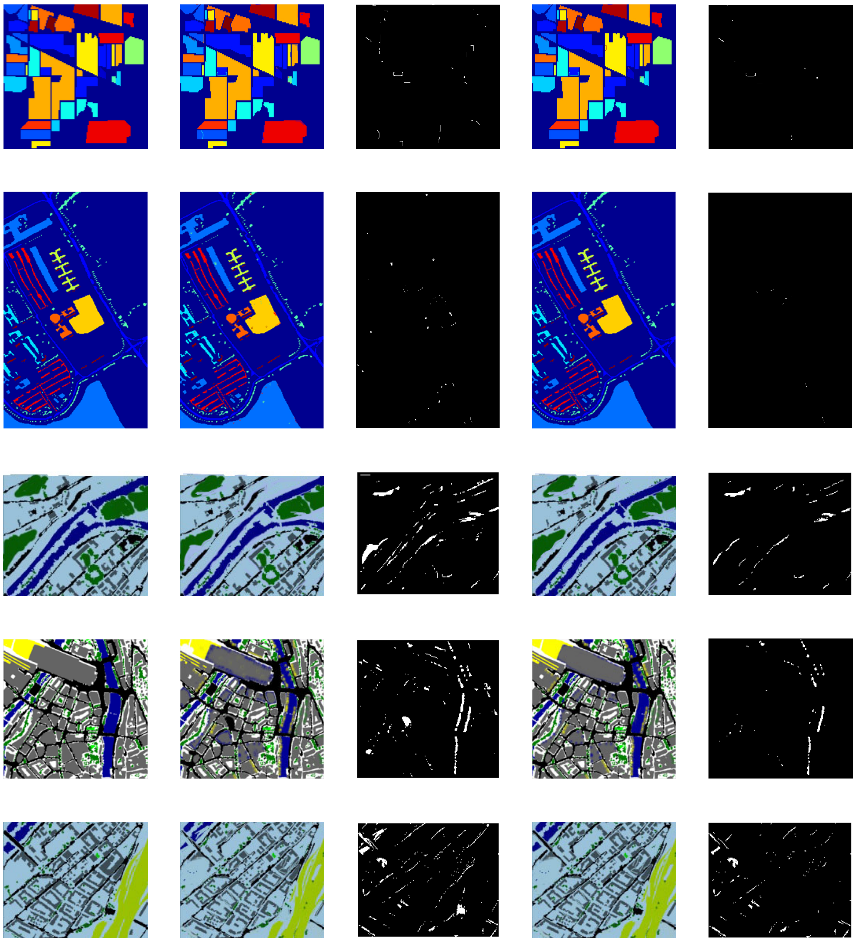


Figure 3.3: (a) Ground truth; (b) 3D-CNN-based classification; (c) difference map with ground truth; (d) improved classification by deep CRF with deconvolution; and (e) difference map after final classification.

3.5.7 Performance Analysis and Parameter Settings

Effect of Few Spectral Bands

In our algorithm, we propose to use spectral groups instead of the whole spectrum to construct the spectral-spatial representation of our hyperspectral data during the first stage. This better characterizes a range of spectral variability among the entire spectral signature of the data. Although during the training of CNNs, the large number of spectral groups results in a large number of feature maps, these are able to capture the local image information well and precisely, as well as can contribute to describing the underlying materials which are very significant for classifying hyperspectral images. Furthermore, we augmented training samples from these band groups which (1) increased the size of the training samples significantly and (2) generated more effective spectral-spatial representation of samples from different wavelengths. During the experiments, we observed that with small spectral groups, we were able to detect a wider range of spectral information from our input data and hence we achieved better classification accuracy than that of using the entire spectrum. Moreover, we also analysed on the number of optimal bands to be added in individual spectral groups by connecting this step with the data augmentation process. Since different wavelength groups capture different underlying material information, we chose the size of the spectral groups by testing on a various number of bands and measuring the corresponding accuracies. We discovered that the optimal number of bands should be 25 for Indian Pines and Pavia University, and 20 for Griffith-USGS. Table 3.9 shows the relative comparison between these two settings of using spectral groups for the initial classification by 3D-CNN. The analysis of choosing the optimal number of bands for respective datasets is given in Fig. 3.4 where the whole spectral cube denotes all the bands used for representing the spectral cube and smaller spectral groups denote the smaller number of bands in the spectral groups.

Table 3.9: Effect of fewer spectral groups

Dataset	Accuracy (%) with Whole Spectral Cube	Accuracy (%) with Smaller Spectral Groups
Indian Pines	96.80 ± 1.01	**99.20 ± 0.09**
Pavia University	97.80 ± 1.05	**99.69 ± 0.03**
Griffith-USGS	79.65 ± 0.65	**84.98 ± 0.86**

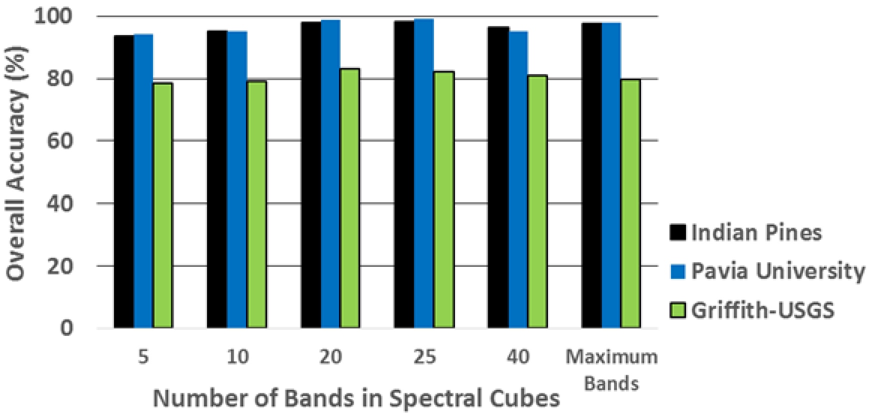


Figure 3.4: Classification results of different numbers of bands in spectral groups.

Effect of Data Augmentation

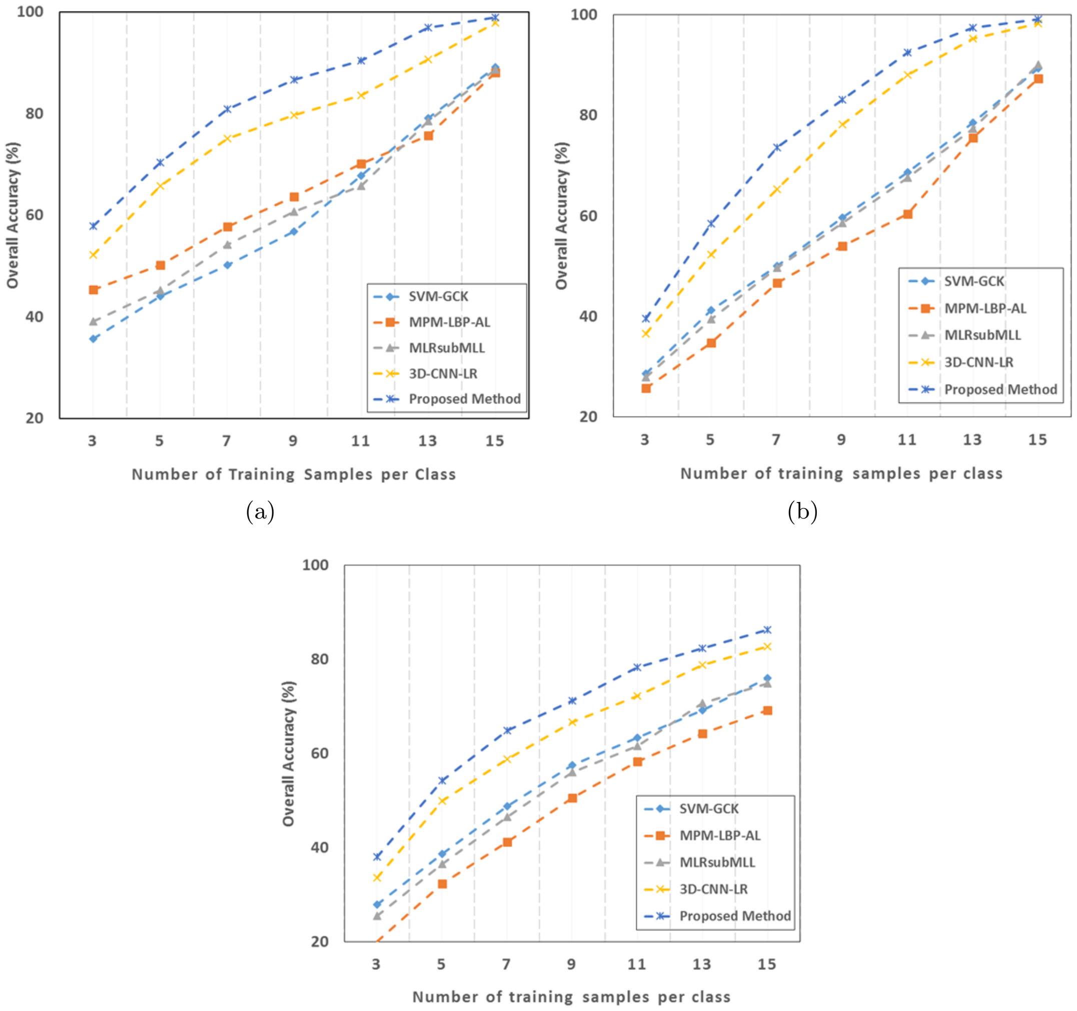
During the experiments, we chose different numbers of training samples and augmented the size accordingly. We observed that increasing the numbers of training samples from different band groups had evidently improved the overall performance of our algorithm. Moreover, we also tested other methods with the same experimental settings and noticed the improved performance achieved by those as well. Fig. 3.5 show the effect of various number of training samples which were used in data augmentation in the overall classification accuracy computed from different spectral groups for all three datasets respectively. Comparisons with other baseline methods for the various number of training samples were also demonstrated in Fig. 3.5. We also reported the overall accuracies obtained by our proposed method with and without augmenting data in Table 3.10 for all three datasets experimented. We observed that, the accuracy improved by almost 35% when only 3 samples were used for training the CNNs. It is quite evident from this analysis that the data augmentation had eventually contributed in improving the performance of the CNNs when limited training data are available.

Table 3.10: Effect of Data Augmentation on OA (%) for all three datasets.

No. of Training	Indian Pines		Pavia University		Griffith-USGS	
Samples/Class	Without Augmenting	With Augmenting	Without Augmenting	With Augmenting	Without Augmenting	With Augmenting
3	19.91	57.89	13.66	39.55	14.25	37.95
5	27.57	70.34	22.58	58.44	18.30	54.21
7	30.63	80.91	29.95	73.65	25.81	64.93
9	35.28	86.60	37.05	83.12	30.05	71.16
11	39.05	90.48	41.15	92.45	33.80	78.35
13	43.85	96.87	43.75	97.41	39.44	82.38
15	47.90	98.89	46.65	99.12	43.25	86.25

Effect of Depth in CNNs

An important observation can be made from the results reported earlier in terms of the depth of the networks. Undoubtedly, depth helps in improving the classification accuracy but adding too many layers introduces the curse of overfitting and may also downgrade the accuracy as well. It is widely accepted that minimizing both training and



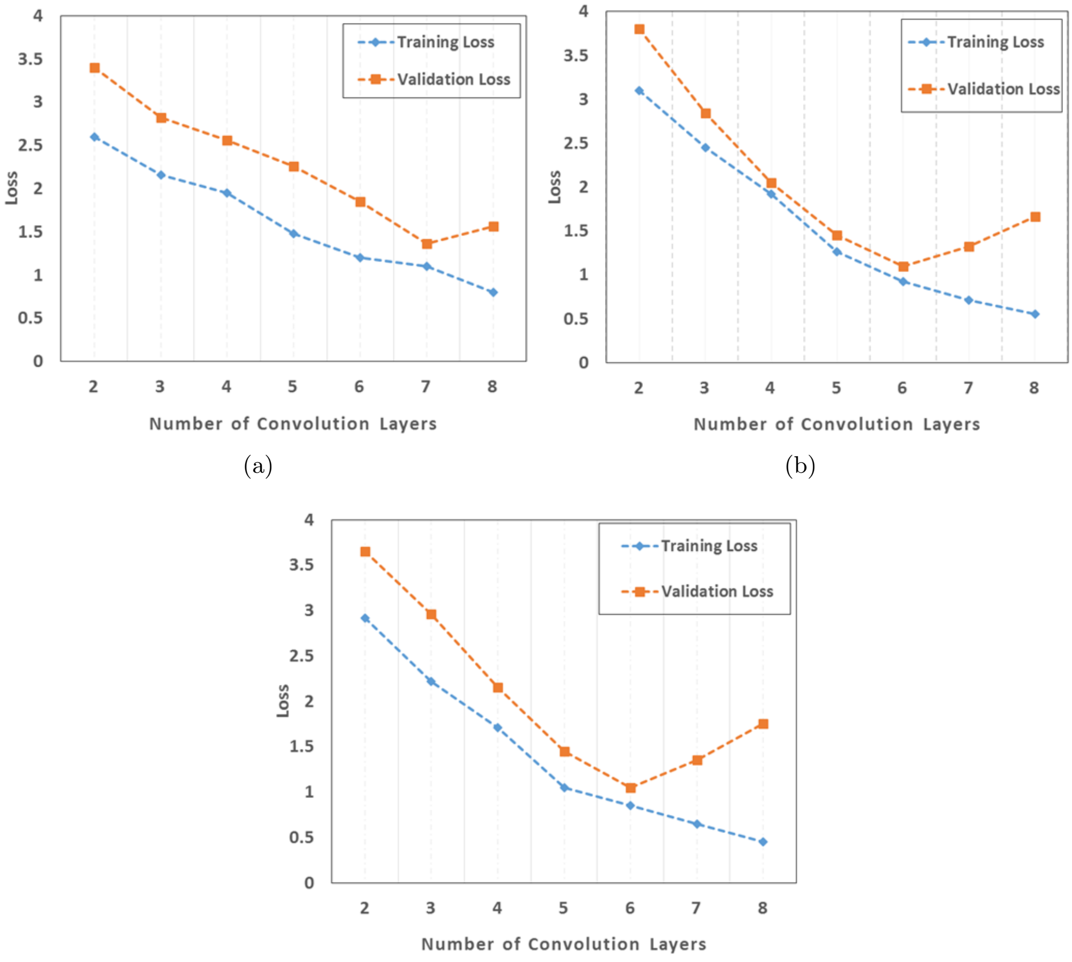
(a)

(b)

(c)

FIGURE 3.5: OAs with different numbers of training samples/class in all baseline methods for (a) Indian Pines, (b) Pavia University, and (c) Griffith-USGS.

validation losses are important in a well trained network. If the training loss is small and the validation loss is large, it means that the network is overfitted and will not generalize well for the testing samples. Therefore, we optimized the CNNs using trial and error approach and determined the number of nodes in the hidden layers, learning rate, kernel size and number of convolution layers. During our experiments, we started with a small number of convolution layers and gradually increase it and monitored the training and validation losses with the changing number of layers. The effect of depth of the convolution layers for the initial classification stage of our algorithm is illustrated in Fig. 3.6 for the three datasets experimented.



(a)

(b)

(c)

FIGURE 3.6: Training and validation losses for (a) Indian Pines, (b) Pavia University, and (c) Griffith-USGS.

Influence of Spatial Size of Kernels

The spatial size of kernels also plays an important role in the final classification performance. Small receptive fields of convolution kernels generally result in better performance because in this way it is possible to learn finer details from the input. During our experiments, we varied the spatial size of the kernels between three to nine. Fig. 3.7 shows that $5 \times 5 \times L$ is an optimal size for all the three datasets, where L is the number of spectral bands in each band group. We found that a larger size kernel such as $9 \times 9 \times L$ ignored and skipped some essential details in the images. On the other hand, a smaller size kernel such as $3 \times 3 \times L$ provided overly detailed local information and therefore, created confusions in classification eventually. Hence, the determination of an optimal size of the kernel is important in finding the most discriminative features for classification.

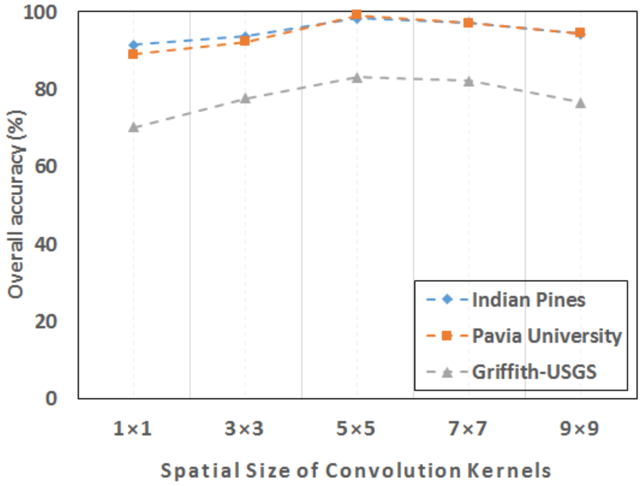


Figure 3.7: Effect of spatial size in convolution kernels

Influence of ReLU

Compared to sigmoid functions, ReLU obtains better performance in terms of both complexity and accuracy [159] (shown in Table 3.11). According to our experiments, we found out that with ReLU, we achieved convergence faster than sigmoid function. For Griffith-USGS, CNN with ReLU reaches convergence almost two times faster than the same network with sigmoid. Performance was also consistently better for the other two datasets with ReLU. Furthermore, the models with ReLU can lead to lower training error at the end of training.

Table 3.11: Effect of ReLU

Dataset	Accuracy (%)		Runtime (in minutes)	
	Sigmoid	ReLU	Sigmoid	ReLU
Indian Pines	98.15	**99.41**	57	**36**
Pavia University	99.04	**99.79**	77	**49**
Griffith-USGS	85.97	**89.13**	912	**512**

3.5.8 Analysis of Computation Cost

Here, we calculate the computational cost of classifying an image with our trained model. The total cost of 3DCNN-CRF combines the computational complexities for (1) generating spectral-spatial features by CNN and (2) improving classification performance by deep CRF. Generally, the convolution operations impose a significant time constraint on the time complexity of 3D-CNN which is computed in terms of the number of convolution layers, number and size of kernels and size of the intermediate feature maps [160]. The generated feature map by 3D-CNN is formulated as a CRF graph in which the voxels are represented as individual nodes. Therefore, the time complexity of CRF is computed in terms of the number of edges between the nodes as well as the size of the label set, which is quadratic in general. However, the use of highly efficient approximations for high-dimensional filtering during the message passing of mean field inference algorithm reduced the time complexity to linear in the number of labels and in the number of edges in the CRF model [153]. Hence, the total time complexity of our algorithm is given by:

$$O\left(\sum_{l=1}^{D} K_{l-1}.R_l^2.K_l.d_l^2\right) + O(E.Y) \tag{3.15}$$

Here, l is the current convolutional layer, D is the number of convolutional layers, K_l is the number of kernels in the l-th layer, K_{l-1} is also known as the number of input channels in the l-th layer, R_l is the spatial size of the kernel and d_l is the spatial size of the intermediate feature maps. E is the number of edges in the CRF graph formulated from the initial CNN and Y is the size of the label set.

We compared the testing time for all methods included in the experiments. Since the baseline methods used in our experiments were implemented on CPU, therefore, for a fair comparison, we also chose to run our algorithm on CPU instead of GPU that is widely used for deep learning approaches. All methods were implemented in Matlab and few C modules, and run on a desktop computer with Intel Core i5-4570 @ 3.2GHz 8G memory, with a Windows 7 system. The results are shown in Table 3.12. The testing stage of the deep learning algorithms is very fast and is close to the time required by other baseline methods. This is an important property for real applications as the model training can be undertaken offline but the application of the trained model on new data has higher efficiency requirements.

Table 3.12: Running time comparison (measured in minutes)

Methods	Dataset	Testing Time
MPM-LBP-AL [139]	Indian Pines	0.63
	Pavia University	0.91
	Griffith-USGS	0.93
MLRsubMLL [81]	Indian Pines	0.58
	Pavia University	0.88
	Griffith-USGS	0.91
WHED [158]	Indian Pines	0.77
	Pavia University	1.14
	Griffith-USGS	1.15
3D-CNN-LR [119]	Indian Pines	0.91
	Pavia University	1.12
	Griffith-USGS	1.09
Proposed Method	Indian Pines	0.79
	Pavia University	1.05
	Griffith-USGS	1.08

3.6 Conclusions

In this chapter, we presented an efficient CRF-CNN based deep learning algorithm for classifying hyperspectral images. To utilize the full strength of deep models for complex computer vision tasks, we constructed a powerful spatial-spectral representation of hyperspectral data. We applied 3D-CNN in a range of more effective spectral-spatial representative band groups to extract initial features. To further facilitate the classification task, we integrated CRF with 3D-CNN into an end-to-end framework in which the parameters of CRF were calculated using CNN, therefore making it a deep CRF. The initial prediction results coming from this 3DCNN-CRF architecture was further improved by using a deconvolution block inside of the CRF pairwise potential calculations. Moreover, to overcome the problem of over-fitting, we employed data augmentation techniques and increased the size of training samples for training the CNNs. This effectively improved the overall performance of our deep network to a significant extent.

In summary, to achieve the improvement of the hyperspectral image classification performance, our proposed 3DCNN-CRF architecture contains several important efficient stages that not only optimize the calculations of such computationally expensive task but also improved the initial prediction results obtained by the initial CNN algorithm. With 3DCNN-CRF, we can fully exploit the usefulness of CRF in the context of classification by integrating it completely inside of a deep learning algorithm. We further evaluated the usefulness of our method by comparing it with several state-of-the-art methods and achieved promising results.

Chapter 4

Combining Unmixing and Deep Feature Learning for Hyperspectral Image Classification

In this chapter, we propose an integrated method which combines unmixing results into a deep feature learning model in order to classify hyperspectral data. The model generates superpixels from abundance estimations of the underlying materials of the image and provides these abundance-guided information as features to a deep model. Our proposed deep model is formulated as an RNN. It receives significant spectral-spatial information in the data to produce better and powerful features so as to achieve improved classification performance.

4.1 Introduction

One of the critical challenges of HSI processing is the problem of mixed pixels. Usually, when the spectral resolution increases, the spatial resolution decreases. In case of high altitude sensors covering wide areas, low spatial resolution is a common problem [161].

These limitations significantly affect the performance of methods used to analyze and process HSI data. In particular, classification tasks suffer greatly due to the problem of mixed pixels in which case a pixel may contain more than one material/class. The combination of mixed and pure pixels also happen in high spatial resolution images [162].

The notion of a pure material can be application dependent. Suppose an HSI contains materials such as bricks, roads, water, plants, soil, cement, which gives a general assumption of the presence of six classes. However, if the percentage of pixels covered by cement is comparatively too small, then it may not be necessary to define an independent class for cement. It also depends on if the estimates on the proportion of cement is indeed required. Similarly, if the application demands distinguishing between two types of plants, then two plant classes must be created based on their spectral signatures. Fortunately, by applying unmixing process, pixels in the spectra can be decomposed into a collection of spectral signatures, called *endmembers*, directly from data without much prior knowledge. These endmembers are represented as a set of fractional estimations called *abundances*. Among different unmixing techniques, methods based on linear mixing model considers each pixel as a linear combination of endmembers.

Linear mixture models can be further divided into three categories: geometry, sparse regression and statistics based. Geometry based methods [163] exploit the geometric relationships among the endmembers and estimate their abundances. Sparsity constraint is often applied to sparse regression methods [164] to select few endmembers with high variance of material reflectance. Spatial regularization term can also be included in the spectral domain using prior information about the endmembers to unmix the data [165]. Statistical unmixing methods such as NMF [166] decompose the image into nonnegative endmember and abundance matrices. Various constraints such as sparsity [167], combination of spectral and spatial constraints [168] for smoothing of endmembers or maintaining manifold structure of unmixed data [169] have been applied.

Along with pixel based unmixing methods, region based methods [170] have also been developed. It is quite interesting to explore the relationships among the regions in an HSI because there is a high possibility that the same endmembers may appear in a local neighborhood or in other homogeneous regions. Therefore, it is important to not only consider the consistency of abundance of those endmembers in a homogeneous region but also discriminate the contributions of those endmembers from other regions across the image. To address this property, Tong *et al.* [171] proposed a novel region based NMF (R-NMF) method that forces consistent abundances within each homogeneous region and also separates the contribution from endmembers among the regions.

Using unmixing results to improve classification performance has also been investigated in the literature where unmixing has been used as a dimensionality reduction process and later classify images [172]. Villa *et al.* [173] proposed a semi-supervised method which uses linear spectral unmixing method to label data samples for classification. In a supervised classification method [174], probabilistic SVM was used to get preliminary classification map so that mixed pixels can be identified. Then a spectral unmixing method based on fully constrained least squares (FCLS) method was adopted to solve sub-pixel mixing problem in the final classification map. A similar approach was also discussed in [175] where the mixture-tuned matched filtering (MTMF) method was used to get the abundance map which was used as the input to the classification step. All these approaches treated the whole spectral information together without investigating further into the contribution from specific wavelength ranges or band groups in the unmixing step. Therefore, abundance information from different range of wavelengths has not been sufficiently explored and utilized to benefit the classification step.

In addition to the existing spectral-spatial classification methods, superpixel techniques have also been used to extract spatial information in the feature extraction process [176]. Superpixels consist of regions containing a set of adjacent pixels that share spectral similarities. The advantage of adopting superpixel-based methods for classification is that the pixels in the same superpixel are more likely to belong to the same class. In this context, it is highly likely that the use of unmixing results can play an important role in generating more effective superpixels. Hence, it would be interesting to investigate the potential of providing such additional information to a deep model to better characterize the data.

In this chapter, we propose to utilize unmixing results in the form of abundance matrices as the input to our model for classifying HSI data. We use a region based NMF method for structure consistency and preservation during the unmixing step [171]. We argue that the derived abundance matrices contain significant information about the underlying endmembers in the regions for classification. Furthermore, instead of providing only one instance of abundance matrix for each endmember, we intend to provide multiple instances of abundance matrices which will be obtained from different band groups along the spectral channel. Such representation is very useful for deep learning framework as the model will be able to receive sufficient information to identify interesting patterns within the data. In this way, the problem of limited training samples of hyperspectral data can be resolved to a significant extent which can facilitate the powerful classification ability of deep learning models. To the best of our knowledge, this is the first work that combines abundance maps across different wavelength groups as an unmixing output with deep feature learning for HSI classification.

In our method, we first obtain abundance matrices for each pixel along the group of bands of an HSI. Then we extend the simple linear iterative clustering (SLIC) algorithm to generate superpixels by utilizing the abundance estimations obtained from the unmixing algorithm. We further introduce KD-estimated PDF to describe the spectral distribution of the superpixels. After that we provide these abundance information-guided superpixels as input to an integrated CNN-CRF model for performing the final classification. In our approach, we formulate mean-field approximate inference for the dense CRF with Gaussian pairwise potentials as an RNN to improve the coarse outputs obtained from a traditional CNN. Our framework is illustrated in Fig. 4.1.

The rest of this chapter is organized as follows. Section 4.2 describes the unmixing method that we extended to generate abundance information for each endmember from different groups of wavelengths. Section 4.3 presents our proposed KD-estimated SLIC algorithm for superpixel extraction. Section 4.4 gives our proposed architecture of the RNN formulated deep model. Section 4.5 presents the experimental results on four datasets along with detailed parameter analysis. Finally, conclusions are drawn in Section 4.6.

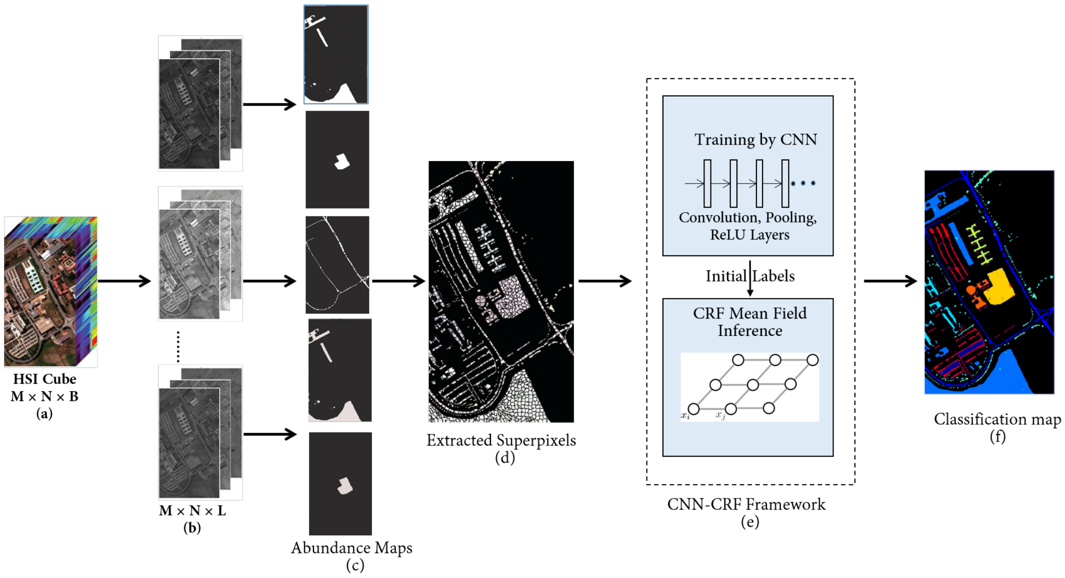


FIGURE 4.1: Our proposed architecture: (a) original hyperspectral cube with B bands. (b) different band groups consisting of L bands ($L \ll B$) each (c) resulting feature maps after region-based NMF (d) resulting superpixels (e) RNN formulated CNN-CRF model (f) classification map.

4.2 Generating Unmixing Results

In this section, we briefly explain the unmixing procedure presented in [171] that is used in our method to generate abundance matrix for each endmember. We extend this approach by considering abundance information from different groups of wavelengths across the spectral channels to capture distinct spectral-spatial estimations of each endmember in the image.

4.2.1 Region Based NMF for Structure Consistency and Preservation

At first, an HSI $\mathbf{H}$ is segmented into a set of Q homogeneous regions $R_1, R_2, \ldots, R_Q$ using a graph-based method [177]. In this method, each pixel in $\mathbf{H}$ is represented as a vertex $v_i \in V$, $i = 1, 2, \ldots, N$ and neighbouring pixels v_i, v_j are connected by an edge whose weight represents the distance between the connecting pixels. This algorithm merges the pixels to construct homogeneous regions by using two important criteria: maximum internal difference D_1 and minimum connecting weight D_2. D_1 is the largest weight in a minimum spanning tree *MST(R)* [178] of a homogeneous region R defined for all $v_i, v_j \in R, (v_i, v_j) \in MST(R)$ as:

$$D_1 = W_{max}(v_i, v_j) + \frac{\alpha}{R} \tag{4.1}$$

α is a parameter that controls the contributions from small regions. D_2 between regions R_1 and R_2 is defined for all $v_i \in R_1, v_j \in R_2, (v_i, v_j) \in MST(R)$ as:

$$D_2 = W_{min}(v_i, v_j) \tag{4.2}$$

If for two homogeneous regions $D_1 > D_2$, these two regions are merged. Otherwise, they are not merged.

After we obtain the homogeneous regions, the mean abundance spectral response t_q is estimated from the mean values of spectral responses for each region R_q. Within homogeneous region, the spectral responses for each pixel should be similar and as a result, the abundance of every pixel should be similar as well. A constraint between mean abundance t_q and estimated abundance vector $\mathbf{a}_q \in \mathbf{A}$ for each pixel of the region is set to apply the structure consistency in the region. With this constraint, the objective function to minimise for region R_q is defined as [171]:

$$\begin{aligned} F(\mathbf{M}_q, \mathbf{A}_q) = \frac{1}{2}||\mathbf{H}_q - \mathbf{M}_q\mathbf{A}_q||_F^2 + \sum_{q=1}^{Q}\sum_{n \in R_q} ||\mathbf{r}_q\tau_q^2||_{1/2} \\ +\eta\sum_{q=1}^{Q}\sum_{n \in R_q} ||\mathbf{a}_n - t_q||_2^2 \end{aligned} \tag{4.3}$$

where $\mathbf{H}_q$ contains the raw spectral responses of pixels in the homogeneous region R_q, $\mathbf{M}_q$ and $\mathbf{A}_q$ are the estimated endmember matrix and abundance matrix, respectively. The second term controls the sparsity of abundance in R_q and τ controls the contribution of this sparsity. The third term is responsible for controlling the consistency of the estimated abundance in R_q in which η controls the contribution from structure consistency.

Next, structure preserving constraint is applied to discover the relationship between the homogeneous regions to separate their contribution across the image. This constraint helps in preserving the local affinity of data distribution both before and after the matrix factorization. Also, it avoids the effect of distant repulsion [179] which is a distortion done by the distant data points. The distant repulsion causes different materials to contain different abundance while local affinity ensures that the same material in different regions shall have similar abundance. Graph regularization is applied later to preserve the structural information where the vertices represent the reflectance at different data points. Details of this approach and the optimization process can be found in [171].

4.2.2 Band Group based Abundance Estimation

In this chapter, we estimate the abundance of each endmember for groups of bands spanning over the entire spectral channels. Therefore, instead of calculating abundance over the entire spectra at once as in [171], we calculate abundances for a number of band groups with the goal to capture subtle spectral-spatial contribution from the image. In this way, each band group leads to distinct estimation of endmembers and their abundance. Such band grouping strategy has several advantages. Firstly, smaller band groups provide better spectral-spatial estimation locally. Secondly, different band groups capture spectral information in different ranges of wavelengths which can contribute in classification performance by providing more material based information to the classifier. Finally, a large number of abundance maps representing the endmembers of the image are expected to benefit the CNN in terms of both sufficient amount of training samples and useful spectral-spatial information as input to start the training with.

We group the original B bands into segments of L bands where $L << B$. Therefore, endmember and abundance matrices can be estimated from each band group in a multi-task manner by the following linear mixture models as:

$$\begin{aligned}
\mathbf{H}_1 &= \mathbf{M}_1\mathbf{A}_1 + \mathbf{E}_1 \\
\mathbf{H}_2 &= \mathbf{M}_2\mathbf{A}_2 + \mathbf{E}_2 \\
&\dots \\
\mathbf{H}_J &= \mathbf{M}_J\mathbf{A}_J + \mathbf{E}_J
\end{aligned} \tag{4.4}$$

where J is the total number of band groups. $\mathbf{M}_g$ and $\mathbf{A}_g$ are the group-wise endmembers and abundances, where $g = 1, \ldots, J$. $\mathbf{E}_g$ is the corresponding additive noise.

For each band group, the unmixing step follows the method in [171] with all parameters remaining the same across band groups. The sparsity constraint τ, however, is band group dependent and is defined as:

$$\tau_g = \frac{1}{\sqrt{L}} \sum_{l=1}^{L} \frac{\sqrt{N} - ||\mathbf{h}_l||_1/||\mathbf{h}_l||_2}{\sqrt{N-1}} \tag{4.5}$$

where $\mathbf{h}_l$ is the spectral responses in the l-th band and N is the total number of pixels in the image.

Furthermore, the local affinity and distance repulsion calculation are also undertaken within each band group. Finally, we obtain $K \times J$ abundance maps where K is the number of endmembers in each band group.

4.3 Forming Superpixels

An ideal superpixel algorithm should result in a properly connected set of pixels which belong to exactly one semantic region. The algorithm should also contain the property of being as regular as possible for features that require spatial support. Therefore, it is desirable to define an energy function that integrates an appearance term for coherency, a smooth constraint term and a connectivity term.

In this chapter, we extend the SLIC algorithm [180] [181], a modified version of K-means algorithm, to produce superpixels by using both the spectral and spatial features of every individual pixel represented by utilizing the abundance matrices obtained from the unmixing algorithm earlier. The parameters of the algorithm control the size and the regularity of the superpixels with fast computation speed and good accuracy. It is also expected to generalize well to multiple spectral bands. Our modified version of SLIC has the following basic steps:

1. Construct a feature vector $\omega(x_i, y_i) = (\vartheta_{x_i}, \vartheta_{y_i}, \mathbf{A}(x_i, y_i))$ for every pixel i in the image, where x_i and y_i are the spatial location of the pixel i, $\mathbf{A}(x_i, y_i)$ is the vector containing the estimated abundance values for each endmember across the group of bands and ϑ is a parameter that trades off the contribution of spatial and spectral information. It can be computed as the ratio $\frac{m}{S}$, where the size of a superpixel is assumed to be $S \times S$ and the value of m controls the regularity of the superpixels. We will illustrate the effect on the choices of values of S and m in the experiments.

2. Define an initial set of cluster centers $C_k = \omega(x_k, y_k)$ on a grid of step size S. After that each cluster center is moved to the lowest gradient position in an $n \times n$ neighborhood.

3. Assign each pixel i to the closest cluster center by computing the Euclidean distances $||\omega(x_i, y_i) - \omega(x_k, y_k)||$. We can accelerate this step by simplifying the search of the cluster centers within a $2S \times 2S$ neighborhood.

4. Instead of taking the mean value of the pixels to update the cluster centers of the superpixels, we introduce a KD-estimated PDF to describe the spectral distribution of the superpixels. The spectral distribution $P_{\mathbf{r}_k}(\mathbf{c})$ of a superpixel $\mathbf{r}_k$ is:
$$P_{\mathbf{r}_k}(\mu_k) = \frac{1}{n_k h} \sum_{i=1}^{k} \psi\left(\frac{\mu_k - \mathbf{c}_{x_i}}{h}\right) \tag{4.6}$$
where μ_k is the mean of the superpixel's spectral feature, $\mathbf{c}_{x_i}$ is the spectral vector of pixel x_i and n_k is the total number of pixels in $\mathbf{r}_k$. $\psi(.)$ places a kernel density around each superpixel and h is the bandwidth of the kernel smoothing window. Here, the widely used Gaussian kernel is adopted to estimate PDF because of its key advantages, such as, continuity, differentiability and locality [30]. The choice of the value of h is very important as it controls the smoothness of the resulting probability density curve. Selecting too large h may result in over-smooth estimator, while a too small h may produce an estimator with statistical instability. For our work, we choose h experimentally such that $h \in [0.05, 0.20]$ for feature values between [0,1]. Once we obtain these KD-estimated probability distribution for the values of the pixels in each superpixel, we update the cluster centers which will be more better representation of the spectral distribution of the superpixels.

5. Repeat the process iteratively until the distance between the successive cluster center updates is below a threshold.

6. Finally, a post-processing step enforces connectivity by reassigning disjoint segments to nearby cluster.

The procedure of our proposed superpixel extraction algorithm is described in Algorithm 3

Algorithm 3: Superpixel Extraction Procedure

```
Data: Hyperspectral image $\mathbf{H}$, Abundance Matrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N$
where $N$ is the total number of pixels in $\mathbf{H}$.
/* Initialization */
1. Construct $\omega(x_i, y_i) = (\vartheta_{x_i}, \vartheta_{y_i}, \mathbf{A}(x_i, y_i))$ for every pixel $i$
2. Initialize cluster centers $C_k = \omega(x_k, y_k)$ at regular grid steps $S$.
3. Move $C_k$ to the lowest gradient position in a $3 \times 3$ neighborhood.
4. Set label $y(i) = -1$ for each pixel $i$.
5. Set distance $d(i) = \infty$ for each pixel $i$.
/* Assignment */
while not converged do
  for each cluster center $C_k$ do
    for each pixel in a $2S \times 2S$ region around $C_k$ do
      Compute $D = ||\omega(x_i, y_i) - \omega(x_k, y_k)||$
      if $D < d(i)$ then
        Set $d(i) = D$
        Set $y(i) = k$
      end if
    end for
  end for
  /* Update */
  1. Compute KD-estimated PDF for every superpixel $R_k$ using Eq. (4.6)
  2. Update cluster centers
end while
  Output: Generated Superpixels $\mathbf{r}_1, \mathbf{r}_2, \ldots$
```

4.4 A CNN-CRF Model for Superpixel Labelling

In this section, we present a classification model that combines the properties of both CNN and CRF in different stages to classify the superpixels obtained earlier. We use CNN to generate an initial classification output which are later provided as input to CRF to improve the results.

4.4.1 Feature Learning and Initial Classification Using CNN

In this section, we explain the process of using the abundance matrix-guided superpixels as the input to the CNN for generating features and later classifying HSI data. In our method, instead of using pixel-based raw data, we treat the superpixels as input to the CNN. These abundance matrices contain the estimation of contribution that each endmember has in each band group. Therefore, our CNN implicitly receives spectral information across the entire spectral bands through these abundance matrices in spite of using convolution operations across the spatial domain only. For Y classes and J band groups, we have $Y \times J$ abundance matrices to be provided as the input to the CNN.

Our CNN includes several convolution layers, pooling layers, batch normalization (BN), fully connected layers and ReLU as the activation function. After every pooling operation, we apply BN to the respective layer of the network (except for the output layer) in order to prevent the model from collapsing all samples to a single point. In this way, we normalize the responses to have zero mean and unit variance over the entire batch. ReLU activation functions are performed at each layer in order to allow gradients to flow backward the layer. After obtaining the convolved features, the feature maps are then flattened into a feature vector and fed into a fully-connected layer which extracts the final learned deep features. Finally, we use logistic regression (LR) as a classifier to generate the required classified labels of the superpixels. The detailed architecture of the CNN is provided in Section 4.5.2.

During the CNN training, all the connections/weights are being updated by using the gradient descent back propagation algorithm. We randomly initialize the model parameters. A cost function is required to update the weights during the training. In our training process, we use mini-batch update procedure which is computed on a mini-batch of inputs [182]:

$$d = -\frac{1}{s}[y_i log(z_i) + (1 - y_i) log(1 - z_i)] \tag{4.7}$$

where s is the mini-batch size, y_i and z_i are the i-th predicted label and label in the mini-batch respectively. We aim to optimize cost d using stochastic gradient descent.

LR uses soft-max function to classify the learned features from the CNN. For the given input X, the probability that the input belongs to class i over all classes in $\mathbb{Y}$ can be calculated as:

$$P(y_i|X, W) = \frac{e^{W_i I}}{\sum\limits_{\mathbb{Y}}} e^{W_Y I} \tag{4.8}$$

where W are the weights of the LR layer.

We propose to train our CNN with local spatial-spectral information through the use of abundance matrices so as to let different wavelength ranges make distinct contribution to the classification outcome. Because the convolution kernels are applied to each of these different abundance matrices, there is a high possibility that the intermediate feature maps will become more meaningful and interesting throughout the training.

4.4.2 Refine Superpixel Labelling Using Conditional Random Fields

In this section, we briefly explain how we used CRF for superpixel-wise labelling. The CRF models superpixel labels as random variables that form an MRF when conditioned upon a global observation which is considered to be the spectral spatial properties of the superpixels [183]. In a fully connected pairwise CRF model, the energy of label assignment of a superpixel is given by the following equation:

$$E(x) = \sum_i \phi(y_i) + \sum_{i \neq j} \psi(y_i, y_j) \tag{4.9}$$

where the first term represents the unary potential of the inverse likelihood of the superpixel i taking a particular label y_i and the second term is the pairwise potential between two superpixels i and j for assigning two labels y_i, y_j simultaneously.

In our model, the unary energies are obtained from the CNN which predicts the labels for the superpixels but it does not consider the smoothness and consistence of the label assignments. The pairwise energies contain a smoothness term that encourages assigning similar labels to superpixels with similar properties. As in [153], we model the pairwise energies as weighted Gaussians as follows:

$$\psi(y_i, y_j) = \varphi(y_i, y_j) \sum_{f=1}^{F} w^{(f)} k_G^{(f)}(v_i, v_j) \tag{4.10}$$

where $\varphi(.)$ is the label compatibility function which encodes a Potts model, i.e., $\varphi(y_i, y_j) = 1_{y_i \neq y_j}$. The Potts model effectively penalizes the case where two superpixels i and j are assigned different labels when $\sum_{f=1}^{F} w^{(f)} k_G^{(f)}$ is large. The efficient approximate inference requires the kernels $k^{(f)}(.,.)$ to be Gaussian kernels computed over elements of the feature vector v_i that describes superpixel i with a scalar weight w^f. By minimizing the CRF energy, we obtain the most probable label assignment for the superpixels which is intractable. Therefore, we use the mean-field approximation algorithm to the CRF distribution for maximum posterior marginal inference. This is done by approximating the CRF distribution $P(X)$ by a simpler distribution $Q(X)$ which is expressed as the product of independent marginal distributions $Q(X) = \prod_i Q_i(X_i)$. We present the steps of this inference algorithm next.

4.4.3 Mean-field Iteration of CRF Inference for Superpixel Labelling

Unlike standard CNNs in which filters are fixed after training, we employ edge-preserving Gaussian spatial and bilateral filters for approximating mean-field inference in each iteration. These filters depend on the original spatial and appearance information of the superpixels. As in [153], we reformulate the steps of the inference as layers. In order to do this, we are required to back-propagate the error differentials to previous layers which are calculated based on the input. The steps are discussed in the following:

Initialization

The first step is the initialization in which the following operation is performed:

$$Q_i(y) \leftarrow \frac{1}{Z_i} exp(U_i(y)) \tag{4.11}$$

where $Z_i = \sum_y exp(U_i(y))$. Note that $U_i(y)$ denotes the negative of the unary energy. This operation is simply applying a soft-max function over the unary potential across all the labels at each superpixel. The error differentials obtained from this output can be passed backward to the unary potential inputs.

Message Passing

The second step is the message passing which is applying F Gaussian filters on the current estimation of the predictions of the superpixels. We use two Gaussian kernels: spatial and bilateral kernels. The filter coefficients are obtained from the spectral and spatial features of the superpixels. This reflects how strongly two superpixels are related to each other. For simplification, we keep the bandwidth values of the kernels fixed. By using back-propagation, we calculate error derivatives on the filter responses and send those through the same Gaussian filters in backward direction.

Weighting Filter Outputs

The next step is to take the weighted sum of the F filter outputs from the previous step for each label of the superpixels. When each label is considered, it can be viewed as the usual convolution with a 1×1 filter with F input channels and one output channel. The error can be calculated since both inputs and outputs are known during back-propagation. This allows an automatic learning of filter weights. To facilitate the process of increasing the ability of discrimination among various classes, we set independent kernel weights for each individual class. Although it increases the number of trainable parameters, the independent weights will complement the relative importance of both kernels in improving class-specific decisions.

Compatibility Transform

In this step, a label compatibility function $\varphi(y, y')$ is applied in order to test the compatibility between two labels y and y' received from the previous step. A fixed penalty is given when two superpixels containing similar properties are assigned different labels, given by the Potts model $\varphi(y, y') = [y \neq y']$, where [.] is the Iverson bracket [184]. This Iverson bracket evaluates the logical proposition and based on the results, the penalty is given. A limitation of this model is that it assigns the same penalty for all pairs of labels. It is not desirable, specifically for remote sensing images where the geographical locations have significant effects in the classification. For example, the pair "crops" and "trees" should have smaller penalty than that of "building" and "water". Therefore, we learn φ directly from data and fix it with the Potts model.

This compatibility transform step can be viewed as another convolution layer where the spatial receptive field of the filter 1 × 1 filter and the number of input and output channels are equal. Updating the weights of the filters during back-propagation will eventually learn φ. As done in other steps, the error differentials from the outputs of this step are transferred backward to the input.

Adding Unary Potentials

In this step, the outputs from the compatibility transform step is subtracted from the initial unary inputs. The error differentials at the end of the step are transferred back to the inputs.

Normalization

Finally, the normalization step of the iteration can be expressed as another softmax operation. The error differentials at the end of the step are transferred back to the inputs using the softmax's backward pass.

The steps of the mean-field inference algorithm is listed in Algorithm 4.

4.4.4 Using Mean-field Iterations as RNN

In this section, we present an end-to-end learning framework after formulate the mean-field iterations as an RNN. As shown in the previous section, we organize each t-th

Algorithm 4: Mean-field Inference Algorithm

```
Data: Set of Labels $y_1, y_2, \ldots$ (initially classifier by CNN).
/* Initialization */
$Q_i^{(1)}(y) \leftarrow \frac{1}{Z_i} exp(U_i(y))$ for all superpixel $i$
while not converged do
    /* Message Passing */
    $Q_i^{(2)}(y) \leftarrow \sum_{j \neq i} k^{(f)}(v_i, v_j) Q_j^{(1)}(y)$ for all kernels and superpixels $i$ and $j$
    /* Weighting Filter Outputs */
    $Q_i^{(3)}(y) \leftarrow \sum_{f} w^{(f)} Q_i^{(2)}(y)$
    /* Compatibility Transform */
    $Q_i^{(4)}(y) \leftarrow \sum_{y' \in L} \varphi(y, y') Q_i^{(3)}(y)$
    /* Adding Unary Potentials */
    $Q_i^{(5)}(y) \leftarrow U_i(y) - Q_i^{(4)}(y)$
    /* Normalization */
    $Q_i^{(6)}(y) \leftarrow \frac{1}{Z_i} exp Q_i^{(5)}(y)$
end while
Output: Improved Classified Labels
```

iteration of the mean-field inference as a stack of layers from which we obtain an estimation of marginal probabilities Q_t. Therefore, given an image $\mathbf{H}$ and the superpixel-wise unary potential values U, we can estimate the next set of marginal probabilities after one mean-field iteration by

$$Q_{t+1} = f_\theta(U, Q_t, \mathbf{H}) \tag{4.12}$$

where the vector θ represents the CRF parameters such as the weights, compatibility function and Gaussian kernel.

Hence, the next mean-field iterations can be implemented by repeating the same stack of layers in such a way that each new iteration receives an estimation of Q_{t-1} marginal probabilities from the previous iteration and the abundance matrix-guided unary potential in their original form. This iterative procedure can be formulated as an RNN as we save the state of each individual mean-field iteration in an equivalent long term short memory (LSTM) gate. This complete system is end-to-end trainable by back propagation algorithm.

In the forward pass of the network, when the execution enters the combined part of CRF and RNN, it falls into the loop of the RNN and perform I number of iterations. In this stage, the computation for updating the weights is performed inside the loop of RNN. After an output is obtained from the loop, the following stages of the deep network can continue the forward pass with a softmax layer included at the end in order to produce the final labels of the superpixels.

During backward pass of the network, the error differentials of the output go through same I number of iterations inside the loop of the RNN before reaching RNN's input. After that it gradually backpropagates toward the CNN that generates the initial U. It is to be noted that the error differentials are calculated inside each iteration of the loop of the mean-field algorithm explained in the previous section. The procedure of mean-filed inference iterations as RNN is illustrated in Fig. 4.2.

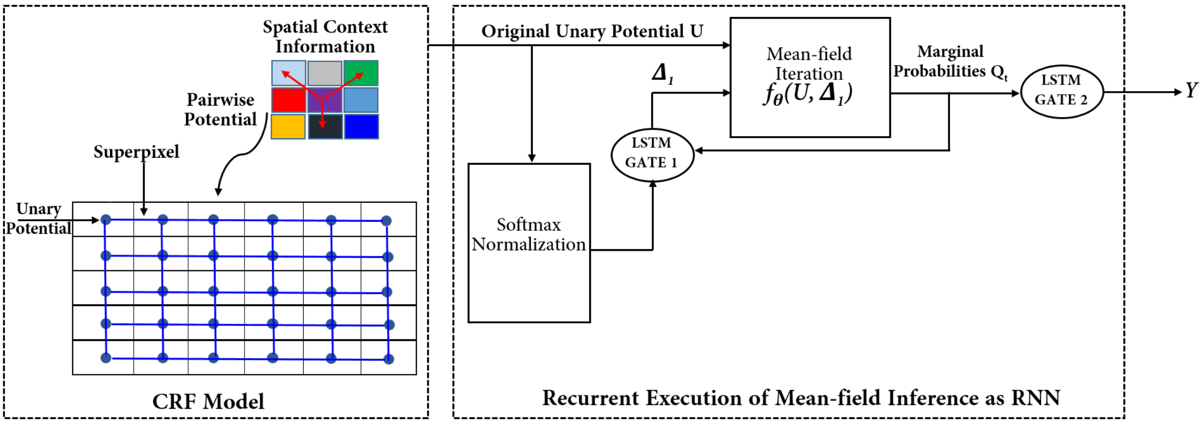


Figure 4.2: CRF mean-field iterations as RNN

4.5 Experiments

Having presented our method in the previous sections, we now demonstrate the effectiveness of our proposed method. A series of experiments have been done to evaluate the performance of the proposed methods based on the following stages:

1. We evaluate the performance of our proposed abundance information-guided superpixel extraction procedure by comparing with standard SLIC algorithm.

2. We evaluate the effectiveness of including band group-based abundance matrices as input to the classification models by testing on widely used classifiers in remote sensing domain.

3. We further evaluate the usefulness of our CNN-CRF model by comparing with other classification methods.

However, before performing the evaluation processes, we define our datasets and tune the parameters to determine optimal values for use in the classifiers.

4.5.1 Data Sets

In the experiments, we used three widely used hyperspectral datasets, i.e., Indian Pines, Pavia University and Salinas. For better evaluation of our proposed method, we used a new dataset "Griffith-USGS" that we introduced in section 3.5.2 in Chapter 3. In our method, we extract abundance-information guided superpixels from each band group along the spectral channels which gives us multiple estimations of abundance information. As a result, we obtain a large number of abundance maps representing the endmembers for each pixel that eventually benefits our deep model with sufficient amount of training samples without using an explicit data augmentation procedure that we implemented in Chapter 3. We split the available samples into training and testing sets and perform a 10-fold cross validation to select training samples for each class. We use 90% of the total number of training samples to learn the weights and biases of the neurons and the remaining 10% to validate and further update the design of the architecture. With the large number of available samples, we are able to address the issue of overfitting and can select small number of training samples which are still sufficient to train our proposed deep model. We report the total number of training and testing samples for each class used on the three datasets in Tables 4.1, 4.2, 4.3 and 4.4.

4.5.2 Parameter Analysis of the Classifier Used

In this chapter, we use unmixing as an important feature extraction approach to be provided abundance maps as the input for classification. That is why we evaluated the

Table 4.1: Number of Training and Testing Samples Distribution for Each Class on Indian Pines Dataset

Class	Training Samples	Testing Samples
Alfalfa	33	29
Corn-notill	670	1309
Corn-mintill	496	685
Corn	140	125
Grass-pasture	305	356
Grass-trees	477	661
Grass-pasture-mowed	15	15
Hay-windrowed	322	389
Oats	16	10
Soybean-notill	766	840
Soybean-mintill	1009	2260
Soybean-clean	339	463
Wheat	194	165
Woods	624	1095
Buildings-Grass-Trees-Drives	289	298
Stone-Steel-Towers	66	59
Total	5761	8759

Table 4.2: Number of Training and Testing Samples Distribution for Each Class on Pavia University Dataset

Class	Training Samples	Testing Samples
Asphalt	3170	5592
Meadows	6293	14890
Gravel	675	1569
Trees	1016	2710
Painted metal sheets	636	1185
Bare Soil	2316	4712
Bitumen	532	1105
Self-Blocking Bricks	1491	3427
Shadows	713	806
Total	16842	35996

effectiveness of including unmixing results not only for our proposed CNN model but also for state-of-the-art classifiers as well. In this regard, we compared and evaluated the performance of RF, SVM, and KNN to observe the rate of improvement these classifiers achieve after incorporating unmixing results in the form of abundance matrices.

Use of appropriate parameters play an important role in classification accuracy with RF, SVM, and KNN. For each classifier, we performed a parameter tuning process to choose the optimal parameters based on the highest classification accuracy. We now present

Table 4.3: Number of Training and Testing Samples Distribution for Each Class on Salinas Dataset

Class	Training Samples	Testing Samples
Brocoli_green_weeds_1	1631	1812
Brocoli_green_weeds_2	3155	3505
Fallow	1522	1691
Fallow_rough_plow	1010	1122
Fallow_smooth	2181	2423
Stubble	3303	3670
Celery	2967	3297
Grapes_untrained	9914	11016
Soil_vinyard_develop	5401	6001
Corn_senesced_green_weeds	2745	3050
Lettuce_romaine_4wk	735	817
Lettuce_romaine_5wk	1534	1704
Lettuce_romaine_6wk	710	789
Lettuce_romaine_7wk	803	892
Vinyard_untrained	6330	7033
Vinyard_vertical_trellis	1419	1577
Total	45359	50399

Table 4.4: Number of Training and Testing Samples Distribution for Each Class on Griffith-USGS Dataset

Class	Training Samples	Testing Samples
Road	2113	1690
Water	2033	1509
Building	1744	1379
Grass	1936	1778
Tree	2109	1624
Soil	1699	1265
Total	11634	9245

the parameter tuning process for each of these classifiers. Also, for a fair comparison with our proposed deep model, we followed the same experimental settings explained in Section 4.5.1 to select the training and testing samples for classification with the following classifiers on each dataset.

Support Vector Machine (SVM)

For remote sensing image covering large land cover areas, radial basis function (RBF) kernel of the SVM classifier is widely used and produces good results. Hence, we used RBF kernel for implementing the SVM classifier for evaluation purposes. For RBF kernel, two parameters need to be set properly: (1) the optimum parameters of cost (C) which is soft margin cost function that controls the influence of misclassification and (2) the kernel width (γ) which is the inverse of the standard deviation of the RBF kernel (Gaussian function) and is used as similarity measure between two points. Larger C may cause over-fitting of the model and an increasing γ may affect the shape of the class-separating hyperplane. We followed the study carried out in [185] and tuned the parameters by taking ten values of C (2^{-2}, 2^{-1}, 2^0,2^1,2^2,2^3,2^4,2^5,2^6,2^7) and ten values of γ (2^{-5}, 2^{-4}, 2^{-3},2^{-2}, 2^{-1}, 2^0,2^1,2^2,2^3,2^4) and tested on all four datasets.

Random Forest (RF)

The number of trees in RF significantly affect the classification performance. Using more than the required number of trees may lead to inaccurate classification results. With the optimal number of trees, RF can provide stable classification output. To find the optimal value for the number of trees, we tested the values 100, 200, 300, 500 and 1000 on all four datasets.

K-Nearest Neighbor (*K*NN)

KNN finds a group of k samples that are nearest to unknown samples based on distance functions. The class of these unknown samples are determined by computing the average of the class attributes of the K nearest neighbor. Hence, the value of k plays a significant role in classification performance. To find the optimal value of K, we experimented from 1 to 20 for all datasets.

Findings from Parameter Analysis

As mentioned before, we experimented on several values of C and γ and the optimal parameters for the SVM model were chosen based on the lowest classification error on each dataset. Tables 4.5, 4.6, 4.7 and 4.8 show the relationship between the classification error and the SVM parameters. According to our experiments, lower classification error was observed for high values of C and low values of γ on each datasets. On the other hand, classification error increased for low values of C and with both high and low values of γ.

TABLE 4.5: Relationship between classification error and parameters (C and γ) of the SVM classifier obtained from Indian Pines. Values shown in blue in every row represent the lowest error for each C against all values of γ.

cost										
128	0.081	0.034	0.044	0.033	0.039	0.069	0.177	0.172	0.172	0.198
64	0.077	0.013	0.037	0.029	0.036	0.048	0.127	0.151	0.161	0.215
32	0.061	0.027	0.031	0.042	0.059	0.056	0.091	0.125	0.154	0.187
16	0.083	0.031	0.021	0.049	0.055	0.061	0.088	0.105	0.145	0.162
8	0.097	0.053	0.049	0.097	0.039	0.063	0.088	0.096	0.105	0.172
4	0.117	0.082	0.087	0.114	0.057	0.085	0.116	0.121	0.133	0.196
2	0.138	0.141	0.136	0.131	0.098	0.119	0.138	0.165	0.188	0.265
1	0.184	0.163	0.155	0.172	0.139	0.147	0.148	0.236	0.298	0.347
0.5	0.225	0.208	0.193	0.201	0.165	0.155	0.149	0.269	0.355	0.409
0.25	0.274	0.248	0.245	0.238	0.231	0.197	0.279	0.351	0.402	0.445
	0.03125	0.0625	0.125	0.25	0.5	1	2	4	8	16

gamma

TABLE 4.6: Relationship between classification error and parameters (C and γ) of the SVM classifier obtained from Pavia University. Values shown in blue in every row represent the lowest error for each C against all values of γ.

cost										
128	0.038	0.055	0.059	0.063	0.071	0.093	0.159	0.191	0.223	0.236
64	0.049	0.045	0.038	0.041	0.049	0.058	0.096	0.098	0.116	0.175
32	0.046	0.044	0.031	0.019	0.038	0.045	0.077	0.112	0.138	0.145
16	0.074	0.064	0.061	0.055	0.021	0.029	0.065	0.166	0.176	0.196
8	0.115	0.095	0.084	0.122	0.084	0.031	0.087	0.171	0.181	0.223
4	0.132	0.104	0.089	0.105	0.098	0.033	0.129	0.189	0.198	0.269
2	0.187	0.165	0.134	0.119	0.098	0.065	0.142	0.194	0.221	0.394
1	0.261	0.225	0.183	0.144	0.119	0.146	0.155	0.216	0.275	0.433
0.5	0.301	0.288	0.227	0.156	0.126	0.161	0.175	0.255	0.349	0.407
0.25	0.433	0.375	0.334	0.217	0.155	0.185	0.266	0.355	0.453	0.526
	0.03125	0.0625	0.125	0.25	0.5	1	2	4	8	16

gamma

In case of finding the optimal parameters of RF classifier, different datasets produced lower classification errors for values between 200 and 400. However, the classification accuracy decreased for all datasets for values greater than 400. We also evaluated the classification accuracy by testing the effect of different number of splits in accordance with the number of trees. Fig. 4.3 shows the effect of the number of trees and the number of splits on the overall accuracy.

Table 4.7: Relationship between classification error and parameters (C and γ) of the SVM classifier obtained from Salinas. Values shown in blue in every row represent the lowest error for each C against all values of γ.

cost										
128	0.085	0.072	0.055	0.084	0.113	0.075	0.165	0.176	0.196	0.322
64	0.077	0.065	0.035	0.044	0.086	0.041	0.144	0.189	0.201	0.336
32	0.071	0.047	0.029	0.017	0.078	0.088	0.183	0.198	0.223	0.341
16	0.023	0.055	0.032	0.021	0.075	0.147	0.193	0.209	0.223	0.355
8	0.039	0.055	0.044	0.029	0.086	0.142	0.223	0.234	0.275	0.397
4	0.043	0.059	0.058	0.040	0.121	0.139	0.224	0.265	0.282	0.393
2	0.049	0.063	0.065	0.056	0.147	0.137	0.224	0.275	0.291	0.393
1	0.054	0.066	0.072	0.066	0.153	0.141	0.277	0.284	0.302	0.391
0.5	0.066	0.071	0.079	0.074	0.167	0.155	0.283	0.294	0.302	0.482
0.25	0.069	0.074	0.081	0.085	0.172	0.175	0.283	0.301	0.352	0.483
	0.03125	0.0625	0.125	0.25	0.5	1	2	4	8	16

gamma

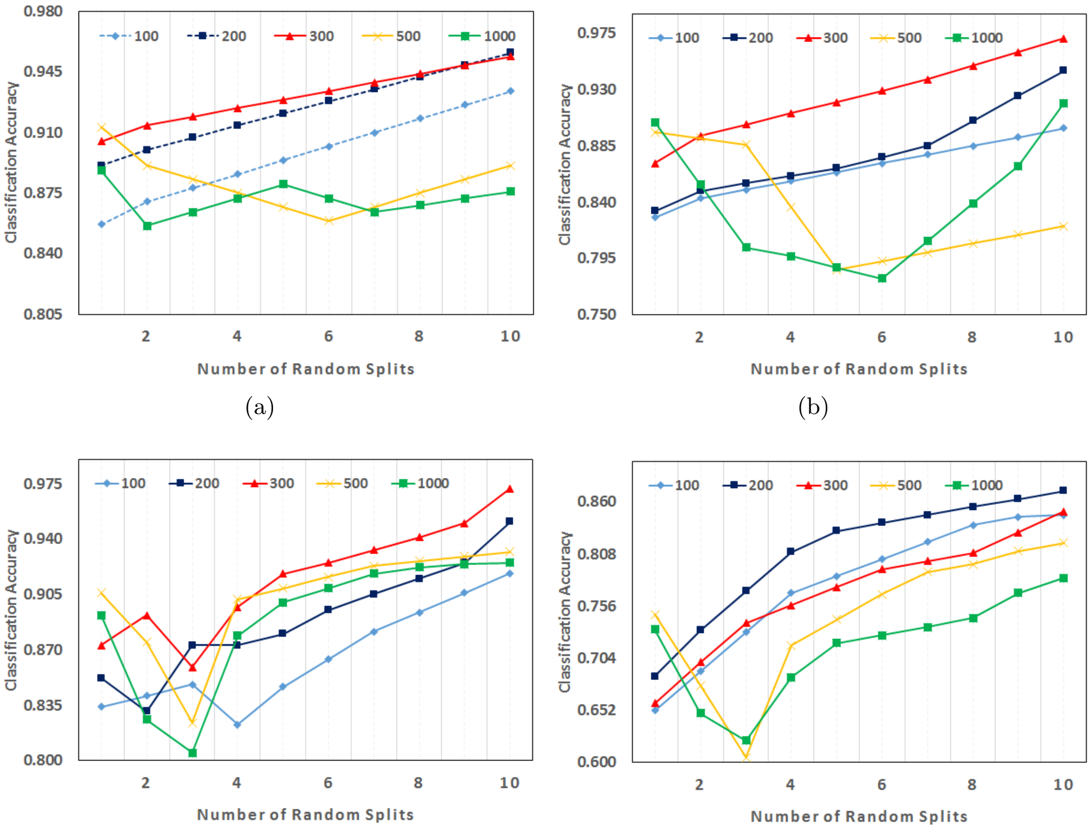
Table 4.8: Relationship between classification error and parameters (C and γ) of the SVM classifier obtained from Griffith-USGS. Values shown in blue in every row represent the lowest error for each C against all values of γ.

cost										
128	0.019	0.027	0.031	0.045	0.075	0.103	0.193	0.242	0.264	0.311
64	0.023	0.025	0.029	0.048	0.058	0.075	0.079	0.253	0.285	0.311
32	0.029	0.018	0.029	0.048	0.031	0.023	0.048	0.265	0.285	0.345
16	0.031	0.023	0.014	0.053	0.055	0.097	0.088	0.281	0.291	0.349
8	0.037	0.035	0.017	0.045	0.069	0.109	0.126	0.284	0.294	0.366
4	0.045	0.033	0.019	0.045	0.071	0.122	0.158	0.292	0.332	0.366
2	0.051	0.038	0.021	0.025	0.083	0.136	0.194	0.292	0.336	0.381
1	0.052	0.045	0.029	0.019	0.096	0.184	0.232	0.333	0.382	0.401
0.5	0.064	0.049	0.032	0.021	0.103	0.194	0.291	0.371	0.391	0.411
0.25	0.075	0.051	0.037	0.023	0.115	0.207	0.299	0.391	0.483	0.483
	0.03125	0.0625	0.125	0.25	0.5	1	2	4	8	16

gamma

According to our experiments for KNN classifier, it is evident that increasing the value of k increases the classification error as well. Although we found different k values for different datasets, smaller values of k produced lower classification error consistently across all datasets. Fig. 4.4 shows the effect of the number of trees and the number of splits on the overall accuracy.

The optimal parameters of the classifiers SVM, RF and KNN for different datasets are provided in Table 4.9.



(a)

(b)

(c)

(d)

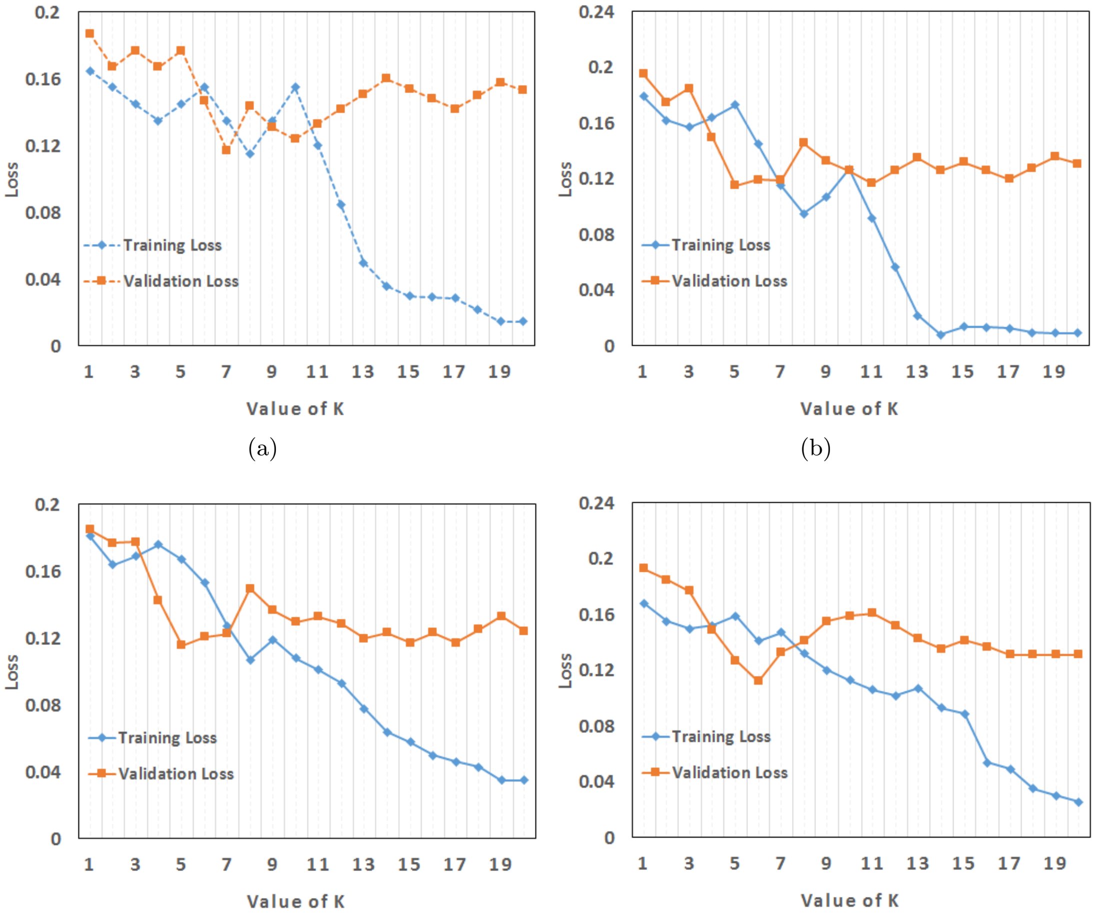
FIGURE 4.3: Effect of the number of trees and the number of random splits on the overall accuracy for RF classification obtained from different datasets: (a) Indian Pines (b) Pavia University (c) Salinas (d) Griffith-USGS.

TABLE 4.9: Optimal Parameters for SVM, RF and kNN

Dataset	Support Vector Machine		Random Forest	K Nearest Neighbor
	Gamma	Cost	Number of Trees	Value of k
Indian Pines	0.0625	64	200	7
Pavia University	0.25	32	300	5
Salinas	0.25	32	300	5
GU-USGS	0.125	16	200	6

Design of the CNNs

For the CNN used in our method, we experimented on different kernel sizes: 5×5, 6×6, 8×8, 10×10, 12×12, 14×14 and 15×15 and found out that smaller kernel sizes resulted in the best testing classification accuracies on each dataset. Depending on the datasets, we adopted three to five convolution layers and two to three pooling layers with 2×2 pooling kernel in each layer. ReLU layers were used as well and cut off the features that



(a) (b) (c) (d)

FIGURE 4.4: Relationship between classification error (y-axis) and k value (x-axis) for KNN classification obtained from different datasets: (a) Indian Pines (b) Pavia University (c) Salinas (d) Griffith-USGS.

were less than 0. The size of the mini-batch was set to 10. For the logistic regression, the learning rate was set to 0.005 and the number of epochs was 650. The weights were randomly initialized and gradually trained using the back propagation algorithm. Each convolution kernel extracted distinct features from the abundance matrices as input which conveyed meaningful structural information about the data. The architecture of the CNNs used in our method is explained in Table 4.10. We decided on the number of convolution and pooling layers after carefully evaluating the training and testing losses on all datasets.

Table 4.10: Architecture of the CNN

Dataset	Layer	Convolution Layer	Pooling Layer	BN	Activation Function
Indian Pines	1	5 × 5	2 × 2	Yes	ReLU
	2	5 × 5	No	No	ReLU
	3	5 × 5	2 × 2	Yes	ReLU
Pavia University	1	6 × 6	2 × 2	Yes	ReLU
	2	6 × 6	No	No	ReLU
	3	6 × 6	2 × 2	Yes	ReLU
	4	6 × 6	2 × 2	Yes	ReLU
Salinas	1	6 × 6	2 × 2	Yes	ReLU
	2	6 × 6	No	No	ReLU
	3	6 × 6	2 × 2	Yes	ReLU
	4	6 × 6	2 × 2	Yes	ReLU
Griffith-USGS	1	5 × 5	2 × 2	Yes	ReLU
	2	5 × 5	No	No	ReLU
	3	5 × 5	2 × 2	Yes	ReLU
	4	5 × 5	No	No	ReLU
	5	5 × 5	2 × 2	Yes	ReLU

4.5.3 Abundance Information-guided Superpixels as Input

We generated multiple abundance matrices for each spectral signature from 20 groups of bands across the spectral channel. Fig. 4.5 shows an example of measuring an endmember "water" with its estimated abundances across various groups of bands. It can be observed that different band groups provide different estimations for the abundances as realized by the brightness of the pixels covering the region, though most of them are quite consistent with each other.

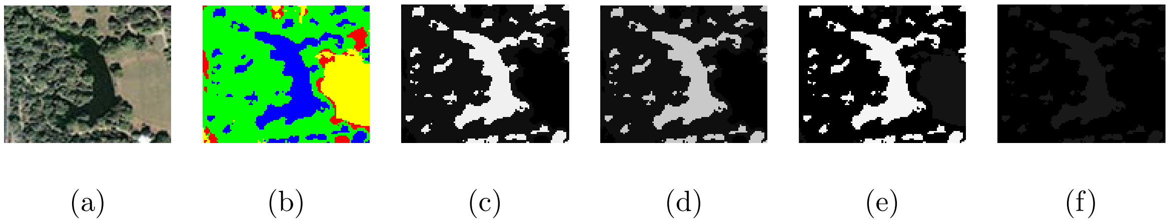


Figure 4.5: (a) Original image; (b) ground truth; (c-f) abundance maps of water in different band groups.

As mentioned in our methodology, instead of taking the mean value of the pixels to update the cluster centers of the superpixels, we introduce a KD-estimated PDF to describe the spectral distribution of the superpixels and update the cluster centers accordingly.

Along with this, the feature vectors representing the pixels contain the estimated abundance values for each endmember across the group of bands. To test the usefulness of including band group-based abundance information into the superpixel extraction procedure, we compared our proposed approach with [186], in which the authors proposed a deep model that was used to perform superpixel-level labelling. We also incorporated the same deep model for classification purposes. Table 4.11 shows the effect of providing unmixing results as input to generate superpixels and finally classifying the image.

Table 4.11: Effect of Unmixing Information-guided Superpixels. Best OAs(%) shown in bold on each dataset.

Dataset	Alam *et al.* [186]	Proposed Method
Indian Pines	96.60 ± 1.32	**99.25 ± 0.30**
Pavia University	97.12 ± 1.07	**99.30 ± 0.22**
Salinas	94.50 ± 1.65	**98.15 ± 0.45**
GU-USGS	75.81 ± 1.46	**83.66 ± 0.83**

From Table 4.11, we see that the inclusion of abundance information-guided superpixels as input increased the classification accuracy of the deep model. The compared method used the standard SLIC algorithm to generate the superpixels whereas we extended the algorithm by introducing KD-estimated PDF to describe the spectral distribution of the superpixels. The iterative update of the cluster centers by employing KD estimation improved the superpixel generation process. Hence, we conclude that the inclusion of abundance information significantly complemented the superpixel extraction and eventually improved the classification performance.

4.5.4 Abundance Information as Input for Traditional Classifiers

Next, we evaluated the performances of the classifiers for using unmixing output as an input to the model. At first, we simply executed the classifiers SVM, RF, kNN with the optimal parameters computed earlier by giving raw spectral information for the pixels. We also tested a standard CNN with the architecture provided in Table 4.10 and provided raw data as input. After this, we replaced the raw spectral information with the band group based abundance matrices computed earlier to train the model. Later, we made a comparison between these two different input information to test the usefulness of unmixing output. For performance evaluation, we calculated the overall accuracy (OA) with the corresponding standard deviations. From Tables 4.12, 4.13, 4.14, 4.15, we see that the classification accuracies of all the classifiers improved quite significantly after including abundance information as input. Also, to justify the usefulness of generating abundance matrices for smaller band groups instead of whole spectral channel at once, we executed our method in two settings: (1) abundance matrices generated from the whole spectral channels and (2) band group based abundance matrices. Results show that band groups lead to superior performance. With different wavelength ranges, the abundance information captured better local information distinct to each group and hence, the variety of spectral-spatial information had been estimated for each underlying endmember. Thanks to these advantages, all the classifiers were able to classify the pixels better than its counterpart where abundance information were collected from the whole spectral channel at once.

Table 4.12: Classification accuracies by SVM. Best OAs(%) shown in bold on each dataset.

Dataset	Raw Input	Abundance Matrices	
		Whole Spectra	Band Group
Indian Pines	90.13 ± 2.71	94.65 ± 1.18	**97.35 ± 1.09**
Pavia University	91.77 ± 1.90	95.69 ± 2.27	**97.90 ± 1.89**
Salinas	89.57 ± 1.60	93.35 ± 1.45	**96.78 ± 2.79**
GU-USGS	67.40 ± 1.89	72.24 ± 2.04	**76.54 ± 2.01**

Table 4.13: Classification accuracies by RF. Best OAs(%) shown in bold on each dataset.

Dataset	Raw Input	Abundance Matrices	
		Whole Spectra	Band Group
Indian Pines	89.37 ± 0.85	93.21 ± 1.56	**96.75 ± 2.15**
Pavia University	92.80 ± 1.05	95.92 ± 0.66	**98.12 ± 0.51**
Salinas	90.07 ± 1.22	94.71 ± 0.75	**96.80 ± 0.65**
GU-USGS	69.10 ± 2.44	75.60 ± 1.50	**77.90 ± 1.75**

4.5.5 Testing Proposed CNN-CRF model

At this stage, we evaluate the effectiveness of the proposed CNN-CRF model by comparing with other spectral-spatial classification methods which included additional steps to improve the initial classification step. Since we introduced CRF in our framework

Table 4.14: Classification accuracies by KNN. Best OAs(%) shown in bold on each dataset.

Dataset	Raw Input	Abundance Matrices	
		Whole Spectra	Band Group
Indian Pines	85.17 ± 2.35	89.80 ± 1.59	**92.88 ± 1.05**
Pavia University	88.70 ± 2.15	91.06 ± 2.80	**93.77 ± 1.60**
Salinas	87.00 ± 1.36	89.95 ± 1.11	**92.15 ± 1.25**
GU-USGS	63.75 ± 1.50	69.50 ± 2.36	**72.65 ± 2.60**

Table 4.15: Classification accuracies by standard CNN. Best OAs(%) shown in bold on each dataset.

Dataset	Raw Input	Abundance Matrices	
		Whole Spectra	Band Group
Indian Pines	91.10 ± 1.42	94.95 ± 0.90	**97.92 ± 0.55**
Pavia University	93.35 ± 1.40	96.69 ± 1.30	**99.05 ± 0.25**
Salinas	91.40 ± 0.90	95.60 ± 1.46	**98.10 ± 0.60**
GU-USGS	74.40 ± 2.90	80.50 ± 1.70	**82.00 ± 2.32**

to improve the initial CNN classification, it is important to test the usefulness of the proposed integration of CRF with CNN.

Our first baseline method RoF-MRF [187] introduces rotation forests, a variation of the standard random forest algorithm that uses feature extraction and subset features to promote both the diversity and the accuracy of the individual classifiers. In this method, four feature extraction methods: PCA, neighborhood preserving embedding (NPE), linear local tangent space alignment (LLTSA) and linearity preserving projection (LPP), are used in rotation forests to obtain the class probabilities based on spectral information. Later, spatial contextual information, modeled by MRF prior, is used to improve the classification results.

Our second baseline method MLR*sub*MLL [81] integrates spectral and spatial information into a Bayesian framework. In this method, a multinomial logistic regression (MLR) algorithm is used to learn the posterior probability distributions from the spectral information. Moreover, a subspace projection method is used to characterize noisy and mixed pixels. Later, spatial information is added using a multilevel logistic MRF prior.

We further made comparisons with "FCLS-SVM" [174] which generates abundance maps from whole spectral channel and uses them to improve the coarse classification by SVM.

Table 4.16: Comparison of classification accuracies with different methods. Best OAs(%) shown in bold on each dataset.

Dataset	RoF-MRF [187]	MLRsubMLL [81]	FCLS-SVM [174]	Proposed Method
Indian Pines	95.71 ± 1.44	93.70 ± 1.75	94.90 ± 1.48	**98.25 ± 0.30**
Pavia University	94.52 ± 0.79	95.65 ± 1.52	96.10 ± 1.15	**99.30 ± 0.22**
Salinas	93.18 ± 2.08	95.05 ± 1.25	93.28 ± 2.00	**98.15 ± 0.45**
GU-USGS	76.17 ± 2.35	79.15 ± 2.75	75.59 ± 2.42	**87.05 ± 0.83**

It does this by applying a fully constrained least squares (FCLS) method to every unlabeled pixel in order to obtain the abundance estimation of each land cover type. Finally, spatial regularization by simulated annealing is performed to obtain the refined classification output.

Table 4.16 shows the comparison of classification accuracies with the mentioned baseline methods. It can be seen that our proposed method outperforms the other methods by 3% - 5% on Indian Pines, Pavia University and Salinas. On Griffith-USGS, our method outperforms the other methods by 8% - 12%. We can draw two significant conclusions from these results. Firstly, the use of abundance matrices provide useful information as input for the classifier. Secondly, CNN is able to learn better features from those abundance matrices as it achieves significantly better classification accuracy than SVM which also considers abundance matrices under the same experimental settings.

Fig. 4.6 shows the intermediate features generated during the CNN training. We can observe that the four convolutions layers used in our model gradually constructed more structured representation of the data.

Fig. 4.7 illustrates the CNN based classification results on our dataset. The first column is the ground truth. The second column is the classification map generated by the CNN. The third column is a binary map that shows the effect of corresponding misclassification obtained by comparing with the ground truth. The white pixels indicate the parts of the image that were not correctly classified. It can be seen that maximum amount of misclassification happens in cases of classifying “road” and “soil”. Comparatively greater spectral similarities shared by these two different materials might possibly have

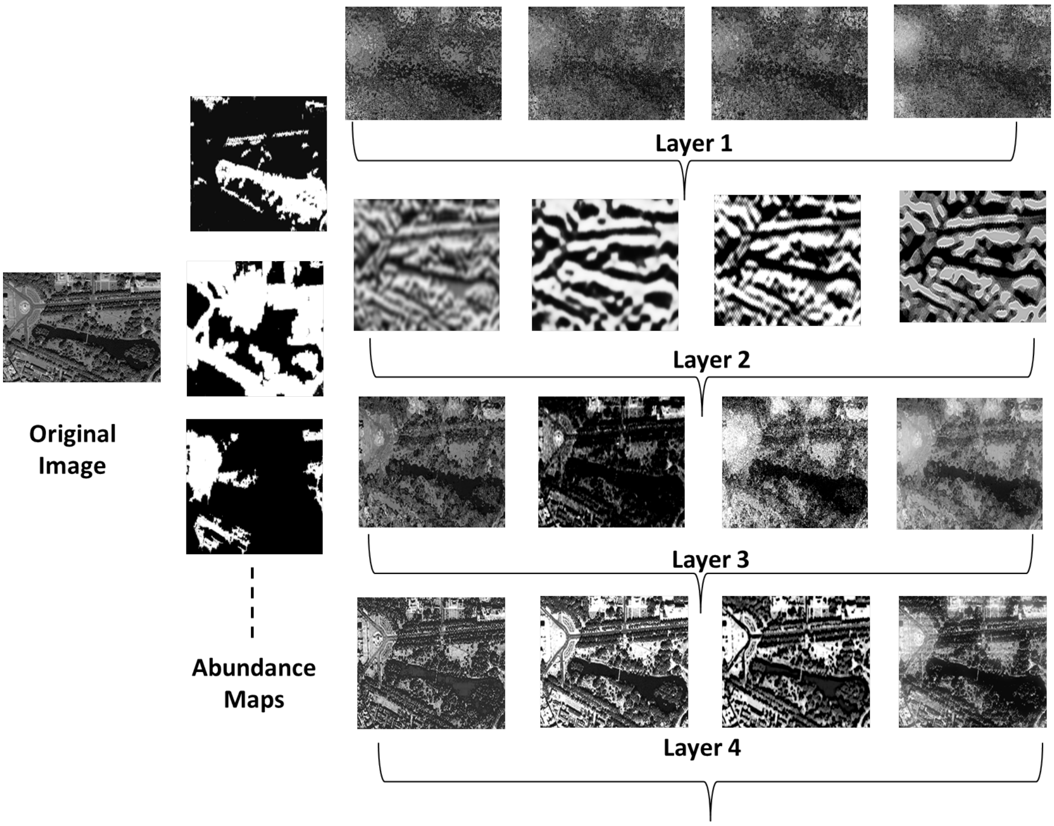


FIGURE 4.6: Working of CNN: sequence of abundance maps generated from the original image and fed into CNN to generate layer-wise features.

led to such misclassification. In the future, we plan to combine more distinctive features in the unmixing and classification approaches to solve such complicated cases.

4.6 Conclusions

In this chapter, we presented a CNN based classification model by incorporating unmixing results during the training procedure of the model. We extended an existing region based structure preserving nonnegative matrix factorization method to estimate the individual spectral responses from different materials in different groups of wavelengths. The estimated abundance maps of the materials were used as important features to generate superpixels. We further extended an existing superpixel extraction algorithm by introducing KD-estimated PDF to describe the spectral distribution of the superpixels and update the cluster centers accordingly. These abundance information-guided superpixels were provided as input to train an CRF-CNN integrated deep model. Instead of learning

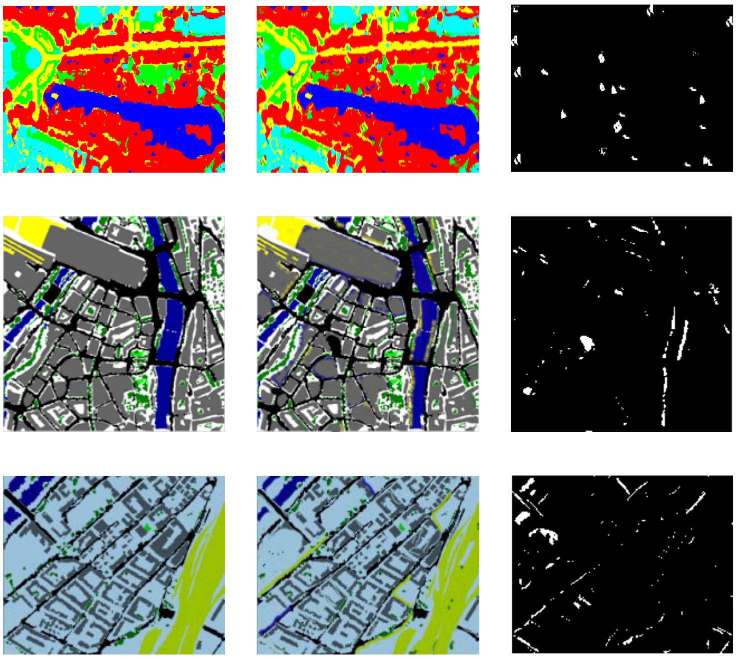


FIGURE 4.7: (a) Ground truth; (b) CNN-based classification; (c) difference map with ground truth.

from raw data, our proposed model receives significant spectral-spatial information in the data to produce better and powerful features so as to achieve improved classification performance. Comparison with several state-of-art methods shows the potential of using unmixing in deep learning-based classification framework.

Chapter 5

Triplet Constrained Generative Adversarial Networks for Hyperspectral Image Classification

In this chapter, we present a GAN-based spectral-spatial method for HSI classification. The proposed model adopts triplet constraints for the data and integrates them into the discriminator to improve its multi-class classification ability. Furthermore, the generator's capacity of producing fake samples is improved by providing intermediate feedback from the discriminator's features. We perform detailed experiments to support the idea of triplet-constrained GAN model in order to improve the classification performance.

5.1 Introduction

Deep models have contributed significantly in HSI classification as we demonstrated in the previous chapters. However, deep models suffer from the problem of overfitting due to limited training samples. Unfortunately, the issue of inadequate training samples is very common in remote sensing applications since collecting ground truth data is both time consuming and expensive. As a result, deep models often perform well during training but fail to accurately classify data during testing. We have proposed to generate virtual samples by using sample fusion and transformation operations in the previous chapter and significantly increased the number of training samples.

Instead of acquiring virtual samples in data preparation, the training samples can also be increased by integrating deep learning-based approaches into the classification framework. Generative Adversarial Networks (GANs) [115] is such a technique that can be adopted as a regularization method to overcome overfitting and increase training samples. GAN, first proposed in [115] by Goodfellow, trains a generator and a discriminator simultaneously. Most traditional deep learning-, based classification models are discriminative in nature, mapping a high-dimensional input to a much simpler output. A generative network instead tries to generate rich, high-dimensional outputs, for instance an image, from a relatively simpler input vector.

The idea introduced by Goodfellow is to train a discriminative model and a generative network simultaneously, by designing their respective cost functions such that their training procedures encourage them to compete against each other. Generator tries to generate new data that look like the original samples. Discriminator tries to discriminate between real data and the fake data from the generator. This discrimination is represented by mapping the data input to scalars that denote the probability that the corresponding input is real data. In the ideal case where the networks have sufficient representational power and the training process converges towards the global optimum, generator will produce samples with a distribution that matches the distribution of the real data, and discriminator will detect any subtle difference between the generated and the real distributions. Through the competition of both networks done in an adversarial manner, discriminator will be trained continuously and effectively. Hence, the problem of overfitting caused due to limited training samples can be avoided.

In spite of the promising architecture of the traditional GAN, very few works have been done on adopting GANs in the field of remote sensing. To handle the extremely time consuming process of labeling huge amount of remote sensing data, GANs can be adopted because the required quantity of training data may be provided by the generator during the training. In this regard, Lin *et al.* [22] proposed a multiple-layer feature-matching generative adversarial networks (MARTA GANs) to learn a representation using only unlabeled data. In [7], Xu *et al.* introduced the scaled exponential linear units instead of ReLU and batch normalization to produce high-quality and large-sized remote sensing images.

Some GAN-based approaches have been introduced for remote sensing classification task in a semi-supervised setting. He *et al.* [23] proposed a semi-supervised learning model in which three-dimensional bilateral filter (3DBF) was adopted to extract the spectral-spatial features from the hyperspectral data. The GANs were subsequently trained on those spectral-spatial features by adding samples from the generator to the features and increasing the dimension of the classification output. Ying *et al.* [24] proposed a semi-supervised 1D-GAN (HSGAN) to enable the automatic extraction of spectral features for HSI classification. In their proposed method, the model is trained on unlabled samples first to contain the features of all samples and then the model is transformed into a classification framework by adding a softmax layer.

Zhu *et al.* [140] introduced a 3D-GAN in which the generated fake samples were used with real samples to increase the number of training samples. The framework includes two schemes: spectral classifier and spectral-spatial classifiers. The adversarial training is adopted by a regularization technique. Along with the task of separating fake samples from real ones, the classification part also includes an additional softmax classifier to perform multi-class classification.

In the multi-class remote sensing image classification setting, the above mentioned works paid little attention to increasing the class-specific discrimination capability of the model. It is not entirely clear how the model is contributing to separating the real classes from fake ones in a GANs framework. Hence, the potential of GANs in a remote sensing classification task is limited to training sample augmentation, rather than improving the performance of multi-class classification.

In this context, spectral-spatial features can be further exploited to design important constraints for the training of GAN-based models to improve the classification capability of the discriminator in separating individual classes. Instead of using additional steps to improve classification performance, spectral-spatial characteristics of the available samples can be compared in the feature space to measure similarities between samples. In this regard, 3D Convolutional Neural Network (CNN) can be learned to build powerful embedding as feature vectors. This embedding is expected to provide useful cues for the subsequent classification of HSI data, i.e., enable minimizing the differences between samples in the same class and maximizing differences between samples in difference classes. This is the motivation of our work.

In this regard, the concept of "triplet constraint" [188] can be adopted to directly learn an embedding from data to a Euclidean space where the distances between samples correspond to data similarity. Here triplet refers to three data samples among which the distances will be measured. The network produces output as a compact embedding using the triplet-based loss function by measuring the distance between the triplet samples. The use of triplet loss in classification can be tricky because if applied naively, it may produce inaccurate results. An essential idea of learning with the triplet loss is the selection of triplets in order to ensure a stable training process. However, there is no standard procedure defined for selecting "good" triplets. Furthermore, mining hard triplets may cause unstable training too. We propose that spectral-spatial properties of the data can play a significant role in selecting the triplets and compute the loss accordingly.

In this chapter, we develop a novel GANs model for hyperspectral remote sensing data classification. With the parallel training processes of generator-discriminator, we increase the training samples and overcome the overfitting problem experienced by CNNs in image classification. Moreover, to improve the multi-class classification accuracy of the discriminator, we include triplet constraints into the loss function. We supply the samples in a batch and learn a Euclidean embedding for the samples using a 3D CNN network. To implement the idea, we extend the "triplet constraint" [188] to remote sensing images. We train the network in such a way that the squared distances in the embedding space correspond directly to the similarity between the samples. Those samples in the same class should have smaller distances and samples in different classes have larger distances. Then 3D CNN-based discriminator computes the loss of the model by separating the positive pair of samples from the negative sample using a distance margin.

In addition, to improve the quality of the fake samples, we propose to use intermediate features of the discriminator to build the perceptual loss. In this way, the generator is expected to produce fake data that are highly similar to the real data and eventually not recognizable as fake samples by the discriminator. We also propose to adopt Wasserstein distance, a function defined to measure the distance between two probability distributions, from WGAN [189] to formulate our objective function. To improve the optimization of the discriminator and to prevent the issue of weight clipping of WGAN, we propose to optimize the expectation using a softmax cross entropy. The main contributions of this paper are as follows:

- A 3D-GAN architecture ("Triplet-3D-GAN") is proposed for classifying remote sensing images by employing GANs and 3D CNN.
- Spectral-spatial triplet constraints are included in generating real samples during the training in order to improve the multi-class classification ability of the discriminator.
- Performance of the generator is further improved by receiving feedback of the intermediate features from the discriminator.
- The adversarial samples are used along with the real samples to adjust the training of the proposed GANs model and thus, address the overfitting problem of CNN for remote sensing images.
- Under the setting of using small number of training samples, our model is tested on standard hyperspectral datasets and achieves the state-of-the-art performance.

The rest of this chapter is organized as follows. Section 5.2 gives a detailed overview of the standard GAN formulation and its variants that are adopted in this work. Section 5.3 provides a description of the proposed triplet construction method for the formulation of discriminator's loss function. Section 5.4 introduces the detailed formulation of our proposed GAN architecture. Section 5.6 presents the experimental results and finally, conclusions are drawn in Section 5.7.

5.2 Background on GAN

A GANs model trains two networks, a discriminator D and a generator G, simultaneously in an adversarial manner. G captures the data distribution and D tries to differentiate between fake samples from real ones by estimating the probability on whether a sample comes from the real data or G. To learn a generator distribution p_g over data x, G samples noise z and builds a mapping function $G(z, \theta_g)$ from a prior noise distribution p_z to the data space, and produces fake samples $\hat{x}$. D receives either real data x or fake samples $\hat{x}$ as input and emits a probability indicating whether the received data is a real training sample or a fake sample drawn from the fake distribution p_z.

During the training of D, the parameters of D are adjusted in order to assign correct labels to both real and fake samples while the weights of G are kept fixed. For every sample $x \in p_r$, where p_r is the data distribution over real sample x, the goal of the training is to maximize $D(x)$ for every sample from the real data and minimize $D(G(z))$ for every sample from the fake samples drawn from G's fake distribution p_z. Therefore, the adversarial loss function of the model during the training of D is expressed as follows:

$$\mathcal{L}(G,D) = E_{x\sim p_r}[logD(x)] + E_{\hat{x}\sim p_z}[log(1-D(G(z)))] \tag{5.1}$$

Here, $\mathcal{L}(G,D)$ is the loss function of the GAN and $E(.)$ is the expectation operator.

During the training of G, the weights of D are kept fixed and the parameters of G are adjusted in order to minimize $log(1-D(G(z)))$ because D produces a probability estimation that ranges between 0 and 1. This is called perceptual loss as it encourages the generated samples to be similar to the samples drawn from the real data distribution. Hence, the aim of the optimization during the training of a GAN is to solve a mimimax problem that is defined as follows:

$$\underset{G}{min}\ \underset{D}{max}\mathcal{L}(G,D) = E_{x\sim p_r}[logD(x)] + E_{\hat{x}\sim p_z}[log(1-D(G(z)))] \tag{5.2}$$

Fig. 5.1 illustrates a typical architecture of a GAN-based classification framework.

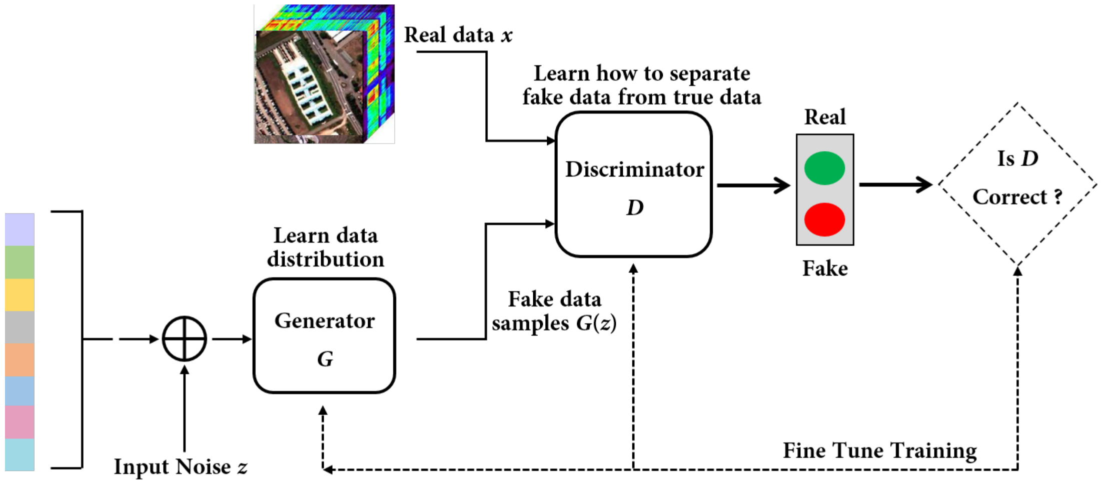


FIGURE 5.1: A typical GAN-based classification framework

However, if D is optimally trained before each generator parameter update, it is guaranteed that $D(x) = 1, \forall x \in p_r$ and $D(x) = 0, \forall x \in p_z$. As a result, the loss function falls to zero and it ends up with no gradient to update the loss during learning iterations. To address this issue, Arjovsky *et al.* [189] proposed an improved GAN called WGAN in which the Earth-Mover (also called Wasserstein-1) distance was used to measure the distance between two distributions as the minimum necessary work to transform one distribution to the other. Since it is intractable to exhaust all the possible joint distributions in $\prod(p_r, p_z)$ to compute $inf_{\gamma \sim \prod(p_r,p_z)}$ (γ is the expected cost of the travelling distance of two points from two distributions), a smart transformation of the formula was proposed based on the Kantorovich-Rubinstein duality to obtain [189]:

$$\underset{G}{min}\,\underset{D\in\kappa}{max}\,\underset{x\sim p_r}{E}[D(x)] - \underset{\tilde{x}\sim p_z}{E}[D(\tilde{x}))] \tag{5.3}$$

where κ is the set of 1-Lipschitz functions. The WGAN results in a critic function whose gradient with respect to its input performs well compared to the standard GAN as the optimization of the generator is easier.

Despite the improvement on the training of GANs, poor samples can still be generated or the model may fail to converge. This happens mainly due to weight clipping within a compact space in WGAN to enforce a Lipschitz constraint which may lead to undesired behaviour. To further improve the training, Gulrajani *et al.* [190] proposed an alternative clipping weights by penalizing the norm of gradient of the critic with respect to its input. Hence, the new objective function is defined as:

$$\mathcal{L} = \underset{\tilde{x} \sim p_z}{E}[D(\tilde{x}))] - \underset{x \sim p_r}{E}[D(x))] + \lambda \underset{\hat{x} \in p_z}{E}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2] \tag{5.4}$$

The first term is the original critic loss and the second term is the gradient penalty. During a training iteration, an interpolated sample is drawn randomly from anywhere on the straight line segment between a real sample and a generated sample. The gradient of D at this point is evaluated and penalizes proportionally to the square deviation from a gradient of 1. This enforces the WGAN requirement that D respects the 1-Lipschitz constraint [190].

5.3 Construction of Triplet Constraint

In this method, we intend to construct an embedding $f(x)$ from a real sample x into a feature space in a way to measure the squared distance between all samples such that the distance between samples belonging to the same class is small and distance between samples of difference classes is large. We design a loss function with an end-to-end learning inside a GAN. The motivation is that the loss encourages the samples of the same class to be projected onto a single point in the embedded space. Moreover, the loss also tries to enforce a margin between each pair of samples from the same class to all other samples. In this way, a manifold is formed by containing the samples for one class and enforcing the discrimination against other classes at the same time.

5.3.1 Triplet Loss

A key concept in our model is the triplet [188]. Each triplet has three samples, with two samples belonging to the same class (positive samples) and one sample belonging to a different class (negative sample). One of the positive samples is termed as an “anchor” sample to which the distance will be compared. In the embedding process, x is embedded into a D-dimensional Euclidean space. Given an image sample x_i^a (anchor), $i \in 1, \cdots, M$ where i is the index of the triplet and M is the number of all possible triplets of samples, we enforce a relationship so it is closer to all positive samples x_i^p than it is to any other negative sample x_i^n (negative) along the spectral channel $\lambda \in \mathbb{B}$, where $\mathbb{B}$ is the set of spectral bands. Formally, this relationship on a triplet $f(x_i^a)_\lambda, f(x_i^p)_\lambda, f(x_i^n)_\lambda \in \mathbb{T}$ is:

$$||f(x_i^a)_\lambda - f(x_i^p)_\lambda||_2^2 + \alpha < ||f(x_i^a)_\lambda - f(x_i^n)_\lambda||_2^2 \tag{5.5}$$

where α is a margin that is enforced between the positive and negative pair of samples and $\mathbb{T}$ is the set of all possible triplets of samples in the training set. Therefore, the triplet loss to be minimized is calculated as:

$$L_t = \sum_i^M (||f(x_i^a)_\lambda - f(x_i^p)_\lambda||_2^2 - ||f(x_i^a)_\lambda - f(x_i^n)_\lambda||_2^2 + \alpha) \tag{5.6}$$

In this way, many triplets are generated to fulfil the constraint in Eq. (5.5). However, using all possible triplets will cause slow convergence and not every triplet may contribute. Hence, it is important to select effective triplets accordingly.

5.3.2 Selecting Triplets

The purpose of including triplet constraints is to improve the multi-class classification performance of our proposed GAN-based model by learning a feature embedding network to extract spectral-spatial features from HSI data. In order to ensure a fast convergence, it is important to select triplets that violate the triplet constraint property defined in Eq. (5.5). This means that we want to select positive samples that satisfy $\textbf{argmax}_{x_i^p}||f(x_i^a)_\lambda - f(x_i^p)_\lambda||_2^2$ in which case the obtained features can discriminate samples with large variation from the same class (i.e., hard positive examples). Similarly, we want to select negative samples that satisfy $\textbf{argmin}_{x_i^n}||f(x_i^a)_\lambda - f(x_i^n)_\lambda||_2^2$ in which case the obtained features can discriminate samples that are spectrally similar but from different classes (i.e., hard negative examples).

Since it is not computationally feasible to obtain **argmax** and **argmin** across the entire training set, we divide our training set into several subsets and save our 3D CNN network in every n step during the training of those subsets to mark network checkpoints. Then, we generate triplets in every n step using the most recent network checkpoint and compute the **argmax** and **argmin** on the subset accordingly. We use small mini-batches to supply the samples which should be a meaningful representation of the positive anchor and negative anchor distances.

In [191], we selected the anchor sample in a random manner and computed the distances accordingly. Selecting anchor sample randomly has some disadvantages. If the selected sample itself is not an effective representative of useful spectral information, the triplet constraint property with other samples in the mini-batch may fail. As a result of not selecting a "good" anchor sample, it introduces chances of obtaining zero or less effective pair of anchor-positive and anchor-negative samples. On the other hand, there may be other samples in the mini-batch which can better describe the spectral properties of the respective classes. Selecting those samples should produce more useful triplets and can improve the multi-class classification of the discriminator significantly.

Therefore, we extend this approach of selecting anchor samples by considering each sample in the mini-batch as an anchor sample. In this way, every sample in the mini-batch will be regarded as an anchor sample once and the distance to the positive and negative samples will be computed accordingly. By doing so, the possibility of selecting good anchor-positive and anchor-negative pair of samples is higher.

A major limitation of the triplet loss is that as the dataset gets larger, the possible number of triplets grows cubically, resulting in an expensive and impractical training. Therefore, it is very important to mine "hard" triplets for learning. In our cases, spectrally similar but different materials can help in building better triplets and hence, we select hard negative samples to form the triplets in our method. In this regard, in case of selecting the positive samples, we consider all anchor-positive pairs in the mini-batches instead of selective anchor-positive pairs to utilize such local structure information of the data.

In case of selecting negative samples, we consider the samples which are spectrally close to the positive samples since it will introduce challenges to the training process. The local minima caused during the training of the model while selecting the hardest negative samples can result in a collapsed model ($f(x) = 0$) [188]. Therefore, we select the negative samples as follows:

$$||f(x_i^a)_\lambda - f(x_i^p)_\lambda||_2^2 < ||f(x_i^a)_\lambda - f(x_i^n)_\lambda||_2^2 \tag{5.7}$$

In this way, we consider the negative samples which are further away from the anchor than the positive samples but the squared distance is still close to the anchor-positive distance. Since these negative samples lie within α, we consider samples whose spectral properties are close to the positive samples. Fig. 5.2 illustrates the process of learning triplet constraint-based embedding by a 3D-CNN.

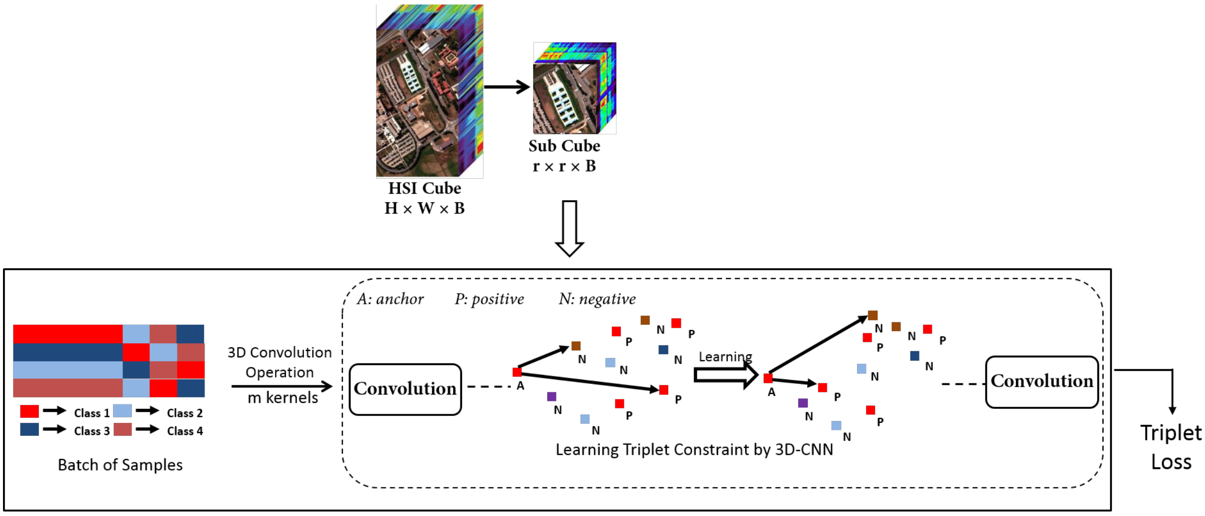


FIGURE 5.2: Learning triplet constraint-based embedding through 3D-CNN

5.4 Formulation of Triplet Constraint-based 3D-GAN

In this section, we present a deep architecture based on the theory of WGAN, whose objective function includes additional loss functions to integrate triplet constraints. We supply training data that satisfy triplet constraint property to the generator. The purpose of doing so is to increase the multi-class classification capabilities of the discriminator since triplets are believed to be useful estimation of spectral-spatial characteristics of HSI data and may contribute considerably in accurate classification. Therefore, the parameters are optimized based on the multi-classification loss. We now present the loss functions of G and D to be used in our proposed model.

5.4.1 Designing Loss Functions

In our model, we supply label of the data to both D and G. The label is expected to add useful information that maximizes the classification abilities of D and generative abilities of G.

G produces fake samples, $\hat{x} = G(z, \theta_z)$ with parameters θ_z, where z obeys a prior noise distribution p_z. G also receives the class labels of our original training data $\mathbb{C} = \{y_1, y_2, \ldots, y_Y\}$, where Y is the number of classes of the HSI data. However, to obtain a more effective perceptual loss, we propose to use the intermediate features from D to build the perceptual loss. In this way, the computational overhead is reduced through the reuse of the extracted features by D. We define the perceptual loss for G as:

$$\mathcal{L}_\rho = \sum_{i=1}^{L}(||E_{x \sim p_r} f_i(x) - E_{\hat{x} \sim p_z} f_i(G(z))||_2^2) \tag{5.8}$$

where, L is the number of convolution layers, $f_i(x)$ is the feature map computed by the i-th convolution layer (after the activation function layer) within D.

In our model, D receives both real data x which are selected after satisfying triplet constraints and fake data $\hat{x}$ generated by G. Therefore, it is very crucial for D to separate real samples from the generated fake ones. Hence, D tries to maximize the log-likelihood of the correct source of data $\mathbb{U} = \{real, fake\}$.

$$\mathcal{L}_{\mathbb{U}} = E[logP(\mathbb{U} = real|x)] + E[logP(\mathbb{U} = fake|\hat{x})] \tag{5.9}$$

The role of D in our model is not only just to differentiate real samples from fake ones but also to accurately classify different classes of the data. We use Wasserstein distance from WGAN [189] to formulate objectives for optimization of D's multi-class classification task. As mentioned in Section 5.2, discriminators in WGAN are constrained to be 1-Lipschitz functions and their losses are constructed using the Kantorovich-Rubinstein duality. Therefore, we formulated the improved WGAN-GP [190] in which a gradient penalty term was included to address the extremely distributed weights because of the weight clipping scheme. Along with this, we propose to remove the last activation in D to prevent the weights from growing too large and we optimize the expectation using softmax cross-entropy. By imposing the gradient penalty [190] together with the removal of the last activation, softmax cross-entropy and BN, an effective and efficient approximation of K-Lipschitz function can be done during D's multi-class classification task.

Therefore, the objective function of the discriminator includes another term containing the log-likelihood of the correct class labels of our HSI data:

$$\mathcal{L}_{\mathbb{C}} = E[logP(\mathbb{C} = y|x)] + E[logP(\mathbb{C} = y|\hat{x})] + \delta \underset{\hat{x} \in p_z}{E}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2] \tag{5.10}$$

where δ is the gradient penalty co-efficient.

As mentioned in Section 5.3.2, we apply triplet constraints on the training samples to select triplets for D to improve its multi-class classification capability. Considering the computation overhead caused by all possible combinations of the triplets, we propose an alternative organization to the standard way of using the triplet loss. The basic idea is: we randomly sample C classes ($C < Y$ where Y is the total number of classes) and then randomly select S samples of each class, thus resulting in a batch of CS samples. Now, for each sample s in the batch, we select all positive and the hard negative samples within the batch when forming the triplets for the loss:

$$\mathcal{L}_{triplet} = \overbrace{\sum_{y=1}^{Y} \sum_{s=1}^{S}}^{\text{All Anchors}} \left[\alpha + \overbrace{\sum_{\substack{p=1 \\ p \neq a}}^{S} ||x_i^a - x_i^p||_2^2}^{\text{All Positives}} - \overbrace{\underset{\substack{j=1,\ldots,Y \\ n=1,\ldots,S \\ j \neq i}}{min} ||x_i^a - x_j^n||_2^2}^{\text{Hard Negatives}} \right] \tag{5.11}$$

where a, p and n denote the anchor, positive and negative samples respectively.

Our proposed approach of constructing triplets in our model is summarized in Algorithm 5.

Adding the triplet loss to the objective function (Eq. 5.9) of D for evaluating correct class labels gives us the overall objective function for our proposed GAN model as:

$$\mathcal{L} = \mathcal{L}_{\mathbb{U}} + \mathcal{L}_{\mathbb{C}} + \mathcal{L}_{triplet} + \mathcal{L}_{\rho} \tag{5.12}$$

Algorithm 5: Triplet Construction Algorithm

```
Data: T Input Samples $\{X_1, X_2, \ldots, X_T\}$, $Y$ target classes in $\{y_1, y_2, \ldots, Y\}$
while training sample $i : 1 \rightarrow T$ do
  for $b = 1, 2, \ldots, B$ do
    Select $C$ classes where $C < Y$
    for $i = 1, 2, \ldots, C$ do
      Draw $S$ samples and add them into Batch $B_b$
    end for
  end for
  for $b = 1, 2, \ldots, B$ do
    while $length(B_b) < S$ do
      Randomly select $x_i, x_j, x_k \in B_b$
      if $x_i \neq$ anchor then
        Select $x_i \leftarrow x_i^a$ as an anchor sample
      end if
      Select $x_j \leftarrow x_j^p$ as a positive sample, where label$(x_i)$ = label$(x_j)$
      Select $x_k \leftarrow x_k^n$ as a negative sample, where label$(x_j) \neq$ label$(x_k)$
      Compute $\sum_{\substack{p=1 \\ p \neq a}}^{S} ||x_i^a - x_i^p||_2^2$
      Compute $\underset{\substack{j=1,\ldots,Y \\ n=1,\ldots,S \\ j \neq i}}{min} ||x_i^a - x_j^n||_2^2$
      Compute $\mathcal{L}_{triplet}$ using Eq. (5.11)
    end while
  end for
end while
Output: Generated Triplets
```

5.5 Network Architecture

We propose a 3D-GAN architecture in this chapter, specifically designed for improving the multi-class classification capability of the model. Both generator and discriminator in our model are in the form of convolutional networks. Our proposed Triplet-3D-GAN extracts effective spectral-spatial characteristics of the HSI data and achieves better classification performance. We now present the architecture of our model followed by describing the classification process.

5.5.1 Generator Architecture

The generator network of our model takes a vector of 100 random numbers drawn from a uniform noise distribution as input and produces an image of size $H \times W$ which is the same size as the real data. The generator network consists of a fully connected layer reshaped into a tensor and used as an input to the convolution stack containing number of fractionally-strided convolutional layers or deconvolution layers. A fractionally-strided convolution can be interpreted as expanding the pixels by inserting zeros in between them. Convolution over the expanded image will produce a larger or upsampled image. We apply BN to each layer of the network, except for the output layer. In order to prevent the generator from collapsing all samples to a single point, we normalize the responses to have zero mean and unit variance over the entire mini-batch, as done in [192]. LeakyReLU activation functions are performed at each layer (not in the output layer) in order to allow gradients to flow backwards through the layer without constrain. The generator network finally produces an image with five channels in the spectral domain, as shown in Fig. 5.3.

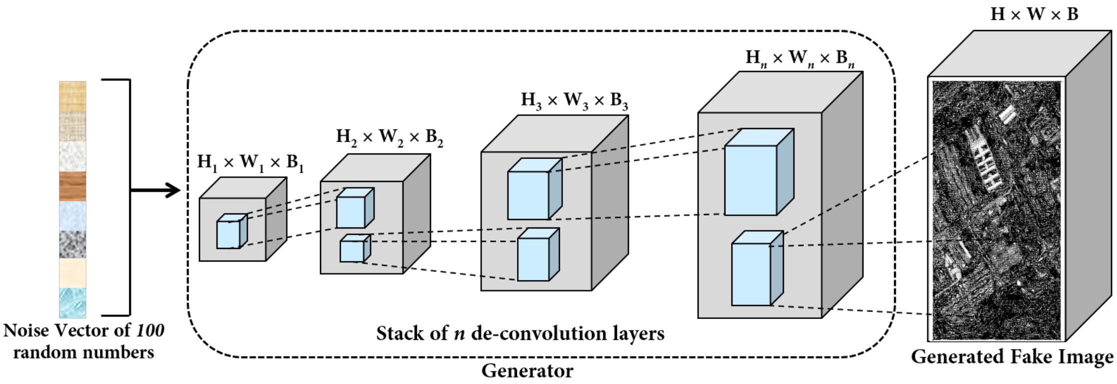

H × W × B
$H_n \times W_n \times B_n$
$H_3 \times W_3 \times B_3$
$H_2 \times W_2 \times B_2$
$H_1 \times W_1 \times B_1$
Noise Vector of *100* random numbers
Stack of *n* de-convolution layers
Generator
Generated Fake Image

FIGURE 5.3: Architecture of the generator.

5.5.2 Discriminator Architecture

In the proposed 3D-GAN model, the discriminator adopts a 3D CNN architecture [193]. During the 3D-CNN training, we use kernels of size $1 \times 1 \times d$, where $d > 1$, to learn the spectral features from the original datacube and to reduce the high dimensions of the HSI data. With this kernel size, the relationship between pixels and their neighbors in the spatial domain is not considered, but the convolution of a kernel size of 1×1 still has the ability to integrate linear combinations of pixels in the spatial neighborhood. Hence, convolution of kernel size of $1 \times 1 \times d$ can extract spectral features and preserve spatial information. During the convolution operations, we use the tradition padding "Same" and "Valid" to reduce the dimensions of the feature maps. The "Same" padding reserves the boundary and produces an output size to be the same as the input size when the stride is one. On the other hand, the "Valid" padding does not apply any padding and assumes that all dimensions are valid so that the input image fully gets covered by the specified kernels and stride. The latter is used to reduce the dimensions of the feature maps and retain extracted spectral features and spatial information.

We illustrate the proposed discriminator architecture in Fig. 5.4. Given an HSI **H** with B channels, a small cube of size $r \times r \times B$ is selected from the original real data as the input to D. Upon receiving the initial input, the first convolution layer which has n kernels of $1 \times 1 \times L$ $(L < B)$ with stride (1, 1, 2) and the "valid" padding, generates n feature maps, each of size $r \times r \times b$. This convolution operation reduces the number of bands to $b = \frac{B-L+1}{2}$. These resulting feature maps are supplied to a block consisting of a stack of l convolution layers. For each of the convolution layers inside this block, we specify k kernels of size $1 \times 1 \times b$ with the "same" padding and stride $1 \times 1 \times 1$. The output of each convolution layer are k feature maps of size $r \times r \times b$. Merging all initial input and output feature maps gives $n + (k \times l)$ feature maps. The resulting output feature maps are further reshaped to produce one feature map with a size of $r \times r \times b$.

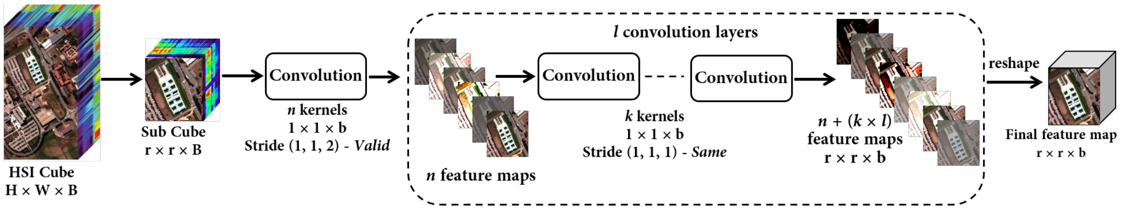


Figure 5.4: Architecture of the discriminator.

5.5.3 Generative Adversarial Samples for Classification

The proposed 3D-GAN is particularly designed for increasing the multi-class classification ability of the discriminator through the use of triplet constraints. D receives both real and fake samples as input with the generated fake data be taken as augmented training samples. However, we apply triplet property for the real data only. In addition to Y for the real data, we add a new class label in order to specify every generated fake sample.

To start with, the generated fake samples are forwarded through the network and are assigned labels by computing the maximum values of the probability vectors. These fake samples can then be used for training the network with the assigned labels, thus increase the number of training samples. Because the fake samples do not belong to any real class, the additional class label is used to classify these samples, making the model a $Y + 1$ classification problem. D uses sigmoid classifier to distinguish real and fake samples and uses a cross-entropy softmax classifier to give the multi-class classification results. The entire process of the proposed 3D GAN-based classification framework is illustrated in Fig. 5.5. Our proposed 3D-GAN architecture is summarized in Algorithm 6.

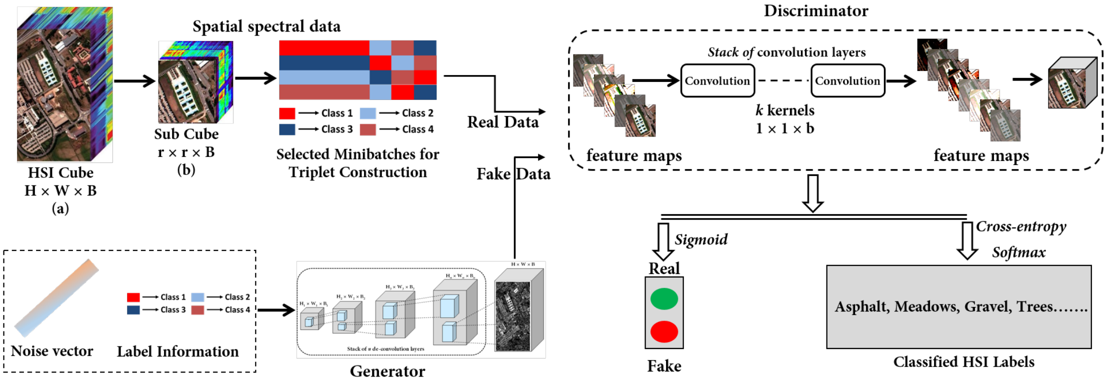


Figure 5.5: 3D GAN-based HSI classification framework

5.6 Experiments

In this section, we present the experimental results on real-world hyperspectral remote sensing images. Then we analyse the performance of the proposed method in comparison with several alternatives. For better evaluation, we organize our experiments in the following stages:

1. We evaluate the effectiveness of the feature extraction stage of a standard GAN model by comparing its classification performance with other feature extraction methods.

Algorithm 6: Triplet Constrained 3D-GAN

```
Data: Gradient penalty co-efficient $\delta$, Number of discriminator iterations $n_D$,
      initial discriminator parameters $\theta_D$, initial generator parameters $\theta_G$, batch
      size $m$ and Adam hyper-parameters $\varrho, \beta_1, \beta_2$.
/* Initialization */
Set default values: $\delta = 10$, $n_D = 5$, $\varrho = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$.
while $\theta_G$ not converged do
  for $t = 1, 2, \ldots, n_D$ do
    for $i = 1, 2, \ldots, m$ do
      Sample a batch from the real data based on $\{x^{(i)}\}_{i=1}^{m} \sim P_r$
      Sample a batch of fake samples based on $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$
      Compute $\mathcal{L}^{(i)} \leftarrow \mathcal{L}_{\mathbb{U}} + \mathcal{L}_{\mathbb{C}} + \mathcal{L}_{triplet}$ using Eqs. (5.9), (5.10) & (5.11)
    end for
    Compute $\theta_D \leftarrow Adam(\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}^{(i)}, \theta_D, \varrho, \beta_1, \beta_2)$
    Compute Sigmoid activation
    Compute Soft-max activation $a = \frac{exp(o)}{\sum_k exp(o_k)}$; where $o$ is the output of the final
    layer of the network and first input to softmax classifier
    Compute error $T = \mathbf{y}_i$-$\mathbf{a}$
    Back-propagate error to compute gradient $\frac{\delta T}{\delta o_j}$
    Update network parameter $\theta_D$
  end for
  Sample a batch of fake data based on $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$
  Compute $\theta_G \leftarrow Adam(\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} -D(G(z)), \theta_G, \varrho, \beta_1, \beta_2)$
end while
Output: Trained Discriminator parameters $\theta_D$
```

2. We evaluate the performance of our proposed triplet constraint-based GAN model by comparing its classification performance with other baseline spectral-spatial classification methods.

3. We evaluate the important stages of our model in order to verify the usefulness of those stages.

5.6.1 Hyperspectral Image Datasets

In the experiments, we used three widely used hyperspectral datasets, i.e., Indian Pines, Pavia University and Salinas, in order to evaluate the effectiveness of our proposed method. For better evaluation of our proposed method, we used a new dataset "Griffith-USGS" that we introduced in Chapter 3. The number of available labelled samples per every class are provided in Tables 5.1, 5.2, 5.3 and 5.4. To select the training and testing samples, we followed the same experimental setting described in 3D-GAN [140]

Table 5.1: Total Number of Available Samples for Each Class on Indian Pines Dataset

Class	Samples
Alfalfa	46
Corn-notill	1428
Corn-mintill	830
Corn	237
Grass-pasture	483
Grass-trees	730
Grass-pasture-mowed	28
Hay-windrowed	478
Oats	20
Soybean-notill	972
Soybean-mintill	2455
Soybean-clean	593
Wheat	205
Woods	1265
Buildings-Grass-Trees-Drives	386
Stone-Steel-Towers	93
Total	10249

Table 5.2: Total Number of Available Samples for Each Class on Pavia University Dataset

Class	Samples
Asphalt	6631
Meadows	18649
Gravel	2099
Trees	3064
Painted metal sheets	1345
Bare Soil	5029
Bitumen	1330
Self-Blocking Bricks	3682
Shadows	947
Total	42776

to make a fair comparison with our method. During the training of GAN, we use 200 training samples to learn weights and biases of each neuron, and 100 training samples as validation samples that are used to guide the design of proper architectures and to identify whether the network is overfitted or not. For testing purposes, we use all the samples in the data sets to verify the effectiveness of the trained network. We use a very limited number of real samples for training and augment the training samples during the course of the GAN training by generating fake samples.

Table 5.3: Total Number of Available Samples for Each Class on Salinas Dataset

Class	Samples
Alfalfa	2009
Corn-notill	3726
Corn-mintill	1976
Corn	1394
Grass-pasture	2678
Grass-trees	3959
Grass-pasture-mowed	3579
Hay-windrowed	11271
Oats	6203
Soybean-notill	3278
Soybean-mintill	1068
Soybean-clean	1927
Wheat	916
Woods	1070
Buildings-Grass-Trees-Drives	7268
Stone-Steel-Towers	1807
Total	54129

Table 5.4: Total Number of Available Samples for Each Class on Griffith-USGS Dataset

Class	Samples
Road	1734
Water	1574
Building	1481
Grass	1917
Tree	1772
Soil	1416
Total	9894

5.6.2 Design of the 3D-GAN Architecture

The detailed design of the generator G and the discriminator D in our model is discussed in this section. G took 100 random numbers drawn from a uniform distribution as an input of size $100 \times 1 \times 1$. The first layer of G was fully connected which is just a matrix multiplication. The result was shaped into a 3D tensor. After that five convolution layers were used to learn its own spatial upsampling and upsample the feature maps to produce a fake sample with a size of $64 \times 64 \times B$ (B is the number of bands for each individual dataset). BN was used in each layer except the last layer. Table 5.5 presents the architectures of the generators used for all datasets.

Table 5.5: Architectures of the generators on all datasets

Dataset	Layer	Convolution	BN	Stride	Padding	Activation Function
Indian Pines, Griffith-USG	1	5 × 5 × 1024	Yes	1 × 1 × 1	No	LeakyReLU
	2	5 × 5 × 512	Yes	1 × 1 × 2	Same	LeakyReLU
	3	5 × 5 × 256	Yes	1 × 1 × 2	Same	LeakyReLU
	4	5 × 5 × 256	Yes	1 × 1 × 1	Same	LeakyReLU
	5	5 × 5 × 200	No	1 × 1 × 2	Same	LeakyReLU
Pavia University	1	5 × 5 × 1024	Yes	1 × 1 × 1	No	LeakyReLU
	2	5 × 5 × 512	Yes	1 × 1 × 2	Same	LeakyReLU
	3	5 × 5 × 256	Yes	1 × 1 × 2	Same	LeakyReLU
	4	5 × 5 × 128	Yes	1 × 1 × 2	Same	LeakyReLU
	5	5 × 5 × 103	No	1 × 1 × 2	Same	LeakyReLU
Salinas	1	5 × 5 × 1024	Yes	1 × 1 × 1	No	LeakyReLU
	2	5 × 5 × 512	Yes	1 × 1 × 2	Same	LeakyReLU
	3	5 × 5 × 256	Yes	1 × 1 × 2	Same	LeakyReLU
	4	5 × 5 × 256	Yes	1 × 1 × 1	Same	LeakyReLU
	5	5 × 5 × 204	No	1 × 1 × 2	Same	LeakyReLU

For D, the first layer was supplied with both real and fake samples. The first two convolution layers used a $1 \times 1 \times L$ kernel with "zero" padding and a stride of 2. The rest of the convolution layers used a $1 \times 1 \times b$ kernel with "same" padding and a stride of 1. BN was used in specific layers of the discriminator network as using it in each layer may cause instability. We added a sigmoid classifier and a softmax classifier in parallel at the end which were used to classify real/fake samples and individual classes of the HSI respectively. Table 5.6 presents the architectures of the discriminators used on all datasets. The size of the mini batch supplied to D was 100 and the learning rate was set to 0.0002. The number of epochs was 600 for Indian Pines and Salinas and 700 for Pavia University and for Griffith-USGS.

5.6.3 Feature Extraction by GAN

In this section, we evaluate the feature extraction stage of a standard GAN by comparing the classification performance with other feature extraction methods. For classification step, we used several widely used classifiers and report their classification performance.

During our experiments, we used several feature extraction methods such as LDA, PCA, ICA and CNN. LDA is a supervised dimensionality reduction method which projects the input data to a linear subspace consisting of the directions to maximize the separation

Table 5.6: Architectures of the discriminators on all datasets

Dataset	Layer	Convolution	BN	Stride	Padding	Activation Function
Indian Pines, Griffith-USGS	1	1 × 1 × 127	No	1 × 1 × 2	Valid	LeakyReLU
	2	1 × 1 × 18	No	1 × 1 × 2	Valid	LeakyReLU
	3	1 × 1 × 10	Yes	1 × 1 × 1	Same	LeakyReLU
	4	1 × 1 × 10	Yes	1 × 1 × 1	Same	LeakyReLU
	5	1 × 1 × 10	Yes	1 × 1 × 1	Same	No
	6	Nodes to classify real/fake				Sigmoid
		Nodes to classify multi classes				Softmax
Pavia University	1	1 × 1 × 30	No	1 × 1 × 2	Valid	LeakyReLU
	2	1 × 1 × 18	No	1 × 1 × 2	Valid	LeakyReLU
	3	1 × 1 × 10	Yes	1 × 1 × 1	Same	LeakyReLU
	4	1 × 1 × 10	Yes	1 × 1 × 1	Same	LeakyReLU
	5	1 × 1 × 10	Yes	1 × 1 × 1	Same	No
	6	Nodes to classify real/fake				Sigmoid
		Nodes to classify multi classes				Softmax
Salinas	1	1 × 1 × 131	No	1 × 1 × 2	Valid	LeakyReLU
	2	1 × 1 × 18	No	1 × 1 × 2	Valid	LeakyReLU
	3	1 × 1 × 10	Yes	1 × 1 × 1	Same	LeakyReLU
	4	1 × 1 × 10	Yes	1 × 1 × 1	Same	LeakyReLU
	5	1 × 1 × 10	Yes	1 × 1 × 1	Same	No
	6	Nodes to classify real/fake				Sigmoid
		Nodes to classify multi classes				Softmax

between classes. The capability of linearly extracting spectral and spatial features has made LDA a reasonable choice for extracting features of HSIs. PCA, a statistical procedure, is one of the most popular dimensionality reduction methods for HSIs. Since the neighboring bands in an HSI are highly correlated, PCA can effectively transform the original data to remove the correlation among the bands by using orthogonal transformations. ICA has also been exploited in hyperspectral remote sensing data analysis. It is based on a well known unsupervised blind source separation (BSS) technique, which identifies statistically independent components by considering only the observation of mixture signals. The independent components can provide useful information related to one or several classes in an HSI.

For deep feature extraction, we chose 1D-CNN architecture to obtain spectral features only. At first, we reduced the number of spectral bands of the original data cube to 10 by PCA. One-dimenstional vectors representing the pixels in the HSI were supplied as input to the model. It consisted of several convolutional and pooling layers. Logistic regression (LR) was applied to adjust the weights and biases in the back-propagation during training. We used two convolution and two pooling layers for all datasets. After the training, the learned features were used in conjunction with different classifiers.

To extract features with GAN, we also chose the architecture of a 1D-GAN to produce spectral features only. All the training data and input noise were in the form of spectral vectors. For a fair comparison, the number of spectral bands of the real data were reduced to 10 by PCA first. G of the model received an input noise of size $1 \times 1 \times 1$ which was passed through three deconvolution layers to be converted into $10 \times 1 \times 1$. D received both fake and real samples as the training data with corresponding labels. A stride of 1 and zero padding were used during the experiment. Three convolution layers were used and BN was included in specific layers as well.

In the experiments, we used RF, SVM, and KNN as the classifiers with the above-mentioned features. The parameter tuning procedure was previously explained in Section 4.5.2 in Chapter 4.

Tables 5.7, 5.8 and 5.9 report the classification accuracies in terms of OA(%) and AA(%) obtained with different feature extraction and classification on all datasets. We report the classification results in the form of mean ± standard deviation. Table 5.7 shows that the deep learning-based models achieve better classification accuracy compared to LDA, PCA and ICA. GAN features, particularly, achieves superior performance over all other feature extraction methods.

Our detailed observations based on the overall accuracy results are given below:

- With SVM as the classifier, GAN features outperforms other feature extraction methods by $5\% - 14\%$ on Indian Pines and Pavia University, $5\% - 15\%$ on Salinas and $7\% - 14\%$ on Griffith-USGS. With RF as the classifier, GAN features outperforms other feature extraction methods by $3\% - 10\%$ on Indian Pines and Pavia University, $4\% - 8\%$ on Salinas and $5\% - 10\%$ on Griffith-USGS. With KNN as the classifier, GAN features outperforms other feature extraction methods by $6\% - 13\%$ on Indian Pines, $9\% - 14\%$ on Pavia University, $7\% - 15\%$ on Salinas and $4\% - 13\%$ on Griffith-USGS.

Table 5.7: Classification accuracies obtained by support vector machine with different feature extraction methods on all datasets. Best accuracies are shown in bold.

Dataset		LDA	PCA	ICA	CNN	GAN
Indian Pines	OA (%)	78.10 ± 1.35	74.17 ± 2.20	79.29 ± 2.47	82.90 ± 2.80	**88.12 ± 1.29**
	AA (%)	74.53 ± 2.80	71.30 ± 1.58	76.00 ± 2.69	79.76 ± 1.15	**84.16 ± 1.35**
Pavia University	OA (%)	83.97 ± 1.70	76.24 ± 1.00	84.02 ± 1.40	84.82 ± 1.19	**90.97 ± 2.00**
	AA (%)	80.36 ± 1.37	72.51 ± 1.70	80.52 ± 0.58	81.00 ± 0.85	**87.81 ± 1.19**
Salinas	OA (%)	84.12 ± 1.60	75.46 ± 1.96	84.50 ± 2.06	85.06 ± 1.91	**90.68 ± 2.29**
	AA (%)	81.09 ± 1.59	70.96 ± 1.93	80.76 ± 1.25	82.88 ± 1.34	**87.76 ± 1.11**
Griffith-USGS	OA (%)	57.46 ± 1.94	54.44 ± 1.38	60.61 ± 1.33	63.60 ± 1.05	**68.64 ± 1.50**
	AA (%)	54.33 ± 2.57	50.18 ± 1.15	57.73 ± 1.94	60.55 ± 1.20	**64.60 ± 2.36**

Table 5.8: Classification accuracies obtained by random forest with different feature extraction methods on all datasets. Best accuracies are shown in bold.

Dataset		LDA	PCA	ICA	CNN	GAN
Indian Pines	OA (%)	85.06 ± 1.36	81.00 ± 1.84	87.14 ± 1.33	85.04 ± 1.45	**91.13 ± 1.55**
	AA (%)	82.35 ± 2.25	77.70 ± 2.25	85.90 ± 1.18	81.09 ± 0.65	**87.34 ± 2.55**
Pavia University	OA (%)	89.19 ± 0.92	83.90 ± 1.50	89.69 ± 1.27	90.89 ± 1.21	**93.67 ± 0.80**
	AA (%)	84.20 ± 1.23	79.70 ± 2.15	85.13 ± 2.10	87.66 ± 2.11	**89.82 ± 2.45**
Salinas	OA (%)	89.83 ± 2.05	84.19 ± 1.96	87.08 ± 1.75	88.71 ± 2.05	**92.31 ± 1.13**
	AA (%)	85.11 ± 1.06	80.42 ± 1.87	83.91 ± 0.85	85.88 ± 1.35	**89.22 ± 0.85**
Griffith-USGS	OA (%)	63.53 ± 1.45	60.39 ± 2.18	65.60 ± 1.77	65.54 ± 1.25	**70.70 ± 1.45**
	AA (%)	59.37 ± 1.95	57.81 ± 1.09	61.90 ± 1.65	61.27 ± 0.75	**66.01 ± 1.54**

- ICA performs consistently well compared to LDA and PCA with all classifiers on each datasets.

- CNN features do not significantly outperform other shallow features. In some cases, the difference in OA between CNN and ICA are very small. Since we did

Table 5.9: Classification accuracies obtained by KNN with different feature extraction methods on all datasets. Best accuracies are shown in bold.

Dataset		LDA	PCA	ICA	CNN	GAN
Indian Pines	OA (%)	76.15 ± 1.20	73.65 ± 1.78	78.11 ± 2.45	80.05 ± 1.70	**86.35 ± 0.35**
	AA (%)	73.90 ± 1.40	70.40 ± 2.15	75.09 ± 0.90	75.16 ± 2.00	**83.68 ± 2.16**
Pavia University	OA (%)	79.27 ± 1.90	74.39 ± 1.88	79.19 ± 1.60	81.46 ± 1.95	**88.84 ± 1.28**
	AA (%)	76.79 ± 1.77	71.30 ± 1.37	77.29 ± 1.78	78.62 ± 2.11	**83.24 ± 0.91**
Salinas	OA (%)	80.19 ± 2.35	74.34 ± 0.77	80.92 ± 1.12	83.74 ± 1.50	**89.00 ± 1.53**
	AA (%)	75.65 ± 1.48	71.03 ± 1.00	76.08 ± 0.75	80.70 ± 1.89	**86.19 ± 1.63**
Griffith-USGS	OA (%)	55.68 ± 0.18	52.50 ± 1.40	56.25 ± 1.39	61.71 ± 1.80	**65.36 ± 1.70**
	AA (%)	52.10 ± 1.07	49.44 ± 1.30	54.92 ± 2.28	59.82 ± 1.86	**63.57 ± 1.65**

not augment the training samples, CNN performed badly on some classes. Hence, it is evident that the small number of training samples restricted the performance.

- In the GAN-based model, we supplied both real and fake samples to the discriminator for training purposes. The higher accuracy of the model, therefore, indicates the usefulness of the generated samples which have increased the number of training samples for the deep model and eventually improve the performance.

5.6.4 Comparing with Other Spectral-Spatial Classification Methods to Evaluate Triplet Constraints

In this section, we evaluate the performance of our proposed triplet constraint-based GAN model by comparing our model with other baseline spectral-spatial classification methods. We first compared our method with [194], a spectral-spatial classification approach based on a novel extrema-oriented connected filtering technique, referred to as extended extinction profiles (EEP). This approach simplifies the input image by discarding insignificant spatial details and preserves the geometrical characteristics of other regions from the first informative features extracted by ICA or PCA. The resulting output is supplied to RF for classification. We experimented on both ICA and PCA and reported the best results on all datasets. According to our experiments, best accuracies were obtained using ICA on Pavia University, Salinas and Griffith-USGS whereas PCA produced best accuracy on Indian Pines.

We further compared our method with few other deep models in order to present a fair evaluation of our method. The first deep model is based on the extended morphological profile (EMP). During our experiments, ten principal components (PC) were extracted from the original data. After that we performed opening and closing operations on the first five PCs on Indian Pines, seven PCs on Pavia University and Salinas and nine PCs on Griffith-USGS to extract spatial information. The generated spatial features and original spectral features are supplied to a standard CNN for classification.

The second deep model 3D-CNN-LR is based on 3D-CNN that learns the signal changes in both spatial and spectral dimensions of HSIs. Proposed by Chen *et al.* [119], the method is able to extract significant discriminative information for classification and exploit powerful structural characteristics for HSI data. The classification accuracy is further improved by using L2 regularization and dropout during training. The number of available training samples are increased by generating virtual samples from real samples.

Finally, we compared our model with 3D-GAN [140], a recent GAN-based model to classify remote sensing images. This model generates fake samples to increase the number of training samples for improving the classification performance. The discriminator in the model reduces the number of spectral bands to three components by PCA and reserves the spatial information. The classification step also includes an additional softmax classifier to perform multi-class classification.

Table 5.10 presents the classification results of our proposed Triplet-3D-GAN and other baseline spectral-spatial methods. Our observation based on the overall accuracy results is given below:

- Triplet-3D-GAN outperforms EEP and EMP-CNN by approximately 5% and 8%

respectively on Indian Pines, approximately 4% and 6% respectively on Pavia University, approximately 3% and 4% respectively on Salinas. However, on Griffith-USGS, it achieves a significant improvement over EEP and EMP-CNN by approximately 16% and 15% respectively.

- Triplet-3D-GAN produces similar results as 3D-GAN [140] and 3D-CNN-LR [119] on Indian Pines and Pavia University and slightly better on Salinas. However, it outperforms both models by approximately 7% for Griffith-USGS. The integration of triplet constraints into the training process of the discriminator for real data has therefore improved the classification performance.
- Observing the increased accuracies by Triplet-3D-GAN, we also conclude that the generated fake samples help the discriminator by supplying sufficient training samples to improve the training process. It also indicates that the quality of the generated samples is better and in turn, supports our idea of including perceptual loss in the generator to improve the sample generation process.

Table 5.10: Comparison on classification accuracies obtained by different spectral-spatial methods. Best accuracies are shown in bold.

Dataset		EEP [194]	EMP-CNN	3D-CNN-LR [119]	3D-GAN [140]	Triplet-3D-GAN
Indian Pines	OA (%)	93.25 ± 1.37	90.75 ± 1.05	**98.25 ± 0.78**	98.10 ± 0.47	98.19 ± 0.05
	AA (%)	93.15 ± 1.60	91.85 ± 2.45	**99.27 ± 0.12**	99.05 ± 0.27	99.15 ± 0.07
Pavia University	OA (%)	94.38 ± 0.50	92.55 ± 1.68	**98.80 ± 0.28**	98.55 ± 0.10	98.75 ± 0.15
	AA (%)	95.26 ± 1.44	93.30 ± 1.96	99.40 ± 0.07	99.20 ± 0.05	**99.45 ± 0.06**
Salinas	OA (%)	95.88 ± 0.29	94.18 ± 1.00	98.05 ± 0.15	97.00 ± 0.85	**98.90 ± 0.10**
	AA (%)	95.56 ± 0.84	94.20 ± 0.65	98.90 ± 0.19	98.83 ± 0.09	**99.25 ± 0.03**
Griffith-USGS	OA (%)	74.77 ± 1.25	75.19 ± 1.80	82.53 ± 0.69	83.15 ± 1.50	**90.35 ± 0.80**
	AA (%)	72.23 ± 2.30	73.70 ± 2.39	83.04 ± 0.91	84.15 ± 0.95	**90.13 ± 0.50**

5.6.5 Testing Important Stages of the Proposed Model

Evaluating Intermediate Features

First, we tested the effectiveness of including intermediate features obtained from the discriminator into generator's perceptual loss computation. We observed in Table 5.10 that Triplet-3D-GAN outperforms other methods significantly on Griffith-USGS compared to other datasets. For indian pines, pavia university and salinas, there was not enough space to significantly improve as the accuracies achieved by other methods on these three datasets were already satisfactory. One of the main reasons of obtaining such high accuracies is that the training and testing samples were selected from the same scene. This increased the possibilities of significant misuse of spatial information during training. Since training and testing samples were selected from different scenes on Griffith-USGS, obtaining high accuracy became more challenging and hence, the usefulness of the proposed model can be better evaluated. Therefore, it is interesting to analyse results on this dataset in more detail to investigate the improvement for individual classes. Hence, this particular experiment includes results from only Griffith-USGS. To support our idea, we conduct experiments in two settings: (1) including intermediate features and (2) excluding intermediate features in the perceptual loss.

Table 5.11: Evaluation of including intermediate features on Griffith-USGS

Class	Without Intermediate Features	With Intermediate Features
Road	65.50 ± 0.97	72.46 ± 2.27
Water	77.88 ± 2.10	78.22 ± 1.16
Building	77.40 ± 1.55	78.39 ± 0.88
Grass	77.85 ± 0.76	78.25 ± 1.00
Tree	74.55 ± 0.54	75.40 ± 1.11
Soil	67.86 ± 0.94	74.59 ± 0.38
OA (%)	73.80 ± 1.69	75.50 ± 1.59
AA (%)	73.51 ± 0.58	76.21 ± 0.50

Table 5.11 reports the classification accuracies obtained by including and excluding the intermediate features into the perceptual loss. Although the accuracies improved slightly for most classes after including intermediate features obtained from the discriminator, we observe that the accuracies of the classes "road" and "Soil" increased significantly. It

Table 5.12: Evaluation of Triplet Constraints on Griffith-USGS (Best Accuracies are shown in bold)

Class	3D-CNN [122]	3D-CNN-LR [119]	Triplet-3D-CNN
Road	63.35 ± 1.71	**73.59 ± 1.05**	72.46 ± 2.27
Water	73.90 ± 0.66	75.11 ± 1.40	**79.22 ± 1.16**
Building	69.88 ± 0.57	**79.12 ± 0.41**	78.39 ± 0.88
Grass	70.15 ± 2.05	73.11 ± 2.71	**78.25 ± 1.00**
Tree	68.27 ± 0.96	70.05 ± 1.37	**76.40 ± 1.11**
Soil	65.33 ± 0.73	65.86 ± 1.26	**74.59 ± 0.38**
OA(%)	66.44 ± 2.21	67.91 ± 1.87	**75.50 ± 1.59**
AA(%)	68.48 ± 1.40	72.80 ± 1.96	**76.55 ± 0.50**

is clear that the sample generation process was benefited to some extent after including intermediate feedback from the discriminator.

Evaluating Triplet Constraints

We used triplet constraint as an important feature embedding to be learned during the training of 3D CNN for classification. The main purpose of using this constraint is to improve the classification performance of 3D CNN instead of adopting additional post-processing stages. Hence, to further validate the effectiveness of using triplets during CNN training, we present additional comparisons with other 3D-CNN-based models. The first baseline method 3D-CNN [122] employed a standard 3D CNN. The second baseline method 3D-CNN-LR [119] used L2 regularization and dropout in the training process to improve the classification results. We included triplet constraints into the standard 3D-CNN which we call "Triplet-3D-CNN" and compared the classification accuracy with 3D-CNN [122] and 3D-CNN-LR [119]. Table 5.12 reports the class-specific OA and AA in order to demonstrate the effectiveness of using triplet constraints. To make fair comparisons, we randomly selected 15% samples for training and the rest of the available samples for testing. Also, we did not include augmented samples in 3D-CNN-LR [119] since for this experiment, we included triplet constraints into our model without augmented samples.

In Table 5.12, we see that Triplet-3D-CNN achieves better classification accuracy compared to other methods. It is quite evident from the results that the use of triplet constraint provided useful spectral-spatial feature embedding for the classifier and improve the classification performance.

Evaluating Augmented Fake Samples

Considering there is a high possibility that the synthesized samples are both realistic and diverse, we used the fake spectra to augment the existing datasets. To support our claim, we now evaluate the effectiveness of including those generated fake samples into the training process. We randomly selected 100 fake samples generated by Triplet-3D-GAN and included those into the training set. The original data had Y classes and the generated fakes samples were supplied with a $Y + 1$-th label for training. We included both real (10% from each class) samples and fake samples into training processes of EMP-CNN and 3D-CNN-LR and measured their classification performances. For 3D-CNN-LR, we did not include the augmented samples obtained by their proposed method. Table 5.13 shows a comparison on accuracies obtained by those two methods with and without augmented samples. It is evident from the results that accuracies increased after including the Triplet-3D-GAN generated augmented samples in training.

TABLE 5.13: Evaluation of GAN-generated Augmented Samples in Training

Dataset		EMP-CNN	EMP-CNN-Augmented	3D-CNN-LR	3D-CNN-LR-Augmented
Indian Pines	OA (%)	83.15 ± 1.20	86.15 ± 0.65	85.08 ± 0.52	88.70 ± 1.00
	AA (%)	83.11 ± 1.09	88.65 ± 0.05	85.29 ± 0.70	90.51 ± 1.03
Pavia University	OA (%)	86.40 ± 1.38	90.05 ± 1.05	89.66 ± 0.71	93.85 ± 1.05
	AA (%)	87.30 ± 1.44	91.62 ± 0.42	90.05 ± 1.56	94.55 ± 0.25
Salinas	OA (%)	86.18 ± 1.13	90.44 ± 1.69	90.16 ± 0.05	93.77 ± 1.80
	AA (%)	85.07 ± 1.24	92.88 ± 0.10	90.85 ± 0.18	94.96 ± 0.43
Griffith-USGS	OA (%)	70.29 ± 0.90	75.20 ± 0.60	72.29 ± 1.40	76.05 ± 1.15
	AA (%)	71.45 ± 1.30	76.66 ± 0.55	71.37 ± 0.90	77.15 ± 0.60

5.6.6 Parameter Analysis

Selection of the Optimal Parameters of SVM, RF and KNN.

To find the optimal parameters for SVM, RF and KNN, we followed the same procedures defined in Section 4.5.2 in Chapter 4. The use of the optimal parameters for SVM are reflected in Table 5.7. The use of the optimal parameters for RF are reflected in Table 5.8 and in Table 5.10 for classifying with EEP. Similarly, the use of the optimal value of k is reflected in Table 5.9. The optimal parameters of the classifiers on different datasets are provided in Table 5.14.

TABLE 5.14: Optimal parameters for SVM, RF and KNN

Dataset	Support Vector Machine		Random Forest	K Nearest Neighbor
	Gamma	Cost	Number of Trees	Value of K
Indian Pines	0.125	32	200	7
Pavia University	0.0625	32	300	5
Salinas	0.25	16	300	5
GU-USGS	0.0625	16	300	6

Analysis on Selection of Triplets

As mentioned earlier, we considered all anchor-positive pairs during training. To validate this option, we compared it with hard anchor-positive pairs in which case we selected positive samples that were relatively close to the anchor samples than other positive samples. Next, we evaluate the effectiveness of considering hard negative samples during the triplet construction instead of taking all negative samples into account. We selected negative samples which were spectrally close to the positive samples and did not consider the samples for whose the squared distances were quite further to the anchor-positive distance. Hence, it will be interesting to observe how the training and testing losses behave in cases of selecting all positive-negative and hard positive-negative. Fig. 5.6 illustrates the comparison which shows that the training losses in (a) for both settings are converging with the increasing iterations. But the testing loss of hard anchor-positive pair setting keeps increasing after a pivoting point at 2000 iterations while that of all anchor-positive pair setting keeps decreasing. Therefore, it can be deduced that selecting hard samples does not cover all kinds of data distributions and on the other hand, all anchor-positive samples avoid this problem by resulting in a more generalized outcome over the testing set. The training losses in (b) for both settings are also converging with the increasing iterations. But the testing loss of hard positive-negative pair setting keeps decreasing after a pivoting point at 2000 iterations, while that of all positive-negative pair

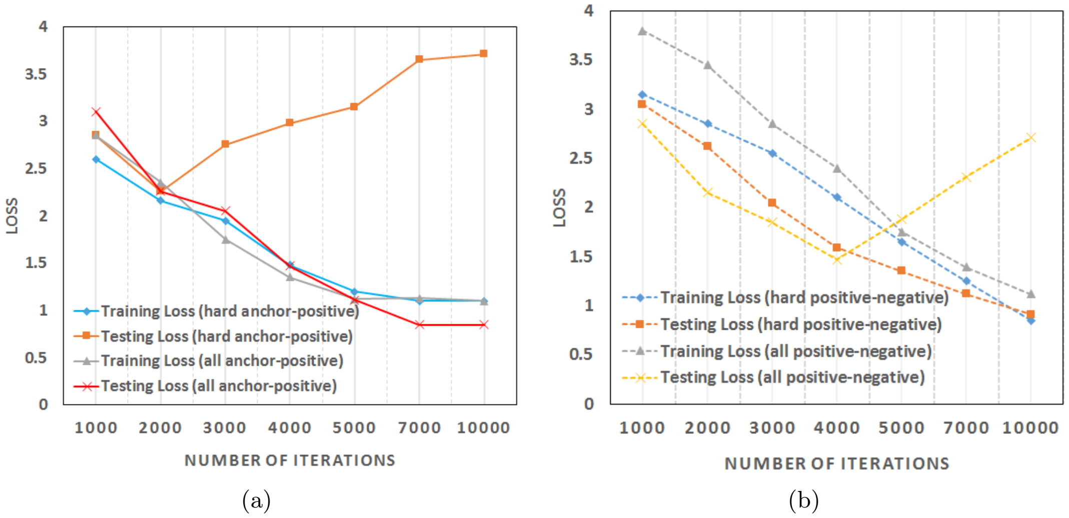


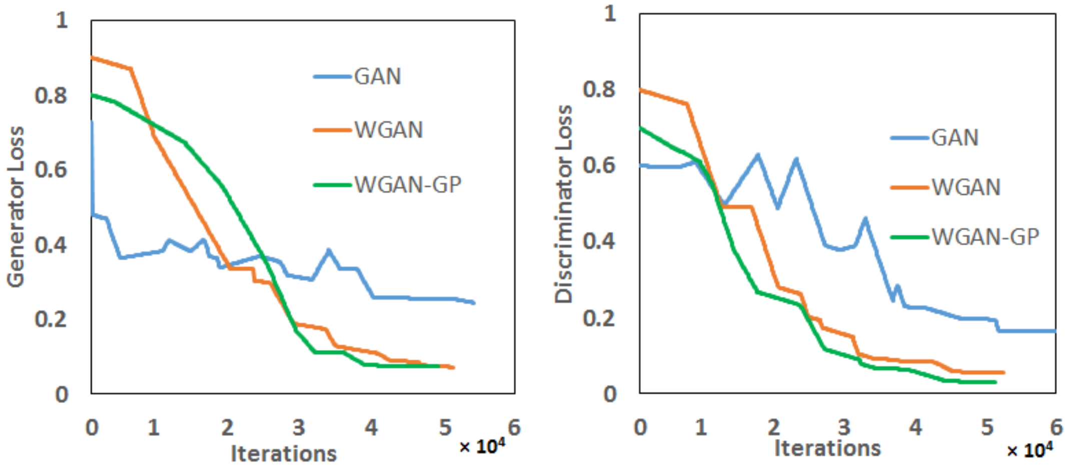
Figure 5.6: Comparison of training and test losses between (a) all anchor-positive and hard anchor-positive pairs, (b) all positive-negative and hard positive-negative pairs.

setting keeps increasing. Therefore, we conclude that selecting hard negative samples captures better spectral-spatial representation of the data distributions and contributes to the classification performance. On the other hand, selecting all negative samples fails to construct better spectral-spatial representation and as a result, does not produce generalized outcome over the testing set.

Evaluating Convergence During Training

In our model, we adopted the improved WGAN which has extended the standard WGAN model by introducing gradient penalty term in the objective function to avoid weight clipping issue. Furthermore, we included feedback from the discriminator's intermediate features into the perceptual loss in the generator. We also added triplet loss in the discriminator's objective function. Because our model is primarily based on the improved WGAN, we still refer to it by the original term WGAN-GP.

In this experiment, we analyse the training losses encountered by the standard GAN, WGAN and WGAN-GP in order to make a comparison on the respective convergences achieved by the three GAN models. This experiment is very important in a sense that the training losses are required to correlate well with the training progress for hyperparameter tuning and detecting overfitting. Fig. 5.7 presents a comparison on the generator and discriminator losses over iterations for these three models.



(a) (b)

FIGURE 5.7: (a) Generator loss (b) discriminator loss over iterations for three models: Original GAN, WGAN with weight clipping and WGAN-GP with gradient penalty.

From both figures, we observe that both WGAN and WGAN-GP losses decrease in a regular pattern with training progress. However, the standard GAN objective results in a fluctuating pattern and therefore, does not coincide with the training progress effectively. Hence, we draw a conclusion that WGAN and WGAN-GP objectives effectively correlate over the training iterations.

5.6.7 Visualization of the Generated Samples

In this section, we present some visualization of the fake samples generated by Triplet-3D-GAN. According to [115], if the discriminator fails to separate real data from fake ones, it can be deduced that the generator achieved better performance and the entire adversarial network reaches the global optimality theory. We present some fake samples generated by the generator in Figs 5.8, 5.9, 5.10 and 5.11 for Indian Pines, Pavia University, Salinas and Griffith-USGS respectively. We observe that there is indeed similarity between the original and fake samples to some extent. From the results, it is evident that the generated samples capture more details over the iterations during the adversarial training.

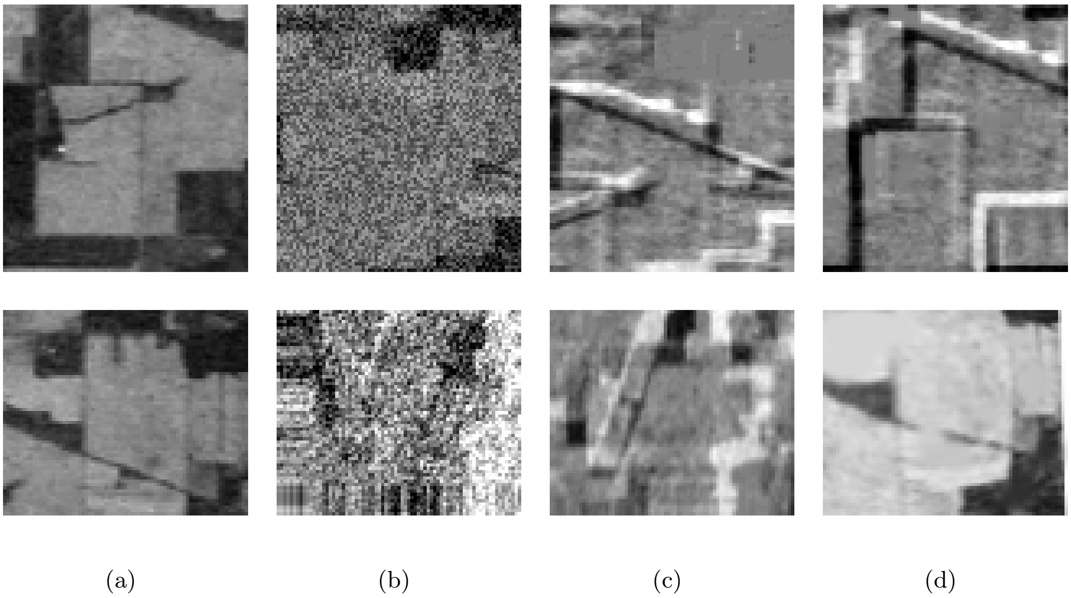


Figure 5.8: Comparison of Real and generated fake data on Indian Pines (First row - Soybeans-min, Second row - Soybeans-notill): (a) Real training data (b) First corresponding fake data (c) Second corresponding fake data (d) Third corresponding fake data

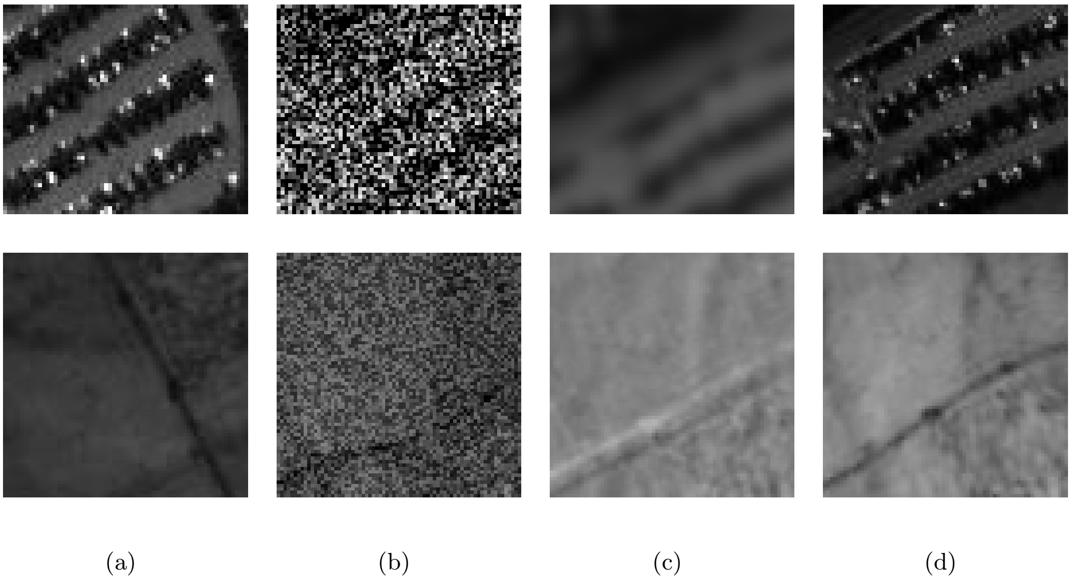


Figure 5.9: Comparison of Real and generated fake data on Pavia University (First row - Bricks, Second row - Meadows): (a) Real training data (b) First corresponding fake data (c) Second corresponding fake data (d) Third corresponding fake data

5.7 Conclusion

In this chapter, we presented a GAN-based spectral-spatial classification method for hyperspectral images. We mainly focused on improving the multi-class classification

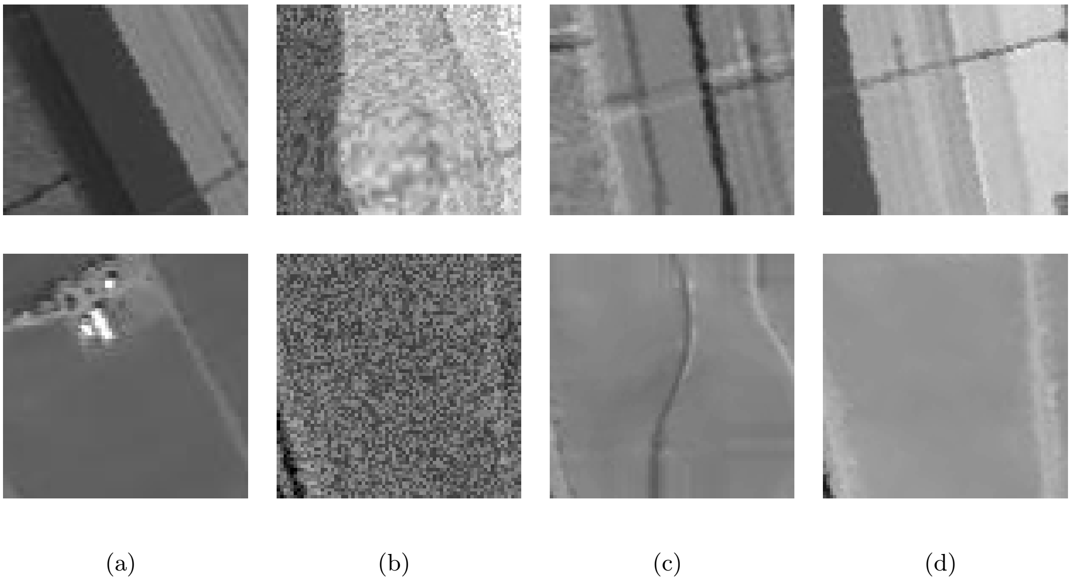

(a) (b) (c) (d)

Figure 5.10: Comparison of Real and generated fake data on Salinas (First row - Stubble, Second row - Soil): (a) Real training data (b) First corresponding fake data (c) Second corresponding fake data (d) Third corresponding fake data

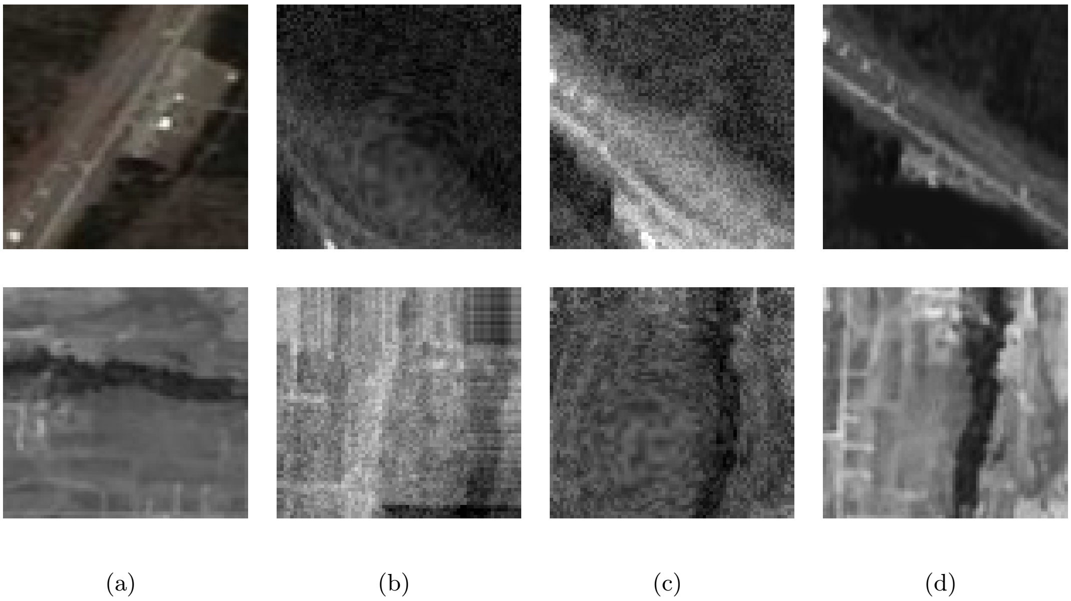

(a) (b) (c) (d)

Figure 5.11: Comparison of Real and generated fake data on Griffith-USGS (First row - Road, Second row - Water): (a) Real training data (b) First corresponding fake data (c) Second corresponding fake data (d) Third corresponding fake data

ability of the discriminator of GAN models since little attention has been paid in the past regarding this. To address this issue, we proposed to adopt the triplet constraint property and extended it to build a useful feature embedding for remote sensing images and further used it for classification task. With the inclusion of such powerful spectral-spatial embeddings, our proposed GAN model achieves better classification accuracy compared to those of traditional CNN models and also, other shallow classification models. Our proposed Triplet-3D-GAN network is based on the WGAN-GP architecture and further includes feedback from discriminator's intermediate features to improve the quality of the generator's sample generation process. Our architecture demonstrated better performance during both feature extraction and classification stages. Since we used the generated fake samples into the discriminator's training stage, it contributed immensely in mitigating the overfitting problem of a deep model. We presented both quantitative and visual (generated fake samples) results which indicate the promising potential of adopting GANs for remote sensing image classification.

Chapter 6

Conclusions

In this chapter, we summarize our main contributions toward the research of deep learning-based spectral-spatial hyperspectral classification models and explore directions for future work.

6.1 Contributions of the Thesis

With the recent development of advanced imaging instruments, HSI system has emerged itself as an effective tool to analyse interesting problems in remote sensing. Focusing on image understanding, the research has long attracted the attention from different researchers because the analysis results, such as the classification, are the basis for many different applications including monitoring quality parameters of agriculture materials, automatic identification of minerals, detecting environmental changes, providing useful security services etc.

The land-cover classification problem, characterizing a given geographical area of interest, is a complex process that requires extracting and analysing useful spectral and spatial information of HSI data. Although extensive research has been done on considering both spectral and spatial information, there is still a high demand on developing novel spectral-spatial classification methods that can effectively extract highly comprehensive and discriminative representation of interested objects.

This thesis introduces three novel deep learning-based models to effectively represent spectral-spatial characteristics of hyperspectral data in the interest of classification of remote sensing images. Each method addresses several fundamental and challenging aspects related to the development of deep models for remote sensing image classification. These methods are derived from traditional approaches in computer vision and then significantly extended to hyperspectral images. The contributions and significance of the proposed methods are summarized as follows.

In Chapter 3, we introduced a novel and optimized deep CRF model, which is formulated in a deep modeling approach and is integrated with the advantages of both CNN and CRF in representing spatial relationships in the data. The usefulness of integrating CNN with CRFF is greatly complemented by the utilization of smaller-sized yet a large number of spectral groups to provide more accurate local spectral-spatial structure description of the data. We fabricated an expressive deep model by employing sample fusion strategy to significantly increase the training samples which essentially addresses the drawback of limited training samples of HSI data. The detailed experiments done in different settings strongly support the idea of the proposed integration of CRF in a deep model to improve classification performance.

In Chapter 4, we explored the possibilities of integrating estimated material-specific abundance information results as input to deep models with a view to improve classification performance. An additional contribution of our method is we generated effective KD-estimated superpixels containing more useful information about specific materials which complemented the unmixing performance. Treated as important cues, these abundance estimations substantially contributed in improving the deep model's capabilities in generating more expressive, high level features of HSI data. Combining the unmixing results and widely used classifiers during experiments clearly indicates that hyperspectral unmixing have outstanding potential in improving the HSI classification performance, in general.

In Chapter 5, we investigated the potential of recently proposed generative adversarial networks into the field of remote sensing and presented a model accordingly. The proposed model specifically considers the possibilities of improving the multi-class classification ability of the discriminator. The inclusion of triplet constraint in building a powerful feature embedding provide discriminative information about the data that eventually contribute in separating the individual classes. Furthermore, the intermediate feedback from the discriminator improve the quality of the generated fake samples that can be used as augmented samples during training, as demonstrated during experiments.

6.2 Future Work

This thesis has proposed three different spectral-spatial approaches to classify hyperspectral data based on deep networks. However, this is an initial work and more work should be investigated in the future.

Based on our discoveries throughout our research, we believe that the excellent structured modelling capabilities of deep CRF potentials can immensely contribute in the task of segmentation as well. It is, in fact, a common practice to use classification as an initial step toward the final segmentation in many computer vision approaches. There is also a high potential of further exploration of combining deep models with hyperspectral unmixing results. However, it is not adequate to handle complex cases such as multiple scattering effects, water-absorbed environments etc. with linear mixture models. One way to deal with such scenarios is to use unsupervised approaches and keep the endmembers and fractional abundances blind.

In these contexts, we are interested in exploring the following possibilities to further exploit the potentials of deep networks for remote sensing image analysis:

- Specifically formulating the CRF potentials to construct semantic segments by exploiting spectral-spatial characteristics of the data for applying in segmentation task.
- Realizing that the adversarial model has excellent abilities to detect discrepancies between the model predictions and the ground-truth, CRF pairwise potentials can be supplied into a GAN model to formulate it as a segmentation task.

- Since CRF inference takes extremely long time to converge, the possibilities of reformulating a large proportion of the inference as convolutions can be explored. This can possibly be achieved by introducing conditional independence assumption to the full connected CRF to reduce the complexities of the pairwise potentials.
- Possibilities of including constraints such as spectral angle distance (SAD) instead of the inner product operators at the intermediate layers to obtain more discriminative features for extracting endmembers in an unmixing model.
- Activation functions such as ReLU and a normalization layer can help in effectively applying sparsity and non-linearity for deep model-based unmixing model.
- In contrast to many unmixing approaches, stochastic gradient-based solver can be adopted to better optimize the endmembers and the corresponding abundances.
- Following our findings of integrating triplet constraints in a deep modelling approach, further research can be carried out in the contexts of selecting triplets and using the generated fake samples since these process are tricky and may introduce additional problems, if not selected carefully.

Research in deep learning is continuously evolving. For remote sensing image applications, several deep models including SAEs, DBNs, GANs have been considerably successful recently and therefore, open the possibilities of introducing new outlooks into this field for further developments. The important limitation of deep models in image analysis is the lack of available public training datasets, which can be a future endeavour of the remote sensing community. For classification and segmentation tasks, the improvement of the network structure remains a challenge to establish a balance between the global context and the local details. Furthermore, in addition to land-cover classification, the applications of remote sensing images for a variety of domains deserve further research in the future.

Publications and Submitted Papers

Journals:

[1] Fahim Irfan Alam, Jun Zhou, Alan Wee-Chung Liew, Xiuping Jia, Jocelyn Chanussot and Yongsheng Gao. Conditional random field and deep feature learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1612-1628, March 2019.

[2] Fahim Irfan Alam, Jun Zhou, Litao Yu, Alan Wee-Chung Liew, Jocelyn Chanussot and Yongsheng Gao. Triplet Constrained Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing.* (submitted for review)

Conferences:

[1] Fahim Irfan Alam, Jun Zhou, Alan Wee-Chung Liew, and Xiuping Jia. CRF learning with CNN features for hyperspectral image segmentation. In *IEEE International Geoscience & Remote Sensing Symposium (IGARSS'16), 2016*, Beijing, 2016, pp. 6890-6893.

[2] Fahim Alam, Jun Zhou, Lei Tong, Alan Wee-Chung Liew, and Yongsheng Gao. Combining unmixing and deep feature learning for hyperspectral image classification. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA'17)*, Sydney, NSW, 2017, pp. 1-8.

[3] Fahim Irfan Alam, Jun Zhou, Alan Wee-Chung Liew, Jun Jo, and Yongsheng Gao. Triplet constrained deep feature extraction for hyperspectral image classification. In *Workshop on Hyperspectral Image and .ignal Processing: Evolution in Remote Sensing*, Amsterdam, The Netherlands, 2018.

Bibliography

[1] J. Liang, J. Zhou, L. Tong, X. Bai, and B. Wang. Material based salient object detection from hyperspectral images. *Pattern Recognition*, 76(C):476–490, 2018.

[2] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, 2012.

[3] A. F. Goetz, G. Vane, J. E. Solomon, and B. N. Rock. Imaging spectrometry for earth remote sensing. *Science*, 228(4704):1147–1153, 1985.

[4] 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*, 46(11):3804–3814, 2008.

[5] C. C. Lelong, P. C. Pinet, and H. Poilvé. Hyperspectral imaging and stress mapping in agriculture: a case study on wheat in beauce (France). *Remote Sensing of Environment*, 66(2):179–191, 1998.

[6] R. G. Resmini, M. E. Kappus, W.S. Aldrich, J. C. Harsanyi, and M. Anderson. Mineral mapping with hyperspectral digital imagery collection experiment HYDICE sensor data at Cuprite, Nevada, USA. *International Journal of Remote Sensing*, 18(7):1553–1570, 1997.

[7] S. Xu, X. Mu, D. Chai, and X. Zhang. Remote sensing image scene classification based on generative adversarial networks. *Remote Sensing Letters*, 9(7):617–626, 2018.

[8] S. Nie, L. Gu, Y. Zheng, A. Lam, N. Ono, and I. Sato. Deeply learned filter response functions for hyperspectral reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] M. T. Kuska, J. Behmann, and A Mahlein. Potential of hyperspectral imaging to detect and identify the impact of chemical warfare compounds on plant tissue. *Pure and Applied Chemistry*, 90(10):1615–1624, 2018.

[10] M. Nathan, A. S. Kabatznik, and A. Mahmood. Hyperspectral imaging for cancer detection and classification. In *2018 3rd Biennial South African Biomedical Engineering Conference (SAIBMEC)*, pages 1–4, 2018.

[11] L. Barroso, F. Burgos-Fernández, X. Delpueyo, M. Ares, S. Royo, J. Malvehy, S. Puig, and M. Vilaseca. Visible and extended near-infrared multispectral imaging for skin cancer diagnosis. *Sensors*, 18(5):1441, 2018.

[12] S. V. Panasyuk, S. Yang, D. V. Faller, D. Ngo, R. A. Lew, J. E. Freeman, and A. E. Rogers. Medical hyperspectral imaging to facilitate residual tumor identification during surgery. *Cancer Biology & Therapy*, 6(3):439–446, 2007.

[13] H. Dalvi, C. Fauteux-Lefebvre, J. Guay, N. Abatzoglou, and R. Gosselin. Concentration monitoring with near infrared chemical imaging in a tableting press. *Journal of Spectral Imaging*, 7(1):a5, 2018.

[14] S. Kim, K. Heinze, and P. Schwille. Fluorescence correlation spectroscopy in living cells. *Nature methods*, 4(11):963–73, 2007.

[15] N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19(1):44–57, 2002.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[17] A. Kirillov, D. Schlesinger, S. Zheng, B. Savchynskyy, P. Torr, and C. Rother. Joint training of generic CNN-CRF models with stochastic optimization. In *13th Asian Conference on Computer Vision*, pages 221–236, 2016.

[18] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4715–4723, 2016.

[19] X. Chu, W. Ouyang, H. Li, and X. Wang. CRF-CNN: Modeling structured information in human pose estimation. In *Advances in Neural Information Processing Systems*, pages 316–324, 2016.

[20] P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock. End-to-end training of hybrid CNN-CRF models for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2339–2348, 2017.

[21] F. Liu, G. Lin, and C. Shen. CRF learning with CNN features for image segmentation. *Pattern Recognition*, 48(10):2983–2992, 2015.

[22] D. Lin, K. Fu, Y. Wang, G. Xu, and X. Sun. MARTA GANs: Unsupervised representation learning for remote sensing image classification. *IEEE Geoscience and Remote Sensing Letters*, 14(11):2092–2096, 2017.

[23] Z. He, H. Liu, Y. Wang, and J. Hu. Generative adversarial networks-based semi-supervised learning for hyperspectral image classification. *Remote Sensing*, 9(10): 1042, 2017.

[24] Y. Zhan, D. Hu, Y. Wang, and X. Yu. Semisupervised hyperspectral image classification based on generative adversarial networks. *IEEE Geoscience and Remote Sensing Letters*, 15(2):212–216, 2018.

[25] F. Borfecchia, M. Pollino, L. D. Cecco, R. Lugari, R. Martini, L. L. Porta, E. Ristoratore, and C. Pascale. C.: Active and passive remote sensing for supporting the evaluation of the urban seismic vulnerability. *Italian Journal of Remote Sensing*, pages 129–141, 2010.

[26] J.B. Campbell. *Introduction to Remote Sensing*, chapter 10, pages 204–242. Guildford Press, 2008.

[27] F. Borfecchia, D. L. CECCO, A. Lugari, S. Martini, L. L. PORTA, E. Ristoratore, and M. Pollino. Active (lidar) and passive (multi/hyperspectral) remote sensing techniques for supporting the evaluation of the urban seismic vulnerability. In *Proceedings of the G4DM 2010 Conference*, pages 1–5, 2010.

[28] A. Goetz, G. Vane, J. E. Solomon, and B. N. Rock. Imaging spectrometry for earth remote sensing. *Science (New York, N.Y.)*, 228:1147–53, 1985.

[29] R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J Chovit, M. Solis, M. R. Olah, and O. Williams. Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing of Environment*, 65(3):227 – 248, 1998.

[30] P. Gamba. A collection of data for urban area characterization. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 69–72, 2004.

[31] L. J. Rickard, R. W. Basedow, E. F. Zalewski, P. R. Silverglate, and M. Landers. Hydice: an airborne system for hyperspectral imaging. In *Proceedings of the Imaging Spectrometry of the Terrestrial Environment, SPIE*, volume 1937, pages 173–179, 1993.

[32] B. Stevenson, R. O'Connor, W. Kendall, A. Stocker, W. Schaff, R. Holasek, D. Even, D. Alexa, J. Salvador, M. Eismann, R. Mack, and P. Kee. The civil air patrol archer hyperspectral sensor system. In *Proceedings of the Airborne ISR Systems and Applications II, SPIE*, volume 5787, pages 17–28, 2005.

[33] J. Pearlman, C. Segal, L. B. Liao, S. L. Carman, M. A. Folkman, W. Browne, L. Ong, and S. G. Ungar. Development and operations of the eo-1 hyperion imaging spectrometer. In *Proceedings of the Earth Observing Systems V, SPIE*, volume 4135, page 243–253, 2000.

[34] K. Staenz. Terrestrial imaging spectroscopy – some future perspectives. In *Proceedings of the 6th EARSeL SIG IS workshop*, pages 1–12, 2009.

[35] G. J. Brelstaff, A. Párraga, T. Troscianko, and D. Carr. Hyperspectral camera system: acquisition and analysis. In *Satellite Remote Sensing II*, pages 150–159, 1995.

[36] S. Mahesh, D. S. Jayas, J. Paliwal, and N. D. G. White. Hyperspectral imaging to classify and monitor quality of agricultural materials. *Journal of Stored Products Research*, 61:17 – 26, 2015.

[37] B. Yousefi, S. Sojasi, C. Ibarra-Castanedo, G. Beaudoin, F. Huot, X. P. V. Maldague, M. Chamberland, and E. Lalonde. Mineral identification in hyperspectral imaging using sparse-pca. In *Proceedings of SPIE 9861, Thermosense: Thermal Infrared Applications XXXVIII*, volume 9861, pages 1–11, 2016.

[38] S. T. Seydi and M. Hasanlou. A new land-cover match-based change detection for hyperspectral imagery. *European Journal of Remote Sensing*, 50(1):517–533, 2017.

[39] S. Freitas, H. Silva, J. Almeida, and E. Silva. Hyperspectral imaging for real-time unmanned aerial vehicle maritime target detection. *Journal of Intelligent & Robotic Systems*, 90(3):551–570, 2018.

[40] X. Prieto-Blanco, C. Montero-Orille, B. Couce, and R. de la Fuente. Optical configurations for imaging spectrometers. *Computational Intelligence for Remote Sensing*, 133:1–25, 2008.

[41] J. E. Fowler. Compressive pushbroom and whiskbroom sensing for hyperspectral remote-sensing imaging. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 684–688, 2014.

[42] B. Guldimann and S. Kraft. Focal plane array spectrometer: miniaturization effort for space optical instruments. In *Proceedings of the SPIE 7930, MOEMS and Miniaturized Systems X*, volume 7930, 2011.

[43] M. F. Hopkins. Four-color pyrometry for metal emissivity characterization. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 2599, pages 294–301, 1996.

[44] N. Gat. Imaging spectroscopy using tunable filters: a review. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4056, pages 50–64, 2000.

[45] X. Wang, Y. Zhang, X. Ma, T. Xu, and G. R. Arce. Compressive spectral imaging system based on liquid crystal tunable filter. *Optics Express*, 26(19):25226–25243, 2018.

[46] D. A. Glenar, D. L. Blaney, and J. J. Hillman. Aims: Acousto-optic imaging spectrometer for spectral mapping of solid surfaces. *Acta Astronautica*, 52(2):389 – 396, 2003.

[47] M. Pollnau. Counter-propagating modes in a fabry-perot-type resonator. *Optics Letters*, 43(20):5033–5036, 2018.

[48] N. Keshava. A survey of spectral unmixing algorithms. *Lincoln Laboratory Journal*, 14(1):55–78, 2003.

[49] J. B. Adams, M. O. Smith, and P. E. Johnson. Spectral mixture modeling: a new analysis of rock and soil types at the viking lander 1 site. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 91(B8):8098–8112, 1986.

[50] C. Theys, N. Dobigeon, J. Tourneret, and H. Lanteri. Linear unmixing of hyperspectral images using a scaled gradient method. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 729–732, 2009.

[51] J. Wang and C. . Chang. Applications of independent component analysis in endmember extraction and abundance quantification for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2601–2616, 2006.

[52] J. M. P. Nascimento and J. M. B. Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, 2005.

[53] B. Hapke. *Theory of reflectance and emittance spectroscopy.* Cambridge University Press, 2012.

[54] D. A. Landgrebe. Multispectral land sensing: where from, where to? *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):414–421, 2005.

[55] Bor-Chen Kuo and D. A. Landgrebe. Nonparametric weighted feature extraction for classification. *IEEE Transactions on Geoscience and Remote Sensing*, 42(5): 1096–1105, 2004.

[56] C. Chion, J. Landry, and L. Da Costa. A genetic-programming-based method for hyperspectral data information extraction: Agricultural applications. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2446–2457, 2008.

[57] C. Daughtry. Discriminating crop residues from soil by shortwave infrared reflectance. *Agronomy Journal*, 93:125–131, 2001.

[58] G. Meera Gandhi, S. Parthiban, Nagaraj Thummalu, and A. Christy. NDVI: Vegetation change detection using remote sensing and gis – a case study of vellore district. *Procedia Computer Science*, 57:1199 – 1210, 2015.

[59] S. K. McFEETERS. The use of the normalized difference water index (ndwi) in the delineation of open water features. *International Journal of Remote Sensing*, 17(7):1425–1432, 1996.

[60] J. Qi, A. Chehbouni, A. R. Huete, Y. H. Kerr, and S. Sorooshian. A modified soil adjusted vegetation index. *Remote Sensing of Environment*, 48(2):119 – 126, 1994.

[61] S. De Backer, P. Kempeneers, W. Debruyn, and P. Scheunders. A band selection technique for spectral classification. *IEEE Geoscience and Remote Sensing Letters*, 2(3):319–323, 2005.

[62] S. B. Serpico and G. Moser. Extraction of spectral channels from hyperspectral images for classification purposes. *IEEE Transactions on Geoscience and Remote Sensing*, 45(2):484–495, 2007.

[63] A. Ifarraguerri and M. W. Prairie. Visual method for spectral band selection. *IEEE Geoscience and Remote Sensing Letters*, 1(2):101–106, 2004.

[64] K. Fukunaga. *Introduction to Statistical Pattern Recognition*, pages 399 – 440. Academic Press, second edition, 1990.

[65] T. V. Bandos, L. Bruzzone, and G. Camps-Valls. Classification of hyperspectral images with regularized linear discriminant analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 47(3):862–873, 2009.

[66] K. Fukunaga and J. M. Mantock. Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(6):671–678, 1983.

[67] J. Yang, P. Yu, and B. Kuo. A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(3):1279–1293, 2010.

[68] H. Huang and B. Kuo. Double nearest proportion feature extraction for hyperspectral-image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4034–4046, 2010.

[69] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Comput.*, 12(10):2385–2404, October 2000.

[70] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce. Locality-preserving discriminant analysis in kernel-induced feature spaces for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 8(5):894–898, 2011.

[71] Y. Qian, M. Ye, and J. Zhou. Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features. *IEEE Transactions on Geoscience and Remote Sensing*, 51(4):2276–2291, 2013.

[72] J. Liang, J. Zhou, and Y. Gao. 3D local derivative pattern for hyperspectral face recognition. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–6, 2015.

[73] J. Wang and C. Chang. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1586–1600, 2006.

[74] N. Wang, B. Du, L. Zhang, and L. Zhang. An abundance characteristic-based independent component analysis for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):416–428, 2015.

[75] A. Ifarraguerri and Chein-I Chang. Unsupervised hyperspectral image analysis with projection pursuit. *IEEE Transactions on Geoscience and Remote Sensing*, 38(6):2529–2538, 2000.

[76] M. M. Crawford, L. Ma, and W. Kim. *Exploring Nonlinear Manifold Learning for Classification of Hyperspectral Data*, pages 207–234. 2011.

[77] Y. Chen, M. Crawford, and J. Ghosh. Improved nonlinear manifold learning for land cover classification via intelligent landmark selection. In *2006 IEEE International Symposium on Geoscience and Remote Sensing*, pages 545–548, 2006.

[78] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[79] F. Bach and M. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2003.

[80] S. L. Al-khafaji, J. Zhou, A. Zia, and A. W. Liew. Spectral-spatial scale invariant feature transform for hyperspectral images. *IEEE Transactions on Image Processing*, 27(2):837–850, 2018.

[81] J. Li, J. Bioucas-Dias, and A. Plaza. Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 50(3):809–823, 2012.

[82] Q. Wang, X. He, and X. Li. Locality and structure regularized low rank representation for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):911–923, 2019.

[83] Q. Wang, Z. Meng, and X. Li. Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 14(11):2077–2081, 2017.

[84] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[85] M. Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

[86] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294 – 300, 2006.

[87] V.F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J.P. Rigol-Sanchez. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67: 93 – 104, 2012.

[88] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778–1790, 2004.

[89] V. N. Vapnik. *The Nature Of Statistical Learning Theory*, volume 6. 1995.

[90] G. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, 1968.

[91] L Bruzzone and B. Demir. *A Review of Modern Approaches to Classification of Remote Sensing Data*, pages 127–143. Springer Netherlands, 2014.

[92] X. Chen, T. Fang, H. Huo, and D. Li. Graph-based feature selection for object-oriented classification in VHR airborne imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 49(1):353–365, 2011.

[93] X. Jia. Class modeling and remote sensing image classification using selected spectral and spatial features. In *Proc. of International Congress of Imagince Science (ICIS)*, pages 302–305, 2006.

[94] M. De Martinao, F. Causa, and S. B. Serpico. Classification of optical high resolution images in urban environment using spectral and textural information. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, volume 1, pages 467–469, 2003.

[95] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6): 610–621, 1973.

[96] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):480–491, 2005.

[97] M. D. Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone. Morphological attribute profiles for the analysis of very high resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(10):3747–3762, 2010.

[98] C. Deng, S. Li, F. Bian, and Y. Yang. Remote sensing image segmentation based on mean shift algorithm with adaptive bandwidth. In *Proceedings of the Second International Conference on Geo-Informatics in Resource Management and Sustainable Ecosystem*, pages 179–185, 2015.

[99] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. Benediktsson. SVM and MRF-based method for accurate classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 7(4):736–740, 2010.

[100] Y. Zhao, L. Zhang, P. Li, and B. Huang. Classification of high spatial resolution imagery using improved Gaussian Markov random-field-based texture features. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5):1458–1468, 2007.

[101] O. Eches, J. Benediktsson, N. Dobigeon, and J. Tourneret. Adaptive Markov random fields for joint unmixing and segmentation of hyperspectral images. *IEEE Transactions on Image Processing*, 22(1):5–16, 2013.

[102] J. Li, P. Marpu, A. Plaza, J. Bioucas-Dias, and J. Benediktsson. Generalized composite kernel framework for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 51(9):4816–4829, 2013.

[103] E. T. Gormus, N. Canagarajah, and A. Achim. Dimensionality reduction of hyperspectral images with wavelet based empirical mode decomposition. In *Proceedings of the 18th IEEE International Conference on Image Processing*, pages 1709–1712, 2011.

[104] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 346–361, 2014.

[105] 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*, 38(2):295–307, 2016.

[106] M. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. In *Proceedings of the International Conference on Learning Representation (ICLR)*, 2013.

[107] W. Ouyang, X. Wang, X. Zeng, Shi Qiu, P. Luo, Y. Tian, H. Li, Shuo Yang, Zhe Wang, Chen-Change Loy, and X. Tang. DeepID-Net: Deformable deep convolutional neural networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2403–2412, 2015.

[108] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, pages 1097–1105, 2012.

[109] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017.

[110] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, pages 315–323, 2013.

[111] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103, 2008.

[112] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pages 1137–1144, 2006.

[113] T. Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1064–1071, 2008.

[114] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[115] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014.

[116] J. T. Springenberg. Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks. *ArXiv e-prints*, arXiv:1511.06390v2, 2015. URL https://arxiv.org/abs/1511.06390.

[117] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. URL http://arxiv.org/abs/1511.06434.

[118] A. Odena. Semi-Supervised Learning with Generative Adversarial Networks. *ArXiv e-prints*, arXiv:1606.01583, 2016. URL https://arxiv.org/abs/1606.01583.

[119] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.

[120] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2094–2107, 2014.

[121] Y. Chen, X. Zhao, and X. Jia. Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2381–2392, 2015.

[122] Y. Li, H. Zhang, and Q. Shen. Spectral spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sensing*, 9(1):67, 2017.

[123] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015:1–12, 2015.

[124] S. Mei, J. Ji, Q. Bi, J. Hou, Q. Du, and W. Li. Integrating spectral and spatial information into deep convolutional neural networks for hyperspectral classification. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 5067–5070, 2016.

[125] X. Ma, H. Wang, J. Geng, and J. Wang. Hyperspectral image classification with small training set by deep network and relative distance prior. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3282–3285, 2016.

[126] P. Zhong, Z. Q. Gong, and C. Schönlieb. A Diversified Deep Belief Network for Hyperspectral Image Classification. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 443–449, 2016.

[127] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 4959–4962, 2015.

[128] J. Yue, W. Zhao, S. Mao, and H. Liu. Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sensing Letters*, 6 (6):468–477, 2015.

[129] W. Zhao, Z. Guo, J. Yue, X. Zhang, and L. Luo. On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery. *International Journal of Remote Sensing*, 36(13):3368–3379, 2015.

[130] E. Aptoula, M. C. Ozdemir, and B. Yanikoglu. Deep learning with attribute profiles for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 13(12):1970–1974, 2016.

[131] H. Liang and Q. Li. Hyperspectral imagery classification using sparse representations of convolutional neural network features. *Remote Sensing*, 8(2):99, 2016.

[132] Z. Lin, Y. Chen, X. Zhao, and G. Wang. Spectral-spatial classification of hyperspectral image using autoencoders. In *Proceedings of the 9th International Conference on Information, Communications Signal Processing*, pages 1–5, 2013.

[133] T. Li, J. Zhang, and Y. Zhang. Classification of hyperspectral image based on deep belief networks. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 5132–5136, 2014.

[134] H. Lee and H. Kwon. Contextual deep CNN based hyperspectral classification. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3322–3325, 2016.

[135] X. Ma, H. Wang, and J. Geng. Spectral–spatial classification of hyperspectral image based on deep auto-encoder. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(9):4073–4085, 2016.

[136] X. Han, Y. Zhong, and L. Zhang. Spatial-spectral classification based on the unsupervised convolutional sparse auto-encoder for hyperspectral remote sensing imagery. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-7:25–31, 2016.

[137] L. Mou, P. Ghamisi, and X. X. Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3639–3655, 2017.

[138] H. Wu and S. Prasad. Convolutional recurrent neural networks for hyperspectral data classification. *Remote Sensing*, 9(3):298, 2017.

[139] J. Li, J. Bioucas-Dias, and A. Plaza. Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):844–856, 2013.

[140] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson. Generative adversarial networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(9):5046–5063, 2018.

[141] P. Ghamisi, M. Mura, and J. Benediktsson. A survey on spectral-spatial classification techniques based on attribute profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2335–2353, 2015.

[142] J. Li, J. Bioucas-Dias, and A. Plaza. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4085–4098, 2010.

[143] J. Yuan, D. Wang, and R. Li. Remote sensing image segmentation by combining spectral and texture features. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):16–24, 2014.

[144] H. Lee and H. Kwon. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Transactions on Image Processing*, 26(10):4843–4855, 2017.

[145] W. Zhao and S. Du. Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8):4544–4554, 2016.

[146] S. Yu, S. Jia, and C. Xu. Convolutional neural networks for hyperspectral image classification. *Neurocomputing*, 219:88 – 98, 2017.

[147] W. Li, G. Wu, F. Zhang, and Q. Du. Hyperspectral image classification using deep pixel-pair features. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):844–853, 2017.

[148] S. C. Douglas. A novel endmember, fractional abundance, and contrast model for hyperspectral imagery. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2164–2168, 2013.

[149] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *Proceedings of the International Conference on Learning Representations*, 2015.

[150] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[151] M. Zeiler, G. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Proceedings of the 2011 International Conference on Computer Vision*, pages 2018–2025, 2011.

[152] G. Lin, C. Shen, A. Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016.

[153] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117. 2011.

[154] A. Kolesnikov, M. Guillaumin, V. Ferrari, and C. H. Lampert. Closed-form training of conditional random fields for large scale image segmentation. In *Proceedings of the 13th European Conference on Computer Vision*, pages 550–565, 2014.

[155] C. Sutton and A. McCallum. Piecewise training for undirected models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 568–575, 2005.

[156] Fast line-of-sight atmospheric analysis of hypercubes (FLAASH). https://www.harrisgeospatial.com/docs/FLAASH.html.

[157] P. Kaiser, J. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 2017.

[158] Y. Tarabalka, J. Chanussot, and J. Benediktsson. Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recognition*, 43 (7):2367–2379, 2010.

[159] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[160] K. He and J. Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5353–5360, 2015.

[161] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, Ea. G. Hansen, and W. M. Porter. The airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing of Environment*, 44(2):127 – 143, 1993.

[162] A. P. Cracknell. Review article synergy in remote sensing-what's in a pixel? *International Journal of Remote Sensing*, 19(11):2025–2047, 1998.

[163] José M Bioucas-Dias. A variable splitting augmented lagrangian approach to linear spectral unmixing. In *Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, 2009.

[164] M. Iordache, J. Dias, and A. Plaza. Collaborative sparse regression for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1): 341–354, 2014.

[165] W. Tang, Z. Shi, Y. Wu, and C. Zhang. Sparse unmixing of hyperspectral data using spectral a priori information. *IEEE Transactions on Geoscience and Remote Sensing*, 53(2):770–783, 2015.

[166] V. P. Pauca, J. Piper, and R. J. Plemmons. Nonnegative matrix factorization for spectral data analysis. *Linear Algebra and its Applications*, 416(1):29–47, 2006.

[167] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly. Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4282–4297, 2011.

[168] S. Jia and Y. Qian. Constrained nonnegative matrix factorization for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 47(1):161–173, 2009.

[169] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li. Manifold regularized sparse NMF for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5):2815–2826, 2013.

[170] G. Martín and A. Plaza. Region-based spatial preprocessing for endmember extraction and spectral unmixing. *IEEE Geoscience and Remote Sensing Letters*, 8 (4):745–749, 2011.

[171] L. Tong, J. Zhou, X. Li, Y. Qian, and Y. Gao. Region-based structure preserving nonnegative matrix factorization for hyperspectral unmixing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(4):1575–1588, 2017.

[172] Yi-Hsing TSENG. Spectral unmixing for the classification of hyperspectral images. In *International Archives of Photogrammetry and Remote Sensing. Vol. XXXIII, Part B7*, pages 1532–1538, 2010.

[173] A. Villa, J. Li, A. Plaza, and J. M. Bioucas-Dias. A new semi-supervised algorithm for hyperspectral image classification based on spectral unmixing concepts. In *2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4, 2011.

[174] A. Villa, J. Chanussot, J. A. Benediktsson, and C. Jutten. Spectral unmixing for the classification of hyperspectral images at a finer spatial resolution. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):521–533, 2011.

[175] I. Dópido and A. Plaza. Unmixing prior to supervised classification of urban hyperspectral images. In *2011 Joint Urban Remote Sensing Event*, pages 97–100, 2011.

[176] L. Fang, S. Li, X. Kang, and J. A. Benediktsson. Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8):4186–4201, 2015.

[177] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[178] J. Nievergelt and K. H. Hinrichs. *Algorithms and Data Structures: With Applications to Graphics and Geometry.* Prentice-Hall, Inc., 1993.

[179] Z. Li, J. Liu, and H. Lu. Structure preserving non-negative matrix factorization for dimensionality reduction. *Computer Vision and Image Understanding*, 117(9): 1175–1189, 2013.

[180] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels. *EPFL Technical Report*, (149300), 2010.

[181] X. Zhang, Selene E. Chew, Z. Xu, and N. D. Cahill. SLIC superpixels for efficient graph-based dimensionality reduction of hyperspectral imagery. *Proc. SPIE*, 9472 (S2), 2015.

[182] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen. Learning contextual dependence with convolutional hierarchical recurrent neural networks. *IEEE Transactions on Image Processing*, 25(7):2983–2996, 2016.

[183] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 282–289, 2001.

[184] M. Waechter, N. Moehrle, and M. Goesele. Let there be color! large-scale texturing of 3D reconstructions. In *Computer Vision – ECCV 2014*, 2014.

[185] C. Li, J. Wang, L. Wang, L. Hu, and P. Gong. Comparison of classification algorithms and training sample sizes in urban land classification with landsat thematic mapper imagery. *Remote Sensing*, 6(2):964–983, 2014.

[186] F. I. Alam, J. Zhou, A. W. Liew, and X. Jia. CRF learning with CNN features for hyperspectral image segmentation. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 6890–6893, 2016.

[187] J. Xia, J. Chanussot, P. Du, and X. He. Spectral–spatial classification for hyperspectral data using rotation forests with local feature extraction and markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5): 2532–2546, 2015.

[188] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[189] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017.

[190] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. *CoRR*, abs/1704.00028, 2017. URL http://arxiv.org/abs/1704.00028.

[191] F. I. Alam, J. Zhou, A. W. Liew, J. Jo, and Y. Gao. Triplet constrained deep feature extraction and classification of hyperspectral images. In *To appear in the proceedings of the 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2018.

[192] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, pages 448–456, 2015.

[193] F. I. Alam, J. Zhou, A. W. Liew, X. Jia, J. Chanussot, and Y. Gao. Conditional random field and deep feature learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1612–1628, 2019.

[194] P. Ghamisi, R. Souza, J. A. Benediktsson, L. Rittner, R. Lotufo, and X. X. Zhu. Hyperspectral data classification using extended extinction profiles. *IEEE Geoscience and Remote Sensing Letters*, 13(11):1641–1645, 2016.