Spectral-Spatial Neural Networks and Probabilistic Graph Models for Hyperspectral Image Classification

by

Zilong Zhong

A thesis thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	**Prof. Shengrui Wang** Computer Science, University of Sherbrooke
Supervisor:	**Prof. Jonathan Li** Systems Design Engineering, University of Waterloo
Supervisor:	**Prof. Alexander Wong** Systems Design Engineering, University of Waterloo
Internal Member:	**Prof. David A. Clausi** Systems Design Engineering, University of Waterloo
Internal Member:	**Prof. John Zelek** Systems Design Engineering, University of Waterloo
Internal-External Member:	**Prof. Michael Chapman** Geomatics, University of Waterloo

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Content from 5 papers are used in this thesis. I was the co-author with major contributions on designing the methods, implementation and writing the papers. Other co-authors also contributed to these works:

Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectra-spatial residual network for hyperspectral image classification: A 3-D deep learning framework", IEEE Transactions on Geoscience and Remote Sensing, 2018. **[ESI Hot & Highly Cited Paper]**

This paper is incorporated in Chapter 3 and 6 of this thesis.

Z. Zhong, J. Li, D. Clausi, and A. Wong, "Generative adversarial networks and conditional random fields for hyperspectral image classification", IEEE Transactions on Cybernetics, DOI: 10.1109/TCYB.2019.2915094, 2019.

This paper is incorporated in Chapter 4 and 6 of this thesis.

Z. Zhong, J. Li, W. Cui, and H. Jiang, "Fully convolutional networks for building and road extraction: Preliminary results", 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2016.

This paper is incorporated in Chapter 3 of this thesis.

Z. Zhong, J. Li, L. Ma, H. Jiang, and H. Zhao, "Deep Residual Networks for Hyperspectral Image Classification", 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017.

This paper is incorporated in Chapter 3 of this thesis.

Z. Zhong, J. Li, "Generative Adversarial Networks and Probabilistic Graph Models for Hyperspectral Image Classification", Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), 2018.

This paper is incorporated in Chapter 5 and 6 of this thesis.

Abstract

Pixel-wise hyperspectral image (HSI) classification has been actively studied since it shares similar characteristics with related computer vision tasks, including image classification, object detection, and semantic segmentation, but also possesses inherent differences. The research surrounding HSI classification sheds light on an approach to bridge computer vision and remote sensing. Modern deep neural networks dominate and repeatedly set new records in all image recognition challenges, largely due to their excellence in extracting discriminative features through multi-layer nonlinear transformation. However, three challenges hinder the direct adoption of convolutional neural networks (CNNs) for HSI classification. First, typical HSIs contain hundreds of spectral channels that encode abundant pixel-wise spectral information, leading to the curse of dimensionality. Second, HSIs usually have relatively small numbers of annotated pixels for training along with large numbers of unlabeled pixels, resulting in the problem of generalization. Third, the scarcity of annotations and the complexity of HSI data induce noisy classification maps, which are a common issue in various types of remotely sensed data interpretation.

Recent studies show that taking the data attributes into the designing of fundamental components of deep neural networks can improve their representational capacity and then facilitates these models to achieve better recognition performance. To the best of our knowledge, no research has exploited this finding or proposed corresponding models for supervised HSI classification given enough labeled HSI data. In cases of limited labeled HSI samples for training, conditional random fields (CRFs) are an effective graph model to impose data-agnostic constraints upon the intermediate outputs of trained discriminators. Although CRFs have been widely used to enhance HSI classification performance, the integration of deep learning and probabilistic graph models in the framework of semi-supervised learning remains an open question.

To this end, this thesis presents supervised spectral-spatial residual networks (SSRNs) and semi-supervised generative adversarial network (GAN) -based models that account for the characteristics of HSIs and make three main contributions. First, spectral and spatial convolution layers are introduced to learn representative HSI features for supervised learning models. Second, generative adversarial networks (GANs) composed of spectral/spatial convolution and transposed-convolution layers are proposed to take advantage of adversarial training using limited amounts of labeled data for semi-supervised learning. Third, fully-connected CRFs are adopted to impose smoothness constraints on the predictions of the trained discriminators of GANs to enhance HSI classification performance. Empirical evidence acquired by experimental comparison to state-of-the-art models validates the effectiveness and generalizability of SSRN, SS-GAN, and GAN-CRF models.

Acknowledgements

I would like to express my deep gratitude to my two co-supervisors, Prof. Jonathan Li and Prof. Alexander Wong for their invaluable advice and amazing help for guiding me to achieve one challenging academic goal after another. I sincerely believe that it is just the beginning and our collaboration will remain strong in the foreseeable future.

Many thanks to my colleagues and friends in Vision and Image Processing lab as well as Mobile Sensing and Geodata Science lab at the University of Waterloo, for establishing a collaborative and efficient environment and always being ready to help. Also, I would like to give special thanks to Nadine Fladd of the Writing and Communication Center, for her professional suggestions and patient explanations during my four-year PhD study.

I would like to thank my thesis committee members Prof. David Clausi and Prof. John Zelek from Systems Design Engineering department at the University of Waterloo, and Prof. Michael Chapman from Ryerson University for their time and commitment. I also would like to thank Prof. Shengrui Wang from Sherbrooke University for his time and acceptance of reviewing my thesis.

Most importantly, I would like to thank my family, including my parents, my wife, and my newborn daughter, for their incredible love and unreserved supports throughout these years.

Table of Contents

List of Tables

List of Figures

List of Nomenclature

β	batch normalization bias to be learned
ϵ	arbitrarily small positive quantity
γ	batch normalization weight to learn
μ	mean of input feature maps
∇	gradient
σ^2	variance of input feature maps
Θ	parameters to be learned
θ	parameters of spectral convolution kernels
θ_D	parameters of discriminator in a GAN
θ_G	parameters of generator in a GAN
ξ	parameters of spatial convolution kernels
CE	cross entropy function
D	discriminator of a GAN
E	energy function
G	generator of a GAN
H	spatial convolution kernels
h	spectral convolution kernels

L_G	loss function of the generator in a GAN
L_{SEMI}	loss function of a GAN
L_{SUP}	loss function of the discriminator in a GAN
$LR(\cdot)$	leaky rectified linear unit
M	models to be designed
P	pairwise term of energy function
$Prob$	probability distribution
Q	approximated probability distribution
$R(\cdot)$	rectified linear unit
U	unary term of energy function

Chapter 1

Introduction

Hyperspectral image (HSI) classification means labeling hyperspectral pixels in imagery as belonging to pre-defined land cover categories. This task forms the cornerstone of a wide range of applications, including object detection, anomaly detection, semantic segmentation, and land-cover mapping [27,47,86,87,100]. HSIs possess two distinctive characteristics different from natural rgb images. First, HSI pixels consist of hundreds of contiguous spectral bands. This abundant spectral information makes the accurate identification of corresponding ground cover classes possible [88]. Second, HSI pixels sampled from homogeneous areas are highly correlated. This spatial correlation provides complementary information to spectral signatures for precise mapping [46]. These two differences prevent machine learning models, e.g. convolutional neural networks (CNNs), that achieve high accuracy for natural image recognition from directly transferring their successes to HSI classification.

Traditional pixel-level HSI classification models mainly concentrate on two steps:

1. **Feature engineering:** Feature engineering methods include feature selection (band selection) and feature extraction [31]. The main objectives of feature engineering are to reduce the high dimensionality of HSI pixels and extract the most discriminative features or bands [85]. Feature extraction approaches usually learn representative features through nonlinear transformation. Unlike feature extraction, feature selection methods try to find the most representative features from raw HSIs without transforming them to retain their physical meaning [69].

2. **Classifier training:** Discriminative and generative are two main approaches to determine the parameters of linear classifiers using the features obtained from the feature

engineering step. Discriminative methods are used to model the projection from input data to their annotations or to model the conditional probabilities, like logistic regression and support vector machine (SVM). In contrast, generative models are adopted for modeling the joint distribution of input data and their labels, like naive Bayes model.

Although the two-step paradigm has been used for HSI classification by a lot of research, these classic methods suffer from two drawbacks:

1. **Low generalizability:** The adoption of dimension reduction methods leads to inevitable loss of information and therefore the extracted or selected features usually do not generalize well to other applications.

2. **Shallow representation:** The shallow learning methods (e.g., logistic regression) or band selection being applied before the linear classifiers has limited representational capacity to fully utilize the abundant spectral and spatial HSI features.

Therefore, the objective of this dissertation is to design specific convolutional blocks that embed HSI attributes along with novel deep neural networks built consisting of these blocks, while achieving high classification accuracy compared to state-of-the-art deep learning models in multiple cases.

1.1 Problem Definition

Suppose an HSI dataset contains n labeled samples $X^{Tr} = \{X_i^{Tr}\}_{i=1}^{n} \in \mathbb{R}^{w \times w \times b}$ and m one-hot labels $y^{Tr} = \{y_j^{Tr}\}_{j=1}^{m} \in \mathbb{R}^{1 \times 1 \times L}$ for training. W and B denote the spatial and spectral sizes of HSI samples, respectively. L represents the total number of land cover categories. Machine learning methods address this task by searching for a non-linear transformation function $F(\cdot; \Theta, M)$ to fit training data $\{X^{Tr}, y^{Tr}\}$ such that this trained model can generalize to unseen HSI samples $X^{Va} = \{X_i^{Va}\}_{i=1}^{l} \in \mathbb{R}^{w \times w \times b}$ and make reasonable predictions. Θ and M denote the parameters to be learned and the model to be designed, respectively. However, the analytical solutions are non-trivial to compute as a result of the high dimension of HSI samples and the hierarchical architecture of neural networks. Therefore, the optimal parameters Θ^* of a given model M are indirectly approximated by an equivalent optimization problem as follows:

$$
\begin{aligned}
\Theta^* &= \arg\min_{\Theta} Loss(y^{Tr}, \hat{y}^{Tr}) \\
&= \arg\min_{\Theta} Loss(y^{Tr}, F(X^{Tr}; \Theta, M)) \\
&= \arg\min_{\Theta} -\frac{1}{N} \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{L} y_{ij}^{Tr} \log F_{ij}(X^{Tr}; \Theta, M),
\end{aligned}
\tag{1.1}
$$

where cross entropy loss is used to measure the discrepancy between predictions $\hat{y}^{Tr} = F(X^{Tr}; \Theta, M)$ and targets y^{Tr}. When $n = m$, in which X^{Tr} contains only labeled samples, HSI classification is a typical supervised learning task. When $n > m$, in which extra unlabelled data are used during training, this problem belongs to semi-supervised learning. Assume that optimal parameters of different models can be learned. To achieve high HSI classification performance, the key point lies in designing network modules with data-specific attributes that can facilitate extracting discriminative spectral-spatial HSI features.

1.2 Challenges and Objectives

Recent studies suggest that CNNs can extract discriminative features from images for vision related tasks and multiple works have adopted deep learning models for HSI classification and achieved promising performance [8, 48, 91]. However, there are still three challenges that hinder deep neural networks from offering precise pixel-wise HSI classification maps:

1. **Hundreds of spectral bands:** The first challenge derives from the characteristics of HSIs, which comprise two spatial dimensions and one spectral dimension, specially the hundreds channels in the spectral dimension make it hard to directly adopt the deep learning models to HSI analysis. Many papers indicate that both spectral and spatial features play important roles in precise HSI interpretation. However, most deep learning applications for HSI classification overlook the distinctive characteristics of this remotely sensed data.

2. **Limited annotated samples:** The second challenge stems from the high cost of and difficulty in obtaining a large amount of labeled data for HSIs. The shortage of labelled pixels limits the classification performance of deep learning models. Many papers indicate that CNNs require a large amount of training data and propose methods to augment training data via adding noise to HSI pixels [8]. Additionally, [48] presents a pixel-pair

approach that samples two pixels independently and couples them as a group for the purpose of enlarging the number of HSI samples for training.

3. **Noisy classification maps:** The third challenge is caused by the complex spatial distribution of HSIs and the similarities between HSI samples. Also, the second challenge contribute to the fact that classification maps generated by CNNs tend to be noisy and have spurious object edges. How to utilize data-agnostic smoothness priors and large numbers of unlabeled HSI pixels to filter out noisy remain open questions to explore.

The objectives of the designed models are to embed HSI attributes into the designing of special convolution blocks, to alleviate the shortage of labeled HSI samples, and to generate less noisy HSI classification outputs, respectively.

1.3 Thesis Contributions

The contributions of this thesis are listed as follows:

1. A novel spectral-spatial residual network (SSRN) that integrates the spectral-spatial attributes of HSIs into the designing of convolution layers is proposed for supervised HSI classification, which is described in Chapter 3,

2. A novel spectral-spatial generative adversarial network (SS-GAN) that constructs a semi-supervised learning framework to take advantage of adversarial training is introduced for semi-supervised HSI classification, which is described in Chapter 4, and

3. A fully connected conditional random field that imposes graph constraints on the softmax outputs of trained models using limited numbers of labeled HSI samples is studied to improve semi-supervised classification outputs, which is described in Chapter 5.

Chapter 2

Overview of HSI Classification Methods

“ In deep learning, the algorithms we use now are versions of the algorithms we were developing in the 1980s, the 1990s. People were very optimistic about them, but it turns out they did not work too well. Now we know the reason is they did not work too well is that we did not have powerful enough computers, we did not have enough data sets to train them. If we want to approach the level of the human brain, we need much more computation, we need better hardware. ”

– Geoffrey Hinton

This chapter presents an overview of HSI representation learning methods to improve classification accuracy, and a discussion about in which situation these methods should be adopted as well as their strengths and shortcomings. The mathematical formulation of designed models and their related works are provided in following chapters. An overview of supervised HSI classification is presented in Chapter 3. An overview of semi-supervised HSI classification is presented in Chapter 4. An overview of graphical probabilistic models for boosting HSI classification results is presented in Chapter 5.

2.1 Types of Representation Learning

Representation learning or feature learning plays the crucial role for finding relevant patterns to be judged from and therefore achieving high HSI classification accuracy in various circumstances. According to how training data being used, HSI classification methods can be classified into four mainstream representation learning branches:

1. Methods based on supervised feature learning [31, 46, 88, 99]
2. Methods based on unsupervised feature learning [83, 84, 103]
3. Methods based on semi-supervised feature learning [83, 84, 103]
4. Methods based on probabilistic graphical constraints [29, 66, 77, 94, 95]

2.2 Supervised Methods

Supervised feature learning methods use sample-label pairs $\{(X_1, Y_1), ...(X_N, Y_N)\}$ to train models M parameterized by Θ, such that trained models $M(\cdot; \Theta)$ represent the mapping from input space of $\{X_i\}$ to target space of $\{Y_i\}$ and this mapping can generalize to unseen testing data $\{X_t\}$. The underlying assumption herein is that the training data $\{X_i\}$ and testing data $\{X_t\}$ are sampled from the same data distribution. The supervised learning methods can be further subdivided into two types: 1) discriminative approaches, 2) generative approaches. Discriminative approaches make predictions on input features without modeling a probability distribution. Traditional discriminative approaches include linear regression, logistic regression, decision trees, and support vector machines [66]. Since these approaches lack of representational capacity, therefore the distinctiveness of extracted features used as input of the classifiers decide classification accuracy. Generative approaches model joint distribution of $\{X, Y\}$ and assume that targets $\{Y\}$ follow certain priors (e.g. Gaussian distribution). Then, the marginal distribution $P(Y|X)$ are used to make prediction on unseen data. Typical generative approaches include Gaussian Mixture model, Hidden Markov model, and Bayesian Networks [59]. Although the prior distribution assumption make the generative model more explainable and robust to missing data, this assumption is not flexible enough to generalize to complicated scenarios. In general, the one-to-one correspondence of data and annotations $\{X_i, Y_i\}$ enables the supervised learning methods to obtain most accurate classification results among all feature learning branches. Methods based on supervised learning are mainly useful for circumstances that the amount of labeled data are large enough given the complexity of selected models to support discriminative feature learning, or high accuracy instead of computationally efficiency is the priority.

2.3 Unsupervised Methods

Unsupervised feature learning methods use only samples $\{X_i\}$ without annotation to learn their shared features that can support the training of classifiers that follows. The underlying assumption herein is that learned features extracted from unlabeled training samples $\{X_i\}$ can increase the distance of samples between different classes and decrease the distance of samples within same classes in feature spaces. Typical unsupervised feature learning approaches include principle component analysis (PCA), auto-encoders(AEs), generative adversarial networks (GANs) [22]. No annotation involvement in feature learning limits the usage of unsupervised learning approaches to achieve high classification accuracy, and therefore this research branch is more useful for data generating tasks [99]. Methods based on unsupervised learning can benefit circumstances that the annotation of samples is hardly available, or the classification task is trivial to the extent that unsupervisedly learned features can be easily clustered into separable groups.

2.4 Semi-supervised Methods

Intuitively, semi-supervised feature learning stands between supervised and unsupervised learning. Semi-supervised methods usually use a small amount of labeled samples $\{X_i, Y_i\}$ and a large amount of unlabeled samples $\{X_j\}$ for training to learn the mapping from data space to target space. To make use of unlabeled data $\{X_j\}$, smoothness assumption or manifold assumption will be used to connect labeled and unlabeled data via iterative propagating predictions from them. Typical semi-supervised feature learning approaches includes generative models, graph-based methods, heuristic approaches. It is worth to note that the difference of generative models used here compared to those used for supervised learning lies in the model parameters Θ are used to fit both labeled $\{X_i\}$ and unlabeled data $\{X_j\}$ during training. The usage of limited amount of annotations make this feature learning branch outperforms its unsupervised counterparts, but underperforms supervised ones. Methods based on semi-supervised learning can benefit circumstances that the limited labels of samples are available and large amount of unlabeled data share similar data distribution with labeled ones used for training.

2.5 Probabilistic Graphical Methods

Probabilistic graphical models are an approach that models joint probability distribution $P(Y, X)$ or conditional probability distribution $P(Y|X)$. This approach integrates graph theory and probability modeling, imposing smoothness constraints as priors on data or label distributions [7, 89]. Typical probabilistic graphical models include Markov random fields (MRFs) and conditional random fields (CRFs). MRFs model joint distribution $P(Y, X)$ and share the advantages and disadvantages of common generative models in the sense that they are suitable for generating synthetic, but not for delivering high classification accuracy. On the other hand, CRFs directly model conditional probability distribution $P(Y|X)$ and is the most widely used graphical approach. The underlying assumption herein is a smoothness prior whereby neighboring pixels with similar spectral signatures tend to have the same labels [94]. Since CRFs can be regarded as a structured generalization of multinomial logistic regression, the conditional probability distribution of a CRF takes the form as follows:

$$Prob(Y|Z) = \frac{\exp(-E(Y|Z))}{\sum_{Y=y} \exp(-E(Y=y|Z))}, \tag{2.1}$$

where Y and Z denote output random variables and their corresponding observed features. $E(\cdot)$ is an energy function that models the joint probability distribution of Y and Z. The optimal random variables can be calculated by the *maximum a posteriori* (MAP) estimation:

$$Y^{MAP} = \underset{Y}{argmax} Prob(Y|Z). \tag{2.2}$$

However, equation (2.2) is intractable to solve directly for image recognition tasks, and then approximation methods like message passing algorithm are used to make it feasible to solve

[36]. Methods based on probabilistic graphical models are useful for circumstances that the smoothness assumption can benefit classification or segmentation results, especially the case that labelled samples are scarce and therefore mapping outputs suffer from noise.

2.6 Discussion

According to the above description, the most promising methods to achieve high HSI classification accuracy are supervise and semi-supervised feature learning approaches. Given enough labeled data for training, supervised feature models are undoubtedly the best one to employ. Given labeled samples is too few to make a reasonable estimation of data distribution, semi-supervised approaches that take advantage unlabeled data appear to be a competitive choice. However, both these two methods involve some drawbacks to overcome.

For supervised methods, the available amount of training and testing samples in the widely studied HSI datasets are relatively small compared to a large number of annotated data in computer vision community [39]. This is problematic because models with high representational capacity need large numbers of samples to train, and small numbers of training samples can yield overfitting to these samples with the loss of generality. Moreover, hundreds of spectral channels of each HSI pixel make the traditional supervised methods more vulnerable to the "curse of dimensionality", which means the number of samples that required to make accurate estimation grow exponentially with the increase of dimension of feature space. Many research works adopted dimension reduction methods to alleviate this phenomenon [88]. However, the dimension reduction methods inevitably lead to the loss of useful information because it is hard to directly judge which features are discriminative.

For semi-supervised methods, the main problem derive from the assumption that labeled and unlabeled data share the same data distribution. If this assumption holds, the unlabeled data is helpful to enhance classification results. Otherwise, unifying labeled and unlabeled data for training brings detriment to classification accuracy. Additionally, since limited annotated samples are used for training, the classification output tend to be noisy. To mitigate the noisy classification maps, CRFs can used to add the graphical smoothness constraint. However, traditional CRFs only impose this prior on neighboring samples and adding global constraints is computationally infeasible.

Therefore, to achieve high classification performance, the proposed supervised models should overcome the "curse of dimensionality" and learn discriminative features from training HSI samples. The proposed semi-supervsied models should find useful unlabeled data for training and incorporate probabilistic graph models that suitable for HSI classification.

Chapter 3

Supervised Spectral-Spatial Networks for pixel-wise Classification

“ We do not believe that having the newest computer or the largest cluster is the key to success, but rather utilizing modern techniques and the latest research with a clear understanding of the problem we are trying to solve. ”

– Jeremy Howard

This chapter presents an end-to-end spectral-spatial residual network (SSRN) that takes raw 3D volumes as input data without feature engineering for hyperspectral image classification. In this network, the spectral and spatial residual blocks consecutively learn discriminative features from abundant spectral signatures and spatial contexts in hyperspectral imagery (HSI). The proposed SSRN is a supervised deep learning framework that alleviates the declining-accuracy phenomenon of other deep learning models. Specifically, the residual blocks connect every other 3D convolutional layer through identity mapping, which facilitates the back propagation of gradients. Furthermore, batch normalization is adopted on every convolutional layer to regularize the learning process and improve the classification performance of trained models.

3.1 Conventional Methods

Classifying every pixel with a certain land cover type is the cornerstone of remotely sensed data analysis, which spans a broad range of applications, including image segmentation, object recognition, land-cover mapping, and anomaly detection [15, 18, 19, 38, 71, 81, 98]. Among various types of remotely sensed data, the attributes of HSIs make them a activated studied objective to bridge the fields of computer vision and remote sensing [27, 43, 47, 74, 75, 86, 87]. Two major characteristics of HSI should be taken into account to obtain discriminative features for HSI classification. First, abundant spectral information, which derives from hundreds of contiguous spectral bands, makes the accurate identification of corresponding ground materials possible [88]. Second, high spatial correlation, which originates from homogeneous areas in HSI, provides complementary information to spectral features for precise mapping [46].

To take advantage of abundant spectral bands, traditional pixel-wise HSI classification models mainly concentrate on two steps: feature engineering and classifier training. Feature engineering methods include feature selection (band selection) and feature extraction [31]. The main objectives of feature engineering are to reduce the high dimensionality of HSI pixels and extract the most discriminative features or bands. Next, general-purpose classifiers are trained using the discriminative features obtained from the feature engineering step. Feature extraction approaches usually learn representative features through nonlinear transformation. For example, [85] integrated multiple features derived from different kinds of dimensionality reduction methods to train SVM classifiers. Unlike feature extraction, feature selection methods try to find the most representative features from raw HSIs without transforming them to retain their physical meaning. For instance, [69] adopted manifold ranking as an unsupervised feature selection method, which chooses the most representative bands for training the classifiers that follow. Moreover, a multitask joint sparse representation based method [83] integrated band selection method with a smooth prior imposed by Markov random field. These two band selection based paradigms used spectral bands from all available pixels for feature selection and can be interpreted as semi-supervised learning methods.

On the other hand, there are two ways to incorporate spatial information for HSI classification: spatialized input and post-processing. The spatialized input methods impose feature engineering step on 3D cuboids obtained from HSI. Many papers suggested that methods expanding input data with more spatial information can improve classification performance [32, 33]. Among these methods, support vector machines (SVMs) are the most commonly used classifiers for HSI classification, because SVMs perform robustly with high dimensional input data [17, 54]. For example, [55] employed a region-based kernel to extract spectral-spatial features on which the learned SVM classifier identifies the categories of hyperspectral pixels. In contrast, the post-processing approaches have taken the prior knowledge of smoothness into consideration that neighboring pixels with similar spectral information are likely to belong to the same land cover categories. For instance, [66] incorporated a probabilistic graphical model as the post-processing step to improve the classification outcomes of kernel SVMs. Although many works use typical classification frameworks, which are composed of feature extractors followed by trainable classifiers, they suffer from two drawbacks. First, the feature engineering step normally does not generalize well to other scenarios. Second, the de facto one-layer nonlinear transformation (e.g., kernel methods) being applied before the linear classifiers has limited representation capacity to fully utilize the abundant spectral and spatial features.

3.2 Deep Learning Methods

In the face of these shortcomings of feature engineering based frameworks, supervised deep learning models have attracted increased attention, due to the fact that the objective functions of deep learning models directly focus on classification in lieu of two independent steps. The fundamental philosophy of deep learning is to let the trained model itself decide which features are more important than other features with fewer constraints imposed by human beings. In other words, deep learning frameworks simultaneously learn feature representation and corresponding classifiers through training process. Furthermore, multilayer neural networks can extract robust and discriminative features of HSI and outperform SVMs [9,10]. For example, the stacked autoencoders (SAEs) were used as feature extractors to capture the representative stacked spectral and spatial features with a greedy layer-wise pre-training strategy [9]. Similarly, the potential of deep belief networks (DBNs) for HSI classification was explored in [10]. However, both models suffer the same problem of spatial information loss, which is caused by the requirement for one-dimensional input data.

Recently, convolutional neural networks (CNNs) and their extensions have obtained unprecedented advances in computer vision tasks [37, 39]. Multiple papers have demonstrated that CNNs can deliver state-of-the-art results using spatialized input for HSI classification [8, 48, 91]. For example, [91] used CNNs to extract spatial features, which were integrated with spectral features that learned from balanced local discriminant embedding, for HSI classification. However, the input of the CNN models are the three principal components of original HSIs, which means the spatial feature extraction process still loses some spectral-spatial information. A CNN-based feature extractor was proposed in [48], which can learn discriminative representations from pixel pairs and use a voting strategy to smooth final classification maps. In addition, 3D CNNs were adopted to extract deep spectral-spatial features directly from raw HSIs and delivered promising classification outcomes [8]. Similarly, [49] further studied 3D CNNs for spectral-spatial classification using input cuboids of HSIs with smaller spatial size. These models generate thematic maps using an approach that can directly process raw HSIs, whereas the classification accuracy of the CNN models decreases when the network becomes deeper.

To resolve this problem, inspired by [24], a supervised spectral-spatial residual network (SSRN[1]) is proposed with consecutive learning blocks that takes the characteristics of HSI into account. The designed spectral and spatial residual blocks extract discriminative spectral-spatial features from HSI cuboids and can be regarded as an extension of convolutional layers in CNNs. The SSRN has a deeper structure than those of 3D CNNs used in [8, 48, 49, 91], and contains shortcut connections between every other convolutional layer. Hence, the SSRN can learn robust spectral-spatial representations from original HSIs. Similar to the SSRN, [41] incorporated residual learning with fully convolutional layers to form a contextual CNN. However, this method fails to distinguish spectral features and spatial features. Thus, this thesis investigates the effectiveness of two types of residual architecture toward the spectral-spatial feature learning for HSI classification, and their robustness in different scenarios.

Compared to a large number of annotated data in computer vision and pattern recognition communities, which play a significant role in the unprecedented success achieved by deep learning models [39], the available amount of training and testing samples in the widely studied HSI datasets are relatively small. Moreover, the unbalanced amounts of differently labeled samples undermine the accuracy of HSI classification. In addition,

[1] https://github.com/zilongzhong/SSRN

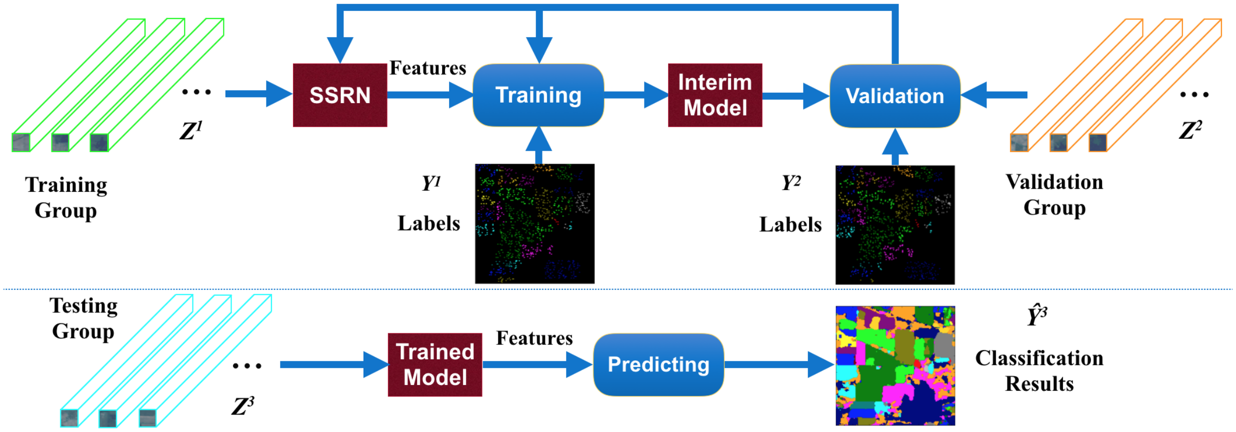


Figure 3.1: Spectral-Spatial Residual Network -based framework for HSI classification. In the upper section, the training group Z^1 and their corresponding labels are used for updating the parameters of network. The validation group Z^2 and their corresponding labels Y^2 are used for monitoring the interim models generated in the training stage. In the lower section, the testing group Z^3 is employed for assessing the optimal trained network.

the input data of SSRN are 3D cuboids of raw HSI and the multidimensional input data brings more challenges. Therefore, this chapter aims to study the generalization ability of the SSRN on HSI datasets with large and small training sizes, high and medium spatial resolution, and various land-cover types with uneven samples for different categories.

3.3 Proposed Network

Figure 3.1 shows the whole deep learning framework of HSI classification based on SSRN. In this framework, all available annotated data are separated into three groups: training, validation, and testing groups for each dataset. Suppose the HSI dataset X contains N labeled pixels $\{x_1, x_2, ..., x_N\} \in \mathbb{R}^{1\times1\times b}$ and $Y = \{y_1, y_2, ..., y_N\} \in \mathbb{R}^{1\times1\times L}$ is the set of corresponding one-hot label vectors, where b and L represent the numbers of spectral bands and land cover categories, respectively. Neighboring cuboids centered at pixels in X form a new group of dataset $Z = \{z_1, z_2, ..., z_N\} \in \mathbb{R}^{w\times w\times b}$. To fully utilize the spectral and spatial information provided by HSIs, the proposed networks take cuboids of size $w \times w \times b$ from raw data as input, where is the short width of 3D cuboids in training group Z^1, validation group Z^2, and testing group Z^3 in Figure 3.1. Their corresponding label vector sets are Y^1, Y^2, and Y^3. For example, the size of HSI cuboids for the Indian Pines dataset is $7 \times 7 \times 200$. Therefore, the objective of training process is to update the parameters of SSRN till the model can make high-accuracy predictions $\hat{Y}^3$ with regard to the ground truth labels Y^3 given the neighboring cuboids Z^3.

After the architecture of deep learning models is built and the hyper parameters for training are configured, the models are trained for hundreds of epochs using the training group Z^1 and their ground truth label vector set Y^1. In this process, the parameters of SSRN are updated through back propagating the gradients of the cross-entropy objective function:

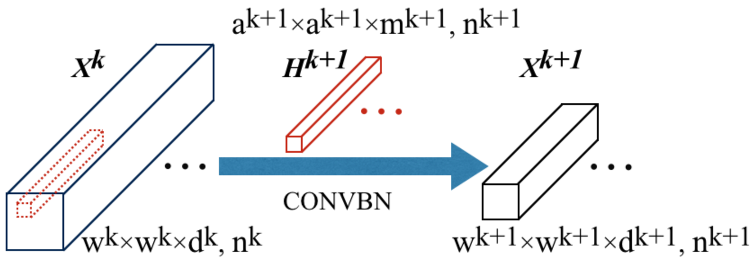
$$\begin{aligned} CE(\hat{y}, y) &= -\sum\nolimits_{i=1}^{L} y_i log \frac{e^{\hat{y}_i}}{\sum\nolimits_{j=1}^{L} e^{\hat{y}_j}} \\ &= \sum\nolimits_{i=1}^{L} y_i log \frac{\sum\nolimits_{j=1}^{L} e^{\hat{y}_j}}{e^{\hat{y}_i}} \\ &= \sum\nolimits_{i=1}^{L} y_i (\log \sum\nolimits_{j=1}^{L} e^{\hat{y}_j} - \hat{y}_i), \end{aligned} \tag{3.1}$$

where $CE(\cdot)$ represents the cross-entropy function. This function measures the difference between predicted label vector $\hat{y} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_L]$ and ground truth label vector $y = [y_1, y_2, ..., y_L]$, which is the vector output of the last fully connected layer without using softmax function.

The validation group Z^2 is used for monitoring training process by measuring the classification performance of interim models, which are intermediate networks generated during the training stage, to select the network with the highest classification accuracy. Finally, the testing group Z^3 is employed for assessing the generalizability of the trained SSRN through calculating classification metrics and visualizing thematic maps.

3.3.1 3D Convolutional Layer with Batch Normalization

Deep learning models consist of multiple layers of nonlinear neurons that can learn hierarchical representations through a large number of labeled images [37]. CNNs have achieved or surpassed human level intelligence in several perception tasks [39, 51], because convolutional layers enable CNNs to learn more discriminative features with sparsity constraint.



$a^{k+1} \times a^{k+1} \times m^{k+1}$, n^{k+1}
X^k
H^{k+1}
X^{k+1}
CONVBN
$w^k \times w^k \times d^k$, n^k
$w^{k+1} \times w^{k+1} \times d^{k+1}$, n^{k+1}

Figure 3.2: 3D Convolutional layer with batch normalization. The $(k+1)$th layer conducts a 3D convolution of input feature cuboids X^k and a convolutional filter bank H^{k+1} and generates output feature cuboids X^{k+1}.

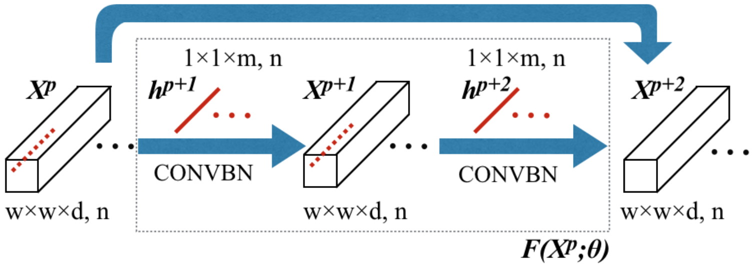


Figure 3.3: Spectral residual block for spectral feature learning. This block includes two successive 3D convolutional layers, and a skip connection directly adds input feature cuboids X^p to output feature cuboids X^{p+2}.

In this thesis, 3D convolutional layers are adopted as the basic element of the SSRN. In addition, batch normalization [28] is conducted at every convolutional layer in SSRN. This strategy makes the training processing of deep learning models more efficient. As shown in Figure 3.2, if the $(k+1)$th 3D convolutional layer has n^k input feature cuboids of size $w^k \times w^k \times d^k$, a convolutional filter bank that contains n^{k+1} convolutional filters of size $a^{k+1} \times a^{k+1} \times m^{k+1}$, and the subsampling strides of (s_1, s_1, s_2) for the convolutional operation, then this layer generate n^{k+1} output feature cuboids of size $w^{k+1} \times w^{k+1} \times d^{k+1}$, where the spatial width $w^{k+1} = \lfloor 1 + (w^k - a^{k+1})/s_1 \rfloor$ and the spectral depth $d^{k+1} = \lfloor 1 + (d^k - m^{k+1})/s_2 \rfloor$. The ith output of $(k+1)$th 3D convolutional layer with batch normalization (CONVBN)

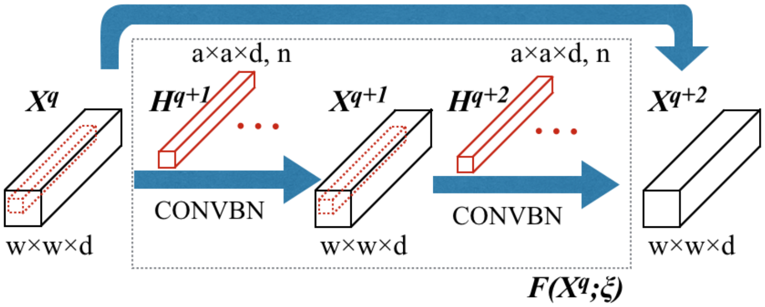


Figure 3.4: Spatial residual block for spatial feature learning. This block includes two successive 3D convolutional layers, and a skip connection directly adds input feature cuboids X^q to the output feature cuboids X^{q+2}.

can be formulated as

$$X_i^{k+1} = R(\sum\nolimits_{j=1}^{n^k} \hat{X}_j^k * H_i^{k+1} + b_i^{k+1}) \tag{3.2}$$

$$\mu(X^k) = \frac{1}{m}\sum\nolimits_{j=1}^{n^k} X_j^k \tag{3.3}$$

$$\sigma^2(X^k) = \frac{1}{m}\sum\nolimits_{j=1}^{n^k} (X_j^k - \mu(X^k))^2 \tag{3.4}$$

$$\tilde{X}^k = \frac{X^k - \mu(X^k)}{\sqrt{\sigma^2(X^k) + \epsilon}} \tag{3.5}$$

$$\hat{X}^k = \gamma\tilde{X} + \beta \tag{3.6}$$

where $X_j^k \in^{w \times w \times d}$ is the jth input feature tensor of the $(k+1)$th layer, $\hat{X}^k$ is the normalization result of batch feature cuboids X^k in the kth layer, $\mu(\cdot)$ and $\sigma^2(\cdot)$ represent the expectation and variance of the input feature tensor, respectively. ϵ is an arbitrarily small positive quantity to avoid the denominator be zero. γ and β are parameters to be learned. H_i^{k+1} and b_i^{k+1} denote the parameters and bias of the ith convolutional filter bank in the $(k+1)$th layer, $*$ represents 3D convolutional operation, and $R(\cdot)$ is the Rectified Linear Unit (ReLU) activation function that sets elements with negative numbers to zero.

3.3.2 Spectral and Spatial Residual Blocks

Although CNN models have been used for HSI classification and achieved state-of-the-art results, it is counterintuitive that, after several layers, the classification accuracy decreases with the increase of convolutional layers [8]. This phenomenon stems from the fact that the representation capacity of CNNs is too high compared to the relative small number of training samples with the same regularization settings. However, this decreasing-accuracy issue can be alleviated by adding shortcut connections between every other layer to build residual blocks [24]. To this end, two residual blocks are designed in a general architecture to consecutively extract spectral and spatial features from raw 3D HSI cuboids, owing to the high spectral resolution and high spatial correlation of HSI. As shown in Figure 3.3, a residual block can be regarded as an extension of two convolutional layers. This architecture enables gradients in higher layers rapidly propagate back to the lower layers, thereby facilitating and regularizing the model training process.

In the spectral residual blocks, as shown in Figure 3.3, convolution kernels/filters of size $1 \times 1 \times m$ are used in successive filter banks h^{p+1} and h^{p+2} for pth and $(p+1)$th layers, respectively. At the same time, the spatial size of 3D feature cuboids X^{p+1} and X^{p+2} is kept at $w \times w$ unchanged through a padding strategy, which means output feature cuboids copy the values from the border area to the padding area after convolutional operation in the spectral dimension. Then, these two convolutional layers build a residual function $F(X^p;\theta)$ instead of directly mapping X^p using a skip connection. The spectral residual architecture can be formulated as follows:

$$X^{p+2} = X^p + F(X^p;\theta) \tag{3.7}$$

$$F(X^p;\theta) = R(\hat{X}^{p+1}) * h^{p+2} + b^{p+2} \tag{3.8}$$

$$X = R(\hat{X}^p) * h^{p+1} + b^{p+1} \tag{3.9}$$

where $\theta = \{h^{p+1}, h^{p+2}, b^{p+1}, b^{p+2}\}$, X^{p+1} represents the n input 3D feature cuboids of $(p+1)$th layer, h^{p+1} and d^{p+1} denote the spectral convolution kernels and bias in the th layer, respectively. In fact, the convolution kernels h^{p+1} and d^{p+1} are composed of 1D vectors, which can be regard as a special case of 3D convolution kernels. The output tensor of the spectral residual block also includes n 3D feature cuboids.

In the spatial residual block, as illustrated in Figure 3.4, focus is primarily placed on the spatial feature extraction using n 3D convolution kernels of size $a \times a \times d$ in successive filter

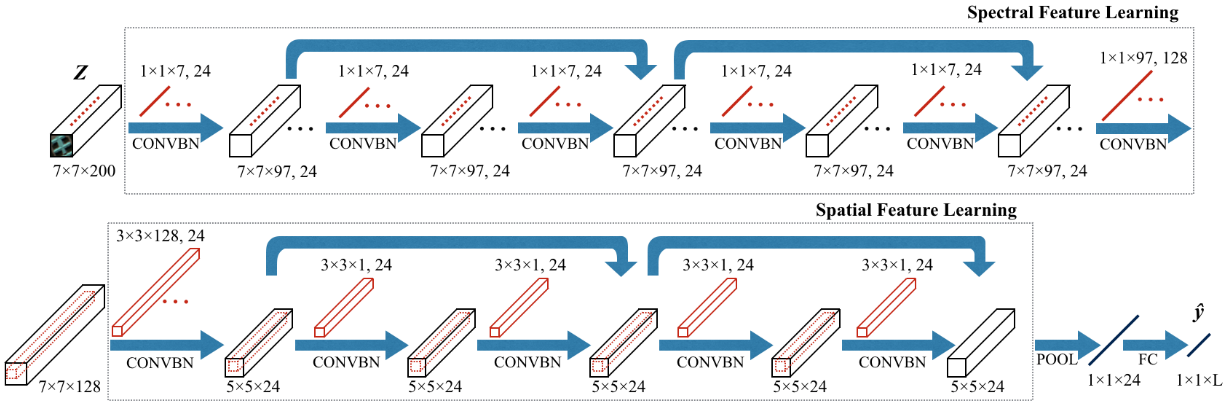


Figure 3.5: Spectral-Spatial Residual Network with a $7 \times 7 \times 200$ input HSI volume. The network includes two spectral and two spatial residual blocks. An average pooling layer and a fully connected layer transform a $5 \times 5 \times 24$ spectral-spatial feature volume into a $1 \times 1 \times L$ output feature vector $\hat{y}$.

banks H^{q+1} and H^{q+2} for the two successive layers. The spectral depth d of these kernels equals to that of the input 3D feature cuboids X^q. The spatial size of feature cuboids X^{q+1} and X^{q+2} is kept unchanged at $w \times w$. Thus, the spatial residual architecture can be formulated as follows:

$$X^{q+2} = X^q + F(X^q; \xi) \tag{3.10}$$

$$F(X^q; \xi) = R(\hat{X}^{q+1}) * H^{q+2} + b^{q+2} \tag{3.11}$$

$$X = R(\hat{X}^q) * H^{q+1} + b^{q+1} \tag{3.12}$$

where $\xi = \{H^{q+1}, H^{q+2}, b^{q+1}, b^{q+2}\}$, X^{q+1} represents the 3D input feature volume in the $(q+1)$th layer, H^{q+1} and b^{q+1} denote the n spatial convolution kernels in the $(q+1)$th layer, respectively. Compared with their spectral counterparts, the convolutional filter banks in spatial residual blocks comprises of 3D tensors. The output of this block is a 3D feature volume.

3.3.3 Spectral-Spatial Residual Network

Considering HSIs contain one spectral dimension and two spatial dimensions, a framework that consecutively extracts spectral and spatial features for pixel-wise HSI classification is proposed. As illustrated in Figure 3.5, the SSRN includes a spectral feature learning section, a spatial feature learning section, an average pooling layer, and a fully connected layer. Compared to CNN, SSRN alleviated the decreasing-accuracy phenomenon by adding skip connections between every other layer to formulate the hierarchical feature representation layers to consecutive residual blocks. The Indian Pines dataset, the 3D samples of which have the size of $7 \times 7 \times 200$, is taken as an example to explain the designed SSRN.

The spectral feature learning section includes two convolutional layers and two spectral residual blocks. In the first convolutional layer, 24 $1 \times 1 \times 7$ spectral convolution kernels with a subsampling stride of $(1, 1, 2)$ convolves the input HSI volume to generate 24 $7 \times 7 \times 97$ feature cuboids. Because the raw input data contains rich and redundant spectral information, $1 \times 1 \times 7$ vector convolution kernels are used in these blocks. This layer reduces the high dimensionality of input cuboids and extract low-level spectral features of HSI. Then, two consecutive spectral residual blocks, which contains four convolutional layers and two identity mappings, use 24 $1 \times 1 \times 7$ vector convolution kernels at each convolutional layers to learn deep spectral representation. In the spectral residual blocks, all convolutional layers use padding to keep the sizes of output feature cuboids the same as input. Following the spectral residual blocks, the last convolutional layer in this learning section, which includes 128 $1 \times 1 \times 97$ spectral convolution kernels for keeping discriminative spectral features, convolves the 24 7×7 feature tensors to produce a 7×7 feature volume as input for spatial feature learning section.

The spatial feature learning section extracts discriminative spatial features using successive 3D convolutional filter banks, where the convolution kernels have the same depth as the input 3D feature volume. The section comprises of a 3D convolutional layer and two spatial residual blocks. The first convolutional layer in this section reduce the spatial size of input feature cuboids and extract low level spatial features with 24 $3 \times 3 \times 128$ spatial convolution kernels, resulting an output $5 \times 5 \times 24$ feature tensor. Then, similar to their spectral counterparts, the two spatial residual blocks learn deep spatial representation with 4 convolutional layers, all of which use 24 $3 \times 3 \times 24$ spatial convolution kernels and keep the sizes of feature cuboids unchanged.

After the above two feature learning sections, an average pooling layer (POOL) transforms the extracted $5 \times 5 \times 24$ spectral-spatial feature volume to a $1 \times 1 \times 24$ feature vector. Next, a fully connected layer (FC) adapts the SSRN to HSI dataset according to the number of land cover categories and generates a output vector $\hat{y} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_L]$. The total numbers of trainable parameters (about 360,000) for the SSRN are much larger than the available training data in the three hyperspectral datasets, which means the network possesses enough capacity to learn the feature representations of HSI but also tend to overfit the training sets. Therefore, batch normalization and dropout [62] are investigated as regularization strategies to further improve the classification performance of SSRN.

3.4 Summary

In this chapter, two specific residual blocks are designed for HSI classification, and a SSRN that consists of two consecutive spectral and spatial learning blocks is proposed. The network configuration and experimental results are reported in Chapter 6. The SSRN adopts residual connections to mitigate the decreasing-accuracy phenomenon and improve the HSI classification accuracy. Two consecutive residual blocks learn spectral and spatial representations separately, through which more discriminative features can be extracted. Therefore, the SSRN shows the effectiveness of accounting for the characteristics of HSIs in order to boost HSI classification performance. Unfortunately, training supervised models requires enough annotations which are expensive in many remote sensing applications. To this end, the subsequent chapters focus on the cases that annotated HSI samples are relatively small, and design a novel semi-supervised framework that incorporates probabilistic graphical constraints to address this task.

Chapter 4

Semi-supervised models for adversarial training

“ Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech. ”

– Yann LeCun [39]

This chapter addresses the hyperspectral image (HSI) classification task with a generative adversarial network (GAN) -based framework, which is a semi-supervised deep learning model, and make two contributions. First, four types of convolutional and transposed convolutional layers are designed that consider the characteristics of HSIs to help with extracting discriminative features from limited numbers of labeled HSI samples. Second, semi-supervised GANs are constructed to alleviate the shortage of training samples by adding labels to them and implicitly reconstructing real HSI data distribution through adversarial training. This semi-supervised framework leverages the merits of discriminative and generative models through a game-theoretical approach.

4.1 Semi-supervised Models for HSI classification

Due to their hundreds of spectral bands, the accurate interpretation of hyperspectral images (HSIs) has attracted significant scholarly attention from the machine learning and remote sensing communities [30, 45, 83, 101]. Recent studies suggest that supervised deep learning models can alleviate challenges caused by the high spectral dimensionality of HSIs and achieve strikingly better classification accuracy [8,50,99]. However, there are still three challenges that prevent deep learning models from offering precise pixel-wise HSI classification maps [65, 82]. First, the high dimensionality of HSI pixels make it hard to directly use the deep learning models for normal optical images in HSI interpretation. Second, the shortage of labeled pixels limits the classification performance of deep learning models. Third, the classification maps generated by deep learning models tend to be noisy and have spurious object edges. In this chapter, these challenges are analyzed and offer several suggestions are offered to mitigate them.

In the face of these difficulties, two common semi-supervised learning methods — graph-based models and generative models — have been adopted to alleviate them [29,77,84,103]. Graph-based models are premised on the smoothness assumption that accentuates geometrically simple classification results. For example, [77] imposed a manifold regularizer on a Laplacian SVM framework to learn spectral-spatial features for HSI image classification. Additionally, [29] proposed a dual hypergraph framework that imposes spectral-spatial constraints by jointly calculating a Laplacian matrix. Although these graph-based semi-supervised methods take both labeled and unlabeled samples into account, they identify HSI pixels based on hand-crafted features. Generally, these features learned from feature engineering steps are difficult to tune or generalize to other cases. Moreover, the performance of these semi-supervised models largely depends on the quality of unlabeled data, which is hard to control or standardize. Recently, a generative model called genera-

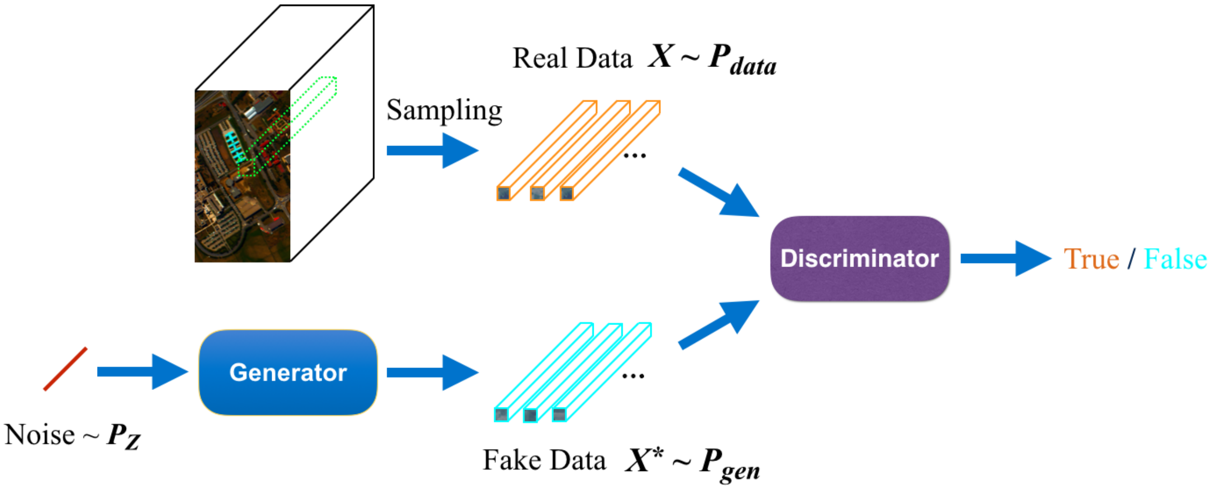


Figure 4.1: The architecture of a generative adversarial network, in which the generator transforms noise vectors to a set of hyperspectral cuboids and the discriminator tries to distinguish true hyperspectral samples from fake ones.

tive adversarial network (GAN) [22] has attracted a lot of attention for image generation. For instance, [84] proposed a semi-supervised 1D-GAN for HSI classification, but ignored the spatial attribute of HSIs that can be used for enhancing classification performance. Moreover, [103] used convolutional neural networks (CNNs) to build generative adversarial networks for HSI classification and achieved very promising results. However, the discriminators used in this thesis only use three principled component analysis (PCA) channels of HSIs and therefore do not fully exploit the spectral characteristic of HSIs.

4.2 Related Work

GANs are unsupervised deep learning models that provide a solution to implicitly estimate real data distribution and correspondingly generate synthetic samples. Recently, there has been increasing interest in GANs for unsupervised learning, especially in regards to generating synthetic images that approximate the distribution of real ones [22, 57]. Compared with traditional generative methods, GANs are not constrained by Markov fields or explicit approximation inference. For instance, a deep convolutional GAN [57] that consists of deep convolutional layers has been proposed to generate high-quality images. The original GAN aims for image generation and its variants have generated astonishing controllable and partially explainable images [59]. The GAN employs a discriminator and a generator to compete with each other [22,96]. Specifically, the generator generates synthetic examples to deceive the discriminator, and the discriminator distinguishes real samples from fake ones. Since their objectives are contradictory, the training of the discriminator and generator of a GAN can be regarded as a process to find a Nash equilibrium through a game-theoretical point of view. Therefore, this GAN training can be formulated as a min-max optimization problem:

$$\min_G \max_D Loss(D, G) = E_{x \sim p_{data}}[\log D(x)] + E_{z \sim p_z}[\log (1 - D(G(z)))], \tag{4.1}$$

where $D(\cdot)$ and $G(\cdot)$ represent softmax outputs of a discriminator and synthetic data generated by a generator, respectively. x and z denote true images and vectors of Gaussian noise, and they follow the distributions of real HSI data and Gaussian noise, respectively. GANs produce very promising image generation results in datasets like the MNIST digit database [40] and the Yale Face database [76], both of which contain compact data distribution and similar image layout.

4.3 Proposed Model

To solve the challenges of HSI classification, a GAN-based semi-supervised deep learning framework is proposed. Suppose a hyperspectral image X contains m pixels $\{x_i\} \in \mathbb{R}^{n_x \times m}$, where n_x represents the number of spectral bands. Then, two groups of HSI cuboids are sampled from X: the labeled group $X^1 = \{X_i^1\} \in \mathbb{R}^{n_x \times w \times w \times m_l}$ and the unlabeled group $X^2 = \{X_i^2\} \in \mathbb{R}^{n_x \times w \times w \times m_u}$, where w, m_l, and m_u are the spatial width of HSI cuboids, the number of labeled, and the number of unlabeled HSI samples, respectively. Since each pixel in X corresponds to a HSI cuboid in $\{X_i^1, X_i^2\}$, therefore $m = m_l + m_u$. The labeled group X^1 has its annotation $Y^1 = \{y_i^1\} \in \mathbb{R}^{(1+n_y) \times m_l}$, where n_y is the number of land cover classes and $y_i^1[0]$ (the first entry in a vector y_i^1) indicates whether the corresponding HSI cuboid is fake (1/0 means fake/real). As shown in Figure 3.1, the whole model is composed of a discriminator, a generator, and a post-processing CRF. Since annotations Y^1 of real HSI samples are used for training, the discriminator and generator form a semi-supervised GAN. The generator transforms noise vectors z to synthetic HSI cuboids $Z = \{Z_i\}$, each sample of which have the same size as those from X^2. The discriminator attempts to distinguish real HSI cuboids X^1 from fake ones Z and to classify real HSI cuboids.

In contrast to updating one discriminative model in supervised deep learning, the train-

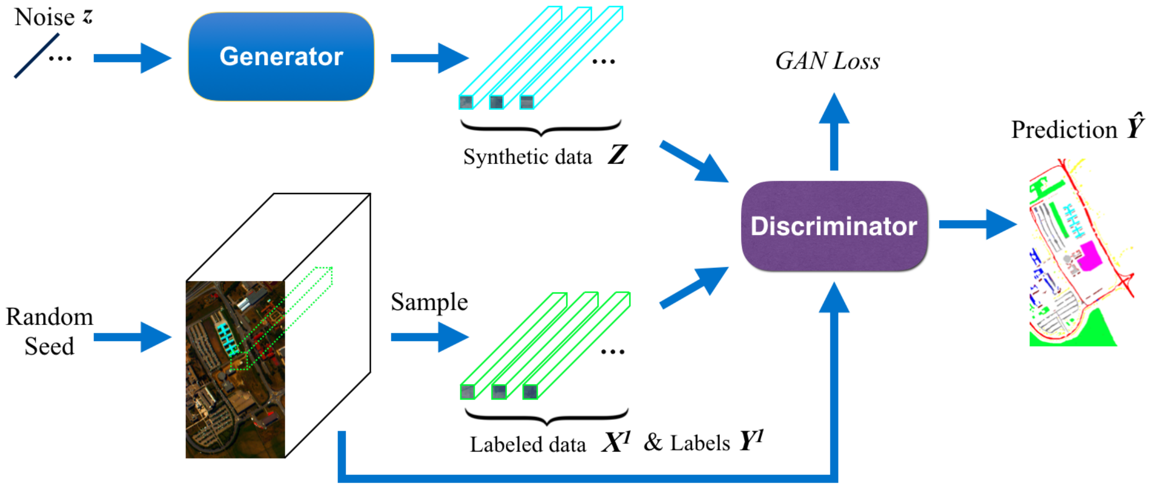


Figure 4.2: A semi-supervised GAN framework for HSI classification. First, in the semi-supervised GAN, a generator transforms noise vectors z to a set of fake HSI cuboids Z, and a discriminator tries to distinguish the categorical information as well as the genuineness of input cuboids that come from X^1 or Z. The HSI prediction $\hat{Y}$ is generated by the trained discriminator of the spectral-spatial GAN.

ing of a GAN involves searching an equilibrium between the generator and discriminator by using stochastic gradient descent or similar methods to optimize the parameters of the GAN. However, GANs are known for their instability in training, and it is almost impossible to find an optimal equilibrium between their generators and discriminators. Therefore, an alternating optimization strategy is adopted that successively updates the parameters of the generator and discriminator in each training iteration to help the discriminator to learn discriminative features using a small amount of labeled data and a large amount of synthetic data produced by the generator. When the training of a GAN is completed, the trained discriminator of the GAN is used to make a prediction about the unlabeled group X^2. Then, a conditional random field is established by using the softmax predictions of the trained discriminator to initialize random variables $Y = \{y_i\} \in \mathbb{R}^{(1+n_y)\times m}$ that are conditioned on the raw HSI X. Last, mean field approximation is used to optimize the conditional random field and get a refined classification map $\hat{Y}$.

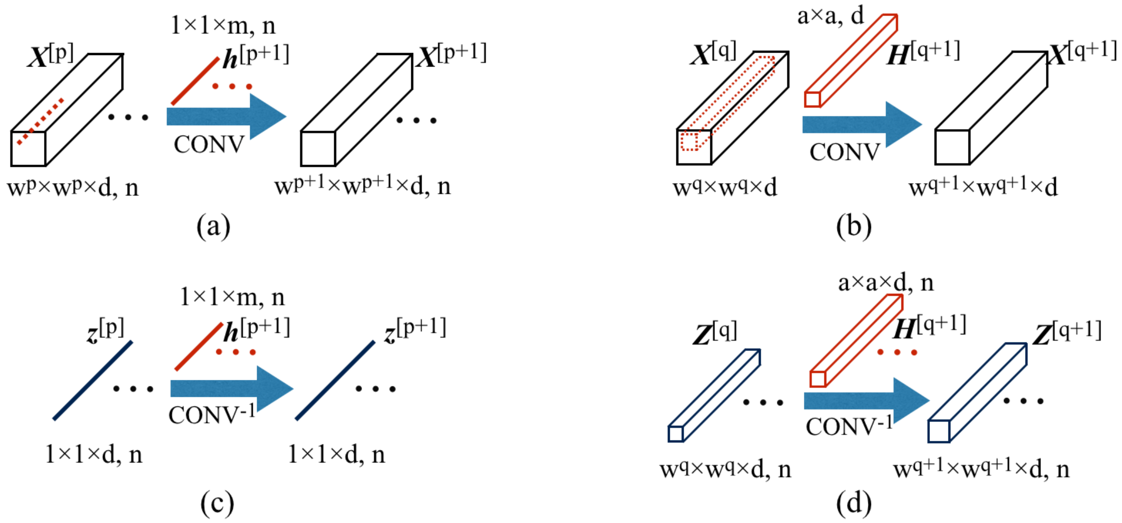


Figure 4.3: Four basic convolutional and transposed convolutional layers aiming for hyperspectral features extraction and generation in semi-supervised GAN-CRF models. (a) - (b) Spectral and spatial convolutional layers in discriminators. (c) - (d) spectral and spatial transposed convolutional layers in generators.

4.3.1 Spectral-Spatial Discriminator and Generator

Discriminative deep learning models, such as CNNs and their extensions, have been used for HSI feature extraction and they have substantially outperformed traditional machine learning methods given enough training data [8, 99]. However, both these approaches ignore the inherent difference in spectral dimensionality between hyperspectral images and common images used in computer vision tasks. Based on the assumption that the sampled HSI data form a low dimensional manifold embedded in a higher dimensional space, multiple models have tried to reduce the high dimensionality of HSI pixels and to learn more efficient representation [86, 91]. However, the dimension reduction process inevitably leads to the loss of useful information.

The specialty of HSI samples lies in its high spectral dimensionality. Recently, in response to this characteristic, [99] implemented a spectral-spatial residual network (SSRN) that considers the characteristics of HSI by consecutively extracting spectral and spatial features and obtained state-of-the-art supervised classification results. Therefore, as illustrated in Figure 3.2 (a)-(b), the idea of spectral and spatial convolution from [99] is extended to the discriminator of a GAN-CRF model. If $X^{[p+1]}$ and $X^{[q+1]}$ represent the feature tensors of $[p+1]$th spectral and $[q+1]$th spatial convolutional layers, then the spectral and spatial convolutional layers of a discriminator can be formulated as follows:

$$X^{[p+1]} = LR(w^{[p+1]} * X^{[p]} + b^{[p+1]}), \tag{4.2}$$

$$X^{[q+1]} = LR(W^{[q+1]} * X^{[q]} + b^{[q+1]}), \tag{4.3}$$

where $w^{[p+1]}$ and $W^{[q+1]}$ represent the $[p+1]$th spectral and $[q+1]$th spatial convolution kernels, respectively. $b^{[p+1]}$ and $b^{[q+1]}$ are the biases of these two layers. $*$ denotes the convolutional operation. $LR(\cdot)$ is a leaky rectified linear unit function:

$$LR(a) = \begin{cases} a, & \text{if a} > 0, \\ 0.2a, & \text{otherwise.} \end{cases} \tag{4.4}$$

In this work, padding tricks is used to keep the spatial size of feature tensors in most convolutional layers unchanged. The goal of adopting spectral-spatial convolutional layers in a GAN-CRF model is to exploit as much information as possible from limited labeled HSI samples. Similarly, the spectral-spatial idea is stretched to transposed convolutional layers. As shown in Figure 3.2 (c)-(d), the spectral and spatial transposed convolutional layers of a generator can be formulated as follows:

$$z^{[p+1]} = R(h^{[p+1]} *^T z^{[p]} + b^{[p+1]}), \tag{4.5}$$

$$Z^{[q+1]} = R(H^{[q+1]} *^T Z^{[q]} + b^{[q+1]}), \tag{4.6}$$

where $h^{[p+1]}$ and $H^{[q+1]}$ represent the $[p+1]$th transposed spectral and $[q+1]$th transposed spatial convolution kernels. $b^{[p+1]}$ and $b^{[q+1]}$ are the biases of these two layers. $*^T$ denotes the transposed convolutional operation. $R(\cdot)$ is the rectified linear unit function:

$$R(a) = \begin{cases} a, & \text{if a} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

As shown in Figure 3.2, in contrast to spatial convolutional layers, the transposed convolutional layers expand the spatial size of feature tensors. In both the discriminator and generator of a GAN-CRF model, batch normalization [28] is applied in all convolutional and transposed convolutional layers to stabilize the training of a GAN.

4.3.2 Semi-supervised GAN

A GAN can be regarded as a combination of discriminative and generative models, where the discriminator focuses on learning discriminative features, and the generator concentrates on implicitly reconstructing real data distribution from random noises. As an example of University of Pavia (UP) dataset shown in Figure 4.1, the discriminator comprises three spectral convolutional layers, three spatial convolutional layers, and a fully connected layer before a vector of softmax outputs. Conversely, the generator consists of a fully connected layer, three transposed spectral convolutional layers, and four spatial transposed convolutional layers to produce a synthetic hyperspectral cuboid.

As the generator of a GAN can produce reasonable synthetic images and utilize them to train the discriminator of the GAN, many research papers have extended the discriminator of GANs to semi-supervised classification [11,59,84]. Similarly, the GAN is generalized to the semi-supervised HSI classification task. Since the labeled hyperspectral cuboid group $X^1 = \{X_i^1\}$ has its corresponding annotation group $Y^1 = \{y_i^1\}$, the prediction of trained discriminators take this form:

$$\hat{Y}^1 = D(X^1; \theta_D), \tag{4.8}$$

each element $\hat{y}_i^1$ of which has $(1 + n_y)$ entries. Specifically, $\hat{y}_i^1[0]$ indicates the genuineness of a hyperspectral cuboid, and $\hat{y}_i^1[1 : n_y]$ is a vector of softmax outputs that shows the probabilities of a hyperspectral cuboid belonging to the n_y land cover classes. Compared to the original GAN that discriminates real data from fake ones, a semi-supervised GAN recognizes the categorical information of HSI cuboids by adding a supervised term to the loss function of a GAN.

It is worth noting that the objectives of an unsupervised GAN and a semi-supervised GAN are different and even partially contradictory. The unsupervised GAN aims for implicitly estimating the true data distribution. On the contrary, the semi-supervised GAN focuses on data generation using limited labeled samples. Therefore, training a semi-supervised GAN jeopardize its image generation capability. As presented in [11], a good semi-supervised GAN requires a bad generator because this generator produces data outside real data distribution, which in turn helps the discriminator recognizes real data more accurately. In this way, the generator that produces synthetic HSI cuboids functions as a regularizer on the discriminator. Therefore, the loss function regarding optimize the discriminator of a GAN for semi-supervised HSI classification takes the form:

$$L_{SEMI}(\theta_D, \theta_G) = L_{SUP}(\theta_D) + L_{D1}(\theta_D) + L_{D2}(\theta_D, \theta_G), \tag{4.9}$$

where θ_D and θ_G are the parameters of a discriminator and a generator, respectively. L_{SEMI} is the total semi-supervised loss for training the discriminator of a semi-supervised GAN, L_{SUP}, L_{D1}, and L_{D2} represent the supervised loss of a discriminator, the unsupervised loss of a discriminator, and the unsupervised loss of a generator, respectively. These three terms are formulated as follows:

$$\begin{aligned} L_{SUP}(\theta_D) &= -E_{X^1 \sim p_{data}} \log D(X^1; \theta_D)[1:n_y] \\ &= -E_{X^1 \sim p_{data}} \log \hat{Y}^1[1:n_y], \end{aligned} \tag{4.10}$$

$$\begin{aligned} L_{D1}(\theta_D) &= -E_{X^1 \sim p_{data}} \log(1 - D(X^1; \theta_D)[0]) \\ &= -E_{X^1 \sim p_{data}} \log(1 - \hat{Y}^1[0]), \end{aligned} \tag{4.11}$$

$$\begin{aligned} L_{D2}(\theta_D, \theta_G) &= -E_{z \sim p_z} \log D(G(z; \theta_G); \theta_D)[0] \\ &= -E_{z \sim p_z} \log D(Z; \theta_D)[0] \\ &= -E_{z \sim p_z} \log \hat{Y}^1[0]. \end{aligned} \tag{4.12}$$

It is worth mentioning that $L_{D1} + L_{D2}$ also is the part of the total semi-supervised loss L_{SEMI} that aims at training the bad generator of a GAN [11]. Correspondingly, the loss function for training the generator of a semi-supervised GAN takes this form:

$$\begin{aligned} L_G(\theta_D, \theta_G) &= -E_{z \sim p_z} \log\left(1 - D(G(z; \theta_G); \theta_D)[0]\right) \\ &= -E_{z \sim p_z} \log\left(1 - D(Z; \theta_D)[0]\right) \\ &= -E_{z \sim p_z} \log(1 - \hat{Y}^1[0]). \end{aligned} \tag{4.13}$$

The training of a semi-supervised GAN involves two alternating steps of stochastic gradient descent (SGD) or similar optimization methods in each iteration. First, the gradients of a discriminator $-\nabla_{\theta_D} L_{SEMI}$ are used to update the parameters θ_D of a discriminator for learning discriminative spectral-spatial HSI features. Second, the gradients of generators $-\nabla_{\theta_D} L_G$ are employed to update the parameters θ_G of a generator for improving the adversarial training of the semi-supervised GAN.

4.4 Summary

This chapter introduces two major challenges of HSI classification and reviews related works with regard to classifying HSI in a semi-supervised manner. Also, the semi-supervised GANs include novel spectral or spatial layers, spectral-spatial discriminators and generators. Section 6.3 offers corresponding model settings, comparative experiments, discussions, and conclusions. By taking the characteristics of training data into account, the discriminators of SS-GANs extract discriminative HSI features and achieve higher classification accuracy. Generators of SS-GANs learn feature representation by producing synthetic HSI samples, and in turn make discriminators more robust to adversaries and learn more discriminative features. Therefore, this adversarial training enables semi-supervised GANs to deliver superior classification outcomes to supervised deep learning models. Additionally, adding large numbers of unlabeled real samples to train semi-supervised GANs marginally improves or even jeopardizes the HSI classification accuracy. Given small numbers of labeled samples, the HSI classification results of SS-GAN suffer from noise in homogeneous areas and therefore further prior constraints are needed for improving semi-supervised HSI classification performance.

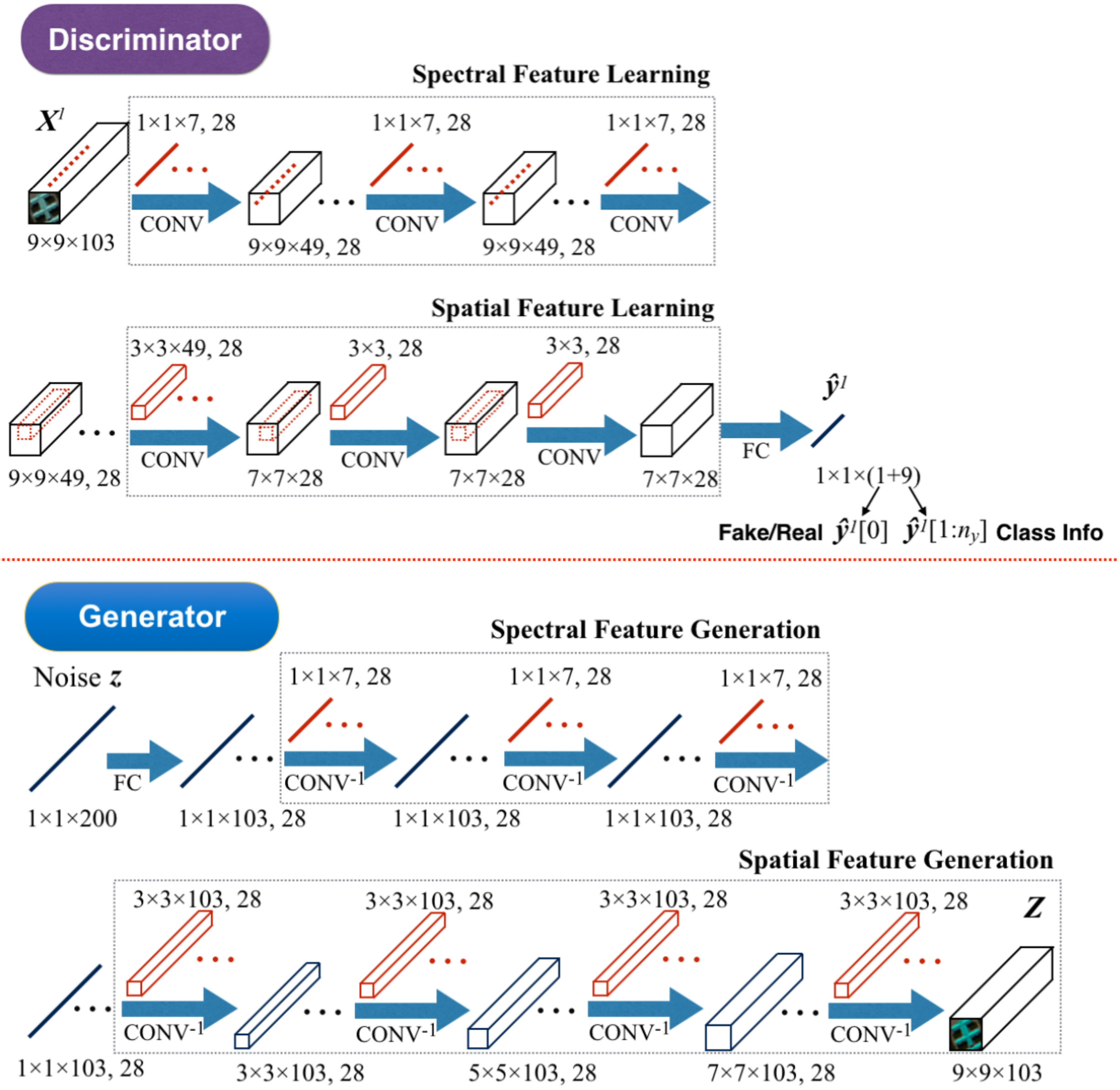


Figure 4.4: A spectral-spatial discriminator (upper), which comprises consecutive spectral and spatial feature learning blocks, outputs a vector that contains a indicative entry of fake or real and categorical probabilities; and a spectral-spatial generator (lower), which comprises consecutive spectral and spatial feature generation blocks, transforms a vector of Gaussian noise to a synthetic HSI cuboid.

Chapter 5

Probabilistic graph models for post-processing

" The framework of probabilistic graphical models provides a mechanism for exploiting structure in complex distributions to describe them compactly, and in a way that allows them to be constructed and utilized effectively. Probabilistic graphical models use a graph-based representation as the basis for compactly encoding a complex distribution over a high-dimensional space."

– Daphne Koller [35]

In this chapter, dense conditional random fields (CRFs) are random variables initialized to the softmax predictions of the trained SS-GANs and conditioned on HSIs to refine classification maps. Then, this framework integrates the semi-supervised deep learning models and probabilistic graph models. Even though very small numbers of labeled training are used HSI samples from the two most challenging and extensively studied datasets (Indian Pines and University of Pavia), the experimental results in Chapter 6 demonstrated that spectral-spatial GAN-CRF (SS-GAN-CRF) models, which adopt dense CRFs as a post-processing step, achieved state-of-the-art accuracy for semi-supervised HSI classification.

5.1 Background

Due to the complexity of HSIs, multiple works utilize the smoothness assumption that favors geometrically simple classification results [6, 14, 16, 42, 44, 66, 77, 80, 95]. For example, [66] incorporated a probabilistic graphical model as the post-processing step to improve the classification outcomes of kernel support vector machines (SVMs). [94] constructed a conditional random field (CRF) with a high-order term to consider more complex relationships between different spectral bands and obtained very promising outcomes. Additionally, [95] incorporated a CRF for pre-processing as well as post-processing to stress the a priori smoothness and refine the classification maps. The integration of probabilistic graphical models and supervised classification models can also be conceived as a way to take the unlabeled samples into account for HSI classification because this step does not require the ground truth annotation of neighboring pixels. However, most CRF based models consider only the short-range correlations of pixels and ignore the long-range ones.

In this chapter, inspired by [22] and [7], a semi-supervised deep learning framework is suggested that consists of a generator, discriminator, and conditional random field built on top of the discriminator. The discriminator and generator form a generative adversarial network based on game theory. Specifically, the discriminator adopts spectral-spatial convolutional layers to learn discriminative features from a small amount of labeled data and unlabeled data, and the generator employs spectral-spatial transposed convolutional layers to reconstruct HSI samples from vectors of Gaussian noise. Unlike traditional semi-supervised models, which require a large amount of unlabeled data for training, the proposed framework is data-efficient because the generator creates a high amount of synthetic data and the discriminator takes a small number of unlabeled samples. In this way, the GAN-CRF model estimates the real data distribution, mitigates the shortage of annotated data, and smooths the semi-supervised learning process. In addition, the output of the discriminator is the unary input term of the subsequent CRF. The binary term of the CRF

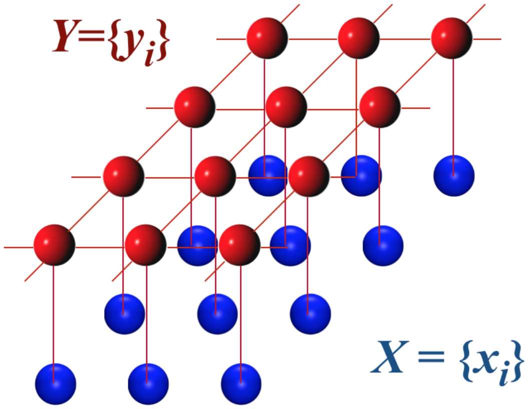


Figure 5.1: A conditional random field is established on two layers. The lower layer X represent the observed hyperspectral pixels, and the upper layer represents the output random variables Y.

imposes an a priori smoothness whereby adjacent pixels are more likely to belong to the same categories. More importantly, the CRF takes on a fully connected form that imposes a random field on the whole classification map and considers the long-range relationship between HSI pixels. Thus, by taking a generative adversarial network and considering the continuity of neighboring pixels, the designed semi-supervised architectures learn local fine-grain representation as well as high-level invariant features of HSI pixels concurrently.

5.2 Conditional Random Field

Graph models have widely been used for remotely sensed image interpretation tasks to effectively impose smoothness constraints on classification or segmentation results [7, 89]. CRFs are graphical models that assume a priori continuity whereby neighboring pixels of similar spectral signatures tend to have the same labels [94]. Since CRFs can be regarded as a structured generalization of multinomial logistic regression, the conditional probability

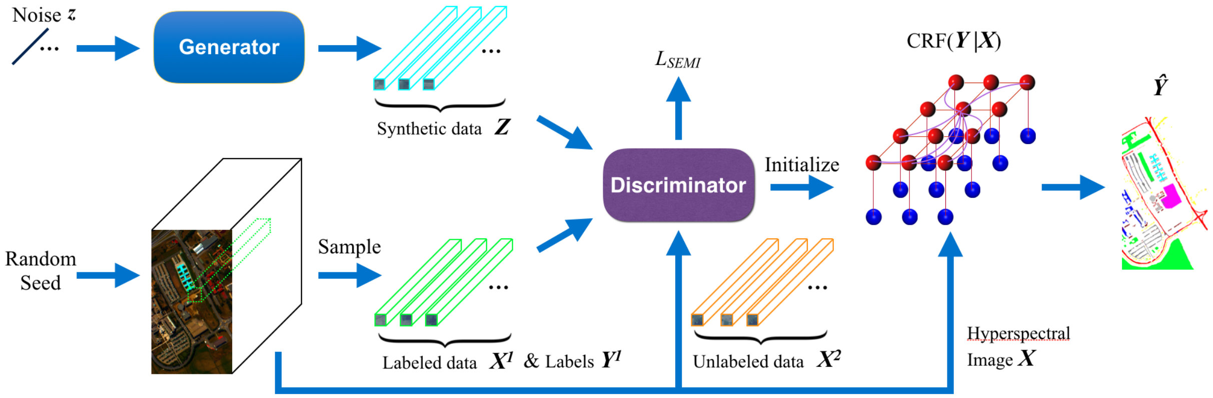


Figure 5.2: A semi-supervised GAN-CRF framework for HSI classification. First, in the semi-supervised GAN, a generator transforms noise vectors z to a set of fake HSI cuboids Z, and a discriminator tries to distinguish the categorical information as well as the genuineness of input cuboids that come from X^1 or Z. Then, a dense CRF is established by using the softmax prediction of the trained discriminator about X^2 to initialize random variables Y, which is conditioned on the HSI data X. Mean field approximation is adopted to offer a refined classification map $\hat{Y}$ for the post-processing CRF.

distribution of a CRF takes this form:

$$Prob(y|X) = \frac{\exp(-E(y|X))}{\sum_y \exp(-E(y|X))}, \tag{5.1}$$

where y and x denote output random variables and their corresponding observed data. $E(\cdot)$ is an energy function that models the joint probability distribution of y and x. The optimal random variables can be calculated by the *maximum a posteriori* (MAP) estimation:

$$y^{MAP} = \underset{y}{argmax}\, Prob(y|X). \tag{5.2}$$

However, although Equation (5.2) usually is an intractable problem, it can be solved through approximation methods [36].

5.3 GAN-CRF model

CRFs have been widely used to post-process image segmentation results because they can exploit the predictions of large numbers of unlabeled pixels to enhance image interpretation performance [6, 92]. Once a semi-supervised GAN has been built, a conditional random field is established by using the softmax predictions of the trained semi-supervised GAN about unlabeled HSI cuboids to initialize random variables $Y = \{y\}$ that are conditioned on observed raw HSI pixels X. According to Equation (5.1), the conditional probability distribution of this CRF takes the form:

$$Prob(y|X) = \frac{\exp(-E(y|X))}{\sum_y \exp(-E(y|X))}. \tag{5.3}$$

As illustrated in Figure 1, given that high correlations exists between HSI pixels $\{x_i\}$ in both short- and long-range, a dense CRF [7] is adopted that includes all pairwise connections between HSI pixels in the pairwise term of energy function to filter salt and pepper noises in homogeneous areas. The energy function of the dense CRF can be formulated as:

$$E(y|X) = \sum_i E(y_i|X) = \sum_i U(y_i|X) + \sum_{ij} P(y_{ij}|X), \tag{5.4}$$

where $U(\cdot)$ and $P(\cdot)$ are the unary and pairwise terms of the energy function that is used to build the dense CRF. y_{set} denotes the set of samples that connect to y. Specifically, the unary term represents the information cost of pixel-wise softmax predictions $\{y_i\}$ and the binary term penalizes the wrong labeling of pixel pairs $\{x_i, x_j\}$ with similar spectral signatures. These two terms are formulated as follows:

$$U(y_i|X_i) = D(X_i; \theta_D), \tag{5.5}$$

$$\begin{aligned} P(y_{ij}|X) &= P(y_i, y_j, x_i, x_j) \\ &= \mu(y_i, y_j) K(x_i, x_j, l_i, l_j), \end{aligned} \tag{5.6}$$

where l_i and l_j denote the locations of x_i and x_j, respectively. $\mu(\cdot)$ is a compatibility function, and $K(\cdot)$ is a bilateral Gaussian kernel function. These two functions take the forms:

$$\mu(y_i, y_j) = \begin{cases} c, & \text{if } \eta(y_i) \neq \eta(y_j) \\ 0, & \text{otherwise} \end{cases} \tag{5.7}$$

$$K(x_i, x_j, l_i, l_j) = \exp(-\frac{(l_i - l_j)^2}{2\theta_\alpha^2} - \frac{(x_i - x_j)^2}{2\theta_\beta^2}), \tag{5.8}$$

where $\eta(\cdot)$ denotes a one-hot function. θ_α and θ_β are two standard deviations of the bilateral Gaussian kernels. c is a constant value that could be manually set. Random variables $Y = \{y_i\}$ of the established dense CRF is initialized to the softmax predictions of the trained discriminators $D(X^2; \theta_D)$ of the semi-supervised GAN according to Equation (4.8).

In a GAN-CRF model, a GAN is utilized to produce softax predictions about unlabeled HSI samples X^2, and the post-processing CRF is independent of the GAN. Specifically, the predictions about a large numbers of unlabeled samples are used to initialize the unary term of the energy function that builds a dense CRF, and therefore the GAN-CRF model is more suitable in the case where only limited labeled samples are available. Because the energy function in Equation (5.4) is an intractable problem, a function $Q(Y|X)$ adopted to approximate the conditional probability distribution $Prob(Y|X)$ of the CRF takes the form:

$$Q(Y|X) = \prod_i Q(y_i|X) \approx Prob(Y|X), \tag{5.9}$$

in which the tractable function $Q(Y|X)$ is close to $Prob(Y|X)$ in terms of KL-distribution divergence. Then, the mean field approximation [36] is used to find an optimal solution of random variables $\hat{Y}$ for the established dense CRF.

5.4 CRF layers for post-processing

Considering that one of the major benefits of deep learning model comes from the end-to-end learning, it is intuitive to incorporate graph constraints into deep learning models. Figure 5.2 shows the architecture of the proposed end-to-end semi-supervised deep learning model. The CRF layers are adopted to approximate fully connected conditional random fields. The CRF layers are convolutional layers corresponding to a mean field approximation, a weighted Gaussian filtering, and a compatibility transformation. The CRF layers are initialized and updated according to intermediate multi-class logistic output and raw input data.

Inspired by [92], this end-to-end model architecture is proposed to combines feature extraction, semi-supervised learning, and graph constraints into a holistic model instead of two independent parts in Section 2.2. The reformulation of CRF as a mean filed approximation can be formulated as the following steps.

First step is message passing that focuses on updating pixel-wise annotation according to neighbouring information:

$$Q_i^{(m)}(l) = \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l), \tag{5.10}$$

Second step is Gaussian filter weighting that decides the contribution of each kernel:

$$\hat{Q}_i(l) = \sum_{m} w^{(m)} Q_i^{(m)}(l), \tag{5.11}$$

Third step is compatibility transformation and adding up the unary term to update the intermediate output of GANs:

$$Q_i^{*}(l) = \sum_{l' \in \mathcal{L}} \mu(l, l') \hat{Q}_i(l) + U_i(l), \tag{5.12}$$

where i denotes the number of iteration. According to these three Equations (3.14)-(3.16), all steps could be implemented using normal layers in deep learning frameworks. Specifically, the CRF layers will be implemented in a recurrent format that that solves a mean field approximation problem and utilize the training processing of deep learning.

Compared with those CRFs adopted in previous articles [90, 93], we adopt the fully connected CRFs which consider the long-range correlations between HSI samples. This property helps GAN-CRF models to better filter noises in the homogeneous areas of some land cover classes. Compared to just a supervised discriminator, a GAN-CRF model integrates the advantages of deep learning models and probabilistic graph models and improves HSI classification accuracy. There are two main reasons for this improvement: 1) the synthetic HSI samples produced by generators help discriminators to learn more robust and discriminative features; 2) the subsequent dense CRFs consider the spectral similarity and spatial closeness of HSI samples to refine the softmax outputs conditional on these samples using the trained discriminators of GANs.

5.5 Summary

In this chapter, dense conditional random fields that impose graph constraints are built on the softmax predictions of trained discriminators to refine HSI classification maps. The GAN-CRF [97] models incorporate the CRF as a post-processing step and build a graph upon the learned features and the softmax outputs of discriminators to refine HSI classification maps. Specifically, the dense CRFs take the classification maps generated by semi-supervised GANs as an initialization and smooth the noisy classification maps by adding a pairwise term that imposes the correlation between similar or neighboring pixels from input HSIs.

There are three differences between the GAN-CRF framework and the original GAN proposed in [22]. First, GAN-CRF models take the spectral-spatial characteristics of HSI data into account for both the discriminators and generators. Second, the discriminators in the semi-supervised framework extend the softmax predictions $\hat{y}$ of a GAN from two classes (fake/real) to $1 + n_y$ classes, where n_y represents the number of land cover classes. Third, a post-processing dense CRF has been built on conditional random variables that are initialized to the softmax outputs of the trained GANs to filter salt and pepper noises in homogenous areas.

Chapter 6

Experimental Results

“ Deep learning algorithms seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower-level features. The objective is to make these higher level representations more abstract, with their individual features more invariant to most of the variations that are typically present in the training distribution, while collectively preserving as much as possible of the information in the input. Ideally, these representations are employed to disentangle the unknown factors of variation that underlie the training distribution. ”

– Yoshua Bengio [2]

In this chapter, the three HSI datasets, specified the model configuring process, and evaluated the proposed methods are introduced, along with classification metrics like overall accuracy (OA), average accuracy (AA), and kappa coefficient (κ). For supervised classification experiments, the Indian Pines (IN), Kennedy Space Centre (KSC), and University of Pavia (UP) datasets are used for assessing the classification performance of the SSRN framework in the cases of unbalanced training data, a small number of training samples and high spatial resolution. In all three cases, experiments are ran for 10 times with randomly selected training data and reported the mean and standard deviation of main classification metrics. For semi-supervised experiments, two challenging HSI datasets (IN and UP datasets) are used, hyper-parameters of semi-supervised GANs are selected, and GAN-CRF models are evaluated. Additionally, training and testing times are recorded of all semi-supervised GANs to quantitatively assess their computational complexity.

6.1 Hyperspectral Image Datasets

Three widely studied datasets are adopted to evaluate the effective and generality of the SSRNs. In the following paragraph, each hyperspectral image dataset is introduced consecutively.

IN dataset: The IN dataset, gathered by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) in 1992 from northwest Indiana, includes 16 vegetation classes and has 145×145 pixels with a resolution of 20 m by pixel. Once the 20 bands corrupted by water absorption effects have been discarded, the remaining 200 bands are adopted for analysis and range from 400 nm to 2500 nm.

KSC dataset: The KSC dataset, collected by AVIRIS in Florida in 1996, contains 13 upland and wetland classes and has 512×614 pixels with a resolution of 18 m by pixel. Once the bands with low signal to noise ratio have been removed, the remaining 176 bands are used for assessment and range from 400 to 2500 nm.

UP dataset:, acquired by Reflective Optics System Imaging Spectrometer (ROSIS-3) in northern Italy in 2001, contains 9 urban land cover types and has 610×340 pixels with a resolution of 1.3 m by pixel. Once the noisy bands have been discarded, the remaining 103 bands are employed for evaluation and ranges from 430 nm to 860 nm.

In the IN and KSC datasets, 20%, 10%, and 70% of the labelled data are randomly assigned to training, validation, and testing groups, respectively. In the UP datasets, the ratio is 10%:10%:80%. In addition, all input data of three HSI datasets are standardized

Table 6.1: Training, Validation and Testing Numbers in IN Dataset

No.	Class	Train.	Val.	Test.
1	Alfalfa	10	1	35
2	Corn-notill	286	131	1011
3	Corn-mintill	166	83	581
4	Corn	48	22	167
5	Grass-pasture	97	42	344
6	Grass-tree	146	69	515
7	Grass-pasture-mowed	6	3	19
8	Hay-windrowed	96	55	327
9	Oats	4	4	12
10	Soybean-notill	195	94	683
11	Soybean-mintill	491	264	1700
12	Soybean-clean	119	56	418
13	Wheat	41	26	138
14	Woods	253	136	876
15	Buildings-Grass-Trees	78	34	274
16	Stone-Steel-Towers	19	5	69
	TOTAL	2055	1025	7169

to mean value with unit variance. Tables 6.1, 6.2, and 6.3 list the training, validation, and testing sample numbers of three datasets, respectively.

6.2 Supervised Classification Using SSRNs

After designing the SSRN framework, the training process that updates the parameters of 3D filter banks are configured through back propagating the gradients of the cost function.

Table 6.2: Training, Validation and Testing Numbers in KSC Dataset

No.	Class	Train.	Val.	Test.
1	Scrub	153	78	530
2	Willow swamp	49	29	165
3	CP hammock	52	28	176
4	Slash pine	51	31	170
5	Oak/Broadleaf	33	18	110
6	Hardwood	46	22	161
7	Swap	21	4	80
8	Graminoid marsh	87	45	299
9	Spartina marsh	104	39	377
10	Cattail marsh	81	40	283
11	Salt marsh	84	39	296
12	Mud flats	101	61	341
13	Water	186	87	654
	TOTAL	1048	521	3642

Next, four factors that control the training process and classification performance of the trained SSRN are analyzed. The four factors are the learning rate, the kernel number of convolutional layers, the regularization method, and the spatial size of the input cuboids. Due the training sets are small, the batch size is set to 16 and adopted the RMSProp optimizer [67] to harness the training process. In the training process of each configuration, the models with the highest classification performance in validation groups were preserved, and the reported results were generated by these optimal models.

First, learning rates control the learning step for each training iteration. Specifically, inappropriate learning rate settings will lead to divergence or slow convergence. Therefore, the grid search is used and each experiment is ran for 200 epochs to find the optimum learning rate from $\{0.01, 0.003, 0.001, 0.0003, 0.0001, 0.00003\}$ for each dataset. Based on the classification outcomes, the optimum learning rates for IN, KSC, and UP datasets are

Table 6.3: Training, Validation and Testing Numbers in UP Dataset

No.	Class	Train.	Val.	Test.
1	Asphalt	664	670	5297
2	Meadows	1865	1810	14974
3	Gravel	210	241	1648
4	Trees	307	333	2424
5	Metal Sheets	135	134	1076
6	Bare Soil	503	500	4026
7	Bitumen	133	133	1046
8	Bricks	369	363	2950
9	Shadows	95	97	755
	TOTAL	4281	4281	34214

0.0003, 0.0001, and 0.0003, respectively.

Second, the kernel numbers of convolutional filter banks decide the representation capacity and computational consumption of SSRN. As shown in Figure 3.5, the proposed network has the same kernel number in each convolutional layer of the spectral and spatial residual blocks. Different kernel numbers from 8 to 32 in an interval of 8 are assessed in each convolutional layer to find a general framework. As shown in Figure 6.1, the models with 24 convolution kernels in each convolutional filter bank achieved the highest classification accuracy in IN and UP datasets, and the model with 16 convolution kernels obtained the best performance in KSC dataset. These results are acquired in 200-epoch training processes for each setting in three datasets.

Third, given there are more parameters than training samples and deep learning models tend to overfit training data, batch normalization and a 50% dropout can be used for regularizing training process. Hence, the models are evaluated without regularization method, with dropout, with batch normalization (BN), and with both dropout and BN under the same condition for 200-epoch training. As shown in Table 6.4, the BN outperforms the dropout in term of mean overall classification accuracy. More importantly, the SSRN performs the best when using both regularization strategies in all three HSI datasets.

Table 6.4: Overall Accuracy (%) of SSRN with Different Regularizers

SSRN	IN	KSC	UP
None	96.41 ± 0.51	97.75 ± 0.54	98.97 ± 0.17
Dropout	95.83 ± 0.52	96.37 ± 0.89	99.02 ± 0.19
BN	97.73 ± 0.42	98.96 ± 0.23	99.42 ± 0.13
Both	97.76 ± 0.38	99.02 ± 0.31	99.59 ± 0.08

Table 6.5: Overall Accuracy (%) of SSRN with Different Input Sizes

Spatial Size	IN	KSC	UP
3×3	75.83 ± 0.14	92.38 ± 0.99	96.81 ± 0.24
5×5	92.83 ± 0.66	96.99 ± 0.55	98.72 ± 0.17
7×7	97.81 ± 0.34	99.01 ± 0.31	99.54 ± 0.11
9×9	98.68 ± 0.29	99.51 ± 0.25	99.73 ± 0.15
11×11	98.70 ± 0.21	99.57 ± 0.54	99.79 ± 0.08

Fourth, to evaluate the influence of the spatialized input, the proposed models are tested using different spatial sizes for input cuboids. Table 6.5 shows that the proposed SSRNs perform robustly for different spatial sizes if these sizes are equal to or larger than 7×7, because the SSRN learns discriminative spatial features of input data. In all three datasets, the classification results increase with the spatial size of input cuboids. The important role of spatial context that this experiment demonstrated is in accordance with results in other publications [47, 55]. Considering the larger input sizes lead to higher classification accuracy, the spatial size of input HSI data is fixed to make a fair comparison between different classification methods.

6.2.1 Classification Results

The SSRN are compared with kernel SVM [70] and state-of-the-art deep learning models, such as SAE [9] and 3D CNN [8]. To demonstrate the effectiveness of the spectral and

Table 6.6: Classification Results of Different Methods for IN Dataset

	SVM	SAE	CNN	CNNL	SPA	SPC	SSRN
OA(%)	81.67	85.47	97.41	95.78	98.01	90.68	**99.19**
	±0.65	±0.58	±0.43	±0.71	±0.37	±0.75	**±0.26**
AA(%)	79.84	86.34	97.39	95.67	98.15	92.00	**98.93**
	±3.37	±1.14	±0.56	±1.23	±0.56	±2.84	**±0.59**
$\kappa \times 100$	78.76	83.42	97.05	95.18	97.73	89.36	**99.07**
	±0.77	±0.66	±0.49	±0.81	±0.42	±0.86	**±0.30**
1	96.78	81.82	100.0	96.17	98.71	83.15	97.82
2	78.74	82.16	97.27	95.31	97.60	86.81	99.17
3	82.26	77.54	98.00	95.31	98.27	87.34	99.53
4	99.03	68.11	92.81	88.58	96.36	91.32	97.79
5	93.75	94.36	99.25	99.24	98.67	97.54	99.24
6	85.96	94.45	99.52	98.72	99.69	97.88	99.51
7	40.00	94.70	97.58	96.13	97.92	89.33	98.70
8	91.80	94.36	99.00	98.58	99.26	90.85	99.85
9	0	82.56	96.95	96.32	100.0	100.0	98.50
10	86.00	81.28	95.38	94.35	97.48	81.92	98.74
11	70.94	84.47	97.72	96.28	98.16	91.68	99.30
12	74.73	83.77	97.13	93.07	95.84	85.14	98.43
13	99.04	96.42	99.65	98.01	99.59	99.72	100.0
14	94.29	92.27	97.95	96.62	98.34	97.44	99.31
15	85.11	80.63	92.30	90.90	96.67	93.43	99.20
16	96.78	81.82	100.0	96.17	97.89	83.15	97.82

spatial residual blocks in the proposed framework, the networks that only contain the spectral feature learning part (SPC) and the ones that only contain spatial feature learning part (SPA) are also tested. Moreover, the longer versions of 3D CNN (denote as CNNL)

Table 6.7: Classification Results of Different Methods for KSC Dataset

	SVM	SAE	CNN	CNNL	SPA	SPC	SSRN
OA(%)	80.29 ±0.58	92.99 ±0.82	97.08 ±0.47	95.45 ±0.45	98.63 ±0.38	97.90 ±0.49	**99.61** **±0.22**
AA(%)	65.64 ±0.86	89.76 ±1.25	95.09 ±0.70	92.56 ±0.99	97.81 ±0.64	96.56 ±0.69	**99.33** **±0.57**
$\kappa\times 100$	77.98 ±0.65	92.18 ±0.91	96.74 ±0.53	94.93 ±0.50	98.47 ±0.42	97.66 ±0.55	**99.56** **±0.25**
1	92.16	93.04	99.00	98.47	99.40	99.11	99.70
2	86.16	92.04	98.48	95.20	99.18	99.19	99.88
3	42.55	85.59	92.16	87.53	95.39	92.60	99.00
4	67.69	72.12	81.84	73.35	93.45	85.49	98.26
5	0	82.20	85.38	77.21	95.70	89.63	99.03
6	54.71	83.15	90.96	90.26	96.27	95.94	99.43
7	0	76.46	93.21	89.63	95.19	96.38	97.03
8	65.12	94.10	98.21	97.28	98.67	98.09	99.54
9	67.82	94.57	99.04	98.05	99.43	99.53	99.70
10	93.40	98.91	99.85	99.40	99.96	99.96	99.96
11	100.0	98.39	98.89	98.72	99.63	99.86	99.80
12	83.75	96.42	99.43	98.63	99.31	99.51	100.0
13	100.0	99.83	99.79	99.48	99.89	99.97	100.0

generated from the SPA models without skip connections are evaluated to study the effect of the designed spatial residual architecture on the decreasing-accuracy phenomenon [8]. To make a fair comparison, the input volume size is set to $7\times 7\times b$ for all methods and tuned these competitors to their optimal settings. 20%, 20%, and 10% labeled 3D HSI cuboids are random selected as training groups for IN, KSC, and UP datasets, respectively.

Tables VI to VIII report the OAs, AAs, Kappa coefficients, and the classification accuracies of all classes for HSI classification. In all three cases, the SSRN achieved the

Table 6.8: Classification Results of Different Methods for UP Dataset

	SVM	SAE	CNN	CNNL	SPA	SPC	SSRN
OA(%)	90.58 ±0.47	94.25 ±0.18	98.85 ±0.15	98.64 ±0.20	99.25 ±0.08	98.88 ±0.22	**99.79** **±0.09**
AA(%)	92.99 ±0.36	93.34 ±0.39	98.40 ±0.30	98.13 ±0.35	98.99 ±0.27	98.40 ±0.27	**99.66** **±0.17**
$\kappa \times 100$	87.21 ±0.70	92.35 ±0.25	98.47 ±0.20	98.20 ±0.26	99.00 ±0.12	98.52 ±0.30	**99.72** **±0.12**
1	87.24	94.59	98.98	98.29	99.25	99.01	99.92
2	89.93	96.44	99.45	99.50	99.58	99.81	99.96
3	86.48	84.57	96.04	94.54	98.06	95.46	98.46
4	99.95	97.37	99.58	99.28	99.76	99.54	99.69
5	95.78	99.60	99.39	99.94	99.50	99.84	99.99
6	97.69	93.39	99.70	99.50	99.74	99.18	99.94
7	95.44	88.57	97.18	96.82	97.87	98.15	99.82
8	84.40	85.66	95.73	95.54	97.44	94.65	99.22
9	100.0	99.88	99.56	99.74	99.74	99.99	99.95

highest classification accuracy and lower standard deviation than 3D CNN. For example, in the KSC dataset, SSRN (99.61%) delivered a roughly 2.5% increase of mean overall classification accuracy compared to CNN (97.08%). All deep learning methods generated obviously better outcomes than the kernel SVM. In all three datasets, the classification results of CNNL were worse than those of CNN. On the other hand, the SPA performed better than CNN. These outcomes showed the proposed spatial residual structures mitigate the declining-accuracy phenomenon. Furthermore, the SSRN constantly performed better than the SPA, because the spectral residual blocks learned spectral representations that are complementary to spatial features. Although there are few training samples for Oats and Grass-pasture-mowed classes in the IN dataset, the SSRN classified the testing data with higher than 98% mean classification accuracy. These results validated the robustness of the designed models in the face of difficult conditions.

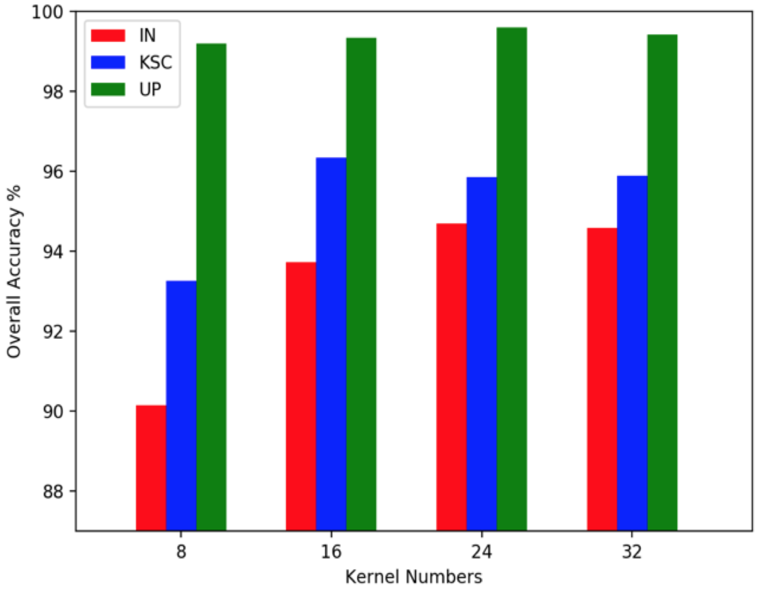


Figure 6.1: Overall accuracy (%) of SSRNs with different kernel numbers in IN, KSC, and UP datasets

Figures 6.2 to 6.4 visualize the classification results of the best trained models in three datasets, along with the false color images of original HSI and their corresponding ground truth maps. In all three cases, the qualitative comparison between different methods is in line with the quantitative comparison in Tables VI to VIII. The SPC generated classification maps with great noise. The SPA generated smoother results but still some dot noises exist in some classes. For example, the SPA reduced the speckles in the Wheat class of IN dataset and the Bare Soil class of UP dataset. Compared to other methods, the SSRN delivered the most accurate and smooth classification maps for all three HSIs, because the SSRN learned discriminative spectral and spatial features consecutively.

To test the robustness and generalizability of the proposed SSRN towards different numbers of training samples, 5%, 10%, 15%, and 20% labeled samples were randomly chosen as training data for IN and KSC datasets, and 4%, 6%, 8%, and 10% for the UP dataset. In Figure 6.5, The overall accuracies of different classifiers using different numbers of training data are illustrated. For a small number of training samples, when the SVM generated inferior overall accuracy, the SSRN still produced high classification accuracy, it is more obvious that SSRN performs the best than other methods because the SSRN extract more discriminative features than other methods. For a large number of training

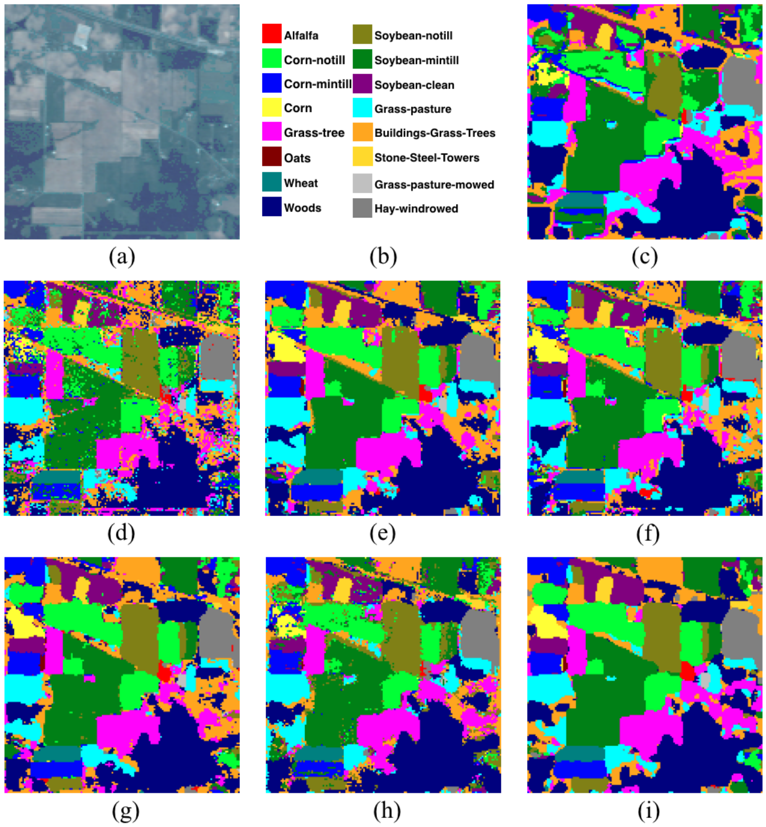


Figure 6.2: Classification results of best models for the IN dataset. (a) False color image. (b) Color table of land cover classes. (c) - (i) Classification results of SVM, SAE, CNN, CNNL, SPA, SPC, and SSRN

samples, the SSRN still generates the best classification outcomes in all three HSI datasets but the improvements are not that clear, simply because the classification accuracy are very high (higher than 99% overall accuracy).

To further validate the effectiveness of residual blocks for mitigating the accuracy-decreasing phenomenon, SSRN models with varying residual blocks were constructed for classifying 3D HSI data. SSRNs are tested with from 2 to 5 blocks and treated spectral and spatial residual blocks differently using the same settings as Tables 6.6, 6.7, and 6.8. In Figure 6.6, the overall classification accuracy differences between the deeper SSRNs and their shallow-layer counterparts are negligible. Therefore, in contrast to the obvious

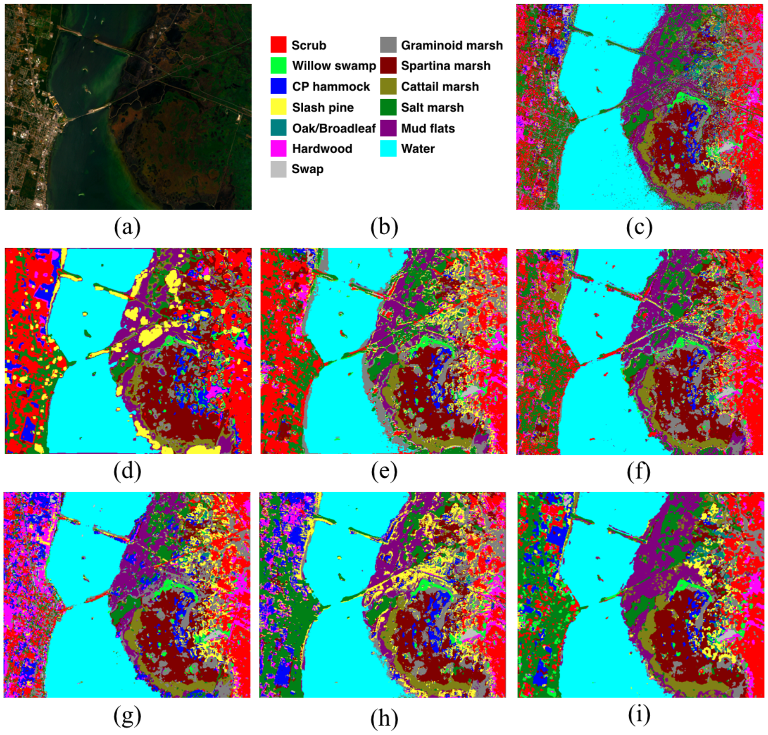
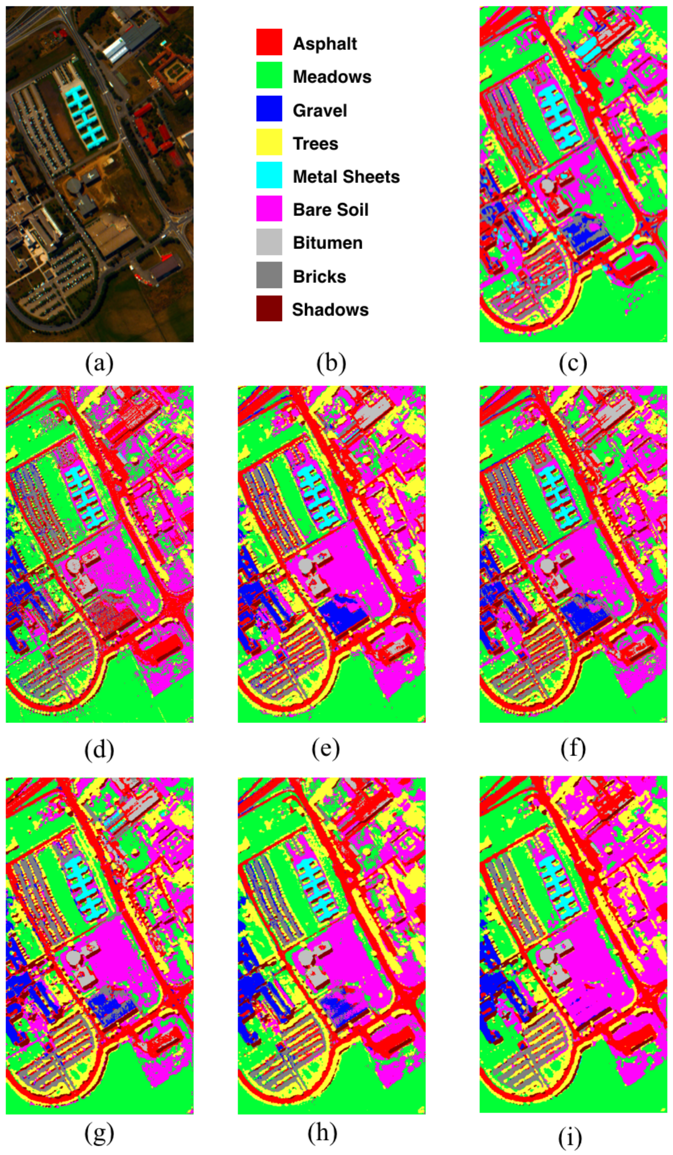


Figure 6.3: Classification results of best models for the KSC dataset. (a) False color image. (b) Color table of land cover classes. (c) - (i) Classification results of SVM, SAE, CNN, CNNL, SPA, SPC, and SSRN

accuracy-decreasing effects reported in [9] and [8], the consistent HSI classification performance of SSRNs with varying layers demonstrated that the residual connections mitigate the decreasing-accuracy effects in other deep learning models.

The training and testing times provide a direct measure of computational efficiency for the SSRN. All experiments were conducted on an MSI GT72S laptop with the GTX 980M graphical processing unit (GPU). Table 6.9 lists the training and testing times of the SSRN and other deep learning models. As presented in Table 6.9, the training times of the spectral section part (SPC) are 5 to 10 times longer than its spatial counterpart (SPA), because the spectral residual blocks preserved abundant features and kept the spatial size unchanged. In other words, the spectral residual blocks in the SSRN requires a larger amount of computational power than their spatial counterparts. The SSRN takes 6 to 10 times longer for training than the CNN, which means the SSRN is more computationally

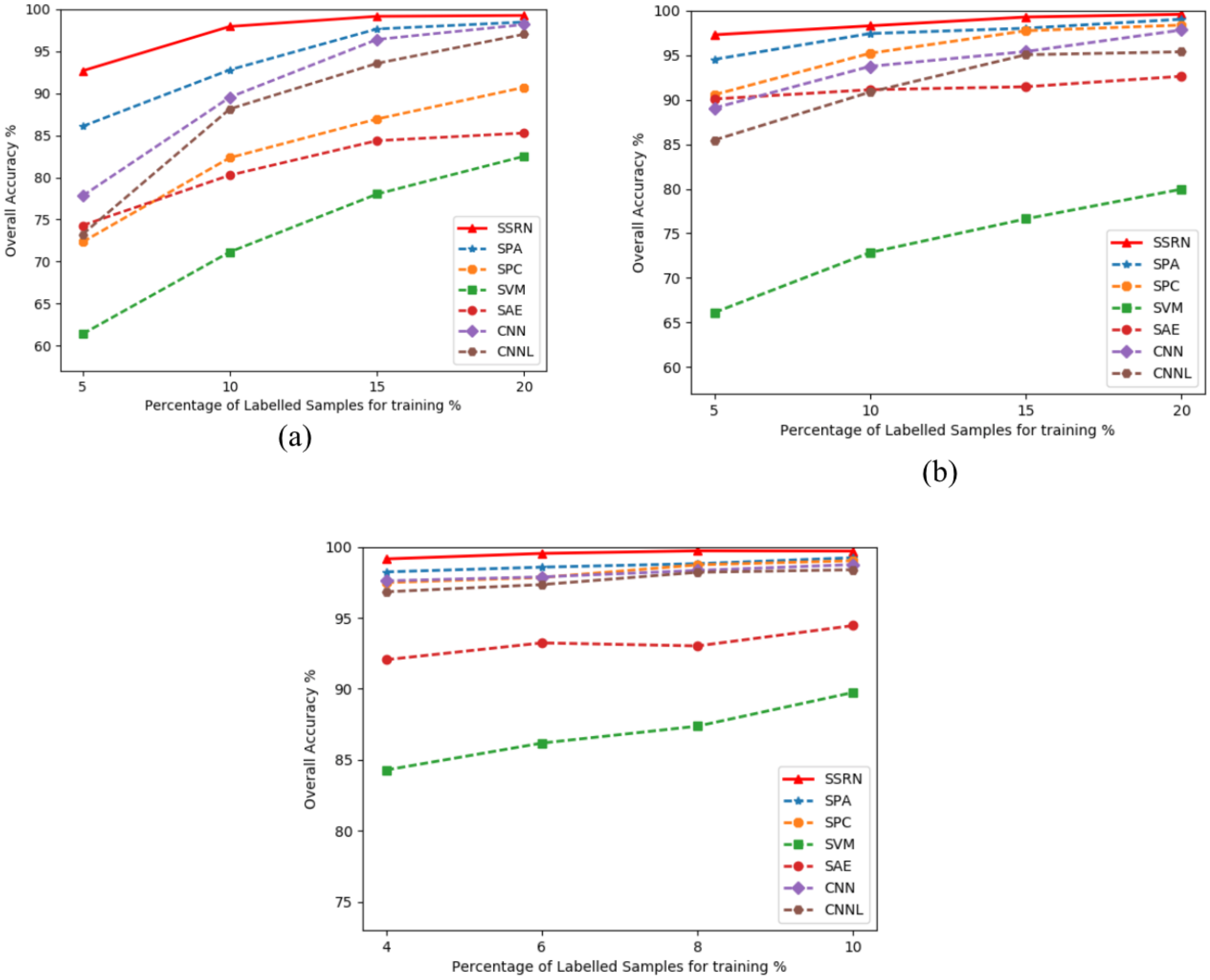


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.4: Classification results of best models for the UP dataset. (a) False color image. (b) Color table of land cover classes. (c) - (i) Classification results of SVM, SAE, CNN, CNNL, SPA, SPC, and SSRN



(a)

(b)

(c)

Figure 6.5: Overall accuracy of different methods with different training data percentages. (a) IN dataset. (b) KSC dataset. (c) UP dataset.

expensive than the CNN. Fortunately, the adoption of GPU has largely alleviated the extra computational costs and reduced the training times.

Table 6.9: Training and Testing Times of Different Models for Three HSI Datasets

		IN	KSC	UP
SAE	Train.(m)	3.5	2.6	12.8
	Test.(s)	2.0	0.8	7.7
CNN	Train.(m)	11.4	4.1	17.0
	Test.(s)	3.1	1.2	8.6
SPA	Train.(m)	10.9	5.4	26.3
	Test.(s)	3.0	1.5	14.5
SPC	Train.(m)	100.5	28.7	123.2
	Test.(s)	21.3	8.9	65.6
SSRN	Train.(m)	106.0	41.1	105.5
	Test.(s)	17.2	4.4	34.5

6.2.2 Discussion

The experimental outcomes validate the effectiveness of the SSRN framework. It is worth noting that different deep learning models usually prefer different hyper parameters, which poses a challenge for deploying these models. However, the classification performance of the SSRN with different settings is stable according to experiment results. Compared to traditional feature engineering based machine learning methods (e.g. kernel SVM), deep learning models have four advantages: first, automatic feature extraction, second, hierarchical non-linear transformation, third, objective functions that directly focus on classification in lieu of two independent steps, and fourth, the ability to utilize computational hardware (especially GPU) efficiently.

Three major differences exist between SSRNs and other deep learning models (e.g. SAE and CNN). First, the SSRN adopts residual connections that improve classification accuracy and make deep learning models much easier to train. Second, the SSRN treats spectral features and spatial features separately in two consecutive blocks, through which more discriminative features can be extracted. Third, owning to batch normalization operation at each convolutional layer, only hundreds of iterations is needed for training the SSRN instead of hundreds of thousands in [49].

Three main factors influence the HSI classification performance of supervised deep learning models: first, the number of training samples; second, the spatial size of input

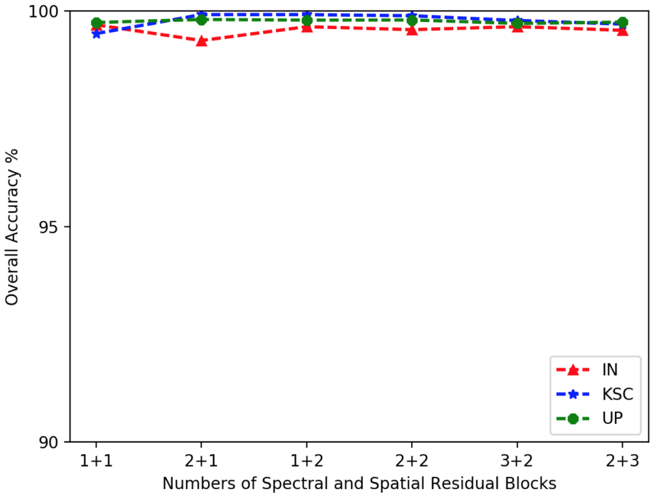


Figure 6.6: Overall accuracy of Spectral-Spatial Neural Networks with varying layers and combinations of residual blocks. The '$x + y$' formation in the horizontal axis denotes a SSRN with x spectral and y spatial residual blocks.

data; and third, the representative capacity of the designed models. Because the SSRN obtained very high classification accuracy for relatively few land cover categories, data augmentation [8] is not employed to further boost the classification performance of the SSRN despite a small number of training samples. Given a fixed model, the more data used for training, and the more information these data contain, the higher classification accuracy deep learning models can generate. Therefore, to make a fair comparison, it is suggested to test different models under the same number of training samples and the same size for each input sample.

6.3 Semi-supervised Classification Results

In this section, two challenging HSI datasets are introduced, set hyper-parameters of semi-supervised GANs, and evaluate GAN-CRF models and their competitors using performance metrics including the classification accuracy of each land cover class, overall accuracy (OA), average accuracy (AA), and kappa coefficient (κ). Additionally, training and testing times of all semi-supervised GANs are recorded to quantitatively assess their computational complexity.

6.3.1 Experimental Datasets

Two most challenging and commonly studied HSI datasets – the Indian Pines (IN) and the University of Pavia (UP) – are used to evaluate the various types of semi-supervised GANs and GAN-CRF models for hyperspectral image classification. In both datasets, $\{100, 150, 200, 250, 300\}$ HSI cuboids are randomly selected with their annotations for training, and used the remaining cuboids for testing.

As shown in Figure 6.11 (a) - (b), the IN dataset contains 16 vegetation classes and has 145×145 pixels with a spatial resolution of 20 m by pixel. 200 hyperspectral bands are used for this study and they range from 400 nm to 2500 nm. As illustrated in Figure 6.13 (a) - (b), the UP dataset includes 9 urban land cover types and has 610×340 pixels with a spatial resolution of 1.3 m by pixel. 103 hyperspectral bands are used for this research and they range from 430 nm to 860 nm. The numbers of labeled HSI samples for each land cover class for the IN and UP datasets can be found in Figures 6.11 and 6.13, respectively. Given their relatively small numbers, the labeled hyperspectral groups X^1 used for training contain at least two samples for each land cover class to avoid the situation that no sampled HSI cuboids are sampled for rare classes, especially in the IN dataset.

6.3.2 Semi-supervised GAN Setting

Figure 4.4 takes the UP dataset as an example to show the discriminator and generator of a semi-supervised GAN for HSI classification. In this semi-supervised GAN, the generator takes a $1 \times 1 \times 200$ vector of Gaussian noise as the input and outputs a $9 \times 9 \times 103$ fake HSI cuboid aiming to make the discriminator classify it as real data. Concurrently, a real $9 \times 9 \times 103$ HSI cuboids is randomly sampled from a raw HSI as the input of the discriminator. In this study, according to the result of a grid search, the learning rate is set to 0.0007, batch size to 50, and the spatial size of sampled HSI cuboids to 9×9. Additionally, the Adam optimizer [34] is adopted to alternatingly train the discriminator and generator. After the hyper-parameters of semi-supervised GANs are configured, three factors that influence the classification performance of semi-supervised GANs are analyzed.

First, the kernel number of convolutional and transposed convolutional layers affects the feature extraction and representation capacity of semi-supervised GANs. As illustrated in Figure 4.4, the discriminator and generator of a semi-supervised GAN have the same kernel number in its convolutional and transposed convolutional layers. Different kernel numbers from 16 to 32 in an interval of 4 are tested for all convolutional or transposed convolutional layers of semi-supervised GANs. As shown in Figure 6.7, the semi-supervised GANs with 24 convolution kernels in each layer achieved the highest classification accuracy using the IN dataset, and their counterparts with 28 convolution kernels obtained the best classification performance using the UP dataset. These results are acquired in the 3000-epoch training for both datasets using randomly sampled 300 HSI cuboids.

Second, the depth of the spectral-spatial discriminators in semi-supervised GANs also impacts their classification performance. Therefore, semi-supervised GANs with from 4 to 8 layers are assessed, which includes spectral and spatial convolutional layers, with the same hyper-parameter setting for each dataset. To make a fair comparison, the generators of semi-supervised GANs are kept to have the same architecture as the generator in Figure 4.4. As demonstrated in Figure 6.8, the semi-supervised GANs with 3 spectral and 3 spatial convolutional layers obtained the highest overall accuracies in both datasets. The fact that classification performance of semi-supervised GANs decreases with more convolutional layers than the optimal '$3 + 3$' architecture shows discriminators with deeper layers overfit the small number of labeled real HSI samples.

Third, to evaluate the influence of unlabeled real HSI cuboids, three types of semi-supervised GANs are tested using different numbers of unlabeled HSI samples for the IN and UP datasets. The three semi-supervised GANs are the spectral GAN (SPC-GAN), and the spatial GAN (SPA-GAN), and the spectral-spatial GAN (SS-GAN). As shown in Figure 4.4, the SS-GAN has both spectral and spatial learning blocks in its discriminator, and the SPC-GAN and SPA-GAN contain only spectral and spatial blocks, respectively. Again, the same setting of generators are used for all semi-supervised GANs as the generator in Figure 4.4. Table 6.10 shows that adding real unlabeled HSI samples for training contributes little to and adding more unlabeled samples even jeopardizes the semi-supervised HSI classification accuracy, which is caused by the different data distribution between labeled and unlabeled HSI samples.

6.3.3 Comparison Results

The proposed semi-supervised GANs are compared to state-of-the-art GAN-based models, such as 1D-GAN [84] , AE-GAN [9], and CNN-GAN [103]. To demonstrate the effectiveness of the spectral-spatial architecture, spectral-spatial GANs (SS-GANs) that comprise three spectral and three spatial convolutional layers are also compared with their variants: SPC-GANs (three spectral layers) and SPA-GANs (three spatial layers). As shown in Figure 4.1, the HSI classification results of the spectral-spatial convolutional neural networks (SS-CNNs) are recorded as important baselines. The generators of all GANs are kept

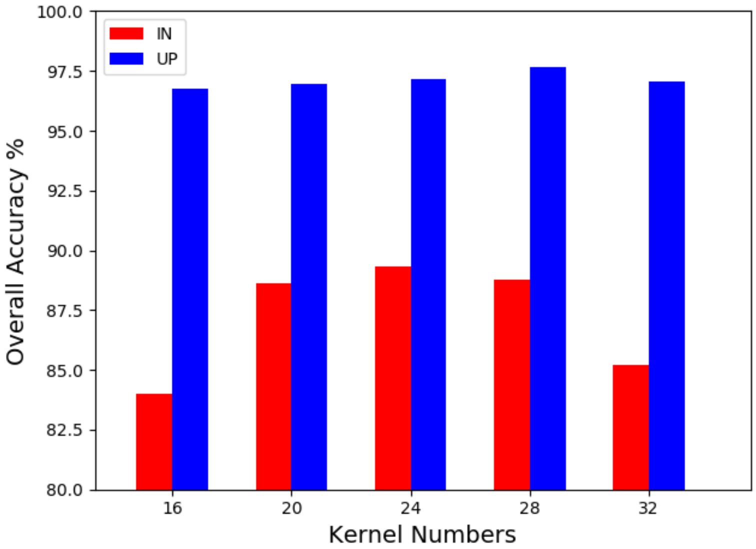


Figure 6.7: Overall accuracies of semi-supervised GANs with different kernel numbers in their convolutiolnal and transposed convolutional layers using 300 labeled HSI samples for training.

Table 6.10: Overall Accuracies (%) of semi-supervised GANs Using Different Numbers of Unlabeled and 200 Labeled HSI Samples in the IN and UP Datasets

Datasets	Models	0	1000	5000
IN	SPC-GAN	**63.21**	62.12	58.96
	SPA-GAN	**73.48**	71.28	67.62
	SS-GAN	81.12	**82.0**	78.0
UP	SPC-GAN	84.24	**84.69**	79.17
	SPA-GAN	91.01	**91.74**	87.35
	SS-GAN	**96.96**	95.76	93.90

the same, which consist of three spectral and four spatial transposed convolutions layers, each of which has 28 convolution kernels. Then, 3000 epochs for all GAN-based models are trained, and the input HSI cuboids is set with the same spatial size of 9 × 9× for

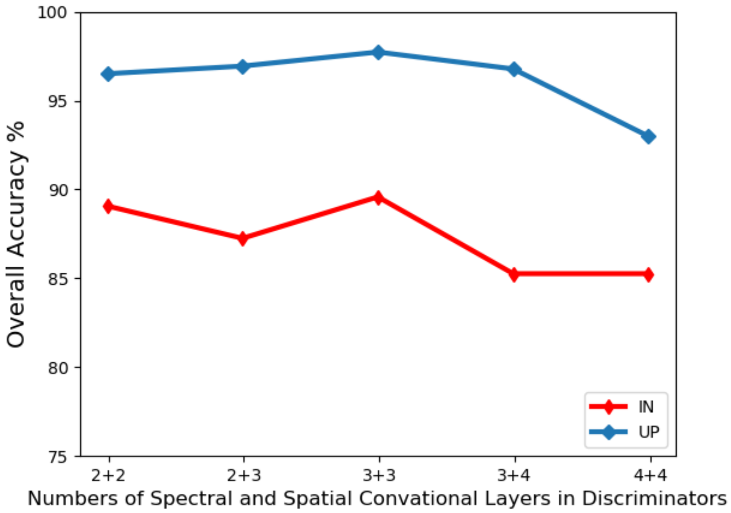


Figure 6.8: Overall accuracies of semi-supervised GANs that contain varying depths of spectral and spatial convolutional layers in their discriminators using 300 labeled HSI samples for training . The $x + y$ formation in the horizontal axis denotes a discriminator with x spectral and y spatial convolutional layers.

all methods that use spatial convolutional layers, and the competitors are tuned to their optimal settings.

Tables 6.11 and 6.12 report the classification performance, including accuracy of all land cover classes, OAs, AAs, and Kappa coefficients, of the IN and UP datasets, respectively. In most cases, the proposed semi-supervised GANs perform better than the state-of-the-art GAN-based models. Interestingly, the supervised benchmark SS-CNNs perform slightly better than SPA-GANs, which shows the discriminative feature learning capacity of spectral and spatial convolutional layers. More importantly, the SS-GANs achieved the highest overall classification accuracies (90.28% and 97.61% OAs for the IN and UP datasets, respectively) among all GAN-based models and the SS-CNNs. It is worth noting that he semi-supervised SS-GANs outperform fully supervised SS-CNNs in IN and UP datasets with 9.21% and 2.57%, respectively, which shows that the generated samples are helpful for improving classification accuracy. These results demonstrate the effectiveness of spectral-spatial convolutional architectures and semi-supervised adversarial training. Additionally, Tables 6.11 and 6.12 also show the training and testing times

Table 6.11: Classification Results, Training, and Testing Times of Different Deep Learning Models Using 300 HSI Samples for the IN Dataset

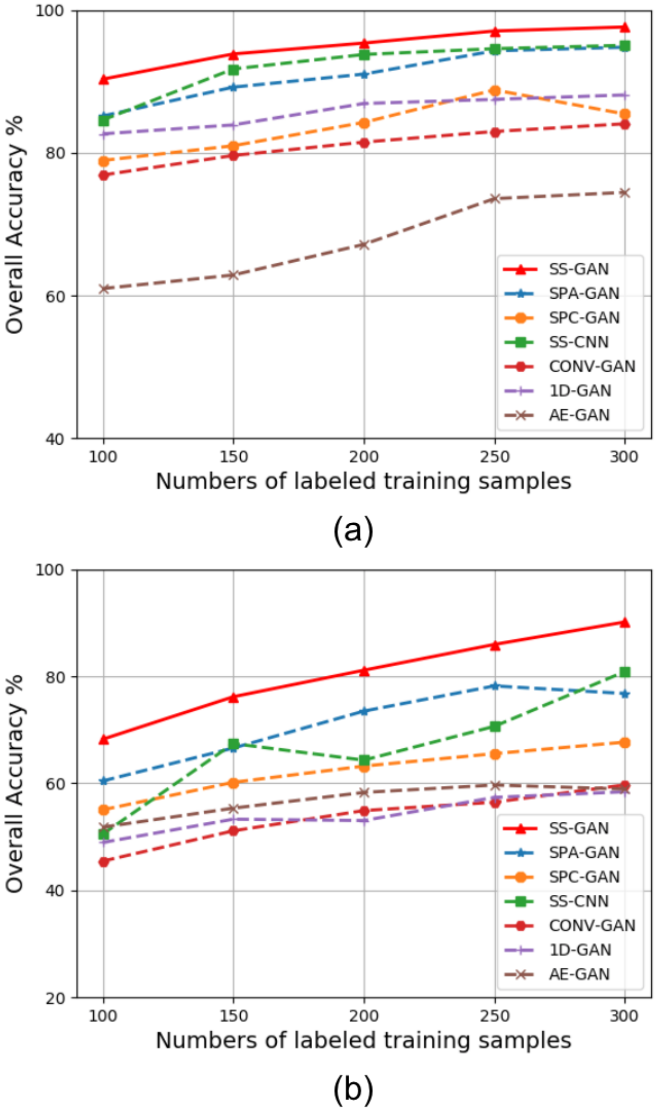
Class	Samples	1D-GAN	AE-GAN	CNN-GAN	SS-CNN	SPC-GAN	SPA-GAN	SS-GAN
1	3	50.00	0	46.94	83.33	66.67	**100.0**	96.43
2	41	51.98	51.20	46.45	77.88	52.71	64.48	**87.29**
3	29	52.41	38.75	43.17	**81.48**	48.55	61.49	77.84
4	7	35.38	22.37	47.66	76.47	56.45	81.56	**92.35**
5	14	68.83	49.74	47.67	78.81	69.44	82.96	**92.64**
6	20	87.30	81.09	63.37	87.14	86.40	93.98	**95.05**
7	2	45.83	0	20.75	42.85	67.86	**82.35**	76.47
8	15	86.86	87.84	79.13	89.45	91.72	90.75	**98.70**
9	3	33.33	0	34.62	**100.0**	42.86	45.45	57.89
10	36	39.29	51.15	61.37	77.94	59.30	78.83	**90.11**
11	64	54.20	64.83	67.49	80.97	72.96	81.60	**95.19**
12	22	45.57	33.00	34.20	62.52	42.82	53.68	**85.74**
13	4	63.75	81.31	69.41	**97.50**	93.71	87.32	93.30
14	28	80.36	74.63	77.32	88.63	79.80	82.32	**92.59**
15	10	39.24	47.91	64.09	76.92	66.76	70.72	**78.74**
16	2	98.63	0	84.29	**100.0**	77.78	94.44	95.29
OA (%)		59.44	60.26	60.68	81.07	67.92	76.65	**90.28**
AA (%)		58.31	42.74	55.93	81.37	67.23	78.23	**87.85**
$\kappa \times 100$		52.06	54.24	55.03	78.21	63.25	73.30	**88.92**
Training (s)		153.85	217.70	64.87	139.55	932.23	233.32	803.23
Testing (s)		0.59	0.60	0.35	4.117	5.88	1.28	5.09

Table 6.12: Classification Results, Training, and Testing Times of Different Deep Learning Models Using 300 HSI Samples for the UP Dataset

Class	Samples	1D-GAN	AE-GAN	CNN-GAN	SS-CNN	SPC-GAN	SPA-GAN	SS-GAN
1	47	84.74	62.51	73.38	**96.07**	84.74	91.10	95.62
2	132	92.50	92.02	90.17	97.57	87.31	96.93	**99.49**
3	15	75.75	39.25	58.09	72.82	60.77	78.84	**89.02**
4	20	93.46	84.55	98.39	**99.37**	97.07	98.94	98.65
5	11	99.55	94.72	99.41	98.97	95.06	99.55	**100.0**
6	35	86.77	62.72	74.21	98.18	86.70	92.71	**99.09**
7	13	82.43	40.46	89.29	96.38	85.86	95.76	**97.10**
8	21	73.79	51.78	83.65	82.81	75.85	86.88	**92.54**
9	6	98.13	66.14	99.30	99.36	96.56	99.79	**100.0**
OA (%)		88.36	75.10	84.23	95.04	85.78	93.97	**97.61**
AA (%)		87.46	66.02	85.10	93.50	85.55	93.39	**96.84**
$\kappa \times 100$		84.41	67.07	78.79	93.40	80.69	91.98	**96.82**
Training (s)		107.27	145.11	64.71	93.45	647.68	159.37	527.46
Testing (s)		2.06	1.34	1.76	14.30	18.38	4.03	15.36

of all models, which indicate the computational costs of these models. All experiments were conducted using an NVIDIA TITAN Xp graphical processing unit (GPU). In both datasets, the SPC-GANs are the slowest to train and the SS-GANs take about 6 times longer for training than SS-CNNs.

To test the robustness of the SS-GANs and their competitors, different numbers of labeled HSI cuboids in an interval of 50 from 100 to 300 are randomly sampled to train these semi-supervised GANs and SS-CNNs for the IN and UP datasets. As shown in Figure 6.9, the classification performance of SPA-GANs is comparable to that of SS-CNNs. AE-GANs perform clearly worse than other models because their fully connected layers fail to take the spectral-spatial characteristics of HSI samples into account. More importantly,


(a)

(b)

Figure 6.9: Overall accuracies of different semi-supervised GANs and the supervised benchmark SS-CNNs using from 100 to 300 HSI samples for training. (a) IN dataset. (b) UP dataset.

the proposed SS-GANs consistently outperform their semi-supervised competitors and SS-CNNs in both datasets. These results demonstrate the importance of accounting for the

Table 6.13: Overall Accuracies (%) of baseline classification results (base.) and different post-processing methods for the IN and UP datasets

	Base.	Mean	Max	Gauss.	Laplace	CRF
IN	86.99	**96.05**	94.53	95.09	94.29	95.16
UP	96.41	96.65	96.47	96.74	96.84	**98.27**

attributes of training data to design deep learning models, which is in line with the report of [99].

To evaluate the post-processing dense CRFs, semi-supervised GANs without CRFs (w/o CRF) are compared with their counterparts with CRFs (w/ CRF). Standard deviations in Equation (5.8) are set as $\theta_\alpha = 2$ and $\theta_\beta = 1$ in both datasets, and set constants in Equation (5.8) $c = 8$ and $c = 10$ for the IN and UP datasets, respectively. Also, the dense CRFs are compared to other alternative post-processing methods, including mean filter, maximum filter, Gaussian filter, and Laplace method. Table 6.13 shows that the CRF delivers comparable overall accuracy improvement for post-processing to the best performed mean filter using the IN dataset, and outperform all other methods using the UP dataset. This is caused by the homogeneous spatial layout of the former dataset and more heterogeneous distribution of the latter dataset. Therefore, the long-range correlation emphasized by CRFs facilitates the classification of HSI samples from heterogeneous areas.

In this study, the three most prominent principal component analysis (PCA) channels of HSI X, which affect only the pairwise term of CRFs, are used instead of raw HSI cuboids to facilitate the mean field approximation. Although As shown in Table 6.14, SS-GANs and SS-GAN-CRF models perform better than their competitors, and GAN-CRF models significantly enhance the classification performance of those models without integrating dense CRFs, Moreover, Figures 6.11 and 6.13 show the classification maps of all semi-supervised GANs and all GAN-CRF models. The qualitative results of these classification maps are in line with the quantitative report of Table 6.14. The SS-GAN-CRF models deliver the most accurate overall classification accuracies(96.30% and 99.31% OAs for the IN and UP datasets, respectively) and smoothest classification maps for both HSI datasets, because the SS-GANs learn the most discriminative spectral-spatial features and dense CRFs consider long-range correlations between similar HSI samples. Therefore, these classification outcomes validate the feasibility of integrating semi-supervised deep learning and graph models given limited labeled HSI samples for training.

Table 6.14: Overall Accuracies (%) of deep learning models and their refined results by adding dense CRFs using 300 labeled HSI samples for training

	IN Dataset		UP Dataset	
Models	w/o CRF	w/ CRF	w/o CRF	w/ CRF
1D-GAN	59.44	70.41	88.36	94.41
AE-GAN	60.26	76.08	75.10	90.44
CNN-GAN	60.28	73.83	84.23	90.42
SS-CNN	81.07	87.66	95.04	98.05
SPC-GAN	68.92	74.64	85.78	88.13
SPA-GAN	76.65	85.64	93.97	97.57
SS-GAN	**90.28**	**96.30**	**97.61**	**99.31**

6.3.4 Discussion

The GAN-CRF models incorporate the CRF as a post-processing step and build a graph upon the learned features and the softmax outputs of discriminators to refine HSI classification maps. Compared with those CRFs adopted in previous articles [90, 93], the fully connected CRFs consider the long-range correlations between HSI samples. This property helps GAN-CRF models to better filter noises in the homogeneous areas of some land cover classes. Compared to just a supervised discriminator, a GAN-CRF model integrates the advantages of deep learning models and probabilistic graph models and improves HSI classification accuracy. There are two main reasons for this improvement: 1) the synthetic HSI samples produced by generators help discriminators to learn more robust and discriminative features; 2) the subsequent dense CRFs consider the spectral similarity and spatial closeness of HSI samples to refine the softmax outputs conditional on these samples using the trained discriminators of GANs.

Four major insights are gained from the semi-supervised HSI classification outcomes of GANs and GAN-CRF models in both datasets. First, by taking the characteristics of training data into account, the discriminators of SS-GANs extract discriminative HSI features and achieve better classification accuracy. Second, generators of SS-GANs learn feature representation by producing synthetic HSI samples, and in turn make discriminators more robust to adversaries and learn more discriminative features. Therefore, this

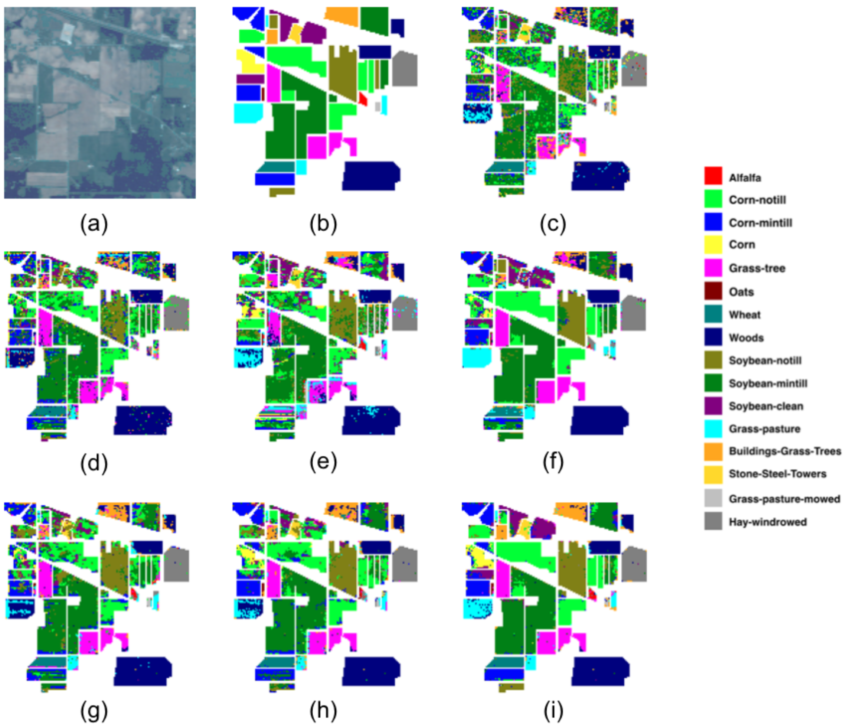


Figure 6.10: Classification results of semi-supervised GAN models, a supervised CNN, and their refined counterparts by adding dense CRFs using 300 labeled HSI samples for the IN dataset. (a) False color image. (b) Ground truth labels. (c) - (i) Classification maps of 1D-GAN, AE-GAN, CNN-GAN, SS-CNN, SPC-GAN, SPC-GAN, and SS-GAN.

adversarial training enables semi-supervised GANs to deliver superior classification outcomes to supervised deep learning models. Third, adding unlabeled real HSI samples to train semi-supervised GANs marginally improves or even jeopardizes the HSI classification results. Fourth, dense CRFs take the classification maps generated by semi-supervised GANs as an initialization and smooth the noisy classification maps by adding a pairwise term that imposes the correlation between similar or neighboring pixels from input HSIs.

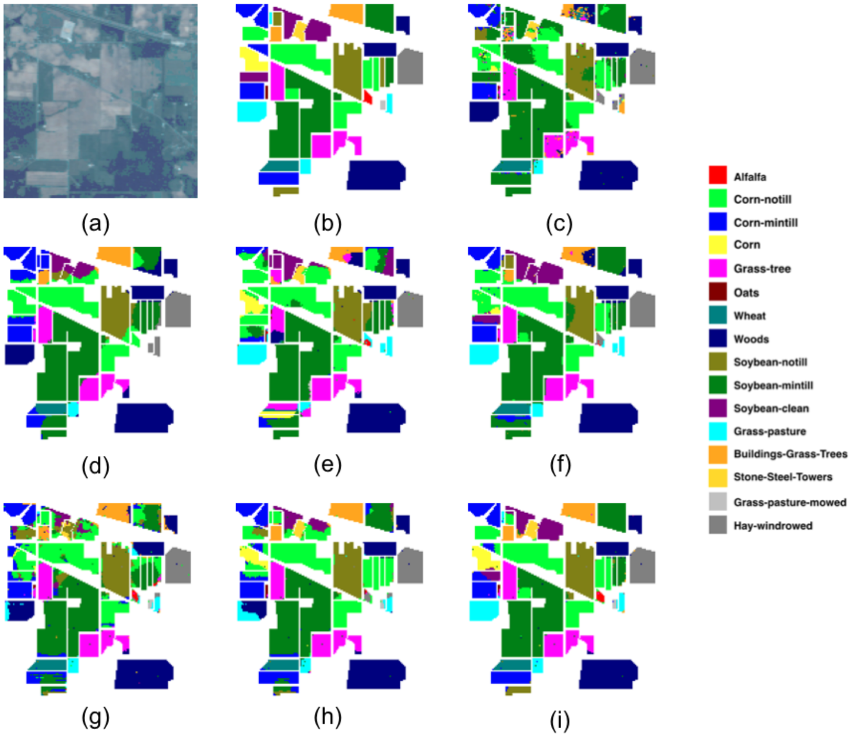


Figure 6.11: Classification results of semi-supervised GAN models and a supervised CNN that adopt dense CRFs for post-processing using 300 labeled HSI samples for the IN dataset. (a) False color image. (b) Ground truth labels. (c) - (i) Classification maps of 1D-GAN-CRF, AE-GAN-CRF, CNN-GAN-CRF, SS-CNN-CRF, SPC-GAN-CRF, SPA-GAN-CRF, and SS-GAN-CRF.

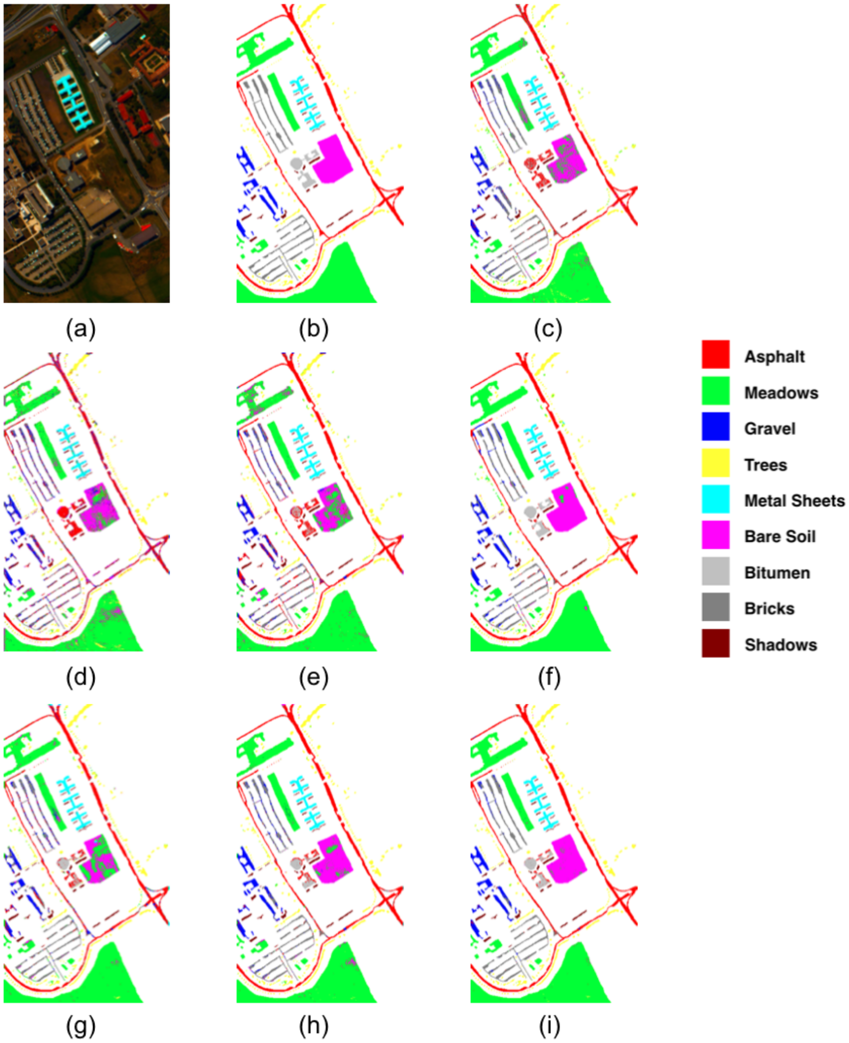


Figure 6.12: Classification results of semi-supervised GAN models, a supervised CNN, and their refined counterparts by adding dense CRFs using 300 labeled HSI samples for the UP dataset. (a) False color image. (b) Ground truth labels. (c) - (i) Classification maps of 1D-GAN, AE-GAN, CNN-GAN, SS-CNN, SPC-GAN, SPC-GAN, and SS-GAN.

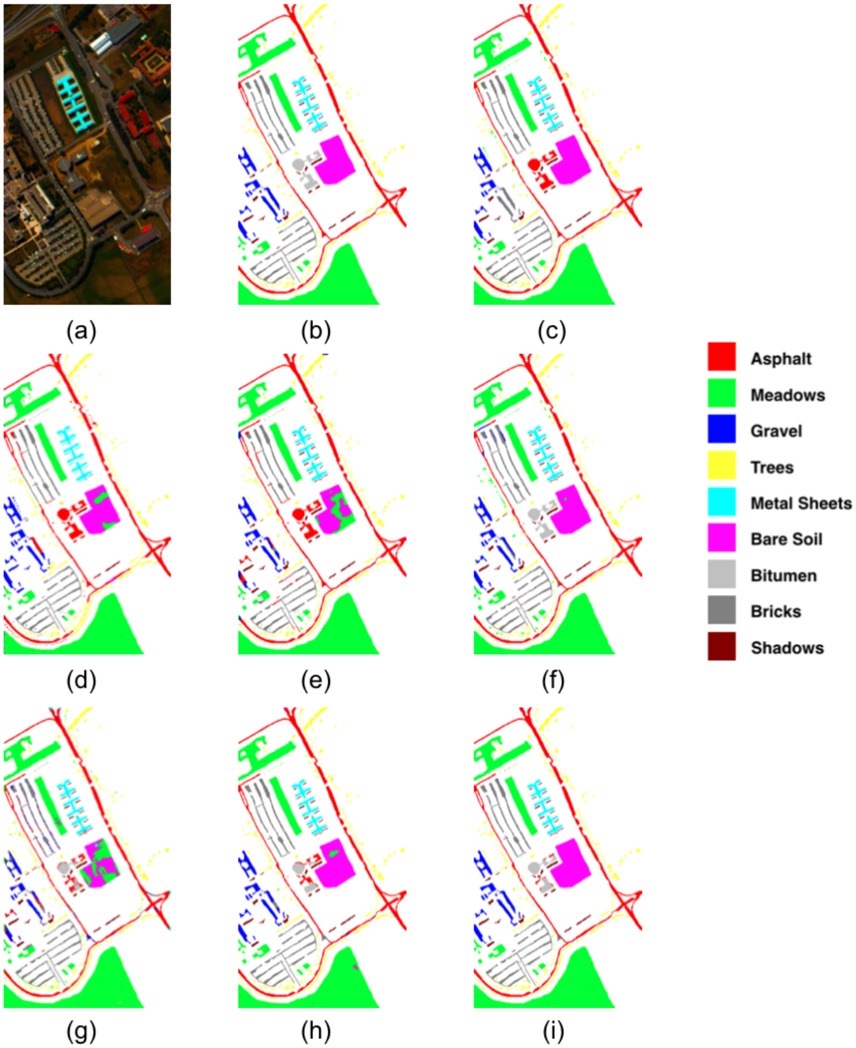


Figure 6.13: Classification results of models that adopt dense CRFs as a post-processing step using 300 labeled HSI samples for the UP dataset. (a) False color image. (b) Ground truth labels. (c) - (i) Classification maps of 1D-GAN-CRF, AE-GAN-CRF, CNN-GAN-CRF, SS-CNN-CRF, SPC-GAN-CRF, SPA-GAN-CRF, and SS-GAN-CRF.

Chapter 7

Conclusions

" Just as the industrial revolution relieved humanity of a lot of physical drudgery (what would your life be like if you had to sew your own clothes?), in the future AI will relieve humanity of mental drudgery. For example, having autonomous cars means we will no longer have to waste 3 years of our lives driving. This will give us more time to spend with loved ones and to pursue more worthy goals! "

– Andrew Ng

In this thesis, a supervised deep learning model SSRN and a semi-supervised model GAN-CRF have been proposed for hyperspectral image classification. The SSRN is composed of consecutive spectral and spatial feature learning blocks, and the GAN-CRF contains a spectral-spatial GAN and a post-processing CRF. Both models account for the characteristics of HSIs and achieve top-ranking performance compared to state-of-the-art deep learning models. In the following sections, the contributions of these two deep learning models are summarized separately.

7.1 Summary of Supervised Model

The essence of deep learning models is learning the representation of input data automatically without feature engineering, because the models themselves can extract discriminative features given appropriate architectural designs and training process settings. Moreover, these hyper parameter settings depend on the number of training samples and the spatial size of each sample. In the cases of HSI classification, one prominent challenge is the shortage of annotations. Thus, this thesis counters this obstacle with the proposed spectral-spatial residual architecture that takes both abundant spectral signatures and spatial contexts into account.

It is suggested that the deep learning methods need a significant amount of labeled data for training [48]. However, the experimental results have demonstrated that the proposed models, which have a spectral-spatial residual architecture and an appropriate regularization strategy, perform vigorously with large numbers as well as limited numbers of training samples. Also, according to the sensitivity test results, the proposed network can extract more discriminative spatial features with larger input cuboids, and simply expanding the sizes of input data will increase the classification accuracy. In other words, HSI classification models using training samples with more spatial information tend to have an advantage over the ones using training data with less spatial information. Therefore, it is advocated that the spatial size of input HSI data should be the same when comparing different classification methods. Considering the consistent performance in three widely studied HSI cases, the SSRN likely can outperform other machine learning competitors for HSI classification under same comparison standards in other cases.

7.2 Summary of Semi-supervised Model

In contrast to the supervised learning model, a semi-supervised GAN-CRF framework is proposed to address three commonly occurring challenges for HSI classification: the high spectral dimensionality of training data, the small numbers of labeled samples, and the noisy classification maps generated by deep learning models. First, four consecutively structured convolutional and transposed convolutional layers are designed to take the spectral-spatial characteristics of HSIs into consideration. Second, semi-supervised GANs, each of which comprises a generator and a discriminator, are established to extract discriminative features and to learn feature representation of HSI samples. Third, a probabilistic graphical model is integrated with a semi-supervised deep learning model to refine HSI classification maps. The experimental results using two of the most widely studied and challenging HSI datasets demonstrate that the spectral-spatial GANs (SS-GANs) perform the best among all semi-supervised GAN-based models and supervised benchmark models, and subsequently that the spectral-spatial GAN-CRF (SS-GAN-CRF) models achieved state-of-the-art performance for semi-supervised HSI classification.

The GAN-CRF models demonstrate an effective way to integrate two mainstream pixel-wise HSI classification methods — deep learning and probabilistic graphical models — and this framework can be easily generalized to other image interpretation cases. These two models have complementary advantages in the sense that deep learning models focus on discriminative feature extraction and implicit feature representation, and graph models emphasize the smoothness prior of images that is crucial for accurate classification and segmentation. However, the GAN-CRF framework presents a two-step setting because the dense CRFs function as a post-processing step to refine the classification maps generated by GANs.

7.3 Thesis Contribution Highlights

Discrimiantive and generative models have complementary advantages in the sense that deep learning models focus on hierarchical feature extraction and implicit feature representation, and graph models emphasize the smoothness prior that is crucial for accurate classification and segmentation of remote sensing images. The contributions of supervised models are fourfold.

- The designed SSRN, which contains consecutive spectral and spatial residual blocks, has alleviated the decreasing-accuracy phenomenon.

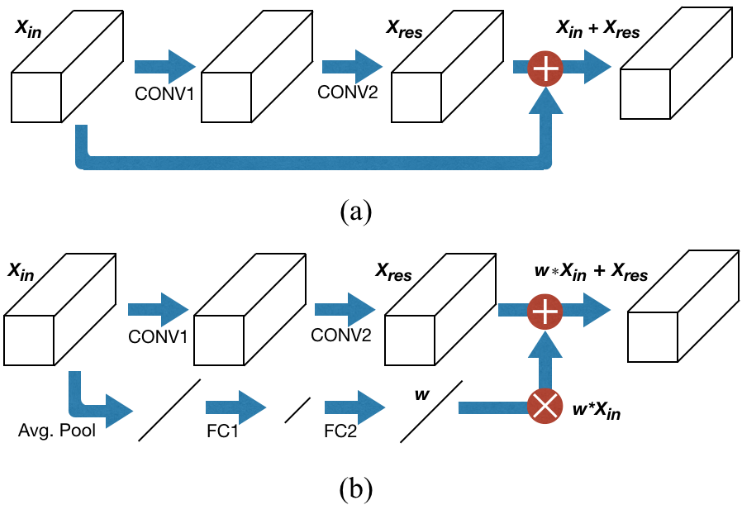
- The experimental results demonstrated that the SSRN performs consistently with the highest classification accuracy for all three types of HSI datasets with different challenges. It is worth noting that this network has delivered robust classification performance using small as well as large numbers of uneven training samples.
- Batch normalization, which is a simple and effective strategy to increase the mathematical stabilization of feature maps, are used for after each convolutional layer to regularize the training process and improved classification accuracy.
- The SSRN achieved state-of-the-art results with limited labeled 3D cuboids as training data in three cases and can easily be generalized to other remote sensing scenarios, because of their uniform structural design and deep feature learning capacity.

For semi-supervised models, four specific convolutional or transposed convolutional layers are designed for semi-supervised HSI classification, and spectral-spatial GANs are proposed that consists of these learning blocks. The main contributions of semi-supervised GAN-CRF models are as follows:

- The spectral-spatial attributes of HSIs are integrated into convolutional and transposed convolutional layers of a semi-supervised GAN to learn discriminative spectral-spatial features of HSI samples.
- Semi-supervised GANs are constructed to alleviate the shortage of labeled data through adversarial training, which is a zero-sum game between the discriminators and generators of GANs.
- The GAN-CRF framework demonstrates an effective way to integrate two mainstream machine leanring methods – deep learning and probabilistic graphical models – and this framework can be generalized to other image interpretation cases.
- The dense CRFs function as a post-processing step to refine the classification maps generated by GANs. In this case, the GAN and CRF are two independent components and this framework is a two-step framework.

7.4 Future Research

The SSRN and GAN-CRF models have established a solid foundation for supervised or semi-supervised feature learning in the context of hyperspectral image analysis, and deliv-



(a)

(b)

Figure 7.1: Residual blocks with and without attention mechanism, which is effective to boost discriminative feature learning and image recognition performance by re-calibrating channels. (a) A normal residual block. (b) A squeeze-and-excitation residual block.

ered promising classification results. Several future research directions could be built on top of these models or ideas presented in this thesis.

1. **Attention mechanism:** Attention mechanism has been proven effective for neural language processing and image recognition tasks [20,68]. For example, a squeeze-and-excitation (SE) module adopts the attention mechanism on image-level categorical information and improves the representational power of residual networks for large-scale classification via re-calibrating channel-wise weights [25]. As illustrated in Figure 7.1, this representational boost is achieved by squeezing out spatial information to generate channel-wise scaling weights. Several recent works also employ the attention mechanism to improve remotely sensed image recognition performance [26,52,72]. Since spectral(channel-wise) information is abundant for HSI, it is worth investigating a simple and effective way to utilize spatial or spectral (channel-wise) attention for achieving better supervised HSI classification performance.

2. **Domain adaptation:** Domain adaptation has been adopted extensively in state-of-the-art artificial intelligence models [13, 61, 63, 79]. The semi-supervised learning is a special case of domain adaptation where the deep learning models learn from

labelled samples during training. In general, neural network layers trained in one dataset embed prior knowledge to enable training new models that have more expressive capacity than those trained from scratch in different datasets. Considering the fact that only a small amount of remotely sensed data is labelled, transferring existing knowledge learned from these labelled samples from one dataset to much larger amounts of unlabelled samples from other datasets presents a valuable research direction.

3. **Segmentation methods:** In this thesis, HSI analysis is defined as a pixel-wise classification problem. However, the HSI samples (input) and classification map pixels (output) correspond spatially to each other. This spatial correspondence connects to an important branch in computer vision: semantic segmentation. The wide adoption of neural networks for classification is also witnessed in the semantic segmentation community [1, 7]. Pixel-wise classification was regarded as the same problem as semantic segmentation, but differences exist. Therefore, the key problem in this research direction rests on how to utilize the difference between classification and segmentation, as well as the data-specific characteristics.

4. **Transductive learning:** The semi-supervised GAN is an inductive learning framework, in which testing data is not accessible during network training. In contrast, transductive learning, in which testing data is accessible during training, could be helpful in real world applications [78]. The transductive learning strategy draws intuition directly from testing data and thus requires fewer labelled annotations. The challenge of transductive learning lies in the different distribution between training and accessible testing samples, resulting in classification performance dependent on testing data. This distribution difference makes transductive learning hard to generalize to unseen cases. Hence, how to improve the generalizability of the semi-supervised GAN models in a transductive manner is a topic worth exploring.

5. **Graph Neural Networks:** Recent works [29,56] suggest that neural networks could recognize patterns from complex and sparse graphs, including brain functional maps, social media networks, and scholar citation networks. Also, conditional random fields impose graph constraints on deep learning models in the proposed GAN-CRF model. Because images are grid-like graphs, deep neural networks that take graphs as input can also be used for HSI classification [56]. Specifically, the graph neural networks (GNNs) account for not only the spatial closeness but also the spectral similarity. Therefore, designing GNNs focus on learning graph features in the context of remotely sensed data remains a challenging and open question [3, 12].

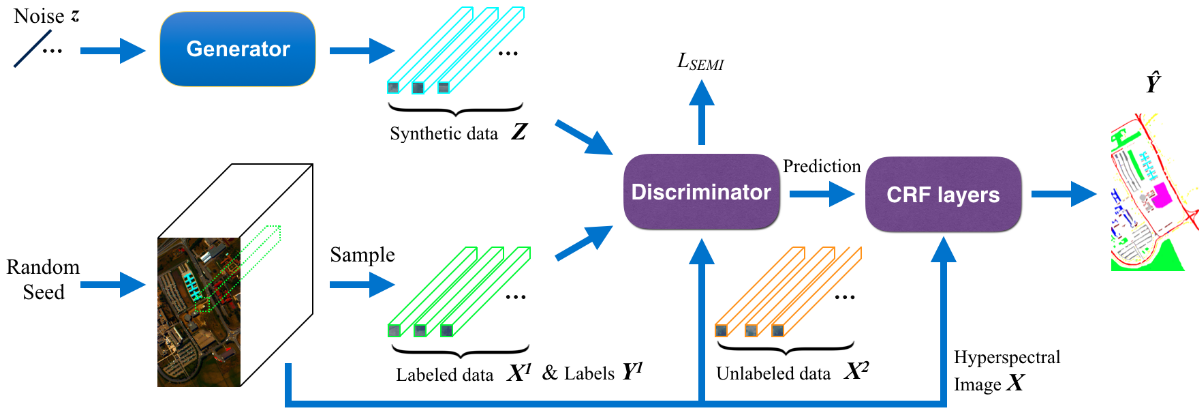


Figure 7.2: An end-to-end trainable semi-supervised GAN-CRF framework for HSI classification. This framework adopts two losses: The GAN loss term remains the same as used in training semi-supervised GANs, and the CRF loss term is the cross entropy between the pixel-wise output predictions of CRF layers and the HSI ground-truth targets.

6. **Integrated Frameworks:** The GAN-CRF model validates the feasibility to integrate GAN and CRF by demonstrating a way to achieve this target [97]. Specifically, a GAN is trained in a semi-supervised way and then the trained discriminator of the GAN is used to produce pixel-wise conditional probability maps. Then, a CRF considers the pixel-wise classification prediction holistically and adds structural constraints upon the discriminator outputs. However, the semi-supervised GAN and the dense CRF are trained separately, because different optimizers are needed for them. Therefore, future research should involve a joint training framework with a redesigned architecture. For example, the discriminator could be a local semantic segmentation network and the generator should be changed accordingly.

   To obtain a unified GAN-CRF model, novel CRF layers should be designed to approximate CRFs. The implementation keystone rests on the information, which includes the forward inferences and backward gradients, propagated between the discriminator of a GAN and the CRF layers. As shown in Figure 7.2, an integral model containing two loss terms is proposed. The GAN loss term remains the same as used in training semi-supervised GANs, and the CRF loss term is the cross entropy between the pixel-wise output predictions of CRF layers and the HSI ground-truth targets. Given that the message passing representation of CRFs can be implemented via arithmetic operations and convolution layers, this research line will be further explored for imposing graph constraints on deep learning models to construct an end-to-end trainable model.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[2] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36, 2012.

[3] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[4] Lorenzo Bruzzone, Mingmin Chi, and Mattia Marconcini. A novel transductive svm for semisupervised classification of remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11):3363–3373, 2006.

[5] Gustavo Camps-Valls, Tatyana V Bandos Marsheva, and Dengyong Zhou. Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3044–3054, 2007.

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

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[8] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram 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.

[9] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, and Yanfeng 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.

[10] Yushi Chen, Xing Zhao, and Xiuping 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.

[11] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, pages 6513–6523, 2017.

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.

[13] Cheng Deng, Xianglong Liu, Chao Li, and Dacheng Tao. Active multi-kernel domain adaptation for hyperspectral image classification. *Pattern Recognition*, 77:306–315, 2018.

[14] Huawu Deng and David A Clausi. Unsupervised segmentation of synthetic aperture radar sea ice imagery using a novel markov random field model. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):528–538, 2005.

[15] Ali El Zaart, Djemel Ziou, Shengrui Wang, and Qingshan Jiang. Segmentation of sar images. *Pattern Recognition*, 35(3):713–724, 2002.

[16] Yuan Fang, Linlin Xu, Junhuan Peng, Honglei Yang, Alexander Wong, and David A Clausi. Unsupervised bayesian classification of a hyperspectral image based on the spectral mixture model and markov random field. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, (99):1–13, 2018.

[17] Mathieu Fauvel, Jón Atli Benediktsson, Jocelyn Chanussot, and Johannes 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.

[18] M-F Auclair Fortier, Djemel Ziou, Costas Armenakis, and Shengrui Wang. Automated correction and updating of road databases from high-resolution imagery. *Canadian Journal of Remote Sensing*, 27(1):76–89, 2001.

[19] Marie-Flavie Auclair Fortier, Djemel Ziou, Costas Armenakis, and Shengrui Wang. Automated updating of road information from aerial images. In *American Society Photogrammetry and Remote Sensing Conference*, pages 16–23, 2000.

[20] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. *arXiv preprint arXiv:1809.02983*, 2018.

[21] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[25] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[26] Yuansheng Hua, Lichao Mou, and Xiao Xiang Zhu. Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional lstm network for multi-label aerial image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149:188–199, 2019.

[27] Xin Huang and Liangpei Zhang. An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 46(12):4173–4185, 2008.

[28] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of Thirty-Second International Conference on Machine Learning*, pages 448–456.

[29] Rongrong Ji, Yue Gao, Richang Hong, Qiong Liu, Dacheng Tao, and Xuelong Li. Spectral-spatial constraint hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(3):1811–1824, 2014.

[30] Sen Jia, Linlin Shen, Jiasong Zhu, and Qingquan Li. A 3-d gabor phase-based coding and matching framework for hyperspectral imagery classification. *IEEE Transactions on Cybernetics*, 48(4):1176–1188, 2018.

[31] Xiuping Jia, Bor-Chen Kuo, and Melba M Crawford. Feature mining for hyperspectral image classification. *Proceedings of IEEE*, 101(3):676–697, 2013.

[32] Xudong Kang, Shutao Li, and Jon Atli Benediktsson. Spectral–spatial hyperspectral image classification with edge-preserving filtering. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2666–2677, 2014.

[33] Mahdi Khodadadzadeh, Jun Li, Antonio Plaza, Hassan Ghassemian, José M Bioucas-Dias, and Xia Li. Spectral–spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(10):6298–6314, 2014.

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] Daphne Koller, Nir Friedman, and Francis Bach. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

[36] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

[38] Tae-Jung Kwon, Jonathan Li, and Alexander Wong. Etvos: An enhanced total variation optimization segmentation approach for sar sea-ice image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):925–934, 2012.

[39] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

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

[41] Hyungtae Lee and Heesung Kwon. Contextual deep cnn based hyperspectral classification. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, pages 3322–3325.

[42] Fan Li, Linlin Xu, Parthipan Siva, Alexander Wong, and David A Clausi. Hyperspectral image classification with limited labeled training samples using enhanced ensemble learning and conditional random fields. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2427–2438, 2015.

[43] Fan Li, Linlin Xu, Alexander Wong, and David A Clausi. Feature extraction for hyperspectral imagery via ensemble localized manifold learning. *IEEE Geoscience and Remote Sensing Letters*, 12(12):2486–2490, 2015.

[44] FY Li, Mohammad Javad Shafiee, Audrey G Chung, Brendan Chwyl, Farnoud Kazemzadeh, Alexander Wong, and J Zelek. High dynamic range map estimation via fully connected random fields with stochastic cliques. In *Proceedings of IEEE International Conference on Image Processing*, pages 2159–2163, 2015.

[45] Hong Li, Guangrun Xiao, Tian Xia, Yuan Y Tang, and Luoqing Li. Hyperspectral image classification using functional data analysis. *IEEE Transactions on Cybernetics*, 44(9):1544–1555, 2014.

[46] Jun Li, José M. Bioucas-Dias, and Antonio 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.

[47] Jun Li, Prashanth Reddy Marpu, Antonio Plaza, José M Bioucas-Dias, and Jon Atli Benediktsson. Generalized composite kernel framework for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 51(9):4816–4829, 2013.

[48] Wei Li, Guodong Wu, Fan Zhang, and Qian Du. Hyperspectral image classification using deep pixel-pair features. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):844–853, 2017.

[49] Ying Li, Haokui Zhang, and Qiang Shen. Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network. *Remote Sensing*, 9(1):67, 2017.

[50] Fulin Luo, Bo Du, Liangpei Zhang, Lefei Zhang, and Dacheng Tao. Feature learning using spectral-spatial hypergraph discriminant analysis for hyperspectral image. *IEEE Transactions on Cybernetics*, 49(7):2406–2419, 2019.

[51] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, and Georg Ostrovski. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[52] Lichao Mou, Yuansheng Hua, and Xiao Xiang Zhu. A relation-augmented fully convolutional network for semantic segmentationin aerial scenes. *arXiv preprint arXiv:1904.05730*, 2019.

[53] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[54] Mahesh Pal and Giles M Foody. Feature selection for classification of hyperspectral data by svm. *IEEE Transactions on Geoscience and Remote Sensing*, 48(5):2297–2307, 2010.

[55] Jiangtao Peng, Yicong Zhou, and CL Philip Chen. Region-kernel-based support vector machines for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(9):4810–4824, 2015.

[56] Anyong Qin, Zhaowei Shang, Jinyu Tian, Yulong Wang, Taiping Zhang, and Yuan Yan Tang. Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 16(2):241–245, 2019.

[57] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[58] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[59] Yunus Saatci and Andrew G Wilson. Bayesian gan. In *Advances in Neural Information Processing Systems*, pages 3622–3631, 2017.

[60] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[61] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[62] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[63] Zhuo Sun, Cheng Wang, Hanyun Wang, and Jonathan Li. Learn multiple-kernel svms for domain adaptation in hyperspectral data. *IEEE Geoscience and Remote Sensing Letters*, 10(5):1224–1228, 2013.

[64] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[65] Yuliya Tarabalka, Jocelyn Chanussot, and Jón Atli Benediktsson. Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(5):1267–1279, 2010.

[66] Yuliya Tarabalka, Mathieu Fauvel, Jocelyn Chanussot, and Jón Atli Benediktsson. Svm- and mrf-based method for accurate classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 7(4):736–740, 2010.

[67] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[69] Qi Wang, Jianzhe Lin, and Yuan Yuan. Salient band selection for hyperspectral image classification via manifold ranking. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1279–1289, 2016.

[70] Björn Waske, Sebastian van der Linden, Jón Atli Benediktsson, Andreas Rabe, and Patrick Hostert. Sensitivity of support vector machines to random feature selection in classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(7):2880–2889, 2010.

[71] Alexander Wong and David A Clausi. Arrsi: Automatic registration of remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5):1483–1493, 2007.

[72] Zhitong Xiong, Yuan Yuan, and Qi Wang. Ai-net: Attention inception neural networks for hyperspectral image classification. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, pages 2647–2650, 2018.

[73] Linlin Xu, David A Clausi, Fan Li, and Alexander Wong. Weakly supervised classification of remotely sensed imagery using label constraint and edge penalty. *IEEE Transactions on Geoscience and Remote Sensing*, 55(3):1424–1436, 2017.

[74] Linlin Xu, Fan Li, Alexander Wong, and David A Clausi. Hyperspectral image denoising using a spatial–spectral monte carlo sampling approach. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):3025–3038, 2015.

[75] Linlin Xu, Alexander Wong, Fan Li, and David A Clausi. Intrinsic representation of hyperspectral imagery for unsupervised feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 54(2):1118–1130, 2015.

[76] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.

[77] Lixia Yang, Shuyuan Yang, Penglei Jin, and Rui Zhang. Semi-supervised hyperspectral image classification using spatio-spectral laplacian support vector machine. *IEEE Geoscience and Remote Sensing Letters*, 11(3):651–655, 2014.

[78] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.

[79] Minchao Ye, Yuntao Qian, Jun Zhou, and Yuan Yan Tang. Dictionary learning-based feature-level domain adaptation for cross-scene hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(3):1544–1562, 2017.

[80] Peter Yu, AK Qin, and David A Clausi. Unsupervised polarimetric sar image segmentation and classification using region growing with edge penalty. *IEEE Transactions on Geoscience and Remote Sensing*, 50(4):1302–1317, 2012.

[81] Qiyao Yu and David A Clausi. Irgs: Image segmentation using edge penalties and region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2126–2139, 2008.

[82] Haoliang Yuan and Yuan Yan Tang. Spectral–spatial shared linear regression for hyperspectral image classification. *IEEE Transactions on Cybernetics*, 47(4):934–945, 2017.

[83] Yuan Yuan, Jianzhe Lin, and Qi Wang. Hyperspectral image classification via multitask joint sparse representation and stepwise mrf optimization. *IEEE Transactions on Cybernetics*, 46(12):2966–2977, 2016.

[84] Ying Zhan, Dan Hu, Yuntao Wang, and Xianchuan Yu. Semisupervised hyperspectral image classification based on generative adversarial networks. *IEEE Geoscience and Remote Sensing Letters*, 15(2):212–216, 2018.

[85] Lefei Zhang, Liangpei Zhang, Dacheng Tao, and Xin Huang. On combining multiple features for hyperspectral remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 50(3):879–893, 2012.

[86] Lefei Zhang, Liangpei Zhang, Dacheng Tao, Xin Huang, and Bo Du. Hyperspectral remote sensing image subpixel target detection based on supervised metric learning. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4955–4965, 2014.

[87] Liangpei Zhang, Lefei Zhang, and Bo Du. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):22–40, 2016.

[88] Liangpei Zhang, Yanfei Zhong, Bo Huang, Jianya Gong, and Pingxiang Li. Dimensionality reduction based on clonal selection for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4172–4186, 2007.

[89] Ji Zhao, Yanfei Zhong, Hong Shu, and Liangpei Zhang. High-resolution image classification integrating spectral-spatial-location cues by conditional random fields. *IEEE Transactions on Image Processing*, 25(9):4033–4045, 2016.

[90] Ji Zhao, Yanfei Zhong, and Liangpei Zhang. Detail-preserving smoothing classifier based on conditional random fields for high spatial resolution remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2440–2452, 2015.

[91] Wenzhi Zhao and Shihong 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.

[92] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[93] Ping Zhong and Runsheng Wang. Learning conditional random fields for classification of hyperspectral images. *IEEE Transactions on Image Processing*, 19(7):1890–1907, 2010.

[94] Ping Zhong and Runsheng Wang. Modeling and classifying hyperspectral imagery by crfs with sparse higher order potentials. *IEEE Transactions on Geoscience and Remote Sensing*, 49(2):688–705, 2011.

[95] Yanfei Zhong, Ji Zhao, and Liangpei Zhang. A hybrid object-oriented conditional random field classification framework for high spatial resolution remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11):7023–7037, 2014.

[96] Zilong Zhong and Jonathan Li. Generative adversarial networks and probabilistic graph models for hyperspectral image classification. In *Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[97] Zilong Zhong, Jonathan Li, David A. Clausi, and Alexander Wong. Generative adversarial networks and conditional random fields for hyperspectral image classification. *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2019.2915094, 2019.

[98] Zilong Zhong, Jonathan Li, Weihong Cui, and Han Jiang. Fully convolutional networks for building and road extraction: Preliminary results. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, pages 1591–1594, 2016.

[99] Zilong Zhong, Jonathan Li, Zhiming Luo, and Michael Chapman. Spectral-spatial residual network for hyperspectral image classification: A 3-d deep learning framework. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):847–858, 2018.

[100] Zilong Zhong, Jonathan Li, Lingfei Ma, Han Jiang, and He Zhao. Deep residual networks for hyperspectral image classification. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, pages 1824–1827, 2017.

[101] Yicong Zhou and Yantao Wei. Learning hierarchical spectral–spatial features for hyperspectral image classification. *IEEE Transactions on Cybernetics*, 46(7):1667–1678, 2016.

[102] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

[103] Lin Zhu, Yushi Chen, Pedram Ghamisi, and Jón Atli Benediktsson. Generative adversarial networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 2018.