DEEPLY LEARNED ATTRIBUTE PROFILES FOR HYPERSPECTRAL PIXEL CLASSIFICATION

by

Murat Can Özdemir

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of
Master of Science

Sabanci University

August 2016

DEEPLY LEARNED ATTRIBUTE PROFILES FOR HYPERSPECTRAL PIXEL CLASSIFICATION

APPROVED BY

Assoc. Prof. Dr. Erchan Aptoula .....
(Thesis Supervisor)

Prof. Dr. Berrin Yanıkoğlu .....
(Thesis Supervisor)

Assoc. Prof. Dr. Koray Kayabol .....

Assoc. Prof. Dr. Selim Balcısoy .....

Asst. Prof. Dr. Kamer Kaya .....

DATE OF APPROVAL: 09/08/2016

...to humanity and beyond the observable universe

Acknowledgments

I would like to thank Mostafa Mehdipour Ghazi for being a good role model and sharing his experience in deep learning with me, setting me on proper footing with experimentation.

I want to express my gratitude to my supervisor Erchan Aptoula for his guidance, motivation, suggestions, superior support and encouragement on my graduate study. It was an unforgettable experience to work with him in this work and in the previous project on the plant identification task.

I would like to thank my supervisor Berrin Yanıkoğlu for her guidance and precious suggestions on my thesis study and on previous collaborations, through which I have mastered essential skills for survival in academia thanks to her level of standards.

I owe special thanks to many friends from numerous bands and choirs for distractions and fun, to my family, and especially to Özlem Muslu for their unconditional love and support at my best and at my worst.

I owe the most special thanks to my professors, especially to Mehmet Keskinöz and Meriç Özcan, who introduced me to bitter pill through numerous interactions and forged the stronger man that I am now.

DEEPLY LEARNED ATTRIBUTE PROFILES FOR HYPERSPECTRAL PIXEL CLASSIFICATION

Murat Can Özdemir

CS, M.Sc. Thesis, 2016

Thesis Supervisors: Erchan Aptoula, Berrin Yanıkoğlu

Keywords: Mathematical Morphology, Convolutional Neural Networks, Deep Learning, Remote Sensing, Extended Attribute Profiles, Hyperspectral Image Classification

Abstract

Hyperspectral Imaging has a large potential for knowledge representation about the real world. Providing a pixel classification algorithm to generate maps with labels has become important in numerous fields since its inception, found use from military surveillance and natural resource observation to crop turnout estimation. In this thesis, within the branch of mathematical morphology, Attribute Profiles (AP) and their extension into the Hyperspectral domain have been used to extract descriptive vectors from each pixel on two hyperspectral datasets. These newly generated feature vectors are then supplied to Convolutional Neural Networks (CNNs), from off-the-shelf AlexNet and GoogLeNet to our proposed networks that would take into account local connectivity of regions, to extract further, higher level abstract features. Bearing in mind that the last layers of CNNs are supplied with softmax classifiers, and using Random Forest (RF) classifiers as a control group for both raw and deeply learned features, experiments are made. The results showed that not only there are significant improvements in numerical results on the Pavia University dataset, but also the classification maps become more robust and more intuitive as different, insightful and compatible attribute profiles are used along with spectral signatures with a CNN that is designed for this purpose.

HİPERSPEKTRAL PİKSEL SINIFLANDIRMA İÇİN DERİN ÖĞRENİLMİŞ ÖZNİTELİK PROFİLLERİ

Murat Can Özdemir

BM, Yüksek Lisans Tezi, 2016

Tez danışmanları: Erchan Aptoula, Berrin Yanıkoğlu

Anahtar Kelimeler: Uzaktan Algılama, Derin Öğrenme, Evrişimsel Sinir Ağları, Matematiksel Biçimbilim, Hiperspektral Görüntü Sınıflandırma, Öznitelik Profilleri

Özet

Hiperspektral Görüntüleme, Uzaktan Algılama araştırmalarında önemli bir yer tutmaktadır. Sınıflandırma haritası oluşturmanın faydaları askeri uygulamalarda, doğal afetlerde ve hatta tarımda uzmanların görsel bilgisine katkı sağlayarak uygulama alanı bulmasını sağlamıştır. Bu tez çalışmasında, sınıflandırma haritası oluşturmak amacıyla, hiperspektral veri kümelerinden, Matematiksel Biçimbilim dalına ait bir yaklaşım olan Öznitelik Profilleri uygulanarak alan ve moment betimleyicileriyle her piksel için öznitelik vektörleri hesaplanmıştır. Veri girdileri, piksele ait spektrum verisi, farklı betimleyicilerden oluşturulan Öznitelik Profilleri ve bunların birleşimini de kapsayacak şekilde hazırlanmıştır. Bu veri girdileri, AlexNet ve GoogLeNet gibi bilinen ağlar ve kendi önerdiğimiz, hiperspektral veri kümelerinde nesnelerin komşuluk bilgisini de göz önüne alan ağlar da dahil olmak üzere beş farklı Evrişimsel Sinir Ağları'nda denenmiş ve derin öznitelikleri çıkarılmıştır. Rasgele Orman sınıflandırıcılarıyla kontrollü olarak yapılan deneylerin sonuçlarında sayısal açıdan Pavia Üniveristesi veri kümesinde büyük ilerlemeler görülmüş ve oluşturulan sınıflandırma haritalarının daha anlaşılır olması sağlanmıştır. Böylece, alan ve moment betimleyicilerden elde edilen Öznitelik Profilleri ve spektral bilginin Evrişimsel Sinir Ağları ile kullanımının önemi gösterilmiştir.

Table of Contents

Acknowledgments v

Abstract vi

Özet vii

1 Introduction 1
1.1 Scope and Motivation . . . 1
1.2 Contributions . . . 3
1.3 Outline . . . 3

2 Background 5
2.1 Introduction: Remote Sensing . . . 5
2.2 Hyperspectral Imaging . . . 6
2.3 Morphological and Attribute Profiles . . . 9
2.3.1 Extension into Hyperspectral Domain . . . 11
2.4 Deep Learning . . . 13
2.4.1 Convolutional Neural Networks . . . 15
2.4.2 Caffe . . . 18
2.5 Literature Review . . . 18

3 Combining Mathematical Morphology with Deep Learning 22
3.1 Rationale . . . 22
3.1.1 Neural Network Selection . . . 23
3.1.2 Ideas for Data Preparation . . . 33
3.1.3 Parameter and Hyperparameter Optimizations . . . 34
3.1.4 Efficiency . . . 35
3.2 Datasets . . . 36
3.2.1 Pavia University Scene . . . 36
3.2.2 Pavia Center Scene . . . 37

4 Results **38**
4.1 Methods . . . 38
4.1.1 Spectral Signatures . . . 38
4.1.2 Extended Attribute Profiles . . . 39
4.1.3 Combination . . . 39
4.1.4 Multidimensional data approach . . . 39
4.1.5 Results . . . 40
4.2 Discussion . . . 44

5 Conclusions and Future Work **45**
5.1 Conclusions . . . 45
5.2 Future work . . . 46

Bibliography **47**

List of Figures

2.1 Hyperspectral data 7
2.2 EAP with area attributes, thickening in successive stages 12
2.3 EAP with area attributes, thinning in successive stages 12
2.4 An application of AlexNet architecture on ImageNet dataset [1] . . . 14
2.5 Connectivity difference between fully connected layers (bottom) and convolutional layers (top). This difference in architecture enables the network to learn from a specific neighbourhood, instead of having input from every neuron in the previous layer. This results in computational, spatial and functional efficiency [2]. 16
2.6 Max pooling layer only cares about its immediate neighborhood, therefore if the layer starts to operate from a neuron to the left, some results might change, but most stay intact [2]. 17

3.1 Rationale 23
3.2 Test 2 approach: $9 \times 9 \times 4$ patches converted to 1×324 [3] 23
3.3 Test 3 approach: Area attribute is used for EAP, resulting in 1×116 24
3.4 Test 4 approach: Area and moment attributes used for EMAP, resulting in 1×148 vectors for each pixel. 25
3.5 Test 5 approach: Addition of spectral profiles to that of Test 4. . . . 26
3.6 AlexNet architecture 27
3.7 GoogLeNet architecture 29
3.8 modAlexNet as a whole 31
3.9 ConfNet as a whole 32
3.10 Feature extraction layer of modAlexNet 33
3.11 An overview of the ideas 34
3.1 Pavia Center dataset 37

3.2 Pavia University dataset 37

4.1 Pavia Center classification maps 41
4.2 Pavia Center classification maps-RF-vector input 41
4.3 Pavia University classification maps-AlexNet-vector input 41
4.4 Pavia University classification maps-GoogLeNet-vector input 42
4.5 Pavia University classification maps-modAlexNet-vector input 42
4.6 Pavia University classification maps-confNet-vector input 42
4.7 Pavia University classification maps-RF-vector input 43
4.8 Pavia University classification maps-multidimensional approach . . . 43

List of Tables

3.1 A comparison of GPUs: 1) Nvidia Quadro K4000 2) GeForce GTX 980M 36

4.1 Pavia Center, best results with kappa statistic, SM = softmax 40

4.2 Pavia University, best results with kappa statistics, SM = softmax . 40

4.3 Pavia University, multidimensional approach 43

Chapter 1

Introduction

Since humanity has taken to the skies, there has been an interest in bird's eye view imaging with different apparatus. Early balloonists made the first attempt as early as 1858. Later on, messenger pigeons, kites, rockets and unmanned balloons were also used to take images. After the start of WWI and WWII, followed by the Cold War, this discipline had been established with serious grounding, due to the applications aimed at military surveillance and reconnaissance that proved immense worth. Modified military airplanes and later on artificial satellites and unmanned aerial vehicles (UAV) were used to collect information remotely using infrared, Doppler, conventional photography and synthetic aperture radar. The development of more complicated signal processing algorithms and sensors that are capable of extracting more precise spectral signatures finally paved the way for the current standards of hyperspectral imaging technology [4].

1.1 Scope and Motivation

In various disciplines, expert decision systems are installed to aid in decision making and automatization. Some of those systems would utilize remote sensing for generation of a classification map, a bird's eye view of the area of interest that is labeled with a finite number of classes. Therefore, this classification task in contemporary hyperspectral imaging is of high significance. A non-exhaustive list of the main challenges of this area is as follows [5], [6]:

- Different sensors: Sensors that have different specifications from one another will inevitably produce different arrays of reflectance values.

- Different lighting: Due to lighting changes during the day, the spectral signature of an item of a particular class will change.

- Different meteorological instances: Atmospheric conditions and presence of a cloud or a different combination of air molecules will produce different spectral signatures for the same object even when all other conditions are fixed. In some occasions this may result in removal of some bands altogether since the image at that band would have become completely useless due to absorption or total reflectance of a specific wavelength [7].

- Different resolutions: This is linked to the general problem of image resolutions. Different settings of image retrieval can distort the resolution and pinpointing of "pure pixels" might prove difficult, which is a desired trait since it will help with training pixel selection in the classification task. Even if the image retrieval part is done perfectly, due to relatively low resolution of these images, there will still be pixels that are "mixed", containing spectral signatures of more than one class. High resolution is problematic as well: At high spectral resolution, due to relatively low number of labelled samples for training and classification, Hughes phenomenon is inevitable, while at high spatial resolution, too many details on the map increases the burden on computations. [8].

- Different locations: Same objects in different locations would have the spectral signature of their background material, which will inevitably be mixed into the response and make its way to the reflectance values that are collected from imaging equipment.

This thesis will solely focus on the pixel classification problem, which is burdened by the problems of this field. In this problem, ground truth pixels are labeled to certain classes of objects and optionally, a training set is also provided. The aim will be to classify the remainder of the instances optimally to generate a classification map for further uses. There are other studies in which these efforts would lead to generalizations about a particular sensor or a class [9], but this study will consist of obtaining two preprocessed hyperspectral datasets and from that point, treating them like machine learning problems while keeping in mind their optical properties.

1.2 Contributions

This thesis will present a comparative study of attribute profiles with area and moment attributes as content descriptors that are used for training, and 5 different Convolutional Neural Networks to extract higher level of features from them, along with other commonly known approaches for a comparison. Since Extended Attribute Profiles (EAPs) are capable of extracting spatial information from hyperspectral images and although CNNs are powerful, they lack their most interesting property of extracting spatial filters when the input images are not grayscale or three-channel color images. The two methods should complement each other. Hence, exploiting spatial information from hyperspectral images while being able to use CNNs for higher levels of abstract features is made possible. The experiments will be done on two different datasets that are acquired from the same sensor over the same city to mitigate with the problems of this field, which would enable a better conclusion of the proposed techniques.

1.3 Outline

The rest of the thesis is organized as follows:

Chapter 2 introduces Remote Sensing, Hyperspectral Imaging and a class of Mathematical Morphological tools called Morphological Profiles (MP) and Attribute Profiles (AP), followed by their extension into the hyperspectral domain. A brief history of neural networks is also given, followed by the construction of convolutional neural networks and its benefits. The tool of choice, Caffe will also be presented. The rest of the chapter will then contain a narrow field literature survey to explore the strategies that are available to solve this problem.

Chapter 3 covers the proposed modus operandi for this problem. The rationale will be explained, followed by definitions of different network architectures and input data preparation stages. At last, working stations will be compared and conclusions will be drawn from those.

In Chapter 4, different experiments that are devised on two different datasets will be explained, their results on evaluation metrics will be presented along with classification maps and conclusions will be drawn.

Chapter 5 provides a summary of the contributions and the results of this thesis, and suggests several potential future research directions.

Chapter 2

Background

This chapter provides the basic concepts of Remote Sensing, Hyperspectral Imaging, Morphological and Attribute Profiles and their extension into the Hyperspectral domain, and Deep Learning Methods. It also includes a survey of published work on the usage of deep learning methods on pixel classification.

2.1 Introduction: Remote Sensing

Remote Sensing is the main area of research that deals with data acquisition through capturing and quantizing force fields or radiation that are reflected from sceneries, and interpreting this aerial viewpoint data in identifying objects, biodiversity, composition of complex bodies and classes of land and water surfaces over the Earth or other heavenly bodies.

Remote Sensing measures electromagnetic energy emanating from distant objects made up of various materials. This often provides rich information about those objects at the tasks of identification, classification and detection. Spatial information can also be incorporated for the abovementioned tasks, which has proved to be useful in the greater field of image processing for a long time [10], [11], [12], [13].

In passive imaging, reflectance data are collected from a range of wavelengths in the electromagnetic spectrum. These can be called multispectral imaging if there are at most ten channels with relatively large differences in wavelength, whereas hyperspectral imaging occurs when hundreds and more of these channels are recorded within a (usually narrowly differing) bandwidth [14]. However, this thesis will contain work on hyperspectral images only.

2.2 Hyperspectral Imaging

Hyperspectral imaging sensors operate on wavelengths from the visible through the middle infrared ranges and have the technology to capture hundreds of spectral channels simultaneously. The collected data from these narrowly separated channels over a bird's eye view over the Earth are stored in pixels. Each pixel in this imaging technique is a vector composed of measurements on specific wavelengths. Hence, the size of each vector is equal to the number of data points, i.e., measurements from the EM spectrum. Since hyperspectral images represent each pixel on hundreds of spectral responses, the resultant spectral information is a reliable spectral signature. This can be used to increase the possibility of accurately discriminating materials of interest with an increased classification accuracy. Recently, this field of imaging is receiving advances with finer spatial resolution, providing even better information than ever [15]. With these information in mind, hyperspectral imaging has a potential for numerous sciences and expertise areas, such as:

- Ecology: Estimation of biomass, carbon and biodiversity are crucial for the monitoring of natural resources. Studying land cover changes can be particularly difficult when densely forested or otherwise prohibitive areas are concerned. Hyperspectral imaging provides rich information to remedy this through remote sensing [16].

- Geology: Measurements can be made over large areas to determine the general composition and abundance of certain minerals, which empowers domain experts in land type classification tasks and provides further insight. [17]

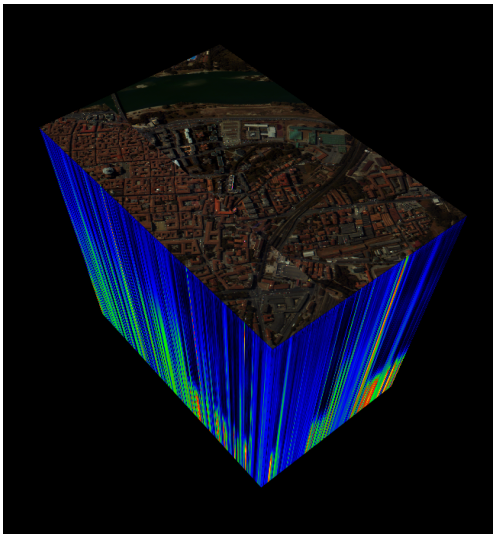
- Mineralogy: Identification and correspondence of different minerals can be understood through the rich information that hyperspectral imaging provides, which comes in handy when looking for a new mineral deposit. A curious investigation is studying the effect of oil and gas leakages on the changes of the spectral signature of nearby vegetation. [18]

- Hydrology: Current state of wetlands can be discovered by the information that hyperspectral imaging provides. Water quality, estuarine environments and coastal zones can be monitored for expert opinion as well. [19]

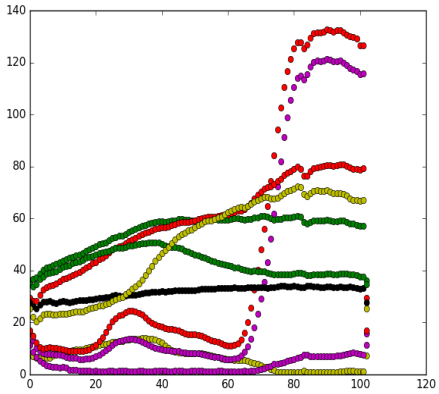
- Agriculture: Hyperspectral data is immensely powerful in the classification task of agricultural classes. Following that, tracking plant health parameters for the purpose of agricultural development is also a favorite area. [20]

- Military applications: While the most popular application of hyperspectral imaging for military applications is target detection, it is useful to obtain a summary of the terrain to most experts, although care must be given for algorithm design, since most convenient ones that are made for the multispectral images are not straight up adaptable to the analysis of hyperspectral images. [21]

Hyperspectral images can be viewed as a stack of images that represents the responses of different wavelengths (spectral channels) from the same scene. Therefore, this stack of images constitute a hyperspectral data tensor. Typical hyperspectral data consists of $n_1 \times n_2 \times d$ pixels where $n_1 \times n_2$ is the number of pixels in each spectral channel as width and height, with d number of different spectral responses. Analyzing hyperspectral data therefore will inevitably have two different perspectives [22]:



(a) Pavia Center Dataset



(b) Spectral response of average reflectance values of the labels of Pavia Center

Figure 2.1: Hyperspectral data

1. Spectral perspective: In this case, each pixel is a vector, containing d values. Each pixel is represented by its spectral signature, which is produced when

total radiance from the object is received and distributed into their respective narrow bands. This detailed spectral signature can be used to accomplish great many deals:

In general, similar materials, even when separated spatially, produce similar spectral signatures. This provides a raw feature vector, readily available for that specific labelled instance, which can be used to group or classify all pixels in that image. This had been the earliest approach to handle hyperspectral data [23].

2. Spatial perspective (or spatial dimension): From this perspective, a hyperspectral data cube consists of d grayscale images with a size of $n_1 \times n_2$.

   In the spatial dimension, in particular for Very High Resolution (VHR) data, the spatial resolution helps to identify different objects of interest on the surface of Earth with greater precision. Bearing in mind that the neighborhood pixels have a strong correlation, due to the fact that they represent the spectral signature of neighboring elements that may be related to each other. Pixels that represent often the same class of object or objects that should belong together in that scene are up for the taking when the neighborhood information is taken into account. However, it is not desired to do this spatial analysis on hundreds of band images, therefore this step is usually coupled with a dimensionality reduction step and a hierarchical representation on the remaining bands.

Multispectral images, which usually has approximately ten channels, has a few useful tools that were used on first hyperspectral images. However, as it turned out, most of the commonly used methods designed for the analysis of grayscale, color or multispectral images are inappropriate and even useless for hyperspectral images [24]. The Hughes Phenomenon/Curse of Dimensionality poses another problem for designing robust statistical estimations. In conclusion, this area of research needs spatial analysis techniques that are designed for this problem and classification problem, due to large feature vectors and not enough training samples, needs to be tackled by employing different machine learning techniques. In this thesis, Attribute Profiles as a spatial analysis technique is presented with their extension

into hyperspectral domain to extract spectral-spatial features, which will be used to train CNNs to obtain higher level meta-features.

2.3 Morphological and Attribute Profiles

Spatial information is fundamentally important in the analysis of remote sensing images of very high spatial resolution (VHR). This high resolution reveals the geometrical features of the structures in a scene with such a great perceptual significance that becomes useful in defining spectral signatures for a specific class, which helps with classification stage by providing a good feature vector. Therefore this advantage aids in the discriminability between different thematic classes, improving the performance in classification tasks. In order to include spectral-spatial features in the image analysis, Pesaresi and Benediktsson [25] introduced the concept of morphological profiles (MPs), which is achieved by stacking filters of multi-scale opening and closing by reconstruction on an image.

The MP were efficient at modelling spectral-spatial information, one of the primary results in their usage can be found in the classification of high-resolution panchromatic IKONOS images [26]. From the MP, Dalla Mura et al.[27] proposed attribute profiles (APs), a definition that contains MPs.

Given a grayscale image $f : E \to \mathbb{Z}, E \subset \mathbb{Z}^2$, its upper level sets are defined as $\{f \geq t\}$ with $t \in \mathbb{Z}$ and the lower sets are defined as the complementary of it. Filtering each of the peak components of the lower and upper level sets according to a predefined logical predicate T_λ^α for an attribute α and a threshold λ is called Attribute Filtering (AF). If the predicate is defined so that the outcome of this filtering is extensive, it is called attribute thickening, $\phi^T(f)$, otherwise it is called attribute thinning, $\gamma^T(f)$.

Attribute profiles are defined through attribute thinning and thickening operations over binary or grayscale images. These thinning and thickening operations are defined to remove connected components (CCs) from an image based on certain criteria. At the binary case, it is defined as the complete removal of CCs, while with grayscale images, the same will happen with peak components. The criteria are defined by attributes and certain thresholds. Attributes can be purely geometric (e.g. area, length of the perimeter, image moments, shape factors), or statistical

(e.g. range, standard deviation, entropy), instead of just structuring elements that are used for morphological profiles. This flexibility improves the modelling of the spectral-spatial information in the image. Attribute thinning and thickening thus will be defined as follows:

$$\gamma^{T}(f)(x) = max\{k : x \in Th_{k}(f)\} \tag{2.1}$$

$$\phi^{T}(f)(x) = min\{k : x \in Th_{k}(f)\} \tag{2.2}$$

In the equations above, $Th_{k}(f) = \cup\{h_{p}(f), p \geq k\}$ is union of all of the results of the level sets at greyscale level k, with $k \in [0, max(f)]$, obtained on greyscale image f. The logical predicate T and CCs of the upper and lower level sets of f, which are represented by $h_{k}(f)$ are used with the threshold value k for a given attribute.

Attribute thinning and thickening thus can generate different outcomes for an image to become profiles, but if the increasingness property is satisfied, which is $f \leq g \rightarrow \gamma^{T}(f) \leq \gamma^{T}(g)$, i.e, if a greyscale image with larger values will generate a larger filtered image with same attribute and threshold than the latter, then these operations can be called attribute opening. Area attribute can be used for this. In fact, this is how topological maps are produced by cartographers who use the area attribute on altitude data. On the other hand, standard deviation attribute may not be available for attribute opening and closing but only for thinning and thickening. For all the points that are made until now, the opposite reasoning holds true between thickening and closing as well.

In spite of this, attribute opening and closing does not determine whether a series of increasing criteria $T^{'} = \{T_{1}, T_{2}, ..., T_{\lambda}\}$ could generate attribute profiles alone, though. If there are correct, increasing thresholds to ensure formal order within the profile, which is $i \leq j \rightarrow T_{i} \subseteq T_{j} \rightarrow \gamma^{T_{i}} \leq \gamma^{T_{j}}$, attribute profile can be constructed. Therefore, attribute profiles Π_{i} can be defined for a series of increasing criterion $T^{'} = \{T_{1}, T_{2}, ..., T_{\lambda}\}$ as follows:

$$AP(f) = \Pi_{i} : \begin{cases} \Pi_{i} = \Pi_{\phi^{T_{\lambda}^{'}}}, & \text{with } \lambda = (n-1+i), \forall\lambda \in [1, n]; \\ \Pi_{i} = \Pi_{\gamma^{T_{\lambda}^{'}}}, & \text{with } \lambda = (i-n-1), \forall\lambda \in [n+1, 2n+1]. \end{cases} \tag{2.3}$$

The reader is encouraged to observe that this notation constructs the attribute profile from a stack of thickening profiles stacked backwards, with the original

grayscale image in the middle, followed by the thinning profiles. Other extensions of attribute thinning and thickening to greyscale images are also possible, leading to different filtering effects (Salembier et al. [28], Urbach et al. [29]). However, within this thesis' scope, the focus will be on the definitions given above on the hyperspectral data, which are composed of greyscale images that represent specific wavelengths.

2.3.1 Extension into Hyperspectral Domain

Hyperspectral data received its fair share of image processing and statistics based approaches aiming at reducing the computational workload. There is a comprehensive review of different techniques that are used for hyperspectral image processing [30]. However, this thesis will focus on how mathematical morphology tools (e.g. APs) can be applied to hyperspectral data for pixel classification purposes.

The task of extending morphological and attribute profiling to hyperspectral domain is not straightforward, since any thresholding operation requires an ordering relation between the elements, which is not natively defined for pixel vectors in hyperspectral data. To mitigate this, either different vector comparison strategies must be explored or the length of each pixel vector must be dropped to reasonable levels so that more commonly known vector comparison metrics can be used. Benediktsson et al. [31] proposed a dimensionality reduction strategy called principal-component analysis (PCA), then computing MP on each of the principal components (PCs) to remedy this issue. Palmason et al. [32], on the other hand, proposed the independent-component analysis (ICA) for dimensionality reduction. Thus, the stacking of the MPs computed on the principal components gave way to extended morphological profile (EMP).

$$EMP = \{MP\left(PC_1\right), MP\left(PC_2\right), ..., MP\left(PC_c\right), \} \tag{2.4}$$

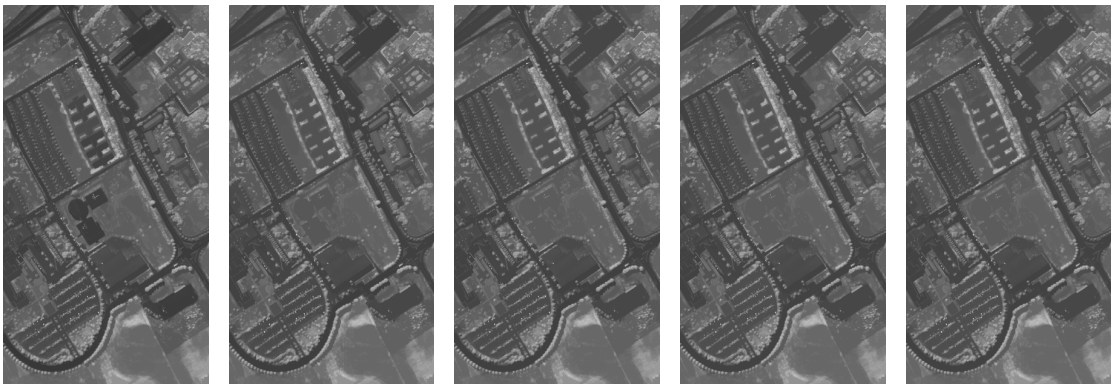
EMP features have already been tried with different classification methods, Benediktsson et al. used neural networks [31], Chan et al. used random forests (RF) [33] and Fauvel et al. used support vector machines (SVM) while adding in spectral information as well [34]. Plaza et al. proposed another approach for extending the concept of MPs to hyperspectral data [35], which is a reduced-vector

ordering scheme based on the spectral purity of pixel vectors, following their earlier work [36].

The extended attribute profile (EAP) and the extended multi-attribute profile (EMAP) [27] are simple extensions of APs to the principal components (PCs) at hand by stacking each AP of PCs, and each EAPs of a specific attribute, respectively.

$$EAP = \{AP\left(PC_1\right), AP\left(PC_2\right), ..., AP\left(PC_c\right)\} \tag{2.5}$$

$$EMAP = \{EAP_{a_1}, EAP_{a_2}, ...\}, \text{where } a_i \text{ are different attributes} \tag{2.6}$$



(a) 1st threshold: 770 (b) 4th threshold: 3076 (c) 6th threshold: 4615 (d) 11th threshold: 8461 (e) 14th threshold: 10769

Figure 2.2: EAP with area attributes, thickening in successive stages

(a) 1st threshold: 770 (b) 4th threshold: 3076 (c) 6th threshold: 4615 (d) 11th threshold: 8461 (e) 14th threshold: 10769

Figure 2.3: EAP with area attributes, thinning in successive stages

2.4 Deep Learning

In this section, a very brief deep learning literature will be presented, with the reasons to use them at the end. For the comprehensive work, one can refer to Schmidhuber's review [37].

A neural network (NN) consists of many neurons that are usually in rigid layers, each being connected to some other neurons, producing a sequence of real-valued activations based on the weighted activities of other neurons that are on their receptive fields. Some neurons may influence the environment by triggering responses. Learning or credit assignment [38] is the determination of weights for NN so that each input will trigger correct behavior, such as the correct classification of millions of images that have thousands of different labels. These behaviors can be so complicated that it might require long chain of layers and non-linear thresholds to be computed, which transforms the activation of all of the network. Deep Learning applications typically have many such stages that needs to be set with correct weights, which empower them to handle these challenges.

Neural networks are akin to linear regression models as both try to estimate parameters for a function that maps the input to the output, which is Gauss' idea of linear regression [39]. Inspired by this idea, various early systems had shown that they can make associations with previous input akin to a neural network starting from 1940s [40]. Advancements in neuroscience around 1960s showed that the visual cortex of animals showed a multi-layered architecture with varying connections [41] [42], which kick-started the beginnings of neural networks [43] [44] and finally, multilayer perceptrons [45] [46] [47]. Fukushima's neocognitons [48] were the pinnacle of that period, before starting "winter of AI" [49] [50].

Deep Learning became a workable idea in 1980s [51] [52] [53] and an active research area in 90s before its computational cost became apparent. There was a problem about vanishing gradients as well, which made training procedure prone to divergences from optimal solutions up to null hypothesis, when the network learns nothing at all. Complexity measurements and prevalence of other classification algorithms as well made research in this area falter. Backpropagation algorithms [54], supervised [55], reinforcement [56] or unsupervised learning [57] strategies were all thrown into the common knowledge pool in the academia to solve this issue, only

to be halted by the lack of appropriate hardware to cut the computational load and other practical problems at the training stage, such as the absence of huge libraries of well labelled instances.

Finally, in 2006, deep neural networks were shown to be capable of training without any problems [58], with an increased efficiency of parameter update calculations that enables them to be trained on purpose. A revival in autoencoders [59] and other unsupervised learning schemes brought this topic back under spotlight, but by far the most influential results stemmed from AlexNet that is submitted to ILSVRC's competition on ImageNet dataset in 2012 [60] and GoogLeNet in 2015 [61], where Alex used Convolutional Neural Networks with new approaches in training them to achieve a great level of accuracy and Google introduced Network in Networks and inception module approaches, which consists of convolutional layers, max pooling layers, normalization and dropout layers. Those networks can also be used to extract features, which are more flexible than a standard set of feature description algorithms such as NWFE, BDFE or SIFT-variants, due to variable window size and freedom of choice in filtering algorithms that can be implemented on the go [37].

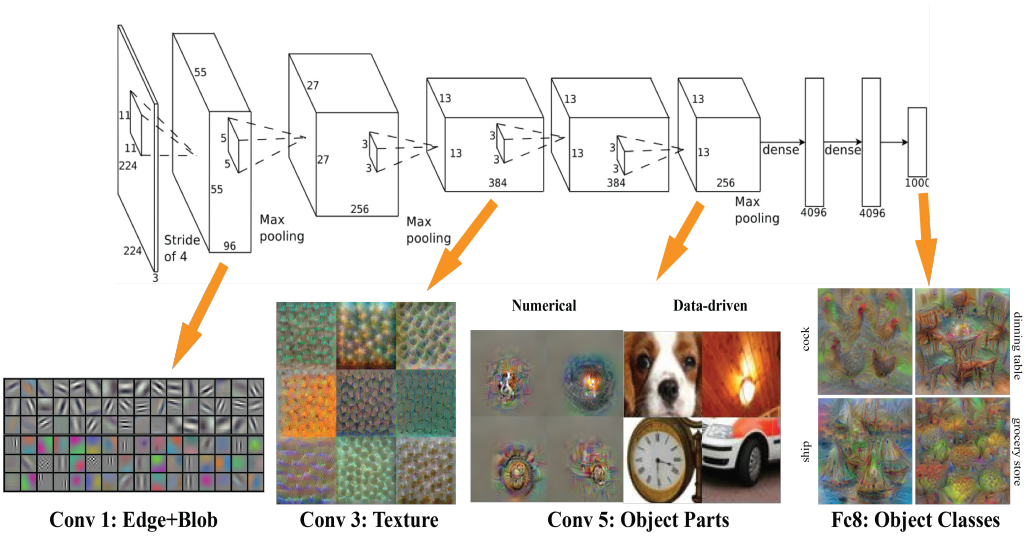


Figure 2.4: An application of AlexNet architecture on ImageNet dataset [1]

Following many advantages of Deep Learning, many more approaches have also been proposed, either by drawing inspiration from nature or statistical and visual models. Interest in CNNs, Recurrent Neural Networks (RNNs) with receptive fields

from neighboring neurons on the same layer to give more options to learn other than one-to-one input output relation, Long Short-term Memory Networks (LSTMs) with memory gates that models memory and neural plasticity, have revived. Recently, ladder networks, which exploit spatio-temporal relationship in the data, are proposed [62] with many supervised and reinforcement learning strategy methods. According to many experts [63] [64] [65], Deep Learning has not even reached the point of losing its effectiveness as "a silver bullet" in many areas of data science.

2.4.1 Convolutional Neural Networks

Convolutional networks are neural networks that have at least one convolutional layer in the totality of their network architecture. In the context of this thesis, it refers to LeNet [59], AlexNet [60], GoogLeNet [61] or their variants, which consists of essentially convolutional, pooling and dropout layers, to count a few. Those three layers are of utmost importance in this thesis due to their intrinsic properties:

- Convolution are aimed to bring three powerful advantages that can help to improve a machine learning system: sparse connectivity, parameter sharing and equivariant representations. Convolution enables working with different receptive fields. Traditional neural network layers use matrix multiplication between input and output layers, but each of the neurons are connected to all of the neurons of the next layer. This would result in large, dense matrix calculations at each layer, which undoubtedly increases computation time for parameter update calculations. Convolutional networks, however, usually have sparse connectivity (also referred to as sparse weights), due to the fact that a convolution operation usually deals with a receptive field that is in the immediate neighborhood. The reason for using this connectivity stems from the fact that while processing an image, the input image might have thousands or millions of pixels, but relevant features detected from the regions of interests such as edges and blobs occupy only tens or hundreds of pixels. Therefore it is only intuitive to store fewer parameters, which would make sparse weight matrices that both reduces the memory requirements of the model and improves its statistical efficiency. It also means that computing the output requires fewer operations. All three advantages usually add up to immense efficiency.

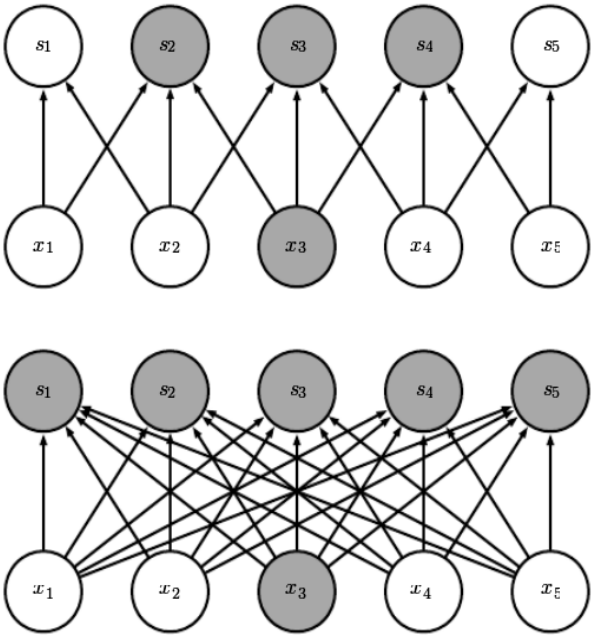


Figure 2.5: Connectivity difference between fully connected layers (bottom) and convolutional layers (top). This difference in architecture enables the network to learn from a specific neighbourhood, instead of having input from every neuron in the previous layer. This results in computational, spatial and functional efficiency [2].

- A pooling function replaces the activation of all neurons at all layers with a statistic that summarizes the neighboring neurons. For example, the max pooling [66] operation assigns the weight of each neuron the maximum value within a rectangular neighborhood. Other pooling functions looks to assign values that are the average of a rectangular neighborhood, the L_2 norm of a rectangular neighborhood, or the distance weighted average, where the distances are measured from a central pixel [67]. In all cases, pooling generates better representations that are invariant to small translations of the input, i.e. if the input alters a little, the values of the layers after pooling operation do not change much. This puts precedence on the presence of a feature rather than its exact location, which is a desired trait for hyperspectral pixel classification.

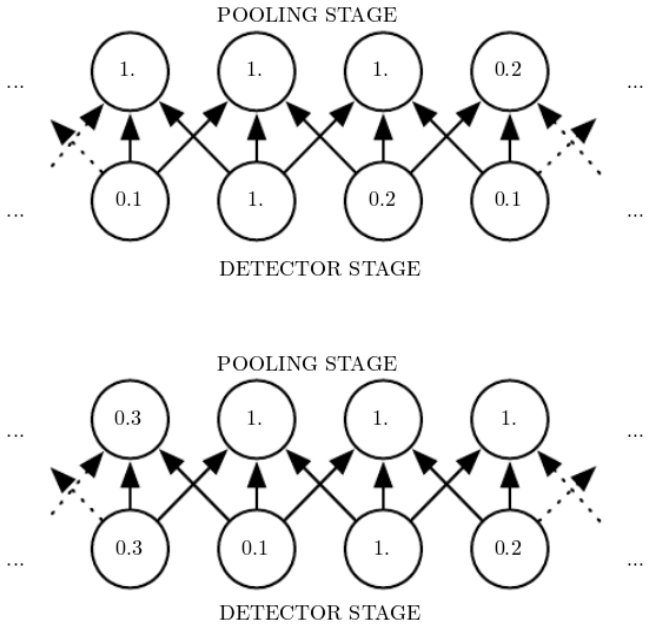


Figure 2.6: Max pooling layer only cares about its immediate neighborhood, therefore if the layer starts to operate from a neuron to the left, some results might change, but most stay intact [2].

- Dropout layers decrease dependence on all neurons in a fully connected network. This is achieved by randomly assigning a fraction of the elements of a descriptor vector to 0. If there is a large object to be discovered in an image, output from a convolutional layer, followed by a pooling layer generates a dense matrix around that object, some of which can be too similar or noisy, leading to overfitting of the learned features. Since in the field of remote sensing inefficient amount of training samples is usually the problem [68], a dropout layer can be considered to switch off a portion of the network randomly to reduce full dependency on all neurons from the previous layer while the full network continues to learn. This also constitutes a form of ensemble averaging over the neighboring regions, which leads to better generalizations.

Weight initializations can be done using a statistical model with carefully selected parameters or these weights can also be learned from another pre-trained model, with an aim to increase the classification performance of the network. Different solving strategies also offer different trade-offs between accuracy and computational time. These topics will be explored further with the tool of choice, Caffe.

2.4.2 Caffe

Caffe is a deep learning framework made with expression, speed, and modularity in mind [69]. It is developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors. It has the main advantage due to the fact that models and optimization are defined by configuration without hard-coding, which leads to faster network generation, which are auto-generated by Google protobuffer compilers from .prototxt files. Through this architecture, many types of layers can be defined, from convolutional layers to different normalization and regularization layers. It utilizes cuDNN libraries [70] for fast GPU implementation and Blas libraries [71] for fast CPU implementation. Many solver types are also supported as well, with all parameters that can be manipulated.

With this tool at hand, this section will be followed by a brief literature survey of the common approaches to solve hyperspectral pixel classification problem:

2.5 Literature Review

Deep learning techniques are prevalent due to the fact that an increase in the layer count can lead to more abstract and complex features, which leads to better results. Obtaining this inspiration from the human visual cortex, since it consists of a dynamic number of neurons that have altering pathways of different depths for each task, and taking advantage of the recently developed parallelized algorithms for solving multilayer neural networks, this approach is now applicable to remote sensing.

Different network architectures are used to achieve the goal of hyperspectral pixel classification. In Chen's first paper [3], autoencoders are used in stacks. An autoencoder (AE) consists of a single visible layer of inputs, a hidden layer, a reconstruction layer which becomes the output and activation functions, which provide nonlinearity. All layer transitions can be formulated by matrix multiplications, which are assumed to be transposes of each other and addition of a bias vector. The authors implemented 5 stacked autoencoder layers with three hidden layers in the middle. Logistic regression is used as activator function. Training is done via back propagation and the inputs are given in three different algorithms, in the first one only

spectral information was used and in the second one the resultant data tensor from a PCA operation that reduced the spectral dimensions to 4 is used and 11×11 patches around each pixel are extracted to be flattened to a 1D vector, while the third algorithm combined both inputs to a 1D vector as input. Error minimization is done by optimizing transformation matrix and the bias vectors for the classification task. Error is calculated from the cross-entropy of input patches while the learning rate is calculated via stochastic gradient descent algorithm. Fine tuning is done by successive learning of each autoencoder layers.

Experimental studies with the Kennedy Space Center and the Pavia hyperspectral images are done in four steps: 1) To test the efficiency of autoencoders, with an initial training phase of 6:2:2 distribution of GT pixels into train, validation and test samples under cross-validation. The performance measures were overall accuracy, average accuracy and kappa statistic. 2) Comparison against state-of-the-art SVM approach. 3) Classification accuracy of spatial-dominated features and 4) The results of the third algorithm. The results showed that a single layer AE with 100 hidden neurons can learn to reconstruct the image until 3500 epochs, while the filters that the model learned showed stark differences across different bands of the spectrum for both images and deep learning obviously took the lead in training and especially testing time, at the same time improved the classifications rates considerably in both spectral and joint spectral-spatial case. The authors did not even finish the pertraining and fine-tuning and claim that if continued on, the accuracy of SAE-LR model will increase.

In Tao's work[72], stacked sparse autoencoders (SSAE) are used, which are usually shallow feature extractors, however now generates sparse representations for the spectrum and enables for multiscale spatial features to be learned. Extracting features through a neural network helps the features to be better as they explain more and complex patterns and classification through Linear SVM therefore becomes possible with a complicated representation of the image at hand. In an attempt to follow the work of Chen [3], they also found out that similar scenes have similar features that can be directly transferable.

Chen's second work [73] tries another approach with a probabilistic neural network system called a Deep Belief Network (DBN) that is composed of three Re-

stricted Boltzmann Machines (RBMs). An RBM in its simplest form consists of one visible input layer and another hidden layer which are fully connected. A network of three stacked RBMs followed by a logistic regression (LR) layer will learn from whether the current layer can reconstruct the information in the previous layer and aims to do it in small learning rates, while checking on cross-entropy.

Romero's work [74] focuses on exploiting the rich information inside the multiple layers of the hyperspectral images. They propose a method for sparse representation of the spectrum by defining two concepts: Population sparsity and Lifetime sparsity, to be determined by a Greedy Layerwise Unsupervised Pretraining (GLUP) algorithm. The algorithm goes through all of the spectrum, extracts N sized patches and comparing it with the previous level of activation within that pact through both sparsities. Following a hysteresis threshold on activation and inhibition levels, the patch is either lit and gets features extracted or goes out. This sparse representation then are used as features to be fed to a Linear SVM to conclude this completely unsupervised method of classification.

Li's approach [75] pertains to a different family of approaches with Gabor wavelets. After a PCA step to extract the first fifty layers of importance, three dimensional versions of Gabor filters were formulated and calculated for each pixel through all remaining spectrum. The resulting bulk of the features are then sent to stacked autoencoders, while learning rate is kept in check by cross-entropy between the original and the reconstructed input vectors in the training phase. GLUP is again present in the training the autoencoder stack, followed by a backpropagation to fine tune with the addition of output sigmoid layer. Since this approach depends on empirical values at every stage and the absence of other comparisons with other state-of-the-art algorithms, it does not add much to the literature.

Convolutional Neural Networks (CNN) are also used in this field for three reasons. In Hu's work [76], CNN is used to get an input of pixel spectrum vector and after passing through a convolutional and pooling layer, it produces 20 different feature maps, which are then fed to fully connected layers (MLP) to classify. This paper is inspired from speech processing, where the input is also a 1D frequency vector. Backpropogation and gradient descent are used for the training on a logarithmic loss function and input normalization. It reports wild improvements on

the results. In Makantasis's paper [77], they introduce multiple convolutional layers and MLP for the classification purposes and give good comparison of the approach versus RBF and Linear SVM, other popular algorithms. Although they describe a method of Random-PCA, which destroys spectral information, they concluded that same objects have similar spectral response with low variance. In Castellucio's paper [78], CaffeNet and GoogLeNet are utilized, which are already trained networks that can either be retrained from scratch, fine tuned for a specific target of outcome or can be used to extract features directly. Their results show that fine tuning those networks is both time efficient and gives more accuracy in classification results.

Chapter 3

Combining Mathematical Morphology with Deep Learning

In this chapter, the approach to classify hyperspectral images will be presented. This thesis' contribution is to combine two well known methods, Extended Attribute Profiles (EAP) and a particular branch of deep learning methods called CNNs, to see if there is an improvement in the classification accuracy. In the particular scheme, AlexNet and GoogLeNet are considered as first trials, followed by two other classifiers, modAlexNet and ConfNet, which modifies and enhances AlexNet with different approaches. Those networks are used to extract features and make the classification through their fully connected and softmax layers, respectively.

3.1 Rationale

In the literature, there are approaches for hyperspectral image classification using CNNs, stacked autoencoders (SAE), Restricted Boltzman Machines (RBM) and multilayer perceptron (MLP) variants. Those networks have also been used to extract features to be fed to Support Vector Machine (SVM) or Random Forest (RF) classifiers. Finally, there are approaches using 2D Gabor Filters to extract features as well. This thesis defends that combining highly nonlinear, complex features that are generated by EAPs with CNNs would achieve greater accuracy, because EAP can handle spatial information and it can be extracted from hyperspectral images which would have hundreds of channels, where CNNs would fail because most kernel operations on CNN are defined on grayscale or RGB images, for 1 or 3 channels. In order to prove this point, this thesis will explore four combinations of linear (raw data, with patches) vs complex features (computed through EAPs) and almost linear (CNN) vs complex classifiers (RF).

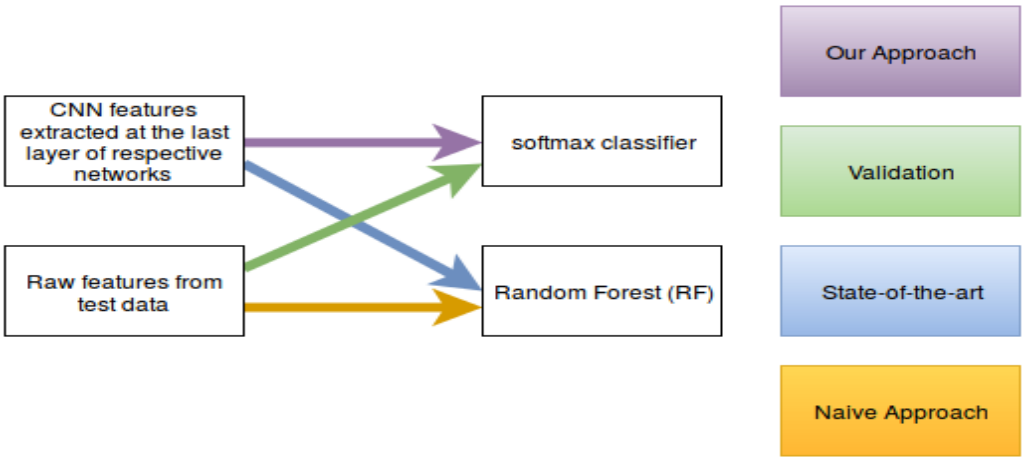


Figure 3.1: Rationale

3.1.1 Neural Network Selection

While doing the experiments, 5 different test scenarios are considered:

- Test 1: Spectral data from each pixel. This has been the standard approach for many years in this field, which results in $S \times 1$ input vectors for hyperspectral images of S channels corresponding to every pixel. This approach is the standard one, explained in Figure 2.1.

- Test 2: 9×9 patches of 4 PCs. This approach has been proposed by Chen et al [3], and it will be tested against the proposed network architectures. Its inputs will be flattened into vector inputs of size 324×1.

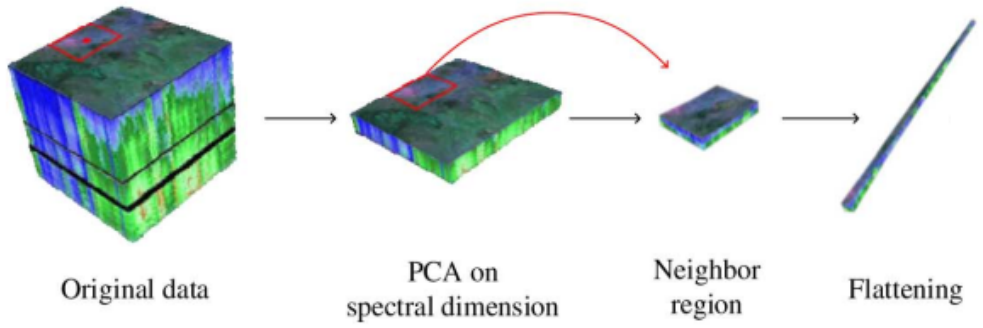


Figure 3.2: Test 2 approach: $9 \times 9 \times 4$ patches converted to 1×324 [3]

- Test 3: EAPs prepared with area as attribute, since area attribute is a widely used increasing attribute. They are extracted from 4 PCs and with different

thresholds and N thickening and thinning operations, totalling to $2N + 1$ profiles and $(8N + 4) \times 1$ input for each pixel.

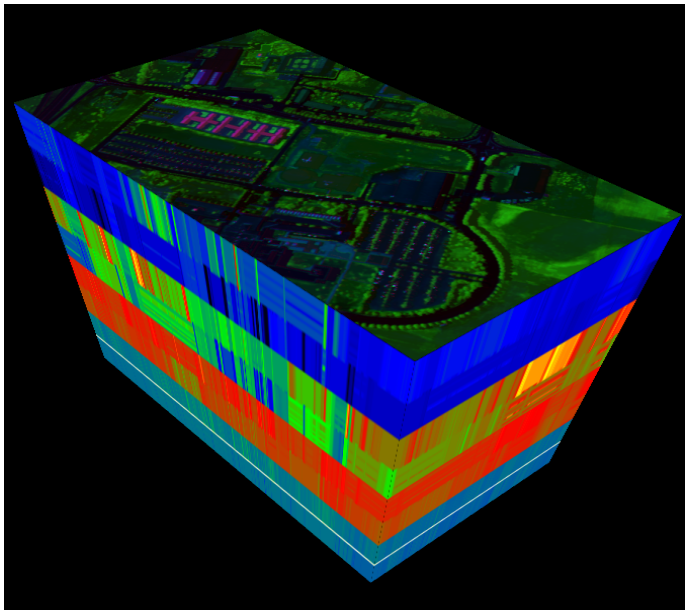


Figure 3.3: Test 3 approach: Area attribute is used for EAP, resulting in 1×116

- Test 4: EMAPs prepared with area and moment as attributes. Original features are preserved from Test 3, while moment profiles are added with K different thresholds for thickening and thinning on 4 PCs. This stage will add $8K$ more components to the input vector, totaling to $(8N + 8K + 4) \times 1$ for each pixel. Moment, unlike area, generates non-increasing profiles, which makes it the second attribute of choice.

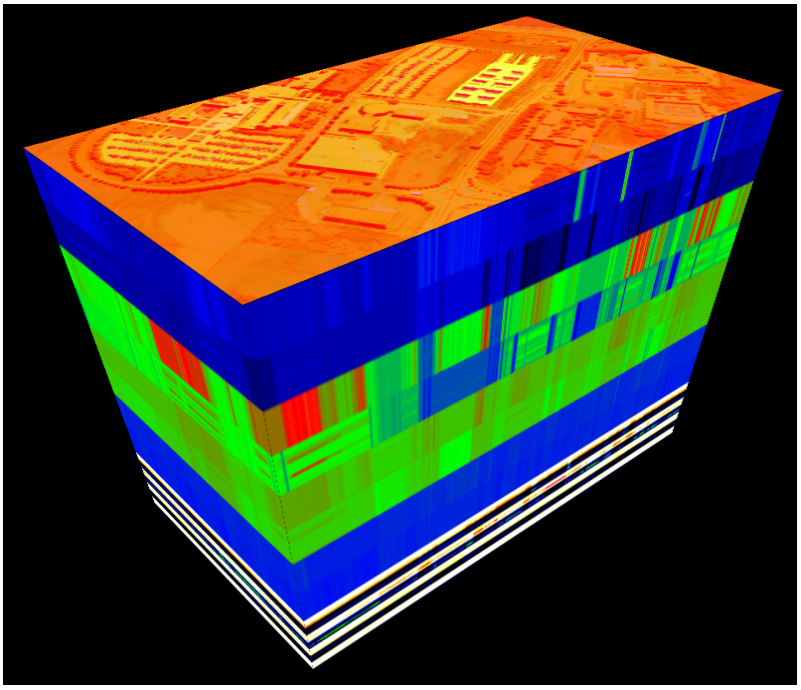


Figure 3.4: Test 4 approach: Area and moment attributes used for EMAP, resulting in 1×148 vectors for each pixel.

- Test 5: EMAPs prepared with area and moment as attributes, combined with spectral data. This stage will have all components from the previous test, with the addition of S channel spectral information, which would result in $(8N + 8K + 4 + S) \times 1$ input for each pixel. This test is aimed to explore whether adding spectral information on top of Test 4 would increase the accuracy.

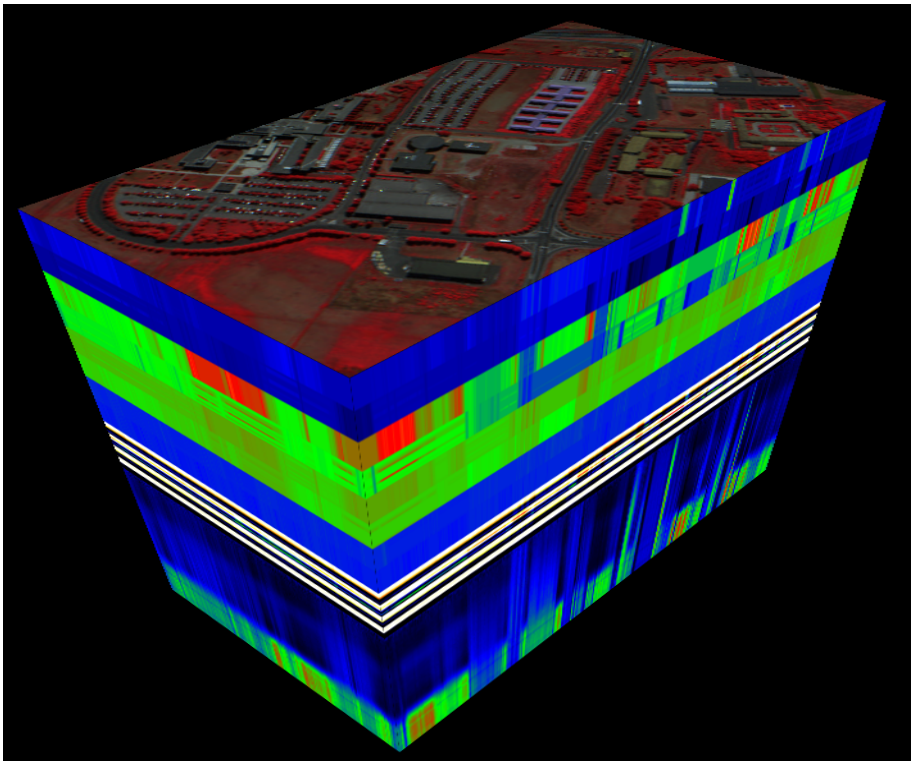


Figure 3.5: Test 5 approach: Addition of spectral profiles to that of Test 4.

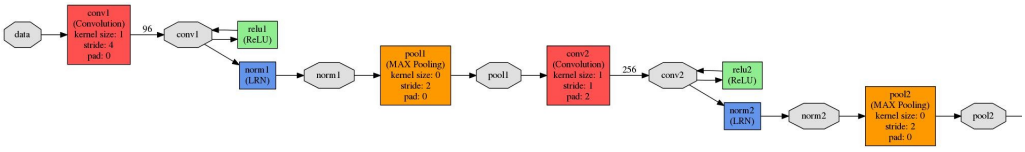
These test scenarios is given as input to 7 different classification scenarios to show their prowess. All network layers with convolution are followed by batch renormalization (LRN) for regularization and Rectified Linear Unit (ReLU) layers to model nonlinearity in perception. All pooling layers are max pooling unless stated otherwise:

1. AlexNet for vector inputs

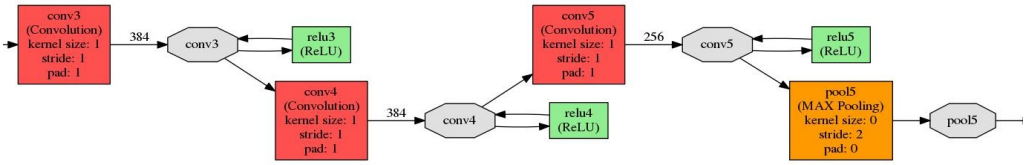
   This network consists of the following stages:

   - conv1 layer: 96 feature maps, produced by 1×11 receptive fields with a stride of 4 pixels,
   - pool1 layer: 1×3 kernel with a stride of 2 pixels,
   - conv2 layer: 256 feature maps, produced by 1×5 receptive fields, padded by 2×2 and grouped by 2,
   - pool2 layer: 1×3 kernel with 2 strides,
   - conv3 layer: 384 feature maps, produced by 1×3 receptive fields, padded by 1×1,

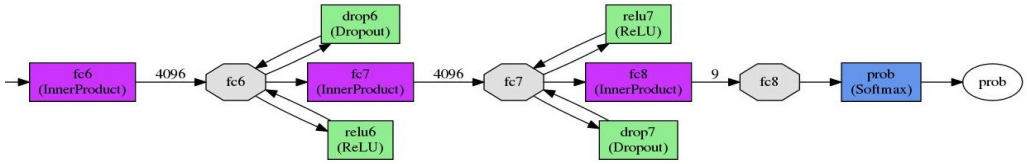
- conv4 layer: 384 feature maps, produced by 1×3 receptive fields, padded by 1×1 and grouped by 2,
- conv5 layer: 256 feature maps, produced by 1×3 receptive fields, padded by 1×1 and grouped by 2,
- pool5 layer: 1×3 kernel with a stride of 2 pixels,
- fc6 and fc7 layers: 4096 neurons each in two fully connected layers, followed by dropout operation of 0.5,
- fc8 layer: Fully connected layer that has a top of 9 classes which will be used for classification.



(a) First level of AlexNet



(b) Second level of AlexNet



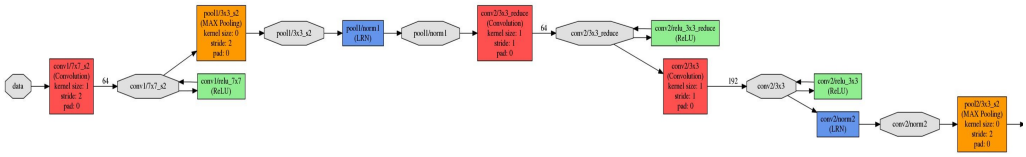
(c) Last level of AlexNet

Figure 3.6: AlexNet architecture

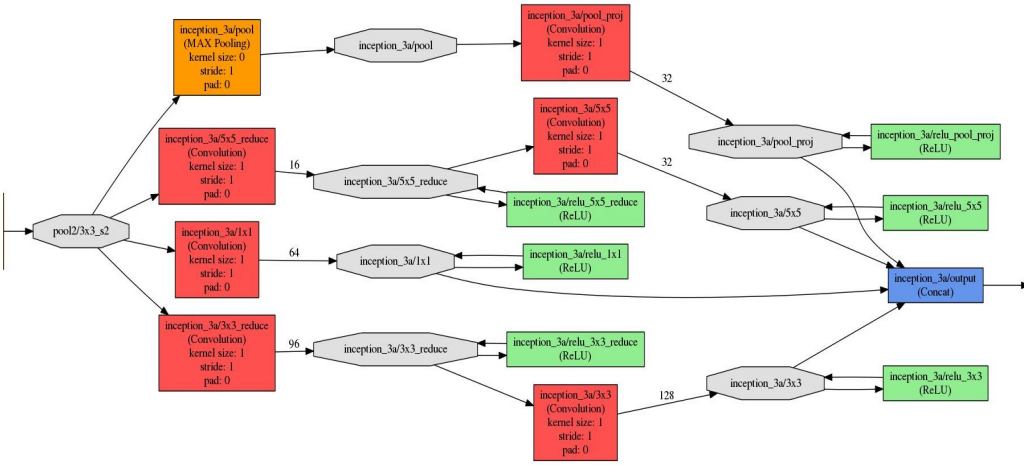
2. GoogLeNet for vector inputs: This network gained attention due to its inception layers approach. Although the first two layers of convolution and pooling are mundane:

- conv1: 64 feature maps, produced by 1×7 receptive fields, padded by 0×3 and a stride of 2 pixels, followed by ReLU
- pool1: 1×3 kernels with a stride of 2 pixels, followed by batch normalization (LRN)
- conv2/reduce: 64 feature maps, produced by 1×1 receptive fields
- conv2: 192 feature maps, produced by 1×3 receptive fields, padded by 0×1, followed by ReLU and LRN

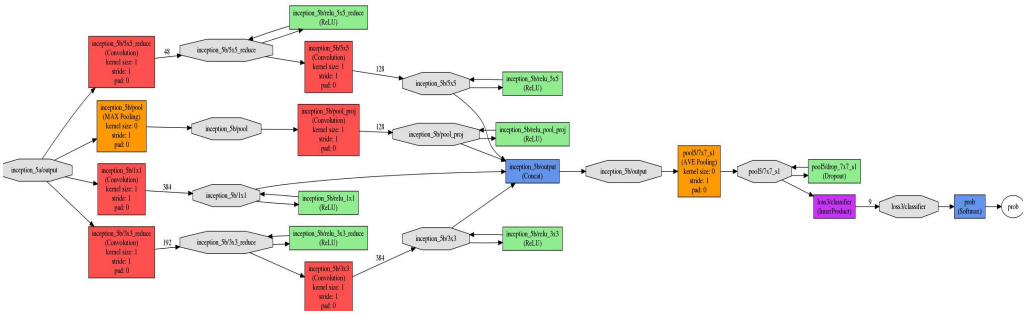
This network branches out and recollects at 9 different instances, always following a distinct pattern of receptive fields of 1×1, 1×3, 1×5 and the pooling layers to be concatenated at the end. By the time the 9th inception would be done, an average pooling step with 1×7 kernel, 0×3 padding and a stride of 1 pixel, followed by a dropout of 0.4 would give way to the last layer of 9 instances which will be used for classification.



(a) First level of GoogLeNet



(b) Repeating inception layers of GoogLeNet: There are 9 of them



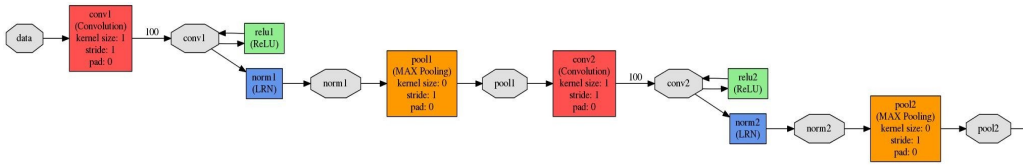
(c) Last level of GoogLeNet

Figure 3.7: GoogLeNet architecture

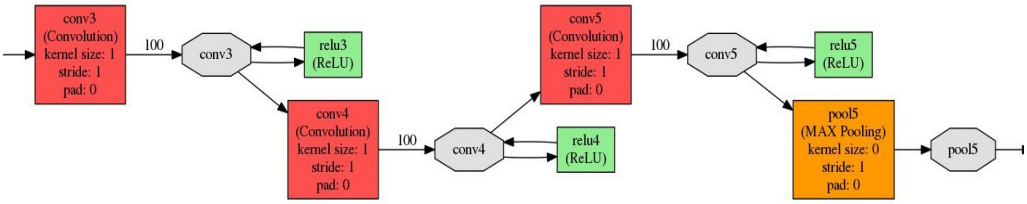
3. modAlexNet: A modified AlexNet, considering local data connectivity

   - conv1 layer: 100 feature maps, produced by 1×5 receptive fields,
   - pool1 layer: 1×2 kernel with a stride of 1 pixel,
   - conv2 layer: 100 feature maps, produced by 1×5 receptive fields, padded by 2×2 and grouped by 2,

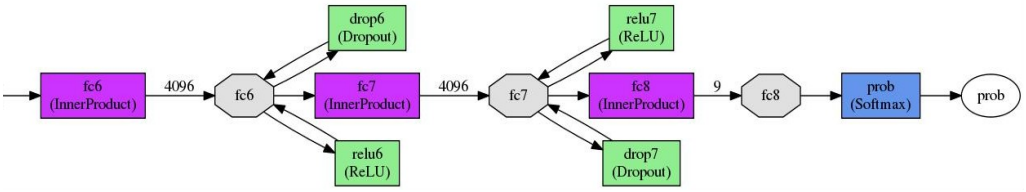
- pool2 layer: 1×2 kernel with a stride of 1 pixel,
- conv3 layer: 100 feature maps, produced by 1×5 receptive fields, padded by 1×1,
- conv4 layer: 100 feature maps, produced by 1×5 receptive fields, padded by 1×1 and grouped by 2,
- conv5 layer: 100 feature maps, produced by 1×5 receptive fields, padded by 1×1 and grouped by 2,
- pool5 layer: 1×2 kernel with a stride of 1 pixel,
- fc6 and fc7 layers: 4096 neurons each in two fully connected layers, always followed by dropout operation of 0.7,
- fc8 layer: Fully connected layer that has a top of 9 classes which will be used for classification.



(a) First level of modAlexNet



(b) Second level of modAlexNet



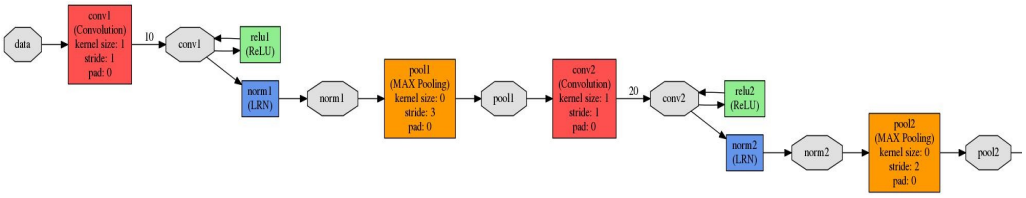
(c) Last level of modAlexNet

Figure 3.8: modAlexNet as a whole

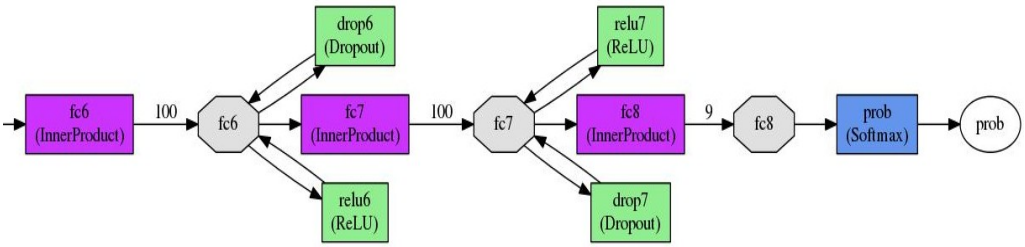
4. ConfNet: A recently proposed lightweight modification of AlexNet [79]. This network has only two convolutional and pooling layers, followed by the usual fully connected layers. Weight initialization scheme is also moved from Gaussian distribution to Xavier distribution, which is defined as follows:

    - conv1 layer: 10 feature maps, produced by 1×5 receptive fields with 1 stride. This layer's weight initialization had the standard deviation of 0.02, instead of the standard 0.01
    - pool1 layer: 1×3 kernel with a stride of 3 pixels,

- conv2 layer: 20 feature maps, produced by 1×5 receptive fields,
- pool2 layer: 1×3 kernel with a stride of 2 pixel,
- fc6 and fc7 layers: 100 neurons each in two fully connected layers, followed by dropout operation of 0.6,
- fc8 layer: Fully connected layer that has a top of 9 classes which will be used for classification.



(a) First level of ConfNet



(b) Second level of ConfNet

Figure 3.9: ConfNet as a whole

5. Raw input vectors fed into RF
6. Raw input vectors fed into softmax classifier
7. Deep features extracted from modified AlexNet's last fully connected layer (4096×1 input) fed into RF.

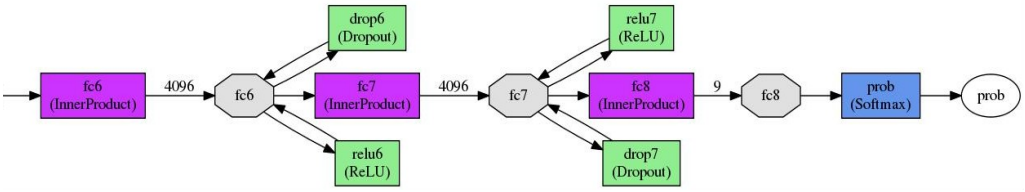


Figure 3.10: Feature extraction layer of modAlexNet

3.1.2 Ideas for Data Preparation

While supplying 1D vector input for each pixels is the simplest way to proceed with the pixel classification tasks, this section will elaborate on different approaches that have been taken during data preparation phase to see their effects on the classification results:

- Idea 0: Extracting EAP to be fed as 1D vector input for each pixel
- Idea A: 11×11 patches that would have all spectrum information, resulting in $121 \times S$ input matrices.
- Idea B: Concatenation of 9×9 patches of EAP images (1D vectors) using area profiles, resulting in 116×81 input matrices.
- Idea C: Concatenation of 9×9 patches of EAP images, resulting in 29x9x9x4->261x36 input matrices.
- Idea D: Multidimensional data input into Caffe environment is achieved by hdf5 data format, thus 9×9 patches with many different strategies became possible, resulting in $9 \times 9 \times C$ hdf5 data cubes, where C is the number of features per pixel. C dimension can represent whole spectrum, resultant channels after a PCA, EAP/EMAP features or a subset of all.

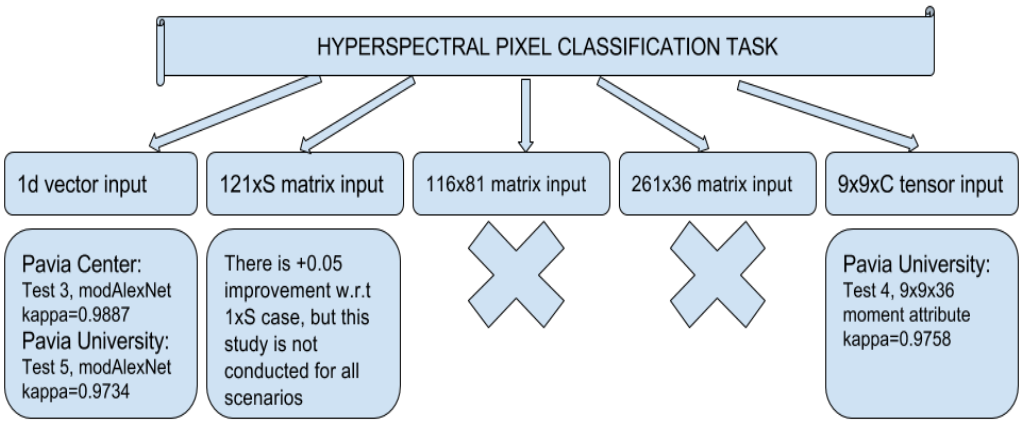


Figure 3.11: An overview of the ideas

As it turns out, only test 0, A and D would work with substantial improvement on accuracy, while test B would overfit and test C would not be learned by any of the networks that are presented up to now. In the end, 1D pixel vector approach and hdf5 multidimensional data input approach are used for the overall analysis and only the results of those two ideas will be presented in the results page.

3.1.3 Parameter and Hyperparameter Optimizations

Different parameters are considered during the training of deep networks, while using default parameters in some places. Fast learning is aimed, i.e., the learning rates are set as high as possible so that the loss function starts to decrease for good around 500-2000 iterations, only fluctuating mildly to produce the best parameters for the network.

- Parameter optimization for Random Forest classifiers are not done, the state-of-the-art assumption of 100 trees with square root of number of instances were used.
- For softmax, logistic regression toolbox of scikit-learn and Weka were utilized with their default parameters.
- For networks, rule of thumb values of learning rate = 0.01 and momentum 0.9 are used, while weight decay, learning strategies and all else remained the same for AlexNet and GoogleNet.

- For modAlexNet, weight decay would be 0.0002 (original value was 0.0005).
- For ConfNet, many observations were made until 10000 iterations each, until settling with Xavier weight initialization scheme, increasing conv1's initialization standard deviation to 0.02, setting weight decay at 0.0002 and solving the network with Adam solver.

During the training, since the learning strategies for all networks based on stochastic gradient descent, best model parameters fluctuated considerably during the training and validation phases. However, it is possible to capture a good model by snapshotting at 5000 iterations and using those saved models for classification purposes. Therefore, it is safe to claim that this procedure is applicable for further studies.

3.1.4 Efficiency

The computations are done on two different machines running on Ubuntu 14.04 LTS. Great care was given to match driver versions to avoid different seeds for random number generations.

1. Station 1: Intel Xeon 2 core processor with NVidia Quadro K4000 GPU
2. Station 2: Intel i7 HQ5700 processor with GeForce GTX 980M GPU

Running time for the first machine lasted 5.5 to 6 hours per training of an input size of 103×1 for 50000 iterations, classifications lasted 25 minutes for each saved model; while on the second machine training of the same scenario lasted for 40 minutes and classification for each saved model took only 7.5 minutes. As input sizes got larger, these numbers grow almost linearly: Reaching to 85 minutes for classification and 11.5 minutes for each classification on the second machine for the input size of 251×1, while it took more than a day on the other machine for training. Although both GPUs had DDR5 memory bus architectures, the main difference here was due to the other specifications of GPUs. Having more CUDA

GPU type	Memory Bandwidth	Memory Storage	CUDA cores	Clock Speed
1	134 GB/s	3 GB	768	400 Mhz
2	160 GB/s	4 GB	1536	1038 Mhz

Table 3.1: A comparison of GPUs: 1) Nvidia Quadro K4000 2) GeForce GTX 980M

cores, with more clock speed and memory bandwidth made the second station eclipse on timing achievements of station 1:

It should be noted that the second GPU remained utilized around 70-80 percent during training and testing of these testing scenarios, using a maximum of 2.5 GBs of memory and 100 Watts of power of its maximum 120 Watts. Should the data be bigger or bigger batch sizes to be used, these timings would improve even more so. However, as the size of the computations grows, there should be more investment in cooling technologies in order to avoid performance loss due to overheating above 75-80 °C, which can be monitored by nvidia-smi.

3.2 Datasets

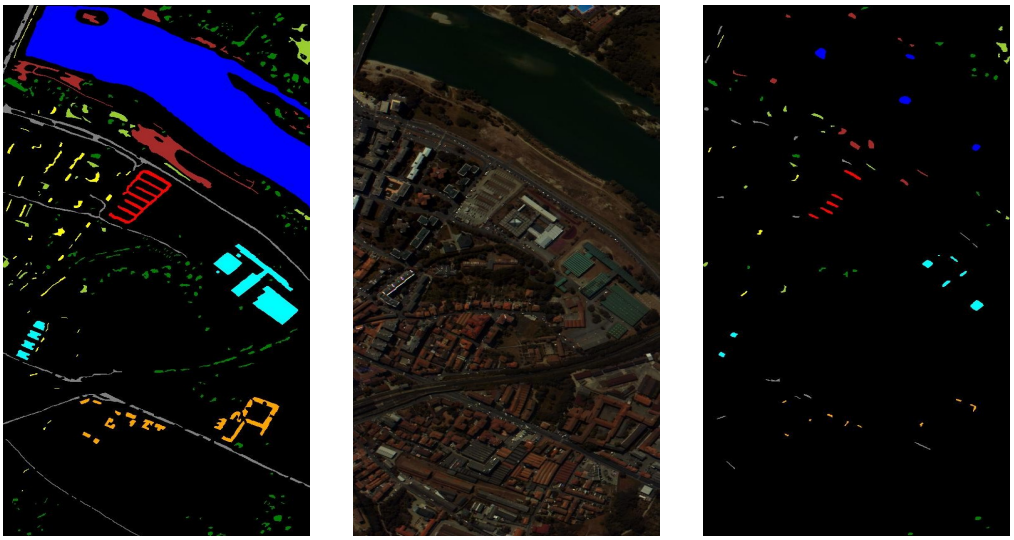
In this thesis, two hyperspectral data tensors that are used are the scenes acquired by the ROSIS sensor during a flight campaign over Pavia, northern Italy. Pavia Centre scene has 102 bands, whereas Pavia University has 103. Pavia Centre is a 1096×715 pixels at spatial dimension, and Pavia University is a 610×340 pixels image, but some of the samples in both images contain no information and have to be discarded before the analysis. The geometric resolution is 1.3 meters. Image ground truths are labelled with 9 different classes each. Pavia scenes were provided by Prof. Paolo Gamba from the Telecommunications and Remote Sensing Laboratory, Pavia university (Italy).

3.2.1 Pavia University Scene

In preparation of Pavia University Scene, original data is used with its 103 spectra present, except at the places where 4 PCs are used, which was done through PCA on the original data to cover at least 99 percent of the original data.

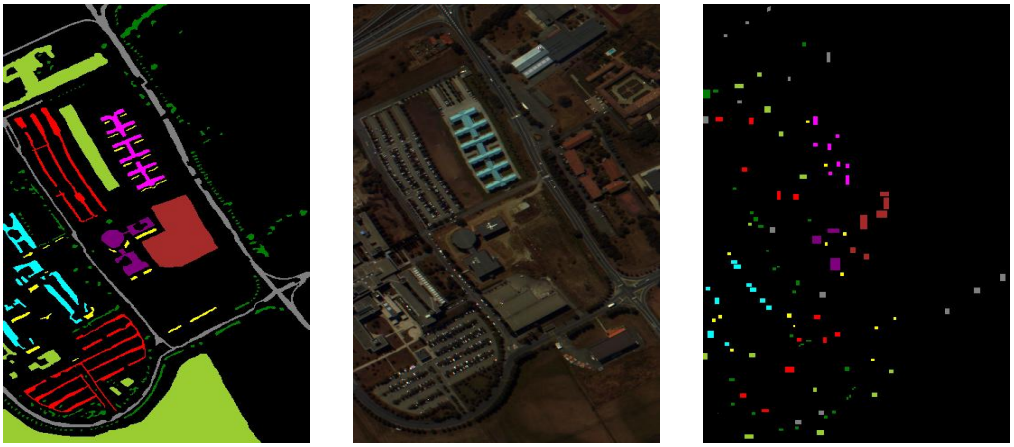
3.2.2 Pavia Center Scene

In preparation of Pavia Center Scene, only right side of the dataset is used as a single image, as seen on the Figure 3.1. Therefore, the final dimensions that are used for this dataset are 1096×489. PCA is done on the dataset, reducing 102 spectral dimensions to cover at least 99 percent of the original data, which happened to be 4 PCs.



(a) Ground Truth (b) False color image (c) Training samples

Figure 3.1: Pavia Center dataset



(a) Ground Truth (b) False color image (c) Training samples

Figure 3.2: Pavia University dataset

Chapter 4

Results

In this chapter, the results of different approaches within this thesis are presented, along with other state-of the art methods to compare the overall results. The experiments covers different network architectures that are either used as feature extractors or direct classification purposes with softmax classifiers combined, or raw features will be fed to softmax and Random Forest classifiers.

4.1 Methods

Given that hyperspectral image datasets contain information from many bands, they are a rich source of indicative power over individual pixels. As it is usually thought in image classification tasks, neighbouring regions are also considered to be of importance, since a group of pixels that are next to each other are more likely to form an object that would have similar labels. Another reason for exploiting neighbourhood is that if this information is never used, a classification algorithm would have given similar results in which the pixels are randomized, which would discard all of spatial information.

4.1.1 Spectral Signatures

In this section, the results obtained using spectral signature per pixel are displayed. Two different approaches are mainly considered:

- Spectral responses: Results in 103×1 pixel vector for Pavia University dataset, while Pavia Center would produce 102×1 pixel vector.

- Neighbourhood information from 4 PCs: 9×9 patches are extracted around each pixel, using 4 PCs, results in 324×1 pixel vector for both datasets after flattening.

The results are given in form of image and table results for comparison.

4.1.2 Extended Attribute Profiles

In this section, the results obtained using area and moment as an attribute are displayed. 14 thickening and 14 thinning profiles are used on 4 PCs, results in 116×1 input vectors for both datasets, while moment profiles are done with 4 thickening and 4 thinning profiles, which results in 32×1 input vectors. The results are given in the form of labelled image after classification and table results for comparison.

4.1.3 Combination

In this section, the approaches from the previous two are presented here. For Pavia University dataset, this would produce 251×1 vector input per pixel, while for Pavia Center University dataset, this would produce 250×1 vector input per pixel. The results are given in form of image and table results for comparison.

4.1.4 Multidimensional data approach

In this section, five different scenarios are considered.

- test1: Full spectral information: The resulting input data is $9 \times 9 \times 103$ per pixel around each original pixel.

- test2: PCA approach: The resulting input data is $9 \times 9 \times 4$ per pixel around each original pixel.

- test3: Area attribute approach: The resulting input data is $9 \times 9 \times 36$, where the third dimension is prepared using all of the 4 PCs. This attribute profiling consists of 9 profiles per PC: 4 thinning, 4 thickening and one original data.

- test4: Moment attribute approach: The resulting input data is $9 \times 9 \times 36$, where the third dimension is prepared using all of the 4 PCs. This attribute

profiling consists of 9 profiles per PC: 4 thinning, 4 thickening and one original data.

- test5: Combination approach: The resulting input data is $9 \times 9 \times 68$, where the third dimension is prepared using the approach from test3 and test4.

4.1.5 Results

Best results for both datasets are presented here.

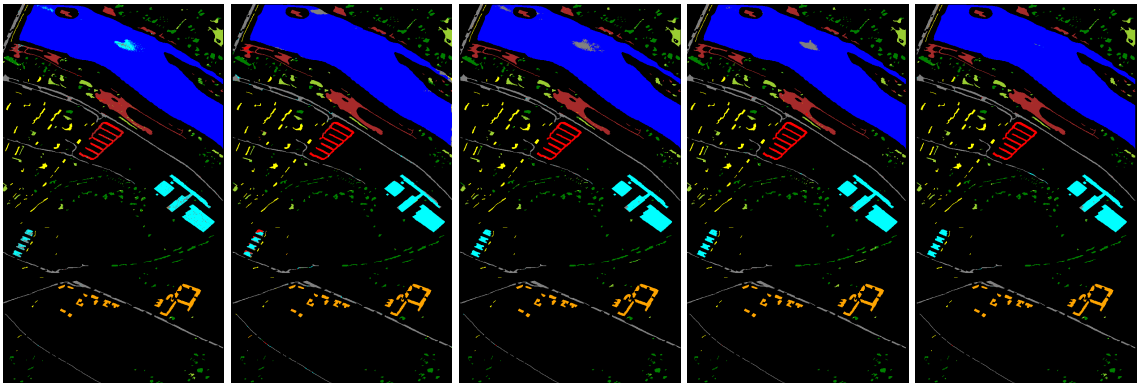
Pavia Center Results							
Test#	raw+SM	raw+RF	AlexNet	GoogleNet	ModAlexNet	CNN+RF	ConfNet
1	0.8624	0.9309	0.9501	0.9450	0.9615	0.9511	0.9654
2	0.9335	0.9532	0.9522	0.9421	0.9646	0.9155	0.9510
3	0.9327	0.9833	0.9724	0.9712	**0.9887**	0.8891	0.9821
4	0.9487	0.9843	0.9801	0.9836	0.9843	0.9080	0.9804
5	0.9511	0.9519	0.9872	0.9792	0.9867	0.9447	0.9871

Table 4.1: Pavia Center, best results with kappa statistic, SM = softmax

Compared to Pavia Center, Pavia University provided a harder challenge in terms of more noisy samples.

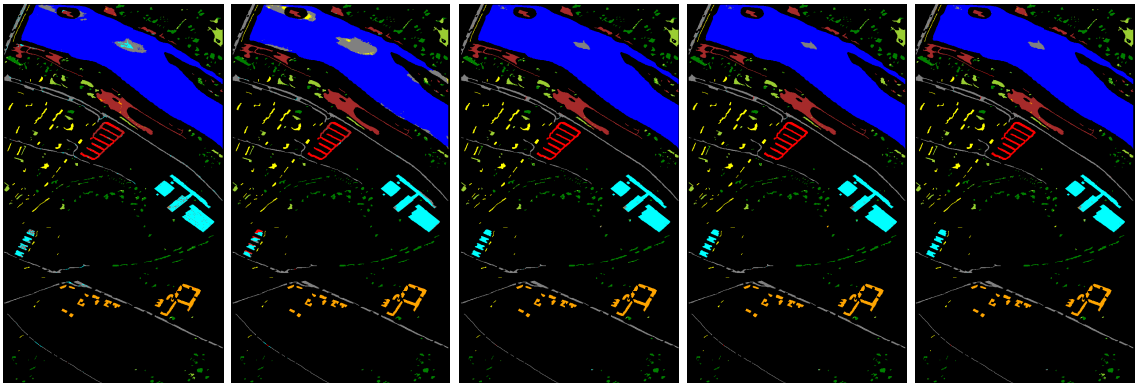
Pavia University Results							
Test#	raw+SM	raw+RF	AlexNet	GoogleNet	ModAlexNet	ConfNet	CNN+RF
1	0.5635	0.6541	0.8100	0.7802	0.7475	0.7858	0.5761
2	0.6174	0.8081	0.8100	0.8243	0.8063	0.8204	0.6117
3	0.7022	0.8847	0.9315	0.9388	0.9230	0.9303	0.8535
4	0.8209	0.8736	0.9488	0.9523	0.9367	0.9516	0.8514
5	0.8361	0.9024	0.9684	0.9602	**0.9734**	0.9575	0.7962

Table 4.2: Pavia University, best results with kappa statistics, SM = softmax



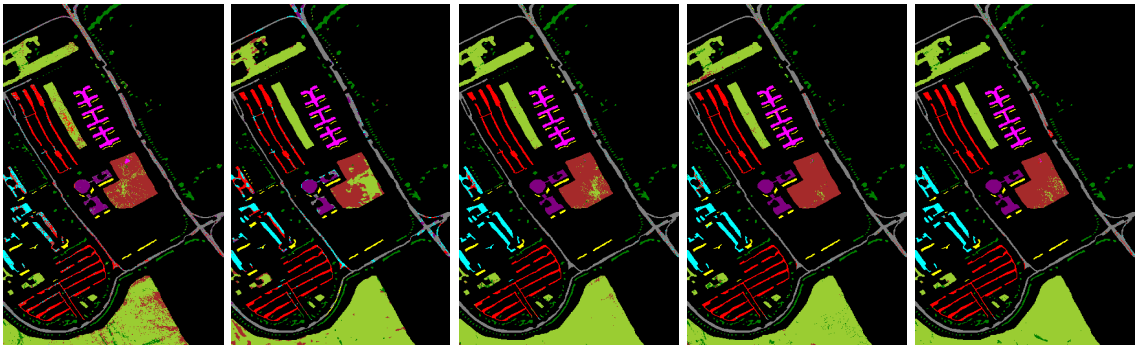
(a) test1 (b) test2 (c) test3 (d) test4 (e) test5

Figure 4.1: Pavia Center classification maps



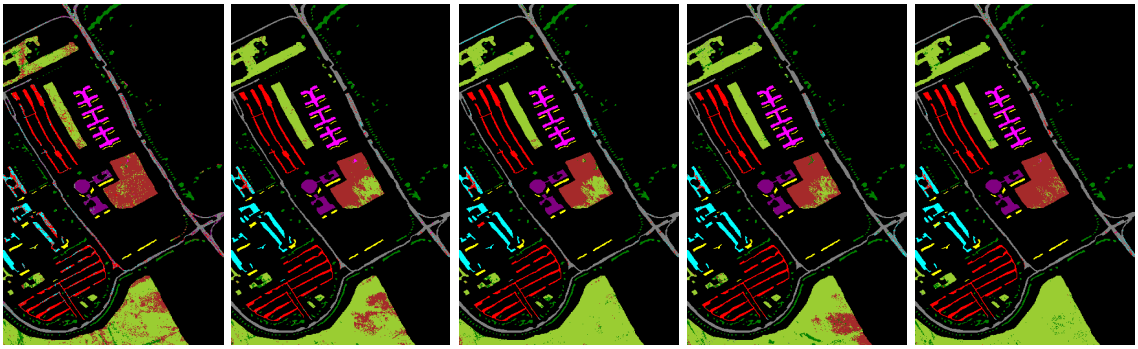
(a) test1 (b) test2 (c) test3 (d) test4 (e) test5

Figure 4.2: Pavia Center classification maps-RF-vector input



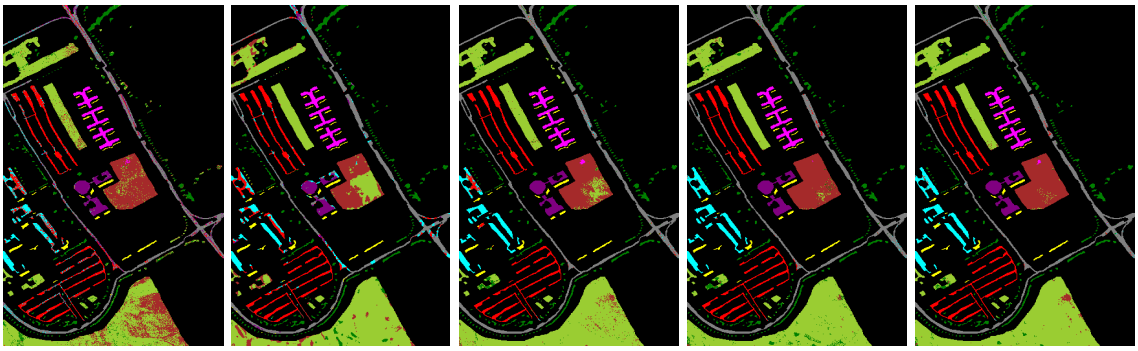
(a) test1 (b) test2 (c) test3 (d) test4 (e) test5

Figure 4.3: Pavia University classification maps-AlexNet-vector input



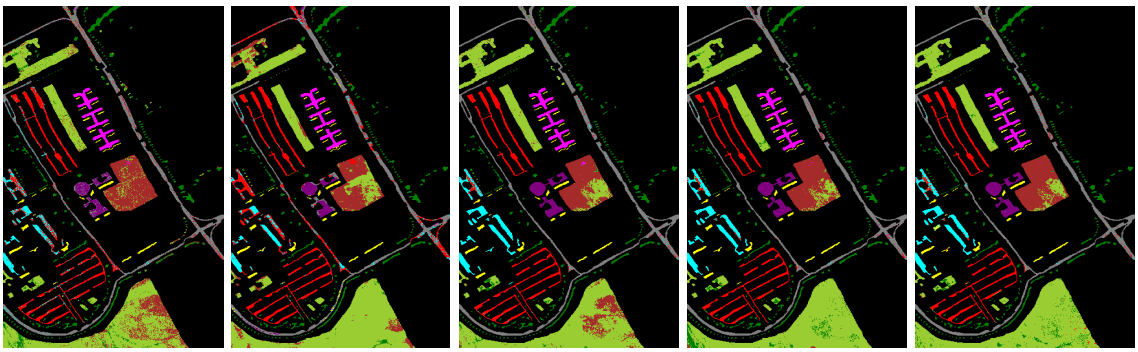
(a) test1 (b) test2 (c) test3 (d) test4 (e) test5

Figure 4.4: Pavia University classification maps-GoogLeNet-vector input



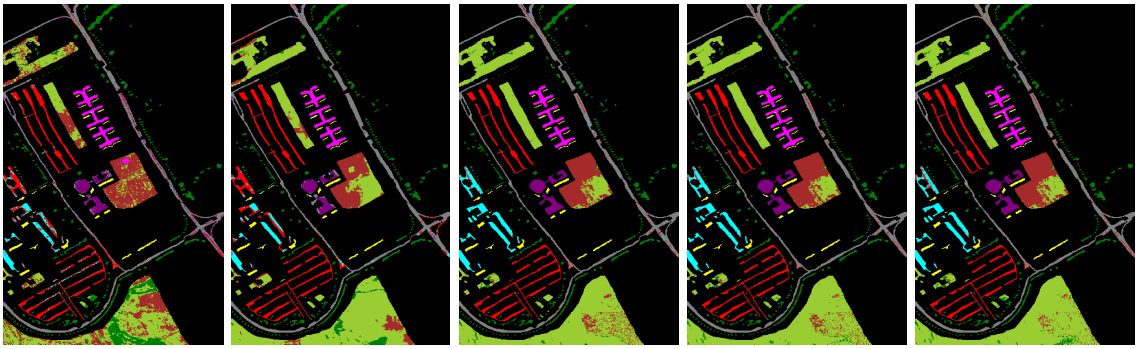
(a) test1 (b) test2 (c) test3 (d) test4 (e) test5

Figure 4.5: Pavia University classification maps-modAlexNet-vector input



(a) test1 (b) test2 (c) test3 (d) test4 (e) test5

Figure 4.6: Pavia University classification maps-confNet-vector input



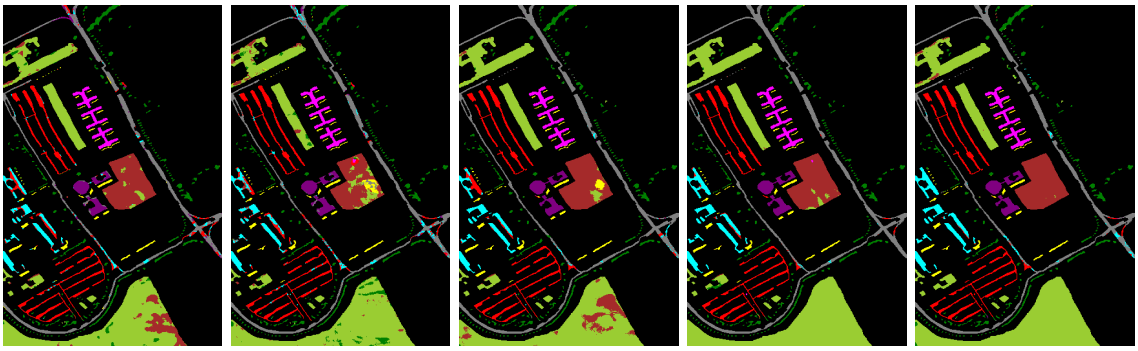
(a) test1 (b) test2 (c) test3 (d) test4 (e) test5

Figure 4.7: Pavia University classification maps-RF-vector input

Multidimensional input to CNN also improved the results with a certain characteristic, which will be discussed later in this chapter.

Pavia University Multidimensional Input Results		
Test#	Kappa	Description
1	0.8670	Patches prepared with whole spectrum ($9 \times 9 \times 103$)
2	0.8615	Patches prepared with PCAd version ($9 \times 9 \times 4$)
3	0.9261	Patches prepared with EAP with area attribute ($9 \times 9 \times 36$)
4	**0.9758**	Patches prepared with EAP with moment attribute ($9 \times 9 \times 36$)
5	0.9694	Patches prepared with EAP with both attributes ($9 \times 9 \times 68$)

Table 4.3: Pavia University, multidimensional approach



(a) test1 (b) test2 (c) test3 (d) test4 (e) test5

Figure 4.8: Pavia University classification maps-multidimensional approach

4.2 Discussion

In the field of hyperspectral imaging, it has always been a preferred method to include spatial domain information to the classification task instead of using only spectral signatures. In this study, it is shown that using neighborhood information improves the accuracy of the mapping by recognizing regions, where a group of pixels belong to same class.

Using CNNs, which has adjustable receptive fields and a sparsely connected network structure to identify regions of interest improved the accuracies versus a classification by RF, which would generate an ensemble of decision trees built by selecting random attributes and hence, makes statistical inference about the individual pixels. RF learners would therefore have no resilience against a misidentified or a heavily mixed pixel, which would introduce sizeable amount of noise into the description of that particular class.

Although the neighborhood information is useful, it is still prone to mislabelled instances and adds noise to the input data. In Pavia University dataset, dirt, grass and tree classes are usually confused by the learners and only with the introduction of EAP and EMAP by combining area and moment attributes helped with better identification of large areas with a single class label, which is usually traded off with errors in small sections of the map, i.e. small patch of the woods on another part of the map would be classified as grass instead.

Using different CNNs enabled a much better in-depth look at how these networks would work at different receptive fields, pooling and dropout operations. AlexNet and GoogLeNet turned out to be a good enough classifiers, yet they lacked the domain knowledge of the modAlexNet, which produced a better generalization of the classification map even under the noise that are described above. ConfNet, although it looked promising on the kappa statistics results, is shown to be no better than AlexNet when it comes to identifying and generalizing large areas with single class label.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis presented a comparative study of attribute profiles with area and moment attributes, and 5 different Convolutional Neural Networks to classify them, along with other commonly known approaches for a comparison. Most popular CNN choices of now, AlexNet and GoogLeNet proved their worth, while newly presented modAlexNet and recent submission of ConfNet provided with different angles in hyperspectral image classification by pixels and patches. Overall, CNNs proved to be better classifiers than off-the-shelf RF and displayed robustness in classification map generation with more interpretable results.

The experiments that were done on two different datasets from Pavia city provided different study opportunities on how hyperspectral imaging works, from data acquisition to dataset and label marking. As contributions despite those difficulties would finish here, proposals will follow for different strategies to build up the work from her.

This thesis solely focused on the pixel classification problem, which is burdened by the problems of this field. In this problem, ground truth pixels were labeled to certain classes of object and optionally, training set was also provided. The aim has been to classify the remainder of the instances optimally to generate a classification map for further uses. This study failed to obtain generalizations about a particular sensor or a class, but this study has been a broad comparative study on different CNNs and MM tools.

5.2 Future work

In this section, future research topics at Deep Learning side of the problem is presented.

- AlexNet and GoogLeNet used 1x5, 1x3 and 1x1 receptive fields, the proposed network modAlexNet used 1x5 receptive fields at all layers, so would ConfNet. Different receptive fields are not covered intensively or local relationship between adjacent spectral layers are not taken into account. It is recognized that some objects should have a distinctive pattern for their spectral signature, which would take other sizes of receptive fields and feature maps.

- Different strategies for pooling and dropout layers can also be explored, since pooling would generalize a region on that image, which is overly represented by the same area and moment attribute values and not all pixels of same label would have the same geometric attributes, i.e. tree class pixels may be grouped at one part of the map or they can be singular tree pixels, which would introduce further noise into training procedure, which might have caused for misclassification of some road pixels and small patches of dirt, trees and even bitumen. Dropout layers are also a must, but too little dropout would focus on the extracted features too much and too much dropout would not be able to learn.

- Neighboring relationships can be further explored by a locally region growing type of a learner [80]. In the cited work, the training phase is done by starting from labeled pixels and then classifying their neighboring unlabeled pixels to grow regions. The authors reportedly have success to model this behavior on a MLP to classify all of the hyperspectral image.

- Feature extraction phases (until fc8) could be refined to include a ranking scheme for population and lifetime sparsity to mitigate for excessive zero'd and same leveled attributes. This will especially come handy at CNN+RF step, since CNN tends to extract all the features from at the end of convolutional layers to fully connected layers and dropout layers, where most of the data would be either too much close to each other or exactly equal to zero due

to dropout. That might even be the reason why CNN+RF approach did not work as planned for ConfNet, which is a very small network with high dropout rate and worked the best for modAlexNet, which is a heavy network with most number of parameters overall.

- Different methods for fine tuning and data augmentation might also be explored. In this thesis, data augmentation is tried for the multidimensional data input phase, but it was only done with training from scratch mode and on training dataset with mirrors and flips at 45°, but adding noise to the input data is also a perfectly valid data augmentation method. For fine tuning, 3 PC patch images can be used with a pretrained network with generic aerial images or color/panchromatic images over the same area and then using this network to fine tune the network and increase the accuracy.

Bibliography

[1] D. Wei. "mNeuron: A Matlab Plugin to Visualize Neurons from Deep Models", 2015. [Online]. Available: http://vision03.csail.mit.edu/cnn_art/index.html

[2] I. G. Y. Bengio and A. Courville, "Deep learning," 2016, book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org

[3] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[4] A. K. Katsaggelos, *Digital image restoration*. Springer Publishing Company, Incorporated, 2012.

[5] G. Camps-Valls, D. Tuia, L. Gomez-Chova, S. Jimenez, and J. Malo, "Remote sensing image processing. synthesis lectures on image, video, and multimedia processing." Morgan and Claypool, Tech. Rep., 2011.

[6] D. Tuia and J. Munoz-Mari, "Learning user's confidence for active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 872–880, 2013.

[7] D. Tuia, F. Ratle, F. Pacifici, M. F. Kanevski, and W. J. Emery, "Active learning methods for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2218–2232, 2009.

[8] J. M. Amigo, "Practical issues of hyperspectral imaging analysis of solid dosage forms," *Analytical and bioanalytical chemistry*, vol. 398, no. 1, pp. 93–109, 2010.

[9] D. Tuia, G. Camps-Valls, G. Matasci, and M. Kanevski, "Learning relevant image features with multiple-kernel classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 10, pp. 3780–3791, 2010.

[10] R. Roesser, "A discrete state-space model for linear image processing," *IEEE Transactions on Automatic Control*, vol. 20, no. 1, pp. 1–10, 1975.

[11] A. R. Weeks, *Fundamentals of electronic image processing*. SPIE Optical Engineering Press Bellingham, 1996.

[12] P. Vogt, K. H. Riitters, C. Estreguil, J. Kozak, T. G. Wade, and J. D. Wickham, "Mapping spatial patterns with morphological image processing," *Landscape ecology*, vol. 22, no. 2, pp. 171–177, 2007.

[13] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[14] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[15] N. Longbotham, F. Pacifici, B. Baugh, and G. Camps-Valls, "Prelaunch assessment of worldview-3 information content," in *IEEE GRSS Workshop on Hyperspectral Image and Signal Processing (WHISPERS)*, 2014, pp. 479–486.

[16] G. P. Asner, D. E. Knapp, T. Kennedy-Bowdoin, M. O. Jones, R. E. Martin, J. Boardman, and C. B. Field, "Carnegie airborne observatory: in-flight fusion of hyperspectral imaging and waveform light detection and ranging for three-dimensional studies of ecosystems," *Journal of Applied Remote Sensing*, vol. 1, no. 1, pp. 013 536–013 536, 2007.

[17] G. R. Hunt, "Spectral signatures of particulate minerals in the visible and near infrared," *Geophysics*, vol. 42, no. 3, pp. 501–513, 1977.

[18] H. Van derWerff, M. Van der Meijde, F. Jansma, F. Van der Meer, and G. J. Groothuis, "A spatial-spectral approach for visualization of vegetation stress resulting from pipeline leakage," *Sensors*, vol. 8, no. 6, pp. 3733–3743, 2008.

[19] A. Barducci, D. Guzzi, P. Marcoionni, and I. Pippi, "Aerospace wetland monitoring by hyperspectral imaging sensors: A case study in the coastal zone of san rossore natural park," *Journal of environmental management*, vol. 90, no. 7, pp. 2278–2286, 2009.

[20] C. C. Lelong, P. C. Pinet, and H. Poilvé, "Hyperspectral imaging and stress mapping in agriculture: a case study on wheat in beauce (france)," *Remote sensing of environment*, vol. 66, no. 2, pp. 179–191, 1998.

[21] N. M. Nasrabadi, "Hyperspectral target detection: An overview of current and future challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 34–44, 2014.

[22] P. Ghamisi, "Spectral and spatial classification of hyperspectral data," Ph.D. dissertation, University of Iceland, 2015.

[23] C.-I. Chang, *Hyperspectral imaging: techniques for spectral detection and classification*. Springer Science & Business Media, 2003, vol. 1.

[24] P. Ghamisi, M. Dalla Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2335–2353, 2015.

[25] J. A. Benediktsson, K. Arnason, and M. Pesaresi, "The use of morphological profiles in classification of data from urban areas," in *Remote Sensing and Data Fusion over Urban Areas, IEEE/ISPRS Joint Workshop 2001.* IEEE, 2001, pp. 30–34.

[26] J. Grodecki, "Ikonos stereo feature extraction–rpc approach," in *ASPRS annual conference St. Louis*, 2001.

[27] M. Dalla Mura, J. Atli Benediktsson, B. Waske, and L. Bruzzone, "Extended profiles with morphological attribute filters for the analysis of hyperspectral data," *International Journal of Remote Sensing*, vol. 31, no. 22, pp. 5975–5991, 2010.

[28] P. Salembier, A. Oliveras, and L. Garrido, "Antiextensive connected operators for image and sequence processing," *IEEE Transactions on Image Processing*, vol. 7, no. 4, pp. 555–570, 1998.

[29] E. R. Urbach, J. B. Roerdink, and M. H. Wilkinson, "Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 272–285, 2007.

[30] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2013.

[31] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 480–491, 2005.

[32] J. A. Palmason, J. A. Benediktsson, J. R. Sveinsson, and J. Chanussot, "Classification of hyperspectral data from urban areas using morphological preprocessing and independent component analysis," in *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS'05.*, vol. 1. IEEE, 2005, pp. 4–pp.

[33] J. C.-W. Chan and D. Paelinckx, "Evaluation of random forest and adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery," *Remote Sensing of Environment*, vol. 112, no. 6, pp. 2999–3011, 2008.

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

[35] A. Plaza, P. Martinez, J. Plaza, and R. Pérez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morpho-

logical transformations," *IEEE Transactions on Geoscience and remote sensing*, vol. 43, no. 3, pp. 466–479, 2005.

[36] A. Plaza, P. Martínez, R. Pérez, and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 9, pp. 2025–2041, 2002.

[37] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[38] R. S. Sutton, "Temporal credit assignment in reinforcement learning," Ph.D. dissertation, University of Massachusetts at Amherst, 1984.

[39] C. F. Gauss, *Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss.* sumtibus Frid. Perthes et IH Besser, 1809.

[40] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[41] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[42] ——, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[43] S. Viglione, "4 applications of pattern recognition technology," *Mathematics in Science and Engineering*, vol. 66, pp. 115–162, 1970.

[44] A. Ivakhnenko, "Polynomial theory of complex systems," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 4, pp. 364–378, 1971.

[45] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[46] R. D. Joseph, *Contributions to perceptron theory.* Cornell Univ., 1961.

[47] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," DTIC Document, Tech. Rep., 1961.

[48] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[49] M. Minsky, "A framework for representing knowledge," *MIT Press*, 1975.

[50] D. Cervier, "Ai: The tumultuous search for artificial intelligence," 1993.

[51] P. J. Werbos, "Applications of advances in nonlinear sensitivity analysis," in *System modeling and optimization*. Springer, 1982, pp. 762–770.

[52] Y. LeCun, "Une procedure d'apprentissage pour reseau a seuil asymmetrique (a learning scheme for asymmetric threshold networks)," 1985.

[53] Y. Le Cun, D. Touresky, G. Hinton, and T. Sejnowski, "A theoretical framework for back-propagation," in *The Connectionist Models Summer School*, vol. 1, 1988, pp. 21–28.

[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," DTIC Document, Tech. Rep., 1985.

[55] H. B. Barlow, "Unsupervised learning," *Neural computation*, vol. 1, no. 3, pp. 295–311, 1989.

[56] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural computation*, vol. 2, no. 4, pp. 490–501, 1990.

[57] R. Battiti, "First-and second-order methods for learning: between steepest descent and newton's method," *Neural computation*, vol. 4, no. 2, pp. 141–166, 1992.

[58] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[59] G. E. Hinton, M. Revow, and P. Dayan, "Recognizing handwritten digits using mixtures of linear models," *Advances in neural information processing systems*, pp. 1015–1022, 1995.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[61] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[62] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 3546–3554.

[63] A. Ng, J. Ngiam, C. Foo, and Y. Mai, "Deep learning," 2014.

[64] P. O. Glauner, "Comparison of training methods for deep neural networks," *arXiv preprint arXiv:1504.06825*, 2015.

[65] Y. Dauphin, "Advances in scaling deep learning algorithms," 2016.

[66] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[67] C. L. Scofield, "N-dimensional coulomb neural network which provides for cumulative learning of internal representations," Jan. 30 1990, uS Patent 4,897,811.

[68] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," 1990.

[69] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embed-

ding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[70] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cudnn: Efficient primitives for deep learning," *arXiv preprint arXiv:1410.0759*, 2014.

[71] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, "Basic linear algebra subprograms for fortran usage," *ACM Transactions on Mathematical Software (TOMS)*, vol. 5, no. 3, pp. 308–323, 1979.

[72] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2438–2442, 2015.

[73] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2381–2392, 2015.

[74] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1349–1362, 2016.

[75] J. Li, L. Bruzzone, and S. Liu, "Deep feature representation for hyperspectral image classification," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2015, pp. 4951–4954.

[76] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, 2015.

[77] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2015, pp. 4959–4962.

[78] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *arXiv preprint arXiv:1508.00092*, 2015.

[79] M. Salman and S. E. Yüksel, "Hyperspectral data classification using deep convolutional neural networks," in *2016 24th Signal Processing and Communication Application Conference (SIU)*. IEEE, 2016, pp. 2129–2132.

[80] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2271–2282, 2010.