

TECHNICAL UNIVERSITY OF CRETE  
SCHOOL OF ELECTRONIC & COMPUTER ENGINEERING  
DIGITAL IMAGE & SIGNAL PROCESSING LABORATORY



**Sparse Representations in Machine Learning and  
Remote Sensing**

by

**Konstantinos T. Karalas**

A thesis submitted for the degree of *Master of Science* in  
*Electronic and Computer Engineering*

Crete, November 2015

Thesis Committee:

Professor Michael Zervakis, Supervisor

Professor Panagiotis Tsakalides, University of Crete

Associate Professor Katerina Mania



# Abstract

Land cover maps are critical for environmental monitoring and urban development among others. Unfortunately, in order to produce such maps significant labor intensive effort is required by human annotators through field-studies. Interestingly, on the other hand, high resolution imaging systems onboard airborne and spaceborne platforms are able to capture rich information in parts of the electromagnetic spectrum that the human eye cannot discern. This remote sensing imagery can be used to overcome the issues associated with field-studies, providing global and up-to-date land cover maps. Typically, during the mapping procedure, each remotely sensed pixel is classified into a single class, leading to very coarse representations. In the past few years, the development of the powerful framework of multi-label learning, where instances may be associated with multiple labels simultaneously, has been successfully applied in various computer vision scenarios. Part of the success is also attributed to the development of hand-crafted features which can dramatically boost the performance under specific conditions, however, these features are very specialized and lack universality.

This thesis introduces a radically novel approach for inferring the complex relationships between multispectral satellite imagery and spectral profiles of different surface materials, exploiting the proliferation of remote sensing imagery, through the introduction of the multi-label classification framework. The adoption of this scheme provides a real-world answer to the scale incompatibility problem between remote sensing imagery and ground-based measurements, since they naturally come in different spatial resolutions. Furthermore, instead of relying on specialized features, we propose the application of deep feature learning with stacked sparse autoencoders, in order to automatically extract meaningful features identifying the

underlying explanatory patterns hidden in low level satellite data.

To validate the merits of the proposed approach, we consider real contemporary data from the European Environment Agency for generating the ground-truth, and multispectral images from the Moderate-resolution Imaging Spectroradiometer sensor for feature extraction. We present results using several state-of-the-art multi-label learning classifiers and evaluate their predictive performance under different challenging scenarios, including cases where training is localized in a specific area and time, while testing takes place on a different location or time instance. Experimental results suggest that the proposed framework can achieve excellent prediction accuracy, even from a limited number of diverse training examples, whereas the application of feature learning leads to more representative features that can significantly boost the performance of multi-label classification problems.



# Περίληψη

Οι χάρτες κάλυψης της γης είναι αποφασιστικής σημασίας για την παρακολούθηση του περιβάλλοντος και της αστικής ανάπτυξης μεταξύ άλλων. Δυστυχώς, για να παραχθούν τέτοιοι χάρτες απαιτείται σημαντική και εντατική ανθρωποπροσπάθεια για τον κατάλληλο σχολιασμό τους μέσα από μελέτες. Κατά έναν ενδιαφέροντα τρόπο, από την άλλη πλευρά, υψηλής ανάλυσης απεικονιστικά συστήματα που βρίσκονται πάνω σε εναέριες και διαστημικές πλατφόρμες, αντλούν πλούσια πληροφορία από μέρη του ηλεκτρομαγνητικού φάσματος που το ανθρώπινο μάτι δε μπορεί να διακρίνει. Αυτές οι τηλεπισκοπικές εικόνες μπορούν να χρησιμοποιηθούν για να ξεπεραστούν τα θέματα που σχετίζονται με τις μελέτες, παρέχοντας παγκόσμιους και ενημερωμένους χάρτες κάλυψης γης. Συνήθως, κατά τη διαδικασία χαρτογράφησης, κάθε τηλεπισκοπικό εικονοστοιχείο ταξινομείται σε μία κλάση, οδηγώντας σε πολύ αδρές αναπαραστάσεις. Τα τελευταία χρόνια, η ανάπτυξη του ισχυρού πλαισίου της εκμάθησης πολλαπλών ετικετών, όπου τα δείγματα μπορούν να συσχετιστούν με πολλές ετικέτες ταυτόχρονα, έχει εφαρμοστεί με επιτυχία σε διάφορα σενάρια υπολογιστικής όρασης. Μέρος της επιτυχίας αυτής αποδίδεται και στην ανάπτυξη χειροποίητων χαρακτηριστικών τα οποία μπορούν να δώσουν δραματική ώθηση στην απόδοση κάτω από συγκεκριμένες συνθήκες, παρόλο που αυτά τα χαρακτηριστικά είναι πολύ εξειδικευμένα και στερούνται καθολικότητας.

Η παρούσα διπλωματική εργασία θέτει στόχο την εισαγωγή μιας ριζικά νέας προσέγγισης για να συνάγει τους σύνθετους δεσμούς μεταξύ των επίκτητων δορυφορικών εικόνων και των φασματικών υπογραφών από διαφορετικά είδη υλικών που βρίσκονται στην επιφάνεια της γης, αξιοποιώντας την ολοένα μεγαλύτερη διαθεσιμότητα τηλεπισκοπικών εικόνων μέσα από την εισαγωγή της ταξινόμησης που συνδέεται με παραπάνω από μία ετικέτες. Η υιοθέτηση αυτού του σχήματος παρέχει μία γνήσια απάντηση στο

πρόβλημα ασυμβατότητας της κλίμακας μεταξύ των τηλεπισκοπικών εικόνων και των χερσαίων μετρήσεων, αφού εκ φύσεως διατίθενται σε διαφορετικές χωρικές αναλύσεις. Επιπλέον, αντί να στηριζόμαστε στα εξειδικευμένα χαρακτηριστικά, προτείνουμε την εφαρμογή της βαθιάς μάθησης χαρακτηριστικών με στοιβαγμένους αραιούς αυτοκωδικοποιητές, έτσι ώστε να εξάγουμε αυτόματα χαρακτηριστικά μεστού περιεχομένου ικανά να αναγνωρίσουν τους διευκρινιστικούς παράγοντες που υποκρύπτονται σε χαμηλού επιπέδου δορυφορικά δεδομένα.

Για την επικύρωση των πλεονεκτημάτων της προσέγγισής μας, θεωρούμε αληθινά και σύγχρονα δεδομένα από τον Ευρωπαϊκό Οργανισμό Περιβάλλοντος για τη δημιουργία του πίνακα αληθείας, και πολυφασματικές εικόνες από τον Μέτριας-ανάλυσης Οπτικό Φασματοφωτομετρικό αισθητήρα για την εξαγωγή των χαρακτηριστικών. Παρουσιάζουμε αποτελέσματα χρησιμοποιώντας μερικούς ταξινομητές μάθησης πολλαπλών ετικετών τελευταίας τεχνολογίας, των οποίων η προβλεπτική ικανότητα αξιολογείται σε διάφορες απαιτητικές συνθήκες, συμπεριλαμβανομένων περιπτώσεων όπου η εκπαίδευση εντοπίζεται σε συγκεκριμένο τόπο και χρόνο, ενώ η εξέταση λαμβάνει χώρα σε διαφορετική τοποθεσία ή χρονική στιγμή. Τα πειραματικά αποτελέσματα αποδεικνύουν πως το προτεινόμενο πλαίσιο μπορεί να επιτύχει μια άριστη ακρίβεια πρόβλεψης, ακόμη και με έναν περιορισμένο αριθμό διαφορετικών δειγμάτων εκπαίδευσης, ενώ η εφαρμογή της εκμάθησης χαρακτηριστικών οδηγεί σε πιο αντιπροσωπευτικά χαρακτηριστικά που μπορούν να δώσουν σημαντική ώθηση στην απόδοση σε προβλήματα πολλαπλών ετικετών.

# Acknowledgements

This master's thesis was conducted to fulfill the requirements of the postgraduate program of the Electronic and Computer Engineering School of the Technical University of Crete. Nevertheless, the main part of this work was carried out at the Institute of Computer Science at Foundation for Research and Technology (FORTH). I am grateful to my supervisor Professor Michalis Zervakis for his flexibility and patience, as well as Professor Panagiotis Tsakalides at FORTH who believed in me from the beginning and supported me greatly. I would also like to thank Grigorios Tsagkatakis for his advice and insights that greatly contributed to the conclusion and enrichment of this thesis. Our numerous discussions have been extremely valuable for me. Finally, I would like to say a big thank you to my family for supporting me totally and unconditionally through all these years of my studies.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Abstract in Greek</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xix</b>
<b>List of Acronyms</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Thesis Motivations . . . . .	2
1.3 Thesis Contributions . . . . .	4
1.4 Thesis Outline . . . . .	6
<b>2 Related Work</b>	<b>9</b>
2.1 Remote Sensing Mapping and Classification . . . . .	9
2.2 Spectral Unmixing . . . . .	10
2.3 Unsupervised Representation Learning . . . . .	13
<b>3 Multi-label Learning</b>	<b>15</b>
3.1 Principal Concept . . . . .	15
3.2 Challenges in Multi-label Learning . . . . .	16
3.3 Problem Statement . . . . .	17

3.4	Classification Algorithms . . . . .	19
3.4.1	Problem transformation methods . . . . .	20
3.4.2	Algorithm adaptation methods . . . . .	21
3.4.3	Ensemble methods . . . . .	22
3.5	Dimensionality Reduction . . . . .	24
3.6	Evaluation Measures . . . . .	25
<b>4</b>	<b>Feature Learning</b>	<b>29</b>
4.1	Artificial Neural Networks . . . . .	29
4.2	Autoencoders Framework . . . . .	31
4.2.1	Single-layer sparse autoncoders . . . . .	32
4.2.2	Deep learning with stacked sparse autoencoders . . . . .	35
<b>5</b>	<b>Dataset Formulation</b>	<b>39</b>
5.1	Key Idea . . . . .	39
5.2	MODIS Data - Obtaining Features . . . . .	39
5.3	CORINE Land Cover Data - Obtaining Labels . . . . .	42
5.4	Dataset Properties . . . . .	45
<b>6</b>	<b>Experimental Results</b>	<b>47</b>
6.1	Multi-label Learning Evaluation . . . . .	47
6.1.1	Experimental settings . . . . .	48
6.1.2	Classification performance with respect to training set . . . . .	48
6.1.3	Introducing the “multi-label confidence maps” . . . . .	50
6.1.4	Comparison of multi-label classification algorithms . . . . .	55
6.1.5	Classification per label . . . . .	57
6.1.6	Classification on different spatial regions and temporal instances	58
6.1.7	Comparison with spectral unmixing . . . . .	60
6.1.8	Experimenting with fusion of the features . . . . .	64
6.1.9	Parameter sensitivity analysis . . . . .	66
6.2	Feature Learning Evaluation . . . . .	67
6.2.1	Data preprocessing . . . . .	69

<b>Contents</b>	<b>xi</b>
6.2.2 Network architecture . . . . .	69
6.2.3 Performance of raw and high quality features . . . . .	70
6.2.4 Impact of layer size and normalization . . . . .	72
6.2.5 Impact of depth . . . . .	74
6.2.6 Visualization of results . . . . .	76
6.2.7 Comparison with PCA . . . . .	76
6.2.8 Model sensitivity . . . . .	77
<b>7 Conclusions and Future Work</b>	<b>82</b>
<b>Appendices</b>	<b>84</b>
<b>A Working with MODIS Reprojection Tool</b>	<b>84</b>
<b>B Processing Geographic Information Systems Data</b>	<b>88</b>
<b>References</b>	<b>92</b>





# List of Figures

3.1	Multi-label classification applications. . . . .	16
3.2	Visual illustration of the multi-label classification process with remotely sensed data. A multi-label training set is generated by annotating multispectral satellite imagery with ground-sampled labels at higher spatial resolutions. Up-to-date land cover predictions are made through the use of multi-label classifiers that produce “multi-label confidence maps” encoding the presence of specific types of land cover. . . . .	18
4.1	Neurons serve as the elementary building blocks of human brain information processing, similar to units of the artificial neural networks.	30
4.2	Visual illustration of the representation learning process with remotely sensed data. An initial feature-mapping with SAEs is performed in order to train a multi-label classifier, whereas we proceed to the testing procedure by exploiting the learned feature-mapping. .	32
4.3	Architecture of an autoencoder with an overcomplete hidden layer. The encoder takes the input $\mathbf{x}$ and computes a prediction of the best value of the latent code $\mathbf{h}$ . The decoder is symmetric to the encoder and computes a reconstruction $\hat{\mathbf{x}}$ from $\mathbf{h}$ . The bias units are not considered for simplicity. . . . .	33

4.4	A 4 layer autoencoder network $[3-4-4-2]$ , where the circles denote the feature units. The black and grey colors are used to denote the inactive and active hidden units, respectively, whereas the white the visible units. The two middle layers constitute an encoder. The bias units are not considered for simplicity. . . . .	36
5.1	Geographic distribution of MODIS h18v04 and h19v04 tiles. The h18v04 region captures South-Central Europe, while h19v04 a large part of the Balkans. These exemplary regions were selected due to the diversity of land cover and the availability of data. . . . .	40
5.2	CLC map for the h19v04 tile of 2000 (CLC2000). . . . .	43
6.1	Classification performance with respect to the amount of training data (out of the 8604) corresponding to a single spatial tile. These experiments illustrate the complex interaction behavior between training data and performance. In general, one can observe that the performance gains are significantly more dramatic when increasing smaller sets of training examples, while the benefits of introducing more training data are moderate once sufficiently data is available. . .	49
6.2	Ground-truth multi-label map for h19v04 of CLC2000 corresponding to a binary matrix indicating which labels are active for each example, <i>i.e.</i> , spatial location. The vertical axis corresponds to specific label as illustrated in Table 5.1, while horizontal axis to specific testing example out of the 3687. . . . .	51
6.3	Multi-label confidence maps for h19v04 of CLC2000 with 128 training samples. The red boxes outline areas where there is significant deviation between the predicted and the ground-truth labels. They highlight that some labels in classification are more sensitive than the others. . . . .	52

6.4	Multi-label confidence maps for h19v04 of CLC2000 with 1024 training samples. Similar to before the red boxes outline areas where there is significant deviation between the predicted and the ground-truth labels. Comparing to the case of 128 training examples shown above, we observe less errors according to the ground-truth map in Fig. 6.2.	52
6.5	Ground-truth multi-label map for h18v04 of CLC2000. The vertical axis corresponds to specific label as illustrating in Table 5.1, while the horizontal axis to specific testing example out of the 7624. . . . .	53
6.6	Multi-label confidence maps for h18v04 of CLC2000 with 128 training samples. . . . .	53
6.7	Multi-label confidence maps for h18v04 of CLC2000 with 1024 training samples. Comparing to the case of 128 training examples shown above, we observe less errors according to the ground-truth map in Fig. 6.5. . . . .	53
6.8	Ground-truth multi-label map for h19v04 of CLC2006. The vertical axis corresponds to specific label as illustrating in Table 5.1, while the horizontal axis to specific testing example out of the 2745. . . . .	54
6.9	Multi-label confidence maps for h19v04 of CLC2006 with 128 training samples. . . . .	54
6.10	Multi-label confidence maps for h19v04 of CLC2006 with 1024 training samples. Comparing to the case of 128 training examples shown above, we observe less errors according to the ground-truth map in Fig. 6.8. . . . .	54

6.11	Multi-label classification performance with respect to amount of training data for tiles originating from different spatial locations using ECC-DT. Naturally, the results verify that training using data from the same spatial location produce the optimal classifier, while changing the location can have dramatic effects on performance. Fortunately, exploiting a mixed training set composed of data from both the corresponding data and location, as well as different location can achieve a very good performance, approaching the “optimal” performance. . . . .	59
6.12	Multi-label classification performance with respect to amount of training data for tiles originating from same spatial locations but different time instances using ECC-DT. Similar to Fig. 6.11 the results are not good when using the training set of the reference tile, whereas by incorporating the mixed training set we achieve a very good performance, approaching the “optimal” one. . . . .	60
6.13	RMSE w.r.t. the number of examples for h19v04 of CLC2000 over 30 random runs. The approximation for all the examined unmixing chains improves, suggesting these data can be also used for unmixing purposes. . . . .	62
6.14	Performance with respect to the number of neighbors for h19v04 of CLC2000 with 1024 training examples. Both adaptation algorithms exhibit similar performance with respect to this parameter, however IBLR has a slightly higher and more robust behavior compared to ML-kNN. . . . .	66
6.15	Classification performance with respect to the number of models for h19v04 of CLC2000 with 1024 training examples. Varying the number of models has a major effect on the classification performance for ensemble methods. In this respect, ECC-DT exhibits a superiority compared to RAKEL-DT for limited number of training examples. . .	67

- 6.16 Performance of multi-label classifiers using the codes learned by increasing the size  $k$  of one hidden layer. The solid horizontal lines correspond to the performance of the classifiers with the initial raw features, whereas the dashed lines correspond to the accuracy achieved with the normalized version of the raw features. Optimal complexity-performance ration is achieved using twice as many of the initial raw features. . . . . 73
- 6.17 Effect of depth on accuracy for models trained with and without unsupervised pretraining using RAkEL-SVM (left) and ECC-SVM (right) classifiers, for 1 to 2 hidden layers in which the hidden layer size has been fixed to 320. Box plots show the distribution of errors associated with 50 different initialization seeds. A box represents 50% of the data, the red central line indicates the median value, whereas the lower and upper boundary lines are the 25th and 75th percentiles. Whiskers extend to the remaining data that are not regarded as abnormal outliers, which are shown individually as red “+”s. . . . . 79
- 6.18 Ground-truth multi-label map for h19v04 of CLC2000 corresponding to a binary matrix indicating which labels are active for each example, *i.e.*, spatial location. The vertical axis corresponds to specific label as illustrating in Table 5.1, while horizontal axis to specific testing example out of the 3000. . . . . 80
- 6.19 Multi-label confidence maps with RAkEL-SVM as top layer classifier incorporating different levels of features. . . . . 80
- 6.20 Multi-label confidence maps with ECC-SVM as top layer classifier incorporating different levels of features. . . . . 80
- 6.21 Comparison of dimensionality reduction methods with PCA and SAEs for multi-label data using ECC-SVM. SAEs perform higher but have also higher computational complexity. . . . . 81

---

6.22	Sensitivity of the SAE model for the single-layer case. Sparsity parameter $\rho$ plays an important role to the final performance for a fixed sparsity weight (left), whereas the value of the cost function reduces primarily due to the size of the hidden layer (right). . . . .	81
A.1	Snapshots of the MRT tool. . . . .	85
A.2	Visual difference between the SIN and the UTM projection for a specific spatial tile. . . . .	86
B.1	The raster and vector data models. Each model stores features in a different way. . . . .	89
B.2	Snapshots of the QGIS. . . . .	90

# List of Tables

5.1	The 20 selected ground-truth labels from CORINE with their original CLC code. The labels represent a wide range of materials that may be contained, from those existing in urban and vegetation areas to those in minerals and beaches. The number of examples in the last column refers to the CLC2000. . . . .	44
5.2	Statistics for multi-label dataset based on the h19v04 tile of CLC2000. . . . .	46
6.1	Performance (mean $\pm$ std) of each multi-label learning algorithm over 10 different 10-fold cross validation experiments. For each metric, $\uparrow$ indicates “higher the better”, whereas $\downarrow$ indicates “lower the better”. Ensemble methods perform overall higher than problem transformation and algorithm adaptation techniques. . . . .	56
6.2	The mean accuracy per label over 10 different 10-fold cross validation experiments. Some labels are more sensitive than others in classification. . . . .	57
6.3	Performance (mean $\pm$ std) of the ensemble methods versus unmixing over 30 random runs. Multi-label classification produces significantly higher results. . . . .	64
6.4	Performance (mean $\pm$ std) of the ensemble methods using feature-level fusion and decision-level fusion with Max or Min Rule over 50 random runs, with 1024 training examples. . . . .	65
6.5	Impact of the quality of features for the classifiers. Higher quality features composed of NDVI and LST yield to improved performance compared to raw surface reflectance. . . . .	71

---

6.6	Impact of depth for a fixed architecture consisting of 320 hidden units per layer. Higher results are obtained for features extracted from deep architectures. . . . .	74
-----	--	----



# List of Acronyms

<b>ANC</b>	Abundance Non-negativity Constraint
<b>ANN</b>	Artificial Neural Networks
<b>ASC</b>	Abundance Sum-to-one Constraint
<b>BFGS</b>	Broyden-Fletcher-Goldfarb-Shanno
<b>BR</b>	Binary Relevance
<b>CC</b>	Classifier Chains
<b>CLC</b>	CORINE Land Cover
<b>CLS</b>	Constrained Least Squares
<b>CORINE</b>	Coordination of Information on the Environment
<b>CRS</b>	Coordinate Reference System
<b>DL</b>	Distinct Labelsets
<b>DT</b>	Decision Trees
<b>ECC</b>	Ensemble of Classifier Chains
<b>EEA</b>	European Environment Agency
<b>EOS</b>	Earth Observation System
<b>EPSG</b>	European Petroleum Survey Group
<b>ETRS</b>	European Terrestrial Reference System

---

<b>EVI</b>	Enhanced Vegetation Index
<b>FCLS</b>	Fully Constrained Least Squares
<b>GIS</b>	Geographic Information System
<b>GLC</b>	Global Land Cover
<b>GPS</b>	Global Positioning System
<b>HDF</b>	Hierarchical Data Format
<b>IBLR</b>	Instance-Based Logistic Regression
<b>KL</b>	Kullback-Leibler
<b>kNN</b>	k-Nearest Neighbors
<b>LMM</b>	Linear Mixing Model
<b>LP</b>	Label Powerset
<b>LST</b>	Land Surface Temperature
<b>ML-kNN</b>	Multi-Label k-Nearest Neighbors
<b>MODIS</b>	MODerate-resolution Imaging Spectroradiometer
<b>MRT</b>	MODIS Reprojection Tool
<b>NDVI</b>	Normalized Difference Vegetation Index
<b>NIR</b>	Near InfraRed
<b>PCA</b>	Principal Component Analysis
<b>PPNMM</b>	Polynomial Post-Nonlinear Mixing Model
<b>RAkEL</b>	RANdom k-LabELsets
<b>SAE</b>	Sparse AutoEncoder
<b>SIN</b>	Sinusoidal

---

<b>SISAL</b>	Simplex Identification via Split Augmented Lagrangian
<b>SUnSAL</b>	Sparse Unmixing by variable Splitting and Augmented Lagrangian
<b>SVM</b>	Support Vector Machines
<b>TIR</b>	Thermal InfraRed
<b>UTM</b>	Universal Transverse Mercator
<b>VCA</b>	Vertex Component Analysis
<b>WGS 84</b>	World Geodetic System 1984



# Chapter 1

## Introduction

### 1.1 Background

Land cover analysis refers to the monitoring of the geophysical and biophysical characteristics of the Earth's surface, a process critical in global environmental sciences studying the ever-changing continental-scale evolution of our planet. Though humans have been modifying land to obtain food and other essentials for thousands of years, current rates, extents and intensities of land cover change are far greater than ever in history, driving unprecedented changes in ecosystems and environmental processes at local, regional, and global scale [1]. These changes encompass the greatest environmental concerns of human populations today, including climate change and pollution of water, soil, and air. Information about land cover is of paramount importance for effective conservation planning and management of biological diversity, since it is used to identify those geographic areas with sufficient quality to support wildlife and biodiversity becoming extinct.

These vital needs mandate an increased effort in creating accurate and timely high spatial resolution land cover maps. Despite the urgency, such endeavors are hindered by various constraints, the most prominent of which is the labor intensive hand-operated process of collecting ground-based data from field surveys. To that end, remote sensing systems represent a major resource for monitoring global-scale variations in land cover [2]. High resolution sensors retrieving optical, synthetic aperture radar, multispectral, and hyperspectral data, are being employed to achieve

this demanding objective. Recently, the use of advanced multispectral and hyperspectral imaging instruments has emerged as a particularly effective approach for distinguishing among physical phenomena by observing differences in their spectral signatures [3, 4].

However, spectral signatures gathered by even such high resolution systems, cannot always capture all the objects on the Earth’s surface, principally due to the fact that surrounding objects are not sufficiently spectrally different (mixing in the signals). Furthermore, machine learning algorithms which are tasked with the automated classification of remotely sensed data are heavily dependent on the choice of data representation (features) on which they are applied [5]. To address this issue researchers have always tried to increase the input quality by incorporating specialized features that ease the burden of learning algorithms. This strategy is particularly evident in computer vision tasks, where carefully designed hand-crafted features, such as Scale Invariant Feature Transform (SIFT) [6] or Histogram of Oriented Gradients (HOG) [7], have shown great effectiveness in a variety of tasks. Analogous features already exist in remote sensing, for example the Normalized Difference Vegetation Index (NDVI) [8], the Enhanced Vegetation Index (EVI) [9], or the Normalized Difference Water Index (NDWI) [10], aiming to provide a stronger indicator of the amount of the photosynthetically active green biomass, of the leaf water content, and of the soil humidity than the pure spectral signatures [11].

## 1.2 Thesis Motivations

During the mapping procedure with remotely sensed data, a classification technique has to be applied in order to annotate the acquired pixels with additional metadata. In typical satellite image classification [12, 13], especially in situations where multiple spectral bands are acquired, each pixel is associated with a single class from a set of two or more classes [14, 15]. Furthermore, in order to sustain a high classification performance under specific conditions, the design of high quality hand-crafted features has been for years the only way, even though such features are characterized by limited generalization ability and require significant human intervention. These

limitations motivate the need for efficient feature representations extracted automatically from data through representation learning [5], a set of techniques which intends to learn useful (*i.e.*, discriminative, robust, smooth) representations of the raw data for the purpose of higher level tasks (*e.g.*, classification, recognition) and minimize the dependency of learning algorithms on feature engineering. We note that learning such features in a domain where the underlying data are subject to many factors of variation is a challenging task [16]. In remote sensing exist many such factors including the ground environmental conditions and cloud contamination, forming a domain full of challenges.

In any case, the conventional way of classification to a single class is guided by an assumption that is often violated in real-life scenarios, because pixels of aerial or space images are simultaneously characterized by multiple classes due to the presence of different materials. Consequently, single class assignment can be unrealistic and leads to ambiguities in the maps. State-of-the-art methods try to uncover the different materials contained in a spectral pixel along with their corresponding proportion values by a process known as spectral unmixing [17]. Despite the significance of spectral unmixing, the majority of approaches that have been proposed rely on extremely limited and outdated hand-labeled datasets, such as the Cuprite mining district data [18]. A consequence of the lack of real data is that typically one artificially applies a theorized forward mixing process and tests the capabilities of the proposed algorithm on performing the inverse process [19]. The utilization of simulated/synthetic data can provide some intuition regarding the capabilities of each approach, however, it is very difficult to generalize the behavior of these algorithms when they are applied under more realistic conditions. One should also keep in mind that performance conclusions based on laboratory-crafted datasets cannot be regarded entirely reliable, since acquisition under a controlled environment inflicts a smaller number of sources of variation than when dealing with real data where isolated and unpredictable situations frequently happen [20].

Nevertheless, nowadays, there is a plethora of large unlabeled remote sensing datasets which remain unexploited, an issue which has raised a lot of research interest. One main obstacle in the utilization for such data is the problem of scale in-

compatibility. Whereas field-based measurements can be conducted at meter scales, distance to the ground and speed of the moving platforms are directly responsible for the considerably lower spatial resolution of remote sensing imagery. This spatial scale incompatibility between ground-based and satellite-based sampling inevitably hinders the exploitation of the acquired measurements. For instance, while Coordination of Information on the Environment (CORINE) land data, collected and distributed by the European Environment Agency (EEA), are available at 100m<sup>2</sup> resolution, the MODerate-resolution Imaging Spectroradiometer (MODIS) multi-spectral satellite instrument by National Aeronautics and Space Administration (NASA) provides usually products at a resolution of 500m<sup>2</sup> per pixel. Similarly, while Global Land Cover (GLC) data provided by the National Geomatics Center of China (NGCC), are available at 30m<sup>2</sup> spatial resolution (GlobeLand30), the PROBA-V mission by the European Space Agency (ESA) supplies 300m<sup>2</sup> multi-spectral images. Despite their lower resolution, airborne and spaceborne platforms can provide imagery at significantly higher temporal sampling rates, as more and more of these platforms are in continuous flights and orbits around the Earth.

### 1.3 Thesis Contributions

In this thesis, we propose a radically novel approach for inferring the complex relationships between multispectral satellite imagery and spectral profiles of different surface materials, exploiting the proliferation of remote sensing imagery, through the introduction of the *multi-label classification* [21, 22], a powerful framework in machine learning. Departing from traditional single-label classification, in multi-label learning each sample is associated with multiple labels simultaneously, while in many cases the labels are also ranked according to their relevance to a given sample, a premise that seems appealing for application in remote sensing. This framework has attracted considerable attention from the data mining community over the last decade due to its numerous contemporary and real-world applications [23]. Traditionally, it is applied in text [24], audio [25], and image classification [26], where a document could belong to several topics, a music song could fit to different gen-



res, and an image could be annotated by many tags, respectively. Inspired by the previous examples, one can cast the prediction of multiple characteristics which are not mutually exclusive in a specified spatial area, as an appealing multi-label classification problem.

In the proposed schema, we jointly model ground-based land cover data and multispectral satellite imagery of different spatial resolutions. This way, we provide a genuine answer to the scale incompatibility problem which arises through the sampling procedures. More specifically, we combine data from the CORINE Land Cover (CLC) maps of 2000 and 2006 compiled by EEA [27] coming at 100m<sup>2</sup> spatial resolution and corresponding to the European environmental landscape annotated by experts, with satellite data products from the MODIS database [28] at 500m<sup>2</sup> spatial resolution. Due to this difference in scale, each multispectral pixel may be associated with multiple labels simultaneously, naturally leading to the case of multi-label annotation.

To the best of our knowledge, this is the first work which applies a multi-label classification scheme in remote sensing data, an approach that can effectively address the issues arising inevitably due to the multiple scales of the data, without requiring the explicit and often unrealistic modeling of the generative single-label processes. A key benefit of our method is accurate and up-to-date high resolution land cover maps, obtained through a new supervised dataset composed of freely available and real data which can leverage the abundance of satellite imagery. The complete dataset will be available online.

We claim that multi-label learning can provide valuable information on remotely sensed data, especially in the case of land cover estimation, where the heterogeneity of different regions suggests an overwhelming amount of mixed pixels. Under the multi-label learning logic, one does not have to assume a priori where the data should lie, in contrast to unmixing algorithms which largely rely on the expected type of mixing [29]. Moreover, multi-label classification provides a wide range of performance evaluation measures which can provide a detailed picture of the whole procedure under different viewpoints, whereas unmixing, being in principle an unsupervised procedure, has not well established arithmetic assessment expressions.

Finally, the multi-label community has highly recognised that exploiting the dependencies between related classes and samples during the classification process is critical in order to improve prediction performance [30, 31], and thus most of the state-of-the-art algorithms take into account such correlations, in contrast to unmixing algorithms. All in all, multi-label classification can be thought as an alternative more realistic model than single-label classification for remotely sensed data, while in parallel it can provide supplementary solutions to the tasks of conventional unmixing.

A second focus of this thesis, is to find “good representations” for satellite data under the aforementioned multi-label learning scenario, since real-world sensor measurements are complex and highly variable leading to limited performance. Instead of relying on specialized features (*e.g.*, NDVI), we propose the application of a particularly successful unsupervised *representation learning* approach to automatically extract meaningful features identifying the underlying explanatory patterns hidden in low level satellite data. This way, we consider the framework of Sparse AutoEncoder (SAE) [32, 33], a type of artificial neural network which employs nonlinear codes and imposes sparsity constraints for representing the original data. The proposed scheme utilizes a series of sparse autoencoders stacked in a greedy layer-wise fashion, in order to train a deep learning model in the context of multi-label satellite image classification.

## 1.4 Thesis Outline

The rest of this thesis is structured as follows:

- Chapter 2 provides an overview of the related state-of-the-art from the literature. We firstly present some benchmark remote sensing sensors used for mapping and classification, highlighting that their design parameters can determine the overall quality of classification performance. Then, we present some frequently employed algorithms for single-label classification in remote sensing, followed by the definition of spectral unmixing problem. Finally, we present feature learning techniques, especially in terms of autoencoders with

different regularization penalty terms, as well as the deep learning version.

- Chapter 3 presents the multi-label classification framework and explains why it can address the challenges of remotely sensed data analysis better than conventional single-label classification. We also discuss the algorithms considered in this thesis presented in discrete categories, along with their internal mechanisms and differentiations. Finally, we explain the evaluation measures we use to evaluate the effectiveness of the algorithms, and thus the merits of our approach.
- Chapter 4 describes the basic theory of feature learning with autoencoders. Then, we present the single-layer sparse autoencoder variant along with its background intuition. Finally, we discuss about deep learning and how this framework can be applied to construct the stacked sparse autoencoders.
- Chapter 5 analyzes the datasets that are employed in this thesis. We explain thoroughly the steps followed for their formulation, along with indicative statistics that reveal their challenges.
- Chapter 6 demonstrates and discusses an extensive set of experimental results. In the first part, we consider the evaluation of the proposed multi-label learning approach on remote sensing data. To that end, we perform some interesting experiments for predicting land cover coverage in different spatial regions or temporal instances with respect to the number of training examples, as well as a comparison with spectral unmixing. The second part affords representation learning experiments, where the best results arise by utilizing the deep learning architectures.
- Chapter 7 provides a synthesis of the contributions, makes concluding remarks, and discusses future work by exposing possible extensions of this thesis.



# Chapter 2

## Related Work

### 2.1 Remote Sensing Mapping and Classification

Since the 90's, satellite data have been extensively used for land cover mapping and classification. Land cover datasets provide a critical input for global models and hence they have gained significant attention. The most established land cover datasets include the GLC2000 [34], the GlobCover [35], and the MODIS land cover product [36] which provide a global mapping, whereas the CORINE project [37] encompasses data for the European continent. Each set is prepared using different data sources, classification algorithms, methodologies or even spatial resolution, leading in many cases to areas of uncertainty (spatial disagreement) [38]. All of the above datasets have been investigated with a plethora of typical [39] as well as more sophisticated classification methods [40]. Notable among them, Support Vector Machines (SVM) demonstrate very good performance for the classification of airborne and satellite imagery with limited training examples, especially by incorporating composite kernels [41].

Apart from learning algorithms, modeling the sensors that are employed in the process is also of crucial importance for the quality of the features and thus the construction of the land cover maps and classification. There are two main categories of optical remote sensing systems: *multispectral* imaging devices which typically acquire 5 to 20 spectral bands, and *hyperspectral* imaging devices which can acquire hundreds of spectral bands. Nevertheless, it has to be underlined, that except for

spectral information, spatial, temporal as well as radiometric resolution properties of the sensor determine dominantly the success of classification [42, 43].

One of the first sensors that provided satellite data at a large scale, was the Advanced Very High Resolution Radiometer (AVHRR) on the National Oceanic and Atmospheric Administration (NOAA) platforms<sup>1</sup>, which triggered many studies on land cover discrimination [44]. Some more recent and broadly used medium resolution remote sensing systems include PROBA-V on SPOT<sup>2</sup>, Thematic Mapper (TM) or Enhanced Thematic Mapper Plus (ETM+) on Landsat<sup>3</sup>, and MODIS aboard the Terra and the Aqua satellites<sup>4</sup>. Note that a higher spatial resolution generally implies smaller coverage area of the system [43]. In order to compensate for the coarse resolution provided by these multispectral instruments, the use of time evolution of surface reflectance (time series) has proven to be valuable and thus is adopted in most relevant studies [45].

On the opposite side, the most explored hyperspectral remote sensing scenes which are appropriate for supervised classification (containing ground-truth tables) were gathered by the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) [46] (*e.g.*, Indian Pines, Salinas Valley, Kennedy Space Center) and the Reflection Optics System Imaging Spectrometer (ROSIS) [47] (*e.g.*, Pavia Center, Pavia University) airborne sensors, which generate 224 and 115 contiguous spectral bands, respectively. Due to the increased number of bands, these sensors achieve a finer spectral resolution and hence dimensionality reduction [48] and sparsity techniques [49] have been extensively applied.

## 2.2 Spectral Unmixing

Under normal operating conditions, in remote sensing imaging systems each pixel (spectral vector) captures and encodes a multitude of signals, a phenomenon attributed to the physical properties of light, its interaction with matter and atmo-

---

<sup>1</sup><http://noaasis.noaa.gov/NOAASIS/ml/avhrr.html>

<sup>2</sup><http://proba-v.vgt.vito.be/>

<sup>3</sup><http://landsat.usgs.gov/>

<sup>4</sup><http://modis.gsfc.nasa.gov/>

sphere, and the technical characteristics associated with the acquisition process [50]. More precisely, on one hand nonlinear mixing of signals occurs when the light scattered by multiple materials in the scene is reflected of additional objects, as well as when two materials are homogeneously mixed [29]. On the other hand, even in the ideal case where the incident light interacts with a single material, linear mixing occurs due to the instrumentation (primarily low spatial resolution of the sensor) and various sources of noise.

Given the mixing of signals, there is a crucial need for a process that can separate the pixel spectra into a collection of pure materials, called endmembers. Spectral unmixing [17] points towards this problem aiming to calculate the number of endmembers (optionally), distinguishing their spectral signatures, and estimating their fractional abundances (proportion of each endmember's presence) in the pixels [29]. Unmixing algorithms work better assuming a specific type of mixing, with the major effort in the last decade to be focused on the Linear Mixing Model (LMM), since, despite its simplicity, is an acceptable approximation of the behavior of the light [29]. Under this model, a mixed pixel is a linear combination of the endmembers signatures weighted by the corresponding fractional abundance, and thus one can assume that such a pixel lies inside the convex hull of its endmembers. Due to physical considerations in the data acquisition process, the unknown fractional abundance vector for a given pixel is subject to the Abundance Non-negativity Constraint (ANC), meaning that the estimated vector cannot be negative, and the Abundance Sum-to-one Constraint (ASC), meaning that the sum of the abundance values must sum up to one.

In order to decompose a mixed pixel spectra, different kinds of endmember extraction algorithms have been developed, essentially divided into two approaches, namely the *geometrical*, and the *statistical*. The geometrical one, exploits the fact that mixed pixels lie inside a simplex (convex geometry shape) and is further divided into two subcategories: the *pure pixel based*, which assume that there is at least one pure pixel per endmember in the training data (*i.e.*, at least one spectral vector on each vertex of the simplex set), and the *minimum volume based*, which have not such a prerequisite and seek to minimize the volume of the simplex (nonconvex

optimization problem). Two well-known examples of the former approach include the N-FINDR [51] algorithm, which seeks for the simplex with the largest volume, and the Vertex Component Analysis (VCA) [52], which consists a robust technique to the presence of weak nonlinearities targeting to find extreme points in the data cloud. A representative paradigm from the latter approach is the Simplex Identification via Split Augmented Lagrangian (SISAL) [53], which solves a sequence of nonsmooth convex subproblems using variable splitting to obtain a constraint formulation, and then applies an augmented Lagrangian technique in order to unmix hyperspectral data in which the pure pixel assumption is violated. Statistical methods model the abundance fractions as random variables and formulate the spectral unmixing as a statistical inference problem, providing a natural framework for representing variability in the endmembers. They include the Independent Component Analysis (ICA) [54], a well-known tool in blind source separation problems, which is yet strongly criticized due to the fact that the abundance fractions associated to each pixel are not statistically independent [55], as well as Bayesian approaches [56], which have the ability to model statistical variability by imposing priors that can constrain solutions to physically meaningful ranges. The statistical methods perform better in highly mixed images than the geometrical approaches (because there are not enough spectral vectors in the simplex facets), however, they have a much higher computational complexity.

The abundance estimation part comprises the last step of the unmixing process. It can be solved via classical convex optimization methods, such as the Constrained Least Squares (CLS), which in this context minimizes the total squared error under the ANC, as well as the Fully Constrained Least Squares (FCLS), which adds the ASC to the CLS problem. Meanwhile, sparse regression approaches have become popular, such as the the Sparse Unmixing by variable Splitting and Augmented Lagrangian (SUnSAL) [57], where sparse linear mixtures of spectra are investigated in a fashion similar to that of compressed sensing [58]. Recently, effort has been also given to study nonlinear mixing models in order to handle specific kinds of nonlinearity, such as the Polynomial Post-Nonlinear Mixing Model (PPNMM) and its associated unmixing algorithm based on the subgradient method proposed in [59].



## 2.3 Unsupervised Representation Learning

In general, representation learning encompasses a variety of methods, most of them based on neural networks that combine linear and nonlinear transformations of the data. This way, autoencoders (or autoassociators) were adopted with impressive success as feature learning architectures, although they were initially studied in the late 80's as a technique for dimensionality reduction by considering a hidden layer with fewer units compared to the input (forming a bottleneck). More recently, extending their initial use, overcomplete basis vectors have been employed to obtain more expressive representations, where the number of features exceeds the number of raw inputs. In this setting, a form of regularization during autoencoder learning is needed in order to avoid trivial solutions where the autoencoder could reconstruct the input perfectly, without needing to extract any meaningful features. Several autoencoder variants have been developed in order to introduce regularization in the latent space, including the denoising [60], the contractive [61], the saturating [62], and the SAE [32,33].

Apart from modifying the regularization penalty term, effort has also been given on the investigation of the impact of other choices on system performance, especially in terms of the network architecture. For instance, recursive networks [63] apply the same set of weights recursively over a structure (directed acyclic graphs), recurrent networks [64] where connections between units form a directed cycle, convolutional networks [65] with whitening transformation and pooling operations for visual tasks [66], and neural networks with rectified hidden units [67].

While it has been shown that one hidden layer can approximate a function to a very high level of precision, this approach becomes impractical due to the large number of the required computational units [68]. Inspired by the human cognitive system, researchers have tried to incorporate depth into learning algorithms, which would allow to achieve function representation more compactly [69], and obtain increasingly more abstract representations. Although theoretical results have been encouraging, in practice, it has been impossible to train sufficiently deep architectures, since gradient-based optimization methods starting from random initial weights tended to get fixated near poor local optima [70].

Deep learning was revolutionized in the past decade, when the strategy of greedy layer-wise unsupervised “pretraining” followed by supervised fine-tuning was introduced [32, 71]. This technique was first applied using Restricted Boltzmann Machines (RBMs) for a digit recognition task, but has proved to be an efficient approach by incorporating autoencoders in various contexts too. Nevertheless, one should keep in mind that deep architectures do not guarantee a superiority over shallow architectures for every type of problem [72], although the behavior in specific settings is under extensive investigation. We should note that the ideas underlying deep learning have been motivated by the way the human brain seems not only to be organized, but also to process received stimuli, which is accomplished through a chain of multiple transformation stages [68]. For example, it has been experimentally shown that for the object recognition tasks, representations produced by deep architectures can resemble those features observed in the first two stages of the visual cortex, *i.e.*, edges and shapes detected by the receptive fields of neurons in V1 and V2 areas.

# Chapter 3

## Multi-label Learning

### 3.1 Principal Concept

In traditional single-class classification and multi-class problems, each instance is relevant to only one class from a set of two or more classes/labels, respectively. However, in many real-world problems, one instance may belong to more than one labels simultaneously. For example, in image classification of Fig. 3.1, a piece of the image is associated with both the label “sea”, and the label “beach”. One distinguishing difference between multi-label learning and binary or multi-class learning, is that the labels in multi-label problems are not mutually exclusive, leading each instance to belong (optionally) to multiple labels. Motivated by the increasing number of contemporary applications, multi-label learning has recently attracted significant attention in the literature [22].

Formally, learning from multi-label examples corresponds to finding a mapping from the features space to the label sets space (*i.e.*, the power set of all labels). Increasing the number of labels makes the task of multi-label learning rather challenging due to the tremendous number of possible label sets. To cope with this issue, it is deemed that exploiting the correlations among different labels during training procedure is of paramount importance [30, 73]. In order to conceptually understand the significance of label dependencies, one can think of two images with a blue background depicting a ship and an airplane. Distinguishing these two images based solely on the color features is a difficult task for a classifier, since both contain

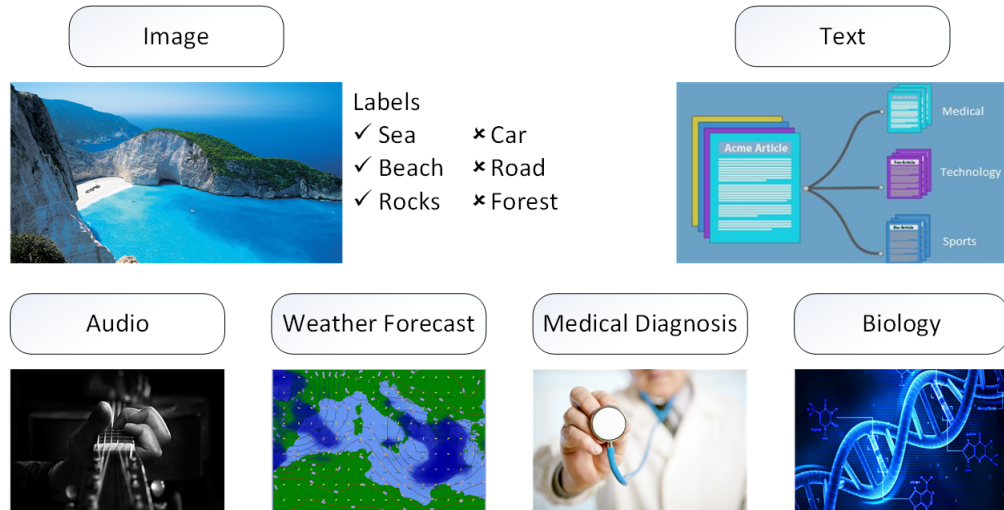


Figure 3.1: Multi-label classification applications.

large regions with blue color. However, if the system is confident enough that the image should be annotated with the “airplane” label, then it is more likely that the region of blue in the image should be annotated with “sky”, rather than “sea”.

## 3.2 Challenges in Multi-label Learning

In comparison with traditional binary and multi-class classification problems, the generality of multi-label classification inevitably causes difficulties. In the following paragraphs we list maybe the two most fundamental challenges which exist in the successful application of multi-label learning under real-world conditions [74].

The first issue concerns the effectiveness of multi-label classification for large-scale problems, meaning cases where the data dimensionality is high and the number of labels is large [75]. Similar to traditional classification techniques, multi-label learning also suffers from the curse of dimensionality, where data points become sparse and far apart from each other when the dimensionality is high. The problem here is even bigger, since features represent all the classes of the whole dataset, whereas many of them are not relevant to a specific class.

The other big challenge concerns the classification imbalance problem [76]. When each label is treated independently, it can be observed that most instances are irrelevant to a specific label. More specifically, since multi-label collections contain

a wide variety of classes, there is an unequal distribution of classes throughout the collection, whereas some of the classes are heavily populated and others contain only a few examples. In other words, not only some labels are much more frequently used than others (inter-class imbalance), but also multi-label data exhibit a strong imbalance between positive and negative examples of each label (inner-class imbalance). In fact, the classification imbalance problem becomes even worse when the number of labels increases. It is extremely difficult to build an accurate classifier for such rare labels employing limited numbers of training instances.

These issues result in two serious problems: high computation cost and low predictive power of the multi-label learning algorithms. Until today, it remains an open research question how to develop techniques that tackle successfully all the aforementioned challenges, while being in parallel computationally efficient [77, 78].

### 3.3 Problem Statement

Let  $\mathcal{S} \subseteq \mathbb{R}^d$  be an instance space of  $d$ -dimensional features, and a finite set of  $m$  labels or classes  $\mathcal{L} = \{\lambda_j \mid j = 1, \dots, m\}$ . Each instance  $\mathbf{x} \in \mathcal{S}$  has multiple class labels in  $Y$ , where  $Y \subseteq \mathcal{L}$ . Given a multi-label training set of  $n$  examples,  $D = \{(\mathbf{x}_i, Y_i) \mid i = 1, \dots, n\}$ , where  $\mathbf{x}_i \in \mathcal{S}$  and  $Y_i \in \mathcal{L}$  known, we assume that each instance is independent and identically distributed (i.i.d.) drawn from an unknown distribution. For each unseen instance  $\mathbf{x}$ , we define  $Z_{\mathbf{x}}$  as the predicted set of labels, and  $r_{\mathbf{x}}(\lambda)$  as the associated ordered listing (rank) of labels, since many machine learning algorithms induce decision rules in the form of a ranking function  $f(\cdot, \cdot)$  which strictly orders all the labels in  $\mathcal{L}$  according to their scores in this function. Therefore, for a given instance  $\mathbf{x}$ ,  $f(\mathbf{x}, \lambda)$  is interpreted as the system's confidence that  $\mathbf{x}$  belongs to label  $\lambda$ . It is said that label  $\lambda_1$  is ranked higher than label  $\lambda_2$ , if and only if  $f(\mathbf{x}_i, \lambda_1) > f(\mathbf{x}_i, \lambda_2)$ . All in all, the objective of multi-label classification algorithms is to estimate a set of decision rules  $\mathcal{H}$  that maximize the probability of  $\mathcal{H}(\mathbf{x}) = Z_{\mathbf{x}}$  for each example  $\mathbf{x}$ . We are based on this notation for the rest of this thesis.

The proposed framework strongly utilizes the merits of existing multi-label learn-

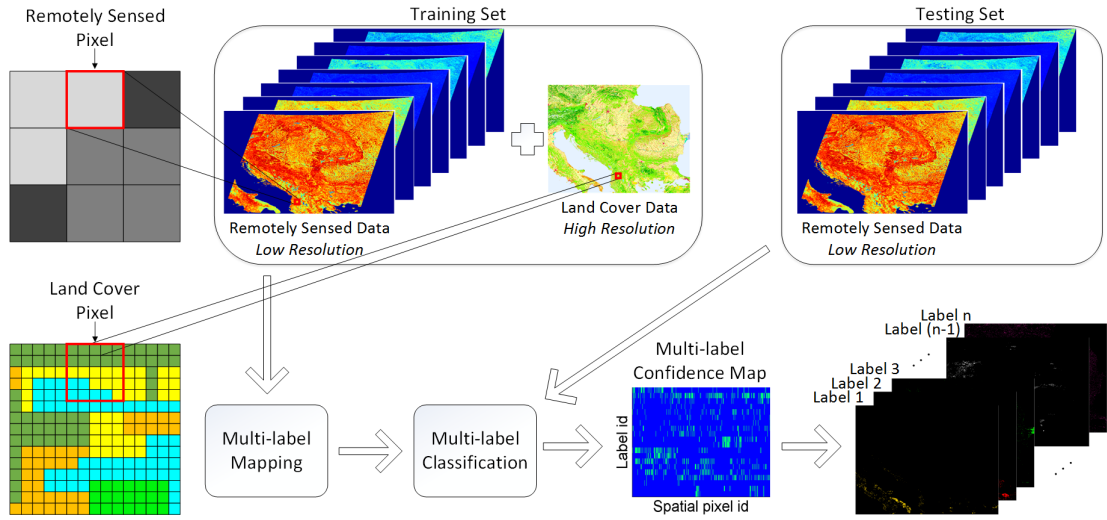


Figure 3.2: Visual illustration of the multi-label classification process with remotely sensed data. A multi-label training set is generated by annotating multispectral satellite imagery with ground-sampled labels at higher spatial resolutions. Up-to-date land cover predictions are made through the use of multi-label classifiers that produce “multi-label confidence maps” encoding the presence of specific types of land cover.

ing algorithms in order to realistically represent the relationships between acquired multispectral remotely sensed data and labels of the CLC programme. In this way, the output of our software system contains a matrix of the predicted labels (bipartitions), together with their ranking which is integrated in an informative visualization termed “*the multi-label confidence map*”. A high level overview of our proposed learning model is depicted in Fig. 3.2. In a nutshell, the key contributions of the proposed system architecture include:

- the formulation of an efficient approach for the combination of high spatial resolution land cover data with low spatial resolution satellite images.
- the development of an architecture capable of using up-to-date remote sensing data and produce land cover maps with minimal labor-intensive manual labeling.
- the systematic evaluation of state-of-the-art multi-label classification approaches on a novel and highly complex dataset.

- potential modalities for extending the scheme to various sources of data, in addition to land cover and multispectral examined in this work.

In the following section, we discuss the design and the theoretical analysis of the multi-label classification algorithms that are involved in this thesis as part of our system.

## 3.4 Classification Algorithms

Active research by the machine learning and the data mining communities has produced a large number of multi-label classification approaches. An extensive review can be found in [21] and in [22]. Existing approaches can be broadly divided into three categories: *problem transformation*, *algorithm adaptation*, and *ensemble methods* [79]. We have experimented with key representative examples from each category of multi-label classification methods.

The intuition underlying the first category, *i.e.* problem transformation methods, is to decompose the original multi-label learning problem into a set of smaller and easier-to-learn binary classification problems, in order to obtain a solution through well-established learning architectures. On the other hand, algorithm adaptation approaches adjust their mechanisms in order to directly tackle multi-label data by employing a type of problem transformation. Researchers usually select to modify algorithms which have been proven to be suitable for a specific domain. Representative techniques which have been adapted for the multi-label case include SVM [80], Boosting [24], Decision Trees (DT) [81], k-Nearest Neighbors (kNN) [82], and Artificial Neural Networks (ANN) [83]. Ensemble methods have appeared more recently and are deployed on top of problem transformation or algorithm adaptation methods as wrappers, improving their generalization ability [84]. Ensemble methods are regarded as the most powerful architectures which gather knowledge from all their components. According to this paradigm, multiple base learners are combined during the training phase to construct an ensemble, whereas a new instance is classified by integrating the outputs of single-label classifiers [85].

### 3.4.1 Problem transformation methods

Binary Relevance (BR) [75] is one of the earliest approaches in multi-label classification [21], where a single-label binary classifier is trained independently for each label, regardless of the rest of the labels (one-versus-all strategy). The method produces the union of the labels predicted by the binary classifiers, with the capability of ranking the labels based on the classifier output scores. In specific, in the BR approach, one trains a set of  $m$  classifiers such that:

$$\mathcal{H}_{BR} = \{h_j \mid h_j(\mathbf{x}) \rightarrow \lambda_j \in \{0, 1\}, j = 1, \dots, m\} . \quad (3.4.1)$$

BR is a straightforward approach for handling multi-label problems and is thus typically employed as a baseline method. The theoretical motivation and intuitive nature of BR are enhanced by additional attractive characteristics, such as moderate computational complexity (polynomial w.r.t. the number of labels), the ability to optimize several loss functions, and the potential of parallel execution [86]. The inherent drawback of the BR approach stems from the underlying assumption that no label correlations are considered during the training phase, which can lead to under or over estimation of the labels present in the testing data, or the identification of multiple labels that never co-occur [87].

Another fundamental yet much less extensively used transformation method is Label Powerset (LP) [21, 75]. Within this approach, each existing combination of labels in the training set is considered as a possible label for the newly transformed multi-class classification problem, *i.e.*, the number of different classes is upper bounded by  $f = \min(n, 2^m)$ , but in practice it is much smaller [88]. For the classification of a new instance, the single-label classifier of LP outputs the most probable class, which can be now translated to a set of labels:

$$\mathcal{H}_{LP} = \{h_j \mid h_j(\mathbf{x}) \rightarrow \lambda_j \in \{0, 1\}, j = 1, \dots, f\} . \quad (3.4.2)$$

In contrast to BR, LP methods can capture inter-relationships among labels, at the cost of significantly higher computational complexity, which scales exponentially with the number of labels. Therefore LP is challenged in domains with a large values of  $n$  and  $m$ . Furthermore, although this method is good at exact matches, it



is prone to overfitting since it can only model labelsets which have been previously observed in the training set [89].

One can see that this type of transformations are universally applicable, since any traditional single-label classifier (DT, SVM, Naive Bayes, etc.) can be employed in order to obtain multi-label predictions. The overall complexity of classification is heavily dependent on the underlying single-label classification algorithm and the number of distinct label collections. Due to these properties, problem transformation methods are very attractive in terms of both scalability and flexibility, while they remain competitive with state-of-the-art methods [87].

### 3.4.2 Algorithm adaptation methods

Multi-Label k-Nearest Neighbors (ML-kNN) [82] constitutes an adaptation of the kNN algorithm for multi-label data following a Bayesian approach. It is a lazy learning approach which is based on retrieving the  $k$  nearest neighbors in the training set and then counting the number of neighbors belonging to each class (*i.e.*, a random variable  $W$ ) [90]. Based on prior and posterior probabilities for the frequency of each label within these neighboring instances, it utilizes the Maximum A Posteriori (MAP) principle to determine the labelset for the unseen sample  $\mathbf{x}$ . The posterior probability of label  $\lambda_j$  is thus given by:

$$P(\lambda_j \in Z_{\mathbf{x}} | W = w) = \frac{P(W = w | \lambda_j \in Z_{\mathbf{x}}) P(\lambda_j \in Z_{\mathbf{x}})}{P(W = w)}. \quad (3.4.3)$$

Then, for each  $\lambda_j \in \mathcal{L}$ , ML-kNN builds a probabilistic classifier  $h_j(\cdot)$  applying the rule:

$$h_j(\mathbf{x}) = \begin{cases} 1 & P(\lambda_j \in Z_{\mathbf{x}} | W = w) > P(\lambda_j \notin Z_{\mathbf{x}} | W = w) \\ 0 & \text{otherwise} . \end{cases} \quad (3.4.4)$$

A classifier's output of 1 indicates that  $\lambda_j$  is active for  $\mathbf{x}$ , while 0 indicates the opposite. Despite the fact that ML-kNN inherits merits from both lazy learning and Bayesian reasoning (*e.g.*, adaptive decision boundary due to the varying neighbors identified for each test instance), it is ignorant of the possible correlations between

labels, thus it is essentially a BR method which learns a single classifier  $h_j(\cdot)$  for each label, independently from the others [22].

The Instance-Based Logistic Regression (IBLR) method [91], is also derived from the family of kNN inspired algorithms, but it integrates instance-based learning and logistic regression. The core idea in this case, is to consider the label information in the neighborhood of a query as “extra features” of that query, and then to treat instance-based learning as a logistic regression problem. For each label  $\lambda_j \in \mathcal{L}$ , the algorithm builds a logistic regression classifier  $h_j(\cdot)$  according to the model:

$$\log \left( \frac{\pi_{\mathbf{x}'}^{(j)}}{1 - \pi_{\mathbf{x}'}^{(j)}} \right) = \omega_{\mathbf{x}'}^{(j)} + \sum_{l=1}^m \alpha_l^{(j)} \cdot \omega_{+l}^{(j)}(\mathbf{x}), \quad (3.4.5)$$

where  $\pi_{\mathbf{x}'}^{(j)}$  denotes the (posterior) probability that  $\lambda_j$  is relevant for  $\mathbf{x}'$ ,  $\omega_{\mathbf{x}'}^{(j)}$  is a bias term,  $\alpha_l^{(j)}$  denotes a coefficient indicating to what extent the relevance of  $\lambda_j$  is influenced by the relevance of  $\lambda_l$ , and  $\omega_{+l}^{(j)}(\mathbf{x})$  is a summary of the presence of label  $\lambda_l$  in the neighborhood of  $\mathbf{x}'$ ,  $\mathcal{N}_k(\mathbf{x}')$ , defined by:

$$\omega_{+l}^{(j)}(\mathbf{x}') = \sum_{\mathbf{x} \in \mathcal{N}_k(\mathbf{x}')} h_l(\mathbf{x}) . \quad (3.4.6)$$

Here,  $h_l(\mathbf{x}) = 1$  if and only if  $\lambda_l$  is associated with  $\mathbf{x}$ , and 0 otherwise. The main advantage of IBLR over ML-kNN is that the former attempts to take into account label inter-dependencies arising by the estimation of regression coefficients.

### 3.4.3 Ensemble methods

Ensemble of Classifier Chains (ECC) [87] has established itself as a powerful learning technique with modest computational complexity. It is based on the successful Classifier Chains (CC) model [87], which involves the training of  $m$  binary classifiers, similar to BR methods. However, unlike the naive BR scheme, in CC, binary classifiers are linked along a “chain”, so that each classifier is build upon the preceding ones. In particular, during the training phase, CC enhances the feature space of each link in the chain with binary features from ground-truth labeling. Since true labels are not known during testing, CC augments the feature vector by all prior BR predictions. Formally, the classification process begins with  $h_1$  which determines  $P(\lambda_1 | \mathbf{x})$  and propagates along the chain for every following classifier  $h_2 \cdots h_j$

predicting:

$$P(\lambda_j \mid \mathbf{x}, \lambda_1, \dots, \lambda_{j-1}) \rightarrow \lambda_j \in \{0, 1\}, j = 2, \dots, m. \quad (3.4.7)$$

The binary feature vector  $(\lambda_1, \dots, \lambda_m)$  represents the predicted label set of  $\mathbf{x}$ ,  $Z_{\mathbf{x}}$ . Despite the incorporation of label information, the prediction accuracy is heavily dependent on the ordering of the labels, since only one direction of dependency between two labels is captured. To overcome this limitation, ECC extends this approach by constructing multiple CC classifiers with random permutations over the label space. Hence, each CC model is likely to be unique and able to give different multi-label predictions, while a good label order is not mandatory. In specific, to obtain the output of ECC, a generic voting scheme is applied, where the sum of the predictions is calculated per label and then a threshold  $t_s$  is applied to select the relevant labels, such that  $\lambda_j \geq t_s$ .

Another effective ensemble-based architecture for solving multi-label classification tasks is the RANdom k-LabELsets (RAkEL) [92], which embodies LP classifiers as base members. The RAkEL system tries to estimate correlations between the labels by training each LP classifier of the ensemble with a small randomly selected (without replacement) k-labelset (meaning size-k subset of the set of labels). This randomness is of primary importance in order to guarantee computational efficiency. For a classification of a new instance, each model provides binary predictions for each label  $\lambda_j$  in the corresponding k-labelset. Let  $E_j$  be the mean of these predictions for each label  $\lambda_j \in \mathcal{L}$ . Then, the output is positive for a given label, if the average decision is greater than a 0.5 threshold:

$$Z_{\mathbf{x}} = \{\lambda_j \mid E_j > 0.5, 1 \leq j \leq m\}. \quad (3.4.8)$$

In other words, when the actual number of votes exceeds half of the maximum number of votes that  $\lambda_j$  receives from the ensemble (majority voting rule), then it is regarded to be relevant. Although RAkEL models label correlations effectively and overcomes the aforementioned disadvantages of the LP transformation, the random selection of subsets is likely to negatively affect the ensemble's performance, since the chosen subsets may not cover all labels or inter-label dependencies [84].

## 3.5 Dimensionality Reduction

Dimensionality reduction techniques extract a small number of features by removing the irrelevant, redundant and noisy information to mitigate the effects of curse of dimensionality. In other words, they aim to transform high-dimensional data into a meaningful representation with a reduced dimensionality which preserves the important properties of the original data, while in parallel requires less storage space and less computational cost in the further processing.

The significance of dimensionality reduction has led the researchers to study the problem extensively in many areas. In general, dimensionality reduction techniques can be categorized based on different criteria [74], *e.g.*, feature extraction/selection, depending on whether a transformation is applied or some features from the original set are chosen, linear/nonlinear, depending on the type of mapping through the data are projected onto a lower-dimensional space, and supervised/unsupervised, depending on whether the label information is used in order to preserve the label discriminatory information after projection or not. It has to be highlighted, that between the obscure boundaries of feature extraction and selection, the promising framework of feature learning (via ANN) has recently arisen aiming to learn automatically more useful representations [5]. The same categorization holds for multi-label data as well.

Principal Component Analysis (PCA) [93], Latent Semantic Indexing/Analysis (LSI/LSA) [94], and Linear Discriminant Analysis (LDA) [95], constitute fundamental examples of linear transformations, whereas Locally Linear Embedding (LLE) [96], Isomap [97], and autoencoders [98], are representative examples of nonlinear. One should keep in mind that unsupervised techniques, such as PCA, LSI, and autoencoders, are directly applicable to multi-label data, whereas others assuming that the classes are mutually exclusive, such as LDA, have to be extended through a problem transformation method (*e.g.*, BR, LP) in order to be able to deal with the multi-label case [99]. Despite the significance of dimensionality reduction in multi-label learning, only a few of the existing techniques represent solutions designed exclusively to tackle multi-label data with their peculiarities [75]. Among them, we distinguish an algorithm called Multi-label Dimensionality reduction via

Dependence Maximization (MDDM) [100], which tries to find a projection such that the dependence between the feature and the corresponding label is maximized after projection. Finally, we mention that besides the feature space dimension reduction, in multi-label learning there is the homologous paradigm of Label Space Dimension Reduction (LSDR) [101], which tries to encode the original label space to a low-dimensional latent space using a decoding process for recovery.

## 3.6 Evaluation Measures

A multi-label classifier produces a set of predicted labels, but many implementations firstly output a score for each label, which is then compared to a threshold to obtain the predictions, as explained in Section 3.3. In this way, ultimately there exist two major tasks in supervised learning of multi-label data: *multi-label classification*, meaning to produce a bipartition of the labels into a relevant (positive) and an irrelevant (negative) set, and *label ranking*, meaning to map instances to a total strict order over a finite set of predefined labels [75]. Consequently, performance evaluation is significantly more complicated compared to the conventional supervised single-label learning and several metrics are required in order to properly evaluate an algorithm. We assume two major categories, namely the *example-based measures* which are calculated separately for each test example and then are averaged across the test set, and the *label-based measures* which evaluate the learning system's performance on each label separately, returning the micro/macro-averaged value across all labels [22].

Let  $p$  be the number of testing multi-label examples. Concerning the first group, we examine six metrics:

- *Hamming Loss* calculates the percentage of misclassified example-label pairs, considering the prediction error (an irrelevant label is predicted) and the missing error (a relevant label is not predicted) given by:

$$\text{Hamming Loss} = \frac{1}{p} \sum_{i=1}^p \frac{|Y_i \Delta Z_i|}{m}, \quad (3.6.9)$$

where  $\Delta$  stands for the symmetric difference between the two sets. The value

is between 0 and 1, with a lower value representing better performance. Due to the typical sparsity in multi-labeling, hamming loss tends to be a lenient metric.

- *Subset Accuracy* evaluates the fraction of correctly classified examples:

$$\text{Subset Accuracy} = \frac{1}{p} \sum_{i=1}^p I(|Y_i| = |Z_i|), \quad (3.6.10)$$

where  $I$  is the indicator function taking values  $I(\text{true}) = 1$  and  $I(\text{false}) = 0$ . Subset accuracy is a very strict accuracy metric since it classifies a sample as correctly predicted if all the predicted labels are identical to the true set of labels.

- *One-error* is a ranking based metric which computes how many examples have irrelevant top-ranked labels according to:

$$\text{One-error} = \frac{1}{p} \sum_{i=1}^p \delta(\arg \min_{\lambda \in \mathcal{L}} r_{\mathbf{x}_i}(\lambda)), \quad (3.6.11)$$

where  $\delta(\lambda) = 1$  if  $\lambda \notin Y_i$  and 0 otherwise.

- *Coverage* reports the average distance which needs to be traversed in order to cover all the relevant labels of the example from the ranked label list:

$$\text{Coverage} = \frac{1}{p} \sum_{i=1}^p \max_{\lambda \in Y_i} r_{\mathbf{x}_i}(\lambda) - 1. \quad (3.6.12)$$

- *Ranking Loss* evaluates the average fraction of labels pairs that are ordered incorrectly:

$$\text{Ranking Loss} = \frac{1}{p} \sum_{i=1}^p \frac{1}{|Y_i||\bar{Y}_i|} |\mathcal{G}_i|, \quad (3.6.13)$$

where  $\mathcal{G}_i = \{(\lambda', \lambda'') : r_{\mathbf{x}_i}(\lambda') > r_{\mathbf{x}_i}(\lambda''), (\lambda', \lambda'') \in Y_i \times \bar{Y}_i\}$ . Here,  $\bar{Y}_i$  denotes the complementary set of  $Y_i$  with respect to  $\mathcal{L}$ . In other words, ranking loss measures the ability to capture the relative order between labels.

- *Average Precision* expresses the percentage of labels ranked above a particular relevant label:

$$\text{Average Precision} = \frac{1}{p} \sum_{i=1}^p \frac{1}{|Y_i|} \sum_{\lambda \in Y_i} \frac{|\{\lambda' \in Y_i : r_{\mathbf{x}_i}(\lambda') < r_{\mathbf{x}_i}(\lambda)\}|}{r_{\mathbf{x}_i}(\lambda)}. \quad (3.6.14)$$

From information retrieval, we know that the evaluation metrics for a binary classification problem are based on the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) test samples. Based on the above, one can compute *Precision* as  $TP/(TP+FP)$ , *Recall* as  $TP/(TP+FN)$ , and the *F-Measure* as the harmonic mean between precision and recall. Extending this concept to multi-label problems, we can derive the corresponding quantities for each label  $\lambda_j \in \mathcal{L}$  calculating the micro-averaging operation [102]:

$$B_{\text{micro}} = B \left( \sum_{j=1}^m TP_{\lambda_j}, \sum_{j=1}^m TN_{\lambda_j}, \sum_{j=1}^m FP_{\lambda_j}, \sum_{j=1}^m FN_{\lambda_j} \right), \quad (3.6.15)$$

as well as the macro-averaging operation [102]:

$$B_{\text{macro}} = \frac{1}{m} \sum_{j=1}^m B(TP_{\lambda_j}, TN_{\lambda_j}, FP_{\lambda_j}, FN_{\lambda_j}), \quad (3.6.16)$$

where  $B$  is one of the previous mentioned classification metrics, and  $TP_{\lambda_j}$ ,  $TN_{\lambda_j}$ ,  $FP_{\lambda_j}$ ,  $FN_{\lambda_j}$  is the number of TP, TN, FP and FN after the binary evaluation for  $\lambda_j$ . Conceptually speaking, micro-averaging gives equal weight to each example and is an indicator of large classes, whereas macro-averaging to each label and gives a sense of effectiveness on small classes [103].

Finally, we consider the *Area Under the Curve* (AUC) metric which is calculated from the Receiver Operating Characteristic (ROC) curve. In case all annotations contain confidence values, the AUC score describes the overall quality of performance independently of individual threshold configurations regarding specific trade-offs between TP and FP [104]. More precisely, let the True Positive Rate (TPR) be defined as  $TP/(TP+FN)$  and the False Positive Rate (FPR) as  $FP/(FP+TN)$ . Then, each point on the ROC curve corresponds to a pair (TPR, FPR) for one threshold, and the area under this ROC curve is called micro-AUC [105] derived as:

$$\text{AUC}_{\text{micro}} = \frac{\left| \left\{ (\mathbf{x}', \mathbf{x}'', \lambda', \lambda'') \mid r_{\mathbf{x}'}(\lambda') \geq r_{\mathbf{x}''}(\lambda''), (\mathbf{x}', \lambda') \in \mathcal{R}^+, (\mathbf{x}'', \lambda'') \in \mathcal{R}^- \right\} \right|}{|\mathcal{R}^+| |\mathcal{R}^-|}, \quad (3.6.17)$$

where  $\mathcal{R}^+ = \{(\mathbf{x}_i, \lambda) \mid \lambda \in Y_i, 1 \leq i \leq p\}$  corresponds to the set of relevant, and  $\mathcal{R}^- = \{(\mathbf{x}_i, \lambda) \mid \lambda \notin Y_i, 1 \leq i \leq p\}$  to the set of irrelevant labels [22]. Subsequently, the macro-averaged AUC is the average AUC of the separate ROC curves for each

class and can be defined as follows:

$$\text{AUC}_{\text{macro}} = \frac{1}{m} \sum_{j=1}^m \frac{|\{(x', x'') \mid r_{x'}(\lambda_j) \geq r_{x''}(\lambda_j), (x', x'') \in \mathcal{Z}_j \times \bar{\mathcal{Z}}_j\}|}{|\mathcal{Z}_j| |\bar{\mathcal{Z}}_j|}, \quad (3.6.18)$$

where  $\mathcal{Z}_j = \{\mathbf{x}_i \mid \lambda_j \in Y_i, 1 \leq i \leq p\}$  is the set of test instances with label  $\lambda_j$ , and  $\bar{\mathcal{Z}}_j = \{\mathbf{x}_i \mid \lambda_j \notin Y_i, 1 \leq i \leq p\}$  is the set of test instances without  $\lambda_j$ . A value of 1 resembles to a perfect system.



# Chapter 4

## Feature Learning

### 4.1 Artificial Neural Networks

Computers and humans have very opposite capacities and strengths. Today’s computers are extremely fast and precise when doing computations, but humans inherently perceive, understand and generalize different things much better. In Metaphysics, Aristotle had famously said “*There is nothing in the intellect that was not previously in the sense*”, meaning that the true source of all our knowledge originates from experience; our evolutionary edge over animals. Far after that, human ability to recognise and hierarchically extract meaningful patterns for distinguishing different objects, has guided scientists to draw their inspiration from the architecture of the human brain in order to solve hard recognition problems.

ANN is a typical example in machine learning which tries to learn and behave in a remarkably similar way to human brain through a simplified mathematical model [106]. A typical brain contains around 100 billion miniscule cells, called neurons, that transmit impulses (electrical charges) from one location to another. According to a naive biological model, each neuron is made up of dendrites which receive signals from other neurons, the cell body which processes those signals, and the axon, a long “cable” that carries information away, as illustrated in Fig. 4.1a. Similarly, ANNs are composed of interconnected artificial neurons called units (or nodes), which represent the actual neurons that we find inside our brains. The neurons in an ANN are organized in layers, whereas each layer can have more than

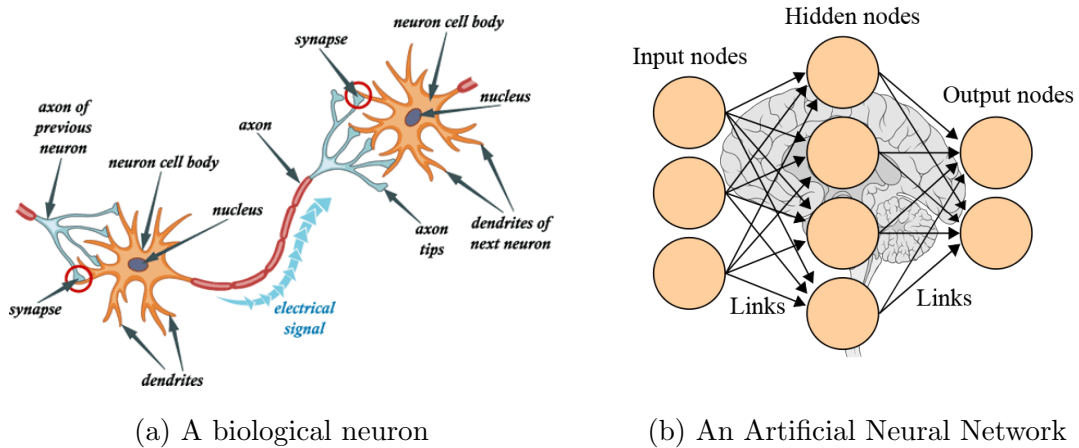


Figure 4.1: Neurons serve as the elementary building blocks of human brain information processing, similar to units of the artificial neural networks.

one neurons in parallel. Correspondingly, the way brain cells trigger one another across tiny junctions (called synapses), is modeled by the modifiable links (called synaptic weights) which are associated with each connection [107], as depicted in Fig. 4.1b. We notice that while this model has been highly successful in machine learning applications, it is a poor model for biological neurons, because it lacks the time-dependence that real neurons exhibit. This way, although some of the earliest biological models took the aforementioned form, nowadays, kinetic models, such as the Hodgkin-Huxley model, have become dominant [108].

There is a big variety of architectures for neural networks regarding the connections between different units and layers [109]. In this thesis, we focus on the feed-forward approach [110], which is by far the most common type of architecture. In this case, the first layer represents the input and the last the output, whereas it might be one or more layers of hidden units in between (see Fig. 4.1b). If more than one hidden layers exist, we have a deep neural network [111]. These networks compute a series of transformations between their input and output. Therefore, at each layer a new representation of the input is obtained; lower layers are relaxed to learn simple and concrete features, whereas higher layers tend to represent complex and abstract features.

In an ANN, the information processing is generally a two-stage procedure. The

first stage forms a linear sum of all the signals that are received from other neurons (likewise with what was mentioned before for biological neurons). The second stage is a nonlinear function, which produces the actual output of the unit and shows how well the information matches to what the neuron has become specialized to detect. These steps can be formally depicted using the following equation:

$$o(\mathbf{x}) = f\left(\sum_i \mathbf{W}_i \mathbf{x}_i + \mathbf{b}\right), \quad (4.1.1)$$

where  $o(\cdot)$  is the output function,  $f(\cdot)$  is a nonlinear transformation function,  $\mathbf{b}$  is a bias value allowing the transformation function a shift to the left or to the right,  $\mathbf{x}$  is the input to the neuron, and  $\mathbf{W}_i$  is the weight of the  $i$ -th input  $\mathbf{x}_i$  indicating the strength of the connection between the neuron and the previous layer.

## 4.2 Autoencoders Framework

The most successful and well-known example of deterministic unsupervised learning methods is the autoencoder. A classical autoencoder is a feed-forward ANN composed of an input and an output layer of the same size with a hidden layer in between. The input pattern has to pass through the hidden layer of the network before it is reconstructed at the output. Typically, the model is trained with back-propagation [112], aiming to learn an approximation  $\hat{\mathbf{x}}$  of the input, which would be ideally more useful compared to the raw input.

The only constraint that can be applied to classical autoencoders is altering the number of hidden units. In case the number of hidden units is smaller than the number of inputs, this constraint would have an effect similar to that of other dimensionality reduction techniques which learn a low-dimensional representation of the original feature set. Nevertheless, different kinds of regularizations can be applied to the cost function during autoencoder training in order to form more expressive features based on overcomplete representations. For instance, SAEs are a special case of the typical autoencoders, where the code is constrained to be sparse, *i.e.* only a small fraction of units are active during training.

Having the SAE as our structural unit, we present the end-to-end design of the

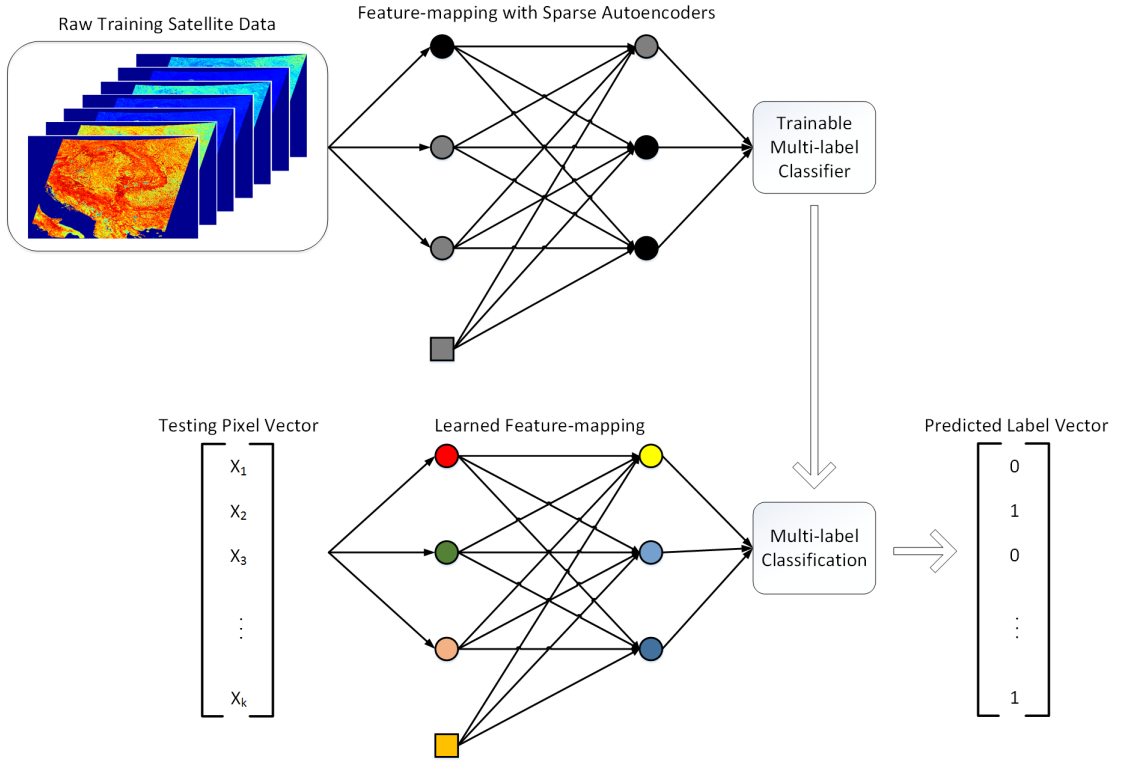


Figure 4.2: Visual illustration of the representation learning process with remotely sensed data. An initial feature-mapping with SAEs is performed in order to train a multi-label classifier, whereas we proceed to the testing procedure by exploiting the learned feature-mapping.

proposed scheme in Fig. 4.2. In specific, it is composed of the following three-stage pipeline:

- preprocessing and normalization of the features.
- feature-mapping using SAEs.
- multi-label classification (see Chapter 3) through the learned feature-mapping.

Through our analysis, we have experimented with several options for each module, trying to evaluate the impact of them to the final performance estimation.

#### 4.2.1 Single-layer sparse autoencoders

The feature mapping that transforms an input pattern  $\mathbf{x} \in \mathbb{R}^d$  into a hidden representation  $\mathbf{h}$  (called code) of  $k$  neurons (units), is defined by the *encoder* function:

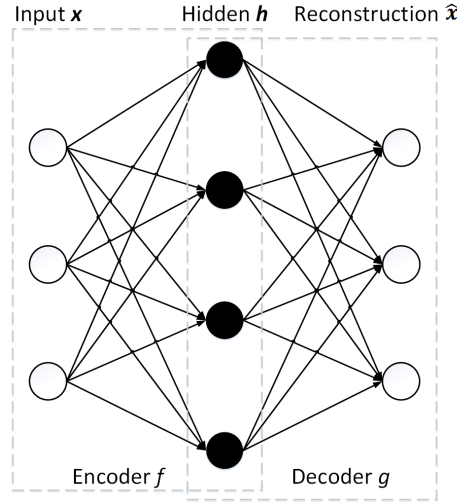


Figure 4.3: Architecture of an autoencoder with an overcomplete hidden layer. The encoder takes the input  $\mathbf{x}$  and computes a prediction of the best value of the latent code  $\mathbf{h}$ . The decoder is symmetric to the encoder and computes a reconstruction  $\hat{\mathbf{x}}$  from  $\mathbf{h}$ . The bias units are not considered for simplicity.

$$f(\mathbf{x}) = \mathbf{h} = \alpha_f(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \quad (4.2.2)$$

where  $\alpha_f : \mathbb{R} \mapsto \mathbb{R}$  is the *activation function* applied component-wise to the input vector. The activation function is usually chosen to be nonlinear; examples include the logistic sigmoid and the hyperbolic tangent. Recently, there is a growing interest in Rectified Linear Units (ReLU), which seem to work better in supervised recognition tasks [67]. The activation function is parametrized by a weight matrix  $\mathbf{W}_1 \in \mathbb{R}^{k \times d}$  with weights learned on the connections from the input to the hidden layer and a bias vector  $\mathbf{b}_1 \in \mathbb{R}^k$ . The network output is then computed by mapping the resulting hidden representation  $\mathbf{h}$  back into a reconstructed vector  $\hat{\mathbf{x}} \in \mathbb{R}^d$  using a separate *decoder* function of the form:

$$g(f(\mathbf{x})) = \hat{\mathbf{x}} = \alpha_g(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2), \quad (4.2.3)$$

where  $\alpha_g$  is the activation function,  $\mathbf{W}_2 \in \mathbb{R}^{d \times k}$  is the decoding matrix and  $\mathbf{b}_2 \in \mathbb{R}^d$  a vector of bias parameters which are learned from the hidden to the output layer. Overall, the architecture of a typical autoencoder with an overcomplete hidden layer is illustrated in Fig. 4.3.

The estimation of the parameters set  $\theta = \{\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2\}$  of an autoencoder, is achieved through the minimization of the reconstruction error between the input and the output according to a specific loss function. Given the training set, a typical loss function seeks to minimize the normalized least squares error, defining the following optimization objective:

$$J_{\text{AE}}(\theta) = \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{2} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \right), \quad (4.2.4)$$

where  $m$  is the number of training examples and  $\|\cdot\|$  is the Euclidean distance. The reconstruction  $\hat{\mathbf{x}}_i$  is implicitly dependent on the parameter set  $\theta$ . More advanced loss functions can also be involved [60]. A weight decay term commonly introduced to the cost function in order to prevent overfitting, has been found to influence marginal our data.

Signal and model sparsity have had a profound impact on signal processing and machine learning due to their numerous advantages, such as robustness, model complexity, generative and discriminative capabilities among others [113, 114]. Furthermore, evidence from neuroscience suggest that sparse networks are closer to biological neurons' responses, since the percentage of neurons being active at the same time is estimated between 1 and 4% of the total [115, 116].

Motivated by these facts, we use the SAE framework in order to promote sparsity in our system. In this way, we define a sparsity constant  $\rho$  and enforce the average latent unit activation to be close to the value of  $\rho$ . This is achieved by penalizing it with the Kullback-Leibler (KL) divergence, a function employed to measure the difference between Bernoulli distributions, namely the expected activation over the training set of hidden unit  $u$  ( $\hat{\rho}_u$ ) and its target value ( $\rho$ ) in our case:

$$\text{KL}(\rho || \hat{\rho}_u) = \rho \log \frac{\rho}{\hat{\rho}_u} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_u}, \quad \hat{\rho}_u = \frac{1}{m} \sum_{i=1}^m [f_u(\mathbf{x}_i)], \quad u = 1, \dots, k, \quad (4.2.5)$$

where  $f_u(\mathbf{x}_i)$  denotes the activation of hidden unit  $u$ . The KL distance reaches its minimum of 0 when  $\hat{\rho}_u = \rho$ , and extends up to infinity as  $\hat{\rho}_u$  increases, enforcing the  $\hat{\rho}_u$  not to significantly deviate from the desired sparsity value  $\rho$ . All in all, the smaller the value of  $\rho$ , the sparser the representation would be. The regularized cost function of a SAE constitutes of the reconstruction loss of a classical autoencoder

with an additional regularization through a *sparsity promoting term* [117] given by:

$$J_{\text{spAE}}(\theta) = J_{\text{AE}}(\theta) + \beta \sum_{j=1}^k KL(\rho || \hat{\rho}_u), \quad (4.2.6)$$

where the hyperparameter  $\beta$  determines the importance of the sparsity regularizer. Note that there have been also developed and other techniques to encourage sparsity in the representation [118].

A particular set of weights is updated by calculating the partial derivatives of  $J_{\text{spAE}}$  and applying the backpropagation algorithm [112]. This way, the training typically converges to a minimum, hopefully a global one, after a small number of iterations. The minimization of the model parameters  $\theta$  can be achieved by conventional optimization algorithms (*e.g.*, gradient descent), as well as with more sophisticated procedures, such as conjugate gradient and Broyden-Fletcher-Goldfarb-Shanno (BFGS) methods to speed up convergence.

### 4.2.2 Deep learning with stacked sparse autoencoders

Deep learning is a special case of representation learning which admits the property that multiple levels of representations are learned hierarchically, leading to more generic and beneficial features. Ultimately, the activity of the first layer neurons corresponds to the low-level features of the input, while higher-level abstract concepts are encoded in the subsequent hidden layers. More specifically, we provide the deep architecture with surface reflectance input data, which are the raw data collected from a remote sensing observation system, and try through a hierarchical approach to learn an “advanced” version of them, which would ideally match the capabilities of high quality hand-crafted features, such as NDVI or EVI. In this way, we aim at bypassing the requirements of empirical design of these features by an expert, and automatically learn representations which can substitute and enhance them. In parallel, due to the unsupervised nature of the processing, the proposed approach is more universal and could also work with other types of targets which are not chlorophyll or water sensitive, such as structures in urban areas, where analogous ratios have not been defined.

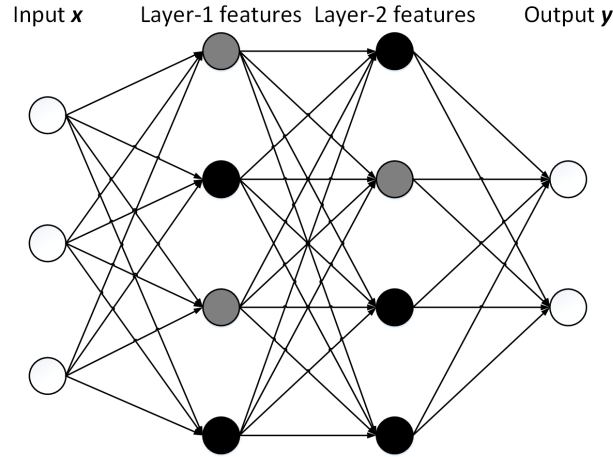


Figure 4.4: A 4 layer autoencoder network  $[3-4-4-2]$ , where the circles denote the feature units. The black and grey colors are used to denote the inactive and active hidden units, respectively, whereas the white the visible units. The two middle layers constitute an encoder. The bias units are not considered for simplicity.

Architectures with two or more hidden layers can be created by stacking single-layer autoencoders on top of each other as depicted in Fig. 4.4. Formally, one starts by training a SAE with the raw data as input. Then, the decoder layer is discarded so that the activations of the hidden units (layer-1 features) become the visible input for training the second autoencoder layer (feed-forward), which in turn produces another representation (layer-2 features). This greedy layer-by-layer process keeps the previous layers fixed and ignores interactions with subsequent layers, thus dramatically reducing the search over the parameter space. While this process can be repeated multiple times, rarely more than three hidden layers are involved. We can formalize a stacked autoencoder according to:

$$\mathbf{h}_L = f_L \left( \cdots f_2 \left( f_1 (\mathbf{x}) \right) \right), \quad (4.2.7)$$

where  $\mathbf{h}_L$  denotes the representation learned by the top layer  $L$ . The output of the entire architecture can be used to feed a stand-alone classifier, offering an improved representation of the data compared to the raw input.

The challenge in deep learning is that the gradient information is difficult to pass efficiently through a series of randomly initialized layers, since a good starting point is hard to identify. Unsupervised pretraining [71] is a recently developed



yet very influential protocol that helps to alleviate this optimization problem by introducing prior knowledge for initializing the weights of each layer, allowing gradients to “flow well”. Autoencoders, being a fundamental example of unsupervised learning, have attracted a lot of attention as a method for pretraining deep neural networks. Formally, we use the SAE as the building block to train one layer at a time, in a bottom up fashion, for a fixed number of updates (epochs). Up until this point, the procedure is completely unsupervised. Supervised refinements are subsequently introduced in the top layer of the deep architecture in order to fine-tune the gradient-based optimization algorithm with respect to a supervised criterion, a process termed fine-tuning phase [69]. As a last optional training stage, it is possible to further optimize the parameters with a global fine-tuning, which uses backpropagation through the whole network architecture at once, however starting from a very good initial model.



# Chapter 5

## Dataset Formulation

### 5.1 Key Idea

Sensors onboard platforms far away from their targets, typically support large fields of view, but cannot provide great detail due to distance from the ground and speed of the platforms. The detail discernible in an image depends on the spatial resolution of the sensor and refers to the smallest possible units that can be detected, *i.e.* pixels. The problem of scale incompatibility arises naturally since traditionally satellite data have a lower spatial resolution than ground-truth label data, which require human annotation through field-studies. A concrete instance is the MODIS data products which have a spatial resolution of  $500\text{m}^2$ , while CORINE land cover data of  $100\text{m}^2$ . This way, multiple CORINE “pixels” correspond to a single MODIS pixel, naturally leading to the case of multi-label annotation, where there are more than ones labels for a single MODIS pixel.

### 5.2 MODIS Data - Obtaining Features

NASA’s MODIS Earth Observation System is considered one of the most valuable sources of remote sensing data, aimed at monitoring and predicting environmental dynamics. The MODIS sensor can achieve global coverage with high temporal resolution, since it is able to scan the entire Earth’s surface (aboard the Terra and Aqua satellites) in one to two days, having a sun-synchronous orbit altitude of 705km. As

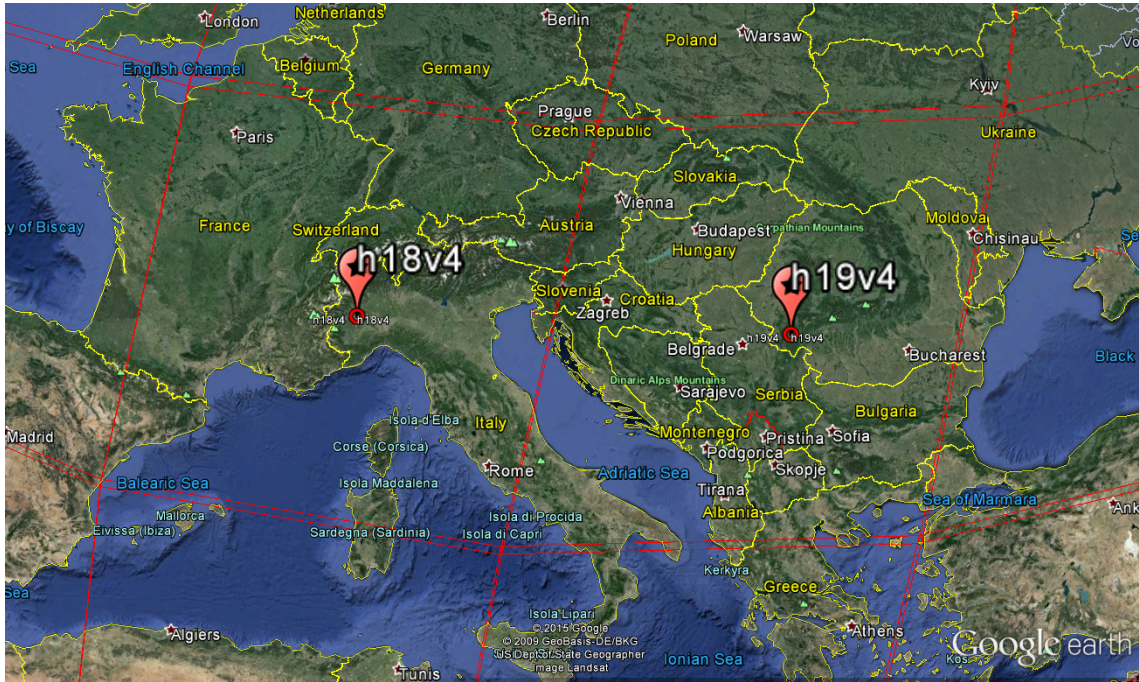


Figure 5.1: Geographic distribution of MODIS h18v04 and h19v04 tiles. The h18v04 region captures South-Central Europe, while h19v04 a large part of the Balkans. These exemplary regions were selected due to the diversity of land cover and the availability of data.

far as spectral resolution is concerned, MODIS acquires data in 36 spectral bands ranging from 400–14400nm. Note that the first two bands have a spatial resolution (pixel size at nadir) of 250m, bands 3 to 7 of 500m, and all the rest bands of 1km approximately. The sensor provides 12 bits radiometric sensitivity and achieves a swath of 2330km (across track) by 10km (along track at nadir). MODIS data are open-access and continuously updated since 2000.

The MODIS land native product files distributed by the Land Processes Distributed Active Archive Center<sup>1</sup> come in the Hierarchical Data Format (HDF) and in Sinusoidal (SIN) projection. As a result, MODIS data are grouped in 460 equal non-overlapping spatial tiles starting at (0,0) in the upper left corner and proceeding to the right (horizontal) and downward (vertical) until the lower right corner at (35,17). Each one of them captures approximately  $1200 \times 1200$  km of real land.

<sup>1</sup><https://lpdaac.usgs.gov/>

Nevertheless, SIN projection is not widely used and thus a common geographic projection is needed for our study. For this reason, we utilized the MODIS Reprojection Tool<sup>2</sup> (MRT) [119], which provides a basic set of routines for transformation of MODIS imagery into standard geographic projections. This way, we re-sampled the original data and changed the projection to Universal Transverse Mercator (UTM) to become compatible with the World Geodetic System 1984 (WGS 84) datum, which is the global coordinate system adopted by the Global Positioning System (GPS), see Appendix A for more. The area of our interest comprises of a central portion of the European continent, namely h19v04 (with the exception of regions from Ukraine and Moldova) and h18v04 image tiles (see Fig. 5.1).

In order to benefit from the high temporal resolution observations of MODIS, while simultaneously mitigating the effects of the low spatial resolution, we consider annual time series to monitor the best possible density and intensity of green vegetation growth. Our model takes into account a well known monitoring tool for vegetation health and dynamics, namely the NDVI [120] from the Level-3 product MOD13A1, collection 5 (500m spatial resolution, 16 days temporal granularity). It is empirically related to the reflectance measurements in the red and Near InfraRed (NIR) portion of the spectrum through the following formula:

$$\text{NDVI} = \frac{\rho_{\text{NIR}} - \rho_{\text{red}}}{\rho_{\text{NIR}} + \rho_{\text{red}}} . \quad (5.2.1)$$

Due to the the high discriminating capabilities of NIR versus visible wavelength, NDVI is more sensitive than a single wavelength and able to separate very well the living from stressed or dead plantation. Therefore, NDVI carries valuable information regarding surface properties and can effectively quantify the “floral” content of an area, *i.e.* the chlorophyll concentrations. Furthermore, as a ratio, it has the advantage of minimizing different types of noise (variations in irradiance, clouds, view angles, atmospheric attenuation and even calibration), but it also leads to insensitivities with respect to vegetation variations over certain land cover conditions [121]. NDVI is designed to standardize the vegetation indices values between -1 and +1, where higher values indicate more photosynthetically active land cover type [122].

---

<sup>2</sup>[https://lpdaac.usgs.gov/tools/modis\\_reprojection\\_tool](https://lpdaac.usgs.gov/tools/modis_reprojection_tool)

We collect all the available measurements from 10 months (March till December) leading to 19 values/features. For the final data calibration, we refer to the quality assurance metadata [123] supplied with the MOD13A1 product in order to assemble only reliable pixels (exclude unprocessed data).

Land Surface Temperature (LST) has been also proved to play a significant role in detecting several climatic, hydrological, ecological and biogeochemical changes [28], which are crucial parameters for land cover. LST observations are retrieved from the Thermal InfraRed (TIR) bands and are able to combine the results of all surface-atmosphere interactions and corresponding energy fluxes, measuring the additive compositions of TIR from background soils and overlying vegetation canopy. This way, whereas NDVI measurements estimate efficiently the vegetation cover, LST is more applicable for targets that are not chlorophyll sensitive [124]. Therefore, we enhance the previously selected examples by adding measurements (feature-level fusion) related to the LST daytime, extending the number of features to 57. The temperature data are included in the Level-3 MOD11A2 product, which stores the average values during an 8 day period on a 1km SIN grid. In order to obtain the same spatial resolution with MOD13A1, an oversampling to 500m spatial resolution is performed. Note that we take into account all available LST values of the 10 months, meaning that there arise abnormal data which are manifested as noise, since atmospheric conditions inevitably cause disturbances in the observations. In this study, we did not consider any available technique for denoising and improving the quality of the time series imagery, since naive techniques such as the Maximum Value Composite (MVC) [125], did not manage to provide higher performance.

### 5.3 CORINE Land Cover Data - Obtaining Labels

The CLC inventory was initiated in 1990 and has been updated in 2000 and 2006, while the latest version of the 2012 update is still under production. CLC consists of





Figure 5.2: CLC map for the h19v04 tile of 2000 (CLC2000).

44 classes<sup>3</sup> (CLC codes at Level 3 of the CORINE nomenclature), including artificial surfaces, agricultural and forest areas, wetlands, and water bodies overall. In this work, we utilize data from 2000<sup>4</sup> and 2006<sup>5</sup> at 100m<sup>2</sup> resolution (Version 17). The QGIS<sup>6</sup> software is employed in order to transform these raster-based Geographic Information System (GIS) data to WGS 84 format, so that become compatible with MODIS data, and subsequently extract the regions corresponding to the h19v04 and the h18v04 tiles through upper left and lower right latitude and longitude coordinates (more in Appendix B).

In order to construct the multi-label dataset, the CLC labels matrix was divided

---

<sup>3</sup><http://ec.europa.eu/agriculture/publi/landscape/about.htm>

<sup>4</sup><http://www.eea.europa.eu/data-and-maps/data/corine-land-cover-2000-raster-3>

<sup>5</sup><http://www.eea.europa.eu/data-and-maps/data/corine-land-cover-2006-raster-3>

<sup>6</sup><http://www.qgis.org/en/site/>

Table 5.1: The 20 selected ground-truth labels from CORINE with their original CLC code. The labels represent a wide range of materials that may be contained, from those existing in urban and vegetation areas to those in minerals and beaches. The number of examples in the last column refers to the CLC2000.

No.	CLC Code	Description	# Examples
1	111	Continuous urban fabric	1242
2	121	Industrial or commercial units	4234
3	122	Road & rail networks & assoc. land	1264
4	124	Airports	272
5	131	Mineral extraction sites	904
6	132	Dump sites	468
7	133	Construction sites	258
8	141	Green urban areas	1181
9	142	Sport and leisure facilities	1172
10	212	Permanently irrigated land	1411
11	213	Rice fields	889
12	223	Olive groves	1730
13	241	Annual crops assoc. with perm. crops	1246
14	322	Moors and heathland	2140
15	331	Beaches, dunes, sands	1258
16	332	Bare rocks	1907
17	411	Inland marshes	2101
18	412	Peat bogs	102
19	421	Salt marshes	736
20	521	Coastal lagoons	521

into non-overlapping blocks using a  $5 \times 5$  grid, since the MODIS pixel size is approximately 25 times the size of a CORINE pixel. As a result, a binary vector per sample is produced, where a value of one indicates that a label is present while a value of zero denotes that a label is absent. We select 20 labels as depicted in Table 5.1 and exclude examples composed of only one label. In this way, we acquire a challenging scenario for the multi-label learning algorithms, while in parallel we mitigate the unpleasant effects of the multi-label learning challenges discussed in Section 3.2 as much as possible.



## 5.4 Dataset Properties

Before running any experiment with multi-label data, it is important to highlight that not all multi-label datasets are equal, even if they have the same number of instances or labels. For example, the number of labels for each example can greatly vary across different datasets and this could highly influence the performance of the multi-label learning algorithms. Therefore, it would be unfair to compare different methods without noting the dataset properties.

This way, we estimate certain statistical metrics [21] in order to obtain a better understanding of the properties of our dataset. Let  $\mathcal{S}$  be the multi-label dataset consisting of  $|s|$  multi-label examples  $(\mathbf{x}_i, Y_i), i = 1, \dots, |s|$ . We take into account the following metrics:

- *Label Cardinality* (LC) calculates the average number of class labels associated with each instance in the dataset:

$$\text{LC}(\mathcal{S}) = \frac{1}{|s|} \sum_{i=1}^{|s|} |Y_i|. \quad (5.4.2)$$

LC is independent of the number of labels  $m$  that exist in the dataset and it is used to denote the number of alternative labels that characterize the  $|s|$  instances of a multi-label dataset. It gives a good idea of the label frequency, but gives no indication of the regularity of the labeling scheme. All in all, the larger the value of LC, the more difficult is to obtain good classification performance.

- *Label Density* (LD) is the cardinality normalized by the number of labels  $m$ :

$$\text{LD}(\mathcal{S}) = \frac{1}{|s|} \sum_{i=1}^{|s|} \frac{|Y_i|}{m}. \quad (5.4.3)$$

LD quantifies how dense (or sparse) the multi-label dataset is.

- *Distinct Labelsets* (DL) is the total count of number of distinct label combinations observed in the dataset:

$$\text{DL}(\mathcal{S}) = \left| \{Y_i \mid \exists \mathbf{x}_i : (\mathbf{x}_i, Y_i) \in \mathcal{S}\} \right|, i = 1, \dots, |s|. \quad (5.4.4)$$

DL expresses the number of different label combinations per example, and it is of key importance for methods that operate on label subsets.

Such inherent dataset statistics reveal the difficulties for each multi-label dataset and lead multi-label algorithms to perform differently based on their underlying assumptions, such as those discussed in Section 3.4. Table 5.2 summarizes the aforementioned statistics for the h19v04 tile of 2000 dataset, whereas it also includes some benchmark multi-label datasets along with their corresponding statistics from a variety of domains. More multi-label datasets are available online<sup>7</sup>. We notice that there are many datasets which vary a lot.

Table 5.2: Statistics for multi-label dataset based on the h19v04 tile of CLC2000.

Dataset	Domain	# Instances	# Labels	# Features	LC	LD	DL
land cover	remote sensing	12291	20	57	2.037	0.102	248
yeast [80]	biology	2417	14	103	4.237	0.303	198
scene [26]	image	2407	6	294	1.074	0.179	15
emotions [25]	music	593	6	72	1.869	0.311	27
mediamill [126]	video	43907	101	120	4.376	0.043	6555

<sup>7</sup><http://mulan.sourceforge.net/datasets-mlc.html>

# Chapter 6

## Experimental Results

### 6.1 Multi-label Learning Evaluation

In this section, we examine the performance of high-level multispectral satellite data products (NDVI and LST) under the following challenging scenarios:

- classification performance given a limited set of training examples.
- classification performance per label, aiming to investigate the label sensitivity.
- classification performance when the training data correspond to a specific *geographic region* and the testing data come from a neighboring region (different spatial tile) of the same year.
- classification performance when the training data correspond to a specific *time instance* and the testing data come from the same spatial location (tile), but from another time instance.
- comparison of multi-label classification with spectral unmixing.
- classification performance based on different types of fusion of the features.

Each experiment has its own distinct value and purpose, whereas the objective is to evaluate the proposed multi-label classification framework when applied under real-life conditions with remotely sensed data. Finally, we perform a sensitivity analysis to determine those parameters which influence the performance of the examined multi-label classification algorithms.

### 6.1.1 Experimental settings

In our analysis, we consider the algorithmic implementations included in the MULAN<sup>1</sup> Java library [127], an open source platform for the evaluation of multi-label algorithms that works on the top of the WEKA<sup>2</sup> framework. We make an initial split of the training to testing examples in the order of 7 : 3, although we are particularly interested in classification with very limited training examples, since under real-life conditions obtaining labeled remotely sensed pixels is a costly process.

In order to make a fair comparison for the algorithms, we applied the same base-level single-label classifier in all problem transformation and ensemble techniques. Particularly, we selected the C4.5 [128] DT learning algorithm (J48 implementation within WEKA), which is a well-known approach producing interpretable classification models. Concerning the individual parameters of the methods, they were instantiated following the recommendations from the literature. The ML-kNN and IBLR algorithms are parametrized by the size of the neighborhood, for which we adopted the value of  $k = 10$ . Zhang et al. [82] and Cheng et. al [91] have shown that this a reasonable choice since it provides balance between complexity and predictive performance. Besides the number of neighbors, ML-kNN requires also a smoothing parameter  $\gamma$  controlling the effect of a uniform prior on the estimation. We use a general rule where  $\gamma$  takes the value of 1, leading to a Laplace smoothing [82]. For the ensemble methods, the basic parameter is the number of component classifiers (models), whereas RAKEL needs an additional parameter which is the size of the labelsets. The number of models was set to 10 for ECC as proposed by Read et al. [87], whereas a rule-of-thumb setting for RAKEL is a size-3 subset with  $2m$  models in combination with the LP classifiers [92].

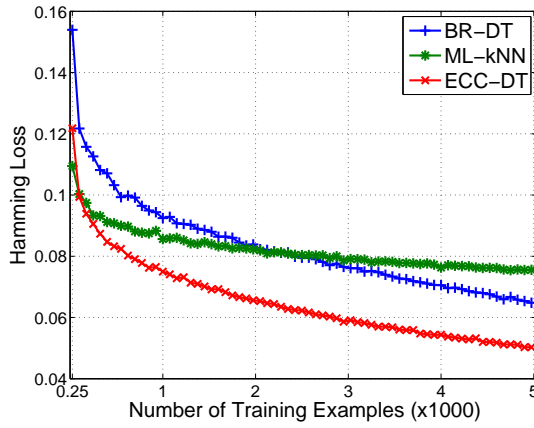
### 6.1.2 Classification performance with respect to training set

The objective of the first set of experiments is to evaluate the generalization capabilities of each learning algorithm. To that end, we evaluate the performance of each

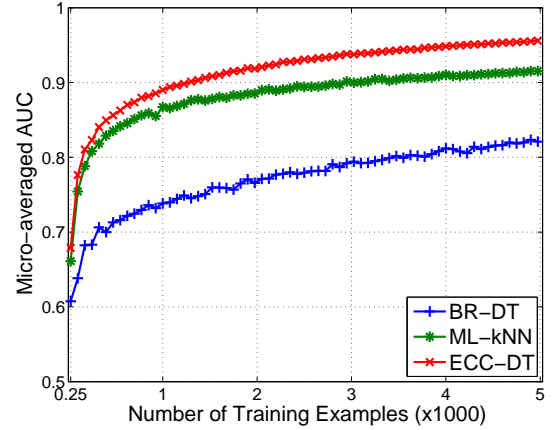
---

<sup>1</sup><http://mulan.sourceforge.net/>

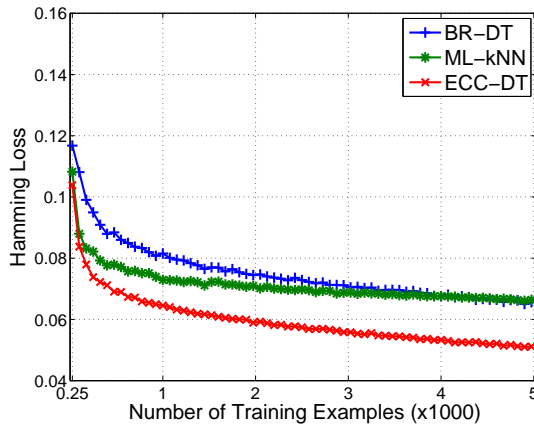
<sup>2</sup><http://www.cs.waikato.ac.nz/ml/weka/>



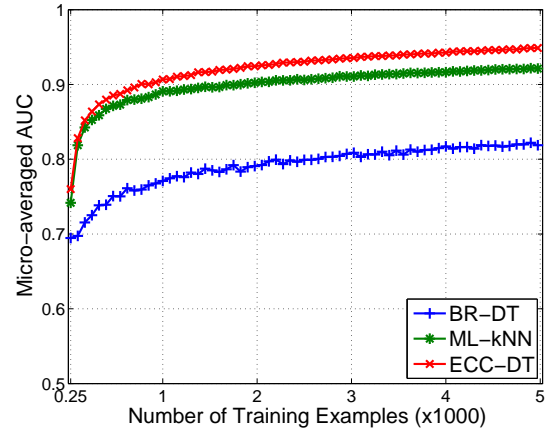
(a) Ham. Loss for h19v04 tile of CLC2000



(b) Micro AUC for h19v04 tile of CLC2000



(c) Ham. Loss for h18v04 tile of CLC2000



(d) Micro AUC for h18v04 tile of CLC2000

Figure 6.1: Classification performance with respect to the amount of training data (out of the 8604) corresponding to a single spatial tile. These experiments illustrate the complex interaction behavior between training data and performance. In general, one can observe that the performance gains are significantly more dramatic when increasing smaller sets of training examples, while the benefits of introducing more training data are moderate once sufficiently data is available.

method as a function of the number of training examples using the NDVI and the LST features. This is a critical parameter, because it is directly related to the cost and manpower required for the classification and understanding of newly acquired remotely sensed images. We consider a varying number of training examples ranging from 25 : 50 : 5000, where the error is averaged over 10 executions.

In Fig. 6.1, we present the performance of multi-label classification for two

specific spatial tiles, namely h19v04 and h18v04, using data from the CLC2000 inventory. In detail, we present the hamming loss and the micro-averaged AUC associated with the BR-DT, the ML-kNN, and the ECC-DT algorithms (one algorithm of each category). These two metrics are highly representative, since the hamming loss belongs to the example-based metrics and can give us an overall intuition of the misclassified instance-label pairs, whereas the AUC is a label-based metric looking at each label independently.

We can see that for both metrics the performance monotonically increases while gradually increasing the training size, with a fast initial rate and then with a slower one. This indicates that the algorithms indeed learn and use information from the training data in order to ameliorate their predictions. Looking closer at each metric, we observe that the performance of the three classifiers according to the AUC metric is quite stable across tiles examined on the same labels (especially for a large number of training examples), whereas slightly differences can be attributed to the variation of the intrinsic spatio-spectral characteristics of each tile. Analogous results arise considering the hamming loss as well. In this case, we further observe that when a large number of training examples is employed, BR-DT outperforms ML-kNN, indicating that in some cases “letting the data speak for themselves” can allow simple algorithms to beat more complex models. However, overall ECC-DT performs better than the other two algorithms in this particular experimental setup, a behavior that has also been observed in other scenarios of multi-label classification [87, 129], due to its internal scheme that benefits from label dependencies. We note that comparable graphs arise also for all the remaining metrics.

### 6.1.3 Introducing the “multi-label confidence maps”

For a better understanding of how each algorithm performs and especially how can these differences in the metrics be better perceived, we introduce the notion of the “multi-label confidence map”. As explained in Section 3.6, multi-label classification is often conducted in a two-stage process where, upon receiving a test instance, real-valued confidence outputs are initially provided for all labels, and then an additional function creates a bipartition of relevant and irrelevant labels [75]. Here, we utilize

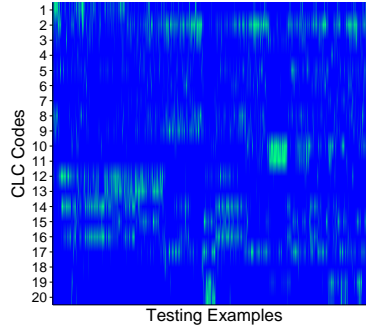


Figure 6.2: Ground-truth multi-label map for h19v04 of CLC2000 corresponding to a binary matrix indicating which labels are active for each example, *i.e.*, spatial location. The vertical axis corresponds to specific label as illustrated in Table 5.1, while horizontal axis to specific testing example out of the 3687.

these confidence outputs of the multi-label learning algorithms in order to produce confidence maps through which one can interpret the results visually (inspired from the abundance maps of spectral unmixing). Each row of this map represents labels with a corresponding CLC code, whereas each column represents an example (spatial location). In Fig. 6.2, we present the ground-truth map for h19v04 of CLC2000. This image is actually a binary matrix, where the value of 1 indicates presence of a specific label for an example, while 0 denotes absence of that label. For instance, regarding the first example (*i.e.*, column), labels 1 and 8 are active, indicating that CLC labels with codes 111 and 141 exist in this pixel.

In Fig. 6.3 and 6.4, we visualize the performance of BR-DT, ML-kNN and ECC-DT classifiers with the help of the “multi-label confidence map”, where each pixel in the map takes a confidence value ranging from 0 to 1 (results averaged over 50 realizations and then scaled to  $[0, 1]$  interval). Values closer to 0 indicate that the label is less likely to be enabled, whereas values closer to 1 indicate that the label has a stronger probability of being active.

Using these maps, we can visually verify the performance is higher due to the use of more training examples, by examining for instance the label 2 associated with the CLC code 121 (*i.e.*, second row). When the algorithms utilize 128 training examples, they assume that almost all samples have this label enabled, something that is not in accordance with the ground-truth data presented in Fig. 6.2. On the contrary,

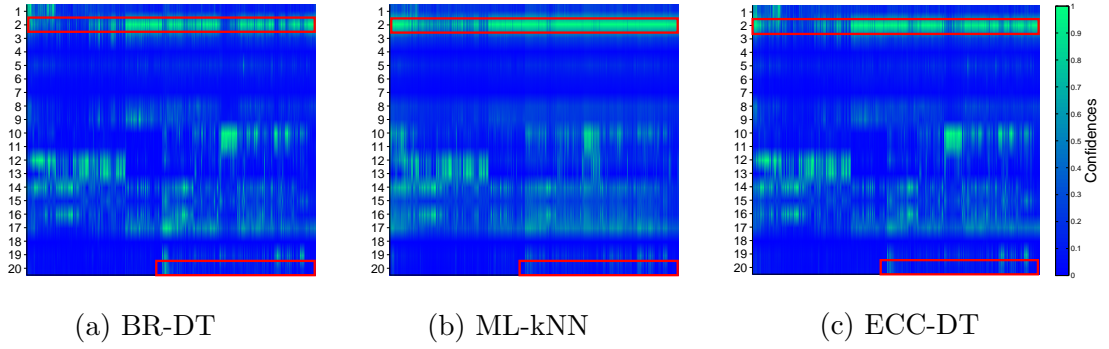


Figure 6.3: Multi-label confidence maps for h19v04 of CLC2000 with 128 training samples. The red boxes outline areas where there is significant deviation between the predicted and the ground-truth labels. They highlight that some labels in classification are more sensitive than the others.

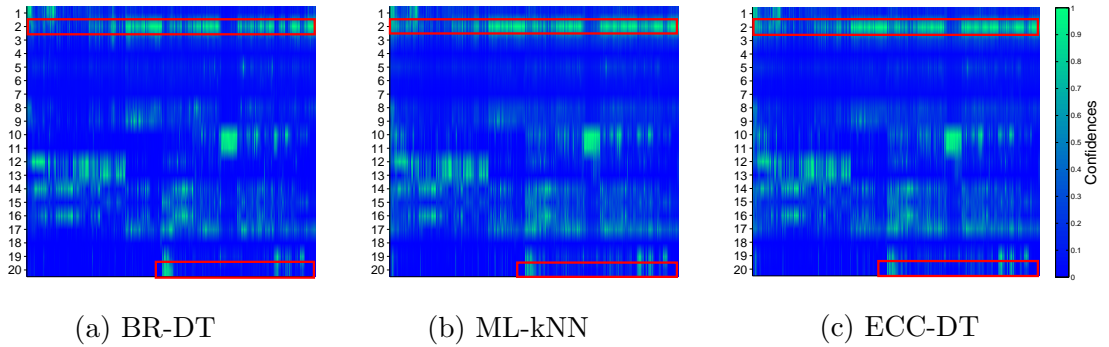


Figure 6.4: Multi-label confidence maps for h19v04 of CLC2000 with 1024 training samples. Similar to before the red boxes outline areas where there is significant deviation between the predicted and the ground-truth labels. Comparing to the case of 128 training examples shown above, we observe less errors according to the ground-truth map in Fig. 6.2.

when the algorithms use 1024 training examples, we can see that their revised predictions become more accurate and reliable, since many of the corresponding matrix positions which were previously close to 1, now obtain a lower confidence value. This observation suggests that for the specific label, many FP examples arise. FP are taken into account by the precision metric, where BR-DT and ECC-DT outperform ML-kNN for the specified number of training examples.

Another illustrative paradigm occurs when we take into account the label 20



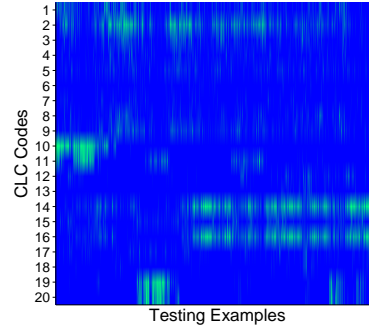


Figure 6.5: Ground-truth multi-label map for h18v04 of CLC2000. The vertical axis corresponds to specific label as illustrating in Table 5.1, while the horizontal axis to specific testing example out of the 7624.

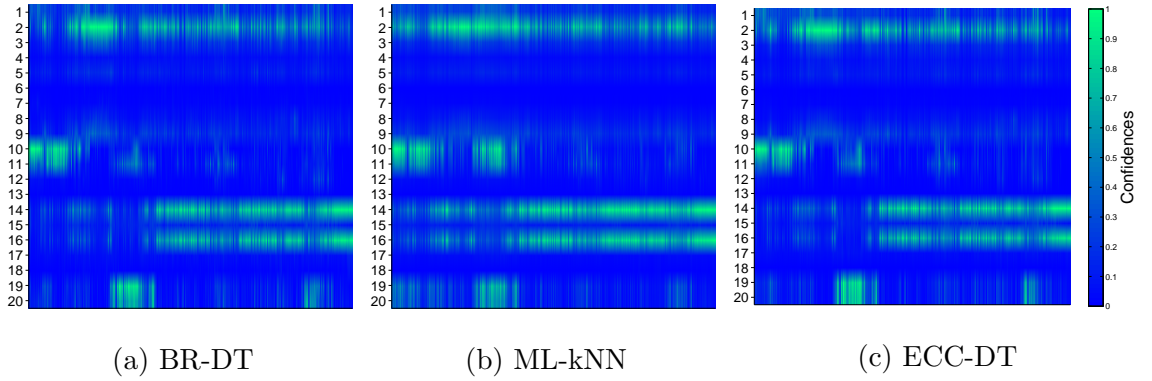


Figure 6.6: Multi-label confidence maps for h18v04 of CLC2000 with 128 training samples.

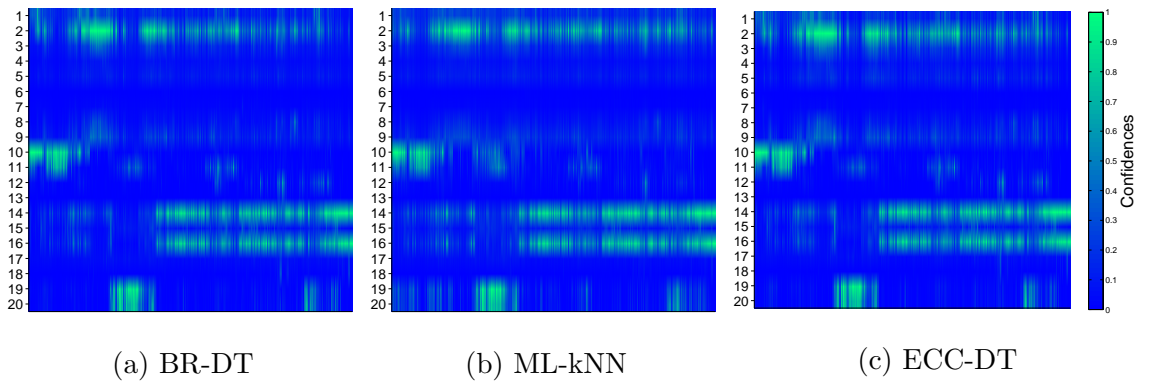


Figure 6.7: Multi-label confidence maps for h18v04 of CLC2000 with 1024 training samples. Comparing to the case of 128 training examples shown above, we observe less errors according to the ground-truth map in Fig. 6.5.

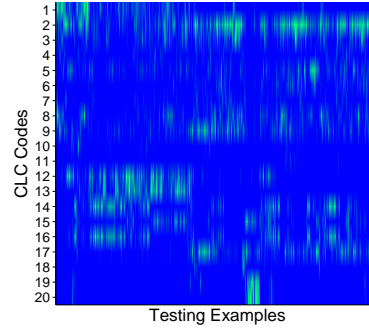


Figure 6.8: Ground-truth multi-label map for h19v04 of CLC2006. The vertical axis corresponds to specific label as illustrating in Table 5.1, while the horizontal axis to specific testing example out of the 2745.

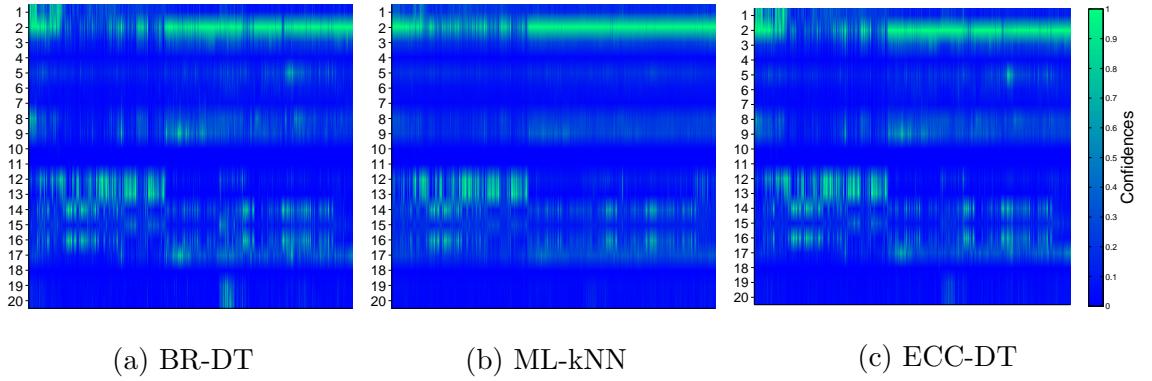


Figure 6.9: Multi-label confidence maps for h19v04 of CLC2006 with 128 training samples.

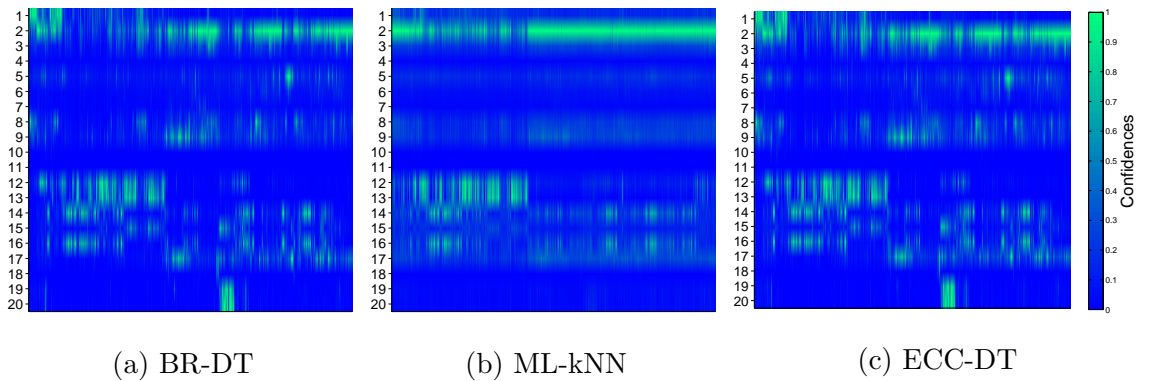


Figure 6.10: Multi-label confidence maps for h19v04 of CLC2006 with 1024 training samples. Comparing to the case of 128 training examples shown above, we observe less errors according to the ground-truth map in Fig. 6.8.

with CLC code 521 (*i.e.*, last row). In this case, we observe that the classification algorithms, cannot detect that this label is present in some pixels when utilizing 128 training examples, however, the prediction improves dramatically with 1024 training examples for all of them. In other words, we have the case of recall and FN, where we show that the BR-DT and the ECC-DT algorithms achieve almost similar performance, whereas ML-kNN exhibits a significant performance lag.

One should keep in mind that such visualizations can be produced for any single spatial tile of a specific time instance. In this manner, we extend the previous experiments by presenting two new sets of multi-label confidence maps for h18v04 tile of CLC2000 in Figures 6.5, 6.6, and 6.7, as well as for h19v04 of CLC2006 in Figures 6.8, 6.9, and 6.10, correspondingly. We observe once more that the algorithms have pronounced troubles in detecting specific labels accurately (*e.g.*, labels 2,19, and 20), but with the incorporation of more training examples the predictions improve remarkably.

#### 6.1.4 Comparison of multi-label classification algorithms

Regarding the h19v04 tile of CLC2000 as our reference region, an overview of the performance is presented in Table 6.1, where we have included all the performance metrics. For the evaluation of experiments, we performed 10 different 10-fold cross validation experiments, whereas the reported results are averages over these 100 runs of the different algorithms.

When considering the problem transformation methods, we observe that BR-DT is better than LP-DT in all example-based metrics except subset accuracy, which is notably high among all examined methods, suggesting that LP-DT is able to capture very well the underlying statistics of the labels. For the label-based metrics, the results are more balanced, since BR-DT is superior in precision and F-measure, whereas LP-DT is slightly better with respect to AUC. Regarding the algorithm adaptation methods, we observe that IBLR has a small lead, but overall ML-kNN and IBLR are on equal footing, since the observed variation in prediction accuracy manifested in the evaluation metrics is of little statistical significance for most of the metrics. Analyzing the last category of ensemble methods, RAkEL-DT

Table 6.1: Performance (mean  $\pm$  std) of each multi-label learning algorithm over 10 different 10-fold cross validation experiments. For each metric,  $\uparrow$  indicates “higher the better”, whereas  $\downarrow$  indicates “lower the better”. Ensemble methods perform overall higher than problem transformation and algorithm adaptation techniques.

Measure	Multi-Label Learning Algorithm					
	BR-DT	LP-DT	ML-kNN	IBLR	RAkEL-DT	ECC-DT
Hamming Loss $\downarrow$	0.044 $\pm$ 0.002	0.047 $\pm$ 0.002	0.067 $\pm$ 0.001	0.067 $\pm$ 0.001	<b>0.030 <math>\pm</math> 0.002</b>	0.033 $\pm$ 0.001
Subset Accuracy $\uparrow$	0.496 $\pm$ 0.013	0.626 $\pm$ 0.015	0.305 $\pm$ 0.010	0.318 $\pm$ 0.014	<b>0.638 <math>\pm</math> 0.014</b>	0.597 $\pm$ 0.013
One-error $\downarrow$	0.189 $\pm$ 0.013	0.289 $\pm$ 0.015	0.269 $\pm$ 0.015	0.268 $\pm$ 0.012	<b>0.098 <math>\pm</math> 0.010</b>	0.108 $\pm$ 0.009
Coverage $\downarrow$	4.793 $\pm$ 0.236	5.743 $\pm$ 0.176	2.987 $\pm$ 0.073	3.008 $\pm$ 0.054	2.436 $\pm$ 0.179	<b>1.939 <math>\pm</math> 0.075</b>
Ranking Loss $\downarrow$	0.120 $\pm$ 0.008	0.177 $\pm$ 0.008	0.072 $\pm$ 0.004	0.073 $\pm$ 0.003	0.044 $\pm$ 0.006	<b>0.030 <math>\pm</math> 0.002</b>
Average Precision $\uparrow$	0.802 $\pm$ 0.010	0.742 $\pm$ 0.012	0.762 $\pm$ 0.009	0.763 $\pm$ 0.006	<b>0.900 <math>\pm</math> 0.008</b>	0.896 $\pm$ 0.005
Macro Precision $\uparrow$	0.770 $\pm$ 0.013	0.731 $\pm$ 0.015	0.679 $\pm$ 0.033	0.669 $\pm$ 0.019	0.874 $\pm$ 0.010	<b>0.897 <math>\pm</math> 0.015</b>
Macro Recall $\uparrow$	0.724 $\pm$ 0.016	0.729 $\pm$ 0.013	0.419 $\pm$ 0.008	0.451 $\pm$ 0.016	<b>0.772 <math>\pm</math> 0.011</b>	0.692 $\pm$ 0.011
Macro F-Measure $\uparrow$	0.743 $\pm$ 0.013	0.727 $\pm$ 0.012	0.483 $\pm$ 0.012	0.520 $\pm$ 0.016	<b>0.814 <math>\pm</math> 0.009</b>	0.766 $\pm$ 0.011
Macro AUC $\uparrow$	0.864 $\pm$ 0.007	0.878 $\pm$ 0.010	0.913 $\pm$ 0.005	0.919 $\pm$ 0.005	0.953 $\pm$ 0.005	<b>0.968 <math>\pm</math> 0.003</b>
Micro Precision $\uparrow$	0.795 $\pm$ 0.009	0.768 $\pm$ 0.013	0.746 $\pm$ 0.014	0.735 $\pm$ 0.010	0.881 $\pm$ 0.009	<b>0.897 <math>\pm</math> 0.006</b>
Micro Recall $\uparrow$	0.768 $\pm$ 0.011	0.766 $\pm$ 0.012	0.516 $\pm$ 0.013	0.531 $\pm$ 0.009	<b>0.820 <math>\pm</math> 0.011</b>	0.761 $\pm$ 0.011
Micro F-Measure $\uparrow$	0.782 $\pm$ 0.009	0.767 $\pm$ 0.012	0.610 $\pm$ 0.009	0.616 $\pm$ 0.009	<b>0.850 <math>\pm</math> 0.009</b>	0.823 $\pm$ 0.008
Micro AUC $\uparrow$	0.882 $\pm$ 0.008	0.891 $\pm$ 0.008	0.942 $\pm$ 0.003	0.940 $\pm$ 0.002	0.967 $\pm$ 0.004	<b>0.980 <math>\pm</math> 0.002</b>
Training Time (sec)	48.19 $\pm$ 1.800	<b>19.46 <math>\pm</math> 0.564</b>	37.40 $\pm$ 2.628	49.03 $\pm$ 1.855	243.3 $\pm$ 5.418	761.8 $\pm$ 134.8
Testing Time (sec)	0.436 $\pm$ 0.035	<b>0.378 <math>\pm</math> 0.094</b>	4.220 $\pm$ 0.333	4.730 $\pm$ 0.472	0.507 $\pm$ 0.058	0.935 $\pm$ 0.178

has a clear advantage when it comes to metrics such as subset accuracy and recall, however, ECC-DT achieves superior performance for precision and AUC. Moreover, we validate that the ensemble frameworks of RAkEL and ECC improve remarkably the performance of the transformation methods on which they are based (*i.e.*, LP and BR, respectively).

All in all, one can argue that the ensemble methods confirmed their reputation as the most powerful class of multi-label classification algorithms, since they achieve a higher and more robust performance compared to other methods. On the other side, this higher performance implies and a substantially higher computational cost, as it is shown in the runtimes reported in Table 6.1 on a typical workstation. Between the remaining two categories, *i.e.* algorithm adaptation and problem transformation methods, results are balanced and largely dependent on the metric which one is

interested to optimize.

### 6.1.5 Classification per label

Table 6.2 shows the classification accuracy of the multi-label algorithms for each of the examined labels separately (as if they were independently predicted), along with the average accuracy in the last column. Based on the ease of predictions, we can rank the labels in the following descending order: 18, 4, 7, 20, 11, 19, 13, 6, 10, 16, 14, 12, 5, 17, 1, 15, 9, 8, 3, and 2. We notice that the hardest labels are the “industrial or commercial units” with a mean accuracy of approximately 86%, followed by “road

Table 6.2: The mean accuracy per label over 10 different 10-fold cross validation experiments. Some labels are more sensitive than others in classification.

Label No.	BR-DT	LP-DT	ML-kNN	IBLR	RAkEL-DT	ECC-DT	Avg.
1	0.940	0.935	0.918	0.919	0.962	0.953	0.938
2	0.870	0.861	0.814	0.815	0.909	0.894	0.861
3	0.930	0.925	0.907	0.907	0.955	0.947	0.928
4	0.984	0.982	0.978	0.978	0.988	0.986	0.983
5	0.947	0.943	0.932	0.931	0.963	0.959	0.945
6	0.970	0.967	0.964	0.964	0.981	0.977	0.971
7	0.982	0.977	0.979	0.978	0.985	0.984	0.981
8	0.932	0.929	0.911	0.910	0.957	0.947	0.931
9	0.934	0.928	0.912	0.911	0.960	0.949	0.932
10	0.971	0.969	0.934	0.935	0.980	0.978	0.961
11	0.981	0.981	0.959	0.959	0.989	0.986	0.976
12	0.952	0.947	0.923	0.923	0.969	0.960	0.945
13	0.977	0.974	0.964	0.963	0.982	0.979	0.973
14	0.959	0.955	0.928	0.928	0.975	0.968	0.952
15	0.942	0.938	0.909	0.909	0.958	0.955	0.935
16	0.960	0.957	0.933	0.935	0.975	0.969	0.954
17	0.918	0.913	0.967	0.965	0.947	0.937	0.941
18	0.995	0.995	0.994	0.994	0.997	0.996	0.995
19	0.980	0.978	0.960	0.960	0.987	0.984	0.975
20	0.983	0.983	0.968	0.969	0.988	0.987	0.979

and rail networks and associated land”, “green urban areas”, and “sport and leisure facilities”, with a mean accuracy of approximately 93%, indicating that in general buildings and urban areas are a hard target for identification. On the contrary, “peat bogs” is the easiest one with a mean accuracy of approximately 99%, followed by “airports”, “construction sites”, and “coastal lagoons”, with a mean accuracy of approximately 98%.

Based on the results of Table 6.2, one can see that the classification model has difficulties discriminating in general artificial surfaces (apart from airports and construction sites), whereas it achieves better performance for wetlands and natural areas. This fact can be attributed to the features we use (especially NDVI), which are more appropriate for detecting targets that follow a periodic profile regarding the natural greenery variations (cycles) throughout a year (time series analysis). On the contrary, urban characteristics, being prone to city’s unpredictable environmental conditions and insensitive to chlorophyll or water factors, violate these standard seasonal changes and thus are much harder to identify.

### 6.1.6 Classification on different spatial regions and temporal instances

In this section, we examine the performance of the algorithms when the training examples are acquired from a specific region at a given year, while testing takes place either on a neighboring region, or on a different time instance. We initially examine the classification efficiency of a neighboring geographic region, since accurate prediction of the labels in another region implies that if we have enough training examples of a specific label, then we can test the presence of this label in unexplored locations, avoiding the high cost of hand-collecting new annotated training examples. Motivated by this, we consider a different experimental setup, where three different types of training sets are used, namely a training set from the same tile (h18v04), a training set from another tile (h19v04), and a mixed training set containing all training examples from the reference tile (h19v04) and only a few (*i.e.*, 1024) training examples from the target tile (h18v04). The classifier that was selected for this set of experiments was the ensemble method ECC-DT.

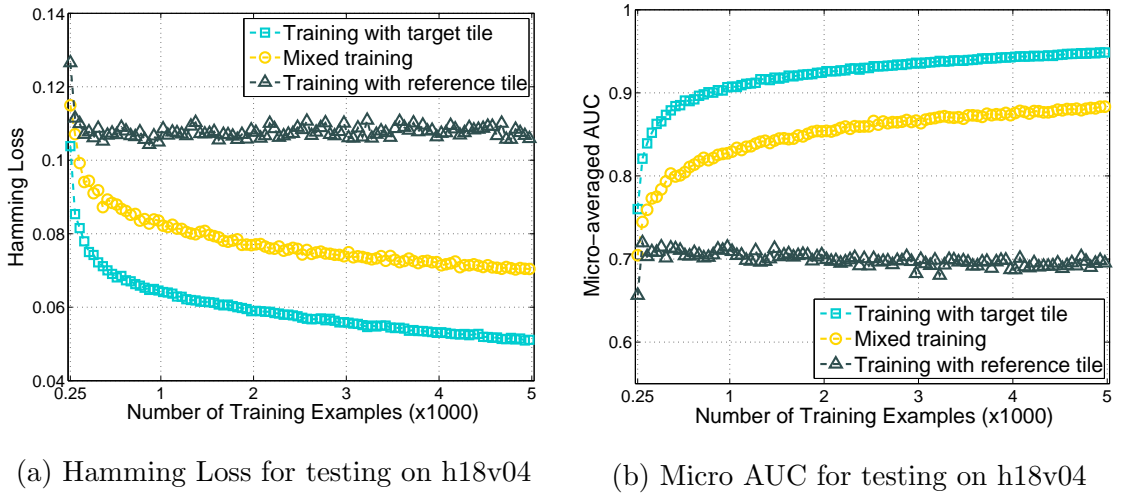


Figure 6.11: Multi-label classification performance with respect to amount of training data for tiles originating from different spatial locations using ECC-DT. Naturally, the results verify that training using data from the same spatial location produce the optimal classifier, while changing the location can have dramatic effects on performance. Fortunately, exploiting a mixed training set composed of data from both the corresponding data and location, as well as different location can achieve a very good performance, approaching the “optimal” performance.

Analyzing Fig. 6.11, we observe that when the training and the testing sets originate from the same tile, the performance is the best that can be achieved. Nevertheless, there is a significant degradation in performance when the training set originates from another tile. We observe that although the performance improves initially, it soon reaches a plateau which is lower than when the same dataset is used for both training and testing. This is where the third described training set comes into play. As we can see, the performance in mixed training conditions is very close to the performance achieved in the benchmark case. This behavior suggests that one can exploit already acquired annotated data, avoiding the effort required for collecting new annotated data, and still achieve a performance comparable to the performance achieved when the typical training-testing scenario is employed.

The last experiment, which we present in Fig. 6.12, examines the predicting performance for data from the h19v04 tile in CLC2006 under three different training sets, namely using a training set from the same tile (h19v04) in the same year



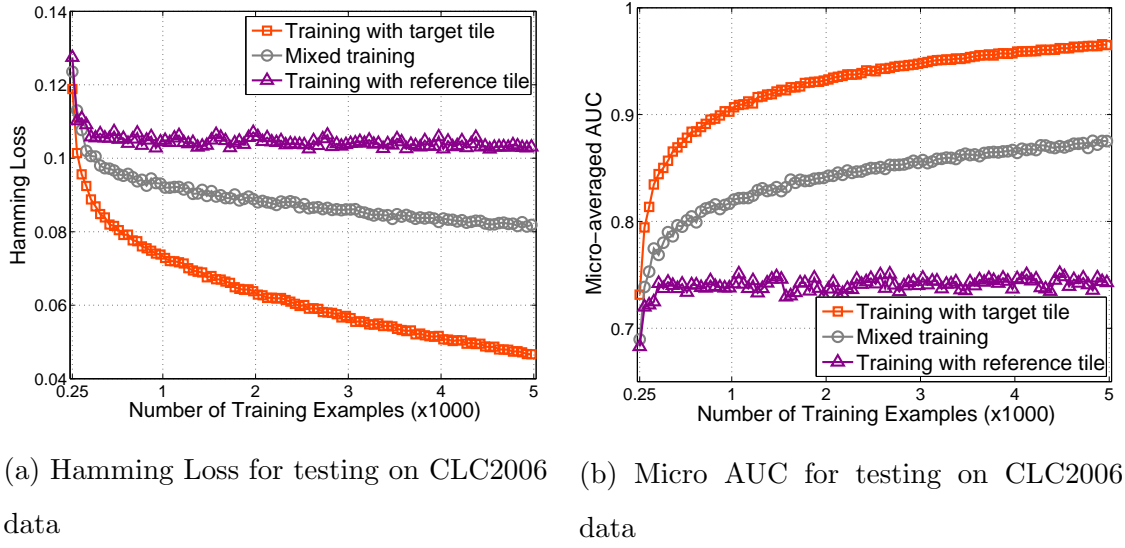


Figure 6.12: Multi-label classification performance with respect to amount of training data for tiles originating from same spatial locations but different time instances using ECC-DT. Similar to Fig. 6.11 the results are not good when using the training set of the reference tile, whereas by incorporating the mixed training set we achieve a very good performance, approaching the “optimal” one.

(2006), using a training set from the same tile (h19v04) in another year (2000), and a training set composed of all training from the reference tile enhanced by 1024 training examples from the target tile. In this case, the objective is to forecast the presence/absence of specific labels in order to understand the temporal evolution of land cover for this region. This is an immensely important scenario since obtaining up-to-date field-based annotation is extremely challenging, causing very low update rates that characterize CLC. This is a problem for land cover maps in general, leading to produced data that are outdated at release time. Similar to the previous case, we observe that the prediction performance when utilizing examples from the reference tile flattens again for the two metrics, but the performance gradient is smoother.

### 6.1.7 Comparison with spectral unmixing

Spectral unmixing and multi-label classification in remote sensing can both operate under the scenario that an observed spectral vector can be actually composed of one



or more materials (in contrast to single-label classification). Nevertheless, a direct comparison between the two methods is very difficult for various reasons. More specifically, the objective of spectral unmixing is the estimation of the abundance (proportion) of each endmember in an observed spectral vector, while multi-label classification aims at estimating a bipartition and a strict total order (ranking) of all labels. Furthermore, from a machine learning perspective, endmember extraction can be achieved either by defining a spectral library (supervised), or by assuming a known number of endmembers and trying to identify their characteristics (semi-supervised), or by trying to estimate both the number and the characteristics of the endmembers from the data (unsupervised). Since no spectral libraries encoding the land cover maps we consider are available, we rely on a semi-supervised approach, where the number of endmembers corresponds to the number of land cover classes we consider. Meanwhile, abundance estimation is also unsupervised, since no information regarding the concentration of the specific land cover labels per pixel exists. On the other hand, multi-label classification adheres to the supervised learning paradigm and strongly utilizes the provided labels during training.

Due to these different learning characteristics of spectral unmixing and multi-label learning, a direct comparison requires certain modifications to be made and extra assumptions to be introduced. Firstly, we consider the semi-supervised version of unmixing, where the number of endmembers is set to be equal to the number of labels  $m$ , in order to be able to utilize the land cover ground-truth. Moreover, in order to support a direct comparison, the output of spectral unmixing, which is the proportions of endmembers, must be converted into a binary format indicating presence or absence of labels. This is accomplished by introducing a threshold  $T$  and by selecting only the endmembers that exceed this threshold. For this reason, we introduce the ASC in the abundance estimation algorithms, in addition to the ANC constraint of endmembers that we also consider. Practically, by introducing the ASC and the ANC, the output of unmixing is probabilistic, and a thresholding operator is applied for selecting only the endmembers that are found to be statistically significant. In order to satisfy the ASC, we convert the 1's indicating the label existence to probabilities that sum up to one (*i.e.*, if two labels are present in

a pixel, we assign to the corresponding positions the value of 0.5). With the above in mind, we proceed to the comparison between spectral unmixing and multi-label classification for real remotely sensed multispectral data.

Regarding spectral unmixing, we consider different combinations of the two underlying processes, namely endmember extraction and abundance estimation, while in all cases, we use the authors' implementation and suggested default settings. This way, in order to unmix the reference tile h19v04 of CLC2000, we initially decompose the measurements into a library  $\mathbf{A} \in \mathbb{R}^{d \times m}$ , where  $d$  is the number of bands/features, and  $m$  is the number of endmembers/labels. In this step, three state-of-the-art algorithms have been incorporated, namely the N-FINDR, the VCA, and the SISAL. Then, for the fractional abundance estimation (inversion step) we evaluate two state-of-the-art methods, namely the SUnSAL which uses sparse regression under the LMM, and a gradient-based algorithm developed in [59] which assumes the PPNMM.

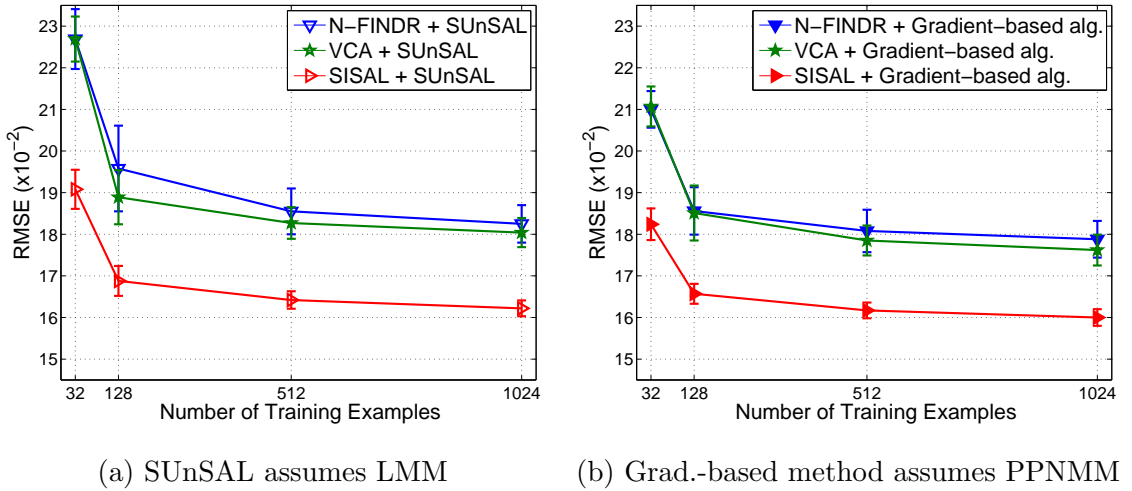


Figure 6.13: RMSE w.r.t. the number of examples for h19v04 of CLC2000 over 30 random runs. The approximation for all the examined unmixing chains improves, suggesting these data can be also used for unmixing purposes.

The first question we seek to answer is whether the process of endmember extraction can be enhanced by introducing a larger number of training examples. The quality of the unmixing procedure is measured by comparing the estimated  $\hat{\alpha}$  and the “actual” abundance vector  $\alpha$  in terms of the error defined by the Root Mean

Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{mp} \sum_{i=1}^p \|\alpha_i - \hat{\alpha}_i\|^2}, \quad (6.1.1)$$

where  $\alpha(i)$  and  $\hat{\alpha}(i)$  is the actual and the estimated abundance vectors of the  $i$ -th testing pixel, respectively. In Fig. 6.13 we observe that the error for all the unmixing chains reduces while increasing the number of training examples. The conclusion we can draw is that introducing more training examples can help spectral unmixing schemes to achieve better prediction by presenting them with a larger pool of data, facilitating the generative processes they employ. In specific, the SISAL method, which is not based on the pure pixel assumption achieves a lower RMSE, which is also characterized with a lower variance. In addition, the gradient-based algorithm assuming the PPNMM captures better the existing nonlinearities and leads to a better approximation of  $\alpha$  than SUnSAL, especially for a small number of training examples. A possible explanation of this behavior, following the reasoning in [17], is that the LMM assumption can be inappropriate for images containing sand, mineral mixtures, trees and vegetation areas [130], elements that are all contained in the selected labels (CLC codes 331, 131, 141, 223, and 241).

Given the best performing unmixing strategy, *i.e.*, SISAL for endmember extraction and the gradient-based algorithm for abundance estimation, we proceed to a comparison between spectral unmixing and multi-label classification versus the ensemble methods, utilizing multi-label classification metrics discussed in Section 3.6. Overall, Table 6.3 demonstrates that the multi-label methods are considerably better than the spectral unmixing ones in terms of the classification measures. Probably, the main reason for the bad behavior of unmixing algorithms, is that most of the endmembers are estimated to have a non-zero proportion (or else most of the labels are estimated to be active) in the testing pixels, producing too many FP examples, thus low precision. This can be attributed to the low spatial resolution and the high mixing of the pixels in our dataset. Regarding the performance of unmixing with respect to the selected threshold, Table 6.3 demonstrates that increasing the threshold leads to higher hamming error, and that it dramatically increases the recall, due to the fact that larger values of the threshold produce a larger number

Table 6.3: Performance (mean  $\pm$  std) of the ensemble methods versus unmixing over 30 random runs. Multi-label classification produces significantly higher results.

Measure	# Training	Spectral Unmixing		Multi-Label Classification	
		$T = 50\%$	$T = 95\%$	RAkEL-DT	ECC-DT
Hamming Loss $\downarrow$	128	$0.240 \pm 0.012$	$0.592 \pm 0.021$	$0.106 \pm 0.004$	$0.096 \pm 0.002$
	1024	$0.257 \pm 0.016$	$0.709 \pm 0.027$	$0.077 \pm 0.001$	$0.075 \pm 0.001$
Micro Precision $\uparrow$	128	$0.097 \pm 0.019$	$0.101 \pm 0.007$	$0.468 \pm 0.023$	$0.554 \pm 0.024$
	1024	$0.100 \pm 0.016$	$0.100 \pm 0.006$	$0.665 \pm 0.009$	$0.724 \pm 0.010$
Micro Recall $\uparrow$	128	$0.164 \pm 0.031$	$0.612 \pm 0.046$	$0.329 \pm 0.019$	$0.284 \pm 0.020$
	1024	$0.191 \pm 0.037$	$0.743 \pm 0.064$	$0.500 \pm 0.008$	$0.429 \pm 0.011$

of FP and a lower number of FN. With respect to the number of training examples, we observe that only the recall metric is increased suggesting that the architecture is able to capitalize on the training examples by identifying a larger portion of true labels.

A higher level snapshot of each method’s behavior can be obtained by comparing the hamming loss metric, which shows that the percentage of misclassified example-label pairs is much higher for unmixing compared to the ensemble multi-label learning algorithms. In general, the results presented in Table 6.3 demonstrate that multi-label classifiers, even if they are not able to produce fractional abundance estimations, achieve much higher and more robust binary predictions, even under such noisy environments.

### 6.1.8 Experimenting with fusion of the features

In real-world applications, the use of heterogeneous features is a practical issue. Nonetheless, how to integrate those features is still one of the main research topics in the areas of pattern recognition and machine learning. In this thesis, we have employed the *feature-level* fusion paradigm for the two kinds of features we take into account, meaning that we simply concatenate the features sets of NDVI and LST into a single feature vector. Besides this approach, one can train a classifier for each type of features, and then combine the predicted results produced by the different

classifiers (with the associated features) via a voting or averaging strategy to obtain the final output. This can be considered as *classifier* or *decision-level* fusion [131]. In this set of experiments, we experiment with some simple such fusion schemata.

More specifically, we initially learn two independent base-level models, one for each feature representation, and then we apply a decision rule in order to exploit the predictions of the base-level models. In this way, we utilize the bipartitions extracted from a multi-label classification algorithm (*e.g.*, RAKEL-DT, ECC-DT) using solely the 19 NDVI, or the 38 LST features. Given the matrices with the predictions, we apply either the Max, or the Min Rule [132] in order to obtain the final bipartition. In the first case (*i.e.*, Max Rule), we formulate the predictions regarding a label to be active as long as it is enabled in one of the classifiers, whereas in the second case (*i.e.*, Min Rule), a label is enabled to a test sample if only the corresponding values in both matrices agree.

Table 6.4: Performance (mean  $\pm$  std) of the ensemble methods using feature-level fusion and decision-level fusion with Max or Min Rule over 50 random runs, with 1024 training examples.

Measure \ Fusion Level	RAKEL-DT			ECC-DT		
	Feature	Decision - Max	Decision - Min	Feature	Decision - Max	Decision - Min
Hamming Loss $\downarrow$	$0.077 \pm 0.001$	$0.159 \pm 0.035$	$0.100 \pm 0.002$	$0.075 \pm 0.001$	$0.129 \pm 0.024$	$0.101 \pm 0.001$
Micro Precision $\uparrow$	$0.665 \pm 0.009$	$0.313 \pm 0.083$	$0.583 \pm 0.097$	$0.724 \pm 0.010$	$0.385 \pm 0.107$	$0.594 \pm 0.101$
Micro Recall $\uparrow$	$0.500 \pm 0.008$	$0.386 \pm 0.055$	$0.044 \pm 0.036$	$0.429 \pm 0.011$	$0.322 \pm 0.033$	$0.027 \pm 0.019$

In Table 6.4, we report three classification measures, regarding the feature-level and the decision-level fusion. We observe that the Min Rule is better than the Max Rule for hamming loss and precision metrics, whereas the Max Rule outweighs in recall, meaning that less FN arise in this case. This way, different rules are more appropriate for different kind of metrics. However, none of these schemata outperforms the baseline (*i.e.*, feature-level fusion), which is higher for all measures, suggesting that if we really want to benefit from decision-level fusion, we have to employ more complex approaches (*e.g.*, stacking). We mention that such simple fusion schemata have not led to improvements of performance also for other works in the past [88].

### 6.1.9 Parameter sensitivity analysis

In order to provide a comprehensive analysis, in this subsection we investigate the effects and influence in performance attributed to the parameter selection process of each considered algorithm. To minimize the effects introduced by other sources of variation, we fix the number of training examples again to 1024 and perform 50 independent trials for each case in order to obtain an informed view of the sensitivity of each method.

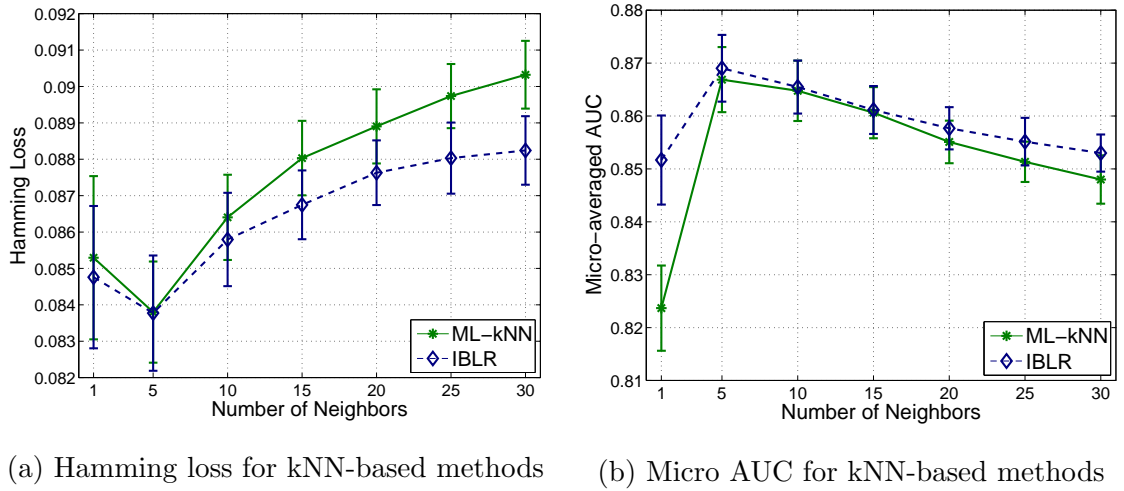
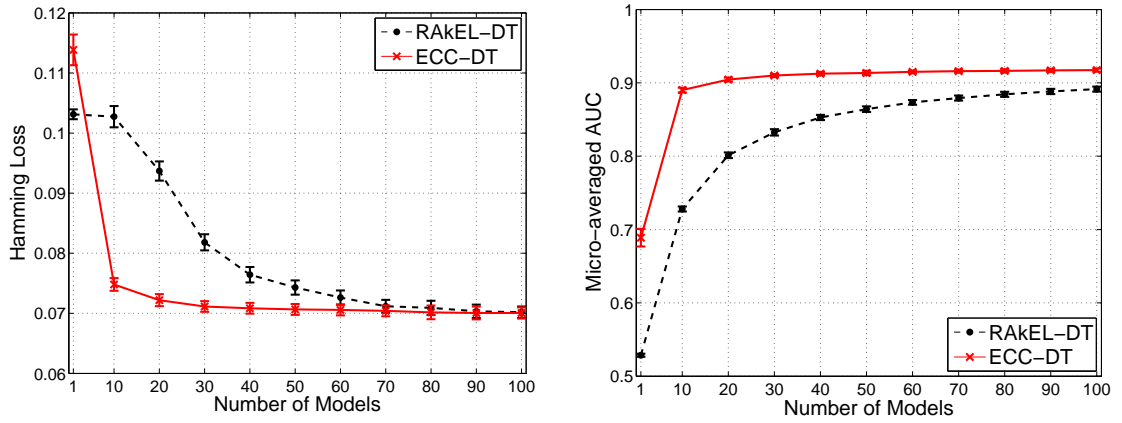


Figure 6.14: Performance with respect to the number of neighbors for h19v04 of CLC2000 with 1024 training examples. Both adaptation algorithms exhibit similar performance with respect to this parameter, however IBLR has a slightly higher and more robust behavior compared to ML-kNN.

Regarding the kNN-based methods, the key parameter that must be defined concerns the number of neighbors that are used during training phase. We observe in Fig. 6.14 that the worst choice for hamming loss corresponds to using 30 neighbors, while for AUC when selecting only one neighbor. Nevertheless, there is convergence between the two measures regarding the point of optimal performance, which is attained with 5 neighbors. Increasing further the size of the neighborhood leads to performance deterioration, since less valuable information and more confusion due to the noise is obtained. Between the two classifiers, we note that IBLR has steadily a better predictable behavior than ML-kNN.

Considering the powerful class of ensemble techniques, we investigate how the



(a) Hamming loss for ensemble methods.

(b) Micro AUC for ensemble methods.

Figure 6.15: Classification performance with respect to the number of models for h19v04 of CLC2000 with 1024 training examples. Varying the number of models has a major effect on the classification performance for ensemble methods. In this respect, ECC-DT exhibits a superiority compared to RAKEL-DT for limited number of training examples.

number of component base classifiers involved in the chain affects the performance. As illustrated in Fig. 6.15, ECC-DT and RAKEL-DT differ significantly in internal design, since the former achieves a performance close its optimal with a small number of classifier chain models, whereas RAKEL-DT is learning progressively with an increased number of models. With the adoption of a large number of models (more than 60), we can observe that the RAKEL-DT approximates the performance achieved by ECC-DT. We note that for a fixed number of models, the complexity of ECC-DT is significantly higher than RAKEL-DT. However, the superiority in performance of ECC-DT for multi-label classification of land cover data is particularly evident when a small number of training examples is incorporated.

## 6.2 Feature Learning Evaluation

The proposed framework of multi-label annotation for remote sensing data is a flexible scheme. Therefore, the procedure described in Chapter 5 can be also adopted successfully for other types of data. In this section, we incorporate surface re-

reflectance data provided by the MODIS product MOD09A1, and associate them with the h19v04 tile of CLC2000 (using the 20 labels of Table 5.1). More specifically, we consider 7 surface reflectance bands, acquired at 500m<sup>2</sup> spatial resolution and having a temporal variability of 8 days. Similar to MOD13A1 and MOD11A2 products, we collect all the available data over a temporal window of six months (May to October), leading to 161 spectral bands in total. We underline that we are particularly interested in this feature set, since these are the data which are provided directly from a satellite imaging system and thus can be obtained and be accessible in short time, without the need of extra processing. Furthermore, we incorporate only 1000 training examples in order to obtain a real-life and challenging scenario.

In the first experiments, we compare these low-level satellite data with the high-quality data used so far (*i.e.*, NDVI and LST) in order to show their differences in performance under the same classification conditions. Then, we perform a pre-processing step through SAEs framework, in order to learn a new more informative representation, and examine the performance of the learned features under the following scenarios:

- classification performance with respect to layer size and normalization.
- classification performance with the exploitation of a second hidden layer (deep learning), where we utilize either the last layer’s features, or the concatenation of features learned in the first and the second layer.
- comparison of undercomplete models with typical dimensionality reduction techniques, such as PCA.

Once again, each experiment has its own distinct value and purpose, while the objective this time is to examine the benefits of learning automatically low-level features for the defined multi-label classification scenario. We further notice that deep learning features can be generally extracted by providing primitive data to a system, since by incorporating hand-crafted features the hierarchical structure of the data needed is lost due to their inherent complex makeup. This way, surface reflectance data are applicable for the deep learning approach and theoretically able to reveal extra valuable underlying information, as they indeed do.



### 6.2.1 Data preprocessing

A critical aspect of SAE models is the need for data normalization. To that end, several normalization steps are usually performed in order to adapt the raw data into appropriate inputs for neural networks. Experimental results have shown that when the input variables are close to zero, neural network training is typically more efficient since convergence is faster and the likelihood of getting stuck in local optima is reduced. We consider normalization of each feature vector  $j$  to  $[0, 1]$  by subtracting the minimum value of each element and dividing it by its range (the difference between the maximum and the minimum value):

$$x_i^j = \frac{x_i^j - \min^j}{\max^j - \min^j}, \quad (6.2.2)$$

where the minimum values and ranges are stored for use in the test set.

### 6.2.2 Network architecture

In order to train a deep neural network there are several hyperparameters which need to be set, including those which specify the structure of the network itself and those which determine how the network is trained. The type of the nonlinearity in the activation function is one of the first such hyperparameters that needs to be considered. We adopt the logistic sigmoid activation  $\alpha_f(\phi) = \alpha_g(\phi) = \sigma(\phi) = 1/(1+e^{-\phi})$  in the hidden layers which has an output range in the interval  $[0,1]$  (and is in accordance to the the initial scaling from Eq. 6.2.2). The bias units are initialized to zero, whereas the initial weights are randomly drawn from a uniform distribution  $U(-\epsilon, \epsilon)$  with  $\epsilon = 4\sqrt{6/(\text{fan-in} + \text{fan-out})}$ , where fan-in is the size of the previous layer and fan-out the number of hidden units in current layer [133]. Tied weights ( $\mathbf{W}_2 = \mathbf{W}_1^\top$ ) are commonly used to reduce the complexity, yet untied ( $\mathbf{W}_2 \neq \mathbf{W}_1^\top$ ) weights seem to generalize better in our case. Therefore, in the following results untied weights are employed in all layers.

Neural network models demand significant effort and time during training, making an exhaustive grid search in the space of hyperparameters intractable. In addition, since the particular dataset we consider has not been explored before, no prior information on where these hyperparameters approximately lie is available. As such,

for the specification of the hyperparameters  $\rho$  and  $\beta$  which control the sparseness of the autoencoder, we first performed a coarse grid search in reasonable values and in all cases, model selection was performed according the minimum Jaccard coefficient in the validation set, which is composed of 20% of the training data (randomly sampled). More specifically, the grid is constructed by considering the set produced by the Cartesian product of  $\rho \in \{0.001, 0.01, 0.1, 0.5, 0.9\}$  and  $\beta \in \{1, 3, 5, 7, 9\}$  values. A more fine-grained search in the vicinity of that tuple  $(\rho, \beta)$  that produced the best score was subsequently considered. The models were trained for 5000 unsupervised learning epochs, while at the supervised learning stage, we use 3000 epochs with early stopping, a typical approach to prevent overfitting [133], where we monitor the validation error every 100 iterations and if it has not decreased for 500 consecutive epochs, early stopping is enabled. Reported results are averaged over 10 Monte-Carlo trials, in order to minimize the effects of the initial random seed. For the implementation of the SAE we considered the framework described in [117] which is also available online<sup>3</sup>. The optimization algorithm used for minimizing the cost function of the SAE was the BFGS gradient descent method with limited-memory variation (L-BFGS)<sup>4</sup> and a stopping criterion of  $10^{-8}$ , a quasi-Newton method for unconstrained optimization that has proved to work well. Concerning the base classifier, for the all following experiments we have used the SVM, which is solved with linear kernel by the Sequential Minimal Optimization (SMO) algorithm that is available within WEKA. This choice has to do with the empirical observation that profits from feature learning are more pronounced for classifiers such as SVM and Logistic Regression, compared to the DT that was used for the experiments of Section 6.1.

### 6.2.3 Performance of raw and high quality features

In land cover classification, vegetation indices provide a stronger indicator of the amount of the photosynthetically active green biomass, than the pure spectral signatures [11]. This way, they enjoy a widespread popularity for many years. In

<sup>3</sup><http://deeplearning.stanford.edu/wiki/index.php/UFLDL-Tutorial>

<sup>4</sup>Mark Schmidt's implementation: <http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html>

Table 6.5: Impact of the quality of features for the classifiers. Higher quality features composed of NDVI and LST yield to improved performance compared to raw surface reflectance.

Algorithm	Features	Hamming Loss ↓	Avg Precision ↑	Mac-F <sub>1</sub> ↑	Mac-AUC ↑	Mic-F <sub>1</sub> ↑	Mic-AUC ↑
RAkEL-SVM	NDVI-LST	$0.086 \pm 0.000$	$0.493 \pm 0.000$	$0.224 \pm 0.000$	$0.615 \pm 0.000$	$0.420 \pm 0.000$	$0.676 \pm 0.000$
	Surf. Refl.	$0.087 \pm 0.000$	$0.435 \pm 0.000$	$0.157 \pm 0.000$	$0.572 \pm 0.000$	$0.367 \pm 0.000$	$0.647 \pm 0.000$
	Norm. Surf. Refl.	$0.084 \pm 0.000$	$0.472 \pm 0.000$	$0.175 \pm 0.000$	$0.586 \pm 0.000$	$0.423 \pm 0.000$	$0.667 \pm 0.000$
ECC-SVM	NDVI-LST	$0.087 \pm 0.000$	$0.594 \pm 0.003$	$0.275 \pm 0.005$	$0.712 \pm 0.006$	$0.474 \pm 0.003$	$0.814 \pm 0.003$
	Surf. Refl.	$0.087 \pm 0.000$	$0.551 \pm 0.004$	$0.191 \pm 0.007$	$0.679 \pm 0.008$	$0.449 \pm 0.007$	$0.794 \pm 0.005$
	Norm. Surf. Refl.	$0.085 \pm 0.000$	$0.593 \pm 0.004$	$0.255 \pm 0.006$	$0.707 \pm 0.005$	$0.486 \pm 0.004$	$0.816 \pm 0.003$

order to obtain a clear understanding regarding the effects of the quality of the features, Table 6.5 presents the performance for different types of input features for both multi-label classifiers considered in this paper. In the table, the first row for each classifier case, corresponds to the optimized hand-crafted features, the second to spectral reflectance values, and the third row to normalized spectral reflectance. The higher quality feature descriptors we consider are the NDVI and the Land Surface Temperature (LST) on the same tile for the same months retrieved from MODIS Terra. The combination of NDVI and LST time-series features is well established in the literature as being a very good indicator which can quantify changes in the representation of vegetation growth and physical characteristics of land cover in general [134].

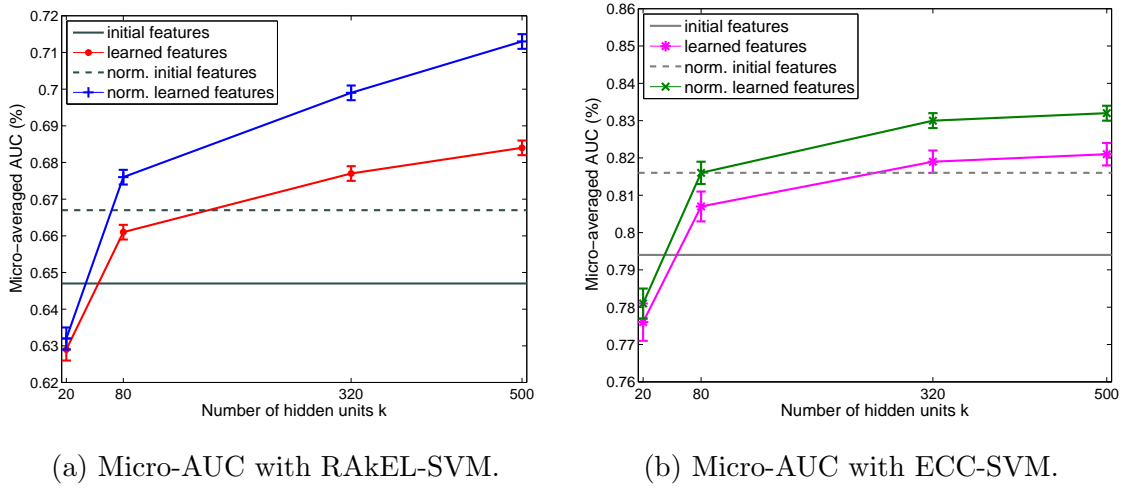
The results show that the feature-level fusion of NDVI and standardized LST acts indeed better than raw surface reflectance. This higher performance achieved by the carefully designed features is the main motivation behind our approach, which aims at formalizing an automated process able to extract more meaningful spectral characteristics from raw satellite data. Moreover, we can see that there is a significant effect of normalization on surface reflectance features, which varies from metric to metric. We emphasize that even if RAkEL and ECC are two of the most powerful schemas for multi-label classification, they perform poorly for some of the measures [79], demonstrating the dramatic challenges associated with the real-world problem we consider in this work.

### 6.2.4 Impact of layer size and normalization

We now move on to our characterization of performance on various axes of parameters, starting with the effect of layer size. Choosing the number of neurons in the hidden layers is a crucial design parameter affecting the overall neural network architecture. No formal rule guaranteeing the optimal selection exists and thus it usually comes down to trial and error. We study feature learning with both undercomplete ( $k < d$ ) and overcomplete ( $k > d$ ) representation models, where  $d = 161$  is the dimensionality of the raw feature space, considering a single-layer architecture as a baseline. More precisely, in the case of undercomplete representations, we perform experiments with half of the initial features,  $k = 80$ , and then with an extreme scenario of only  $k = 20$  hidden units. Correspondingly, in the case of overcomplete representations, we double the capacity of the model in breadth,  $k = 320$ , and then we experiment on a highly overcomplete model where the latent code has  $k = 500$  units.

Figure 6.16 presents the micro-AUC score with respect to the number of hidden units used in shallow architectures, where more specifically, Figure 6.16a demonstrates the classification performance of the RAkEL with SVM as base classifier (RAkEL-SVM), whereas Figure 6.16b introduces SVM as part of an Ensemble Classifier Chain (ECC-SVM). Regarding the baselines, the solid horizontal lines correspond to the performance of the classifiers using the initial raw features, whereas the dotted lines correspond to the normalized version of the initial raw features. An observation evident in both figures is the large gap between the solid and the dotted lines suggesting that the initial normalization step of the surface reflectance data before their introduction to the classifiers can have a dramatic impact on the performance. This is in line with the fact that algorithms that work with distances and make parametric assumptions regarding the distribution of the data, such as SVM or logistic regression, are affected positively by a normalized input space in general. SVM assumes also that the data it works with lie in a standard range, thus the normalization of feature vectors is crucial. We should also notice that the computational time is much smaller with the use of normalization.

Overall, for both classifiers, 20 units are too few to adequately encode the signals



(a) Micro-AUC with RAkEL-SVM.

(b) Micro-AUC with ECC-SVM.

Figure 6.16: Performance of multi-label classifiers using the codes learned by increasing the size  $k$  of one hidden layer. The solid horizontal lines correspond to the performance of the classifiers with the initial raw features, whereas the dashed lines correspond to the accuracy achieved with the normalized version of the raw features. Optimal complexity-performance ratio is achieved using twice as many of the initial raw features.

in the hidden layer resulting in significant degradation performance. By increasing the number of hidden units to 80, the performance in both schemes surpasses the score achieved using raw un-normalized features as inputs. However, the gain offered by this feature learning, is outweighed to some extent by the effort of normalization of the raw input data, as indicated by the dotted line. On the contrary, by considering 320 units, the performance of the feature learning scheme increases and slightly surpasses the baseline, whereas with the incorporation of 500 units, the improvement is marginal and comes at a higher computational cost. A key observation point is that the normalization after the feature-mapping can also play a significant role and boost the performance of classifiers. In detail, we observe that in the case of the undercomplete feature learning architectures, the micro-AUC does not significantly change with or without this normalization step. Nonetheless, it is evident that in the case of overcomplete systems, the performance is higher and can clearly outperform the enhanced baseline versions with the normalized feature vectors. Note that different hyperparameters are needed in the overcomplete case whether we use

normalization or not for optimal performance.

With respect to the different classifiers, one can easily notice the dominance of the ECC scheme compared to the RAkEL approach. Moreover, ECC is less affected by the normalization steps, but has a greater variance on the results. Last, we have to mention that we need the contribution of such powerful ensemble multi-label learning schemes in order to achieve reasonable performance, due to both the limited training examples and the many factors of variation that inhere in our real dataset, allowing us to test the limits of current state-of-the-art classifiers.

### 6.2.5 Impact of depth

In this set of experiments, we focus on the impact of depth, *i.e.*, the number of hidden layers, with respect to the classification performance. In our setup, we employ the same number of hidden units for all layers, which has been suggested that generally leads to better performance compared to decreasing (pyramid) or increasing (inverted pyramid) network architectures [70, 133].

Table 6.6: Impact of depth for a fixed architecture consisting of 320 hidden units per layer. Higher results are obtained for features extracted from deep architectures.

Algorithm	Depth	Hamming Loss ↓	Avg Precision ↑	Mac-F <sub>1</sub> ↑	Mac-AUC ↑	Mic-F <sub>1</sub> ↑	Mic-AUC ↑
RAkEL-SVM	1	0.081 ± 0.000	0.521 ± 0.003	0.265 ± 0.004	0.620 ± 0.002	0.475 ± 0.004	0.699 ± 0.002
	2	0.082 ± 0.000	0.553 ± 0.003	0.330 ± 0.005	0.661 ± 0.003	0.504 ± 0.004	0.731 ± 0.002
	1 & 2	0.082 ± 0.000	0.568 ± 0.004	0.360 ± 0.004	0.676 ± 0.003	0.518 ± 0.003	0.743 ± 0.003
ECC-SVM	1	0.084 ± 0.000	0.623 ± 0.003	0.330 ± 0.007	0.732 ± 0.003	0.521 ± 0.003	0.830 ± 0.002
	2	0.087 ± 0.001	0.628 ± 0.003	0.377 ± 0.005	0.748 ± 0.003	0.530 ± 0.004	0.832 ± 0.002
	1 & 2	0.086 ± 0.000	0.635 ± 0.003	0.397 ± 0.005	0.757 ± 0.003	0.539 ± 0.003	0.835 ± 0.003

Table 6.6, provides a comprehensive numerical evaluation of the two classification schemes, namely RAkEL-SVM and ECC-SVM under different evaluation metrics. The experiments concern the features extracted from the feature learning system; either from a single-layer autoencoder (rows indicated with Depth 1), or a level-2 stacked autoencoder which obeys the properties of deep learning (rows indicated with Depth 2). The results demonstrate that both RAkEL-SVM and ECC-SVM can benefit from the additional hidden layer to gain extra valuable discriminative

information. Regarding the depth of the network, the gain is significant for all metrics except hamming loss, which improves only for the first hidden layer. We noticed also that the mean value of the cost function is smaller from the first to the second hidden layer, which can serve as a proxy of the final system’s performance. In addition, we have also considered the “concatenated” representation for autoencoders (rows indicated with Depth 1 & 2 in Table 6.6), where we utilize the concatenation of both layers of the network. This way, the final features introduced to the classifier correspond to the combination of the first and the second hidden layer, instead of the traditional “replacement-based” representation, where only the top-layer features are used. We observe that the model can take further advantage from this kind of representation and the more features exhibiting an improved performance, but in a higher computational cost.

We have to highlight that a sparser representation has to be enforced for the second than the first hidden layer, which suggests that in this case the sparseness property in the representation can indeed help overcomplete architectures, since without the use of this type of regularization, the deep models cannot achieve performance beyond the one achieved by a single-layer architecture. Furthermore, the performance achieved with deep learning of 320 units is better compared to the single-layer case where we have 500 hidden units, further promoting the motivation for deep architectures. Finally, when the feature learning procedure is involved, the performance is substantially higher for all measures compared to the surface reflectance baselines and the higher quality features (NDVI–LST) shown in Table 6.5. In a nutshell, these results suggest that to really benefit from sparse overcomplete models and produce useful representations, one must consider departing from shallow to deep learning architectures.

Figure 6.17 shows the evolution of performance as we increase the number of hidden layers from the first to the second with and without the use of pretraining via SAEs. We use the same set of hyperparameters for both models. The performance of the models with unsupervised pretraining is higher, whereas the advantage is more pronounced in deep architectures. In parallel, the pretraining procedure clearly reduces the variance of the performance, leading to more robust results. All in all,

RAkEL model seems to be more affected from the pretraining procedure, as well as to benefit more from the second hidden layer in comparison to the ECC.

### 6.2.6 Visualization of results

The results can be again interpreted visually through the notion of the multi-label confidence maps presented in Subsection 6.1.3. This time, the goal is to reveal the differences in classification performance arising from the use of different features. More specifically, firstly we show the results from the classification performance with the initial (raw) features, then with level-1 learned features through SAEs, and finally with level-2 learned features through stacked SAEs. The ground-truth map used for this set of experiments is presented in Fig. 6.18, whereas as top layer classifier we incorporate either the RAkEL-SVM (Fig. 6.19), or the ECC-SVM (Fig. 6.20).

Examining the ground-truth map of Fig. 6.18 very closely, we observe that there are labels of high frequency (*e.g.*, 2, 12, 13, 14, 16, 17), but most of the labels arise sparsely (*e.g.*, 1, 3 – 11, 18 – 20). When the algorithms utilize low-level raw features (Figures 6.19a and 6.20a), they can recognise only the labels of a higher frequency in the samples. Nevertheless, through the hierarchical learning process, they can gradually recognise successfully more examples having labels of a lower frequency. In other words, labels such as 10, 11, and 19, as well as 5, 8, 9, and 20, can only be revealed through the feature learning process for RAkEL-SVM and ECC-SVM, respectively. In this way, we can visually verify that the performance improves while increasing the number of hidden layers. Another fact that is particularly evident through the maps is the superiority of ECC-SVM, since through RAkEL-SVM labels such as 5, 8, 9, and 20, cannot be recognised successfully (except for the second layer features).

### 6.2.7 Comparison with PCA

In the specific case where the autoencoder has a hidden layer with a smaller hidden size than the input, the neural network is forced to represent the input in a lower



dimensional space, *i.e.*, compress it. In this set of experiments, we focus on such cases, trying to compare undercomplete models produced via SAEs, with PCA, a benchmark technique for dimensionality reduction which also follows the unsupervised learning framework, but with a linear mapping.

Figure 6.21 presents how PCA and SAEs behave as multi-label dimensionality reduction techniques with respect to the number of hidden units and the number of principal components, respectively, implying the final dimensionality size  $k$  of the lower-dimensional space. The considered evaluation performance metrics are the hamming loss and the micro-averaged AUC, while the classification scheme applied after the dimensionality reduction procedure is the ECC-SVM. In specific, we perform experiments according to an extreme scenario where we set the dimensionality to  $k = 5$ , and proceed until we reach half of the original feature space ( $k = 80$ ). We observe that single-layer SAEs achieve higher performance for both metrics, meaning that the underlying data are really complex with relationships that a linear transformation cannot efficiently capture. The biggest difference is noticed when experimenting with extreme cases of limited feature size. However, the gain through autencoders is inadequate to strongly suggest the use of autoencoders in any case, particularly if one considers the computational complexity which is significantly higher with autoencoders. We mention that this behavior has been also observed in other real-world applications. In other words, despite the theoretical advantages of nonlinear reduction algorithms, it has been commonly reported that they are not capable of outperforming the conventional linear algorithms on many real-world datasets, although they perform very well on artificial ones [135].

### 6.2.8 Model sensitivity

In this part, we investigate the sensitivity of the feature learning scheme with respect to the sparsity parameter  $\rho$ . More specifically, Figure 6.22a demonstrates the micro-averaged AUC score for the ECC-SVM classifier versus some representative values of the  $\rho$ , for a fixed sparsity weight ( $\beta = 1$ ). We observe that the undercomplete models are more sensitive to the hyperparameter settings, since a change for  $\rho$  entails a dramatic change for the performance. From the other side, the overcomplete

models seem more robust for different values of the sparsity parameter, but do not benefit a lot from the regularization. Overall, the hyperparameter  $\rho$  can highly affect the final performance, whereas the impact of regularization is more prominent in the undercomplete case for a single-layer architecture. In any case, as illustrated by the figure, different hyperparameters combinations can lie on a wide range, indicating that SAEs are quite sensitive models and thus hyperparameters settings have to be chosen very carefully.

Figure 6.22b investigates the impact of the number of hidden units with respect to the generalization performance of SAEs as it is encoded in the cost function. We observe that the system seems to be primarily affected by the number of hidden units compared to the sparsity of the connections. By increasing the number of hidden units, the autoencoder ends-up learning a very good approximation of the identity, but the specific regularization technique does not provide much additional interpretation of the data in order to boost the performance of the subsequent classification algorithm. Intuitively, this means that for very sparse models (large value of  $\beta$  and small value of  $\rho$ ), the algorithm tends to learn very specific features that classifiers are not capable of generalizing, thus achieving better classification rates.

ged AUC (%)

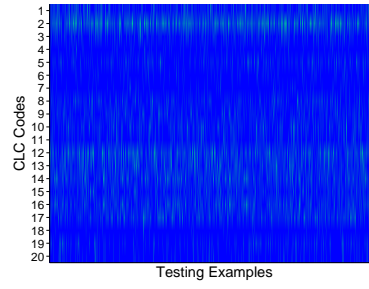


Figure 6.18: Ground-truth multi-label map for h19v04 of CLC2000 corresponding to a binary matrix indicating which labels are active for each example, *i.e.*, spatial location. The vertical axis corresponds to specific label as illustrating in Table 5.1, while horizontal axis to specific testing example out of the 3000.

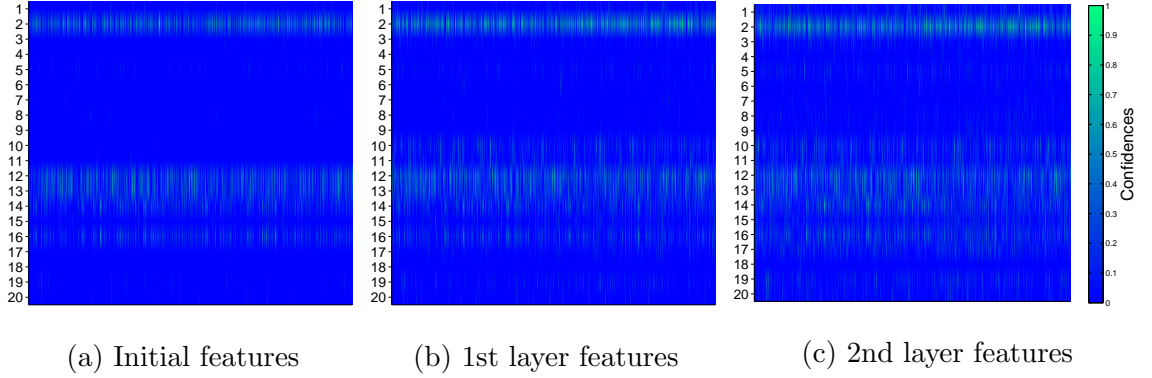


Figure 6.19: Multi-label confidence maps with RAKEL-SVM as top layer classifier incorporating different levels of features.

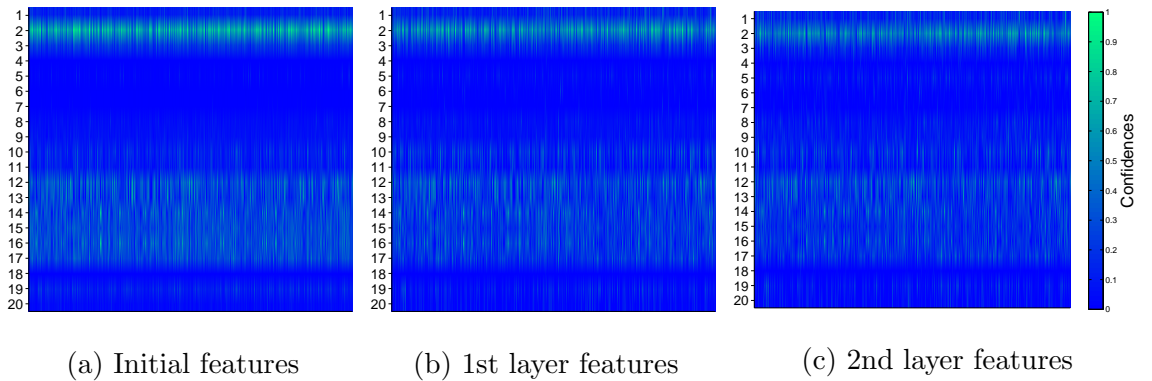


Figure 6.20: Multi-label confidence maps with ECC-SVM as top layer classifier incorporating different levels of features.

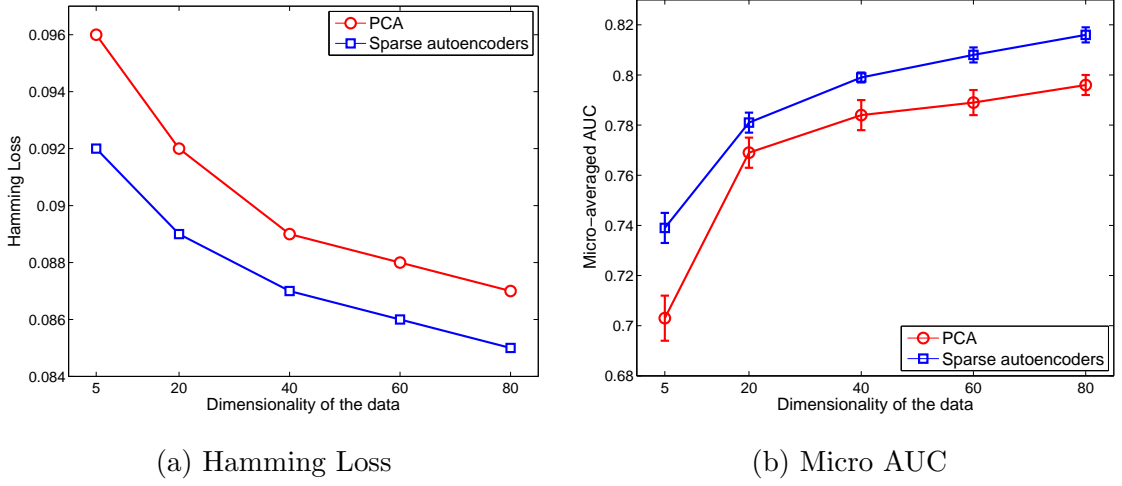


Figure 6.21: Comparison of dimensionality reduction methods with PCA and SAEs for multi-label data using ECC-SVM. SAEs perform higher but have also higher computational complexity.

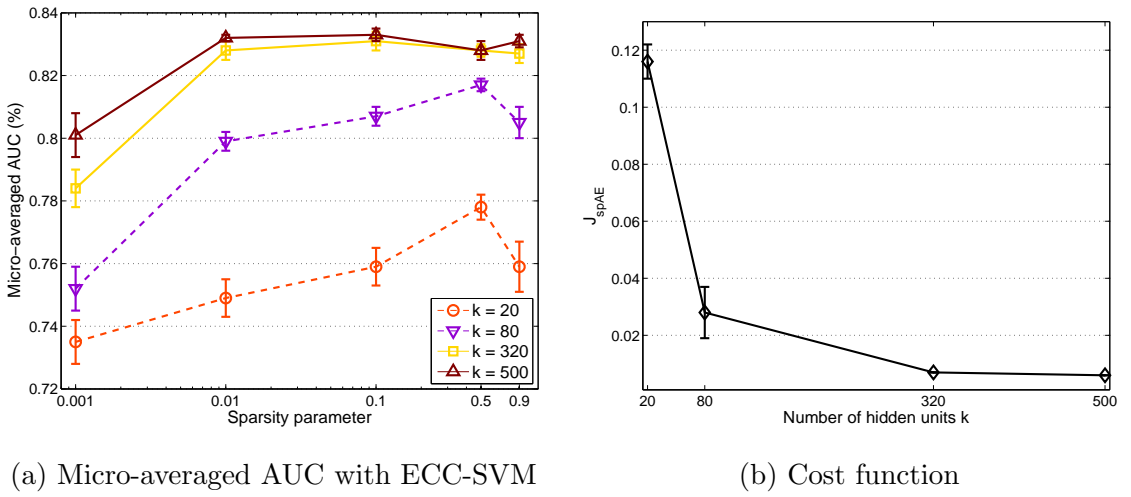


Figure 6.22: Sensitivity of the SAE model for the single-layer case. Sparsity parameter  $\rho$  plays an important role to the final performance for a fixed sparsity weight (left), whereas the value of the cost function reduces primarily due to the size of the hidden layer (right).

# Chapter 7

## Conclusions and Future Work

In this thesis, we presented a radically different approach in satellite-based land cover identification, where we cast the problem as an instance of multi-label classification. Multi-label classification in this specific domain provides supplementary solutions to the important problem of spectral unmixing, however, unlike state-of-the-art schemes, the proposed formulation utilizes publicly available labels in conjunction with contemporary satellite data, and provides a real-world answer to maintaining up-to-date land cover maps. Furthermore, the proposed scheme provides potential modalities for extensions to various sources of data, in addition to land cover and multispectral examined in this thesis.

We considered an extensive set of experiments, employing state-of-the-art multi-label learning algorithms under diverse and challenging scenarios. The experimental results suggest that a small number of training examples is sufficient for achieving satisfying performance in the situation where training and testing data from a specified region on a given time is considered. However, the results deteriorate when testing takes place on a different spatial region or from another instance in time. We demonstrate that by encompassing a limited number of examples of the target-tile at the target-time, the performance improves remarkably, offering a solid answer to the issues related to the cost and time required for gathering annotated ground-truth data. It should be noted that the proposed formulation can fully exploit the existence of ground-truth data, which means in parallel that this approach cannot be applied in cases where labeled data are unavailable, *e.g.*, unmixing of the Mars

surface. An interesting direction for future work is to consider sophisticated techniques in order to reduce the effect of the unavoidable disturbances in the time-series imagery, such as those presented in [136]. Apart from this, the enrichment of the training set with measurements from other (compatible) instruments, can be considered as a particularly appealing aspect to improve the predictions regarding another spatial tile or temporal instance.

The thesis also focused on the effects that the quality of satellite data representation can have on a learning algorithm, an issue of extreme importance. Carefully designed hand-engineered features can significantly aid in the more discriminative representation of the remote sensing data, such as multispectral images employed in our case. However, the specificity of these features may limit their generalization capacity to different data sources and learning objectives. To address this issue, we propose the introduction of feature learning directly from data. Results presented in this thesis suggest that feature learning, in our case SAE networks, can significantly boost the performance, even when a single hidden layer is involved. Furthermore, experiments indicate that stacking layers over the raw input data can further improve the performance leading to state-of-the-art performance in solving a truly hard learning problem including real data that exhibit many facets of variation. Future directions include experimenting with other types of regularization for the autoencoders, as well as extending this work to consider the nature of time-series and spatial coherence, in order to better exploit the temporal and spatial characteristics of the remotely sensed features, respectively. Finally, a supervised finetuning with backpropagation to the whole architecture, can be also adopted to greatly improve the performance, but at a significantly higher execution time.

In addition to the value of this work in the remote sensing community, we have also effectively produced a new class of datasets composed of satellite and geographic data. Therefore, we offer the research community the possibility to evaluate different multi-label classification schemes on alternative remote sensing datasets which provide a more appropriate and beneficial formulation compared to the single-label models that have been explored in the literature so far.

# Appendix A

## Working with MODIS Reprojection Tool

The MRT<sup>1</sup> tool was developed to support individual work with higher level MODIS Land products (Levels 2G, 3, and 4) [119]. MODIS data products are free and distributed in HDF-Earth Observation System (HDF-EOS) format. These products are projected to a global tiled-based SIN grid formulation, where each tile is approximately 10° latitude<sup>2</sup> by 10° longitude<sup>3</sup> and non overlapping (Fig. A.1b). MRT accepts raw binary or tiled MODIS products in HDF-EOS format, whereas output file formats include raw binary, HDF-EOS, and GeoTIFF. The tool is compiled for use on multiple operating systems aiming to provide map projections, resampling, format conversions, mosaicing, and spatio-spectral subsetting options. Note that MRT extracts automatically useful information once an HDF-EOS file is loaded, displaying basic file specifications, such as the number of available bands, and the geographic coordinates of the rectangle.

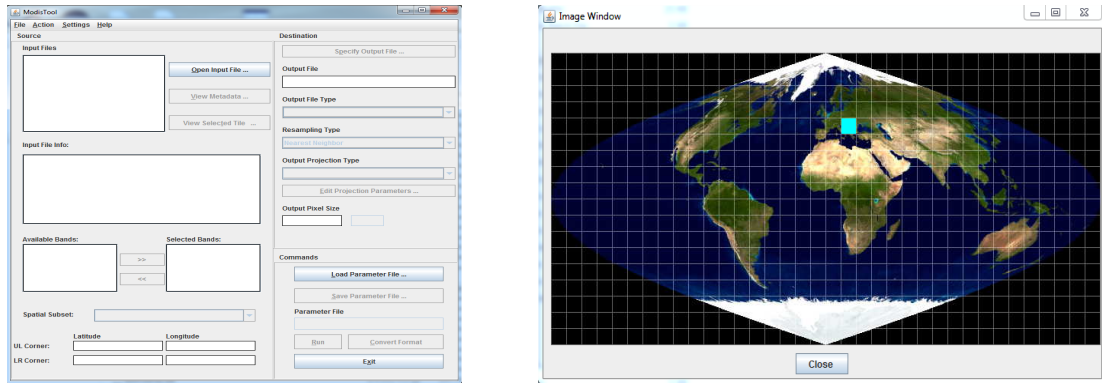
---

<sup>1</sup>[https://lpdaac.usgs.gov/tools/modis\\_reprojection\\_tool](https://lpdaac.usgs.gov/tools/modis_reprojection_tool)

<sup>2</sup>Latitude is the angular distance measured north or south in degrees from the equator. By convention, the equator is 0°, the north rotational pole is 90°N, and the south rotational pole is 90°S. Therefore, assuming a north facing map, lines of latitude are horizontal increasing from left to right (east to west).

<sup>3</sup>Longitude is the angular distance measured east or west in degrees from the prime meridian at Greenwich, England, to a maximum of 180°. Longitude lines run up and down increasing from down (south pole) to up (north pole), assuming a map with the north on top.





(a) MRT primary interface invoked from Graphical User Interface (GUI). (b) The View Selected Tile popup of MRT illustrating the h19v04 tile.

Figure A.1: Snapshots of the MRT tool.

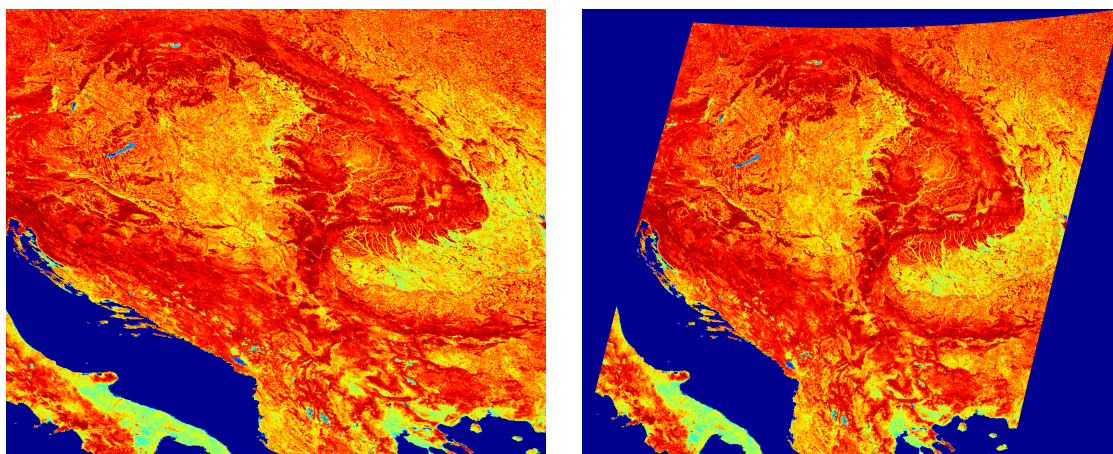
We provide the MRT with HDF files of distinct tiles referring to a specific time instance of the defined time series. Each file is subject to the following processing steps. We resample the original data according to Nearest Neighbor in order to ensure that we do not average quality bits. Then, we change the projection (*i.e.* how the 3D Earth's curved surface is mapped on a 2D planar surface) to UTM<sup>4</sup>, and edit the projection parameters to WGS 84 datum<sup>5</sup>, a standard Coordinate Reference System (CRS)<sup>6</sup> for the Earth adopted globally in cartography, geodesy, and navigation by the GPS<sup>7</sup>. The output is forced to be again an HDF file. In Fig. A.2a we show the h19v04 tile in SIN projection, whereas Fig. A.2b depicts the same tile after the aforementioned processing steps.

<sup>4</sup>UTM is a map projection which uses a 2D Cartesian coordinate system to give locations on the surface of the Earth, dividing it into 60 Zones of the same size. UTM is based on the original cylindrical Transverse Mercator projection.

<sup>5</sup>Geodetic datum (or geodetic system) is a coordinate system (mathematical model) with a reference surface, used to fit the Earth to an ellipsoid. For example, WGS 84 datum is a standard spheroidal reference surface (ellipsoid) for raw altitude data, which uses a gravitational equipotential surface (the geoid) in order to estimate the nominal sea level, and thus define a place.

<sup>6</sup>[http://docs.oracle.com/cd/B28359\\_01/appdev.111/b28400/sdo\\_cs\\_concepts.htm](http://docs.oracle.com/cd/B28359_01/appdev.111/b28400/sdo_cs_concepts.htm)

<sup>7</sup>The Earth is an imperfect ellipsoid, thus localised datums can give a more accurate representation of the area of coverage than WGS 84. Nevertheless, the benefits of a global system outweigh the greater accuracy.



(a) The original HDF-EOS file representing the NDVI bands of the h19v04 tile in SIN projection.

(b) The processed HDF-EOS file representing the NDVI bands of the h19v04 tile in UTM projection with a WGS 84 datum.

Figure A.2: Visual difference between the SIN and the UTM projection for a specific spatial tile.



# Appendix B

## Processing Geographic Information Systems Data

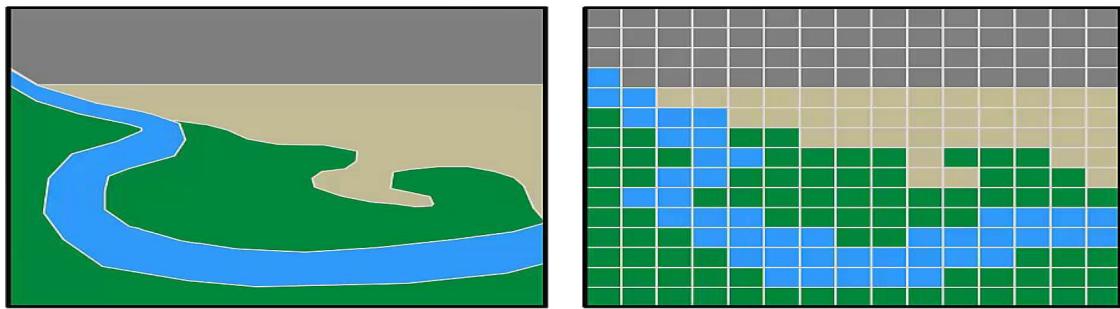
GIS primarily use two categories of digital data: *vector* and *raster* data format. Vector data model is a representation of the world using individual points, which (for 2D data) are stored as pairs of (x,y) coordinates. These points may be joined in a particular order to create lines, or joined into closed rings to create polygons. Vector models are useful for storing data that have discrete boundaries, such as country borders, land parcels, and streets. From the other side, raster data model is a representation of the world as a surface divided into a regular grid of (usually square) pixels (or cells), where each pixel has an associated value. Raster models are useful for storing data that vary continuously, as in an aerial photograph, a satellite image, a surface of chemical concentrations, or an elevation above sea level. All in all, vector and raster datasets have different strengths and weaknesses, whereas a representation through vector and raster format of the same image is presented in Fig. B.1.

CLC data 2000<sup>1</sup> and 2006<sup>2</sup> Version 17 are distributed by the EEA in raster format. Since raster data in GIS are good for showing continually varying information, they highly apply for land cover estimation which changes over time. The data are

---

<sup>1</sup><http://www.eea.europa.eu/data-and-maps/data/corine-land-cover-2000-raster-3>

<sup>2</sup><http://www.eea.europa.eu/data-and-maps/data/corine-land-cover-2006-raster-3>



(a) Vector data are focused on modelling discrete features with precise shapes and boundaries. (b) Raster data are focused on modelling continuous phenomena and images of the Earth.

Figure B.1: The raster and vector data models. Each model stores features in a different way.

actually matrices of discrete cells that represent features (land cover type) on the Earth's surface. Each cell in the raster grid is the same size and constitutes a pixel. The size of pixels in a raster determines its spatial resolution, *i.e.*, the ability to resolve two points as separate in an image, which in the case of the CLC maps is either  $100 \times 100$  or  $250 \times 250$  meters.

By checking the online metadata of the inventories provided by EEA, we see that the grid included in the GeoTIFF file is based on the CRS of the European Terrestrial Reference System (ETRS) projection 89 (ETRS89)<sup>3</sup>/ETRS - Lambert Azimuthal Equal Area (LAEA), which is also known in the European Petroleum Survey Group (EPSG) Geodetic Parameter Dataset under the identifier EPSG:3035<sup>4</sup>. The .zip files, apart from the GeoTIFF file, include a layer (.lyr) and a projection (.prj) file, as well as the CLC legend in Excel (.xls) and text (.txt) file formats. Note that CLC2000 and CLC2006 do not have the same geographic coverage according the metadata, since some of the countries did not provide data for the 2006 update.

QGIS<sup>5</sup> is a free and open-source GIS application, which can visualize, manage, edit, and analyse data, as well as compose printable maps. It can run under different

<sup>3</sup>The ETRS89 datum is used in Europe, as the NAD83 datum is used in North America. These datums are almost identical to WGS 84.

<sup>4</sup><http://epsg-registry.org/>

<sup>5</sup><http://www.qgis.org/en/site/>

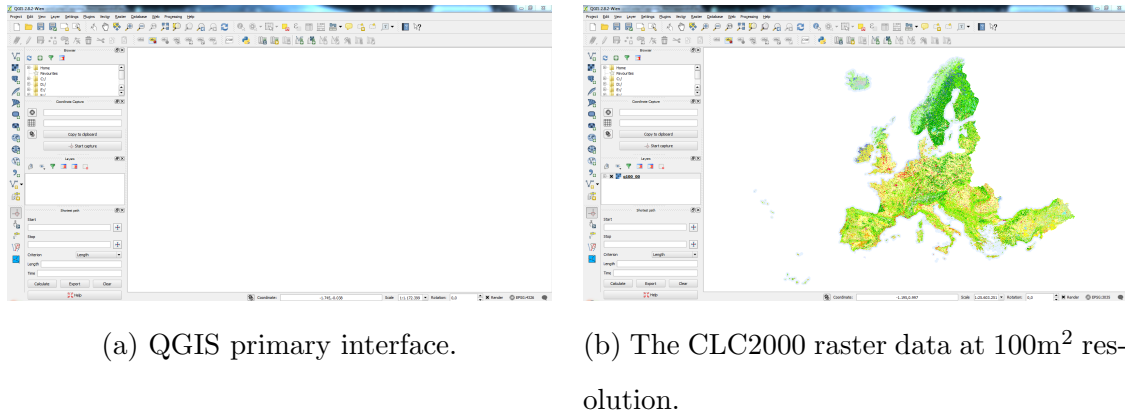


Figure B.2: Snapshots of the QGIS.

operating systems and supports numerous vector, raster, and database formats and functionalities. We use the QGIS in order to read the CLC raster data in the following way. The process starts by loading the 100m<sup>2</sup> original GeoTIFF image (through Layer → Add Layer → Add Raster Layer). In Fig. B.2 we show the CLC2000 as results from the original raster data. In the next step, we have to store the current map in the WGS 84 datum (right click on g100\_00 in the Layers pane → Save As, and set CRS as “Default CRS (EPSG:4326 - WGS 84<sup>6</sup>)”). Then, save this new reprojected map in GeoTIFF format. Later on, we noticed that the reprojection process can be achieved from an alternative way either (Raster → Projections → Warp, and edit Target SRS field), producing slightly different results.

Given that we have already converted a MODIS tile in UTM projection with WGS 84 datum (see Appendix A), we know the geographic coordinates, since they are automatically shown in the main window of the MRT tool once we load an appropriate HDF-EOS file, as Upper Left (UL)/Lower Right (LR) latitude and longitude positions. We utilize these coordinates in order to crop the CLC map, extracting the area corresponding to a whole MODIS tile (*e.g.*, h19v04, h18v04) which entirely belongs in the coverage area of the CLC map (Raster → Extraction → Clipper<sup>7</sup>, and save as .tif file).

<sup>6</sup>EPSG:4326 is just the EPSG identifier of WGS84!

<sup>7</sup>Latitudes and longitudes exist on a spherical globe and that’s why we need to project them onto a (theoretically) flat map (x and y coordinates). For QGIS Lat = y and Long = x. Therefore,

As already stated, the spatial resolution is inversely proportional to the size of the pixels in the raster grid. Furthermore, the pixel size obviously influences the size of the matrix. Consequently, for a given matrix size, a finer grid supports a higher spatial resolution. In other words, when we have two images referring to the same geographic area and the one is of a higher spatial resolution, then this image is a finer matrix with a greater detail containing more pixels. In this way, more than ones pixels (labels) of CORINE can be assigned to each multispectral pixel of MODIS, since the CLC maps enjoy a higher spatial resolution. Although the GeoTIFF file produced with the aforementioned procedure in QGIS refers to the same location with a predefined MODIS tile, when we load it into the MATLAB workspace, there is not the expected analogy between the matrix containing the labels and the matrix with the feature data. This incompatibility can be attributed to various reasons, such as the error that all projections introduce, or the fact that MODIS spatial resolution is not exactly  $500\text{m}^2$ .

In order to acquire the appropriate proportional association between the two matrices according to their corresponding spatial resolutions, we practically need a tool that allows us to perform calculations and modifications on the basis of the existing raster pixel values. The associated tool provided by the QGIS application is termed Raster Calculator, and is accessible via Raster  $\rightarrow$  Raster Calculator. More specifically, we utilize its capability to set manually the number of rows and columns of the result layer, thus to define the exact resolution, *i.e.*, matrix size, of the calculation area (select the desired raster band from the left pane, click to “Current layer extent”, set desired rows and columns of the result layer, and select the GeoTIFF as the output format). Knowing the size of the MODIS image tile (measured as an array in MATLAB), and assuming that the MODIS pixel is approximately  $5 \times 5$  times bigger than the CORINE pixel, we are finally able to form the data matrices properly (*e.g.*, if the size of the MODIS image is  $2504 \times 3028$ , then the size of the desired CLC labels matrix should be  $12520 \times 15140$ ).

---

MRT UL Corner Longitude  $\leftrightarrow$  Clipper 1 x, MRT UL Corner Latitude  $\leftrightarrow$  Clipper 1 y, MRT LR Corner Longitude  $\leftrightarrow$  Clipper 2 x, MRT LR Corner Latitude  $\leftrightarrow$  Clipper 2 y.

# References

- [1] A. Di Gregorio, *Land Cover Classification System—Classification concepts and user manual for Software version 2*. Rome, Italy: Food and Agriculture Organization of the United Nations, 2005.
- [2] I. McCallum, M. Obersteiner, S. Nilsson, and A. Shvidenko, “A spatial comparison of four satellite derived 1km global land cover datasets,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 8, no. 4, pp. 246–255, 2006.
- [3] D. Manolakis and G. Shaw, “Detection algorithms for hyperspectral imaging applications,” *Signal Processing Magazine, IEEE*, vol. 19, no. 1, pp. 29–43, Jan 2002.
- [4] M. A. Friedl, D. Sulla-Menashe, B. Tan, A. Schneider, N. Ramankutty, A. Sibley, and X. Huang, “MODIS collection 5 global land cover: Algorithm refinements and characterization of new datasets,” *Remote Sensing of Environment*, vol. 114, no. 1, pp. 168–182, 2010.
- [5] Y. Bengio, A. Courville, and P. Vincent, “Representation Learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.
- [6] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, June 2005, pp. 886–893 vol. 1.
- [8] S. N. Goward, B. Markham, D. G. Dye, W. Dulaney, and J. Yang, “Normalized difference vegetation index measurements from the advanced very high resolution radiometer,” *Remote Sensing of Environment*, vol. 35, no. 2, pp. 257 – 277, 1991.
- [9] W. Zhengxing, L. Chuang, and H. Alfredo, “From AVHRR-NDVI to MODIS-EVI: Advances in vegetation index research,” *Acta Ecologica Sinica*, vol. 23, no. 5, pp. 979 – 987, 2003.
- [10] B.-c. Gao, “NDWI — a normalized difference water index for remote sensing of vegetation liquid water from space,” *Remote Sensing of Environment*, vol. 58, no. 3, pp. 257 – 266, 1996.



- [11] C. J. Tucker, "Red and photographic infrared linear combinations for monitoring vegetation," *Remote Sensing of Environment*, vol. 8, no. 2, pp. 127 – 150, 1979.
- [12] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International journal of Remote sensing*, vol. 28, no. 5, pp. 823–870, 2007.
- [13] P. Mertikas and M. E. Zervakis, "Exemplifying the theory of evidence in remote sensing image classification," *International Journal of Remote Sensing*, vol. 22, no. 6, pp. 1081–1095, 2001.
- [14] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sensing of Environment*, vol. 113, Supplement 1, no. 0, pp. S110 – S122, 2009, imaging Spectroscopy Special Issue.
- [15] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. Atli Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 45–54, Jan 2014.
- [16] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Int. Conf. on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 473–480.
- [17] N. Keshava and J. Mustard, "Spectral unmixing," *Signal Processing Magazine, IEEE*, vol. 19, no. 1, pp. 44–57, Jan 2002.
- [18] R. N. Clark, G. A. Swayze, K. E. Livo, R. F. Kokaly, S. J. Sutley, J. B. Dalton, R. R. McDougal, and C. A. Gent, "Imaging spectroscopy: Earth and planetary remote sensing with the usgs tetracorder and expert systems," *Journal of Geophysical Research: Planets*, vol. 108, no. E12, 2003.
- [19] J. Li and J. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, vol. 3, July 2008, pp. III – 250–III – 253.
- [20] C. Salvaggio and C. J. Miller, "Comparison of field- and laboratory-collected midwave and longwave infrared emissivity spectra/data reduction techniques," pp. 549–558, 2001.
- [21] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.
- [22] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, Aug 2014.

- [23] A. Santos, A. Canuto, and A. Neto, “A comparative analysis of classification methods to multi-label tasks in different application domains,” *Int. J. Comput. Inform. Syst. Indust. Manag. Appl.*, vol. 3, pp. 218–227, 2011.
- [24] R. Schapire and Y. Singer, “Boostexter: A boosting-based system for text categorization,” *Machine Learning*, vol. 39, no. 2-3, pp. 135–168, 2000.
- [25] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas, “Multi-label classification of music into emotions.” in *ISMIR*, vol. 8, 2008, pp. 325–330.
- [26] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, “Learning multi-label scene classification,” *Pattern Recognition*, vol. 37, no. 9, pp. 1757 – 1771, 2004.
- [27] M. Bossard, J. Feranec, J. Otahel *et al.*, “Corine land cover technical guide: Addendum 2000,” 2000.
- [28] C. Justice, E. Vermote, J. Townshend, R. DeFries, D. Roy, D. Hall, V. Salomonson, J. Privette, G. Riggs, A. Strahler, W. Lucht, R. Myneni, Y. Knyazikhin, S. Running, R. Nemani, Z. Wan, A. Huete, W. Van Leeuwen, R. Wolfe, L. Giglio, J.-P. Muller, P. Lewis, and M. Barnsley, “The moderate resolution imaging spectroradiometer (modis): land remote sensing for global change research,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 4, pp. 1228–1249, Jul 1998.
- [29] J. M. Bioucas-Dias, A. Plaza, S. Member, N. Dobigeon, M. Parente, Q. Du, S. Member, P. Gader, and J. Chanussot, “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 354–379, 2012.
- [30] K. Dembczyński, W. Waegeman, W. Cheng, and E. HG’Ollermeier, “On label dependence and loss minimization in multi-label classification,” *Machine Learning*, vol. 88, no. 1-2, pp. 5–45, 2012.
- [31] M.-L. Zhang and K. Zhang, “Multi-label learning by exploiting label dependency,” in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’10. New York, NY, USA: ACM, 2010, pp. 999–1008.
- [32] C. Poultney, S. Chopra, and Y. Lecun, “Efficient learning of sparse representations with an energy-based model,” in *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2006.
- [33] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng, “Measuring invariances in deep networks,” in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 646–654.

- [34] E. Bartholomé and A. S. Belward, “GLC2000: a new approach to global land cover mapping from earth observation data,” *International Journal of Remote Sensing*, vol. 26, no. 9, pp. 1959–1977, 2005.
- [35] O. Arino, D. Gross, F. Ranera, L. Bourg, M. Leroy, P. Bicheron, J. Latham, A. Di Gregorio, C. Brockman, R. Witt, P. Defourny, C. Vancutsem, M. Herold, J. Sambale, F. Achard, L. Durieux, S. Plummer, and J.-L. Weber, “GlobCover: ESA service for global land cover from MERIS,” in *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, July 2007, pp. 2412–2415.
- [36] M. Friedl, D. McIver, J. Hodges, X. Zhang, D. Muchoney, A. Strahler, C. Woodcock, S. Gopal, A. Schneider, A. Cooper, A. Baccini, F. Gao, and C. Schaaf, “Global land cover mapping from MODIS: algorithms and early results,” *Remote Sensing of Environment*, vol. 83, no. 1, pp. 287 – 302, 2002, the Moderate Resolution Imaging Spectroradiometer (MODIS): a new generation of Land Surface Monitoring.
- [37] G. Büttner, J. Feranec, G. Jaffrain, L. Mari, G. Maucha, and T. Soukup, “The CORINE land cover 2000 project,” *EARSeL eProceedings*, vol. 3, no. 3, pp. 331–346, 2004.
- [38] C. Giri, Z. Zhu, and B. Reed, “A comparative analysis of the Global Land Cover 2000 and MODIS land cover data sets,” *Remote Sensing of Environment*, vol. 94, no. 1, pp. 123 – 132, 2005.
- [39] J. Guo, J. Zhang, Y. Zhang, and Y. Cao, “Study on the comparison of the land cover classification for multitemporal modis images,” in *Earth Observation and Remote Sensing Applications, 2008. EORSA 2008. International Workshop on*, June 2008, pp. 1–6.
- [40] V. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. Rigol-Sanchez, “An assessment of the effectiveness of a random forest classifier for land-cover classification,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, no. 0, pp. 93 – 104, 2012.
- [41] G. Mountrakis, J. Im, and C. Ogole, “Support vector machines in remote sensing: A review,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 3, pp. 247 – 259, 2011.
- [42] P. Teillet, K. Staenz, and D. William, “Effects of spectral, spatial, and radiometric characteristics on remote sensing vegetation indices of forested regions,” *Remote Sensing of Environment*, vol. 61, no. 1, pp. 139 – 149, 1997.
- [43] *Remote Sensing Satellites*. John Wiley & Sons Ltd, 2014, pp. 524–576.
- [44] R. DeFries, M. Hansen, and J. Townshend, “Global discrimination of land cover types from metrics derived from avhrr pathfinder data,” *Remote Sensing of Environment*, vol. 54, no. 3, pp. 209 – 222, 1995.

- [45] X. Yang and C. P. Lo, "Using a time series of satellite imagery to detect land use and land cover changes in the atlanta, georgia metropolitan area," *International Journal of Remote Sensing*, vol. 23, no. 9, pp. 1775–1798, 2002.
- [46] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen, and W. M. Porter, "The airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sensing of Environment*, vol. 44, no. 2, pp. 127 – 143, 1993, airborne Imaging Spectrometry.
- [47] P. Gege, D. Beran, W. Mooshuber, J. Schulz, and H. Van Der Piepen, "System analysis and performance of the new version of the imaging spectrometer rosis," in *Proceedings of the First EARSeL Workshop on Imaging Spectroscopy*. University of Zurich Remote Sensing Laboratories, 1998, pp. 29–35.
- [48] G. Tzagkarakis, G. Tsagkatakis, J.-L. Starck, and P. Tsakalides, "Compressive video classification in a low-dimensional manifold with learned distance metric," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, Aug 2012, pp. 155–159.
- [49] G. Tsagkatakis and A. Savakis, "Sparse representations and distance learning for attribute based category recognition," in *Trends and Topics in Computer Vision*, ser. Lecture Notes in Computer Science, K. Kutulakos, Ed. Springer Berlin Heidelberg, 2012, vol. 6553, pp. 29–42.
- [50] N. Keshava, "A survey of spectral unmixing algorithms," *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 55–78, 2003.
- [51] M. E. Winter, "N-findr: an algorithm for fast autonomous spectral end-member determination in hyperspectral data," pp. 266–275, 1999.
- [52] J. Nascimento and J. Bioucas-Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 4, pp. 898–910, April 2005.
- [53] J. Bioucas-Dias, "A variable splitting augmented lagrangian approach to linear spectral unmixing," in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS '09. First Workshop on*, Aug 2009, pp. 1–4.
- [54] P. Comon, C. Jutten, and J. Herault, "Blind separation of sources, part ii: Problems statement," *Signal Processing*, vol. 24, no. 1, pp. 11 – 20, 1991.
- [55] J. Nascimento and J. Bioucas Dias, "Does independent component analysis play a role in unmixing hyperspectral data?" *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 1, pp. 175–187, Jan 2005.
- [56] K. Themelis, A. Rontogiannis, and K. Koutroumbas, "A novel hierarchical bayesian approach for sparse semisupervised hyperspectral unmixing," *Signal Processing, IEEE Transactions on*, vol. 60, no. 2, pp. 585–599, Feb 2012.

- [57] J. Bioucas-Dias and M. Figueiredo, “Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing,” in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on*, June 2010, pp. 1–4.
- [58] G. Tsagkatakis and P. Tsakalides, “Compressed hyperspectral sensing,” pp. 940 307–940 307–9, 2015.
- [59] Y. Altmann, A. Halimi, N. Dobigeon, and J.-Y. Tourneret, “Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery,” *Image Processing, IEEE Transactions on*, vol. 21, no. 6, pp. 3017–3025, June 2012.
- [60] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.
- [61] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contracting auto-encoders: Explicit invariance during feature extraction,” in *Int. Conf. on Machine Learning*, 2011.
- [62] R. Goroshin and Y. LeCun, “Saturating auto-encoder,” *CoRR*, vol. abs/1301.3577, 2013.
- [63] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, “Semi-supervised recursive autoencoders for predicting sentiment distributions,” in *Conf. on Empirical Methods in Natural Language Processing*, ser. EMNLP ’11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 151–161.
- [64] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 6645–6649.
- [65] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
- [66] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” in *Int. Conf. on Computer Vision*, Sept 2009, pp. 2146–2153.
- [67] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, G. J. Gordon and D. B. Dunson, Eds., vol. 15. Journal of Machine Learning Research - Workshop and Conference Proceedings, 2011, pp. 315–323.
- [68] Y. Bengio, “Learning Deep Architectures for AI,” *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

- [69] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, U. D. Montréal, and M. Québec, “Greedy layer-wise training of deep networks,” in *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2007.
- [70] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, “Exploring strategies for training deep neural networks,” *J. Mach. Learn. Res.*, vol. 10, pp. 1–40, Jun. 2009.
- [71] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [72] R. Salakhutdinov and I. Murray, “On the quantitative analysis of deep belief networks,” in *Int. Conf. on Machine Learning*, ser. ICML ’08. New York, NY, USA: ACM, 2008, pp. 872–879.
- [73] S. Godbole and S. Sarawagi, “Discriminative methods for multi-labeled classification,” in *Advances in Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science, H. Dai, R. Srikant, and C. Zhang, Eds. Springer Berlin Heidelberg, 2004, vol. 3056, pp. 22–30.
- [74] L. Sun, S. Ji, and Ye, *Multi-Label Dimensionality Reduction (Chapman & Hall/CRC Machine Learning & Pattern Recognition)*, 1st ed., 2013.
- [75] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Mining multi-label data,” in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer US, 2010, pp. 667–685.
- [76] N. V. Chawla, N. Japkowicz, and A. Kotcz, “Editorial: Special issue on learning from imbalanced data sets,” *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 1–6, Jun. 2004.
- [77] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Effective and efficient multilabel classification in domains with large number of labels,” in *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD’08)*, 2008.
- [78] W. Bi and J. Kwok, “Efficient multi-label classification with many labels,” in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, S. Dasgupta and D. Mcallester, Eds., vol. 28, no. 3. JMLR Workshop and Conference Proceedings, May 2013, pp. 405–413.
- [79] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Deroski, “An extensive experimental comparison of methods for multi-label learning,” *Pattern Recognition*, vol. 45, no. 9, pp. 3084–3104, Sep. 2012.
- [80] A. Elisseeff and J. Weston, “A kernel method for multi-labelled classification,” in *In Advances in Neural Information Processing Systems 14*. MIT Press, 2001, pp. 681–687.
- [81] A. Clare and R. King, “Knowledge discovery in multi-label phenotype data,” in *Principles of Data Mining and Knowledge Discovery*, ser. Lecture Notes in Computer Science, L. De Raedt and A. Siebes, Eds. Springer Berlin Heidelberg, 2001, vol. 2168, pp. 42–53.

- [82] M.-L. Zhang and Z.-H. Zhou, “ML-KNN: A lazy learning approach to multi-label learning,” *Pattern Recognition*, vol. 40, p. 2007, 2007.
- [83] M.-L. Zhang, “ML-rbf: Rbf neural networks for multi-label learning,” *Neural Processing Letters*, vol. 29, no. 2, pp. 61–74, 2009.
- [84] L. Rokach, A. Schclar, and E. Itach, “Ensemble methods for multi-label classification,” *Expert Systems with Applications*, vol. 41, no. 16, pp. 7507 – 7523, 2014.
- [85] T. G. Dietterich, “Ensemble methods in machine learning,” in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2000, vol. 1857, pp. 1–15.
- [86] O. Luaces, J. Díez, J. Barranquero, J. del Coz, and A. Bahamonde, “Binary relevance efficacy for multilabel classification,” *Progress in Artificial Intelligence*, vol. 1, no. 4, pp. 303–313, 2012.
- [87] J. Read, B. Pfahringer, G. Holmes, and E. Frank, “Classifier chains for multi-label classification,” vol. 85, no. 3, 2011, pp. 335–359.
- [88] A. Dimou, G. Tsoumakas, V. Mezaris, I. Kompatsiaris, and L. Vlahavas, “An empirical study of multi-label learning methods for video annotation,” in *Content-Based Multimedia Indexing, 2009. CBMI '09. Seventh International Workshop on*, June 2009, pp. 19–24.
- [89] M. D. Turner, C. Chakrabarti, T. B. Jones, J. F. Xu, P. T. Fox, G. F. Luger, A. R. Laird, and J. A. Turner, “Automated annotation of functional imaging experiments via multi-label classification,” *Frontiers in neuroscience*, vol. 7, 2013.
- [90] T.-H. Chiang and H.-Y. Lo, “A ranking-based knn approach for multi-label classification.”
- [91] W. Cheng and E. Hüllermeier, “Combining instance-based learning and logistic regression for multilabel classification,” *Mach. Learn.*, vol. 76, no. 2-3, pp. 211–225, Sep. 2009.
- [92] G. Tsoumakas, I. Katakis, and L. Vlahavas, “Random k-labelsets for multilabel classification,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 7, pp. 1079–1089, July 2011.
- [93] I. Jolliffe, *Principal Component Analysis*. John Wiley & Sons, Ltd, 2014.
- [94] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala, “Latent semantic indexing: A probabilistic analysis,” in *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, ser. PODS '98. New York, NY, USA: ACM, 1998, pp. 159–168.
- [95] A. J. Izenman, “Linear discriminant analysis,” in *Modern Multivariate Statistical Techniques*, ser. Springer Texts in Statistics. Springer New York, 2008, pp. 237–280.
- [96] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

- [97] M. Balasubramanian and E. L. Schwartz, "The Isomap algorithm and topological stability," *Science*, vol. 295, no. 5552, p. 7, 2002.
- [98] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [99] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee, "A comparison of multi-label feature selection methods using the problem transformation approach," *Electronic Notes in Theoretical Computer Science*, vol. 292, pp. 135 – 151, 2013, proceedings of the XXXVIII Latin American Conference in Informatics (CLEI).
- [100] Y. Zhang and Z.-H. Zhou, "Multilabel dimensionality reduction via dependence maximization," *ACM Trans. Knowl. Discov. Data*, vol. 4, no. 3, pp. 14:1–14:21, Oct. 2010.
- [101] Y. nan Chen and H. tien Lin, "Feature-aware label space dimension reduction for multi-label classification," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1529–1537.
- [102] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information Retrieval*, vol. 1, no. 1-2, pp. 69–90, 1999.
- [103] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.
- [104] S. Nowak, H. Lukashevich, P. Dunker, and S. Rüger, "Performance measures for multilabel evaluation: A case study in the area of image classification," in *Proceedings of the International Conference on Multimedia Information Retrieval*, ser. MIR '10. New York, NY, USA: ACM, 2010, pp. 35–44.
- [105] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. K. Hadley, A. S. Hadley, and M. G. Betts, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," *The Journal of the Acoustical Society of America*, vol. 131, no. 6, pp. 4640–4650, 2012.
- [106] D. W. Patterson, *Artificial Neural Networks: Theory and Applications*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.
- [107] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362.
- [108] D. Mishra, A. Yadav, S. Ray, and P. Kalra, "Exploring biological neuron models," *Directions, The Research Magazine of IIT Kanpur*, vol. 7, no. 3, pp. 13–22, 2006.
- [109] A. K. Jain, J. Mao, and K. Mohiuddin, "Artificial Neural Networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31 – 44, 1996.



- [110] D. Svozil, V. Kvasnicka, and J. Pospichal, “Introduction to multi-layer feed-forward neural networks,” *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43 – 62, 1997.
- [111] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, Nov 2012.
- [112] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. Montavon, G. Orr, and K.-R. Müller, Eds. Springer Berlin Heidelberg, 2012, vol. 7700, pp. 9–48.
- [113] K. Fotiadou, G. Tsagkatakis, and P. Tsakalides, “Low light image enhancement via sparse representations,” in *Image Analysis and Recognition*. Springer International Publishing, 2014, pp. 84–93.
- [114] G. Tsagkatakis and A. Savakis, “Sparse representations and distance learning for attribute based category recognition,” in *Trends and Topics in Computer Vision*. Springer Berlin Heidelberg, 2012, pp. 29–42.
- [115] P. Lennie, “The cost of cortical computation,” *Current Biology*, vol. 13, no. 6, 2003.
- [116] P. Petrantonakis and P. Poirazi, “A compressed sensing perspective of hippocampal function,” *Frontiers in systems neuroscience*, vol. 8, 2014.
- [117] A. Ng, “Sparse autoencoder,” *CS294A Lecture notes*, vol. 72, 2011.
- [118] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, “Learning invariant features through topographic filter maps,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2009, pp. 1605–1612.
- [119] J. Dwyer and G. Schmidt, “The MODIS Reprojection Tool,” in *Earth Science Satellite Remote Sensing*, J. Qu, W. Gao, M. Kafatos, R. Murphy, and V. Salomonson, Eds. Springer Berlin Heidelberg, 2006, pp. 162–177.
- [120] R. B. Myneni, C. Keeling, C. Tucker, G. Asrar, R. Nemani *et al.*, “Increased plant growth in the northern high latitudes from 1981 to 1991,” *Nature*, vol. 386, no. 6626, 1997.
- [121] A. J. Ramon Solano, Kamel Didan and A. Huete, “MODIS vegetation index user’s guide (MOD13 series),” May 2010.
- [122] P. J. Sellers, “Canopy reflectance, photosynthesis and transpiration,” *International Journal of Remote Sensing*, vol. 6, no. 8, pp. 1335–1372, 1985.
- [123] C. Justice, J. Townshend, E. Vermote, E. Masuoka, R. Wolfe, N. Saleous, D. Roy, and J. Morisette, “An overview of MODIS land data processing and product status,” *Remote Sensing of Environment*, vol. 83, no. 1, pp. 3 – 15, 2002, the Moderate Resolution Imaging Spectroradiometer (MODIS): a new generation of Land Surface Monitoring.

- [124] Z.-L. Li, B.-H. Tang, H. Wu, H. Ren, G. Yan, Z. Wan, I. F. Trigo, and J. A. Sobrino, "Satellite-derived land surface temperature: Current status and perspectives," *Remote Sensing of Environment*, vol. 131, pp. 14 – 37, 2013.
- [125] B. N. Holben, "Characteristics of maximum-value composite images from temporal AVHRR data," *International Journal of Remote Sensing*, vol. 7, no. 11, pp. 1417–1434, 1986.
- [126] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proceedings of the 14th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '06. New York, NY, USA: ACM, 2006, pp. 421–430.
- [127] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "MULAN: A java library for multi-label learning," *J. Mach. Learn. Res.*, vol. 12, pp. 2411–2414, Jul. 2011.
- [128] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
- [129] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 10, pp. 1338–1351, Oct 2006.
- [130] Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Nonlinearity detection in hyperspectral images using a polynomial post-nonlinear mixing model," *Image Processing, IEEE Transactions on*, vol. 22, no. 4, pp. 1267–1276, April 2013.
- [131] Y.-R. Yeh, T.-C. Lin, Y.-Y. Chung, and Y.-C. Wang, "A novel multiple kernel learning framework for heterogeneous feature fusion and variable selection," *Multimedia, IEEE Transactions on*, vol. 14, no. 3, pp. 563–574, June 2012.
- [132] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 226–239, Mar 1998.
- [133] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Springer Berlin Heidelberg, 2012, vol. 7700, pp. 437–478.
- [134] W. P. Wan, Z. and X. Li, "Using MODIS Land Surface Temperature and Normalized Difference Vegetation Index products for monitoring drought in the southern great plains, USA," *International Journal of Remote Sensing*, vol. 25, no. 1, pp. 61–72, 2004.
- [135] L. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," 2008.
- [136] L. Geng, M. Ma, X. Wang, W. Yu, S. Jia, and H. Wang, "Comparison of eight techniques for reconstructing multi-satellite sensor time-series ndvi data sets in the heihe river basin, china," *Remote Sensing*, vol. 6, no. 3, pp. 2024–2049, 2014.