SEMI SUPERVISED MACHINE LEARNING AND DEEP LEARNING BASED ANALYSIS FOR HYPERSPECTRAL REMOTE SENSING IMAGES

A Dissertation

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Electrical and Computer Engineering

By

Souvick Mukherjee

August 2019

SEMI SUPERVISED MACHINE LEARNING AND DEEP LEARNING BASED ANALYSIS FOR HYPERSPECTRAL REMOTE SENSING IMAGES

_____

Souvick Mukherjee

Approved:

_____

Chair of the Committee,
Dr. Saurabh Prasad, Assistant Professor,
Electrical and Computer Engineering

Committee Members:

_____

Dr. David Mayerich, Assistant Professor,
Electrical and Computer Engineering

_____

Dr. Thomas Hebert, Associate Professor,
Electrical and Computer Engineering

_____

Dr. Miao Pan, Associate Professor,
Electrical and Computer Engineering

_____

Dr. Demetrio Labate, Professor,
Mathematics

_____

Dr. Dalton Lunga, Research Scientist,
Oak Ridge National Laboratory

_____

Dr. Suresh K. Khator, Associate Dean,
Cullen College of Engineering

_____

Dr. Badrinath Roysam, Chair,
Electrical and Computer Engineering

SEMI SUPERVISED MACHINE LEARNING AND DEEP LEARNING BASED ANALYSIS FOR HYPERSPECTRAL REMOTE SENSING IMAGES

An Abstract

of a

Dissertation

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Electrical and Computer Engineering

By

Souvick Mukherjee

August 2019

Abstract

Hyperspectral Image Analysis has been an active area of research, especially in scenarios where discriminative features from classes having similar spectral characteristics have to be learned. We propose and implement novel machine learning techniques to address research problems in the field of Hyperspectral Image Analysis using remote sensing images. Each chapter in this dissertation presents a novel method from the field of machine learning with the end goal of robust classification of Hyperspectral Remote Sensing Images.

We describe common problems faced in the field of Hyperspectral Image Analysis, and addresses those problems by proposing novel techniques. One common problem is the lack of large quantities of labeled data, which leads to the problem of models overfitting to the limited number of labeled training samples. We propose a spatial-spectral unsupervised feature extraction / reduction approach in Chapter 2 of this dissertation. Another approach to address the specific problem of the lack of large quantities of labeled data samples is to use the large number of available unlabeled data samples to perform Semi-Supervised learning. Towards this goal, we propose a Semi-Supervised feature extraction / reduction approach in Chapter 3 of this dissertation. Following the same idea, and inspired by the recent advancements in the field of Deep Learning, we also propose a Semi-Supervised Deep Learning approach in Chapter 4 of this dissertation. Another recent development in the field of Deep Learning for color image analysis involves new variants of neural network architectures called Capsule Neural Networks, which can capture the spatial information along with the underlying context from the original images in a much more robust manner. We propose Semi-Supervised Capsule Neural Networks tailored towards hyperspectral image analysis in Chapter 5 of this dissertation. In the final Chapter

of this dissertation, Chapter 6, we propose an algorithm to perform label expansion for Semi-Supervised Deep Learning tasks, applied to the domain of large scale Road Segmentation of big cities (we show our results for Road Segmentation in the city of Las Vegas and Caracas, the capital of Venezuela).

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine learning based techniques have been used to address research problems in the field of Hyperspectral Image Analysis using remote sensing images, for a long time. Recently, with advances in the field of Deep Learning, Deep Learning has also been extensively used for Analysis of Hyperspectral Remote Sensing Images [4].

This dissertation describes general problems faced in the field of Hyperspectral Image Analysis, and addresses those problems by proposing and implementing novel techniques. Several mathematically based approaches have been proposed to address the major issues with hyperspectral remote sensing data, over the last few decades. It is well known that for hyperspectral images the lack of large quantities of labeled samples lead to the problem of overfitting on the training dataset and very poor performance on the test dataset [5, 6]. The loss of spatial details / information / context in the extracted features, with respect to the original images is another major issue. Researchers have proposed several novel statistically based approaches as spatial-spectral feature extraction and classification, semisupervised feature extraction and classification, to address the issues as mentioned before [7, 8]. With the intention of incorporating spatial details / context into the extracted features, we propose and implement a spatial-spectral unsupervised feature extraction / reduction approach

in Chapter 2 of this dissertation. Hyperspectral images contain a very large number of unlabled data samples. One way to exploit the structure of the underlying data is to use the unlabeled data samples in addition to the labeled data samples to perform Semi-Supervised learning / classification. Following this idea, in Chapter 3 of this dissertation we propose and implement a feature extraction / dimension reduction algorithm. Inspired by the recent advancements in the field of Deep Learning [9, 10, 11, 12], we also propose and implement a Semi-Supervised Deep Learning approach in Chapter 4 of this dissertation. Following some more recent developments in the field of Deep Learning, where researchers propose new types of neural networks called Capsule Neural Networks [13, 1, 14], which can preserve the spatial details from the original images, we propose and implement Semi-Supervised Capsule Neural Networks in Chapter 5 of this dissertation. In the penultimate Chapter of this dissertation, Chapter 6, we propose an algorithm to perform label expansion for Semi-Supervised Deep Learning tasks, applied to the domain of large scale Road Segmentation of big cities (we show our results for Road Segmentation in the city of Las Vegas and Caracas: the capital of Venezuela). The final Chapter of this dissertation, Chapter 7, contains the Conclusions and Future Work.

Contributions of this Dissertation:

• Chapter 1 of this Dissertation introduces the problem of Hyperspectral Image Analysis in the context of Remote Sensing Images.

• Chapter 2 of this dissertation proposes an algorithm which incorporates spatial details / context into the extracted features. We propose and implement a spatial-spectral unsupervised feature extraction / reduction approach.

• Chapter 3 of this dissertation proposes an algorithm which incorporates spatial details / context into the extracted features, in a similar manner as the previous Chapter. However, here we propose and implement a spatial-spectral Semi-Supervised feature extraction / reduction approach to make use of the available labeled samples.

• Chapter 4 of this dissertation proposes and implements a Semi-Supervised Deep Learning algorithm.

• Chapter 5 of this dissertation proposes an algorithm which can preserve the spatial context of the objects from the original images using a Semi-Supervised Learning approach.

• Chapter 6 of this dissertation proposes an algorithm to perform label expansion for Semi-Supervised Deep Learning tasks, applied to the domain of large scale Road Segmentation of big cities (we show our results for Road Segmentation in the city of Las Vegas and Caracas: the capital of Venezuela).

• The final chapter of this dissertation, Chapter 7, contains the Conclusions and Future Work.

Chapter 2

Unsupervised Local Angle Distance Preserving Embeddings with Spatial Context for Hyperspectral Image Analysis

Dimensionality reduction is an important pre-processing step for hyperspectral image analysis. High dimensional signals as hyperspectral signals have a large proportion of redundant information and it is necessary to find a good low dimensional projection subspace in order to train robust classifiers. The intra-class samples in hyperspectral or high-dimensional data often exhibit a large variability. This prevents the back-end classifier from learning a robust discriminative representation of the data. Dimensionality reduction also helps to potentially address the issue of computational and time constraints when training a classifier. It is also known that a large number of hyperspectral images are impacted by illumination variances and low quality pixels due to shadows or faults in capturing the images. In other words, intra-class samples may differ from each other due to illumination differences between

different parts of the image, likewise confusing the back-end classifier and leading to classification errors. Moreover, hyperspectral images tend to have a large number of unlabeled samples compared to labeled samples. Labeling samples in such images is costly as human intervention is required. To address these issues we propose an unsupervised spatial-spectral dimensionality reduction algorithm based on angular distances as a preprocessing step to spatial-spectral angular distance based back-end classification algorithms.

2.1 Introduction

Hyperspectral images capture a scene at a large number of wavelengths in contrast to RGB images. The high-dimensionality of the data normally helps to distinguish between classes which potentially have same reflectances, when captured at lower wavelength resolutions. Most of the hyperspectral images capture redundant information due to the very detailed spectral resolution. Hence, when traditional machine learning algorithms are used for training classifiers with such data, they are known to exhibit the curse of dimensionality. The redundant information in the images tends to increase the sparsity in the data, which in turn leads to an increase in the computational and time complexity of the algorithms. In a high dimensional space the data samples belonging to the same classes appear to exhibit different reflectance signatures, thereby making it difficult for the classifier to learn. This is traditionally known as the curse of dimensionality. To overcome such problems, usually the high-dimensional data is projected onto low dimensional subspaces by applying feature reduction or dimension reduction approaches. Several such feature reduction approaches are already in use as - [15, 16, 17, 18, 19, 20, 21, 22]. Supervised dimension reduction algorithms as [23, 24, 25, 26], which utilize the labeled data generally try to focus on reducing the within-class variance and increase the between class vari-

ance of the data samples. Whereas, the goal of unsupervised dimension reduction approaches [27] is to find an inherent structure in the data and arrange it, so as to reduce a distance measure between similar samples and increase the same distance measure between dissimilar samples. It has also been observed that certain hyperspectral images in which the intra-class samples are impacted highly by illumination differences and shadows, tend to work well with Angular feature extraction algorithms such as [24, 28] than Euclidean feature extraction algorithms [29, 30]. Traditionally, it has also been observed that spatial-spectral feature extraction algorithms perform better than only-spectral based algorithms [31, 32, 33, 34, 35, 36]. This can be attributed to the fact that knowing the spatial neighbors of a pixel of interest in a hyperspectral image, can provide knowledge about the class or label of the pixel of interest. Introducing spatial-contextual knowledge into the feature extraction or classification algorithms tends to reduce noisy classification results. The goal of this chapter is two fold: (1) To present a spatial-spectral unsupervised feature reduction algorithm, based on Angular distances instead of Euclidean distances, (2) To kernelize the proposed algorithms, so as to make them work when the datasets are governed by non-linear distributions.

The chapter is organized as follows: Section 4.1 introduces the problem of feature reduction specific to the context of Hyperspectral Image Analysis. Section 2 gives a brief perspective or description of the work related to our proposed method. Section 2.3 presents the proposed work. Section 2.4 describes the back-end classifier used to classify the features extracted by the proposed algorithms. Section 2.5 and Section 2.6 discuss about the Experimental Settings with Results and Conclusion, respectively.

2.2 Motivation and Related Work

2.2.1 Motivation

For the hyperspectral images, where the pixels are impacted by illumination differences and shadows it is well known that angular distance based algorithms play a more important role than euclidean distance based algorithms [24, 28]. This can be attributed to the fact that projecting the sample pixels of interest onto a unit hypersphere removes the illumination differences from the pixels and preserves the shapes of the objects from the original hyperspectral image, thereby generating illumination invariant features from the pixels of interest during training. Figure 2.1 depicts this observation. This phenomenon helps the back-end classifier to learn better by eliminating the intra-class variance in the reflectances of pixels belonging to the same class, which happens due to the presence of shadows, or low-quality pixels in the image. In simple words, intra-class samples from an image must not have different reflectances due to illumination differences between different regions in the image, the l_2 normalization approach to project the features onto a hypersphere, as discussed in this chapter, addresses this problem by removing the illumination differences and generating illumination invariant features. More recently, a similar observation in the field of face recognition, where the final classifier was learning only about the high quality facial images and ignoring the low quality facial images during training, led researchers to develop an angle based classifier [37], in the field of deep learning. Several other angular distance based classifiers have also been proposed recently in the field of deep learning, with the intention of addressing similar issues as discussed in this chapter [38, 39, 40].

It is also known that spatial-spectral algorithms perform better than algorithms which only focus on utilizing the spectral properties of the data. Thus we extend our proposed algorithm by embedding spatial-contextual information to it. In simple

words, we preserve the spatial neighbors of the pixels of interest from the original higher dimensional space, to the final lower dimensional subspace, by introducing spatial-contextual information to our proposed method, through the use of fixed sized rectangular windows or superpixels surrounding the pixels of interest.

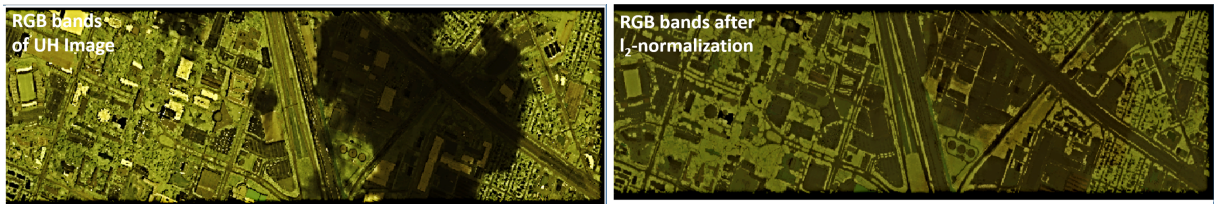


Figure 2.1: The effect of projecting samples onto a unit hypersphere. (Left) The original image impacted by illumination differences/clouds. (Right) l_2-normalized image where illumination difference is removed

2.2.2 Related Work

A semi-supervised discriminative feature reduction / extraction method related to this work has also been published by our group [28], but it assumes the availability of abundant labeled training samples, which is not always available for hyperspectral images. It is known that hyperspectral images contain abundant unlabeled samples, and we propose an unsupervised dimension reduction / feature extraction method in this chapter to exploit the information contained in them.

The proposed work is also inspired from and built upon the arXiv pre-print as proposed in [41], which uses fixed sized rectangular windows to incorporate spatial context to the pixels instead of superpixels as used in our proposed work. Superpixels can capture the spatially neighboring spectrally similar pixels and incorporate spatial context to the pixels in a much robust manner. In our proposed work we also kernelize our algorithms to make the features linearly separable in an infinite dimensional hyperspace. Our proposed feature extraction / dimension reduction algorithm is entirely unsupervised, in the sense that it does not need any unlabeled data samples

to find the corresponding projection matrix or lower dimensional projection subspace. Our contribution in this chapter is to introduce spatial contextual information in the dimension reduction algorithm by embedding superpixels, which are suited to capture the spectrally similar spatial neighbors. We also kernelize all our proposed algorithms so that they are able to produce more robust features when the datasets are inherently governed by non-linearity.

Local Angular Discriminant Analysis: LADA

LADA proposed in [42] finds a lower dimensional subspace by minimizing the ratio of within-class to between-class angular distance of its samples. Let $\tilde{x}_i \in \mathbb{R}^d$ be the i^{th} training sample, which is normalized and projected onto the surface of a unit hypersphere, $\boldsymbol{T} \in \mathbb{R}^{d\times r}$ be the projection matrix to be obtained, where d, is the dimension of original higher-dimensional space and r, is the dimension of the obtained lower-dimensional subspace. l and n_l be the class labels and the number of samples belonging to a certain class, respectively. The the optimization equation for finding the subspace corresponding to the LADA algorithm can be formulated as a generalized eigen-value problem as:

$$\boldsymbol{T}_{LADA} \approx \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\operatorname{argmin}} \Big[\big(\boldsymbol{T}^t \boldsymbol{O}^{(\mathrm{lw})} \boldsymbol{T} \big)^{-1} \big(\boldsymbol{T}^t \boldsymbol{O}^{(\mathrm{lb})} \boldsymbol{T} \big) \Big] \quad , \tag{2.1}$$

where $\boldsymbol{O^{(lw)}}$ and $\boldsymbol{O^{(lb)}}$ are the within-class and between-class angular outer product matrices, respectively, as:

$$\boldsymbol{O}^{(lw)} = \sum_{i,j=1}^{n} \tilde{W}_{ij}^{(lw)} \tilde{x}_i \tilde{x}_j^t \text{ and} \tag{2.2}$$

$$\boldsymbol{O}^{(lb)} = \sum_{i,j=1}^{n} \tilde{W}_{ij}^{(lb)} \tilde{x}_i \tilde{x}_j^t \quad , \tag{2.3}$$

where the within-class ($\tilde{\boldsymbol{W}}^{(\mathrm{lw})}$) and between-class ($\tilde{\boldsymbol{W}}^{(\mathrm{lb})}$) angular affinity matri-

ces, between samples x_i and x_j, are given by

$$\tilde{W}_{ij}^{(\mathrm{lw})} = \begin{cases} \tilde{W}_{ij}/n_l, & \text{if } y_i, y_j = l, \\ 0, & \text{if } y_i \neq y_j, \quad \text{and} \end{cases} \tag{2.4}$$

$$\tilde{W}_{ij}^{(\mathrm{lb})} = \begin{cases} \tilde{W}_{ij}(1/n - 1/n_l), & \text{if } y_i, y_j = l, \\ 1/n, & \text{if } y_i \neq y_j. \end{cases} \tag{2.5}$$

where the angular affinity $\tilde{W}_{ij} \in [0, 1]$, depicts the affinity between samples $\tilde{x}_i$ and $\tilde{x}_j$.

$$\tilde{W}_{ij} = \exp\left(-\frac{\|\tilde{x}_i - \tilde{x}_j\|^2}{\sigma}\right), \tag{2.6}$$

where σ is the parameter in the heat kernel, $\boldsymbol{T}$ is the projection matrix and r is the dimension of the reduced lower dimensional subspace.

2.3 Proposed Work: Superpixel Driven Spatial Context based LSPP and Kernel SLSPP

Figure 4.2 shows a simple block diagram representation of the proposed algorithms as described in this section. The yellow region in the figure below represents the spatial neighborhood for the unlabeled red pixel. The unlabeled image is segmented using the mentioned superpixel algorithm and utilized by the proposed SLSPP algorithm to find the projection subspace. The projection matrix obtained from the proposed feature extraction algorithm and unlabeled samples is then used to transform the labeled samples. The obtained features are then used for the purpose of classification using the SOMP classification algorithm. Figure 3.2 provides an overview of the feature embeddings for the algorithms described in this section. Upper half of the

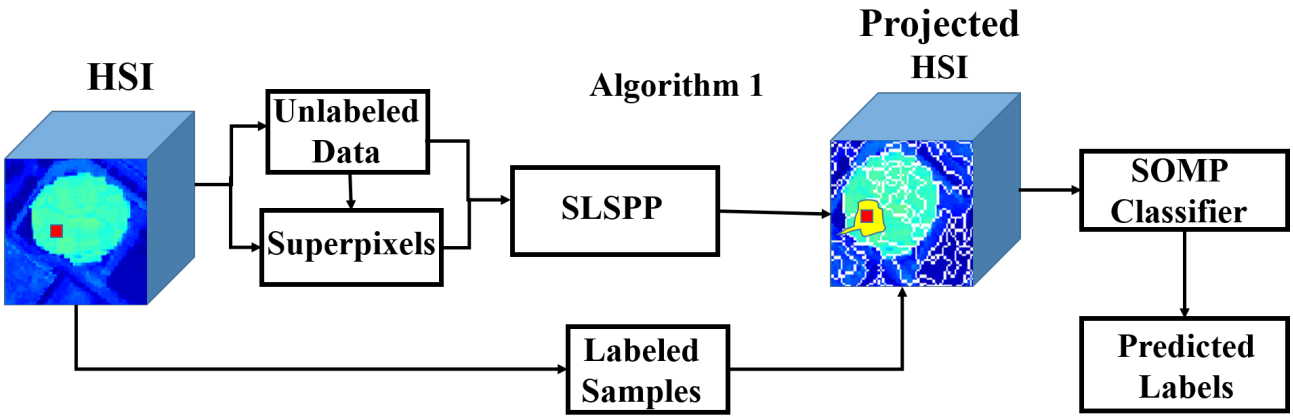


Figure 2.2: Block diagram representation of the proposed architecture

figure shows how the features get aligned when the LSPP algorithm is used for feature reduction and the lower half of the figure shows how introducing spatial-context in the LSPP algorithm helps to generate more robust features.

2.3.1 Algorithm

Algorithm 5 briefly describes the flow of processes used by us to implement our proposed algorithms. Entropy Rate (ER) superpixel generation is used to oversegment the image. In Algorithm 5 the entropy rate term $\mathcal{H}(A)$ increases with the addition of any edge to the set A, but the increase is larger when selecting edges that form compact and homogeneous clusters as described in [43]. The balancing term $\mathcal{B}(A)$ helps to generate clusters of similar sizes when the number of clusters are fixed. Thus, the algorithm tends to find clusters which are compact, homogeneous and of similar sizes in nature. Since the objective function in [43] increases monotonically, the number of connected components (clusters) in the graph (N_A) is exactly found to be equal to the number of desired superpixels ($\mathcal{N}$), due to the additional constraints.

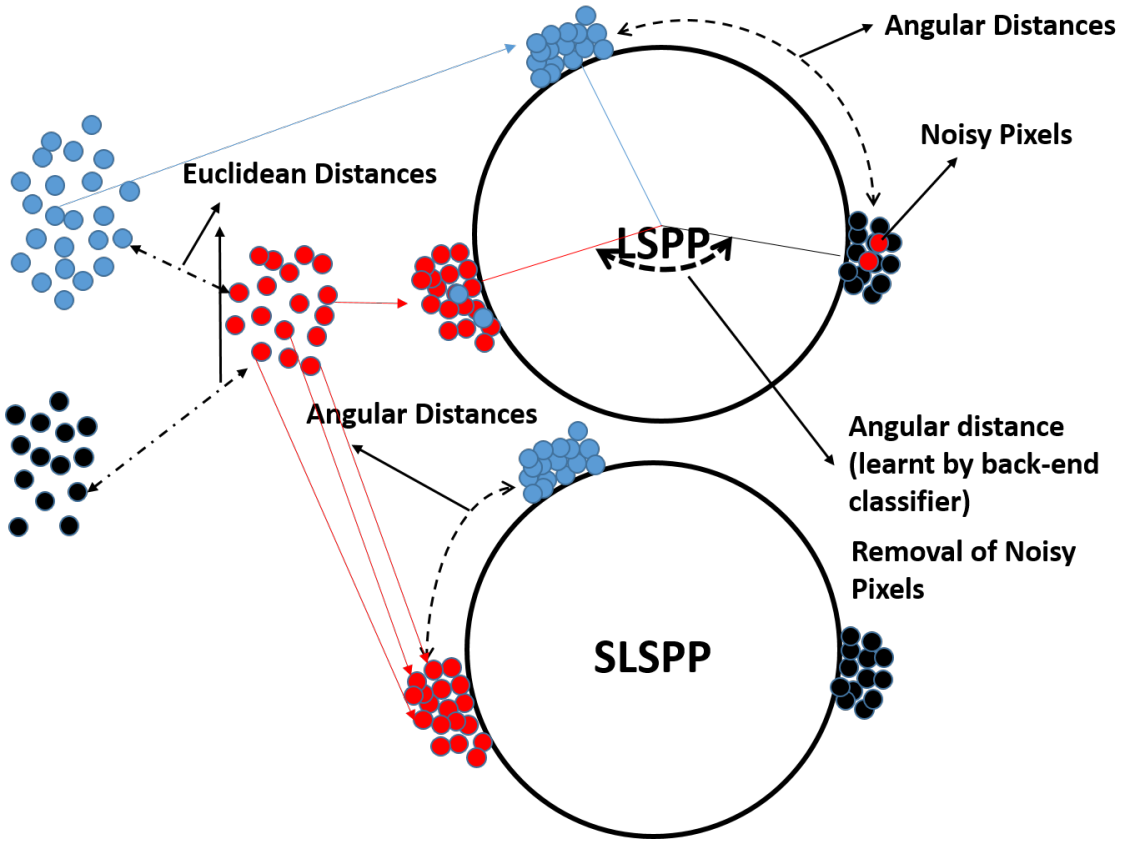


Figure 2.3: Neighborhood embedding of points for purely spectral (top) and spatial-spectral (bottom) feature extraction algorithms

Local Spectral angle Preserving Projections: LSPP

The optimization problem as proposed for the unsupervised Local Spectral angle Preserving Projection (LSPP) can be written as in equation (2.7). Let $\tilde{x}_i$ be an l_2-normalized training sample and $\boldsymbol{T} \in \mathbb{R}^{d \times r}$ be the $d \times r$ projection matrix, where r is the reduced dimensionality, then the objective function of LSPP is reduced to:

$$\begin{aligned} \boldsymbol{T}_{LSPP} &= \sum_i \sum_j \tilde{W}_{ij} (\boldsymbol{T}^t \tilde{x}_i)^t (\boldsymbol{T}^t \tilde{x}_j) \\ \boldsymbol{T}_{LSPP} &\approx \sum_i \sum_j tr[\tilde{W}_{ij} (\boldsymbol{T}^t \tilde{x}_i)^t (\boldsymbol{T}^t \tilde{x}_j)] \\ \boldsymbol{T}_{LSPP} &\approx \sum_i \sum_j tr[\tilde{W}_{ij} (\boldsymbol{T}^t \tilde{x}_j)(\boldsymbol{T}^t \tilde{x}_i)^t] \ , \\ \boldsymbol{T}_{LSPP} &\approx \sum_i \sum_j tr[\boldsymbol{T}^t (\tilde{W}_{ij} \tilde{x}_j \tilde{x}_i^t) \boldsymbol{T}] \approx tr[\boldsymbol{T}^t \tilde{\boldsymbol{X}} \tilde{\boldsymbol{W}} \tilde{\boldsymbol{X}}^t \boldsymbol{T}] \end{aligned} \tag{2.7}$$

$$\boldsymbol{T}_{LSPP} \approx \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\operatorname{argmax}} \left[\boldsymbol{T}^t \tilde{\boldsymbol{X}} \, \tilde{\boldsymbol{W}} \, \tilde{\boldsymbol{X}}^{\boldsymbol{t}} \boldsymbol{T} \right] \ s.t. \ (\boldsymbol{T}^t \tilde{\boldsymbol{X}} \, \tilde{\boldsymbol{D}} \, \tilde{\boldsymbol{X}}^{\boldsymbol{t}} \boldsymbol{T}) = \boldsymbol{I} \ , \text{ and} \tag{2.8}$$

$$\tilde{\boldsymbol{X}} \, \tilde{\boldsymbol{W}} \, \tilde{\boldsymbol{X}}^{t} \psi = \lambda \tilde{\boldsymbol{X}} \, \tilde{\boldsymbol{D}} \, \tilde{\boldsymbol{X}}^{\boldsymbol{t}} \psi \ , \tag{2.9}$$

where the angular affinity $\tilde{W}_{ij} \in [0, 1]$, depicts the affinity between samples $\tilde{x}_i$ and $\tilde{x}_j$, $\tilde{W}_{ij} = \exp\left(-\frac{\|\tilde{x}_i - \tilde{x}_j\|^2}{\sigma}\right)$, σ is the parameter in the heat kernel, $\tilde{\mathbf{X}}$ represents a matrix containing l_2-normalized unlabeled samples and $\tilde{\mathbf{W}}$ represents the affinity matrix of all those normalized unlabeled samples. The constraint $(\boldsymbol{T}^t \tilde{\boldsymbol{X}} \, \tilde{\boldsymbol{D}} \, \tilde{\boldsymbol{X}}^{\boldsymbol{t}} \boldsymbol{T}) = \boldsymbol{I}, where \ \tilde{D}_{ii} = \sum_j \tilde{W}_{ij}$ is imposed to avoid biases caused by different samples. Smaller the value of $\tilde{D}_{ii}$, corresponding to the i^{th} training sample, more important the i^{th} training sample is. The projection matrix $\boldsymbol{T}$ are the eigenvectors corresponding to the r largest eigenvalues, obtained after solving Equation (3.11), by simple eigen-value decomposition.

Spatial Local Spectral angle Preserving Projections (SLSPP) using Superpixels

The optimization problem for the proposed Spatial-LSPP algorithm can be written as in equation (2.10). Let $\tilde{x}_k, k \in \Omega_i$ be the l_2-normalized spatial neighborhood as defined by a superpixel, of labeled samples around a normalized labeled training sample $\tilde{x}_i$ and $\boldsymbol{T} \in \mathbb{R}^{d \times r}$ be the $d \times r$ projection matrix, where r is the reduced dimensionality, then the objective function of SLSPP is reduced to:

$$\begin{aligned} \boldsymbol{T}_{SLSPP} &= \sum_i \sum_{k \in \Omega_i} \tilde{W}_{ik} (\boldsymbol{T}^t \tilde{x}_i)^t (\boldsymbol{T}^t \tilde{x}_k) \\ \boldsymbol{T}_{SLSPP} &\approx \sum_i \sum_{k \in \Omega_i} tr[\tilde{W}_{ik} (\boldsymbol{T}^t \tilde{x}_i)^t (\boldsymbol{T}^t \tilde{x}_k)] \\ \boldsymbol{T}_{SLSPP} &\approx \sum_i \sum_{k \in \Omega_i} tr[\tilde{W}_{ik} (\boldsymbol{T}^t \tilde{x}_k)(\boldsymbol{T}^t \tilde{x}_i)^t] \\ \boldsymbol{T}_{SLSPP} &\approx \sum_i \sum_{k \in \Omega_i} tr[\boldsymbol{T}^t (\tilde{W}_{ik} \tilde{x}_k \tilde{x}_i^t) \boldsymbol{T}] \approx tr[\boldsymbol{T}^t \tilde{M} \boldsymbol{T}] \end{aligned} \tag{2.10}$$

where $\tilde{\boldsymbol{M}} = \sum_{\boldsymbol{i}} \sum_{\boldsymbol{k} \in \boldsymbol{\Omega_i}} (\tilde{\boldsymbol{W}}_{\boldsymbol{ik}} \tilde{\boldsymbol{x}}_{\boldsymbol{k}} \tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{t}})$

$$\boldsymbol{T}_{\boldsymbol{SLSPP}} \approx \underset{\boldsymbol{T} \in \mathbb{R}^{\boldsymbol{d} \times \boldsymbol{r}}}{\mathbf{argmax}} \left[\boldsymbol{T}^{\boldsymbol{t}} \tilde{\boldsymbol{M}} \boldsymbol{T} \right] \quad s.t. \quad \boldsymbol{T}^{\boldsymbol{t}} \tilde{\boldsymbol{X}} \tilde{\boldsymbol{D}}_{\boldsymbol{sp}} \tilde{\boldsymbol{X}}^{\boldsymbol{t}} \boldsymbol{T} = \boldsymbol{I} \ , \tag{2.11}$$

where $\tilde{\mathbf{X}}$ represents a matrix containing unlabeled samples and $\tilde{\mathbf{W}}$ represents the affinity matrix of all those unlabeled samples.

Kernel SLSPP

Samples from different classes may not always be linearly separable in the original space due to the inherent non-linear structure of the data. For such instances the SLSPP algorithm will fail to find a subspace that can angularly separate the samples. Formulating SLSPP in a Reproducible Kernel Hilbert Space (RKHS) $\mathcal{H}$ will overcome

Algorithm 1: *Pseudo code of the proposed feature extraction algorithms*

Input:

• Image: $I \in \mathbb{R}^{rw \times cl \times d}$

• Number of superpixels to be generated: $\mathcal{N}$

{*Superpixel Segmentation*} ***[28]***

• Generate compact, homogeneous and balanced Entropy-Rate superpixels: $X_i = \{x_j\}_{j=1}^{n_i}$ for $i = 1, 2, ..., \mathcal{N}$ (where i : superpixel index and j : pixel index in the i^{th} superpixel) from [43, 44]:

$$\textbf{max}_A \mathcal{H}(A) + \lambda \mathcal{B}(A)$$
$$\textbf{\textit{s.t.}}\ A \subseteq E,\ N_A \geq \mathcal{N} \text{ and } \lambda \geq 0$$

where A: selected edge set for segmenting the graph, $\mathcal{H}(A)$: Entropy Rate term, $\mathcal{B}(A)$: balancing term, λ: weight of the balancing term, E: Edges in the graph and N_A: number of connected components in the graph

{*Spectral-angle based Merging*} ***[28]***

• Generate merged superpixel corresponding to or encompassing each pixel:

for all $i \in 1, 2, ..., \mathcal{N}$ **do**
  for all $j \in 1, 2, ...,$ neighbors of superpixel i **do**
    $\theta = cos^{-1}\left[E[X_i] \odot E[X_j]^T / \left(||E[X_i]||\ ||E[X_j]||\right)\right]$
    if $\theta \leq \delta$ (minimum angle) **do**
      $\{X_i\} = \{X_i \cup X_j\}$
    end if
  end for
end for Where E represents the mean over a window of superpixels.

{*Feature Extraction*}

• Extract training samples: $\{\tilde{x}\}_{i=1}^{n} \in \mathbb{R}^d$ - from I; n: number of samples

• Compute $\tilde{\boldsymbol{M}}$ using unlabeled data from $\tilde{\boldsymbol{X}}$ as defined in Equation 2.10.

• Form an objective function to minimize the optimization Equation for SLSPP:

$$\boldsymbol{T}_{SLSPP} \approx \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\text{argmax}} \left[\boldsymbol{T}^t \tilde{\boldsymbol{M}} \boldsymbol{T}\right] \quad s.t. \quad \boldsymbol{T}^t \tilde{\boldsymbol{X}} \tilde{\boldsymbol{D}}_{sp} \tilde{\boldsymbol{X}}^t \boldsymbol{T} = \boldsymbol{I} \tag{2.12}$$

Where $\tilde{\boldsymbol{M}} = \sum_i \sum_{k \in \Omega_i} (\tilde{W}_{ik} \tilde{x}_k \tilde{x}_i^t)$

Output:

• Training points after projection: $\{p\}_{i=1}^{n} \in \mathbb{R}^r$

• Projection Matrix: $\boldsymbol{T} \in \mathbb{R}^{d \times r}$.

this limitation.

By applying the *kernel trick* [45], SLSPP can be extended to its kernel variant. Let n be the number of available samples and m be the number of neighbors for

each of those samples. Then, the term $\left(\boldsymbol{M} = \sum_{i} \sum_{k \in \Omega_i} \tilde{\boldsymbol{W}}_{ik} \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_i^t\right)$ can be simplified to $\sum_{s=1}^{n} \tilde{\boldsymbol{Z}}_{\boldsymbol{s}} \tilde{\boldsymbol{W}}_{\boldsymbol{s}} \tilde{\boldsymbol{X}}_{\boldsymbol{s}}^{\boldsymbol{t}}$ by using basic matrix algebra. Where:

$$\tilde{\boldsymbol{Z}} = \begin{bmatrix} \tilde{\boldsymbol{Z}}_{\boldsymbol{1}} \\ \tilde{\boldsymbol{Z}}_{\boldsymbol{2}} \\ \tilde{\boldsymbol{Z}}_{\boldsymbol{3}} \\ \tilde{\boldsymbol{Z}}_{\boldsymbol{n}} \end{bmatrix} = \begin{bmatrix} \tilde{X}_{1,1} & \tilde{X}_{1,2} & \tilde{X}_{1,3} \dots & \tilde{X}_{1,m} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{X}_{n,1} & \tilde{X}_{n,2} & \tilde{X}_{n,3} \dots & \tilde{X}_{n,m} \end{bmatrix}$$

$$\tilde{\boldsymbol{W}} = \begin{bmatrix} \tilde{\boldsymbol{W}}_{\boldsymbol{1}} & \tilde{\boldsymbol{W}}_{\boldsymbol{2}} & \tilde{\boldsymbol{W}}_{\boldsymbol{3}} & \dots & \tilde{\boldsymbol{W}}_{\boldsymbol{n}} \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{W}_{1,1} & \tilde{W}_{2,1} & \tilde{W}_{3,1} \dots & \tilde{W}_{n,1} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{W}_{1,m} & \tilde{W}_{2,m} & \tilde{W}_{3,m} \dots & \tilde{W}_{n,m} \end{bmatrix} \text{ and}$$

$$\tilde{\boldsymbol{X}}^{\boldsymbol{t}} = \begin{bmatrix} \tilde{X}_1^t & \tilde{X}_2^t & \tilde{X}_3^t \dots & \tilde{X}_n^t \end{bmatrix}$$

Substituting equations 2.10 and 2.11 into equation 3.11, for spatially constrained LSPP, the eigen-value problem simplifies to:

$$\left(\sum_{s=1}^{n} \tilde{\boldsymbol{Z}}_{\boldsymbol{s}} \tilde{\boldsymbol{W}}_{\boldsymbol{s}} \tilde{\boldsymbol{X}}_{\boldsymbol{s}}{}^{t}\right) \boldsymbol{\psi} = \lambda \left[\tilde{\boldsymbol{X}} \tilde{\boldsymbol{D}}_{\boldsymbol{sp}} \tilde{\boldsymbol{X}}^{t}\right] \boldsymbol{\psi} \ , \tag{2.13}$$

where $\tilde{\mathbf{X}}$ represents a matrix containing unlabeled samples, $\tilde{\mathbf{Z}}$ represents a matrix in which each row contains all the neighbors of individual pixels of interest, and $\tilde{\mathbf{W}}$ represents the affinity matrix for unlabeled samples (where each column of $\tilde{\mathbf{W}}$ represents the affinity of a pixel of interest with all its spatial-neighbors).

By multiplying $\tilde{\boldsymbol{X}}^t$ from the left and $\tilde{\boldsymbol{X}}$ from the right side of Equation (3.3.2), we obtain the following generalized eigenvalue problem.

$$\left(\sum_{s=1}^{n} \tilde{\boldsymbol{K}}_{\boldsymbol{Z}_s} \tilde{\boldsymbol{W}}_s \tilde{\boldsymbol{K}}_{\boldsymbol{X}_s}\right) \boldsymbol{\psi} = \lambda \left[\tilde{\boldsymbol{K}} \boldsymbol{D}_{\boldsymbol{sp}} \tilde{\boldsymbol{K}}\right] \boldsymbol{\psi} \ , \tag{2.14}$$

where $\tilde{\boldsymbol{K}}$ is a symmetric kernel matrix between elements of $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{X}}$; $\tilde{\boldsymbol{K}}_{\boldsymbol{X}_s}$ represents the kernel matrix between elements of $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{X}}_{\boldsymbol{s}}$; and $\tilde{\boldsymbol{K}}_{\boldsymbol{Z}_s}$ represents the kernel matrix between elements of $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Z}}_{\boldsymbol{s}}$. Here $\tilde{K}_{ij} = \kappa(\tilde{x}_i, \tilde{x}_j) = \langle \tilde{x}_i, \tilde{x}_j \rangle$ represents a simple linear kernel, although it can be replaced with any valid (nonlinear) Mercer kernel. A commonly used non-linear kernel function is the Gaussian radial basis function (RBF) which is defined as: $\kappa(\tilde{x}_i, \tilde{x}_j) = \exp\Big(-\frac{\|\tilde{x}_i - \tilde{x}_j\|^2}{2\sigma^2}\Big)$, where σ is a free parameter. Similar to SLSPP, the projection matrix or the eigenvectors corresponding to the r largest eigenvalues are found.

2.3.2 Obtaining optimal Spatial Neighbors using Superpixels

Entropy Rate Superpixels: Superpixel Segmentation is an important module for many Computer Vision applications as object recognition [46], image segmentation [47, 48] and single view 3-D reconstruction [49, 50]. A superpixel is commonly defined as a perceptually uniform region in the image. The goal of designing the optimization function for superpixel segmentation usually considers the following features: (1) Every superpixel should overlap with only one object; (2) The set of superpixel boundaries should be a superset of object boundaries; (3) The mapping from pixels to superpixels should not reduce the achievable performance of the intended application; (4) The above properties should be obtained with as few superpixels as possible. The Entropy Rate Superpixel Segmentation approach was proposed in [43], to represent the superpixel segmentation problem as an optimization problem on graph topology. The objective function is based on the entropy rate of a random walk in a graph.

The objective function of the ER-Superpixel algorithm contains two terms: (1) The Entropy Rate of random walk on a graph (which favors the formulation of compact and homogeneous clusters), and (2) A balancing term (which favors clusters of similar sizes). An efficient solution with a bound on the optimality of the solution was derived in the mentioned chapter.

The Entropy Rate superpixel [43] algorithm was modified to over-segment hyperspectral images instead of RGB images in [44]. We use superpixels generated by this modified version of the Entropy Rate superpixel algorithm to define the spatial neighorhood for each pixel.

Merging spectrally similar superpixels: Oversegmentation generates very small sized superpixels as some spatially adjacent pixels belonging to the same class gets segmented to multiple superpixels. From Equations 2.10 and 2.11, we find that very small sized superpixels would negatively impact the quality of the subspace projection as they would force the angular outer product matrix $\boldsymbol{M}$ in SLSPP to be calculated using a very small number of neighboring pixels (the points belonging to the same superpixel). In other words the number of samples $\{x_k, k \in \Omega_i\}$ belonging to the spatial neighborhood of a training sample x_i would be very small for very small sized superpixels. To negate this effect we merge the very small superpixels with spatially neighboring superpixels which have similar spectral angles, to form larger superpixels. All the applications using superpixels from hereon utilize these modified merged superpixels as describe above.

2.4 Sparse Representation-Based Classification using Orthogonal Matching Pursuit

2.4.1 Simultaneous Orthogonal Matching Pursuit (SOMP) Classifier

The SOMP [51] classifier is a sparse representation based classification method using the orthogonal matching pursuit algorithm, as shown in *Equations 2.15 and 2.16*. In addition to utilizing the class label of the pixel of interest it utilizes the samples surrounding a particular pixel of interest in order to classify that particular pixel. Thus this type of classifier is specially suited to explore the information contained in the neighboring pixels surrounding a pixel of interest while making a decision during classification.

The classification method employed after feature extraction using the SLSPP algorithm and its variants is SOMP based sparse representation classifier. SLSPP and its other derivatives as proposed in this chapter minimizes the angular distance between the spatially-neighboring points belonging to the same angular neighborhood and preserves the angular distance between the pixels belonging to different angular neighborhoods in the projected lower dimensional subspace. This implies that spatial neighbors in the projected lower dimensional subspace are more likely to be spectrally similar pixels belonging to the same spatial neighborhood in the original space and by using the SOMP classifier we can exploit the local neighborhood structures very efficiently. We need a spatial-classifier as SOMP at the back-end so as to utilize the spatial information preserved by our proposed dimension reduction algorithm, from the original space, in the reduced feature subspace. Any pixel-wise classifier would disregard the spatial information between the samples in the feature subspace and result in poor classification accuracies in comparison to spatial classifiers. Also since

we are proposing a feature reduction algorithm, we keep the back-end classifier to be the same (SOMP) for all the other feature reduction algorithms, during comparison. This was done in order to make the comparisons fair.

The extracted features generated using Algorithm 5 are used for training the SOMP classifier. Assume $\boldsymbol{S}$ to be the set of training samples, $\boldsymbol{A}$ to be the set of test samples - obtained from the group of spatial neighbors provided to the SOMP algorithm, and $\boldsymbol{C}$ to be the coefficient matrix learned through SOMP [51]. The class residuals for each class (Equation 2.15) are calculated using the SOMP algorithm as described in [51], and class-labels (Equation 2.16) are calculated by selecting the class corresponding to the minimal residual, as in traditional Sparse Representation based Classifier (SRC). Here c is the number of classes and δ_k is an indicator activating entries corresponding to the k'th class in the coefficient matrix. The Equation for SOMP is given as follows,

$$\mathbf{r}_k(\mathbf{S}) = \|\mathbf{S} - \mathbf{A}\delta_k(\mathbf{C})\|_2, \quad k = 1, 2, \ldots, c \ , \text{ and} \tag{2.15}$$

$$\omega = \underset{k=1,2,\ldots,c}{\operatorname{argmin}} \left(\mathbf{r}_k(\mathbf{S})\right) \ . \tag{2.16}$$

2.5 Experimental Settings and Results

2.5.1 Hyperspectral Datasets

We validate our proposed methods on two well known datasets: (1) The University of Pavia dataset and the (2) University of Houston dataset.

Grid Search Technique: We divide our datasets into three different parts with no overlap between the samples (samples are randomly selected) - (1) The training subset set, (2) the validation subset, and (3) the testing subset. First we tune our

model to find the free parameters using only the validation subset. Then we train the model using the training dataset and test using the unseen testing dataset, in order to evaluate the performance of the algorithms.

University of Pavia Data

An image covering the University of Pavia in Italy, captured using the Reflective Optics System Imaging (ROSIS) sensor [52], was the first hyperspectral dataset to be captured. It has 103 spectral bands ranging from 430 nm to 860 nm containing 9 classes of interest. It has a spatial coverage of 610×340 pixels and a spatial resolution of 1.3 m.

University of Houston Data

An image of the University of Houston and the neighboring urban area was captured using the ITRES-CASI (Compact Airbone Spectrographic Imager) 1500 hyperspectral imager. It has 144 spectral bands spanning the visible and near-infrared spectrum from 380 nm to 1050 nm. It has a spatial coverage of 349×1905 pixels, with a spatial resolution of 2.5 m. Fifteen different classes of interest were identified in the data.

2.5.2 Experimental Setup

The algorithms proposed in this chapter are shown to be better than other state of the art feature extraction methods in terms of learning effective subspaces. We compare the proposed algorithms with a wide variety of other traditional and state-of-the-art feature extraction algorithms — Principal Component Analysis (PCA) [53], Local Fisher Discriminant Analysis (LFDA) [54], Locality Preserving Projections (LPP) [55], Local Angular Discriminant Analysis (LADA) [42] and their kernel variants, Semi-Supervised LFDA (SELF) [56]; and Spatial-Spectral- Superpixel based multi-kernel SVM (SC-MK) [57].

Comparison with simple Fixed size Rectangular Windows: Using rectangular windows as spatial neighborhoods does not improve the accuracies to the extent when superpixels are used as spatial neighborhoods. This is understandable, as rectangular windows might contain samples belonging to multiple classes, which can be categorically avoided by using superpixels (which are designed for the purpose of segmenting the image to generate spectrally similar spatial-neighborhoods).

2.5.3 Results and Analysis

Tables 3.2 and 5.3 show the overall classification accuracies as a function of the number of training samples for the University of Pavia and the University of Houston datasets, respectively. 100 samples per class were chosen for testing. We notice for all cases, the overall classification accuracies improve significantly when our proposed dimension reduction approaches are used instead of the baseline or traditional state-of-the-art methods.

2.6 Conclusion

In this work we propose an unsupervised approach to utilize the unlabeled samples during feature extraction and also propose a general method which can be applied to utilize the spatial information by embedding the spectrally similar spatial neighborhoods using small fixed sized rectangular windows or superpixels, for angular discriminant based spatial-spectral feature extraction algorithms such as LSPP. We also kernelize our proposed linear algorithm in order to extract features from data samples which are governed by non-linear decision surfaces. We show that embedding the spatial neighborhoods (as quantified by superpixels) during the process of feature extraction significantly improves the classification accuracies by applying the neighborhood embedding algorithms to LSPP, and we also show that embedding spatial

Table 2.1: Overall accuracies (%) for the University of Pavia data.

Algorithm / N	***10***	***20***	***30***	***40***	***50***
PCA	66.02±2.6	69.21±3.1	71.23±2	72.62±1.2	73.46±2.7
KPCA	67.86±3.5	71.79±2.2	72.48±1.8	75.60±1.7	73.36±2
LFDA	55.44±3	68.17±3	72.27±1.8	73.89±1.3	73.73±2.2
KLFDA	81.34±2.2	86.21±2.3	90.14±2.2	91.58±1.6	92.11±1.2
LPP	36.38±2.5	67.68±2.5	74.48±2.2	78.6±2.2	79.79±2.1
LADA	74.54±2.5	77.81±2.4	79.18±2.5	79.14±1.7	81.12±1.9
KLADA	79.92±3.2	83.81±2.5	86.36±1.4	85.38±1.1	87±1.6
SELF	72.42±3.09	75.74±2.8	78.51±3.0	82.13±1.9	81.92±2.4
SC-MK	75±4.1	83.72±1.2	86.95±2.2	90.2±1.7	91.27±1.3
LSPP.	74.01±3	78.12±2.5	79.39±1.2	79.43±1.7	80.8±2.2
SLSPP-rect.	78.27±1.8	80.04±2.2	81.76±1.8	79.86±1.4	81.46±2.2
KSLSPP-rect. (proposed)	86.58±2.7	91.47±1.2	92.84±1.6	93.16±0.8	93.62±1.1
SLSPP-sp. (proposed)	**80.92±1.5**	**83.07±2.4**	**84.5±1.4**	**84.32±1.9**	**85.37±1.6**
KSLSPP-sp. (proposed)	**88.41±1.3**	**92.21±1.3**	**94.26±0.6**	**94.60±0.8**	**94.69±0.7**

information using superpixels produces more robust features compared to embedding spatial information using fixed sized rectangular windows. This makes sense intuitively as superpixels are known to extract spectrally homogeneous spatial neighbors by over-segmenting the image, while fixed sized rectangular windows may suffer from impure spectral pixels, due to the presence of pixels from multiple classes in one rectangular window. We also clearly show the benefit of kernelizing our proposed algorithms compared to the corresponding linear variants. We show that our proposed methods are able to extract more robust features which can train the back-end classifier in a more effective manner and produce higher classification accuracies than other state of the art baseline feature extraction or dimension reduction approaches.

Table 2.2: Overall accuracies (%) for the University of Houston data.

Algorithm / N	***10***	***20***	***30***	***40***	***50***
PCA	65.81±2.5	69.10±2	69.95±1.7	71.56±1.8	72.39±1.9
KPCA	64.59±1.7	68.95±1.7	69.85±2.1	71.34±1	72.52±1.6
LFDA	57.37±5.4	74.13±3	76.51±2.6	78.53±1.3	78.9±2.5
KLFDA	78.75±2.4	82.17±2.5	83.79±2.3	85.72±2.7	84.55±2.2
LPP	50.87±2.8	75.61±2	81.35±2.1	85.41±1.7	88.65±2.6
KLPP	79.87±4.2	86.62±1.2	90.53±1.4	90.11±1.4	92.02±1.6
LADA	83.37±2.6	88.14±1.9	90.10±1.5	91.93±1.7	92.70±1.3
KLADA	83.89±2.2	90.77±1.5	92.69±0.7	94.91±1.4	96.19±0.8
SELF	74.47±2.8	78.93±2.3	79.5±2.4	80.47±1.8	82.17±1.7
SC-MK	86.26±1.9	91.32±1.7	93.56±1.0	94.6±0.9	95.18±1.3
LSPP.	82.39±1.5	87.75±1.9	90.71±1.2	92.75±0.8	93.94±1.3
SLSPP-rect.	88.49±1.3	92.89±1.2	94.15±1	95.49±0.7	96.25±0.7
KSLSPP-rect. (proposed)	88.23±1.3	93.45±1.2	95.47±0.8	97.07±1	97.52±0.7
SLSPP-sp. (proposed)	**89.03±2**	**94.14±1.1**	**95.7±0.8**	**96.96±0.8**	**97.67±0.7**
KSLSPP-sp. (proposed)	**89.20±0.6**	**94.67±1.2**	**96.58±0.7**	**98.47±0.5**	**98.67±0.4**

Chapter 3

Semisupervised Spatial-Spectral Angular Discriminant Analysis for Hyperspectral Image Classification

3.1 Introduction

Hyperspectral images are typically captured at a very large number of wavelengths in the visible, near infrared and short-wave infrared region of the electromagnetic spectrum. Some information added to the image by many of these wavelengths (channels) is redundant and hence feature extraction is an important pre-processing step prior to classification. This is done in order to decrease the computational burden as well as train the classifier more effectively using a small number of available labeled training samples. Over the last two decades many algorithms have been proposed to extract useful features from images. Various feature extraction or subspace learning algorithms have been used in the literature as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) [53] and their variants [58]. Various manifold learning algorithms, such as Local Linear Embedding (LLE) [59], ISOMAP

[60], Laplacian Eigenmap [61], Locality Preserving Projection (LPP) [55] and Local Fisher Discriminant Analysis (LFDA) [54] were designed to preserve the local structure in the resulting embeddings by the nearest neighbors of every point on the manifold. It is well known [62] that hyperspectral data also lies on a low-dimensional manifold embedded in a high-dimensional space. Traditional dimension reduction algorithms utilize the Euclidean distance information for finding the subspace projection. Different from most traditional feature extraction algorithms, Angular Discriminant Analysis [42] was proposed in order to separate between-class samples and reduce the distance between the within class-samples in an angular sense. Local Angular Discriminant Analysis (LADA), which preserves the local structures in the resulting embeddings for ADA was also proposed.

Spatial-spectral algorithms have been used widely in the field of hyperspectral imaging with the purpose of generating robust features which utilize both the spatial context as well as the spectral information from the pixels in the image [31, 32, 33, 34]. The aim of this chapter is three fold— (1) To propose a method to capture the spatial contextual information for subspace learning, (2) To propose a semi-supervised algorithm that utilizes unlabeled samples for improving the subspaces learned and (3) To kernelize our algorithms to account for the presence of non-linearities in the underlying data topology. We compare our proposed methods with different state of the art methods and show that our algorithms are able to produce better features which train the back-end classifier in a more effective manner. As we showed in our previous work with LADA, LADA subspaces are beneficial to classifiers that leverage sparsity [63, 51, 64]. Hence, at the backend, we use a sparse representation classifier based on Simultaneous Orthogonal Matching Pursuit (SOMP) [51] that uses spatial information coupled with a sparse representation classifier.

The outline of this chapter is as follows. In Section 1 we briefly introduce the problem and provide the background. Section 2 briefly describes the LADA algorithm

from [42]. Section 3 proposes the semi-supervised spatial-spectral subspace learning algorithm. Section 4 provides a brief description of the SOMP classifier. Section 5 describes the datasets, experimental setup and results. Section 6 concludes this chapter.

3.2 Related Work: Local Angular Discriminant Analysis: LADA

LADA seeks to find an "optimal" subspace where the ratio of the angular distance between the within-class samples and the between-class samples gets minimized. Let $\tilde{x}_i \in \mathbb{R}^d$ be the normalized i-th training sample and $T \in \mathbb{R}^{d \times r}$ be the $d \times r$ projection matrix, where r is the reduced dimensionality. Let l denote the individual class labels and n_l denote the number of samples belonging to that class. Then the optimization function of LADA can be reduced to a generalized eigenvalue problem as

$$\boldsymbol{T}_{LADA} \approx \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\operatorname{argmin}} \left[\left(\boldsymbol{T}^t \boldsymbol{O}^{(\mathrm{lw})} \boldsymbol{T} \right)^{-1} \left(\boldsymbol{T}^t \boldsymbol{O}^{(\mathrm{lb})} \boldsymbol{T} \right) \right] , \tag{3.1}$$

Where the within class and between class angular scatter matrices $O^{(lw)}$ and $O^{(lb)}$ respectively are defined as

$$\boldsymbol{O}^{(lw)} = \sum_{i,j=1}^{n} \tilde{\boldsymbol{W}}_{ij}^{(lw)} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_j^t, \text{ and} \tag{3.2}$$

$$\boldsymbol{O}^{(lb)} = \sum_{i,j=1}^{n} \tilde{\boldsymbol{W}}_{ij}^{(lb)} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_j^t , \tag{3.3}$$

where the normalized weight matrices are defined as

$$\tilde{\boldsymbol{W}}_{ij}^{(\mathrm{lw})} = \begin{cases} \tilde{\boldsymbol{W}}_{ij}/n_l, & \text{if } y_i, y_j = l, \\ 0, & \text{if } y_i \neq y_j, \end{cases} \tag{3.4}$$

$$\tilde{\boldsymbol{W}}_{ij}^{(\mathrm{lb})} = \begin{cases} \tilde{\boldsymbol{W}}_{ij}(1/n - 1/n_l), & \text{if } y_i, y_j = l, \\ 1/n, & \text{if } y_i \neq y_j. \end{cases} \tag{3.5}$$

The normalized affinity $\tilde{\boldsymbol{W}}_{ij} \in [0, 1]$ between $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ is defined as

$$\tilde{W}_{ij} = \exp\left(-\frac{\|\tilde{x}_i - \tilde{x}_j\|^2}{\sigma}\right), \tag{3.6}$$

where σ is the parameter in the normalized heat kernel. The projection matrix T are the eigenvectors corresponding to the r smallest eigenvalues.

3.3 Proposed Work: Semi-supervised Spatial LADA: SSLADA

3.3.1 Motivation

For hyperspectral images it is well known that providing a spatial context to spectral classification algorithms is helpful [34, 65, 66, 67]. This is based on the fact that the spatially neighboring pixels in images belong to the same classes of interest, hence, they are spectrally similar. In order to utilize the spatial neighbors of a pixel of interest in the lower dimensional subspace we preserve the spatial neighbors of the pixels of interest for our algorithms. With this is in mind we propose a spatial-spectral feature extraction method that builds upon pixel-level supervised LADA.

Figure 4.2 shows a simple block diagram representation of the proposed algorithms

as described in this section. The yellow region in the figure below represents the spatial neighborhood for the red pixel. As shown in the figure, in this chapter: (1) We propose a semi-supervised subspace learning algorithm which leverages both labeled and unlabeled samples to learn the embedding, (2) We add spatial context to the resulting embedding and (3) We kernelize the resulting projections to implement them in the Reproducible Kernel Hilbert Space (RKHS).

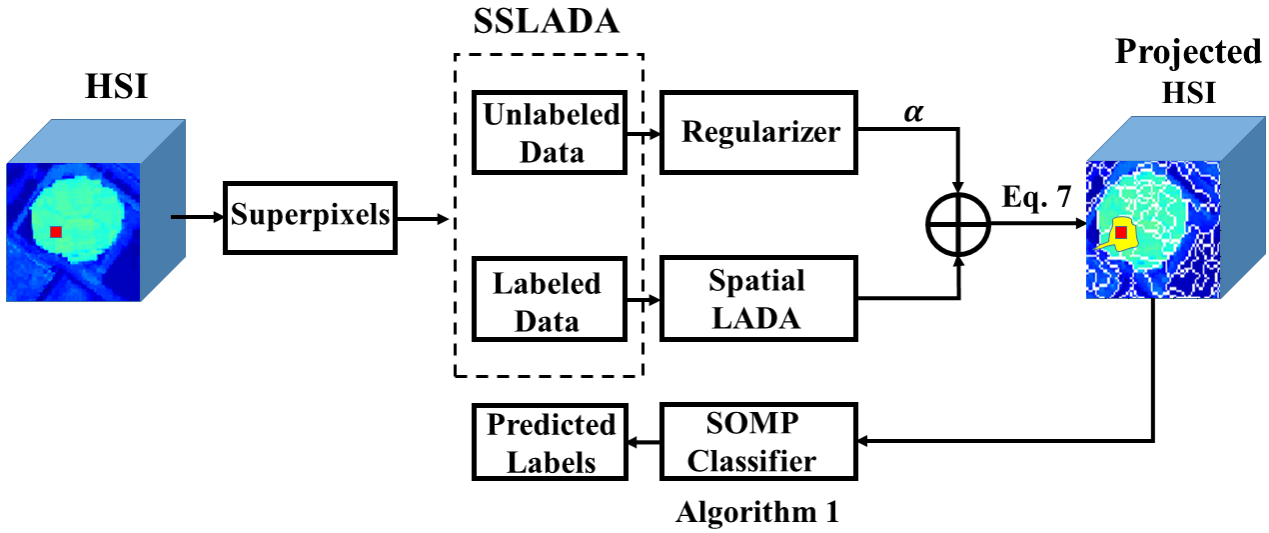


Figure 3.1: Block diagram representation of the proposed architecture

Figure 3.2 provides an overview of the proposed algorithms described in this section. Upper half of the figure shows a part of the University of Houston image. The red boundaries overlayed on the image show superpixel segmentation boundaries. The white pixels represent one particular class from the image. All pixels which are located inside one superpixel represent a spatial neigborhood. We visually depict how spatially constraining the within-class angular scatter matrix effects the angular distance between the projected points belonging to the same class (depicted by the white pixels) on an l_2-normalized hyperspehere, through this figure. For the spatial-spectral algorithm the within-class angular scatter matrix is calculated using only the spatial neighbors which belong to the same class as the pixel of interest but for the spectral-only algorithms the samples do not have to belong to the same spatial neigh-

borhood as the pixel of interest, they only have to belong to the same class. The black points are the pixels of interest and the blue points surrounding them are the spatial neighborhoods extracted from superpixels. Since each point is best represented by its spatial neighbors which belong to the same class, we propose that in the normalized hypersphere for spatial-spectral algorithm, the within-class samples will come closer to a much higher degree than for the purely spectral algorithms as shown in Figure 3.2. Our results in later sections visualizing the samples (from University of Houston Image) projected onto the hyperspheres clearly demonstrate that this claim holds.

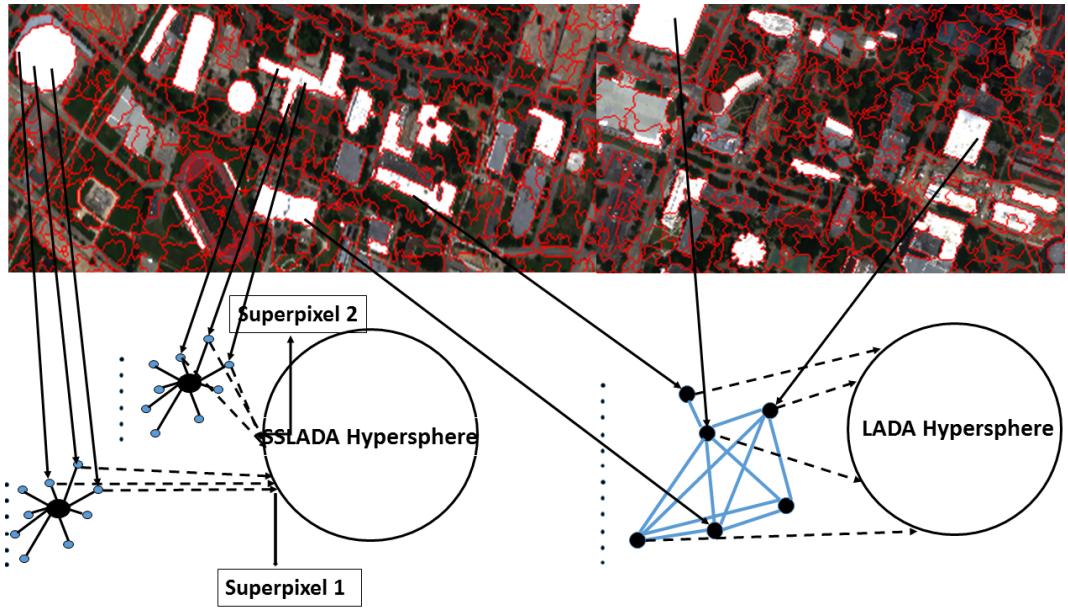


Figure 3.2: Neighborhood embedding of a point for spatial-spectral (left) and purely spectral (right) feature extraction algorithms (only for within-class affinities)

LADA is a supervised feature extraction algorithm which requires class labels of individual pixels during implementation. Traditionally, hyperspectral images contain few labeled samples as it is difficult to obtain a large number of samples with class labels. Thus the aim to find a projection matrix so as to minimize the ratio of $O^{(lb)}$ and $O^{(lt)}$, will not work when the number of labeled training samples are not sufficient, as overfitting will occur. A usual way to prevent overfitting is to impose a regularizer

[68]. Based on a similar idea, a model of Linear Discriminant Analysis (LDA) with the popular Tikhonov regularizer, referred to as Regularized Discriminant Analysis (RDA), has been proposed in [69]. The Semi-supervised Discriminant Analysis (SDA) algorithm based on another regularized version of LDA has been proposed in [58]. Although the number of labeled samples available for the purpose of training and testing is low, hyperspectral images generally contain a large number of unlabeled pixels present in the image. For this reason semi-supervised approaches which can utilize the labeled data to minimize the separability between pixels belonging to the same class (and / or maximize the separability between pixels belonging to different classes) and the unlabeled data to cluster the pixels depending upon the inherent angular or euclidean structure of the data, are very popular for hyperspectral image analysis [70, 6, 71]. The premise of our approach is that in the resulting subspace, spatial neighbors will be clustered together, thus preserving spatial-spectral affinities. Based on this idea, we propose a semi-supervised approach and embed the spatial neighborhood to the individual pixels of interest in our Semisupervised-Spatial LADA (SSLADA) algorithm.

3.3.2 Algorithm

Algorithm 5 briefly describes the flow of processes used by us to implement our proposed algorithms. Entropy Rate (ER) superpixel generation is used to oversegment the image. In Algorithm 5 the entropy rate term $\mathcal{H}(A)$ increases with the addition of any edge to the set A, but the increase is larger when selecting edges that form compact and homogeneous clusters as described in [43]. The balancing term $\mathcal{B}(A)$ helps to generate clusters of similar sizes when the number of clusters are fixed. Thus, the algorithm tends to find clusters which are compact, homogeneous and of similar sizes in nature. Since the objective function in [43] increases monotonically, the number of connected components (clusters) in the graph (N_A) is exactly found to

be equal to the number of desired superpixels ($\mathcal{N}$), due to the additional constraints.

Algorithm 1: *Pseudo code of the proposed feature extraction*

Input:

• Image: $I \in \mathbb{R}^{rw \times cl \times d}$

• Ground Truth: $Y \in \mathbb{I}^{rw \times cl}$

• Number of superpixels to be generated: $\mathcal{N}$

{*Superpixel Segmentation*}

• Generate compact, homogeneous and balanced Entropy-Rate superpixels: $X_i = \{x_j\}_{j=1}^{n_i}$ for $i = 1, 2, ..., \mathcal{N}$ (where i : superpixel index and j : pixel index in the i^{th} superpixel) from [43, 44]:

$$\textbf{max}_A \mathcal{H}(A) + \lambda \mathcal{B}(A)$$

$$\textbf{\textit{s.t.}}\ A \subseteq E,\ N_A \geq \mathcal{N} \text{ and } \lambda \geq 0$$

where A: selected edge set for segmenting the graph, $\mathcal{H}(A)$: Entropy Rate term, $\mathcal{B}(A)$: balancing term, λ: weight of the balancing term, E: Edges in the graph and N_A: number of connected components in the graph

---

{*Spectral-angle based Merging*}

• Generate merged superpixel corresponding to or encompassing each pixel:

for all $i \in 1, 2, ..., \mathcal{N}$ **do**
  for all $j \in 1, 2, ...,$ neighbors of superpixel i **do**
    $\theta = cos^{-1}\left[E[X_i] \odot E[X_j]^T / \left(||E[X_i]||\ ||E[X_j]||\right)\right]$
    if $\theta \leq \delta$ (minimum angle) **do**
      $\{X_i\} = \{X_i \cup X_j\}$
    end if
  end for
end for

{*Feature Extraction*}

• Extract training samples: $\{\tilde{x}\}_{i=1}^{n} \in \mathbb{R}^d$ - from I and Y. n: number of samples

• Compute O^{lb} and O^{lt} using labeled data from X; Compute $\tilde{W}_u$ using unlabeled data from $\tilde{X}_u$ as defined in Equations 3.9, 3.10 and 3.7 respectively.

• Form an objective function to minimize the ratio of within-class to between-class angular scatter by utilizing both the labeled and unlabeled data and using Equation 3.7:

$$\boldsymbol{T}_{SSLADA}$$
$$= \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\text{argmin}} \left[\frac{tr(\boldsymbol{T}^t \boldsymbol{O}^{(\text{lb})} \boldsymbol{T})}{tr(\boldsymbol{T}^t (\boldsymbol{O}^{(\text{lt})} + \boldsymbol{\alpha} \tilde{\boldsymbol{X}}_{\boldsymbol{u}} \tilde{\boldsymbol{W}}_{\boldsymbol{u}} \tilde{\boldsymbol{X}}_{\boldsymbol{u}}^{\boldsymbol{t}}) \boldsymbol{T})} \right]$$
$$\approx \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\text{argmin}} \left(tr \left[\frac{(\boldsymbol{T}^t \boldsymbol{O}^{(\text{lb})} \boldsymbol{T})}{(\boldsymbol{T}^t (\boldsymbol{O}^{(\text{lt})} + \boldsymbol{\alpha} \tilde{\boldsymbol{X}}_{\boldsymbol{u}} \tilde{\boldsymbol{W}}_{\boldsymbol{u}} \tilde{\boldsymbol{X}}_{\boldsymbol{u}}^{\boldsymbol{t}}) \boldsymbol{T})} \right] \right)$$

Output:

• Training points after projection: $\{p\}_{i=1}^{n} \in \mathbb{R}^r$

• Projection Matrix: $T \in \mathbb{R}^{d \times r}$.

Linear SSLADA

The optimization problem for the proposed regularized Semisupervised-Spatial LADA can be written as in equation (3.7). Let $\tilde{x}_k, k \in \Omega_i$ be the l_2-normalized spatial neighborhood of labeled samples around a normalized labeled training sample $\tilde{x}_i$ and $T \in \mathbb{R}^{d \times r}$ be the $d \times r$ projection matrix, where r is the reduced dimensionality, then the objective function of SSLADA is reduced to

$$\boldsymbol{T}_{SSLADA} \approx \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\operatorname{argmin}} \left[(\boldsymbol{T}^t (\boldsymbol{O}^{(\mathrm{lt})} + \boldsymbol{\alpha} \tilde{\boldsymbol{X}}_{\boldsymbol{u}} \tilde{\boldsymbol{W}}_{\boldsymbol{u}} \tilde{\boldsymbol{X}}_{\boldsymbol{u}}^{\boldsymbol{t}}) \boldsymbol{T})^{-1} (\boldsymbol{T}^t \boldsymbol{O}^{(\mathrm{lb})} \boldsymbol{T}) \right] , \tag{3.7}$$

$$\boldsymbol{O}^{(\mathrm{lw})} = \sum_i \sum_{k \in \Omega_i} \tilde{\boldsymbol{W}}_{ik}^{(\mathrm{lw})} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_k^t , \tag{3.8}$$

$$\boldsymbol{O}^{(\mathrm{lb})} = \sum_{i,j=1}^{n} \tilde{\boldsymbol{W}}_{ij}^{(\mathrm{lb})} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_j^t , \tag{3.9}$$

$$\boldsymbol{O}^{(\mathrm{lt})} = \boldsymbol{O}^{(\mathrm{lw})} + \boldsymbol{O}^{(\mathrm{lb})} , \text{ and} \tag{3.10}$$

$$\boldsymbol{O}^{(\mathrm{lb})} \psi = \lambda (\boldsymbol{O}^{(\mathrm{lt})} + \boldsymbol{\alpha} \tilde{\boldsymbol{X}}_{\boldsymbol{u}} \tilde{\boldsymbol{W}}_{\boldsymbol{u}} \tilde{\boldsymbol{X}}_{\boldsymbol{u}}^{\boldsymbol{t}}) \psi , \tag{3.11}$$

where $\tilde{X}_u$ represents a matrix containing all unlabeled samples and $\tilde{W}_u$ represents the affinity matrix of all those unlabeled samples. The projection matrix T are the eigenvectors corresponding to the r smallest eigenvalues, obtained after solving Equation (3.11), by simple eigen-value decomposition. The objective function in equation (3.1) remains equivalent if the within class scatter matrix $O^{(lw)}$ is replaced by the total scatter matrix $O^{(lt)}$. We propose SSLADA governed by the objective function in equation

(3.7), based on this idea. Equation (3.7) uses unlabeled samples in addition to the labeled samples and the total scatter matrix instead of the within-class scatter matrix when compared to Equation (3.1). Use of the unlabeled samples improves the projection as shown by the results. Singularity of the within-class scatter matrix is a known problem of the LDA algorithm, which makes the ratio of the between-class scatter matrix and the within-class scatter matrix undefined or unbounded [72, 73, 74]. For our algorithm we replace the within-class scatter matrix with the total scatter matrix, as in [58, 75], so that the optimization function in Equation (3.1) is bounded when either of the between-class angular scatter matrix (O^{lb}) or within-class angular scatter matrix (O^{lw}) is non-zero. The regularizer term $\tilde{X}_u \tilde{W}_u \tilde{X}_u^t$ incorporates our prior knowledge about the underlying data distribution. When a set of unlabeled samples are available, our intention is to construct a regularizer which can incorporate the inherent angular structures present in the data and preserve those in the embedding. The term $T\tilde{X}_u \tilde{W}_u \tilde{X}_u^t T^{-1}$ in equation (3.7) portrays the unsupervised part of the algorithm, which helps to reduce the angular distance between the spatial neighbors in the lower dimensional projected subspace. In other words spatial neighbors in the higher dimensional subspace remain spatial neighbors in the projected subspace due to the presence of this term. This term in the objective function is natural because if two unlabeled data points have very low spectral angle difference between them in the projected subspace, implying a greater value of the term $T\tilde{X}_u \tilde{W}_u \tilde{X}_u^t T^{-1}$, then they are likely to belong to the same class. Moreover, data points lying on dense subgraphs in the angular space are also likely to belong to the same classes. Thus, the goal is to maximize this term by adding its scaled value to the within-class or total scatter matrix, in the projected lower dimensional subspace. All the other terms are based on the supervised part of the algorithm. The coefficient α, which controls the balance between supervised and unsupervised components of the algorithm, is determined by a grid search technique and its value is fixed to the value which results

in the highest classification accuracies.

Kernel SSLADA

Samples from different classes may not always be linearly separable in the original space due to the inherent non-linear structure of the data. For such instances the SSLADA algorithm will fail to find a subspace that can angularly separate the between class samples. Formulating SSLADA in a Reproducible Kernel Hilbert Space (RKHS) $\mathcal{H}$ will overcome this limitation.

By applying the *kernel trick* [45], SSLADA can be extended to its kernel variant.

Let n be the number of available samples and m be the number of neighbors for each of those samples. Then, the term $\left(\sum_i \sum_{k\in\Omega_i} \tilde{\boldsymbol{W}}_{ik}^{(\mathrm{lt})} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_k^t\right)$ can be simplified to $\sum_{s=1}^{n} \tilde{\boldsymbol{X}}_{\boldsymbol{s}} \tilde{\boldsymbol{W}}_{\boldsymbol{s}}^{\boldsymbol{lw}} \tilde{\boldsymbol{Z}}_{\boldsymbol{s}}^{\boldsymbol{t}}$ by using basic matrix algebra. Where:

$$\tilde{\boldsymbol{X}} = \begin{bmatrix} \tilde{X}_1 & \tilde{X}_2 & \tilde{X}_3 \dots & \tilde{X}_n \end{bmatrix},$$

$$\tilde{\boldsymbol{W}^{lw}} = \begin{bmatrix} \tilde{\boldsymbol{W}}_{\mathbf{1}} \\ \vdots \\ \tilde{\boldsymbol{W}}_{\boldsymbol{n}} \end{bmatrix} = \begin{bmatrix} \tilde{W}_{1,1} & \tilde{W}_{1,2} & \tilde{W}_{1,3} \dots & \tilde{W}_{1,m} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{W}_{n,1} & \tilde{W}_{n,2} & \tilde{W}_{n,3} \dots & \tilde{W}_{n,m} \end{bmatrix} \text{ and}$$

$$\tilde{\boldsymbol{Z}}^{\boldsymbol{t}} = \begin{bmatrix} \tilde{\boldsymbol{Z}}_{\mathbf{1}}^{\boldsymbol{t}} & \tilde{\boldsymbol{Z}}_{\mathbf{2}}^{\boldsymbol{t}} & \tilde{\boldsymbol{Z}}_{\mathbf{3}}^{\boldsymbol{t}} \dots & \tilde{\boldsymbol{Z}}_{\boldsymbol{n}}^{\boldsymbol{t}} \end{bmatrix},$$

$$= \begin{bmatrix} \tilde{X}_{1,1} & \tilde{X}_{2,1} & \tilde{X}_{3,1} \dots & \tilde{X}_{n,1} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{X}_{1,m} & \tilde{X}_{2,m} & \tilde{X}_{3,m} \dots & \tilde{X}_{n,m} \end{bmatrix}$$

Where $\tilde{W}_{i,j}$ is as defined in Equation 3.6.

$\tilde{X}$ represents a matrix containing all labeled samples, $\tilde{Z}$ represents a matrix in which each row contains all the neighbors of individual pixels of interest, $\tilde{X}_u$ represents a matrix containing all unlabeled samples and $\tilde{W}_u$ represents the affinity matrix for all unlabeled samples.

By multiplying $\tilde{\boldsymbol{X}}^t$ from the left and $\tilde{\boldsymbol{X}}$ from the right side of Equation (3.3.2), we obtain the following generalized eigenvalue problem.

$$\tilde{\boldsymbol{K}}\,\tilde{\boldsymbol{W}}^{(\mathrm{lb})}\tilde{\boldsymbol{K}}\boldsymbol{\psi} = \lambda\left[\left(\sum_{s=1}^{n}\tilde{\boldsymbol{K}}_{\boldsymbol{X}_s}\,\tilde{\boldsymbol{W}}_{\boldsymbol{s}}{}^{(\mathrm{lw})}\tilde{\boldsymbol{K}}_{\boldsymbol{Z}_s}\right)+\tilde{\boldsymbol{K}}\;\tilde{\boldsymbol{W}}^{(\mathrm{lb})}\tilde{\boldsymbol{K}}+\boldsymbol{\alpha}\tilde{\boldsymbol{K}}_{\boldsymbol{u}}\,\tilde{\boldsymbol{W}}_{\boldsymbol{u}}\,\tilde{\boldsymbol{K}}_{\boldsymbol{u}}\right]\boldsymbol{\psi} \quad (3.12)$$

where $\tilde{K}$ is a symmetric kernel matrix between elements of $\tilde{X}$ and $\tilde{X}$; $\tilde{K}_{X_s}$ represents the kernel matrix between elements of $\tilde{X}$ and $\tilde{X}_s$; $\tilde{K}_{Z_s}$ represents the kernel matrix between elements of $\tilde{X}$ and $\tilde{Z}_s$; and $\tilde{K_u}$ represents the kernel matrix between elements of $\tilde{X}$ and $\tilde{X}_u$. Here $\tilde{\boldsymbol{K}}_{ij} = \kappa(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) = \langle\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j\rangle$ represents a simple linear kernel, although it can be replaced with any valid (nonlinear) Mercer kernel. A commonly used non-linear kernel function is the Gaussian radial basis function (RBF) which is defined as

$$\kappa(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) = \exp\Big(-\frac{\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2}{2\sigma^2}\Big), \quad (3.13)$$

where σ is a free parameter.

Similar to SSLADA, the projection matrix or the eigenvectors corresponding to the r smallest eigenvalues are found.

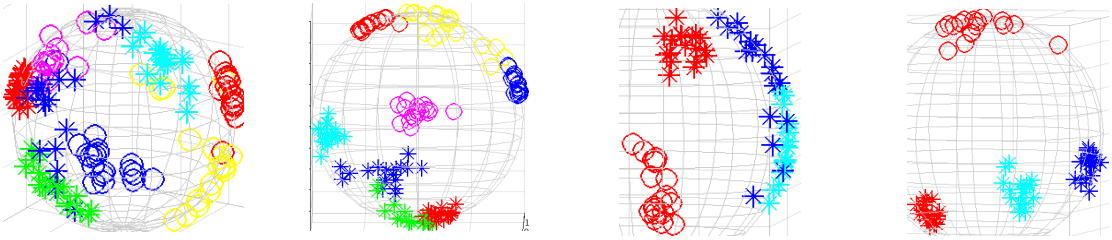
3.3.3 Visualization of Embeddings

The superpixels were obtained by implementing the Entropy Rate superpixel algorithm. The superpixels so obtained were used in designing the dimension reduction algorithm to find the projection matrix (only the components in the direction of the 3 smallest eigen-values after solving the optimization problem were retained). The samples from the original space were projected using this projection matrix and they were visualized in a hypersphere using our proposed algorithms (SSLADA-sp. and KSSLADA-sp.) and the nearest baseline algorithms (LADA and KLADA), for comparison. The points are shown in the hypersphere for the purpose of better visualization. As the pixels lie in a 3 dimensional space (corresponding to the three smallest eigenvalues resulting from their optimization functions as described in detail earlier), we plotted them on the hyperspheres to make visualization easier for the readers. The samples were projected onto a unit normalized sphere by using an l_2 norm on the data points.

SpecTIR Dataset: We performed the analysis with the SpecTIR Dataset. Figure 3.3 below shows 18 random samples per class chosen from the SpecTIR image in the projected subspaces.

From Figures 3.3a and 3.3b we notice that the within-class samples come closer to each other to a higher degree in the SSLADA-sp. projected subspace compared to the LADA projected subspace (samples belonging to the classes represented by the blue, yellow and magenta circles; green, blue, cyan and red crosses show this explicitly). We also notice that between class samples move farther apart (even get separated) from each other in the projected subspace of SSLADA-sp. than the subspace of LADA (samples belonging to the classes represented by the magenta circles, red, blue and cyan crosses; red, yellow circles and cyan crosses; blue circles, blue and green crosses, show this explicitly). We perform similar analysis with KLADA and KSSLADA and observe similar results. The samples represented by the cyan, red and blue crosses

come closer to each other for KSSLADA-sp. compared to KLADA. Also the samples represented by cyan, red and blue crosses move farther apart (even get separated) from each other for KSSLADA-sp. compared to KLADA. This is shown in Figures 3.3c and 3.3d.



(a) LADA (b) SSLADA-sp. (c) KLADA (d) KSSLADA-sp.

Figure 3.3: Samples from the SpecTIR image after projection by (a) LADA, (b) SSLADA-sp, (c) KLADA and (d) KSSLADA-sp.

The red, yellow, blue and magenta circles represent Sesbania (rattlebox), Sabal Mexicana (palm tree), Upland Grass and Phragmites Austrails respectively. The blue, green, cyan and red crosses represent Juncus Roemerianus, Batis maritima / Distichlis Spicata, Baccharis halimifolia and Distichlis Spicata respectively. This helps the classifier to distinguish between samples belonging to different classes and predict samples which belong to the same classes. We attribute these observations to the nature of our proposed algorithm as described in the previous sections. In short, we conclude that there is an overall benefit when we use our proposed feature extraction algorithms.

Next we describe the way using which we provide spatial context to our algorithms.

3.3.4 Obtaining optimal Spatial Neighbors using Superpixels

The Entropy Rate superpixel [43] algorithm was modified to over-segment hyperspectral images instead of RGB images in [44]. Initially we use superpixels generated

by this modified version of the Entropy Rate superpixel algorithm to define the spatial neighorhood for each pixel. We first generate a very large number of superpixels in order to make sure that class boundaries in the image are well respected and we do not have pixels belonging to multiple classes inside a particular superpixel.

Merging spectrally similar superpixels: We deliberately oversegment the image so that each superpixel contains samples from only one class. However, this generates very small sized superpixels as some spatially adjacent pixels belonging to one class get segmented to multiple superpixels. From Equation 3.8 we find that very small sized superpixels would negatively impact the quality of the subspace projection as they would force the within-class angular scatter matrix $O^{(lw)}$ in SSLADA to be calculated using a very small number of neighboring pixels (the points belonging to the same superpixel). In other words the number of samples $\{x_k, k \in \Omega_i\}$ belonging to the spatial neighborhood of a training sample x_i would be very small for very small sized superpixels. To negate this effect we merge the very small superpixels with spatially neighboring superpixels which have similar spectral angles. The upper half of Figure 3.2 shows a part of the University of Houston image after segmentation and merging. The superpixels are obtained in the same way as described in Section 2.3.2.

3.4 Sparse Representation-Based Classification using Orthogonal Matching Pursuit

3.4.1 SOMP Classifier

The simultaneous orthogonal matching pursuit is a sparse representation based classification method using the orthogonal matching pursuit algorithm. In addition to utilizing the class label of the pixel of interest it utilizes the samples surrounding a particular pixel of interest in order to classify that particular pixel. Thus this type of

classifier is specially suited to explore the information contained in the neighboring pixels surrounding a pixel of interest while making a decision during classification. Assume A_i contains spatial neighborhood samples (based on superpixels) around p_i (inclusive of p_i), S contains the spatial neighborhood samples (based on superpixels) around p_t (inclusive of p_t) and K is the sparsity level. Here, $A_i = \{p_k\}_{k\in\Omega_i}$, $S = \{p_k\}_{k\in\Omega_t}$, Ω_i : spatial neighborhood around the training pixel x_i and Ω_t : spatial neighborhood around the test pixel x_t. SOMP estimates the coefficient $\hat{C}$ based on the K mostly correlated training samples in A.

Classification via SOMP: The classification method employed after feature extraction using SSLADA algorithm and its variants is Simultaneous Orthogonal Matching Pursuit as described in algorithm 2. SSLADA and its other derivatives as proposed in this chapter minimizes the angular distance between the spatially-neighboring points belonging to the same class and maximizes the angular distance between the pixels belonging to different classes in the projected lower dimensional subspace. This implies that spatial neighbors in the projected lower dimensional subspace are more likely to be spectrally similar pixels belonging to the same spatial neighborhood in the original space and by using the SOMP classifier we can exploit the local neighborhood structures very efficiently. We need a spatial-classifier such as SOMP at the back-end so as to utilize the spatial information preserved by our proposed dimension reduction algorithm, from the original space, in the reduced feature subspace. Any pixel-wise classifier would disregard the spatial information between the samples in the feature subspace and result in poor classification accuracies in comparison to spatial classifiers. Also since we are proposing a feature reduction algorithm, we keep the back-end classifier to be the same (SOMP) for all the other feature reduction algorithms, during comparison. This was done in order to make the comparisons fair.

The extracted features generated using Algorithm 5 are used for training the

SOMP classifier as described next, in Algorithm 3. The performance of the feature extraction methods are evaluated by analyzing the classification accuracies.

Algorithm 2: *SOMP*

Input: Projected training data points $A = \{A_i\}_{i=1}^{n}$, projected test data S and row sparsity level K.

Initialize $R^0 = S$, $\Lambda^0 = \emptyset$, and the iteration counter $m = 1$.

while $m \leq K$ **do**

Update the support set $\Lambda^m = \Lambda^{m-1} \cup \lambda$ by solving

$$\lambda = \underset{i=1,2,\dots,n}{\operatorname{argmax}} \| A_i^t R^{m-1} \|_{2,1}.$$

Derive the coefficient matrix C^m based on

$$C^m = \left(A_{\Lambda^m}^t A_{\Lambda^m} \right)^{-1} A_{\Lambda^m}^t S$$

Update the residual matrix R^m

$$R^m = S - A_{\Lambda^m} C^m$$

$m \leftarrow m + 1$

end while

Calculate the coefficient matrix: $\hat{C} = C^{m-1}$.

Calculate residuals for each class

$$r_k(S) = \| S - A\delta_k(\hat{C}) \|_2, \quad k = 1, 2, \dots, c$$

Determine the class label of S based on

$$\omega = \underset{k=1,2,\dots,c}{\operatorname{argmin}} (r_k(S)).$$

Output: A class label ω.

3.5 Experimental Settings and Results

3.5.1 Hyperspectral Datasets

We validate the proposed algorithms by applying them on two well known hyperspectral datasets—(1) The University of Houston dataset (2) The University of Pavia dataset and a dataset published by our lab to study the coastal Wetlands of Galveston, Texas (3) The SpecTIR dataset. All parameters from the datasets were obtained by using a cross-validation or grid search technique which resulted in the maximum accuracy values.

Grid Search Technique: The labeled data was randomly divided into 3 subsets: Training set, validation set and the testing set. The training data was used to train the algorithms and validation set was used to tune the free parameters to values which resulted in the best classification accuracies. Testing results as reported in this chapter were acquired using the testing data.

University of Houston Data

The dataset covering the University of Houston campus and the neighboring urban area was captured using an ITRES-CASI (Compact Airbone Spectrographic Imager) 1500 hyperspectral imager. It covers 144 spectral bands spanning the visible and near-infrared spectrum from 380 nm to 1050 nm. The image has a spatial size of 349×1905 with a spatial resolution of 2.5 m. Fifteen different classes of interest were identified in the data. Free parameters for the algorithm were learned using a grid search technique.

University of Pavia Data

The first Hyperspectral dataset covering the University of Pavia in Italy was captured using the Reflective Optics System Imaging (ROSIS) sensor [52]. The image

contains 103 spectral bands spanning from 430 nm to 860 nm with 9 classes of interest. It has a spatial coverage of 610 × 340 pixels with a spatial resolution of 1.3 m. Free parameters for the algorithm were learned using a grid search technique.

SpecTIR Airborne Data

The data was acquired in a study conducted in the coastal wetland of Galveston, Texas. Wetland vegetation, which is a crucial part of the wetland ecosystem, is found to change dramatically with respect to both coverage and species distribution. Marshes in Mission-Aransas estuary used to be dominated by smooth cordgrass, but are now mostly covered by black mangroves. Such transitions in coastal vegetation are likely to influence the quality of coastal wetlands for supporting shrimps, fishes, birds and change the ability of the coastal habitats to buffer wind and wave energy. Therefore, mapping the wetland species would help us to better manage the endangered wetland ecosystems.

An airborne hyperspectral image was captured using the ProSpecTIR VS sensor on August 14, 2015. The image is captured at 360 wavelength bands ranging from 400 nm to 2450 nm, at a spectral resolution of 5 nm. The radiance data were radiometrically and spectrally calibrated before they were converted to the reflectance data using the ATCOR 4 software. The output reflectance data of multiple flight lines were eventually geo-corrected and mosaiced as an image with spatial coverage of 3462×5037 pixels at a 1 m spatial resolution.

Since labeling samples through photo-interpretation is not applicable in this situation, a field survey was made on September 16, 2016. ***Upland grass, St. Augustine grass, Sesbania / rattlebox, Upland tree, Phragmites austrails, Sabal mexicana / palm tree, Spartina alterniflora, Juncus roemerianus, Batis maritima / Distichlis spicata, Distichlis spicata, Baccharis halimifolia*** and ***Avicennia germinans / black mangrove*** were identified to be the 12 dis-

tinct classes. The dataset contains a total of 7219 labeled samples. Free parameters for the algorithm were learned using a grid search technique.

3.5.2 Experimental Setup

The algorithms proposed in this chapter are shown to be better than other state of the art feature extraction methods in terms of learning effective subspaces. We compare the proposed algorithms with other feature extraction algorithms — Principal Component Analysis (PCA) [53], Local Fisher Discriminant Analysis (LFDA) [54], Locality Preserving Projections (LPP) [55], Local Angular Discriminant Analysis (LADA) [42] and their kernel variants. We also compare our algorithms to a semi-supervised feature extraction algorithm — SELF [56] and a spatial-spectral classification algorithm based on entropy-rate superpixels and multiple-kernel SVM's — SC-MK [57]. The algorithms are evaluated as a function of the number of training samples. For every case, the training samples are used to train the feature extraction algorithm in order to extract the relevant features, using which the classification algorithm is trained and used to predict the class labels of the test samples. The training and test samples were generated using a repeated random subsampling method. The number of labeled training samples for each case, N, is mentioned in the columns of Table 5.3, Table 3.2 and Table 3.3. The number of unlabeled training samples used for each case for semi-supervised methods is fixed to be equal to the number of labeled training samples. The number of test samples was fixed to 100 when the University of Houston and University of Pavia datasets were used and for the SpecTIR dataset all the available labeled samples excluding those used for training were used as test samples. All the reported accuracies are the average accuracies of 10 repeated random subsampling results. The notation LADA-SOMP means that LADA is used as the feature extraction algorithm and SOMP is used as the back-end classifier. All the other notations have similar meanings.

Comparison with simple Fixed size Rectangular Windows: We use fixed sized rectangular windows around each pixel of interest to define its spatial neighborhood in order to observe if very basic spatial neighborhoods (simple rectangular windows) can improve the subspace projections as well. We use rectangular windows which are of sizes equal to the mean size of all superpixels after merging, in order to keep our experimental designs equivalent. Our results show that simple rectangular windows also provide an improvement in accuracy over LADA but the gain is not as high as compared to the case when superpixels are used. This happens because even for small sized rectangular windows there is a chance of generating rectangular windows which contain pixels belonging to multiple classes. This introduces inaccuracy in the calculation of the within-class outer product scatter matrix ($O^{(lw)}$) in our algorithms, which, assume that all pixels inside the generated patches belong to the same class.

For the case of superpixels this problem (mixing of pixels belonging to different classes) can be avoided to a significant degree as superpixels are able to generate patches in the image which contain spectrally similar pixels. This idea gets reflected in the results which show that the superpixel based SSLADA leads to higher classification accuracies compared to the rectangular window based SSLADA. It is interesting to note that even though the rectangular window based SSLADA performs slightly poorly compared to the superpixel based SSLADA, it still outperforms LADA in terms of generating features which can train the classifier more effectively. Thus by providing a spatial context (albeit in a very simplistic manner) to the spectral angle based feature extraction method LADA, we can considerably improve the quality of subspace projection.

3.5.3 Results and Analysis

The best linear algorithm among all linear algorithms and the best kernel algo-

rithm among all kernel algorithms are shown using bold black symbols.

University of Houston Dataset: Table 5.3 shows the overall classification accuracies as a function of the number of training samples for the University of Houston dataset. Table 3.4 shows the class specific accuracies when 10 samples are used for training and 100 samples are used for testing for the University of Houston dataset. We notice from Table 5.3 that for all cases, the overall classification accuracies improve when our proposed dimension reduction approaches are used instead of the baseline methods. We notice from Table 3.4 that for the classes which are harder to classify as — *Residential area, Roads, Parking lot 1 and Parking lot 2* — our algorithms significantly outperform the other baseline algorithms.

University of Pavia Dataset: Table 3.2 shows the overall classification accuracies as a function of the number of training samples for the University of Pavia dataset. Table 3.5 shows the class specific accuracies when the number of training samples is 10 and 100 test samples per class are used for the University of Pavia dataset. We notice from Table 3.2 that for all except one case, the overall classification accuracies improve when our proposed dimension reduction approaches are used instead of the baseline methods. There is one case when the kernel LFDA algorithm performs slightly better than the kernel version of our proposed algorithm, but even for that case the classification accuracy produced by the proposed algorithm is still very close to the best accuracy (difference of 0.42% compared to KLFDA, when the number of training samples per class is 40). For this specific case angular distances do not offer significant advantages as classes are already well separated in the euclidean space. This is further suggested by the reported observations, where completely deviating from the trends observed using the University of Houston and SpecTIR datasets, Euclidean distance based KLFDA algorithm performs better than the angular distance based KLADA algorithm, for the University of Pavia dataset. This could simply be due to the fact that the University of Pavia dataset is impacted

less by illumination differences and other factors which are exploited by angular distance based algorithms. This happens as this dataset has a separate class representing 'Shadows', which implies that majority of the other classes are not impacted by unwanted shadows or illumination differences. We notice from Table 3.5 that for the classes which are harder to classify as — *Asphalt, Gravel, Soil and Bricks* — our algorithms significantly outperform the other baseline algorithms.

SpecTIR Dataset: Table 3.3 shows the overall classification accuracies as a function of the number of training samples for the SpecTIR dataset acquired by our lab. Table 3.6 shows the class specific accuracies when the number of training samples is 12 and all available test samples are used during classification. We notice from Table 3.3 that for all cases, the overall classification accuracies improve when our proposed dimension reduction approaches are used instead of the baseline methods. We notice from Table 3.6 that for the classes which are harder to classify as — *Sabal mexicana (palm tree) and Avicennia germinans (black mangrove)* — our algorithms significantly outperform the other baseline algorithms.

Tables 3.7, 3.8 and 3.9 show the overall accuracy as a function of the reduced dimensionality of the data for the University of Houston, University of Pavia and SpecTIR datasets, respectively. **For both Table 3.7 and 3.8:** 50 samples per class were used for training the algorithms and testing was done on 100 randomly selected samples. **For Table 3.9:** 30 samples per class were used for training and all the other labeled samples were used for testing.

We notice that the overall accuracy does not change significantly with respect to the reduced dimensionality after the dimensionality increases beyond 20. Thus, we limit our experiments to compare only 2 algorithms here and focus primarily on the more interesting observations where the overall accuracy is a function of the number of training samples per class. In our comparison our proposed KSSLADA-sp. algorithm produces the highest classification accuracy followed by the SC-MK algorithm.

However, we do not need feature reduction for SC-MK, as feature reduction doesn't theoretically benefit SVMs, since the kernel algorithm ultimately projects the data to an infinite dimensional subspace. Thus we limit the comparison of the linear version of our proposed algorithm (SSLADA-sp.) only to the SELF algorithm.

Figures 3.4-3.5 show the classification maps of the entire datasets for the University of Houston and University of Pavia images, respectively, with our proposed KSSLADA-sp. algorithm and the best baseline algorithm SC-MK. For the University of Pavia classification maps - Asphalt roads (depicted by gray color) are much better classified using the proposed KSSLADA-sp. algorithm than SC-MK algorithm. The asphalt roads in the classification map using our algorithm are less impacted by noise. For the University of Houston classification maps - The SC-MK algorithm misclassifies parking lot 2 (depicted by light blue color) as commercial area (depicted by greyish blue color) or assigns other classes to the pixels belonging to parking lot 2.

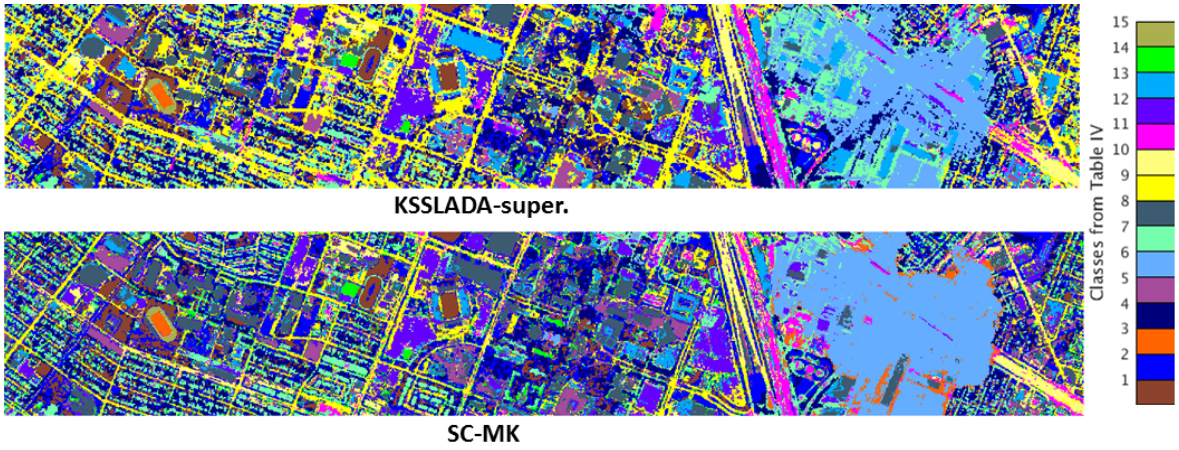


Figure 3.4: Classification maps with the proposed KSSLADA-sp. and baseline SC-MK algorithm for University of Houston dataset

Table 3.1: Overall accuracies (%) for the University of Houston data.

Algorithm / N	***10***	***20***	***30***	***40***	***50***
PCA	65.81±2.5	69.10±2	69.95±1.7	71.56±1.8	72.39±1.9
KPCA	64.59±1.7	68.95±1.7	69.85±2.1	71.34±1	72.52±1.6
LFDA	57.37±5.4	74.13±3	76.51±2.6	78.53±1.3	78.9±2.5
KLFDA	78.75±2.4	82.17±2.5	83.79±2.3	85.72±2.7	84.55±2.2
LPP	50.87±2.8	75.61±2	81.35±2.1	85.41±1.7	88.65±2.6
KLPP	79.87±4.2	86.62±1.2	90.53±1.4	90.11±1.4	92.02±1.6
LADA	83.37±2.6	88.14±1.9	90.10±1.5	91.93±1.7	92.70±1.3
KLADA	83.89±2.2	90.77±1.5	92.69±0.7	94.91±1.4	96.19±0.8
SELF	74.47±2.8	78.93±2.3	79.5±2.4	80.47±1.8	82.17±1.7
SC-MK	86.26±1.9	91.32±1.7	93.56±1.0	94.6±0.9	95.18±1.3
No feature reduction	77.47±2.3	83.59±1.2	86.19±1.1	88.33±0.9	90.04±1
SSLADA-rect. (proposed)	83.95±2.3	89.33±1.9	90.75±1.1	92.58±1.6	93.60±1.4
KSSLADA-rect. (proposed)	85.81±1.7	91.45±1.7	93.87±1.5	96.07±1.2	97.29±0.7
SSLADA-sp. (proposed)	**87.1±1.8**	**92.53±1.3**	**94±0.8**	**95.63±1.2**	**97.3±1**
KSSLADA-sp. (proposed)	**87.82±1.6**	**93.28±0.9**	**94.69±0.8**	**96.67±0.6**	**98.2±0.7**

Table 3.2: Overall accuracies (%) for the University of Pavia data.

Algorithm / N	***10***	***20***	***30***	***40***	***50***
PCA	66.02±2.6	69.21±3.1	71.23±2	72.62±1.2	73.46±2.7
KPCA	67.86±3.5	71.79±2.2	72.48±1.8	75.60±1.7	73.36±2
LFDA	55.44±3	68.17±3	72.27±1.8	73.89±1.3	73.73±2.2
KLFDA	81.34±2.2	86.21±2.3	90.14±2.2	**91.58±1.6**	92.11±1.2
LPP	36.38±2.5	67.68±2.5	74.48±2.2	78.6±2.2	79.79±2.1
LADA	74.54±2.5	77.81±2.4	79.18±2.5	79.14±1.7	81.12±1.9
KLADA	79.92±3.2	83.81±2.5	86.36±1.4	85.38±1.1	87±1.6
SELF	72.42±3.09	75.74±2.8	78.51±3.0	82.13±1.9	81.92±2.4
SC-MK	75±4.1	83.72±1.2	86.95±2.2	90.2±1.7	91.27±1.3
No feature reduction	70.5±2.1	75.92±1.4	75.24±1.9	76.77±1.9	78.7±1.3
SSLADA-rect. (proposed)	75.40±3	78.83±2.6	80.53±2.3	80.42±1.8	81.77±2.2
KSSLADA-rect. (proposed)	82.24±3.2	86.53±2.3	88.61±1.9	89.59±1.1	90.82±1.5
SSLADA-sp. (proposed)	**77.51±1.9**	**80.38±2.1**	**82.07±2**	**82.02±1.3**	**83.67±2.3**
KSSLADA-sp. (proposed)	**83.5±2.7**	**87.92±1.5**	**90.53±1.6**	91.16±1.3	**92.13±1**

Table 3.3: Overall accuracies (%) for the SpecTIR dataset.

Algorithm / N	***6***	***12***	***18***	***24***	***30***
PCA	59.49±3.5	62.59±3.6	64.7±4	64.24±2	64.27±2.7
KPCA	66.88±4.2	69.89±1.8	70.99±1.5	71.9±1.3	72.28±1.3
LFDA	66.38±3.8	71.93±3	73.48±3	77.51±2.8	78.43±2
KLFDA	71.79±3	74.91±2.3	76.77±2.8	76.7±2.7	77.57±1.6
LPP	49.05±4	51.47±3.4	77.27±3.5	80.79±2.1	81.2±2
KLPP	66.31±4.3	72.11±2.7	81.38±0.8	82.21±2.3	82.76±1.3
LADA	75.99±4.4	78.42±3.2	77.84±2.5	79.48±2.7	79.99±1.2
KLADA	75±3.2	78.32±2.3	80.02±1.2	79.56±2	80.47±1.2
SELF	60.99±2.9	73.03±2.5	77.11±2.2	76.87±2.7	76.03±2.4
SC-MK	69.34±2.6	72.13±1.37	72.42±1.73	74.16±2.9	73.86±2.3
No feature reduction	65.41±2.8	69.2±1.8	70±1.2	72.1±1.8	71.8±1.6
SSLADA-rect. (proposed)	75.19±3.3	81.21±2.7	80.87±2.3	82.07±2.4	82.9±1.7
KSSLADA-rect. (proposed)	77.49±2.1	80.03±2.5	83.76±1.8	**83.77±2.9**	85.9±1.5
SSLADA-sp. (proposed)	**77.37±3.4**	**82.57±1.5**	**82.76±1.9**	**83.71±2.6**	**84.67±1.6**
KSSLADA-sp. (proposed)	**77.68±3**	**81.82±1**	**83.96±2.4**	83.73±1.5	**86.05±1.2**

Table 3.4: Class-specific accuracies (%) for the University of Houston data.

Classes / Algorithm	***LADA***	***KLADA***	***SELF***	***SC-MK***	***SSLADAs***	***KSSLADAs***
1. Grass-healthy	97.8	98.8	99.4	96.1	95.6	98.2
2. Grass-stressed	94.2	99.2	89.5	98.7	98.5	98.1
3. Grass-synthetic	100	100	96.8	98.3	100	100
4. Tree	98.3	99.4	94.6	98.7	97	99.6
5. Soil	99.4	97.2	98.1	93.9	99.9	99.9
6. Water	84.7	90.6	82.9	95.9	94.1	97.6
7. Residential	73.9	74.3	55.1	80	85.2	82.9
8. Commercial	72.3	60.6	56.5	61.8	71.7	59
9. Road	56.3	72.4	40.5	79	63.2	79.5
10. Highway	87.9	76.3	73.9	87.2	85.5	88.2
11. Railway	70	77.2	66	84.7	83.9	95.5
12. Parking Lot 1	66.2	60.1	55.6	68.7	68.9	62.7
13. Parking Lot 2	49.7	52.3	32.3	55.2	63.7	56.9
14. Tennis Court	99.7	100	95.7	97	100	99.7
15. Running Track	100	100	80.1	98.8	99.3	99.5
Overall Accuracy	83.36	83.89	74.5	86.3	87.1	87.82

Table 3.5: Class-specific accuracies (%) for the University of Pavia data.

Classes / Algorithm	***LADA***	***KLADA***	***SELF***	***SC-MK***	***SSLADAs***	***KSSLADAs***
1. Asphalt	44	74.3	38.4	68.6	45.8	76.8
2. Meadows	70	81.9	66.6	74	71.3	73.1
3. Gravel	75.5	85.7	73	70.5	76.9	85.5
4. Trees	88.2	51.9	84.1	89.1	86.2	76.1
5. Metal Sheets	100	91.1	99.2	99.1	99.4	97.6
6. Soil	70.5	78.1	63.5	70.6	71.2	80.7
7. Bitumen	90.1	95.5	86	84.1	91.2	90.5
8. Bricks	57.8	84.6	60	70.5	59.7	72.6
9. Shadows	74.6	76	81	99	95.9	98.6
Overall Accuracy	74.54	79.92	72.4	75	77.51	83.5

Table 3.6: Class-specific accuracies (%) for the SpecTIR data.

Classes / Algorithm	***LADA***	***KLADA***	***SELF***	***SC-MK***	***SSLADAs***	***KSSLADAs***
1. Upland grass	63.4	96.4	98.4	25.2	98.2	97.4
2. St. Augustine grass	40.2	52.8	42	32.8	58.5	60
3. Sesbania (rattlebox)	97.4	97.5	90	77.8	89.2	92.9
4. Upland tree	54.9	57.4	59	52.7	52.5	45.4
5. Phragmites austrails	23.6	20.4	43.5	25.9	34.4	22.6
6. Sabal mexicana (palm tree)	3.4	5	0.9	17.1	17.5	47.7
7. Spartina alterniflora	61.4	53.5	65.4	58.9	63.6	75.9
8. Juncus roemerianus	90.2	97.7	97.9	96.7	87.7	95.93
9. Batis maritima / Distichlis spicata	100	100	99.4	100	100	99.6
10. Distichlis spicata	99.9	100	94.7	99.7	99.9	100
11. Baccharis halimifolia	95.2	89	71.4	91.6	97.9	95.1
12. Avicennia germinans (black mangrove)	89.3	82.1	73	81.1	88.2	85
Overall Accuracy	78.4	78.3	73	72.1	82.6	81.9

Table 3.7: Overall accuracies (%) for the University of Houston data versus reduced dimensionality of the data (50 training samples per class used)

Algorithm / r	***10***	***20***	***40***	***80***	***100***
SELF	78.51±1.7	82.17±1.7	81.49±1.3	81.59±1.7	81.63±2
SSLADA-sp.	92.73±1.1	97.3±1	95.97±0.9	96.41±0.5	96.48±0.9

Table 3.8: Overall accuracies (%) for the University of Pavia data versus reduced dimensionality of the data (50 training samples per class used)

Algorithm / r	***10***	***20***	***40***	***80***	***100***
SELF	76.21±3.6	81.92±2.4	80.57±2.2	80.04±1.5	78.88±3.1
SSLADA-sp.	77.83±1.9	83.67±2.3	82.39±2	81.11±2	81.83±1.8

Table 3.9: Overall accuracies (%) of the SpecTIR data versus reduced dimensionality of the data (30 training samples per class used)

Algorithm / r	***20***	***40***	***80***	***160***	***200***
SELF	76.03±2.4	76.64±2.4	74.72±2.8	74.67±2.5	73.7±2.6
SSLADA-sp.	84.67±1.6	82.96±2.5	81.6±2.3	82.17±2.1	81.52±2.9

Figure 3.5: Classification maps with the proposed KSSLADA-sp. and baseline SC-MK algorithm for University of Pavia dataset

3.6 Conclusion

In this work we propose a semi supervised approach to utilize the unlabeled samples during supervised feature extraction and also propose a general method which can be applied to utilize the spatial information by embedding the spectrally similar spatial neighborhoods using superpixels and small fixed sized rectangular windows, for angular discriminant based feature extraction algorithms such as LADA. We also kernelize our proposed linear algorithm in order to extract features from data which are non-linearly separable. We show that embedding the spatial neighborhoods during the process of feature extraction significantly improves the classification accuracies by applying the neighborhood embedding algorithms to LADA. We show that our proposed methods are able to extract better features which train the classifier in a more effective manner and produce higher classification accuracies than other state of the art baseline feature extraction or dimension reduction approaches.

Chapter 4

A Spatial-Spectral Semisupervised Deep Learning Framework using Siamese Networks and Angular Loss

Deep learning has gained popularity in recent times in the field of feature-extraction, object-identification, object-tracking, change-detection, image-classification, spatio-temporal-data analysis, and hyperspectral imaging. Most of the supervised tasks using deep learning require a large number of labeled samples, barring which the model tends to overfit and do not generalize well to the test data. Semi-supervised learning is very beneficial for hyperspectral images which contain abundant unlabeled data samples in comparison to labeled data. Furthermore, it is known that for datasets in which samples are related to each other in all three dimensions such as videos, three-dimensional biological images and hyperspectral images, the use of spatial-spectral / spatial-temporal based deep learning strategies, which can exploit the relationship between pixels in all three-dimensions, has also seen a rise in the past

few years. Moreover, to date, deep feature extraction and classification has been done using euclidean distance based metrics. Foray into the field of angular feature extraction and classification, which is known to work better when samples are impacted by resolution or illumination differences, has not yet been made. We propose a novel spatial-spectral semisupervised deep learning approach based on angular distances by projecting the deep features onto the surface of an l_2-normalized unit hypersphere.

4.1 Introduction

Deep learning has been an area of research since many years [76]. The rise of accelerated computational power has led to a growing interest and revival of deep learning based image analysis methods [77]. The remarkable results in the field of image classification based tasks [77] in recent times has made deep learning popular in the field of hyperspectral image classification as well. To date, a majority of the work has exploited euclidean distances for deep feature extraction and classification. To counter the major disadvantage of overfitting in the field of deep learning i.e. to prevent overfitting due to a low number of available labeled data, many research groups have explored the field of semi-supervised deep learning. For hyperspectral image classification, since the unlabeled samples are available very easily compared to the labeled samples, semi-supervised deep learning is a very important and a burgeoning field of research. There are papers in recent literature which have already started to explore the field of semi-supervised deep learning in the domain of hyperspectral image analysis [78, 71]. Our contribution in this chapter is three fold: (1) We propose a 1D-Semi-Supervised Spectral deep feature extraction and classification method to make use of the unlabeled samples in addition to the labeled samples to learn several million parameters from our deep network, (2) To exploit the Spatial-Spectral relationship between the pixels we integrate the above setup with 3D-CNN's which

are capable of learning three-dimensional spatial-spectral filters from the data and (3) We also make our objective function for deep feature extraction and the back-end classification softmax function to utilize angular distances between pixels / frames instead of euclidean distances, which is normally used. To the best of our knowledge even though there has been a foray in the first field, to pre-train deep networks using unsupervised objective functions in the past, and the second field, to exploit the spatial-temporal [79, 80] and spatial-spectral [81, 82] relationship between pixels by using 3D-CNN's, it is still an emerging area within hyperspectral image analysis. To the best of our knowledge, angular distance based deep feature extraction and classification of hyperspectral images has not been developed or studied previously either.

4.2 Related Work: Semi-Supervised Deep Learning

Deep Learning has been used in the field of machine learning for a long time [83, 84]. Recently, due to the improved hardware availability it has been possible to train the deep models in realistically short periods of time. However, most supervised models still require a large number of labeled samples during training, in order to learn models which generalize well on the validation / testing data. Thus the use of unlabeled data is of prime importance in the field of deep learning.

Semi-supervised deep learning based on discriminative learning [78, 85] have been proposed. Discriminative models aim to directly map the inputs and outputs of systems and avoid any modeling of the underlying distributions. In the paper [85], the authors use an Euclidean distance based metric learning approach to perform deep semi-supervised learning. They implement a loss function for their deep network, which reduces the euclidean distance between the deep features of similar samples and

increases the euclidean distance between the deep features of the dissimilar samples. Other than discriminative learning, generative learning approaches have also been used to perform unsupervised and semi-supervised deep learning [86, 87, 88, 89, 90, 91]. Generative models produce a probability distribution over all variables in a system and manipulate it to perform classification. Recently, approaches combining generative and discriminative models as [92, 93, 94, 95, 96] have been used to perform semi-supervised learning. These models penalize the generative model as long as the samples drawn from it do not perform well in a discriminative model, in a mini-max optimization game. In [92], the authors train feed-forward neural networks with additional penalty from an auto-encoder. [78] proposed a pre-training strategy where the initial layers of the deep network are trained using data and cluster labels and then a few tunable layers are added at the end of the network to learn from the limited number of training samples. This paper uses the strategy of pre-training and transfer learning in an efficient manner. By pre-training using cluster labels the initial layers of the network are able to learn filters which will be able to discriminate between samples belonging to different clusters and group samples which belong to the same cluster. The same filters (along with a few more learnable filters at the end) are then used to discriminate between samples belonging to different classes (due to oversegmentation of the image, we assume that samples belonging to different classes belong to different clusters) by fine-tuning on the limited number of labeled samples. Though an efficient strategy, this method is not optimized for hyperspectral image classification: (1) It neglects the spatial information contained in hyperspectral image cubes by vectorizing and using spectral data pixels; (2) It neglects the use of regularizers that are directly related to unsupervised embedding algorithms, which is known to be a very important approach in the field of semisupervised learning [85, 97, 98, 99, 100]; (3) It extracts features and performs classification using euclidean distance based metrics, which are not optimal for hyperspectral images impacted

by shadows and illumination differences. In this chapter we propose a deep neural network which will address each of the mentioned issues.

4.3 Proposed Work: Semi-Supervised Deep Learning via Angular distance embedding

4.3.1 Motivation

It well known that for hyperspectral datasets, features which are Angularly discriminative perform better than features which are generated based on Euclidean distance based metrics, in terms of classification performance on data which have illumination differences between pixels belonging to the same class [24]. 4.1 clearly shows this phenomenon. Furthermore, it has also been observed that task specific unsupervised pre-training to learn the millions of parameters in deep neural networks improves the deep features and increases the classification performance of the back-end softmax classifier (which is generally used for classifying the extracted deep features) [78]. It can also be hypothesized that the availability of a large number of unlabeled samples or unlabeled frames along with a deeper network can help to increase the complexity of the deep layers during the pre-training stage, without leading to the problem of over-fitting. We expect that the framework we present in this chapter will allow us to construct deeper networks with a large number of trainable parameters that can be learned from unlabeled data in a semi-supervised manner. Futhermore, we also know that utilizing the spatial contextual information from images helps to improve the generation of more robust features. Thus we propose a Spatial-Spectral-Semi-Supervised-Angular feature extractor followed by an Angular-distance metric based back-end Softmax classifier in our current chapter.

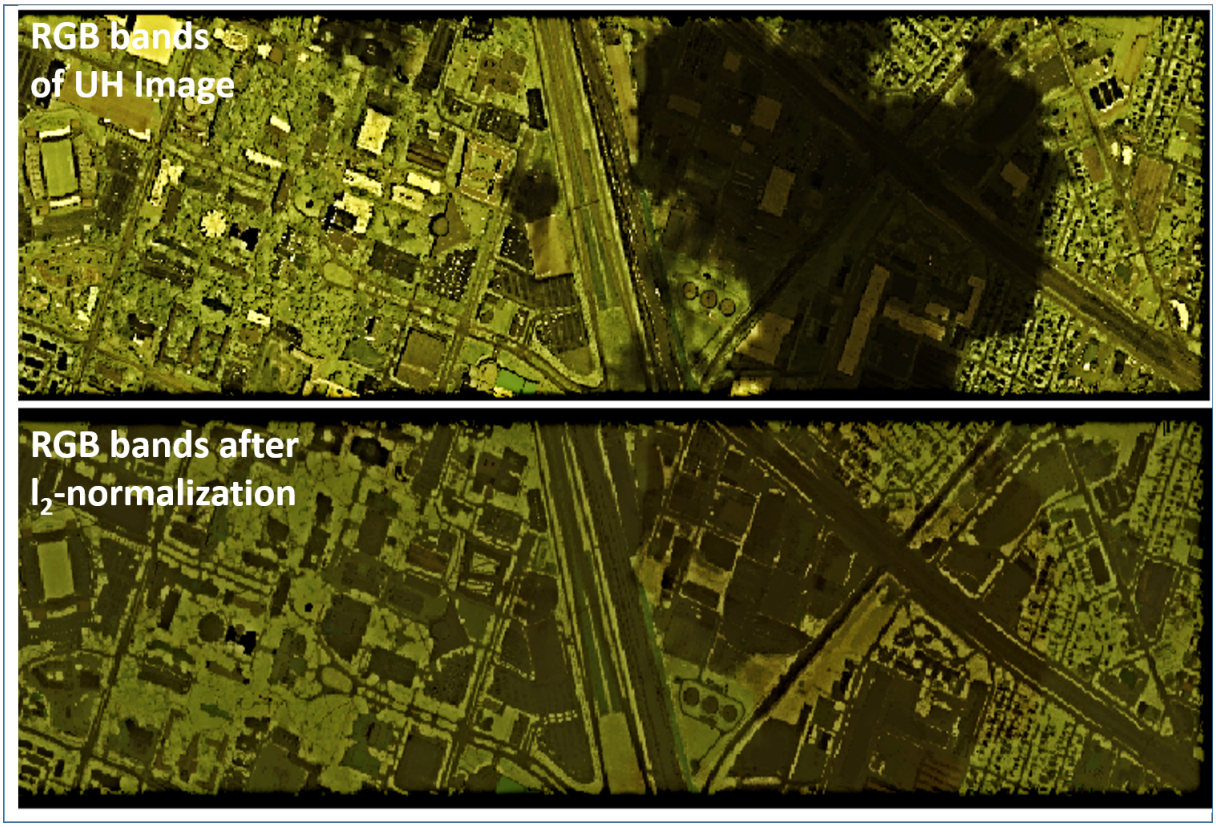


Figure 4.1: Removal of clouds from UH image after l_2-normalization, i.e. after samples are projected onto a unit hypersphere

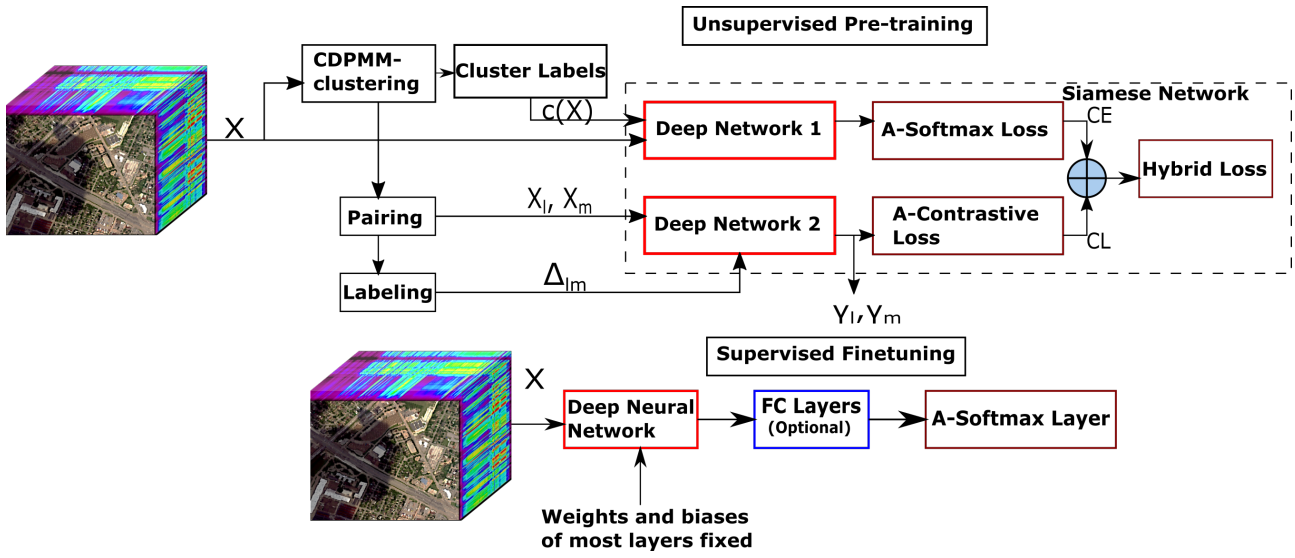


Figure 4.2: Block diagram representation of the proposed 3D Network architecture (CE and CL functions are as defined in Equation 4.1. A-Contrastive and A-Softmax Loss are as defined in Equations 4.3 and 4.7, respectively)

4.3.2 Algorithm

The block diagram representation of our proposed work is shown in Figure 4.2. The algorithm for the proposed method is described in Section 4.3.2 under the *Explanation* paragraph.

As shown in the figures and in Section 4.3.2 under the *Explanation* paragraph we first cluster the unlabeled samples from our image using a Constrained Dirichlet Process Mixture Model (C-DPMM) based approach as proposed in [101]. DPMM based models find clusters by minimizing an optimzation function, which considers: (1) The distance of the samples of interest from other clusters, (2) The density of samples in nearby clusters and (3) The cost of forming new clusters. The optimization function is formed using Bayesian probabilistic approaches. C-DPMM is a variant of DPMM having additional no-link (between inter-class samples) and to-link (between intra-cluster samples) constraints. Next we perform unsupervised pre-training using a joint loss function which combines the angular cross-entropy and angular contrastive loss functions, using a siamese neural network [102]. Finally, we add a few tunable layers at the end of the network to learn the specialized features from unseen data (having a very low number of labeled samples) to perform supervised fine-tuning.

Explanation:

Preprocessing: The input hyperspectral image and ground truth is available: $Image \in \mathbb{R}^{rw \times cl \times d}$ and Ground Truth: $Y \in \mathbb{R}^{rw \times cl}$. Clustering of the image is done by applying the C-DPMM algorithm [101] to generate i different clusters denoted by: $\zeta_i = \{z_j\}_{j=1}^{n_i}$ for $i = 1, 2, ..., \mathcal{N}$ (where i : cluster index and j : pixel index within the i^{th} cluster, ζ are the different clusters, z represents the individual pixels within a cluster, $\mathcal{N}$ is the total number of clusters and n_i are the total number of points in each cluster). We then form spatial-rectangular-window-frames X_d and X_e around the unlabeled sample pixels x_d and x_e, respectively. We form a set G containing pairs of spatial-window-frames according to the cluster label of the central pixel in each frame. Let

G = [ ] initially, and:

if $x_d \implies \zeta_i$ and $x_e \implies \zeta_i$, then

$G = G \subset \{X_d,\, X_e,\, \Delta_1 = 1\}$

else

$G = G \subset \{X_d,\, X_e,\, \Delta_0 = 0\}$

Where G contains similar and dissimilar spatial window frames generated according to the cluster labels of the central pixel of each frame; and $\Delta_1 = 1$ if the frames are similar, else $\Delta_0 = 0$.

Unsupervised Pre-training: We acquire $\gamma_l = f_l(X_l)$, $\gamma_m = f_m(X_m)$, where γ_l and γ_m are the feature descriptors generated from the penultimate layer of the deep neural networks, produced by non-linear mapping (parameterized by the deep neural network) of the window frames X_l and X_m, respectively. We then constrain these features to lie on the surface of a unit hypersphere by performing $l_2 - normalization$. Finally, we minimize the angularly discriminative loss function (inspired by the contrastive loss function defined for Euclidean space as proposed in [85]):

$$\begin{aligned} &H(c, c') + \max_{\cos(\angle\gamma_l - \angle\gamma_m)} [\Delta_{lm}(\cos(\angle\gamma_l - \angle\gamma_m))^2 \\ &+(1 - \Delta_{lm})(\min(0, \Theta - \cos(\angle\gamma_l - \angle\gamma_m))^2)] \ , \end{aligned} \tag{4.1}$$

where $H(c, c')$ is the categorical cross-entropy function when only cluster labels are used to train the sample frames, c is the true cluster label of the sample frame, c' is the cluster label predicted by the deep network, $\Delta_{lm} = 1$ or 0 depending on whether the central pixels of X_l and X_m have the same cluster labels or not, Θ makes sure that the final deep features of dissimilar samples are separated from each other by at least some minimum angular distance.

Supervised Fine-tuning: We then proceed by fixing the layer parameters (weights and biases) of all the layers before the $l_2 - normalization$ layer, to the final layer pa-

rameters after optimization of the unsupervised pre-training loss. Finally, we optimize the loss function for Angular Softmax Classifier

$$\min \left[-\frac{1}{M} \sum_{p=1}^{M} \log \frac{\exp(W_{y_p}^T f(X_p) + b_{y_p})}{\sum_{q=1}^{\Xi} \exp(W_q^T f(X_p) + b_q)} \right] \\ s.t. \qquad ||f(x_p)||_2 = 1 \quad \forall p = 1, 2, ..., M \ , \tag{4.2}$$

where X_p is the input rectangular frame in batch of size M, y_p is the corresponding class label of the p^{th} sample, $f(X_p)$ is the feature descriptor obtained from the penultimate layer of the deep neural network (layer before the Angular Softmax Classifier), Ξ is the number of classes, W and b are the weights and bias of the last layer of the network, which is the softmax classifier. The decision boundary of Angular Softmax Classifier is calculated using the features which are projected onto a hypersphere and separated based on angular distances instead of euclidean distances.

4.3.3 Angular Deep feature extraction by Pre-training

Clustering and Pairing via CDPMM

As we propose a general framework, our model can use any state-of-the-art clustering methods. But in our work here we use Constrained-DPMM based clustering as proposed in [101].

Angular Loss Function

Inspired by the Euclidean distance based metric learning approach from [85] we propose a joint angular based loss function for the purpose of pre-training most of the initial layers of our deep network.

$$H(c, c') + \max_{\cos(\angle\gamma_l - \angle\gamma_m)} [\Delta_{lm}(\cos(\angle\gamma_l - \angle\gamma_m))^2 + (1 - \Delta_{lm})(\min(0, \Theta - \cos(\angle\gamma_l - \angle\gamma_m))^2)] \ , \tag{4.3}$$

where $H(c, c')$ is the categorical cross-entropy function when only cluster labels are used to train the sample frames, c is the true cluster label of the sample frame, c' is the cluster label predicted by the deep network, γ_l and γ_m are the non-linear feature descriptors generated by the penultimate layer (before the angular softmax classifier) of the deep neural network corresponding to the input frames X_l and X_m respectively, $\Delta_{lm} = 1$ or 0 depending on whether X_l and X_m have the same cluster indices or not, Θ makes sure that the final deep features of dissimilar samples are separated from each other by at least some minimum angular distance.

4.4 Classification of Angularly Discriminative features

4.4.1 Softmax Classification

Normally an euclidean distance based softmax classifier is used at the back-end to classify the extracted deep features. The equation for the Softmax Classifier will be

$$\mathbb{P}_s = \frac{\exp(W_s^T f(x) + b_s)}{\sum_r^{\Xi} \exp(W_r^T f(x) + b_r)} \ , \tag{4.4}$$

Where W and b are the weights and biases for the final softmax classification layer; and $f(x)$ is the non-linear mapping parameterized by the penultimate layer of the deep neural network (just before the final softmax classification layer) whose input is

x. The predicted label will be assigned to class s if $\mathbb{P}_s > \mathbb{P}_r \ \forall r \in \{1, 2, 3, 4, ..., \Xi\}$. Where Ξ is the total number of classes in the dataset. Thus the optimization function for the softmax classifier would be

$$\min \left[-\frac{1}{M} \sum_{p=1}^{M} \log \frac{\exp(W_{y_p}^T f(X_p) + b_{y_p})}{\sum_{q=1}^{\Xi} \exp(W_q^T f(X_p) + b_q)} \right] , \tag{4.5}$$

where all the symbols are as defined before in Section 4.3.2 under the *Explanation* section.

4.4.2 Angular Softmax Classification

It has been seen that angle based classifiers [39, 103] work better when data samples are impacted by illumination differences (if samples belonging to the same class have illumination differences between them), especially in the field of Hyperspectral Image Analysis [24]. Euclidean distance based classifiers fit to the high-resolution data samples but completely ignore the low-resolution data samples [103]. Features which are discriminative in an angular space or the surface of a hypersphere do not face the same problem. Thus, we implement an angular softmax classifier, by introducing an l_2-normalization layer just before the final softmax classification layer and after the penultimate layer in the deep neural network. This projects the deep features onto the surface of a unit hypersphere, as shown in Figure 4.3. On a hypersphere, minimizing the softmax loss is equivalent to maximizing the cosine similarity between intra-class samples, and minimizing it for inter-class samples. The angular softmax loss (as proposed in [103]) is also able to model the difficult cases with intra-class illumination variance within samples in a more robust manner, as all the angular features have the same l_2-norm. Qualitatively, as seen before, projecting the samples onto the surface of a hypersphere removes the illumination differences or shadows between samples. The equation for angular softmax will then become (as proposed

in [103])

$$\mathbb{P}_s = \frac{\exp(W_s^T f(x) + b_s)}{\sum_r^{\Xi} \exp(W_r^T f(x) + b_r)} \quad s.t. \quad ||f(x)||_2 = 1 \ . \tag{4.6}$$

Thus the optimization function for the angular softmax classifier would be:

$$\begin{aligned} \min \left[-\frac{1}{M} \sum_{p=1}^{M} \log \frac{\exp(W_{y_p}^T f(X_p) + b_{y_p})}{\sum_{q=1}^{\Xi} \exp(W_q^T f(X_p) + b_q)} \right] \\ s.t. \qquad ||f(x_p)||_2 = 1 \quad \forall p = 1, 2, ..., M \ . \end{aligned} \tag{4.7}$$

where all the symbols are as defined before in Section 4.3.2 under the *Explanation* section.

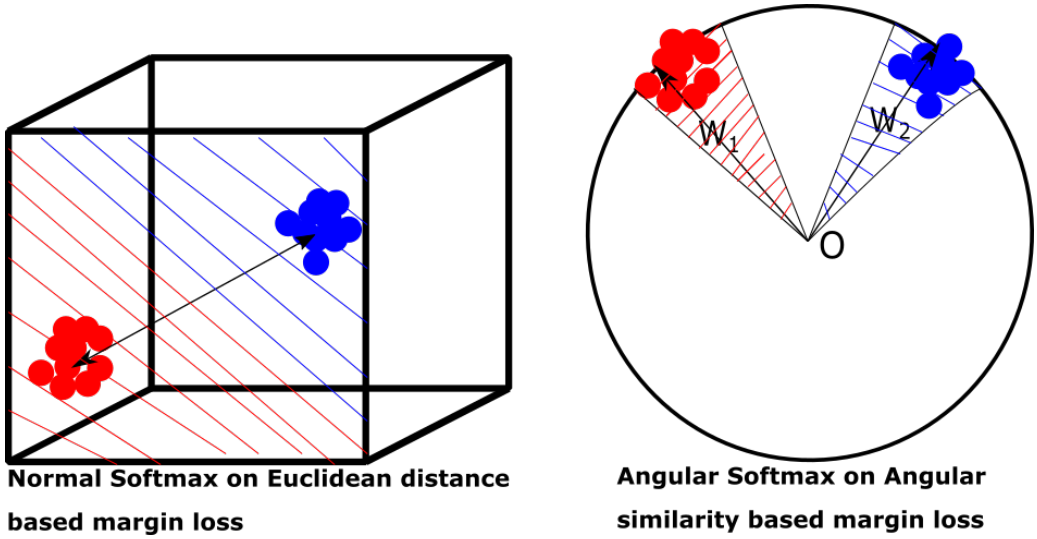


Figure 4.3: Visual representation of the data samples from different classes being separated by original softmax and angular softmax

4.5 Experimental Settings and Results

4.5.1 Hyperspectral Datasets

We validate our proposed algorithms on two datasets: (1) The well-known urban University of Houston dataset from 2013 and (2) The SpecTIR Wetlands dataset which was acquired by our lab at University of Houston, and which captures the Wetlands of Galveston in 2015.

Dataset Partitioning and Parameter Optimization: The entire data was randomly partitioned into three subsets including the - training, validation and testing datasets. It was made sure that the random samples are non-overlapping and also belong to different spatial parts of the image. Since we are using a spatial-spectral approach for classification, it is important to make sure that the training and the validation / testing datasets do not have any spatial overlap between them. This is done to prevent testing on the training data. After obtaining the point samples, ***window-size*** $\times$ ***window-size*** frames surrounding the individual pixels were acquired. These were then used as the training, validation and testing frames. All the free parameters were obtained by tuning them to result in the highest validation accuracy on the validation data frames.

University of Houston Data

The University of Houston dataset captures the campus and the neighboring urban area, using an ITRES-CASI (Compact Airbone Spectrographic Imager) 1500 hyperspectral imager. It covers 144 spectral bands spanning the visible and near-infrared spectrum from 380 nm to 1050 nm. The image has a spatial size of 349×1905 with a spatial resolution of 2.5 m. It contains 15 different classes of interest.

SpecTIR Airborne Data

The data was acquired in a study conducted in 2015, in the coastal wetlands of Galveston, Texas. Wetland vegetation, a crucial part of wetland ecosystem is found to have an immense impact on the species coverage and distribution. Marshes in Mission-Aransas estuary which used to be dominated by smooth cordgrass are now covered mostly by black mangroves. Such sudden drastic changes tend to influence the quality of coastal wetlands, which support a wide variety of marine / aquatic animals as shrimps, fishes, birds and have an impact on the ability of the coastal habitats to buffer wind and wave energy. Therefore, mapping and monitoring the wetland ecosystems will help us to better manage and monitor the endangered wetland ecosystems.

An airborne hyperspectral image was captured using the ProSpecTIR VS sensor on August 14, 2015. Ranging from 400 nm to 2450 nm, the image was captured at 360 wavelength bands, at a spectral resolution of 5 nm. The radiance data were radiometrically and spectrally calibrated before they were converted to the reflectance data using the ATCOR 4 software. The output reflectance data of multiple flight lines were eventually geo-corrected and mosaiced as an image with spatial coverage of 3462×5037 pixels at a 1 m spatial resolution.

Since labeling samples through photo-interpretation was not possible in this situation, a field survey was made on September 16, 2016. ***Upland grass, St. Augustine grass, Sesbania / rattlebox, Upland tree, Phragmites austrails, Sabal mexicana / palm tree, Spartina alterniflora, Juncus roemerianus, Batis maritima / Distichlis spicata, Distichlis spicata, Baccharis halimifolia***, ***Avicennia germinans / black mangrove, Roads, Sand, Soil, Rocks and Urban constructions*** were identified to be the 17 distinct classes. The dataset contains a total of 7219 labeled samples.

4.5.2 Network Architecture

The Deep Neural Network (DNN) architectures used for all our Experiments are as shown in Table 5.1.

Table 4.1: Network Architecture of 3D Deep Neural Networks

Dataset	***Layer***	***Kernels***	***Filters***	***ReLU***	***Pooling***	***Dropout***
	$Conv\ 3D$-1	$32 \times 2 \times 2$	32	Yes	No	50%
	$Conv\ 3D$-2	$32 \times 2 \times 2$	64	Yes	No	50%
UH	$Conv\ 3D$-3	$32 \times 2 \times 2$	64	Yes	$2 \times 2 \times 2$	50%
	$Conv\ 3D$-4	$32 \times 2 \times 2$	128	Yes	$2 \times 2 \times 2$	50%
	$l_2\ norm$-5	–	–	–	–	–
	$Softmax$-6	–	15	–	–	–
	$Conv\ 3D$-1	$32 \times 2 \times 2$	32	Yes	No	50%
	$Conv\ 3D$-2	$32 \times 2 \times 2$	64	Yes	No	50%
	$Conv\ 3D$-3	$32 \times 2 \times 2$	64	Yes	No	50%
Wetlands	$Conv\ 3D$-4	$32 \times 2 \times 2$	128	Yes	$2 \times 2 \times 2$	50%
	$Conv\ 3D$-5	$32 \times 2 \times 2$	128	Yes	$2 \times 2 \times 2$	50%
	$l_2\ norm$-6	–	–	–	–	–
	$Softmax$-7	–	17	–	–	–

4.5.3 Experimental Setup

We compare our proposed algorithms with several other state-of-the-art algorithms. The comparisons show that our methods extract better features compared to other methods, as our methods result in higher classification accuracy values. The training, validation and test frames were generated by using a repeated random subsampling method. We use 10 labeled samples / frames per class and a total of 50,000

unlabeled samples / frames for training, validate on 20 labeled samples / frames per class, and test on 100 labeled samples / frames per class, for all datasets. We found that a frame size of 5×5 works best for our datasets. We run each of the experiments 5 times using a random sample selection strategy and report the average results showing the mean and standard deviations. We compare our methods with - (1) Supervised Spectral classification methods including the KNN and SVM wih RBF kernel; (2) Discriminative Semi-Supervised-Spectral classification methods including Label-Propagation [97] - [It propagates labels along the high-density areas defined by unlabeled data], Transductive SVM's (T-SVM) [104] - [Were proposed to modify SVM's with the aim of max-margin classification ensuring minimum number of unlabeled data samples near the margins], Laplacian SVM's (LapSVM) [105] - [The loss function is a combination of the supervised loss function of normal SVM's and an additional term which introduces a regularization term on the geometry of both supervised and unsupervised samples by using the graph Laplacian], and PL-SSDL(CDPMM) [78] - [Uses cluster labels and data samples to pre-train the initial layers of a DNN, to perform transfer learning capable of learning filters which can discriminate between clusters]; (3) A Generative Semi-Supervised-Spectral classifier - Ladder Networks [106] - [Uses a discriminative approach to learn from labeled samples and a generative approach which aims to minimize the difference between encoder inputs and decoder outputs at each stage. All the layers of the encoder-decoder network share lateral connections with each other]. For the sake of clarity we disintegrate our approach to a set of basic steps and show the results for each and every step, thereby making it clear about how the addition of an approach / concept improves the final overall classification accuracy on a particular dataset.

4.5.4 Results and Analysis

1-D Classification (at the level of pixels): For the sake of comparison we implement our model using 1D CNN's, having the exact same parameters and configuration as the one with 3D CNN's. For normal softmax classification when pixel level training is done, we observed that we get performance similar to other baseline algorithms. Angular Softmax substantially increases the accuracy for the UH dataset and slightly increases the accuracy for the SpecTIR dataset. With pre-training using only pseudo-labels we are able to further boost the performance for both the datasets to a large extent. We also observe that by adding the second contrastive angular based loss as shown in Equation [4.3], we are able to further boost the performance for both the datasets.

3-D Classification (at the level of patches): To utilize the spatial-spectral features from the data we use 3D CNN's. For both the datasets we observe a significant boost in performance when we move to 3D CNN's compared to 1D CNN's. The Angular-Softmax based 3D CNN gives us performance similar to the normal-Euclidean based 3D CNN for the UH dataset, but there is an improvement for the Wetlands SpecTIR dataset. Pre-training the initial layers of the 3D CNN network with only the cluster labels significantly boosts the performance for both the datasets. Similar to the observations for 1D CNN's, we observe that adding the contrastive angular based loss function improves the accuracy for both the datasets.

Classification Maps: As observed from Figures 5.5 and 4.5, depicting the classification maps for the UH and SpecTIR Images, respectively, we see that our proposed method can preserve the inherent / coherent clusters from the original image in a much better manner than the baseline method. The finer details from all regions in the maps are much better reconstructed using our method. The baseline method shows an inclination to smooth through the boundaries of different objects.

Specifically for the UH Image represented by Figure 5.5, the commercial buildings

under shadows are much better reconstructed using our methods compared to the baselines. The baseline method erroneously misclassifies the commercial buildings as to belong to several other unrelated classes. Moreover, the baseline method misclassifies several roads under the clouded region as Synthetic Grass, whereas our methods correctly classify those pixels as roads. We also notice that our method performs much better at the borders of the clouded region compared to the baseline method. The area impacted by cloud which is misclassified in the map created using our algorithm is less than the case when the other baseline method is used. In simple words, clouds have a much severe effect on the maps constructed using baseline methods compared to our methods. Moreover, the baseline method erroneously classifies many roads and highways as railways, our proposed method can reconstruct these classes in a much better manner. It is also known that most of the grass inside particular stadiums should be well-manicured healthy grass. Our method classifies the corresponding pixels correctly, but the baseline methods do not. It shows a tendency to mislabel many pixels as to belong to the class depicted by stressed-grass. The tennis court is much better reconstructed using our method compared to the baseline. The classes depicting roads, highways and residential area are much better represented using our proposed methods compared to the baseline methods.

Specifically for the SpecTIR image as represented by Figure 4.5, a large part of the image near the center is incorrectly labeled as Urban area when baseline method is used for classification, and it is known that the corresponding area does not belong to the urban class. The class depicting roads are much better depicted using our method compared to the baseline method. Moreover, it is known that in the original image most roads are bordered by soil. This is represented in class maps reconstructed using our proposed method in a much better manner compared to the baseline method. The baseline method erroneously classifies the soil bordering some roads as Distichlis spicata, represented by white color as shown in the maps above.

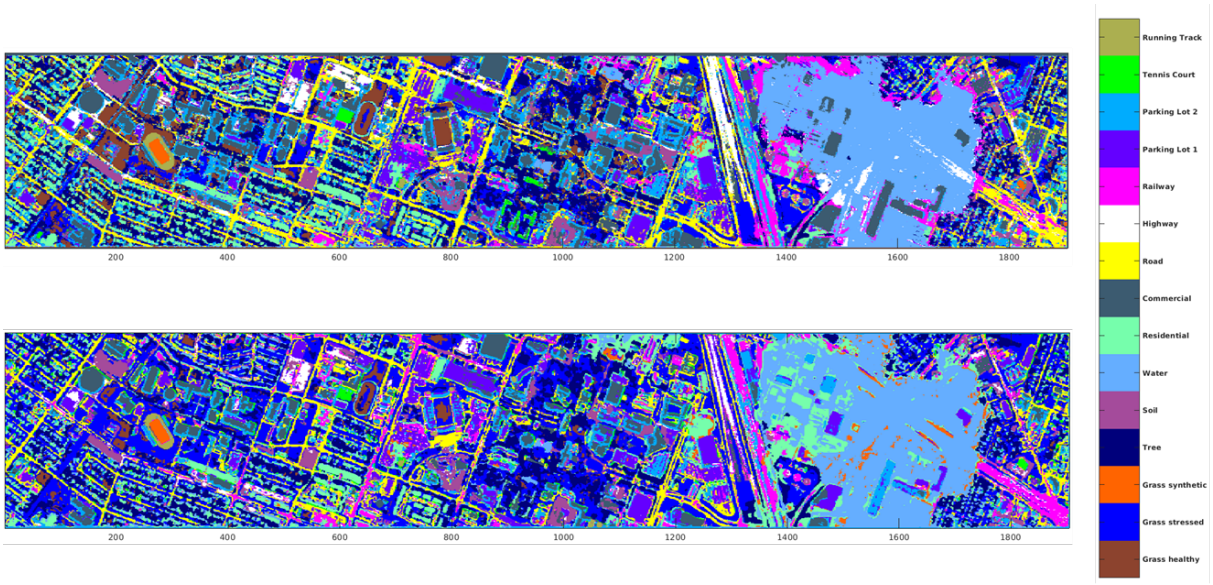


Figure 4.4: Classification maps of UH image dataset with our proposed (last row in Table 1) algorithm (top) and with the baseline method of 3D-CNN with normal softmax (bottom)

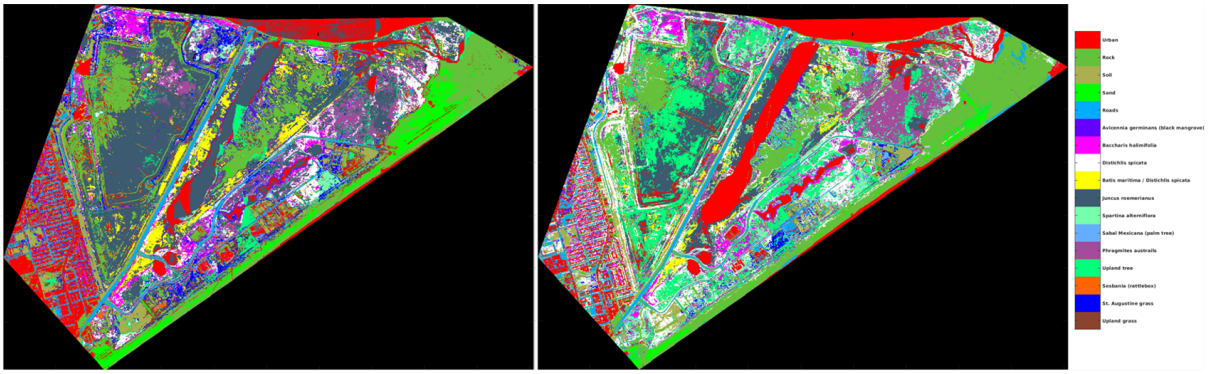


Figure 4.5: Classification Maps of SpecTIR Image Dataset with our proposed (last row in Table 2) algorithm (left) and with the baseline method of 3D-CNN with normal Softmax (right)

Table 4.2: Overall accuracies (%) for the University of Houston data.

	Algorithm	*Accuracy*
Baselines	***kNN***	67.60±1.05
	SVM	71.67±1.58
	Label Propagation	67.56±0.81
	TSVM	72.17±2.26
	LapSVM	74.29±1.09
	Ladder Networks	72.00±1.24
	PL-SSDL (CDPMM)	77.07±1.31
	No pre-training normal softmax	69.41±2.7
Proposed	***No pre-training angular softmax***	74.59±1.71
	ASSDL with pre-training using angular softmax (proposed)	79.88±0.76
	ASSDL with pre-training using angular softmax and contrastive loss (proposed)	81.73±1.23
	3D-CNN with normal Softmax	75.73±1.09
	3D-CNN with Angular Softmax (proposed)	75.61±1.37
	3D-CNN with Angular Softmax and Pre-trained using pseudo-labels only (proposed)	81.88±1.5
	3D-CNN with Angular Softmax and Pre-trained using pseudo-labels and contrastive-loss (proposed)	**82.37±1.46**

Table 4.3: Overall accuracies (%) for the Wetlands data.

	Algorithm	*Accuracy*
Baselines	***kNN***	66.11±1.46
	SVM	72.72±1.19
	Label Propagation	65.54±1.04
	TSVM	74.67±0.7
	LapSVM	77.17±1.33
	Ladder Networks	64.63±1.54
	PL-SSDL (CDPMM)	77.97±2.7
	No pre-training normal softmax	67.56±2.04
Proposed	***No pre-training angular softmax***	68.22±1.24
	ASSDL with pre-training using angular softmax (proposed)	78.22±0.8
	ASSDL with pre-training using angular softmax and contrastive loss (proposed)	78.45±1.2
	3D-CNN with normal Softmax	75.25±1.05
	3D-CNN with Angular Softmax (proposed)	76.94±1.89
	3D-CNN with Angular Softmax and Pre-trained using pseudo-labels only (proposed)	77.52±1.08
	3D-CNN with Angular Softmax and Pre-trained using pseudo-labels and contrastive-loss (proposed)	**81.79±0.6**

4.6 Conclusion

The results show that deep features learned from the large number of available unsupervised data samples or frames can be helpful and used for pre-training the neural network. The classification accuracy improves if the millions of learnable

parameters are learned from the large number of unlabeled data samples or frames instead of the very low number of labeled data samples or frames.

We also show that angularly discriminative deep feature extraction along with an angular softmax classification layer at the back-end can be very helpful for hyperspectral image classification. The results show that our proposed algorithms are capable of generating results which are better than most of the state-of-the-art algorithms.

The 3D filters of the 3D CNN's preserve the spatial-spectral neighbors from the original hyperspectral image, in the final deep feature space. Therefore, using 3D-CNN's to utilize the spatial features in addition to the spectral features, makes the deep features more robust and leads to better overall classification accuracy values. This shows the importance of exploiting the spatial-contextual information between the sample pixels in hyperspectral images.

Chapter 5

Deep Feature Extraction by Semisupervised Capsule Neural Networks for Hyperspectral Image Classification

Deep Neural Networks (DNN's) have been known to suffer from the problem of over-fitting on a limited amount of training data and deliver poor performance on the validation or test datasets. Convolutional Neural Network (CNN) architectures were proposed in the late eighty's with the intent of solving this problem. In recent years, CNN's have emerged as the building blocks of most of the state-of-the-art DNN architectures. In order to counter the problem of over-fitting to the limited number of training samples or reduce the number of parameters in the networks and also introduce a certain degree of invariance between the intra-class features extracted, different forms of pooling operations are introduced in the Deep Neural Networks which use CNN's as their building blocks. Recently, it has been clearly shown and scientifically proven that arbitrarily eliminating features using random pooling oper-

ations severely degrade the quality of the extracted deep features. Capsule Neural Networks have been proposed with a scientific rational approach based on routing algorithms, which work to reduce the parameters between the different layers in the networks by an intelligent routing mechanism, and optimize for the coupling between the neurons of those layers. Due to this efficient pooling strategy, it has been shown that Capsule Neural Networks perform better than the state-of-the-art CNN's when limited number of training samples are available. Since, hyperspectral remote sensing images contain come with limited ground truth (due to the inherent cost of labeling), we propose the use of Capsule Neural Networks for performing hyperspectral remote sensing image classification. Moreover, due to the availability of very large quantities of unlabeled data samples we propose a semisupervised framework, which can exploit the inherent structure in the dataset to perform the image classification.

5.1 Introduction

Recent advances in the field of machine learning have shown that the use of deep neural networks can result in significant improvements for tasks such as - image classification, segmentation, object detection and hyperspectral image analysis. Most of the deep networks use architectures similar to the CNN + Pooling architecture as proposed in [76]. The architecture in [107] was shown to reduce the problem of overfitting in DNN's, by introducing a pooling layer in between the Convolutional layers, in order to reduce the number of learnable parameters in the network. The primitive form of pooling layer generally implemented by the state-of-the-art max-pooling or average-pooling mechanisms, allow neurons in the later layer to ignore all but the most active feature detector in a local pooling window in the layer below or averages all the feature detectors in the pooling window, respectively. This primeval form of operation introduces an invariance between the intra-class features during the process

of deep feature extraction. But, Intra-class variance may generally be very useful to separate features belonging to different classes, or in other words - Intra-class variance or spatial details may be very important for inter-class separation. In such cases, loss of intra-class detail information degrades the quality of the features extracted by the deep neural network. For this reason, over the last few decades many researchers have questioned the rationale behind the arbitrary pooling operations even though the simple operation has been known to produce good results. A better strategy proposed was to introduce covariance or equivariance [108, 109], instead of intra-class feature invariance. Recently many different intelligent and adaptive spatial pooling strategies have been developed in order to replace the arbitrary forms of pooling operations as maximum or average pooling [110, 111, 112]. More recently, [113, 14, 1, 13, 114] have shown that the arbitrary pooling operation which randomly neglects most of the features from previous layers, can be replaced by routing algorithms which have strong fundamental scientific principles governing them. [1] show that such networks can extract more robust features and outperform the state-of-the-art CNN based deep neural networks. Moreover, in this chapter we show that due to the strategic pooling approach behind the proposed neural networks as shown in papers describing Capsule Neural Networks [115, 113], the networks can work much better in the absence of a large number of labeled training samples, compared to the state-of-the-art CNN's. Since, it is known that the high dimensionality and a very limited number of labeled training samples [5] makes the problem of hyperspectral remote sensing image classification [116, 117, 118] extremely challenging, we propose the use of Capsule Neural Networks (as they can extract features which are more robust in the absence of a large number of labeled training samples) to perform hyperspectral remote sensing image classification and show that they perform better than the state-of-the-art convolutional neural network architectures (both qualitatively and quantitatively), which are traditionally in use.

Remote Sensing using Sensors as the Hyperion Imaging Spectrometer has been captured and studied for a long time, for Earth Observation (EO) applications. Hyperspectral Images can capture spectral level differences between different objects / classes but RGB images cannot. For this reason, hyperspectral images having high spectral resolution and capturing hundreds of observation channels are studied and captured. Hyperspectral remote sensing images have their own set of problems: Large spatial variability of hyperspectral signatures over land cover classes, atmospheric effects / inteference and the curse of dimensionality [119, 120, 121]. It has already been proven that deep learning models may not be effective to extract robust features from hyperspectral images unless abundant training samples are available [122], our results in this chapter confirm this fact. Additionally, we also know that despite having very low number of labeled training samples, hyperspectral images contain a very large number of unlabeled samples. Following these arguments, we extend the use of Capsule Neural Networks to propose a Semi-Supervised Capsule Neural Network in order to utilize the unlabeled samples in the datasets, and exploit the inherent structure between the unlabeled data samples.

5.2 Related Work: Capsule Neural Networks

Neural networks have been known to suffer from the problem of overfitting since its inception. Near the end of the last century, [76] proposed Convolutional Neural Network architectures to overcome this problem. Pooling layers in the form of Maximum-Pooling or Average-Pooling layers were inserted between subsequent layers in order to reduce the number of trainable parameters in such networks and overcome the problem of overfitting. The pooling layers in convolutional neural networks also helps to introduce a degree of translational invariance in the architecture. This strategy is known to work very well for known problems in the field of machine learning

such as image segmentation, image classification and object detection. However, this very primitive form of routing which allows neurons in subsequent layers to ignore all but the most active feature or find the average of the features belonging to the local pooling window in the preceding layer, can be euphemistically termed as a random arbitrary operation, defying the statistical properties governing the data. Moreover, it is known that the pooling operation causes the network to lose spatial information from the image being analyzed [123, 124]. Arbitrarily eliminating the features generated from convolutional layers in an unintelligent manner causes the network to loose much of the local spatial-information from the image. Capsule Neural networks were proposed to perform intelligent pooling operation by applying a novel scientifically sound routing mechanism, which would allow the network to preserve the local spatial information or details. For this operation, the scalar output feature detectors generated by CNN's are replaced by vector output capsules and the arbitrary pooling operation is replaced by an intelligent routing by agreement mechanism. Capsules are groups of neurons whose activity vectors represent the instantiation parameters for specific types of entities as objects or object parts, belonging to a certain class.

Capsule Neural Networks were originally proposed in [13], mainly to address the issues of arbitrary pooling operations and loss of local spatial information between the subsequent layers of neural networks. However, the network proposed in [13] did not perform well and led to further research and proposal of the Capsule Neural Networks as in proposed in [1]. [1] shows that the Capsule Neural Network produces state-of-the-art performance on many known and state-of-the-art datasets in the field of computer vision. The primary idea behind moving away from the state-of-the-art neural network architecture as proposed in [76] and moving towards the architecture as proposed in [1], is to introduce the idea of intelligent pooling which can preserve the local spatial information from the original image. This is done by replacing the scalar output feature detectors produced by CNN's with vector output capsules and

replacing the arbitrary pooling operations as maximum / average - pooling with an intelligent dynamic routing mechanism, based on a routing by agreement algorithm.

[125] uses Capsule Neural Networks for hyperspectral image classification. Different from that work, in this chapter our primary objective is to perform Semi Supervised Learning using Capsule Neural Networks, by exploiting the information contained in the abundantly available unsupervised samples from the hyperspectral remote sensing images. We also make a direct comparison of the features learned by our proposed Capsule Networks and state-of-the-art 2D-CNN + Maxpooling Networks. Our comparison shows that our proposed Capsule Networks are able to learn features of higher quality and are also able to capture illumination invariant features from images which are severely impacted by illumination variances, as the University of Houston hyperspectral image.

5.2.1 Pseudocode for Dynamic Routing Algorithm [1]

Algorithm 5 shows the algorithm governing the dynamic routing process which replaces the arbitrary pooling operation in the traditional deep neural network architectures.

The squashing function is applied to the output vector of a capsule s_j as shown in Algorithm 5, in order to make sure that the maximum length of all output length-vectors are equal to 1. The squashing function makes sure that when the predicted output is correct the length of the corresponding output capsule vector is 1, and the length of the corresponding output vector is close to 0, otherwise. c_{ij}'s as shown in Algorithm 5 are the coupling coefficients and they are determined by the iterative dynamic routing mechanism. The coupling coefficients are found from the log-prior probabilities (b_{ij}) which denote the coupling between capsules i and j. The Softmax

Algorithm 1: *Pseudo code of the dynamic routing algorithm [1]*

Input:

• Output of capsule from previous i^{th} layer: u_i

• Number of routing iterations: r

• Layer notation: l

{Iterative Dynamic Routing Algorithm}

• Find the prediction vectors $u_{j|i}$ by multiplying the output of the previous capsule layer u_i with the weight matrix W_{ij}

$u_{j|i} = W_{ij}u_i$

• Let b_{ij}'s be the log probabilities that represent that the lower level capsule i and the next higher level capsule j are coupled

for all capsules i in layer l and capsules j in layer $(l+1)$:

Let $b_{ij} = 0$

for r iterations do:

for capsule i in layer l: $c_i = \text{softmax}(b_i)$

for capsule j in layer $l+1$: $s_j = \sum_i c_{ij}u_{j|i}$

for capsule j in layer $l+1$ apply the squashing function:

$v_j = \frac{||s_j||^2}{1+||s_j||^2} \cdot \frac{s_j}{||s_j||}$

for capsule i in layer l and capsule j in layer $l+1$:

$b_{ij} = b_{ij} + u_{j|i} \cdot v_j$

return v_j

Output:

• The squashed length vector from capsules v_j

operator will then lead to the equation as follows

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{5.1}$$

The softmax operator on b_{ij} makes sure that all the coupling coefficients between the neurons in the two consecutive capsule layers sum up to 1. In simple words, it makes sure that a neuron is fractionally connected (or weighted) to other neurons from the preceding layer, such that the sum of all fractions (or weights) is equal to 1. k represents the total number of available connections which can be made between successive capsule layers.

In simple words, the iterative dynamic routing process makes sure that the coupling coefficients between the lower level and higher level capsules keep on increasing if the dot product between the output vectors generated by the corresponding lower and higher level capsule layers is large, and the coupling coefficients keep on decreasing otherwise. It implies that if there is an agreement between the vectors generated by the lower and higher level capsule layers, then the coupling coefficients between the capsules belonging to the consecutive capsule layers keep on increasing, and the coefficients decrease otherwise. This is why the iterative dynamic routing mechanism is said to work using a *routing by agreement* algorithm.

5.2.2 Block Diagram of Capsule Neural Networks [1]

Figures 5.1 and 5.2 show the block diagram representation and the decoder structure representation of the Capsule Neural Networks as used in our proposed work, and as proposed in [1]. The Capsule Network as shown in Figure 5.1, has 3 layers: The initial Relu-Convolutional Layer, the intermediate Primary Capsule Layer and the final Classification Layer. For future research work, it would be interesting to observe the effect of going deeper with Capsule Layers. The decoder as shown in Figure

5.2 focuses on reducing the Euclidean distance measure between the image and the reconstructed output obtained from the deep neural network. This is a constraint which makes sure that the deep neural network learns features which can accurately reproduce the input image from the corresponding deep features. The length of the

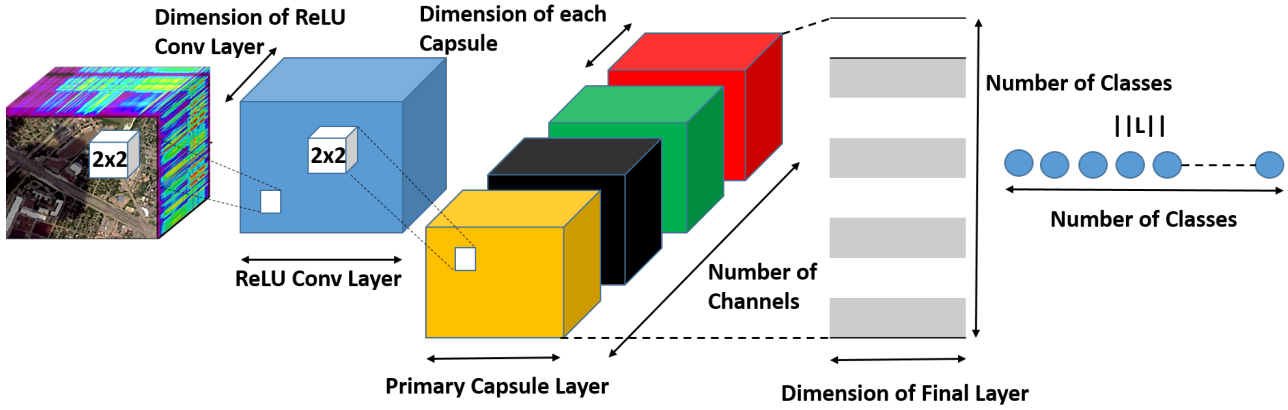


Figure 5.1: CapsNet with 3 layers as proposed in [1] and used in this work.

activity vector of each capsule in Final Layer indicates the presence of an instance of each class and is used to calculate the classification loss. W_{ij} is the weight matrix between each capsule u_i from Primary Capsule Layer and v_j from the Final Capsule Layer. The Euclidean distance between the image and the output of a Reconstruction

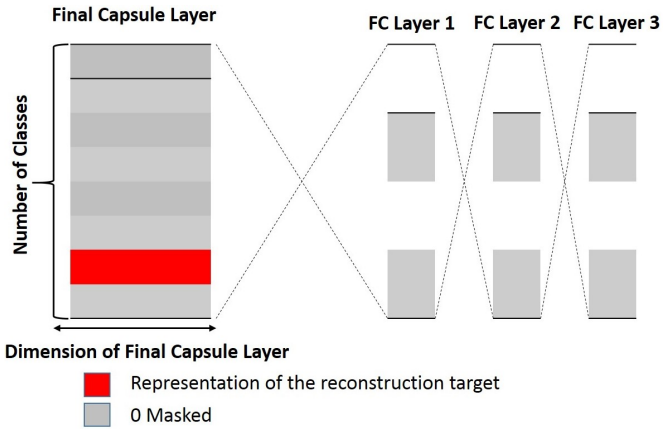


Figure 5.2: Decoder structure to reconstruct the Hyperspectral Image from the Final Capsule Layer feature representation [1].

layer (which uses the deep features extracted by this network) is minimized during

training. True label is used as reconstruction target during training.

5.2.3 Margin Based Loss function [1]

The loss function governing the described network is a margin based loss function which makes sure that the length of the corresponding capsule vector generated for the correct matching class of interest is no less than 0.9 units, when the length of the vector can be any number from 0 to 1 units. It also makes sure that the length of the capsule vector for all non-matching classes other than the class of interest in no more than 0.1 units. In simple words, the capsule vector is large, almost equal to 1, when the predicted class of interest matches with the true class, and is very small, almost close to zero, otherwise. The margin based loss function is defined as follows:

$L_k \ = \ T_k \ \max(0, m^+ - ||v_k||)^2 \ + \ \lambda(1 - T_k) \ \max(0, ||v_k|| - m^-)^2$

Where L_k is the loss of the network, v_k is the deep feature extracted from the last capsule layer as described earlier, and $T_k = 1$ iff a sample of class k is present and $m^+ = 0.9$ and $m^- = 0.1$. The λ down weighting of the loss for the absent classes stops the initial learning from shrinking the lengths of the activity vector of all the higher layer capsules. Here $\lambda = 0.5$ is used. The total loss is simply the sum of losses of all the higher level capsules from the last capsule layer.

5.3 Proposed Work: Semi-supervised Capsule Neural Networks for Hyperspectral Image Classification

Since the recent revival of deep learning based algorithms in the field of machine learning and computer vision [77], hyperspectral image analysis has also benefited significantly from the use of deep learning based algorithms. It is also known that

Hyperspectral remote sensing images benefit from the use of spatial information contained in the image pixels [28, 31, 32, 33]. Exploiting the spatial information between pixels from the image has been shown to substantially improve the robustness of the extracted features, compared to the cases when only spectral information of the pixels were used to learn the features. Moreover, it is also known that hyperspectral image analysis can benefit significantly from the use of spatial-spectral deep learning architectures compared to the use of only spectral based deep learning methods. We have also seen this explicitly in our recently submitted research work, submitted to a different journal.

More recently, Capsule Neural Networks have been used in the field of Computer Vision [1, 14, 126], Biological Image Analysis [127, 128] and others, but this is the first time that it is being used in the field of Hyperspectral Image Analysis for semi supervised feature extraction.

We expand the proposed Capsule Neural Networks to perform Hyperspectral Image Classification instead of RGB image classification as performed in the Computer Vision Community. Our image is of size $m \times n \times d$, where m is the number of rows in the image, n is the number of columns and d is the number of wavelengths at which the image is captured (or the dimensionality of the image). We input rectangular frames surrounding the pixels belonging to specific classes of interest to the first 2D-Convolutional Layer in the Deep Capsule Neural Network. The 2D-Convolutional Layer has d number of input channels.

It is known that unsupervised pre-training using pseudo-labels or cluster labels can help to improve the performance compared to purely supervised classification for hyperspectral remote sensing images, when the number of labeled samples during supervised classification is very low [78]. Semi-supervised learning helps to generalize the model and prevent over-fitting in the absence of a high number of labeled training samples. In our recent work which is currently under review in a different journal, we

also found that more the complexity of the unsupervised pre-training network, more is the robustness of the features generated. Following this, we hypothesize that in the presence of a low number of labeled training samples Capsule Neural Networks (which are much more complex than the traditionally used CNN architectures), can generalize much better than the state-of-the-art Convolutional Neural Networks, and thus we propose the use of Capsule Neural Networks for Semi-Supervised learning in order to extract more robust features during the unsupervised pretraining stage. This will help to generate more robust features from the abundantly available unsupervised data by exploiting the inherent structure and the statistical properties governing the data.

The goal of this chapter is two fold: First, we perform hyperspectral image classification using capsule neural networks and show that the features extracted are better than those extracted from the corresponding traditionally used 2-D CNN's. Next, we propose a semisupervised architecture in order to perform semisupervised learning to make the extracted deep features more robust and improve the subsequent classification performance. We provide both - detailed quantitative and qualitative comparisons to demonstrate our ideas and validate our conjectures or hypothesis.

5.3.1 Motivation

Since we know that Capsule Neural Networks can preserve the spatial details from the images as compared to the state-of-the-art Convolutional Neural Networks and Pooling operations, and we also know that spatial details are very important in hyperspectral images, we propose the use of Capsule Networks to perform hyperspectral image classification. We understand that our hypothesis is only valid for hyperspectral images having high spatial resolution and may not work with images having low spatial resolution, but most of the hyperspectral images captured in modern times have high spatial resolution or enormous amounts of spatial details in them. More-

over, if the intra-class details are important for performing inter-class discrimination during classification then traditionally used state-of-art CNN and pooling operations will not generate robust deep features, due to the loss if information. This is because the pooling operation introduces an invariance between the intra-class features and consequently causes the network to lose important spatial context which could be extremely useful for discriminating between inter-class features during classification.

In general Capsule Neural Networks generate more robust features due to the preservation of spatial context from the images compared to Convolution Neural Networks and the primitive pooling operations. In this chapter, we propose the use of the abundantly available unsupervised data samples to learn the robust features in order to perform semisupervised classification for hyperspectral images. We anticipate that these features will be much more robust than the features learned through the normal CNN-Pooling operations when the same network configuration and same set of training samples are used for pretraining and finetuning the corresponding neural networks, following the exact same semisupervised learning strategy. Our results clearly show that our hypothesis is in fact true and a fact.

It is also known that semisupervised learning increases the quality of extracted features and makes them more robust compared to purely supervised learning [28] and we implement this to improve the features extracted by purely supervised Capsule Neural Networks.

Comparison of features extracted using Semisupervised Capsule Networks and Semisupervised 2D-CNN's: Figures - 5.3 and 5.4 show the features extracted by the proposed Semisupervised Capsule Neural Networks and the traditionally used 2D-CNN based Networks. It can be clearly observed that the features extracted by Capsule Neural Networks are much more discriminative than those extracted by the CNN + Pooling based network. Capsule Networks are able to preserve the object boundaries (objectness) of the distinct objects from the original hyperspectral image

in a much better way. Qualitatively these features appear to preserve the information from the original image in a much better way than the corresponding 2D-convolutional neural networks. Most of the features extracted by the 2D-CNN appear to be darker and in a sense lose much of the spatial details and information from the original image, leading to the reduction of discriminativeness between the inter-class features. Furthermore, as observed from the feature map, as shown in Figure 5.3, and as described in the paragraph below, we observe that certain dimensions of the penultimate layer of the Capsule Neural Network captures illumination invariant features from the University of Houston hyperspectral image, which is severely impacted by cloud shadows. These cloud shadows are known to drastically degrade the performance during Hyperspectral Image Classification, as they introduce varying noises between the intra-class samples, extracted from the well-lit and cloudy region in the image. In simple words, intra-class samples belonging to different spatial neighborhoods of the image, and impacted by different degrees of illumination, will naturally have very different distributions governing them. The difference in distributions for such intra-class samples makes it extremely difficult for any feature-extractor and classifier to classify them as to belong to the same class. This decreases the robustness of the extracted features and leads to classification errors. These observations clearly motivate for the use of Semisupervised Capsule Neural Networks for Hyperspectral Image Classification over the traditionally used Convolutional Neural Networks.

Features extracted using Semisupervised Capsule Networks which capture the specific property of Illumination invariance: The original Capsule Neural Network paper [1] shows that different dimensions of the penultimate *DigitCaps* Layer capture different properties which inherently govern the MNIST dataset / MNIST training samples, such as - scale and thickness, stroke thickness, local skew, width and translation - of the digits. For our hyperspectral image which captures the University of Houston and its neighboring urban area, we notice that the Semisu-

pervised Caspule Neural Network as proposed here is able to capture illumination invariant features as shown in Figure 5.3. Several dimensions of the penultimate layer of the Capsule Neural Network capture these illumination invariant features as shown. The features show that Capsule Networks are able to learn features after removing the clouds and learning about the shapes of the objects underneath the clouds. The shapes of the distinct objects are preserved from the original image in the extracted features. Our intuitive explanation for this observation is: There are certain properties as shapes and sizes of objects which govern the hyperspectral image being studied, and these properties are illumination invariant. Here, we are observing the feature dimensions which are governed by those specific properties which are illumination invariant. As shown in Figure 5.3, we observe that Capsule Neural Networks are able to extract such high-quality illumination invariant features (which as shown in Figure 5.4, is not the case for traditionally used 2D-CNN + Pooling based deep network architectures). This property is known to be very desirable during hyperspectral image analysis.

5.4 Experimental Settings and Results

5.4.1 Hyperspectral Datasets

We validate our proposed algorithms on two datasets: (1) The well-known urban University of Houston dataset from 2013 and (2) The SpecTIR Wetlands dataset which was acquired by our lab at University of Houston, and which captures the Wetlands of Galveston in 2015.

Dataset Partitioning and Parameter Optimization: The entire data was randomly partitioned into three subsets including the - training, validation and testing datasets. It was made sure that the random samples are non-overlapping and also belong to different spatial parts of the image. Since we are using a spatial-spectral

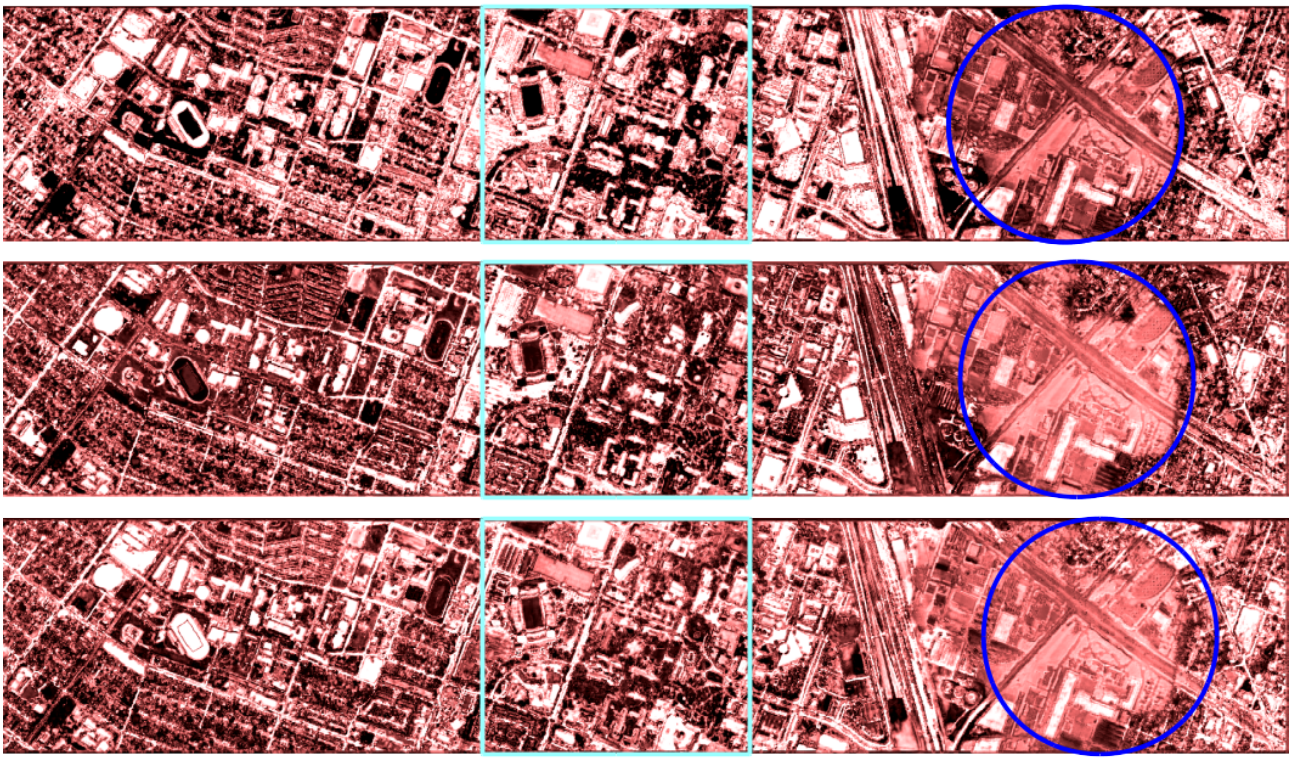


Figure 5.3: 3 Feature maps from the penultimate layer of the Capsule Neural Network, showing the features captured in multiple dimensions / capturing illumination invariant features.

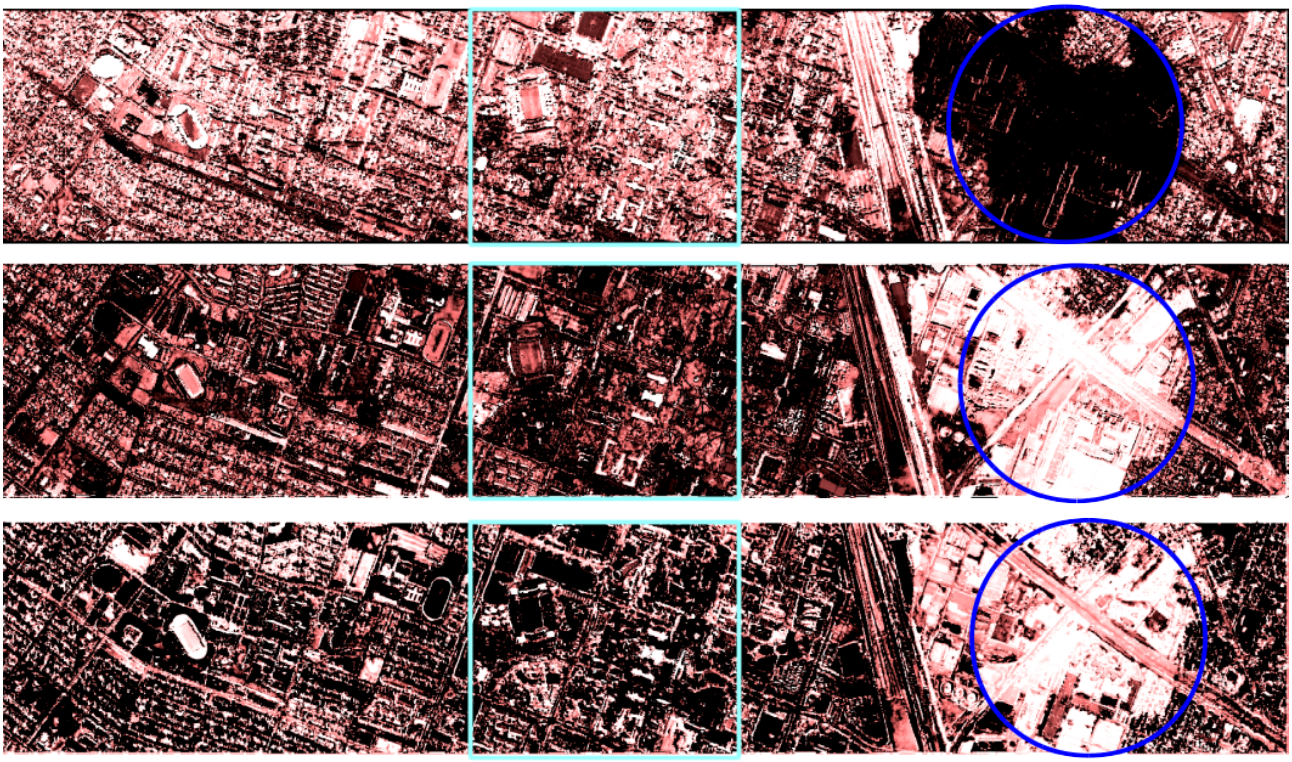


Figure 5.4: 3 Feature maps from the penultimate layer of the traditional 2D-Convolutional Neural Network, showing the features captured in multiple dimensions. Features lacking illumination invariance.

approach for classification, it is important to make sure that the training and the validation / testing datasets do not have any spatial overlap between them. This is done to prevent testing on the training data. After obtaining the point samples, ***window-size*×*window-size*** frames surrounding the individual pixels were acquired. These were then used as the training, validation and testing frames. All the free parameters were obtained by tuning them to result in the highest validation accuracy on the validation data frames.

University of Houston Data

The University of Houston dataset captures the campus and the neighboring urban area, using an ITRES-CASI (Compact Airbone Spectrographic Imager) 1500 hyperspectral imager. It covers 144 spectral bands spanning the visible and near-infrared spectrum from 380 nm to 1050 nm. The image has a spatial size of 349×1905 with a spatial resolution of 2.5 m. ***Grass-Healthy, Grass-Stressed, Grass-Synthetic, Trees, Soil, Water, Residential area, Commercial area, Roads, Highways, Railways, Parking Lot 1, Parking Lot 2, Tennis Courts and Running Tracks*** were identified to be the 15 different classes of interest.

SpecTIR Airborne Data

The data was acquired in a study conducted in 2015, in the coastal wetlands of Galveston, Texas. Wetland vegetation, a crucial part of wetland ecosystem is found to have an immense impact on the species coverage and distribution. Marshes in Mission-Aransas estuary which used to be dominated by smooth cordgrass are now covered mostly by black mangroves. Such sudden drastic changes tend to influence the quality of coastal wetlands, which support a wide variety of marine / aquatic animals as shrimps, fishes, birds and have an impact on the ability of the coastal habitats to buffer wind and wave energy. Therefore, mapping and monitoring the

wetland ecosystems will help us to better manage and monitor the endangered wetland ecosystems.

An airborne hyperspectral image was captured using the ProSpecTIR VS sensor on August 14, 2015. Ranging from 400 nm to 2450 nm, the image was captured at 360 wavelength bands, at a spectral resolution of 5 nm. The radiance data were radiometrically and spectrally calibrated before they were converted to the reflectance data using the ATCOR 4 software. The output reflectance data of multiple flight lines were eventually geo-corrected and mosaiced as an image with spatial coverage of 3462×5037 pixels at a 1 m spatial resolution.

Since labeling samples through photo-interpretation was not possible in this situation, a field survey was made on September 16, 2016. ***Upland grass, St. Augustine grass, Sesbania / rattlebox, Upland tree, Phragmites austrails, Sabal mexicana / palm tree, Spartina alterniflora, Juncus roemerianus, Batis maritima / Distichlis spicata, Distichlis spicata, Baccharis halimifolia***, ***Avicennia germinans / black mangrove, Roads, Sand, Soil, Rocks and Urban constructions*** were identified to be the 17 distinct classes. The dataset contains a total of 7219 labeled samples.

5.4.2 Network Architecture

The Deep Neural Network (DNN) architectures based on Capsule Neural Networks as used for all our Experiments are as shown in Table 5.1.

5.4.3 Experimental Setup

We compare our proposed algorithms with several other state-of-the-art algorithms. The comparisons show that our methods extract better features compared to other methods, as quantitatively our methods result in higher classification accuracy values, and qualitatively our methods extract features which are more appealing vi-

Table 5.1: Network Architecture of Deep Capsule Neural Networks

Dataset	*Layer*	*Kernels*	*Filters*	*Activation*	*Dropout*
	Conv-2D-1	2×2	4096	Yes	50%
	Conv-2D (Primary Capsule)-2	2×2	dimension-of-capsules (80) × number-channels (128)	$ReLU$	No
	Routing+Classification-3	–	dimensions (256) × number-classes (15)	–	–
UH	Decoder-Network-4 (FC-1)	–	dimensions (512)	$ReLU$	–
	Decoder-Network-5 (FC-2)	–	dimensions (1024)	$ReLU$	–
	Decoder-Network-6 (FC-3)	–	Input-Shape	$Sigmoid$	–
	Conv-2D-1	2×2	64	$ReLU$	50%
	Conv-2D (Primary Capsule)-2	2×2	dimension-of-capsules (60) × number-channels (32)	No	No
	Routing+Classification-3	–	dimensions (64) × number-classes (17)	–	–
SpecTIR	Decoder-Network-4 (FC-1)	–	dimensions (8)	$ReLU$	–
	Decoder-Network-5 (FC-2)	–	dimensions (16)	$ReLU$	–
	Decoder-Network-6 (FC-3)	–	Input-Shape	$Sigmoid$	–

FC: Fully-Connected Layer

sually. The training, validation and test frames were generated by using a repeated random subsampling method. We use 10 labeled samples / frames per class, validate on 20 labeled samples / frames per class, and test on 100 labeled samples / frames per class, for all datasets. We found that a frame size of 5×5 works best for our datasets. We run each of the experiments 5 times using a random sample selection strategy and report the average results showing the mean and standard deviations. We compare our methods with - (1) Supervised Spectral classification methods including the KNN and SVM wih RBF kernel; (2) Discriminative Semi-Supervised-Spectral classification methods including Label-Propagation [97] which propagates labels along the high-density areas defined by unlabeled data, Transductive SVM's (T-SVM) [104] which were proposed to modify SVM's with the aim of max-margin classification ensuring minimum number of unlabeled data samples near the margins, Laplacian SVM's (LapSVM) [105] where the loss function is a combination of the supervised loss function of normal SVM's and an additional term which introduces a regularization term on the geometry of both supervised and unsupervised samples by using the graph Laplacian; (3) A Generative Semi-Supervised-Spectral classifier, Ladder Networks [106] which use a discriminative approach to learn from labeled samples and

a generative approach which aims to minimize the difference between encoder inputs and decoder outputs at each stage. All the layers of the encoder-decoder network share lateral connections with each other. For the sake of clarity we disintegrate our approach to a set of basic steps and show the results for each and every step, thereby making it clear about how the addition of an approach / concept improves the final overall classification accuracy on a particular dataset.

5.4.4 Results and Analysis

For the sake of comparison with the proposed Capsule Neural Networks we implement a model using 2D-CNN + Max-Pooling, having the exact same parameters and configuration as the one with Capsule Neural Network, only difference being in the substitution of the routing layer with a max-pooling layer. For classification with 2D-CNN's, we observe that we get performance similar to other baseline algorithms. The proposed Capsule Neural Network performs better than all the proposed baseline methods (including the 2D-CNN's) for Hyperspectral Image Classification, using the two well-known datasets as described. Since Capsule Neural Networks can preserve the spatial information from the original image in the extracted deep features, we also note that hyperspectral images which have high spatial-resolution perform much better during classification using Capsule Neural Networks compared to using the state-of-the-art Convolutional Neural Networks. Thus, the improvement of classification performance for the SpecTIR dataset is much larger than the improvement for UH dataset, as the spatial resolution of the SpecTIR image is higher than that of the UH image, and as more spatial details from the SpecTIR dataset are preserved by the Capsule Neural Networks as compared to the UH dataset. Moreover, as hypothesized we show that for the same set of training samples and same layer parameters, the features learned by Capsule Neural Networks during Semisupervised learning are qualitatively much more robust than those learned by the traditional

2D-Convolutional Neural Networks. Due to the qualitative robustness of the unsupervised features, Semisupervised classification performance using Capsule Neural Networks is much better than compared to the traditionally used Semisupervised 2D-Convolutional Neural Networks. This leads to higher classification accuracy values and extraction of features which are of higher quality.

Table 5.2: Overall accuracies (%) for the University of Houston data.

	Algorithm	***Accuracy***
(Baselines)	***kNN***	67.60±1.0
	SVM	71.67±1.6
	Label Propagation	67.56±0.8
	TSVM	72.17±2.2
	LapSVM	74.29±1.1
	Ladder Networks	72.00±1.2
	Traditional 2-D CNN-Pooling	73.47±1.0
	Semisupervised 2-D CNN-Pooling	81.12±2.0
	Capsule Neural Networks	76.82±1.3
(Proposed)	***Semisupervised Capsule Neural Networks***	**83.00±1.0**

5.5 Conclusion and Future Work

The search space to optimize the network used to perform Hyperspectral Image Classification here is very large. We work within our environment respecting certain computational constraints in order to tune the network parameters. The results as shown here show us that Capsule Neural Networks are able to produce state-of-the-

Table 5.3: Overall accuracies (%) for the Wetlands data.

	Algorithm	*Accuracy*
(Baselines)	***kNN***	66.11±1.4
	SVM	72.72±1.1
	Label Propagation	65.54±1.0
	TSVM	74.87±0.7
	LapSVM	77.17±1.3
	Ladder Networks	64.63±1.5
	Traditional 2-D CNN-Pooling	66.54±1.9
	Semisupervised 2-D CNN-Pooling	75.52±1.6
	Capsule Neural Networks	76.41±1.9
(Proposed)	***Semisupervised Capsule Neural Networks***	**80.95±0.9**

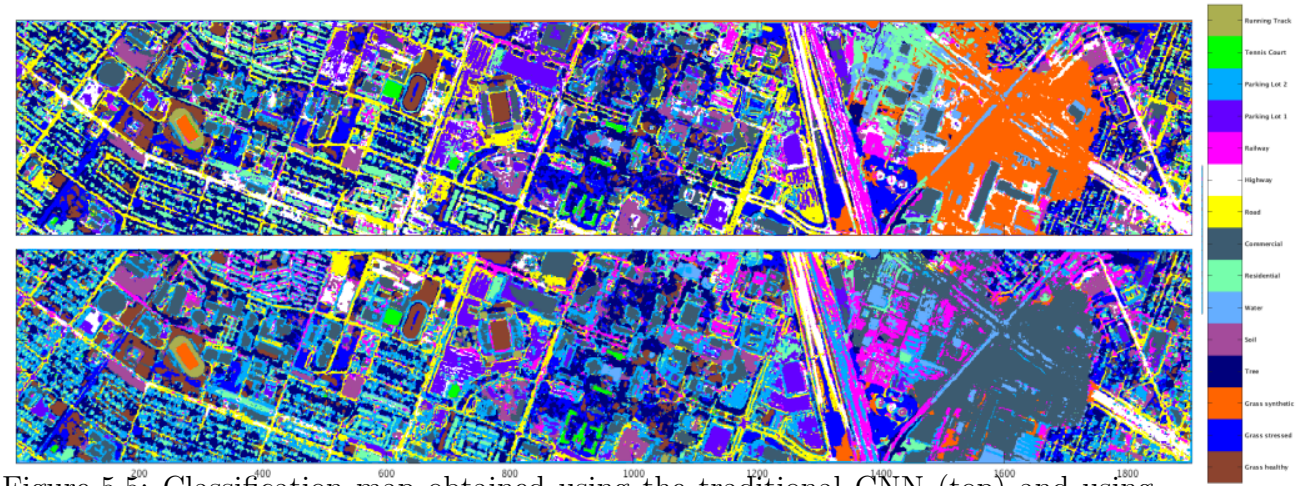


Figure 5.5: Classification map obtained using the traditional CNN (top) and using the proposed Semisupervised Capsule Network (bottom)

art performance on Semisupervised Hyperspectral Image Classification tasks using the two well known datasets. For all cases the performance produced by Capsule Neural Networks is much better than the state-of-the-art 2D-CNN's. This observation is in accordance with the results and observations from the Computer Vision community [1, 14]. Future research concerning efficient and intelligent ways to tune the Capsule Neural Networks will be an interesting domain to explore. As hypothesized, we show that Capsule Neuural Networks outperform the traditionally used 2D-CNN's both for purely supervised as well as semisupervised classification tasks. We show that the quality of the features extracted by the proposed Semisupervised Capsule Neural Networks are much better than those extracted by the traditionally used Semisupervised 2D-Convolutional Neural Networks. We also show that certain feature dimensions of the penultimate Capsule Layer is able to capture illumination invariant features by eliminating the impact of cloud shadows and capturing the shape and size based properties of the objects underneath those shadows. This observation leads us to hypothesize that Capsule Neural Networks will be extremely useful for domain adaptation tasks, as domain adaptation requires us to generate feature properties which are domain invariant. Since some dimensions of the learned filters in Capsule Networks can capture specific property invariant features, as shown in this chapter, we think that it is going to be very beneficial for domain adaptation tasks. Some authors have already started to investigate this for domain adaptation and cross domain learning tasks [129, 130, 131, 132, 133], but we think that the Capsule Neural Networks hold a much greater promise in the near future.

Chapter 6

For single pixel wide labeled datasets: Towards a more robust approach for Semantic Segmentation

Semantic Segmentation of roads after learning road networks is an important domain of research. However, the training data for road segmentation models generally depict only the center of the roads and not the entire roads. This is primarily done in order to reduce the cost of labeling the roads, which differ widely in terms of shapes. Due to the inconsistency between different road geometries (unlike building segmentation, where most of the buildings can be labeled using small polygonal units) it is not possible to label roads using polygonal units, and generally line units are used. Lines to depict center of the roads, used as ground truth, will confuse the deep neural network and train the semantic segmentation models incorrectly, as pixels belonging to the non-central part of the roads are incorrectly depicted as non-roads. In this chapter we propose a method to expand the ground truth to the edge of homogeneous

roads and cover the entire roads in the image, instead of only the center of the roads. Our proposed method is general and can be applied to any images (other than roads), in order to generate abundant ground truth information from single pixel wide supervised labeled images. This idea is of great significance in scenarios where acquiring abundant labeled data is a very expensive process and requires costly human intervention. We evaluate our proposed algorithm on the well known large scale SpaceNet: Vegas Road Segmentation Dataset, SpaceNet: Shanghai Road Segmentation Dataset, as well as the large scale Caracas Road Segmentation Dataset (which captures the city of Caracas in the country of Venezuela, and was captured using the Worldview-2 sattelite by DigitalGlobe. We introduce this Road Segmentation dataset for the first time from our lab located in Oak Ridge National Laboratory).

6.1 Introduction

Barring a few state-of-the-art Road Segmentlation datasets as KITTI [134], most remote sensing imagery data have pixel-level segmentation ground truth labels, which demarcate only the center of the wide roads using a single pixel wide line [135]. Label miss-assignment is a well known problem in the field of road segmentation. The ground truth for road segmentation images only label the center of the roads and mislabel all the other pixels in the roads as non-roads. This drastically degrades the prediction performance, as the model learns to incorrectly assign parts of roads that do not belong to the center of the roads to the background class or non-roads. Whilst, this problem has non been completely eradicated so far, many solutions have been proposed by different researchers working in the field of road segmentation. One such approach is to apply the principle of flood-filling to expand the central one pixel wide ground truths to all the spatially connected pixels having similar color [2]. While this method improves the performance of semantic road segmentation, it is still arbitrary

in a sense where only the color of the pixels are taken into account when determining the road clusters or groups of pixels belonging to the roads. Moreover, the ground truth is still not able to capture the entire roads, and is especially very inaccurate near the edges as shown in Figure 6.1. In this chapter our goal is four fold: (1) Propose a method which will be able to detect roads and background to a much higher degree of precision, from the raw images, by learning only from a limited number of labeled data samples or by learning from single pixel wide supervised datasets, (2) Wide area deployment to expand the labels from their thin one pixel wide versions depicted by the center of the roads, to correctly capture the entire roads/drivable surfaces, (3) Pre-processing of the generated labels to smooth out the center of roads, and also respect the road boundaries/edges and (4) An accelerated city-scale mapping for big data semantic segmentation. We use these newly generated ground truth labels, extracted by using our proposed approach, to train a well known deep network in order to perform semantic segmentation of roads. The prediction and evaluation results from the deep network shows the benefit of using our approach for improving the accuracy of the related semantic road segmentation tasks.

6.2 Related Work

Semantic Segmentation has been studied by researchers since a long time [136, 137], but the recent revival of deep learning methods [77], has seen a diverse set of semantic segmentation tasks and various domains benefiting immensely [138, 139, 140, 141, 142, 143, 144, 145, 146, 147]. Road Segmentation, belonging to the domain of semantic segmentation has also been an area of research for a long time, and in a similar way has also gone through significant boosts in performance improvement over the last few years, especially after the revival of deep learning methods. Several papers including novel datasets in the literature have been proposed with the idea of

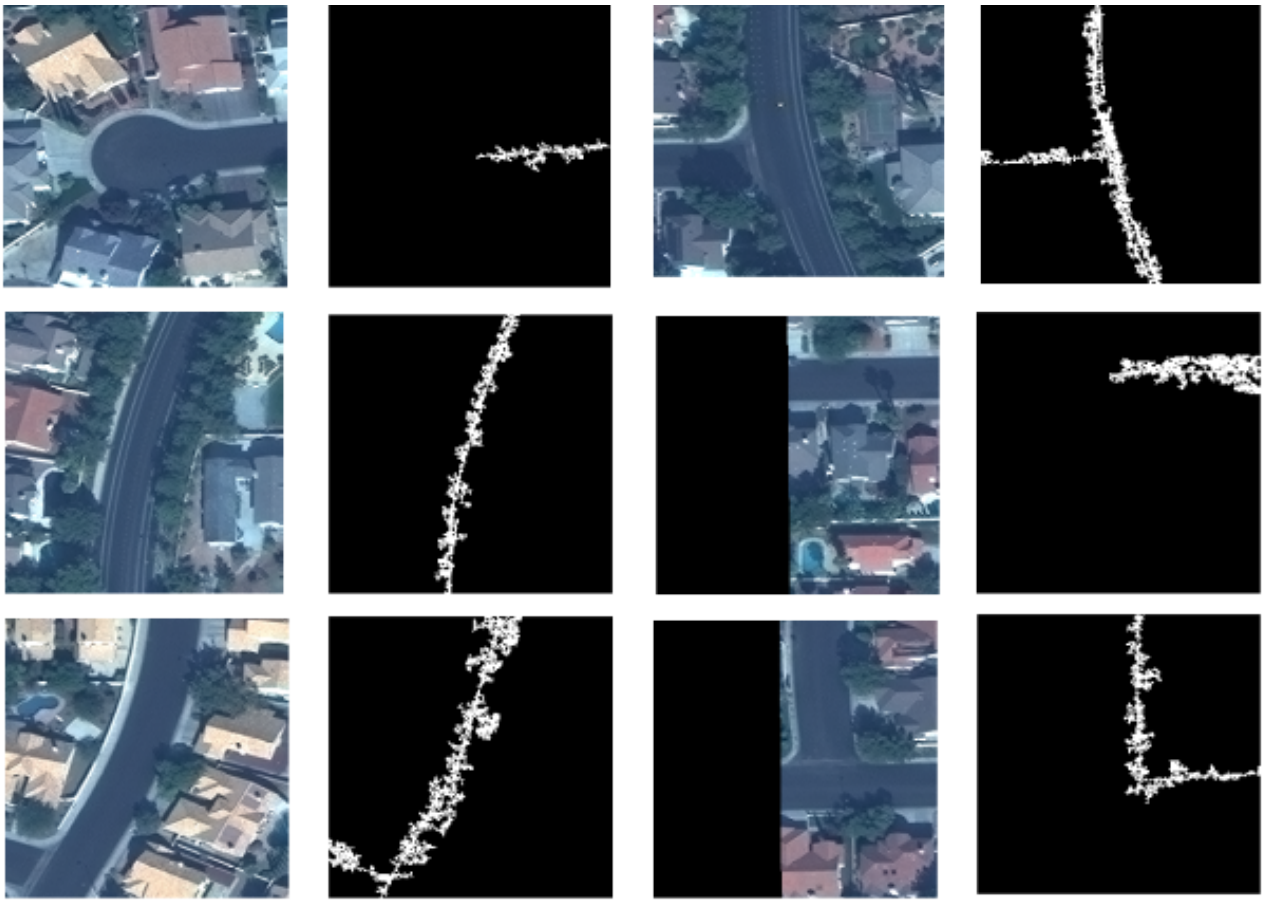


Figure 6.1: Motivation for the proposed work: State-of-the-art expansion of one pixel wide road labels using NVIDIA's flood-filling approach [2]. Image chips are egenrated from SpaceNet: Vegas dataset.

performing road segmentation for — autonomous driving [148, 149, 150, 151], vehicle identification [152, 153, 154, 155, 156, 157, 158], obstacle detection [159, 160, 161, 162, 163] and road-lane-detection [164, 165, 166, 167, 134].

In this chapter we strive towards obtaining correctly labeled ground truth labels for Road Segmentation tasks, by expanding the original single pixel ground truth labels to encompass the entire width of the roads. Our proposed method is especially applicable in scenarios where we have limited number of labeled training data, and require unsupervised learning methods to obtain more labeled training samples, for the purpose of training a deep neural network without overfitting on the limited number of available training samples.

6.3 Proposed Work

6.3.1 Pseudocode for creating clusters corresponding to the single pixel wide ground truths

Algorithm 5 shows the algorithm governing the proposed semi-supervised ground truth expansion method. The generated ground truth along with the original 16 bit images are then used to train a deep neural network as proposed in [3] to perform semantic road segmentation.

6.3.2 Block Diagram of the proposed approach

The Block Diagram Representation of our proposed approach is shown in Figure 6.2. Our proposed algorithm takes the image and corresponding line based single pixel ground truth as the inputs. We expand the single pixel wide ground truths to the edges of the roads, in order to cover the entire roads as shown in the Figure. We then use the newly obtained ground truth, along with the original 16 bit images, in

Algorithm 1: *Pseudo code of the proposed algorithm*

Input:

• RGB Image: $I \in r \times c \times 3$
• 1 pixel wide Ground Truth: $Y \in r \times c$
• Number of clusters: n
• Minimum distance for merging clusters: d
• Kernel window for smoothing: $k \times k$

{*Hierarchial Density Based Clustering Algorithm [168]*}

• Cluster input image I to generate n clusters

{*Merging Based on Wasserstein similarity [169, 170]*}

• Assume the n clusters to represent a distribution for each of the Red, Green and Blue bands For clusters $1, 2, ..., n$, and image pixels $I_1, I_2, ..., n$ which belong to the corresponding clusters respectively. We assume:
for each of *Red*, *Green* and *Blue* color bands:

$P_{1,band} = I_{1,band}$;
$P_{2,band} = I_{2,band}$;
$P_{3,band} = I_{3,band}$;
$\vdots$
$P_{n,band} = I_{n,band}$

end for

Where $P_{1,band}$ represents the color distribution for the Red, Green or Blue colored band of cluster 1, $P_{2,band}$ represents the color distribution for the Red, Green or Blue colored band of cluster 2 and so on.

• Compare each of the obtained n clusters (generated by each band) to calculate the $n \times n$ Wasserstein similarity matrix between clusters (there will be 3 such similarity matrices representing the Red, Green and Blue bands respectively).
• Wasserstein Similarity Matrix between color matrices, for clusters i and j, represented by color distributions P_i and P_j is denoted as:
W: $W(\mathbb{P}_i, \mathbb{P}_j) = \inf_{\gamma \in \prod(\mathbb{P}_i, \mathbb{P}_j)} \ \mathbb{E}_{(x,y) \approx \gamma}[\ ||x - y||\ ]$
• Merge clusters based on the similarity matrix W: $W(\mathbb{P}_i, \mathbb{P}_j)$:
for each of *Red*, *Green* and *Blue* color bands:

for two clusters i, j:
if $W(\mathbb{P}_i, \mathbb{P}_j) \leq d$:
i = j

end for

• Where $\prod(\mathbb{P}_i,\ \mathbb{P}_j)$ is the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_i$ and $\mathbb{P}_j$. Intuitively, from the field of optimal transport, $\gamma(x, y)$ indicates how much "mass" or "earth" must be transported from x to y in order to transform the distribution $\mathbb{P}_i$ into the distribution $\mathbb{P}_j$. The EM distance is then the "cost" of the optimal transport plan

{*Nearest neighbor based smoothing [171]*}

• Find the $m + 1$ samples within the $k \times k$ rectangular kernel window
• Perform nearest neighbor smoothing as follows:
$Y(X_l) = 1/(m + 1) \times ||X_0 + X_1 + ... + X_l + ... + X_m||$
Where X_0, X_1, ..., X_l, ..., X_m represent the $m + 1$ samples belonging to the neighborhood of the pixel X_l, and contained within the $k \times k$ kernel. $Y(X_l)$ is the approximated smoothed value of X_l, obtained by performing the kernel smoothing operation over its neighborhood.
• Where X_0 is the 1^{st} closest neighbor of the pixel X_i, X_1 is the 2^{nd} closest neighbor of the pixel X_i, ... , and X_m is the $(m + 1)^{th}$ closest of the pixel X_i

Output:

• The smoothed wide roads covering the entire width of roads: $\mathbb{L}$

order to train the Deeplab Semantic Segmentation Network [3]. We use very high atrous rates to capture features using larger filters, in order to capture the global details from the original images to a very high degree of precision. The parameters and settings for our experiments are described in details in the Experimental Setup Section of this chapter.

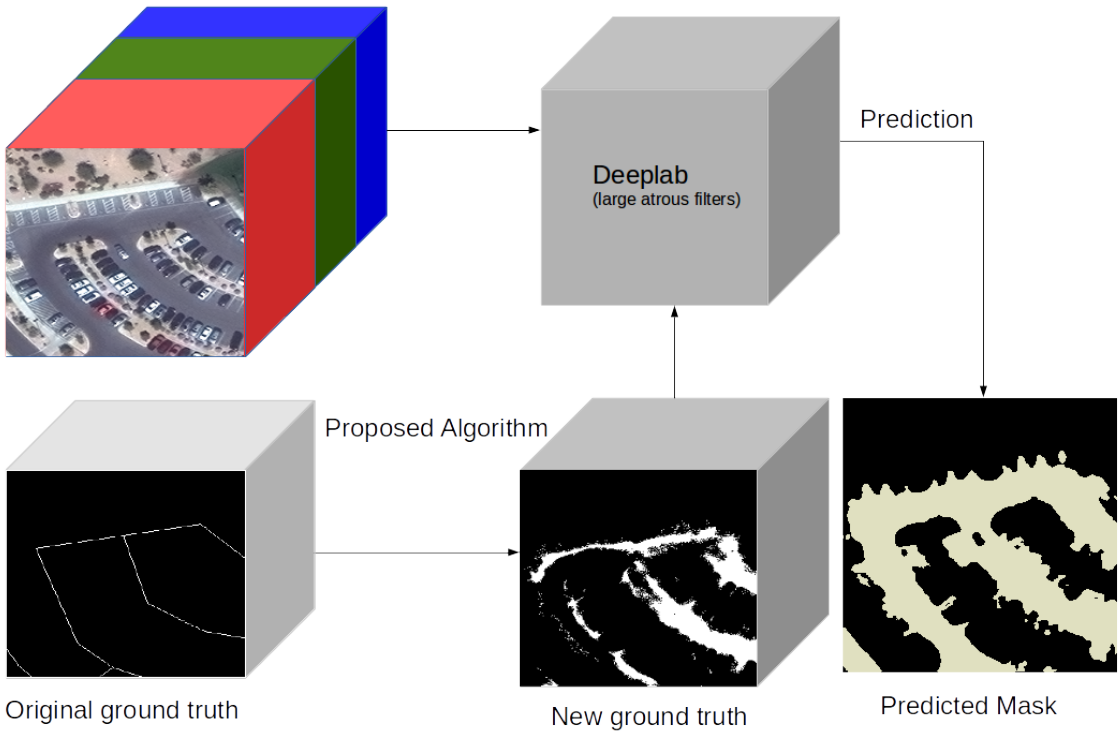


Figure 6.2: Block Diagram Representation of our proposed approach

The block diagram in Figure 6.3 shows an approach from [3] which uses atrous convolutional filters to expand the filter sizes, so that the global properties and spatial context from the original images are captured in the final deep feature space. A number of zeros are appended to the filters in between the non-zero elements of the filters, in order to increase the filter sizes so as to capture lager spatial contexts, without increasing the number of filter parameters. We use this Semantic Segmentation network [3] for performing Semantic Segmentation using the original rich 16 bit image and the expanded labels generated by our algorithm.

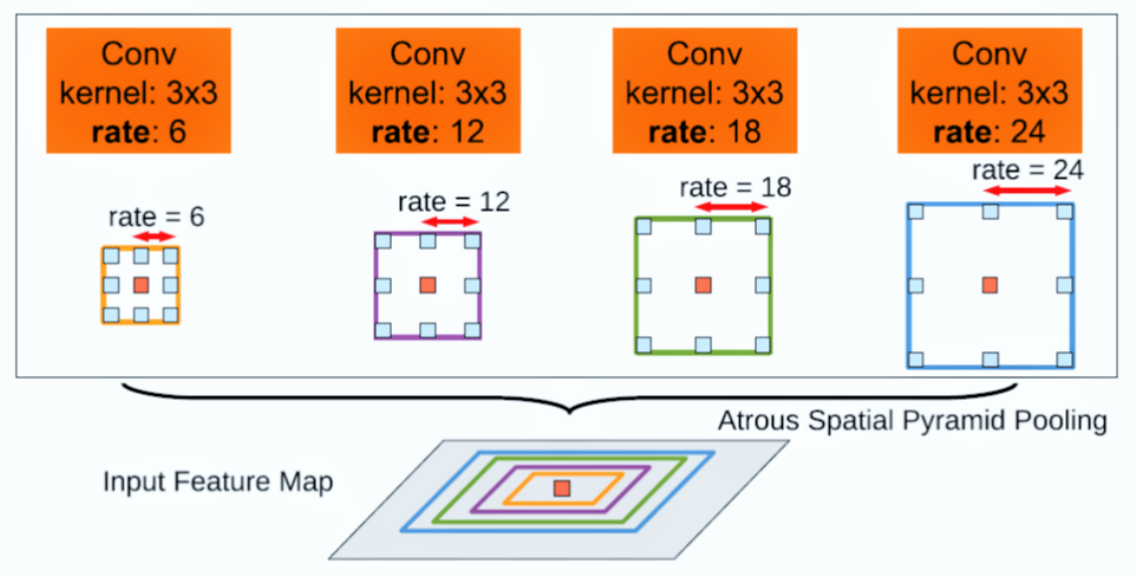


Figure 6.3: Block Diagram from the Deeplab paper [3]

6.3.3 Motivation

Since its inception [172, 173, 76], deep learning has been known to suffer from some well-known problems as: The lack of labeled training data leading to overfitting of the learned models on the training data and producing weak models due to overfitting on single pixel wide labeled datasets. In this chapter we try to analyze the cases where we have single pixel wide labeled ground truths denoting only the center pixel of the roads and expand the ground truth to cover the entire roads using our proposed algorithms. Figure 6.4 depicts the effectiveness of our approach to expand the single pixel wide thin label ground truths to the entire width of the roads. This helps us to reduce the mislabeled data, and effectively helps to improve the robustness of the deep learning model which is trained based on these labels and the corresponding images.

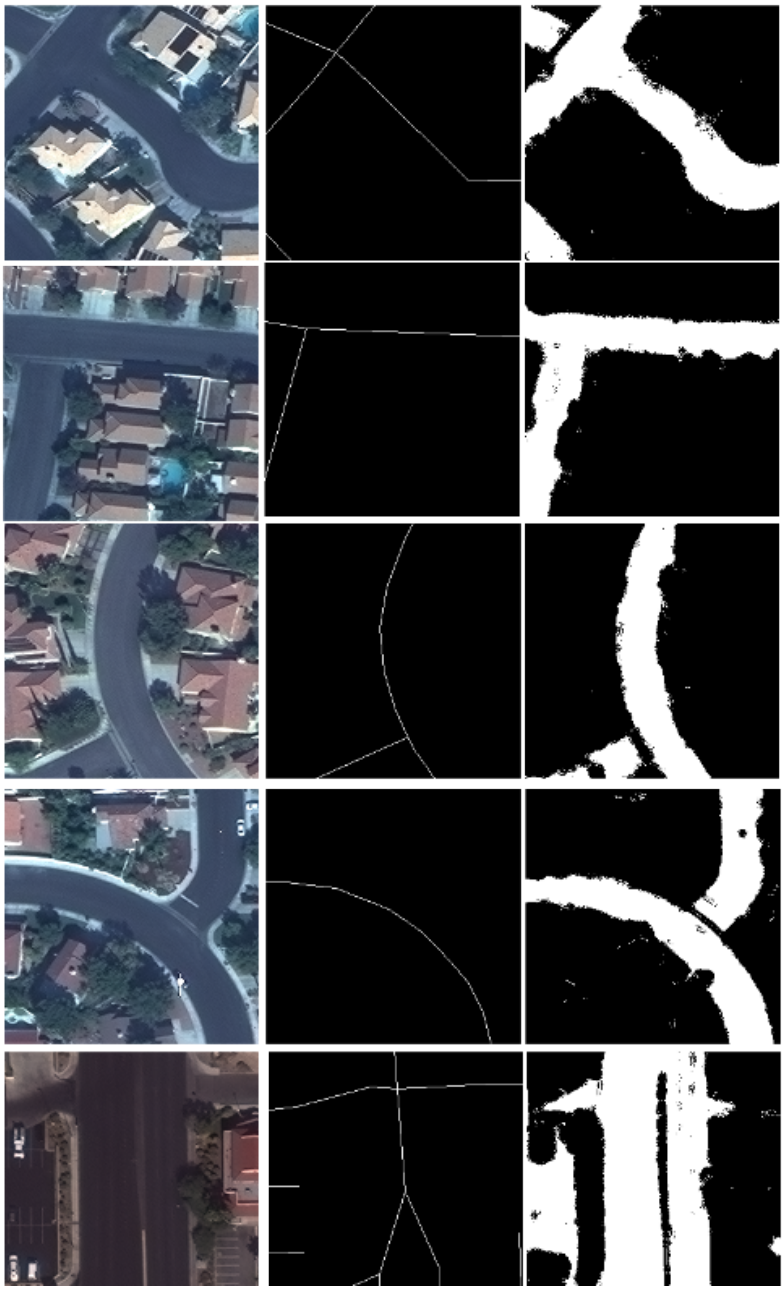


Figure 6.4: RGB images from SpaceNet3: Vegas Dataset (left), corresponding 1 pixel wide ground truth labels for roads (middle), generated thick labels covering entire roads using our proposed approach (right)

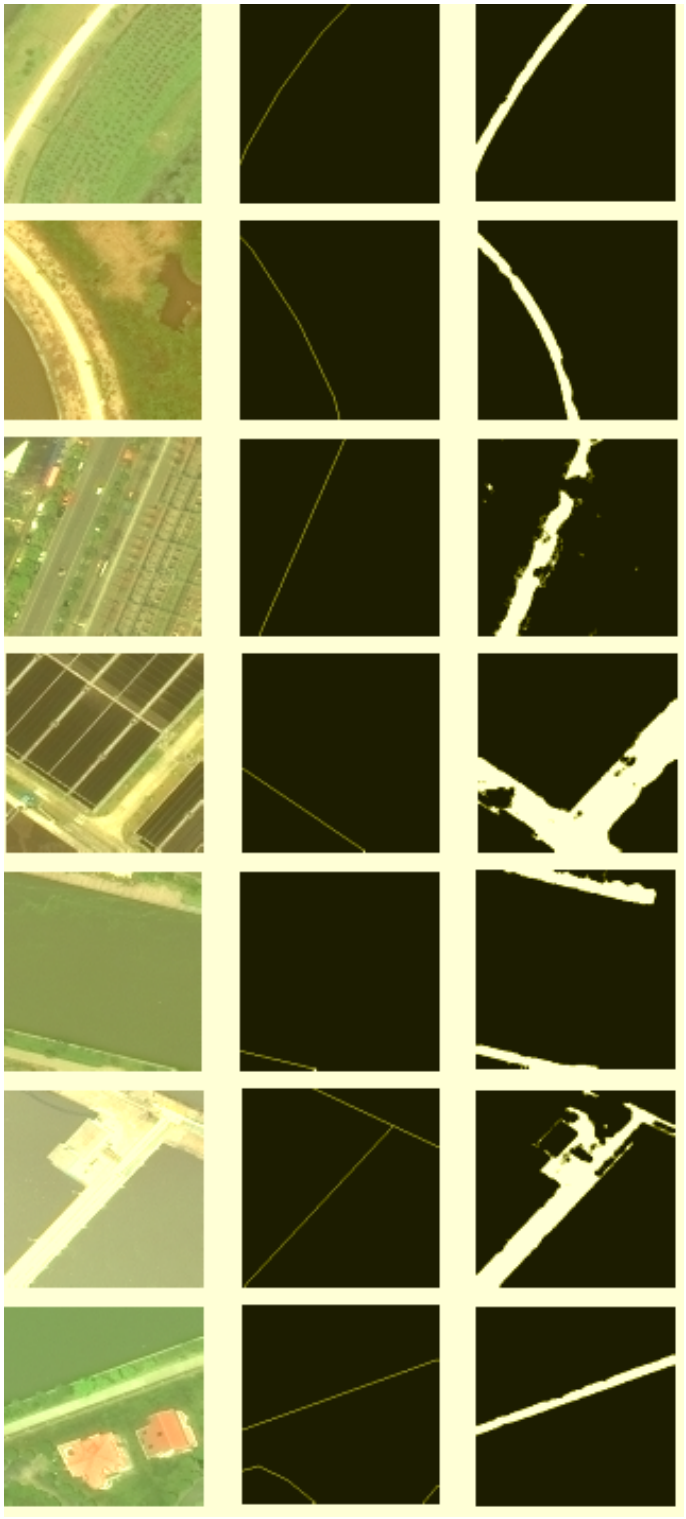


Figure 6.5: RGB images from SpaceNet3: Shanghai Dataset (left), corresponding 1 pixel wide ground truth labels for roads (middle), generated thick labels covering entire roads using our proposed approach (right)

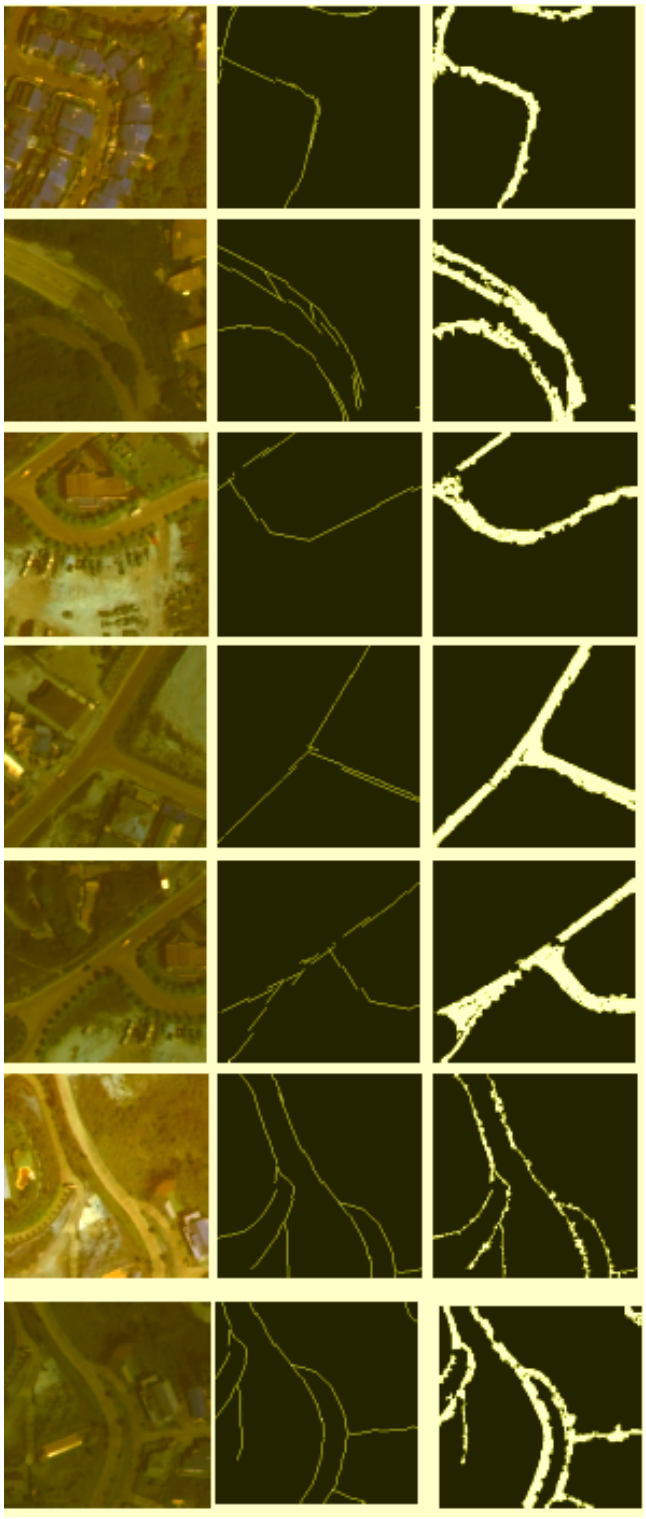


Figure 6.6: RGB images from Venezuela: Caracas Dataset (left), corresponding 1 pixel wide ground truth labels for roads (middle), generated thick labels covering entire roads using our proposed approach (right)

6.3.4 Spectrally pure Clustering

Clustering is an unsupervised technique used for grouping spectrally similar pixels together, so as to preserve the purity of the groups in order to avoid mixing of pixels from different objects. Many papers have been proposed in the literature as [174, 175, 176, 177, 178, 179, 180, 181, 182], which analyze various forms of clustering, so as to group spectrally similar pixels together using different statistical measures. In this chapter we use the Hierarchial Density Based Clustering Algorithm as proposed in [168] to perform clustering. The Hierarchial Density Based Clustering Algorithm is based on the Density Based Spatial Clustering of Applications with Noise (DBSCAN [183]) Algorithm. It converts the DBSCAN algorithm into a hierarchical clustering algorithm, and then uses a technique to perform clustering based on stability of the clusters. We make sure that the extracted clusters are very small so as to avoid obtaining clusters which have pixels from multiple objects / classes.

Known statistical properties governing data distributions as the density of groups of pixels belonging to same objects are used as paradigms to formulate algorithms to perform unsupervised clustering of images. However, most of the existing methods suffer from *"flat"* labeling of the data objects (*DBSCAN [183]* and *DENCLUE [184]*), based on a global density threshold. A global density threshold cannot lead to the generation of optimum clusters in datasets having clusters belonging to very different density levels, other methods as *gSkeletonClu [185]* are not able to automatically extract clusters which can be interpreted easily so as to represent the most significant clusters, other methods as *gSkeletonClu [185]* are limited to work for specific classes of problems as networks, and as *DECODE [186]* and *Generalized Single Linkage [187]* work for point sets in the real coordinate space. Most of the methods depend on multiple critical input parameters which are very difficult to tune [183, 184, 186, 187, 188]. To solve these problems the authors of [168] propose a clustering method which can work automatically to find clusters of widely varying densities from images,

requiring only the minimum number of samples in the smallest cluster as the input. The authors of [168] further show that their method is relatively tolerant to changes in the input parameter.

6.3.5 Wasserstein Distance based merging of clusters

Wasserstein distance or Kantorovich-Rubinstein metric is a distance function defined between distributions on a given metric space M. Intuitively, if each distribution is viewed as a unit amount of "dirt" piled on M, the metric gives us the minimum cost of turning one pile into the other, or turning one distribution into the other. This metric is assumed to be the amount of dirt that needs to be moved times the mean distance it has moved. As a result of this analogy this distance if also popularly known as the Earth mover's distance in the field of computer science. We obtain small clusters to preserve the spectral purity of the clusters as explained in the preceding paragraphs. But such pure and small clusters do not accurately represent the entire road from the original images. For this reason, we merge the clusters based on a distance metric governed by the fundamentals of the optimal transport theory, known as the Wasserstein distance. The problem of optimal transport theory was first defined by french mathematician Gaspard Monge in 1781 [189]. In simple words, it formulates a way of redistributing mass, such as a pile of soil and optimizes how that mass can be transported or reshaped to form a mound with minimal effort. This particular problem remained unsolved for a period of 200 years (it was not even known whether this problem was solvable during those years), until recent mathematical advancements in 1980's and 1990's. Since then the field of optimal transport theory has flourished and it has been applied to several other relevant domains of research as: PDE's, geometry, statistics, economics and image processing [190, 191, 192, 193, 194, 195]. More recently, it has gained immense popularity and relevance in the field of deep learning [196, 197, 198, 199].

We use the Wasserstein distance metric to calculate the similarity between different cluster sets and merge the cluster sets which are similar to each other. In this paragraph we show in the figures how the Wasserstein similarity metric looks for the similar and dissimilar clusters. We see that clusters which have visually similar objects are quantitatively deemed to be similar by the Wasserstein similarity metric, and the clusters which are visually dissimilar are quantitatively deemed to be dissimilar. The Wasserstein distance metric would then be given by

$$W(\mathbb{P}_i, \mathbb{P}_j) = \inf_{\gamma \in \prod(\mathbb{P}_i, \mathbb{P}_j)} \mathbb{E}_{(x,y)\approx\gamma}[\ ||x - y||\ ]\ , \tag{6.1}$$

where $\prod(\mathbb{P}_i,\ \mathbb{P}_j)$ is the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_i$ and $\mathbb{P}_j$. Intuitively, from the field of optimal transport, $\gamma(x, y)$ indicates how much "mass" or "earth" must be transported from x to y in order to transform the distribution $\mathbb{P}_i$ into the distribution $\mathbb{P}_j$. The EM distance is then the "cost" of the optimal transport plan.

Benefit of Wasserstein Distance: The primary benefit of using the Wasserstein distance is that the distance metric or similarity between individual clusters is calculated irrespective of the number of samples in the individual clusters. Since this distance measures the distance between multiple distributions irrespective of the number of samples in each distribution, it is ideal for our case. This is because the clusters generated from our experiments are very heterogeneous with respect to the number of samples in each cluster, and we want to measure the similarity between such clusters irrespective of the number of samples in the individual clusters.

After obtaining the spectrally pure clusters we assume each of the three bands: Red, Green and Blue bands for each of the clusters to represent a color distribution. Since it is known that clusters can have widely varying density of samples depending on the scale of the object they represent, we propose the comparison of the color

distributions (and not a direct comparison between the widely varying number of samples representing different clusters) obtained from the clusters. The Wasserstein distance metric is a well known distance metric used for the comparison of multiple distributions, and it calculates the amount of "work" required to convert one distribution into the other. In our proposed work we use this similarity metric to compare between the Red, Green and Blue bands of color distributions obtained from the different clusters. A small Wasserstein distance for each of the three color bands imply that the distributions governing the clusters are similar. We merge the clusters if they belong to similar distributions, else they remain as they are. This method helps us to expand the ground truth for the roads to cover the entire roads, starting from the one pixel representing the center of the roads, as represented by the Figures. The distance based merging of the clusters help us to connect the similar clusters to each other.

6.3.6 Nearest Neighbor based smoothing

Even after merging the small clusters to form large groups of objects, we still have noisy labels. To eliminate this we implement a smoothing constraint and smooth out the noisy labels. We implement this smoothing algorithm since we realize that roads are mostly smooth except at the boundaries/edges. We implement a smoothing algorithm with a small rectangular kernel, having a size of 7×7, so as to smooth the pixels belonging to the road and simultaneously respect the road boundaries/edges. This smoothing operator leads us to get the labels as shown below. We finally use this version of the generated ground truth, along with the corresponding original 16-bit images to train the back-end semantic segmentation network.

A kernel smoother is a statistical technique to estimate a real valued function f: $\mathbb{R}^p \rightarrow \mathbb{R}$ as the weighted average of the neighboring observed data. The weight is defined by a kernel K, such that closer points are given higher weights. The

estimated function obtained as a result is smooth. The nearest neighbor smoother is based on the following idea: For each point X_i, take m nearest neighbors, and estimate the value of $Y(X_i)$ by averaging the values of these neighbors. The kernel size k is determined empirically from the dataset. Higher the value of k, larger is the kernel size, resulting in smoother images. A very large value of k, causes the edges and separate objects in the images to smoothen out. Thus, we empirically find the optimum value of k so as to smoothen the roads, as simultaneously respect the road boundaries. Mathematically,

$$Y(X_i) = 1/(m+1) \times ||X_0 + X_1 + ... + X_i + ... + X_m|| \ , \tag{6.2}$$

where X_0 is the 1^{st} closest neighbor of the pixel X_i, X_1 is the 2^{nd} closest neighbor of the pixel X_i, ... , and X_m is the $(m+1)^{th}$ closest of the pixel X_i.

6.3.7 Deeplab based training

Most of the deep networks proposed since [172, 173, 76], have been known to be known to suffer from the problem of overfitting. To address this issue and also incorporate a degree of translational invariance to the extracted deep features, most of the state-of-the-art deep networks utilize the pooling layers (either max-pooling or average-pooling). But since its inception and more recently, the science governing such arbitrary pooling operations have been questioned by a large number of scientists/researchers [13, 110, 200, 1]. More recently, many of the proposed deep neural networks as Capsule Neural Networks [1, 14], have been proposed to replace this arbitrary idea of pooling with a statistically and scientifically sound approach, which is governed by a routing algorithm between deep layers. In the field of semantic segmentation, researchers have replaced the pooling operations with strided convolutional networks, moreover to preserve or utilize a large field of view from the original

images, atrous convolutions have been used within the proposed deep networks.

Atrous convolutions since their inception, have been used in the field of wavelets [201, 202, 203, 204], for the purpose of enlarging the field of view of the filters to incorporate larger global context within the learned deep features, without increasing the filter parameters. The term *"a-trous algorithm"* is a French word, meaning *"holed algorithm"*. In simple words, it grows the small sized filters by incorporating zeros in between the filter elements, thereby increasing the filter sizes, without increasing the number of non-zero parameters, which are required to represent the filters. More recently, they have been used to capture features at a large field of view, so as to capture the important global details from the images without compromising the local details, in the field of deep learning [3]. The primary benefit of using atrous convolutions in deep networks is that they allow us to increase the filter size so as to capture global details from training images, by increasing the field of view, without increasing the number of parameters in the deep neural networks. This helps the deep network to capture the important global details from the images and capture the spatial context of the objects with respect to each other without requiring large number of training parameters [3], which as seen from literature inevitably leads to the problem of overfitting on the training dataset [205, 206, 9].

Deeplab [3] was recently proposed for performing Semnatic Segmentation, and it incorporates both strided convolutions and atrous convolutions as discussed before. For performing the backend segmentation utilizing our proposed algorithm, we use this approach as the state-of-the-art method for performing Semantic Segmentation of Roads.

6.3.8 Training and Testing using 16 bit images

Most of the recent state-of-the-art images are aquired using the 16 bit data format. The 16 bit data format is exceptionally rich from the perspective of representing the

variation of spectral content, and also from the perspective of capturing the fine spatial details from any particular scene of interest. Any 16 bit image can represent $65536^3 = 2.8147498e + 14$, distinct shades of colors.

However, in most of the backend deep learning libraries as Tensorflow, most of the default configurations of the modules operate using 8 bit images. 8 bit RGB images can only represent $256^3 = 16777216$ distinct color shades. Due to this reason most of the deep learning approaches including the recently proposed state-of-the-art Semantic Segmentation approaches [3, 207], downconvert the original spectrally and spatially rich 16 bit images to the corresponding 8 bit images, in order to learn the corresponding deep features using the in built deep learning libraries as tensorflow, in which the default configuration can handle only the 8 bit downconverted data. Most of the recently proposed Deep Learning Frameworks have enough parameters to learn direct representations of the large, rich and detailed 16 bit original images, and downconversion of the original images to 8 bit is not required. In simple words, downconverting the rich 16 bit representations to the corresponding 8 bit representations result in a loss of useful spatial / spectral information, and decreases the robustness of the learned deep features.

We have seen that 16 bit representations are able to represent 16777216 times more color variations compared to the corresponding 8 bit representations. To address this issue, we modify the default configuration of the original Tensorflow library, in order to utilize the original 16 bit images. Our network directly learns the deep features from the original 16 bit images and the corresponding thick labels as generated by our proposed algorithm. We show the impact of learning from the original detailed 16 bit images, instead of the downconverted 8 bit images, using Figure 6.7. For both the cases, the newly generated expanded ground truth, obtained from our proposed algorithm is used for training the deep network. We notice that the predictions when learning is done using the original 16 bit images, are much detailed, less impacted

by noise, and are able to capture detailed representations from the original images, when compared to the predictions when learning is done using the downconverted 8 bit images. More precisely, for the image at the top the 16 bit predictions are much less noisy compared to the 8 bit predictions. For the second image from the top, the 16 bit predictions represent the edges and the crevices of the parking lot in a much detailed manner compared to the corresponding 8 bit predictions. For the third image from the top, the 16 bit prediction accurately captures the road around the tree, at the bottom left end of the image, in a much better manner compared to the corresponding 8 bit prediction. For the 8 bit image, the road around the tree is mislabeled as non-road, and this erroneously creates a detachment in the generated prediction labels for the road. For the bottom most image, the 16 bit prediction is again much less impacted by noise than the corresponding 8 bit prediction.

6.4 Experimental Settings and Datasets

Modern satellite images can capture detailed images of entire countries. Millions of miles of Roads still remain unmapped in those satellite images. Non-profit organizations as - The Humanitarian Open Street Map Team, The Missing Maps Project have been mapping large areas from satellite images. Machine Learning techniques hold a great promise of accelerating the process of road mapping, by automating the process and requiring minimal human intervention. Advancing automatic feature extraction and road segmentation algorithms will tremendously help in formulating disaster responses and developing disaster management strategies. It would help to combat natural disasters as some of the recent natural disasters: The recent flooding in Bangladesh, hurricane Harvey in Texas, hurricane Irma in Florida and hurricane Maria in Puerto Rico. It will also significantly boost in unleashing the power of robust state-of-the-art machine learning algorithms applied to remote sensing applications

in both the private and public sectors. The SpaceNet Road Detection and Routing challenge was designed to assist the development of techniques for generating road maps from high quality satellite images [135]. The SpaceNet Roads dataset was especially created for this purpose. The dataset maps 8000 km of roads *centerlines*, which is the ground truth for the dataset. All roads were digitized from the existing SpaceNet data: 30 cm GSD worldview 3 satellite imagery, over the four cities of — Las Vegas, Paris, Shanghai and Khartoum.

SpaceNet Road Segmentation Dataset: Vegas dataset

Las Vegas is one of the four cities captured by the road segmentation dataset. Las Vegas has a total of 3685.0 km of roads. The data is distributed as 1300×1300 pixelated image tiles, of 16 bit images. Figure 6.4 shows the results of expanding the 1 pixel wide lined road labels using our proposed approach, when the SpaceNet: Vegas Road segmentation image dataset is used as the input images.

SpaceNet Road Segmentation Dataset: Shanghai dataset

Shanghai is one of the four cities captured by the road segmentation dataset. Shanghai has a total of 3537.9 km of roads. The data is distributed as 1300×1300 pixelated image tiles, of 16 bit images. Figure 6.5 shows the results of expanding the 1 pixel wide lined road labels using our proposed approach, when the SpaceNet: Shanghai Road segmentation image dataset is used as the input images.

Venezuela Caracas Road Dataset

High quality spatially rich 16 bit images capturing the capital city of Venezuela i.e. Caracas has been captured using the WorldView-2 dataset by DigitalGlobe. This dataset is being first introduced by our laboratory located at the Oak Ridge National Laboratory in this thesis. We actively participated in creating the ground truth road

maps for this dataset. Figure 6.6 shows the results of expanding the 1 pixel wide lined road labels using our proposed approach, when the Venezuela: Caracas Road segmentation image dataset is used as the input images.

6.5 Results and Analysis

In Figure 6.8 the numbers below each image show the Wasserstein dissimilarity metric between images from the top row and other images, for each of the Red, Green and Blue Bands.

Table 6.1: Overall Mean-IOU (%) for the SpaceNet3 Las Vegas data.

Algorithm	***Accuracy***
NVIDIA's Floodfill *	49.63
Proposed Approach	**65.42**

* https://devblogs.nvidia.com/solving-spacenet-road-detection-challenge-deep-learning/

Table 6.2: Overall Mean-IOU (%) for the Caracas data.

Algorithm	***Accuracy***
NVIDIA's Floodfill *	74.39
Proposed Approach	**80.87**

* https://devblogs.nvidia.com/solving-spacenet-road-detection-challenge-deep-learning/

6.5.1 Challenges for large scale road segmentation

Large scale road segmentation datasets generally have high intra-class variance between samples. The SpaceNet dataset as used in this chapter, has 7 different classes of interest within the general road class. The general roads can be further separated into: Motorways, Primary Roads, Secondary Roads, Tertiary Roads, Residential Area

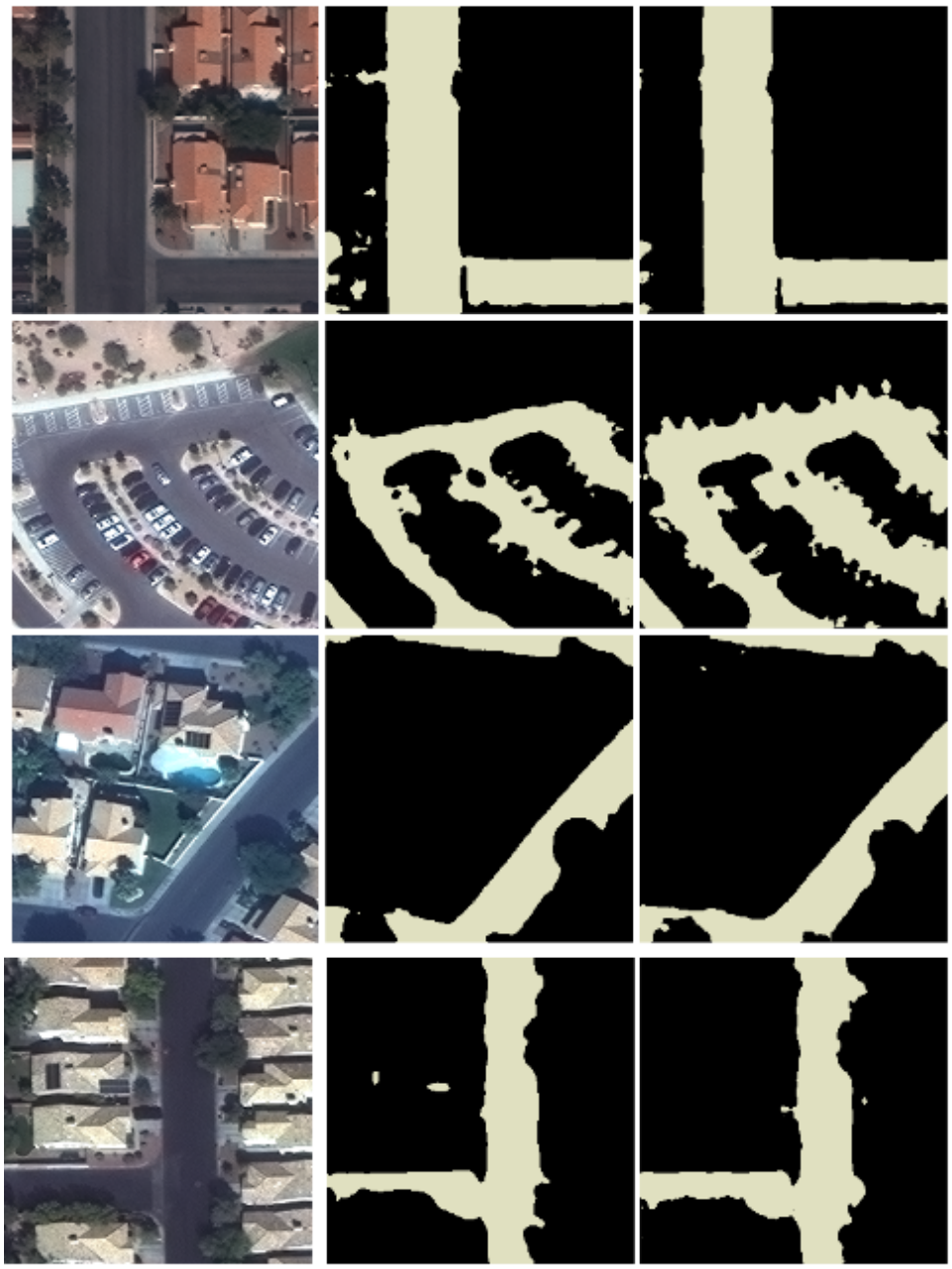


Figure 6.7: Predictions by the proposed method for Vegas Roads. Original RGB images (left), Predictions when training and testing using downconverted 8 bit images (middle) and original 16 bit images (right).

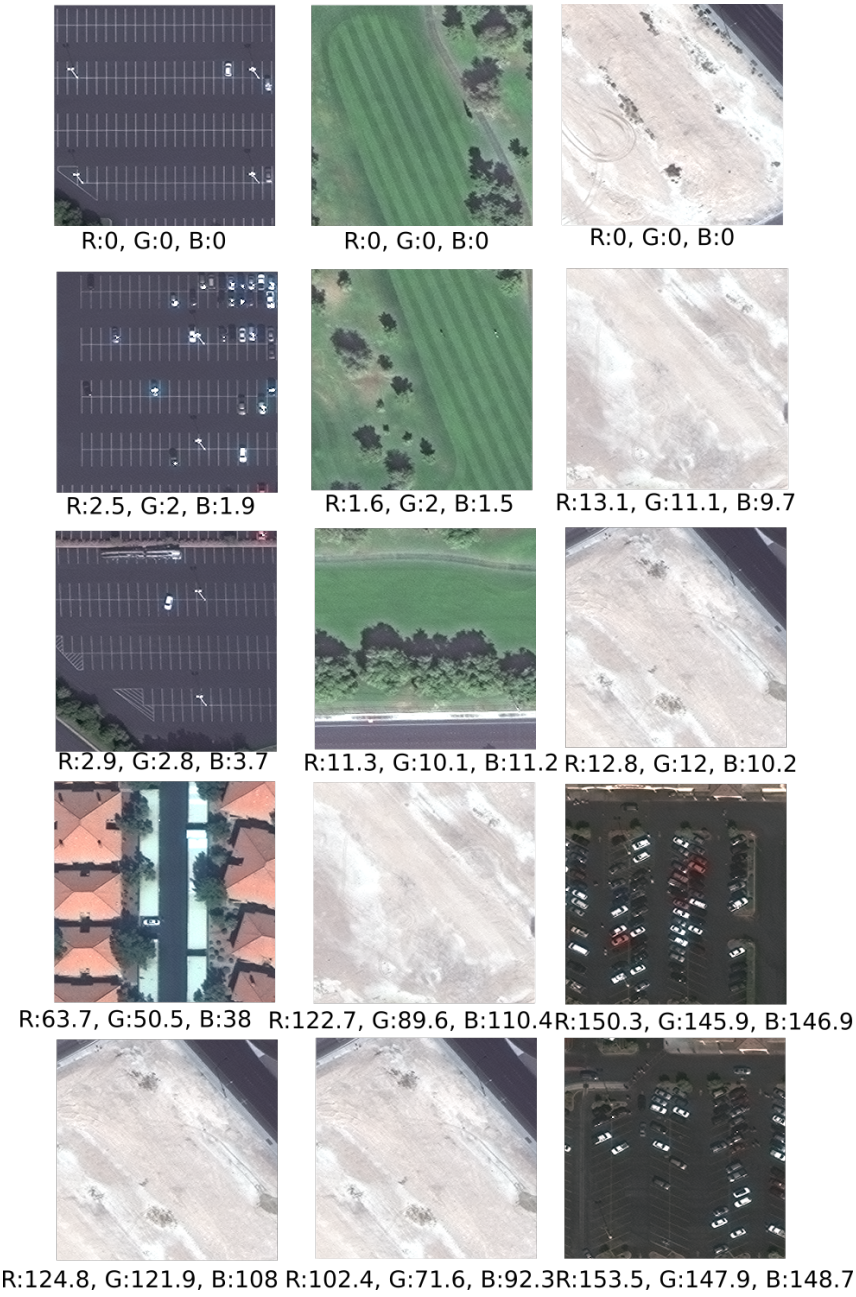


Figure 6.8: Wasserstein Distance Matching between RGB images. Original RGB images (top row), RGB images for comparison (all other rows). Each column shows a distinct comparison.

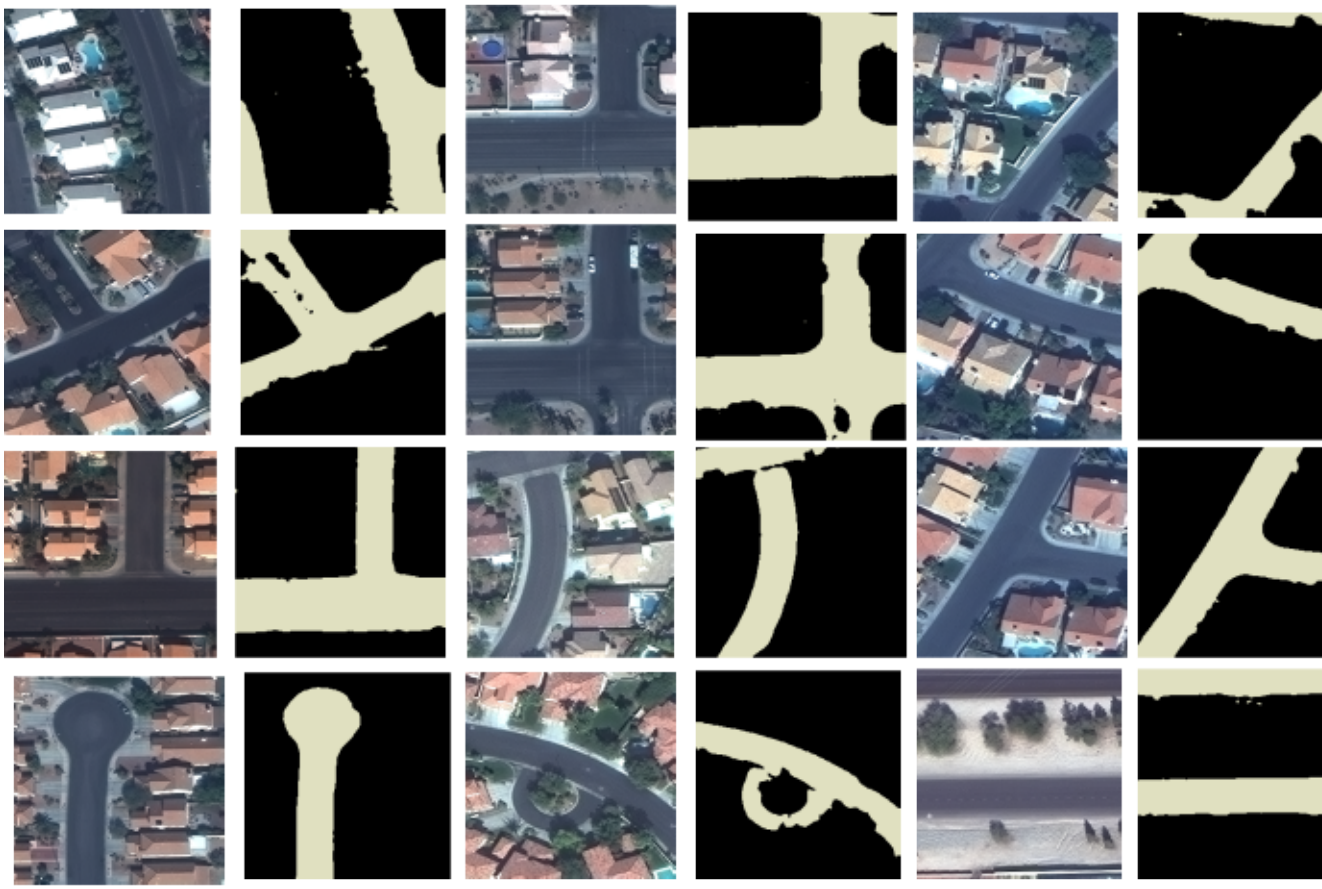


Figure 6.9: Visualization of predictions by the proposed method for SpaceNet, Vegas Road Segmentation Dataset when trained and tested with 16 bit images.

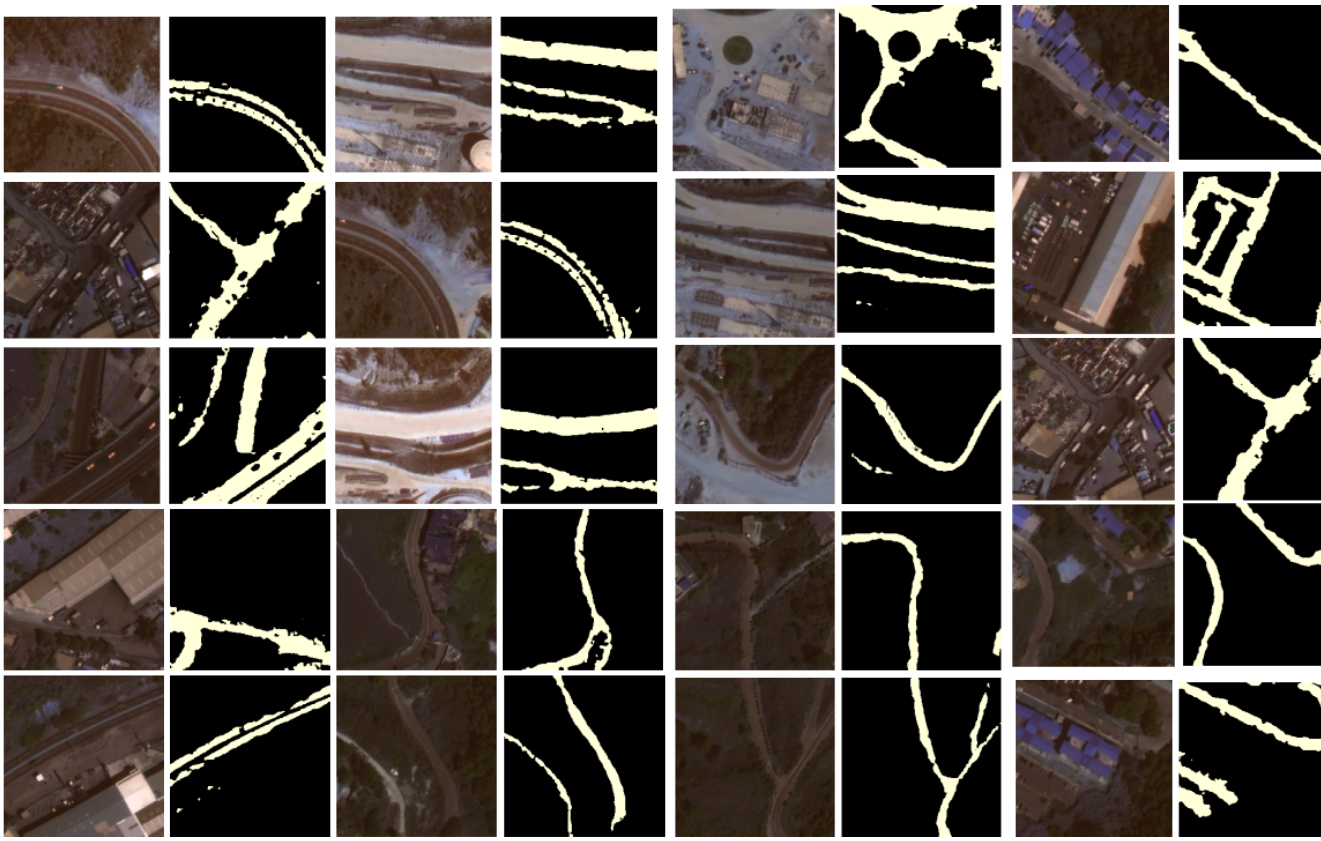


Figure 6.10: Visualization of predictions by the proposed method for in house Venezuela: Caracas Road Segmentation Dataset when trained and tested with 16 bit images.

Roads, Unclassified Roads and Cart Tracks. The high degree of intra-class variance makes it difficult for any feature extractor or classifier to extract robust features from the original images. This problem can be avoided by using a large number of training samples. Using a large number of training samples, will in turn make the process of training the deep network very slow. To avoid this problem [3] uses byte array data types for training, validation and testing of the deep networks. For our proposed work, as described in this chapter, we also follow the same approach.

6.5.2 Wasserstein Distance based similarity

Figure 6.8 shows the Wasserstein distance matching between several images. The top row represents 3 distinct RGB images. Each column shows a distinct comparison between the image from the top row and all the other images below the top row, which belong to the same column. The dissimilarity metrics, as shown below each image for each of the Red (R), Green (G) and Blue (B) bands individually, clearly show that visually similar images have a low dissimilarity value between them. Images which are visually very different from each other have a high value of Wasserstein dissimilarity metric. As described in previous paragraphs, we merge similar clusters using this Wasserstein distance metric, and let the dissimilar clusters remain separated.

6.5.3 Visualization of Road Segmentation

Figure 6.9 and Figure 6.10 shows the 16 bit RGB images and the corresponding segmentation mask predicted by our proposed algorithm as described in this chapter. We notice that the generated or predicted segmentation masks are of very high visual quality, such that they are able capture the finely detailed representations of the roads from the original RGB images.

6.6 Conclusion and Future Work

In this chapter we present a novel method to expand the road segmentation ground truth from the original 1 pixel thin labels, in order to cover the entire roads instead of the center of the roads. Previously, this has been done by [2]. This helps to correctly label the incorrect ground truth pixels and helps to train a more robust model. We show that our methods generate visually accurate segmentation masks.

Chapter 7

Conclusion and Future Work

In this thesis we focus on developing novel Spatial-Spectral Semi-Supervised machine learning techniques used for image analysis. We show that the proposed approaches which use the inherent structure of the underlying unlabled data samples can train feature extractors and classifiers which are much more robust than the corresponding learning methods which make use of only the labeled data samples. Furthermore, we clearly show that our proposed Spatial-Spectral Semi-Supervised Deep Learning methods can perform better than the traditional machine learning methods and corresponding state-of-the-art Deep Learning methods. Another major observation of this thesis is that using the Spatial properties of remote sensing images can help us to learn more robust feature extractors and classifiers compared to using only the Spectral properties of the data. From Chapter 5 of this dissertation, we also observe that by using intelligent pooling mechanisms and Semi-Supervised Deep Learning strategies we can train more robust deep feature extractors and classifiers. In Chapter 6 of this dissertation we also explore novel Semi-Supervised Deep Learning strategies for large scale city wise Semantic Segmentation problems. The task under consideration being Semantic Segmentation of Roads. We show that by using our novel techniques, we can segment the roads from the images in a much robust

manner compared to using the original one pixel wide supervised ground truth labels.

A general observation in the the field of Deep Learning is: Since the recent revival of Deep Learning methods, researchers have started to explore the use of unsupervised data to train their models. This is primarily done in order to exploit the inherent / underlying structure or distribution of the data so as to prevent overfitting of the deep learning models on the limited number of training samples. In this thesis our focus is primarily on exploring Semi-Supervised Learning strategies for traditional machine learning methods and novel deep learning methods. The field of Semi-Supervised Deep Learning has received a significant boost in terms of novel important contributions, but we believe that there is still a large space remaining for researchers to explore in this particular field.

Chapters in this dissertation appear in (or are in preparation for submission) as the following publications [28, 208, 209].

Assumptions and Limitations:

• Chapter 2 of this dissertation assumes that the unlabeled samples in the image can be discriminated from each other based on a similarity metric. The choice of the similarity metric which governs the back-end superpixel algorithm determines the effectiveness of the proposed algorithm.

• Chapter 3 of this dissertation has the same limitations as Chapter 2, as mentioned before.

• Chapter 4 of this dissertation uses a Siamese Neural Network for training the Deep Network. It assumes that both the branches of the Siamese Network will ultimately converge in the same direction. If the branches of the Siamese Network learn opposing features, the the Network will fail.

• Chapter 5 of this dissertation uses a Capsule Neural Network to train the Deep Network. A limitation of this chapter is that we did not experiment with the depth of the Capsule Network, due to time constraints. It would be interesting to observe whether the features become more robust if the depth of the network is increased.

• Chapter 6 of this dissertation assumes the availability of enough single pixel wide ground truth labels for creating label expansion on the training set. It also assumes that the spectral properties of the Roads are different from the spectral properties of non-roads. The approach fails when roads are covered by trees or urban constructions.

Bibliography

[1] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[2] Solving spacenet road detection challenge with deep learning: Jonathan howe, may casterline and abel brown. last modified february 20, 2018. (https://devblogs.nvidia.com/solving-spacenet-road-detection-challenge-deep-learning/).

[3] 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.

[4] Farideh Foroozandeh Shahraki and Saurabh Prasad. Graph convolutional neural networks for hyperspectral data classification. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 968–972. IEEE, 2018.

[5] Gustavo Camps-Valls and Lorenzo Bruzzone. Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6):1351–1362, 2005.

[6] 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.

[7] Gustavo Camps-Valls, Devis Tuia, Lorenzo Bruzzone, and Jón Atli Benediktsson. Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. *IEEE Signal Processing Magazine*, 31(1):45–54, 2014.

[8] 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.

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

[10] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[11] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[12] Li Deng and Dong Yu. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[13] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[14] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018.

[15] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[16] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[17] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.

[18] Hua Yu and Jie Yang. A direct lda algorithm for high-dimensional datawith application to face recognition. *Pattern recognition*, 34(10):2067–2070, 2001.

[19] Anil K Jain and Farshid Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern recognition*, 24(12):1167–1186, 1991.

[20] Steve De Backer, Antoine Naud, and Paul Scheunders. Non-linear dimensionality reduction techniques for unsupervised feature extraction. *Pattern Recognition Letters*, 19(8):711–720, 1998.

[21] Sameh M Yamany, Aly A Farag, and Shin-Yi Hsu. A fuzzy hyperspectral classifier for automatic target recognition (ATR) systems. *Pattern Recognition Letters*, 20(11-13):1431–1438, 1999.

[22] Shixin Yu, Steve De Backer, and Paul Scheunders. Genetic feature selection combined with composite fuzzy nearest neighbor classifiers for hyperspectral satellite imagery. *Pattern Recognition Letters*, 23(1-3):183–190, 2002.

[23] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 905–912. ACM, 2006.

[24] Minshan Cui and Saurabh Prasad. Angular discriminant analysis for hyperspectral image classification. *IEEE Journal of Selected Topics in Signal Processing*, 9(6):1003–1015, 2015.

[25] Ming Li and Baozong Yuan. 2d-lda: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532, 2005.

[26] Qian Du and Chein-I Chang. A linear constrained distance-based discriminant analysis for hyperspectral image classification. *Pattern Recognition*, 34(2):361–373, 2001.

[27] Dewen Hu, Guiyu Feng, and Zongtan Zhou. Two-dimensional locality preserving projections (2DLPP) with its application to palmprint recognition. *Pattern recognition*, 40(1):339–342, 2007.

[28] Souvick Mukherjee, Minshan Cui, and Saurabh Prasad. Spatially constrained semisupervised local angular discriminant analysis for hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2017.

[29] Saurabh Prasad and Lori Mann Bruce. Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geoscience and Remote Sensing Letters*, 5(4):625–629, 2008.

[30] Yuliya Tarabalka, Jón Atli Benediktsson, and Jocelyn Chanussot. Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques. *IEEE Transactions on Geoscience and Remote Sensing*, 47(8):2973–2987, 2009.

[31] 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.

[32] Antonio Plaza, Pablo Martinez, Rosa Pérez, and Javier Plaza. Spatial/spectral endmember extraction by multidimensional morphological operations. *IEEE Transactions on Geoscience and Remote Sensing*, 40(9):2025–2041, 2002.

[33] Minshan Cui, Saurabh Prasad, Wei Li, and Lori M Bruce. Locality preserving genetic algorithms for spatial-spectral hyperspectral image classification. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*, 6(3):1688–1697, 2013.

[34] Yuliya Tarabalka, Jón Atli Benediktsson, and Jocelyn Chanussot. Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques. *IEEE Transactions on Geoscience and Remote Sensing*, 47(8):2973–2987, 2009.

[35] Mathieu Fauvel, Yuliya Tarabalka, Jon Atli Benediktsson, Jocelyn Chanussot, and James C Tilton. Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3):652–675, 2013.

[36] K Muneeswaran, L Ganesan, S Arumugam, and K Ruba Soundar. Texture image segmentation using combined features from spatial and spectral distribution. *Pattern Recognition Letters*, 27(7):755–764, 2006.

[37] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[38] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, pages 507–516, 2016.

[39] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2017.

[40] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[41] Minshan Cui and Saurabh Prasad. Spatial context based angular information preserving projection for hyperspectral image classification. *arXiv preprint arXiv:1607.04593*, 2016.

[42] Minshan Cui and Saurabh Prasad. Angular discriminant analysis for hyperspectral image classification. *IEEE Journal of Selected Topics in Signal Processing*, 9(6):1003–1015, 2015.

[43] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2097–2104. IEEE, 2011.

[44] Tanu Priya, Saurabh Prasad, and Hao Wu. Superpixels for spatially reinforced bayesian classification of hyperspectral images. *IEEE Geosci. Remote Sensing Lett.*, 12(5):1071–1075, 2015.

[45] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[46] Greg Mori, Xiaofeng Ren, Alexei A Efros, and Jitendra Malik. Recovering human body configurations: Combining segmentation and recognition. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[47] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, page 10. IEEE, 2003.

[48] Pushmeet Kohli and Philip HS Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.

[49] Derek Hoiem, Andrew N Stein, Alexei A Efros, and Martial Hebert. Recovering occlusion boundaries from a single image. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[50] Srikumar Ramalingam, Pushmeet Kohli, Karteek Alahari, and Philip HS Torr. Exact inference in multi-label crfs with higher order cliques. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[51] Joel A Tropp, Anna C Gilbert, and Martin J Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal processing*, 86(3):572–588, 2006.

[52] Paolo Gamba. A collection of data for urban area characterization. In *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, volume 1. IEEE, 2004.

[53] Saurabh Prasad and Lori Mann Bruce. Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geoscience and Remote Sensing Letters*, 5(4):625–629, 2008.

[54] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 905–912. ACM, 2006.

[55] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.

[56] Masashi Sugiyama, Tsuyoshi Idé, Shinichi Nakajima, and Jun Sese. Semi-supervised local fisher discriminant analysis for dimensionality reduction. *Machine learning*, 78(1-2):35, 2010.

[57] Leyuan Fang, Shutao Li, Xudong Kang, and Jón Atli Benediktsson. Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8):4186–4201, 2015.

[58] Deng Cai, Xiaofei He, and Jiawei Han. Semi-supervised discriminant analysis. In *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007.*, pages 1–7. IEEE, 2007.

[59] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[60] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[61] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591, 2002.

[62] Charles M Bachmann, Thomas L Ainsworth, and Robert A Fusina. Exploiting manifold geometry in hyperspectral imagery. *IEEE transactions on Geoscience and Remote Sensing*, 43(3):441–454, 2005.

[63] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

[64] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[65] Yi Chen, Nasser M Nasrabadi, and Trac D Tran. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10):3973–3985, 2011.

[66] Jun Li, José M Bioucas-Dias, and Antonio Plaza. Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 50(3):809–823, 2012.

[67] Mathieu Fauvel, Yuliya Tarabalka, Jon Atli Benediktsson, Jocelyn Chanussot, and James C Tilton. Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3):652–675, 2013.

[68] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

[69] Jerome H Friedman. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175, 1989.

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

[71] Frédéric Ratle, Gustavo Camps-Valls, and Jason Weston. Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(5):2271–2282, 2010.

[72] Li-Fen Chen, Hong-Yuan Mark Liao, Ming-Tat Ko, Ja-Chen Lin, and Gwo-Jong Yu. A new lda-based face recognition system which can solve the small sample size problem. *Pattern recognition*, 33(10):1713–1726, 2000.

[73] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *Advances in neural information processing systems*, pages 1569–1576, 2005.

[74] Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma. Solving the small sample size problem of lda. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 29–32. IEEE, 2002.

[75] Hua Yu and Jie Yang. A direct lda algorithm for high-dimensional datawith application to face recognition. *Pattern recognition*, 34(10):2067–2070, 2001.

[76] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

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

[78] Hao Wu and Saurabh Prasad. Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Transactions on Image Processing*, 27(3):1259–1270, 2018.

[79] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[80] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE, 2011.

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

[82] 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.

[83] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[84] Joseph J Atick and A Norman Redlich. Towards a theory of early visual processing. *Neural Computation*, 2(3):308–320, 1990.

[85] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012.

[86] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

[87] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[88] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

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

[90] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[91] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics, 2011.

[92] Marc'Aurelio Ranzato and Martin Szummer. Semi-supervised learning of compact document representations with deep networks. In *Proceedings of the 25th international conference on Machine learning*, pages 792–799. ACM, 2008.

[93] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

[94] 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.

[95] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[96] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

[97] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. 2002.

[98] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *AISTATS*, volume 2005, pages 57–64. Citeseer, 2005.

[99] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in neural information processing systems*, pages 601–608, 2003.

[100] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

[101] Andreas Vlachos, Anna Korhonen, and Zoubin Ghahramani. Unsupervised and constrained dirichlet process mixture models for verb clustering. In *Proceedings of the workshop on geometrical models of natural language semantics*, pages 74–82. Association for Computational Linguistics, 2009.

[102] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[103] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[104] 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.

[105] Luis Gómez-Chova, Gustavo Camps-Valls, Jordi Munoz-Mari, and Javier Calpe. Semisupervised image classification with laplacian support vector machines. *IEEE Geoscience and Remote Sensing Letters*, 5(3):336–340, 2008.

[106] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[107] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[108] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *European Conference on Computer Vision*, pages 100–117. Springer, 2016.

[109] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[110] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[111] Yinglu Liu, Yan-Ming Zhang, Xu-Yao Zhang, and Cheng-Lin Liu. Adaptive spatial pooling for image classification. *Pattern Recognition*, 55:58–67, 2016.

[112] Yi-Hsuan Tsai, Onur C Hamsici, and Ming-Hsuan Yang. Adaptive region pooling for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–739, 2015.

[113] Ruilong Chen, Md Asif Jalal, Lyudmila Mihaylova, and Roger K Moore. Learning capsules for vehicle logo recognition. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 565–572. IEEE, 2018.

[114] 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.

[115] Md Asif Jalal, Ruilong Chen, Roger K Moore, and Lyudmila Mihaylova. American sign language posture understanding with deep neural networks. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 573–579. IEEE, 2018.

[116] Vittorio E Brando and Arnold G Dekker. Satellite hyperspectral remote sensing for estimating estuarine and coastal water quality. *IEEE transactions on geoscience and remote sensing*, 41(6):1378–1387, 2003.

[117] Alexander FH Goetz. Three decades of hyperspectral remote sensing of the earth: A personal view. *Remote Sensing of Environment*, 113:S5–S16, 2009.

[118] 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, 2011.

[119] Farid Melgani and Lorenzo Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on geoscience and remote sensing*, 42(8):1778–1790, 2004.

[120] Michael Theodore Eismann. Hyperspectral remote sensing. SPIE Bellingham, 2012.

[121] José M Bioucas-Dias, Antonio Plaza, Gustavo Camps-Valls, Paul Scheunders, Nasser Nasrabadi, and Jocelyn Chanussot. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and remote sensing magazine*, 1(2):6–36, 2013.

[122] Qishuo Gao, Samsung Lim, and Xiuping Jia. Hyperspectral image classification using convolutional neural networks and multiple feature learning. *Remote Sensing*, 10(2):299, 2018.

[123] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005.

[124] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *null*, pages 2169–2178. IEEE, 2006.

[125] Mercedes E Paoletti, Juan Mario Haut, Ruben Fernandez-Beltran, Javier Plaza, Antonio Plaza, Jun Li, and Filiberto Pla. Capsule networks for hyperspectral

image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4):2145–2160, 2018.

[126] Edgar Xi, Selina Bing, and Yang Jin. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*, 2017.

[127] Parnian Afshar, Arash Mohammadi, and Konstantinos N Plataniotis. Brain tumor type classification via capsule networks. *arXiv preprint arXiv:1802.10200*, 2018.

[128] Kai Qiao, Chi Zhang, Linyuan Wang, Bin Yan, Jian Chen, Lei Zeng, and Li Tong. Accurate reconstruction of image stimuli from human fmri based on the decoding model with capsule network architecture. *arXiv preprint arXiv:1801.00602*, 2018.

[129] Yash Upadhyay and Paul Schrater. Generative adversarial network architectures for image synthesis using capsule networks. *arXiv preprint arXiv:1806.03796*, 2018.

[130] Ayush Jaiswal, Wael AbdAlmageed, and Premkumar Natarajan. Capsulegan: Generative adversarial capsule network. *arXiv preprint arXiv:1802.06167*, 2018.

[131] Raeid Saqur and Sal Vivona. Capsgan: Using dynamic routing for generative adversarial networks. *arXiv preprint arXiv:1806.03968*, 2018.

[132] Bowen Zhang, XU Xiaofei, Min Yang, and XiaoJun Chen. Cross-domain sentiment classification by capsule network with semantic rules. *IEEE Access*, 2018.

[133] Wei-Lin Hsiao and Kristen Grauman. Creating capsule wardrobes from fashion images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7161–7170, 2018.

[134] 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.

[135] SpaceNet on Amazon Web Services (AWS). The SpaceNet Catalog. Last modified April 30, 2018. `https://spacenetchallenge.github.io/datasets/datasetHomePage.html`.

[136] Chuang Gu and Ming-Chieh Lee. Semiautomatic segmentation and tracking of semantic video objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):572–584, 1998.

[137] Bilge Gunsel, Ahmet Mufit Ferman, and A Murat Tekalp. Temporal video segmentation using unsupervised clustering and semantic object tracking. *Journal of Electronic Imaging*, 7(3):592–605, 1998.

[138] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[139] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[140] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[141] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[142] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *European Conference on Computer Vision*, pages 430–443. Springer, 2012.

[143] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.

[144] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[145] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016.

[146] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

[147] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016.

[148] Miguel Angel Sotelo, Francisco Javier Rodriguez, Luis Magdalena, Luis Miguel Bergasa, and Luciano Boquete. A color vision-based lane tracking system for autonomous driving on unmarked roads. *Autonomous Robots*, 16(1):95–116, 2004.

[149] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020. IEEE, 2018.

[150] Tsai Hong Hong, Christopher Rasmussen, Tommy Chang, and Michael Shneier. Road detection and tracking for autonomous mobile robots. In *Unmanned Ground Vehicle Technology IV*, volume 4715, pages 311–320. International Society for Optics and Photonics, 2002.

[151] Christopher Rasmussen. Combining laser range, color, and texture cues for autonomous road following. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 4, pages 4320–4325. IEEE, 2002.

[152] Massimo Bertozzi, Alberto Broggi, Massimo Cellario, Alessandra Fascioli, Paolo Lombardi, and Marco Porta. Artificial vision in road vehicles. *Proceedings of the IEEE*, 90(7):1258–1271, 2002.

[153] Yen-Lin Chen, Yuan-Hsin Chen, Chao-Jung Chen, and Bing-Fei Wu. Nighttime vehicle detection for driver assistance and autonomous vehicles. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 687–690. IEEE, 2006.

[154] Chung-Lin Huang and Wen-Chieh Liao. A vision-based vehicle identification system. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 364–367. IEEE, 2004.

[155] Sang Kyoon Kim, Dae Wook Kim, and Hang Joon Kim. A recognition of vehicle license plate using a genetic algorithm based segmentation. In *Proceedings of 3rd*

IEEE International Conference on Image Processing, volume 2, pages 661–664. IEEE, 1996.

[156] Chengcui Zhang, S-C Chen, M-L Shyu, and Srinivas Peeta. Adaptive background learning for vehicle detection and spatio-temporal tracking. In *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, volume 2, pages 797–801. IEEE, 2003.

[157] Hyo Jong Lee. Neural network approach to identify model of vehicles. In *International Symposium on Neural Networks*, pages 66–72. Springer, 2006.

[158] Venugopal KR and LM Patnaik. Moving vehicle identification using background registration technique for traffic surveillance. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, 2008.

[159] Dan Levi, Noa Garnett, Ethan Fetaya, and Israel Herzlyia. Stixelnet: A deep convolutional network for obstacle detection and road segmentation. In *BMVC*, pages 109–1, 2015.

[160] Paolo Lombardi, Michele Zanin, and Stefano Messelodi. Unified stereovision for ground, road, and obstacle detection. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 783–788. IEEE, 2005.

[161] Roberto Manduchi, Andres Castano, Ashit Talukder, and Larry Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous robots*, 18(1):81–102, 2005.

[162] Sebastian Ramos, Stefan Gehrig, Peter Pinggera, Uwe Franke, and Carsten Rother. Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1025–1032. IEEE, 2017.

[163] Serge Beucher and Michel Bilodeau. Road segmentation and obstacle detection by a fast watershed transformation. In *Proceedings of the Intelligent Vehicles' 94 Symposium*, pages 296–301. IEEE, 1994.

[164] Yue Wang, Eam Khwang Teoh, and Dinggang Shen. Lane detection and tracking using b-snake. *Image and Vision computing*, 22(4):269–280, 2004.

[165] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent progress in road and lane detection: a survey. *Machine vision and applications*, 25(3):727–745, 2014.

[166] Kuo-Yu Chiu and Sheng-Fuu Lin. Lane detection using color-based segmentation. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 706–711. IEEE, 2005.

[167] Dong-Joong Kang and Mun-Ho Jung. Road lane segmentation using dynamic programming for active safety vehicles. *Pattern Recognition Letters*, 24(16):3177–3185, 2003.

[168] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[169] Elizaveta Levina and Peter Bickel. The earth mover's distance is the mallows distance: Some insights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 251–256. IEEE, 2001.

[170] CL Mallows. A note on asymptotic joint normality. *The Annals of Mathematical Statistics*, 43(2):508–515, 1972.

[171] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. The Elements of Statistical Learning. 1, 2001.

[172] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[173] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[174] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[175] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[176] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[177] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[178] Julio F Navarro, Carlos S Frenk, and Simon DM White. A universal density profile from hierarchical clustering. *The Astrophysical Journal*, 490(2):493, 1997.

[179] Robert C Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461, 2010.

[180] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[181] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

[182] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review. *Information Systems*, 53:16–38, 2015.

[183] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[184] Alexander Hinneburg and Daniel A Keim. A general approach to clustering in large databases with noise. *Knowledge and Information Systems*, 5(4):387–415, 2003.

[185] Heli Sun, Jianbin Huang, Jiawei Han, Hongbo Deng, Peixiang Zhao, and Boqin Feng. gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration. In *2010 IEEE International Conference on Data Mining*, pages 481–490. IEEE, 2010.

[186] Tao Pei, Ajay Jasra, David J Hand, A-Xing Zhu, and Chenghu Zhou. Decode: a new method for discovering clusters of different densities in spatial data. *Data Mining and Knowledge Discovery*, 18(3):337, 2009.

[187] Werner Stuetzle and Rebecca Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 19(2):397–418, 2010.

[188] Jörg Sander, Xuejie Qin, Zhiyong Lu, Nan Niu, and Alex Kovarsky. Automatic extraction of clusters from hierarchical clustering representations. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 75–87. Springer, 2003.

[189] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie royale des sciences de Paris*, 1781.

[190] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[191] Luigi Ambrosio and Nicola Gigli. A users guide to optimal transport. In *Modelling and optimisation of flows on networks*, pages 1–155. Springer, 2013.

[192] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[193] Alfred Galichon. *Optimal transport methods in economics*. Princeton University Press, 2016.

[194] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55:58–63, 2015.

[195] Luigi Ambrosio. Lecture notes on optimal transport problems. In *Mathematical aspects of evolving interfaces*, pages 1–52. Springer, 2003.

[196] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[197] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[198] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[199] Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K Kalra, Yi Zhang, Ling Sun, and Ge Wang. Low-dose

ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE transactions on medical imaging*, 37(6):1348–1357, 2018.

[200] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[201] Mark J Shensa. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992.

[202] Pierre Dutilleux. An implementation of the algorithme à trous to compute the wavelet transform. In *Wavelets*, pages 298–304. Springer, 1990.

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

[204] M González-Audícana, X Otazu, O Fors, and A Seco. Comparison between mallat's and the à trous discrete wavelet transform based algorithms for the fusion of multispectral and panchromatic images. *International Journal of Remote Sensing*, 26(3):595–614, 2005.

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

[206] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

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

[208] Souvick Mukherjee and Saurabh Prasad. A spatial-spectral semisupervised deep learning framework using siamese networks and angular loss. *Under Review in the Journal of Elsevier's Computer Vision and Image Understanding*, 2019.

[209] Souvick Mukherjee and Saurabh Prasad. Deep feature extraction by semisupervised capsule neural networks for hyperspectral image classi

cation. *In preparation to be submitted to IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.