

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEEP LEARNING APPLIED TO REMOTE SENSING
IMAGE SEQUENCE ANALYSIS

MELİKE İLTERALP
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

GEBZE
2022

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**DEEP LEARNING APPLIED TO REMOTE
SENSING IMAGE SEQUENCE ANALYSIS**

MELİKE İLTERALP
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

THESIS SUPERVISOR
PROF. DR. ERCHAN APTOULA

GEBZE
2022

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UZAKTAN ALGILANMIŞ GÖRÜNTÜ
DİZGİLERİNİN DERİN ÖĞRENME İLE
ÇÖZÜMLENMESİ

MELİKE İLTERALP
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
PROF. DR. ERCHAN APTOULA

GEBZE
2022

GEBZE TEKNİK ÜNİVERSİTESİ	YÜKSEK LİSANS JÜRİ ONAY FORMU
----------------------------------	--------------------------------------

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu’nun 07/02/2022 tarih ve 2022/08 sayılı kararıyla oluşturulan jüri tarafından 04/03/2022 tarihinde tez savunma sınavı yapılan Melike İltıralp’in tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Prof. Dr. Erchan APTOULA

ÜYE : Prof. Dr. Yusuf Sinan AKGÜL

ÜYE : Doç. Dr. Seniha Esen YÜKSEL

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu’nun
...../...../..... tarih ve/..... sayılı kararı.

SUMMARY

Advances in sensor technology and changes in data policy have led to an ever-increasing volume of remote sensing images. However, the optimal exploitation of remote sensing images for their analysis still remains a problem due to their characteristics. Deep learning techniques come as a prominent choice for such analyzes due to their state-of-the-art results in various domains. However, remote sensing images require adapted techniques to fully exploit their characteristic.

This thesis investigates the development of adapted deep learning techniques for the analysis of remote sensing image sequences, taking into account the characteristics of these images. Four deep learning methods are proposed, each targeting at least two characteristics of remote sensing images for their analysis. The first approach proposes a multi-task learning model for water quality estimation using labeled and unlabeled data. The second approach treats pixels as scale series and couples them with an LSTM network for pixel-level classification. The third approach combines a hierarchical tree-based representation with graph convolutional neural networks for remote sensing image classification. The last method proposes an alternative pooling mechanism for segmentation networks.

The models of the first and second approaches are trained and tested with their datasets and compared with their alternatives. The first approach achieved a substantial increase in estimation performance and demonstrated that harnessing the unlabeled data improves water quality estimation, while a significant increase in generalization performance for pixel-level classification was observed with the second approach compared to its alternatives. The proposed methods are presented for the third and fourth approaches.

Keywords: Deep Learning, Remote Sensing, Image Analysis, Semi-Supervised Learning, Multi-Task Learning, Time-Series, Multi-Scale.

ÖZET

Sensör teknolojilerindeki yenilikler ve veri politikalarındaki gelişmeler sayesinde uzaktan algılanmış görüntüler günbegün çoğalmaktadır. Bu artış, birçok farklı alandaki problemin çözümü için yeri doldurulamaz bilgi sağlayan bu görüntülerin, uygun şekilde çözümlenmesine yönelik ihtiyacı da beraberinde getirmektedir. Ancak, uzaktan algılanmış görüntülerin, kendine özgün karakteristiğinden en uygun faydayı sağlayacak şekilde çözümlenmesi hâlâ çözülememiş bir sorundur. Derin öğrenme teknikleri, birçok alanda bilinen en iyi başarı düzeyini sağladığı için bu görüntülerin çözümlenmesi için uygun bir tercih olarak öne çıkmaktadır. Ancak, uzaktan algılanmış görüntüler, karakteristiklerinden tam anlamıyla faydalanabilmek için özelleşmiş teknikleri gerektirir.

Bu tez kapsamında, uzaktan algılanmış görüntü dizgilerinin bahsi geçen karakteristiğini dikkate alarak çözümlenmesi için özelleşmiş derin öğrenme teknikleri geliştirilmiştir. Bu amaçla, her biri bahsi geçen görüntülerin en az iki karakteristiğini ele alan dört derin öğrenme yaklaşımı önerilmiştir. İlk yaklaşımda piksel seviyesinde sınıflandırma için görüntünün pikselleri ölçek-serisi şeklinde ele alınarak bir Uzun Kısa Süreli Bellek ağı ile birlikte kullanılmıştır. İkinci yaklaşımda su kalitesinin kestirimi için etiketli ve etiketsiz veriyi bir arada kullanan bir çoklu-görev öğrenen model önerilmiştir. Üçüncü yaklaşım bahsi geçen görüntülerin çözümlenmesi için hiyerarşik ağaç tabanlı temsil ile çizge evrişimli sinir ağlarını bir araya getirmektedir. Son yaklaşımda bölütleme ağlarının temel bileşenlerinden biri olan havuzlama katmanı için bir alternatif önerilmiştir.

Birinci ve ikinci yaklaşımın modelleri veri kümeleriyle eğitilip sınanmış ve alternatifleriyle karşılaştırılmıştır. Bunun sonucunda, ilk yaklaşımda su kalitesi kestiriminde belirgin bir başarıml elde edilmiş ve etiketlenmemiş verilerden faydalanmanın kestirimi iyileştirdiği gösterilmiştir. İkinci yaklaşımda piksel seviyesinde sınıflandırmada alternatiflerine göre genelleme performansında kayda değer bir artış gözlemlenmiştir. Üçüncü ve dördüncü yaklaşım için önerilen yöntemler sunulmuştur.

Anahtar Kelimeler: Derin Öğrenme, Uzaktan Algılama, Görüntü Çözümleme, Yarı-Gözetimli Öğrenme, Çoklu-Görev Öğrenme, Zaman Serileri, Çoklu-Ölçek.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Prof. Erchan Aptoula, for his constant support and guidance throughout my master's studies. Apart from his expertise in the field, his mentorship and enthusiasm have encouraged me through this thesis and all related publications. I am also grateful for the funding support from the MULTISCALE project of TUBITAK (project number: 118E258). I would like to extend my appreciation to my jury committee members Assoc. Prof. Seniha Esen Yüksel Erdem and Prof. Dr. Yusuf Sinan Akgül for their comments, time, and patience throughout the process.

I am very grateful for the support I received from my family, my brother Mert, my father İsmet and my mother Ayşen, to whom I owe special thanks for her unwavering belief in me all the time. I am also indebted to the family members whose names I could not mention. Without their understanding and encouragement, this thesis would not have been possible.

During my studies at GTU, I was fortunate to have my beloved friends and colleagues. Firstly, I owe special thanks to Beyza Türkmen, Ferda Abbasoğlu, and Özgü Göksu not only for sharing their technical skills with me, but also for being there for me during all the ups and downs of the process. I would also like to extend my appreciation to Fatma Zehra Sarı and Usama Derbashi for making the whole process more bearable and fun. I am also indebted to Sibel Gülmez and Hadi Alizadeh, together with Ferda, who patiently helped me with the technical issues. My appreciation also extends to the members of our Coffee & Paper reading group.

I would also like to acknowledge the support of my friends Ece Fırat Uzman, Yeşim Arı Dalman and Gökhan Çiçekbilek. Finally, it has been a great pleasure to work with my dear friends from Focusmate: Hilal Eda Arpacı, Oumaima Zamharir, Naeimeh Najafizadeh, Andrew, Selin, and Adriana Bautista. Thank you so much for bearing with me through all these sessions!

TABLE of CONTENTS

	<u>Page</u>
SUMMARY	v
ÖZET	vi
ACKNOWLEDGEMENTS	vii
TABLE of CONTENTS	viii
LIST of ABBREVIATIONS and ACRONYMS	x
LIST of FIGURES	xi
LIST of TABLES	xiii
1. INTRODUCTION	1
1.1. Problem Definition	1
1.2. Aim and Scope	2
1.3. Contributions	4
1.4. Outline	5
2. LITERATURE REVIEW	6
3. DEEP SEMI-SUPERVISED MULTI-TASK LEARNING APPROACH FOR CHLOROPHYLL-A RETRIEVAL FROM REMOTE SENSING IMAGES	9
3.1. Problem Definition	9
3.2. Background	10
3.2.1. Shallow Approaches	10
3.2.2. Deep Learning Approaches	11
3.3. Methodology	13
3.3.1. Study Area and Dataset	14
3.3.2. Training Procedure	16
3.3.3. Input Shape	17
3.3.4. Network Architecture	17
3.3.5. Loss Functions	17
3.3.6. Multi-Task Loss	21
3.4. Experiments & Results	22
3.4.1. Data Split	22
3.4.2. Alternative Methods	23

3.4.3.	Hyperparameters	23
3.4.4.	Results	24
3.4.5.	Ablation Studies	27
3.5.	Discussion & Limitations	29
3.6.	Conclusion	30
4.	PROCESSING ATTRIBUTE PROFILES AS SCALE SERIES FOR REMOTE SENSING IMAGE CLASSIFICATION	32
4.1.	Problem Definition	32
4.2.	Background	33
4.2.1.	Hierarchical Tree Representation	34
4.2.2.	Attribute Profiles	36
4.2.3.	Long-Short Term Memory Network	40
4.3.	Methodology	41
4.3.1.	APs as Input to LSTM	41
4.3.2.	Network Architecture	41
4.4.	Experiments	43
4.4.1.	Datasets	43
4.4.2.	A Common Validation Misconduct	46
4.4.3.	Implementation Details	48
4.4.4.	Alternative Methods	49
4.5.	Results & Discussion	50
4.5.1.	Results	50
4.5.2.	Discussion	50
4.6.	Conclusion	52
5.	CONCLUSION AND FUTURE WORK	54
5.1.	Conclusion	54
5.2.	Future Work	56
	REFERENCES	57
	BIOGRAPHY	65
	APPENDICES	66

LIST of ABBREVIATIONS and ACRONYMS

<u>Abbreviations</u>	<u>Explanations and Acronyms</u>
2D	: 2 Dimensional
3D	: 3 Dimensional
AP	: Attribute Profile
Chl-a	: Chlorophyll-a
CNN	: Convolutional Neural Network
Conv	: Convolutional Layer
DL	: Deep Learning
EAP	: Extended Attribute Profile
FC	: Fully Connected Layer
GCN	: Graph Convolutional Network
GPU	: Graphical Processing Unit
K	: One Thousand
LSTM	: Long Short-Term Memory
MAE	: Mean Absolute Error
ML	: Machine Learning
MLP	: Multilayer Perceptron
MSE	: Mean Squared Error
$\mu\text{g/L}$	: Microgram/Liter
NASA	: National Aeronautics and Space Administration
PCA	: Principal Component Analysis
R^2	: Coefficient of Determination
RF	: Random Forest
RMSE	: Root Mean Square Error
RMSProp:	: Root Mean Square Propagation
RNN	: Recurrent Neural Network
RS	: Remote Sensing
Tanh	: Hyperbolic Tangent

LIST of FIGURES

<u>Figure No:</u>	<u>Page</u>
3.1: Study area, Lake Balik situated in Samsun, a province in northern Turkey and its in-situ measurement sites.	14
3.2: The distribution of Chl-a concentration in Lake Balik over time. Each point in the graph represents the average of 10 monitoring sites, while the vertical bar indicates the range of values across sites, where mean = 35.82 $\mu\text{g/L}$, standard deviation = 19.43 $\mu\text{g/L}$, minimum = 11.04 $\mu\text{g/L}$, and maximum = 108.35 $\mu\text{g/L}$.	15
3.3: The pink areas show the lake masks used to employ unlabeled samples in the ablation studies centered on each Chl-a monitoring site with (a) an 11 pixel diameter circle, (b) a 45 pixel diameter circle, and (c) the entire lake surface, yielding 28,160, 503,040, 2,364,928 unlabeled samples in 32 images, respectively.	19
3.4: The outline of the proposed multi-task learning network architecture, where Conv and FC denote the convolutional layer and the fully connected layer, respectively.	20
3.5: Chl-a estimation maps in $\mu\text{g/L}$ created by the proposed multi-task learning model for spring, summer, fall, and winter 2017.	26
3.6: The relationship between observed Chl-a values and those estimated by the proposed multi-task learning model.	26
3.7: Acquisition month prediction performance of the proposed multi-task learning model with different number of unlabeled samples.	31
4.1: The scale difference in size between the instances of cars (yellow) and vegetation (green) classes can be observed, with both images coming from the Vaihingen Dataset.	34
4.2: The input grayscale image (a), its min-tree (b) and max-tree (c) representations.	37
4.3: The input grayscale image (top left) and its α -tree representation (right).	38

- 4.4: Outline of the proposed pixel description strategy and architecture of the 42
model, where PCA, PC, FC, and ReLU denote principal component
analysis, principal component image, fully connected layer, and
rectified linear unit, respectively.
- 4.5: The Pavia University dataset, the training set, and the ground truth, as 44
commonly encountered in the state-of-the-art, from left to right.
- 4.6: The Reykjavik dataset, the training set, and the ground truth, as 45
commonly encountered in the state-of-the-art, from left to right.
- 4.7: The Reykjavik dataset train and test sets, horizontal split (left), and 46
vertical split (right).
- 4.8: The Pavia University dataset train and test sets, horizontal split (left), 47
and vertical split (right).

LIST of TABLES

<u>Table No:</u>	<u>Page</u>
3.1: The number of samples per month for each dataset.	19
3.2: Mean Chl-a estimation results across 10 folds of cross-validation with the best results in bold where, MAE, RMSE, and * denote the coefficient of determination, mean absolute error, root mean square error, and the proposed approach, respectively.	25
3.3: Chl-a estimation performance of the proposed multi-task learning model under 10-fold cross validation with different number of shared layers, where R^2 , MAE, and RMSE denote the coefficient of determination, mean absolute error, and root mean square error, respectively. The best results are indicated in bold (patch size = 3×3).	27
3.4: Chl-a estimation performance of the proposed multi-task learning model under 10-fold cross validation with different patch sizes, where R^2 , MAE, and RMSE denote the coefficient of determination, mean absolute error, and root mean square error, respectively. The best results are indicated in bold (number of shared layers=4).	28
3.5: Chl-a estimation performance of the proposed multi-task learning model under 10-fold cross validation with different number of unlabeled samples, where R^2 , MAE, and RMSE denote the coefficient of determination, mean absolute error, and root mean square error, respectively. The best results are indicated in bold (number of shared layers = 4, patch size 3×3).	28
3.6: Chl-a estimation performance of the proposed multi-task learning model under 10-fold cross validation with different task loss weighting schemes, where R^2 , MAE, and RMSE denote the coefficient of determination, mean absolute error, and root mean	29

square error, respectively. The best results are indicated in bold (number of shared layers=4, patch size 3×3).

- 4.1: The number of samples per class in the standard version of the Pavia University Dataset, and its horizontal and vertical splits. 44
- 4.2: The number of samples per class in the standard version of the Reykjavik Dataset, and its horizontal and vertical splits. 45
- 4.3: Pixel classification results in the form of Kappa statistics $\kappa \times 100$ 51 with the best results in bold for the Pavia University and Reykjavik Datasets, where RF, CNN, LSTM, and * denote random forest, convolutional neural network, long-short term memory network, and the proposed approach, respectively.

1. INTRODUCTION

The history of remote sensing (RS) begins with the invention of the photographic camera, which photographers attached to balloons in the 1850s to obtain images of the Earth's surface [1]. A famous attempt was the attachment of cameras to a fleet of pigeons in the early 20th century. The value of aerial photography was recognized during World War I, as analysis of aerial images enabled the correct identification of enemy installations [2].

With the launch of the first artificial satellite, Sputnik-1, in 1957, scientists became aware of the possibility of observing the Earth from space, which eventually led to the foundation of the National Aeronautics and Space Administration (NASA). The establishment of NASA led to more mature approaches that eventually resulted in the 1972 launch of Landsat-1, the first satellite for Earth observation. This milestone marked the beginning of modern remote sensing studies.

1.1. Problem Definition

Recent advances in Earth observation, such as the new satellite missions like the Copernicus Programme, have resulted in an abundance of remote sensing images with ever increasing spectral bands and temporal instances, as well as finer spatial resolutions [3]. In addition, changes in data policy, as in the case of Landsat, have resulted in remote sensing data being freely available to the public [4].

Remote sensing images provides invaluable information for solving a variety of problems on Earth, from monitoring urban development and identifying anthropogenic pressures to understanding the effects of climate change [5]. This irreplaceable position of remote sensing images via providing information to a wide range of problems on Earth and the drastic increase in their volume have made availing from them to the maximum possible extent a key quest [3]. What lies in the basis of this quest is the analysis of remote sensing images in order to interpret them.

The main challenge in this analysis is the exploitation of the intrinsic characteristics of remote sensing images, that is to bring together their multi-scale, multi-/hyperspectral, and multi-temporal properties so that they can introduce complementary information to each other when available. Moreover, the effect of

exploiting such properties can be amplified by using unlabeled data, which are amply available compared to the few number of labeled data.

1.2. Aim and Scope

A major and current challenge in this analysis is the classification of remote sensing images. Deep learning (DL) has made a breakthrough entry to the world of image classification in 2012 by reducing the error rate in object classification by half [6], which led to its fast adoption for computer vision problems. DL is a family of artificial neural network-based learning algorithms consisting of multiple connected layers that transform input data into an estimation of the output through the neurons operating in each layer by learning progressively higher-level features from the input [7]. The architecture of a DL network consists of these layers stacked on top of each other, hence coining the term deep.

DL methods are popular due to their ability of learning high-level representations from raw input data, and therefore eliminating the need for hand-crafted features on which the performance of the learning algorithm heavily depends [8]. As a result, DL methods have achieved state-of-the-art results in several domains where the reader is referred to [9] for more details. Considering the breakthrough results of DL methods in various domains, especially in problems containing sequential and spatial data [10], and the progressive increase in the volume of remote sensing data, the application of DL methods to the analysis of remote sensing images has become very appealing. However, remote sensing images has the following intrinsic characteristics that make the direct application of well-performing DL methods unsuitable for their analysis, thus necessitating the development of DL methods specifically for this analysis:

- Multi-scale appearance of the phenomena

The altitude of Earth observation sensors varies from a few to hundreds of kilometers. Accordingly, the scale of the captured phenomena on the image varies, resulting in the instances of the same phenomenon being appeared in a very different number of pixels on the image. This poses a challenge to the existing DL approaches, as they were developed for computer vision problems where instances of a class appear at a single scale in their datasets [11]. In order to deal this with issue, RS images require

DL approaches that are robust to the multi-scale appearance of phenomena compared to existing DL techniques.

- Necessity of the exploitation of the spatial context

Due to the multi-scale problem mentioned above, instances of the same phenomenon can appear in a very different number of pixels. This issue requires DL approaches that can exploit contextual information by considering the spatial relationship available to each pixel, rather than treating pixels independently [12]. This contextual information can also be extended hierarchically by considering the relationship between different scales of phenomena [13].

- Necessity of the exploitation of the temporal context

Temporal information in RS images can provide either essential or complementary information, depending on the problem domain. However, existing DL methods for analyzing multi-temporal images are not directly applicable to RS images, i.e., RS images cannot be processed as in the case of videos, since misalignments within phenomena may occur due to altitude differences and pixel intensities may change for the same spectral band as illumination may vary depending on weather conditions [14]. Therefore, DL approaches that take advantage of the temporal properties of RS images are needed.

- High number of spectral bands

RS images are usually consisted of a large number of bands because their imaging systems acquire data at different wavelengths. Although they provide the opportunity to observe surfaces with rich spectral content, these bands contain highly redundant information because they are acquired from the same surface region. As a naive approach, existing DL models can be adapted for this problem by increasing the number of neurons in the first layer, but this introduces unnecessary computational complexity in training. Instead, DL approaches that account for this spectral density of RS images are needed.

- Heterogeneity of RS images in time and space

Instances of the same phenomenon on the Earth's surface have very different characteristics due to different geographic, climatological, and anthropogenic

conditions, which cause them to appear very differently on the captured RS images as well [11]. This effect is multiplied in the case of multi-temporal RS images. This problem also causes learning algorithms trained for one region to work well for that particular region but fail for other regions in different geographies that share the same semantic categories [10]. Therefore, DL approaches that target this heterogeneity of RS images are needed.

- **Paucity of labeled data**

Obtaining RS images is tedious and labeling them may require expert level of knowledge, as a large phenomenon can be represented with a small number of pixels, as some RS datasets have a spatial resolution of up to 30 meters. Labeling may also require additional steps, e.g., water quality estimation requires collecting water samples and analyzing them in the laboratory. Therefore, DL techniques are needed that can exploit the available data in RS images, labeled or unlabeled, to the maximum extent possible.

As a result, four DL techniques are proposed for the analysis of RS images with exploiting their above-mentioned characteristics.

1.3. Contributions

The contributions of this thesis are following:

- i) A multi-task learning network for water quality estimation is presented to address the lack of labeled data in multi-temporal RS images by using the unlabeled image locations along with the labeled ones.
- ii) A pixel-level classification technique is proposed to exploit the multi-scale appearance of phenomena in RS images. For this purpose, attribute profiles are used to describe pixels as scale-series and coupled with a long short-term memory network to exploit their sequential nature, similar to time series. It also highlights a common validation misconduct encountered in RS image datasets that leads to misleadingly high generalization performance, and instead considers the heterogeneous appearance of phenomena.

- iii) A graph convolutional neural network-based approach for pixel-level classification is proposed that considers the multi-scale appearance of phenomena and exploits spatial context.
- iv) A special pooling mechanism is proposed to address the issues of contour precision and missing small phenomena, which convey essential information for the RS images due to their multi-scale appearance, caused by multiple downsampling with the existing pooling layers in segmentation networks.

1.4. Outline

This thesis is organized as follows: The following chapter, Chapter 2, provides information on the state-of-the-art. Chapter 3 presents the first approach to address the limited availability of labeled data in multi-temporal RS image analysis by proposing a multi-task learning network in the context of water quality estimation that can exploit unlabeled data, and shows that harnessing such data improves water quality estimation with comparing it to the alternatives. Chapter 4 proposes a pixel description strategy that takes into account the multi-scale appearance of RS images by combining shallow and deep methods. In this approach, pixels are described using their attribute profile-based scale-series in combination with an LSTM network to exploit their sequential nature, and the results are presented in comparison to the alternatives. Chapter 5 concludes with a summary of the proposed approaches and provides an overview of future studies. Appendices A and B contain publications related to this thesis, and funding information. Appendix C combines the hierarchical tree representation with a GCN for pixel-level classification to take advantage of the spatial context and multi-scale appearance of phenomena. Appendix D addresses an alternative pooling layer that specializes in RS images by considering their multi-scale appearance. It can be used with any segmentation network by replacing the pooling layers and explains the planned experiments.

2. LITERATURE REVIEW

For the spatial-spectral description of the RS images, [15] proposed a method to circumvent the need to determine thresholds for attribute profiles (AP) for the hierarchical tree representation and investigated alternative tree representations. An AP is a sequence of numerical measures describing the hierarchy of nested, connected components that comprise the pixel under study with respect to selected attributes at the predefined thresholds. APs have the disadvantage of requiring a well-chosen threshold for each attribute. In the study, the authors used a whole sequence of nodes, from the root to the node containing the pixel under study, which is called lineage. They selected 5 features per pixel for description, namely:

- 2 attribute values of the nodes where the absolute difference of the attributes is maximum.
- Level of maximum difference between attribute values of nodes in the tree.
- Intensity of the selected pixel
- Attribute value of the selected pixel's node

[16] studied the evolution of hierarchical image representations with component trees and introduced partitioning and inclusion trees. They described their characteristics, construction algorithms, and differences. [17] demonstrated how braid structures can be related to the representation of multimodal images. They showed that the presented braid structure is able to handle multi-modal information to produce more accurate results. In the pioneering work of [18], a novel approach for semi-supervised classification of data with graph structures is presented. They showed that a graph convolutional neural network is capable of encoding both graph structure and node features in a way that is useful for semi-supervised classification. [19] proposed a semi-supervised learning approach that takes into account both adjacency nodes in a graph and neighbour pixels in the spatial domain, and showed that this can significantly improve the performance of hyperspectral image classification. [20] developed a method that goes beyond regular image grids by learning pixel-to-region assignment, showing that regions that are relatively distant in 2D space can be connected by successive graph convolutions.

Studies on water quality estimation using remote sensing can be divided into two categories: semi-analytical and semi-empirical techniques [21]. For an overview of water quality estimation techniques, the reader is advised to read [22], which contains a comprehensive account of the subject.

Semi-analytical approaches are based on radiative transfer, which models the propagation of light in the water column [23]. The use of the radiative transfer equation (RTE) provides a description of the optical properties of the water body [21]. RTEs can be solved using lookup tables and inversion methods, which in turn provide the Chl-a estimation. However, both methods require detailed information about the water constituents in the lake [22]. Another disadvantage of these methods is that they are very sensitive to the correct calculation of optically active constituents in the lake, which can affect the accuracy of the method and eventually lead to errors in the determination of water quality parameters [24].

Semi-empirical approaches, on the other hand, estimate Chl-a values directly using spectral bands instead of using transfer models on them [21]. This direct approach makes them robust to water constituent calculations and does not require prior information about lake characteristics [22]. They work by establishing a regression relationship between the feature vectors generated by a learning algorithm from the spectral values of the pixels for the selected lake sites and their corresponding Chl-a values from the in-situ water measurements. Subsequently, this regression operation is applied to the feature vectors of each remaining pixel to produce a Chl-a estimation map for the entire lake. Examples of this approach include support vector regression [25], a multilayer perceptron [26], and Gaussian processes [27]. Other approaches include the use of linear and nonlinear combinations of spectral bands in place of feature vectors [28], especially for green-blue, far-red, and near-infrared bands [29], and the application of expert-derived band ratios [30]. For a comprehensive overview of band-based approaches, the reader is referred to [31].

The estimation performance of the semi-empirical approaches is tightly limited by the machine learning (ML) algorithm employed. Since the ML algorithms in general are going through a period of change with the breakthrough of DL methods, they are leaving their place in favor of DL methods [32]. This is happening because DL methods have achieved the state-of-the-art results in many challenging computer vision problems in various domains [9], which has finally brought them the attention

they deserve. This breakthrough impact of DL methods has also been resonating in the field of water quality studies for Chl-a estimation.

Particularly, [33] employed a deep neural network (DNN) consisting of 6 hidden layers to the spectral signatures of multispectral images with a spatial resolution of 30 m and estimated water quality parameters of lakes in the United States. Their approach is one of the first applications of DL to water quality estimation, which improved the estimation performance of all water quality parameters compared to other methods. As the first application of CNNs in this context, [34] treated water quality estimation as a classification problem and developed a CNN architecture that takes as input the patch of the pixel under study, whose label is predicted as one of the 6 water quality levels from the multispectral images of two lakes in China. After that, further studies employed CNNs for Chl-a estimation, but they treated the problem as a regression, which is more inherent to this problem rather than classification [35, 36]. Other approaches focused on exploiting the temporal aspect of the problem, which is consistent with the nature of multi-temporal remote sensing images and utilized long short-term memory (LSTM) networks either alone [37] or together with CNNs [38]. Finally, a comprehensive study by [21] developed a mixture density network consisting of 5 layers to estimate Chl-a levels and examined lakes from 10 different countries.

3. DEEP SEMI-SUPERVISED MULTI-TASK LEARNING APPROACH FOR CHLOROPHYLL-A RETRIEVAL FROM REMOTE SENSING IMAGES

3.1. Problem Definition

Lakes are essential for the environment and human life, being part of the carbon cycle, maintaining the ecosystem and providing drinking water [5]. However, despite their importance, they face many threats from humans and the environment. Algal bloom, as one of the major threats, which can produce toxins hazardous to humans and wildlife, reduce biodiversity, and cause changes in the ecosystem [22]. Chlorophyll-a (Chl-a) is a commonly used water quality parameter to estimate the concentration of algal blooms in a water body. Therefore, monitoring such water quality parameters to identify the above threats is of great importance for the sustainability of the lake environments.

Conventional water quality monitoring methods involve taking samples from the water body and analyzing them in the laboratory, making them labor-intensive, time-consuming, and costly methods. Although in-situ sampling gives accurate measurements for the selected lake sites, it cannot reflect the entire water body in time and space due to the heterogeneous distribution of phenomena that affect water quality, such as algal blooms [39].

Moreover, it is practically impossible to sample from every point of the water body [40, 41]. Remote sensing provides spatial and temporal views of Earth surfaces at repetitive time intervals that can be used to mitigate the shortcomings of conventional water quality monitoring. Furthermore, they are cost-effective, as satellites such as Sentinel 2 and Landsat provide remote sensing data free of charge and in the public domain.

Therefore, it is quasi-essential to use satellite images as they convey such useful properties and can provide pixel-level Chl-a estimations for the entire lake surface. Here raises the questions of how to exploit satellite images that does not contain Chl-a values, and how to couple them with a limited number of Chl-a measurement samples to make them useful for the estimations.

One of the main issues of remote sensing applications, similar to computer vision, is the insufficient number of labeled samples, which can lead to overfitting of the data by the model. This issue has been multiplied by the "data hungriness" of current DL models, which employ millions of samples as input. Therefore, this paucity of data mandates the usage of all available data to the utmost possible extent, which may include the entire lake surface without Chl-a measurements. This is an unorthodox approach that has not yet been explored in the context of water quality estimation.

Accordingly, the contribution of this section is a novel architecture for Chl-a estimation that exploits the unlabeled image locations by the means of in-situ Chl-a measurements in accommodation with the labeled image locations. To this end, a multi-task learning architecture is proposed that employs Chl-a estimation by regression as the main task for labeled samples and acquisition month classification as an auxiliary task for unlabeled samples. The proposed architecture consists of a shared feature extractor, two task-specific branches that produce outputs for each task, and a combined loss function to train the model with these tasks. The proposed model is tested with 3 years of in-situ measurements collected from Lake Balık in Turkey and the corresponding multispectral Sentinel-2 images. In addition to comparing the proposed multi-task model with current water quality estimation studies, ablation studies are also conducted to ensure that the proposed model achieves the best estimation performance.

Finally, this chapter addresses two previously mentioned problems of RS image analysis, namely how to deal with the limited number of labeled data and how to exploit the multi-temporal data by proposing an adapted DL model that benefits from these properties of such images.

3.2. Background

3.2.1. Shallow Approaches

To provide a baseline for estimation of parameters, band ratio algorithms [21, 31, 36, 41] and support vector regressor (SVR) [33, 35, 38] are commonly used in water quality studies. In this section, preliminary information about these two prominent shallow techniques is provided.

3.2.1.1. Band Ratio Algorithms

Surface topography, slope, clouds, and shadows can cause the spectral bands of remote sensing images to have different illuminations for similar surfaces [42]. Therefore, spectral band ratioing is introduced to compensate for these illumination variations, since they are constant between different bands, and to exploit the changes between bands, which are not constant, to obtain information about the characteristics of the surface.

Band ratio algorithms are computationally inexpensive, easy to use, and provide reasonably good performance, which still makes them a prominent choice [29, 31]. However, they have the disadvantage of requiring domain expert knowledge to derive them manually for each sensor and need to be adapted for different water bodies and seasons of the year [43].

3.2.1.2. Support Vector Regression

Support vector regression (SVR) is a popular regression technique developed by Cortes and Vapnik and is an adaptation of support vector machine (SVM), a commonly used supervised machine learning method [44]. Originally, SVR was proposed to convert low-dimensional, linearly inseparable data into a high-dimensional space in which the data is linearly separable using kernel functions. It works by approximating a function with a margin of tolerance between the observed and estimated values.

In water quality analysis, SVR is often used to benchmark the proposed estimation model [33, 35, 38]. Despite SVR's ability to perform well on a small number of data, it is not robust when the data have many outliers because it attempts to approximate a function with as many points as possible within the margin. SVR is also not suitable for large datasets because its training complexity is proportional to the number of samples.

3.2.2. Deep Learning Approaches

Deep learning is attracting growing interest in many fields because of its ability to learn abstract representations from raw data without requiring domain experts to structure the data so that the learning algorithm can detect its patterns, and without

manually designing feature extractors. In this section, background information on two DL techniques, namely the multilayer perceptron (MLP) and the convolutional neural network (CNN), are provided.

3.2.2.1. Multilayer Perceptron

A multilayer perceptron is a deep feedforward neural network consisting of an input layer, several hidden layers, and an output layer. These layers are connected in a feedforward manner so that information propagates in only one direction, from the input layer to the output layer. Each layer of the network consists of multiple neurons, where each neuron transforms its input with a nonlinear activation function and feeds its output to each neuron in the next layer, repeatedly until the output is reached [45]. The connections between neurons are defined with the weights, which eventually leads to an estimate for the output. To train the network, these weights are adjusted using the backpropagation algorithm [46]. It consists in computing the error between the estimated output and the targeted output and updating each weight gradually, starting with the output layer to backward direction, to minimize this error. Backpropagation uses a nonlinear technique called gradient descent that operates by taking the derivative of each activation function. The training phase of the network with the adjustment of weights continues until a desired threshold is reached.

3.2.2.2. Convolutional Neural Network

Convolutional neural networks (CNN) is another deep neural network that extends MLP by exploiting spatial/temporal context and introducing translation invariance [8]. Moreover, compared to MLPs, CNNs have better regularization capability because in the latter architecture, each neuron in one layer is connected to all neurons in the next layer, making them more inclined to overfitting.

Inspired by the human visual system, a CNN model learns information in a hierarchical structure in which the lower layers respond to primitive features, e.g., edges, orientations, intensity changes, while the higher layers combine these lower-level representations from the previous layers to learn more abstract features, e.g., object level. This is achieved by three mechanisms specific to the CNN architecture, namely the local receptive field, shared weight and pooling [8]. The first attribute, the

local receptive field, ensures that each neuron receives input only from the receptive field of the previous layer, which is usually a square-shaped area, e.g., 5×5 pixels. Thus, although the size of the receptive field is limited, the area it represents within the input grows progressively towards the higher layers by combining multiple convolutional layers, contributing to a more abstract representation. The second attribute of the CNN, shared weight, is used to reduce the number of trainable parameters. It follows the assumption that features learned for input position (x, y) are also useful for position $(x + 1, y + 1)$, allowing the same filter to be used within a receptive field. Sharing weights not only drastically reduces the number of trainable parameters, but also introduces the capability of translational equivariance into the CNN, allowing it to detect objects without a fixed position in the image. Finally, pooling within the CNN downsamples the features within a spatial/temporal window to a single value by taking either the maximum or the average of the features. This not only reduces the number of trainable model parameters, but also helps with overfitting since it is not possible to recover the original values within the pooling kernel (unpooling) except for the selected value.

3.3. Methodology

It is known that Chl-a concentration of inland waters is positively correlated with surface water temperature, wind forces, and fertilizer runoff with precipitation, as well as other meteorological aspects [47]. This means that a correlation between Chl-a levels and months of the year should be expected (i.e., higher concentrations at high temperatures and after rainfall). Given this information, the hypothesis is that a network trained to predict the month of the year for the given remote sensing sample (without its Chl-a measurement) will yield a better Chl-a estimation performance as it will also learn information useful for Chl-a estimation.

This section presents the formation of the labeled and unlabeled datasets and the input shape, the proposed architecture, the loss functions of each task, and the final multi-task loss, in that order.

3.3.1. Study Area and Dataset

3.3.1.1. Study Area

Lake Balik situated in Samsun a province in northern Turkey, is used for water quality experiments as presented in Figure 3.1. It is a shoreline lake, close to Black Sea and is part of the Kizilirmak Delta, which has been designated as a Ramsar site since 1998 and hosts 75% of Turkey's bird species during the year [48]. Lake Balik has a total area of 13.9 km² and an average depth of 1.5 meters, which can reach up to 3 meters when it is connected to the sea. Around the lake there are agricultural lands and irrigation canals that cause fertilizer and pesticide runoff and threaten the water quality of the lake.

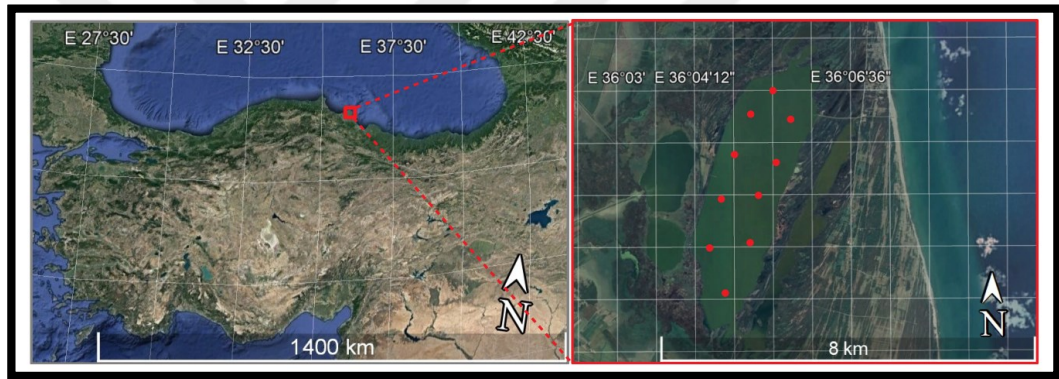


Figure 3.1: Study area, Lake Balik situated in Samsun, a province in northern Turkey and its in-situ measurement sites.

3.3.1.2. Dataset Formation

Samples were analyzed in accordance with EPA-445 [49]. Specifically, 1-liter water samples were collected from a depth of 0.2-0.3 m below the lake surface, filtered with 0.7 μ m Whatman GF/F filter pads, and stored in the dark at -20°C until analyzed in the laboratory. Because algal populations, and therefore Chl-a concentrations, are prone to change within short periods of time, filtration was done under dim light shortly after sampling. Chl-a was extracted for 24 hours using 6 mL of 90% acetone [49]. Chl-a concentration was measured fluorometrically using a TD-7210 Fluorometer (Turner Designs, CA, USA) equipped with a white daylight lamp and a

properly calibrated chlorophyll optical kit after being corrected for pheophytin. The dataset was collected by Prof. Sema Ariman and her team from the Department of Meteorological Engineering Samsun University, Turkey, and provided by them.

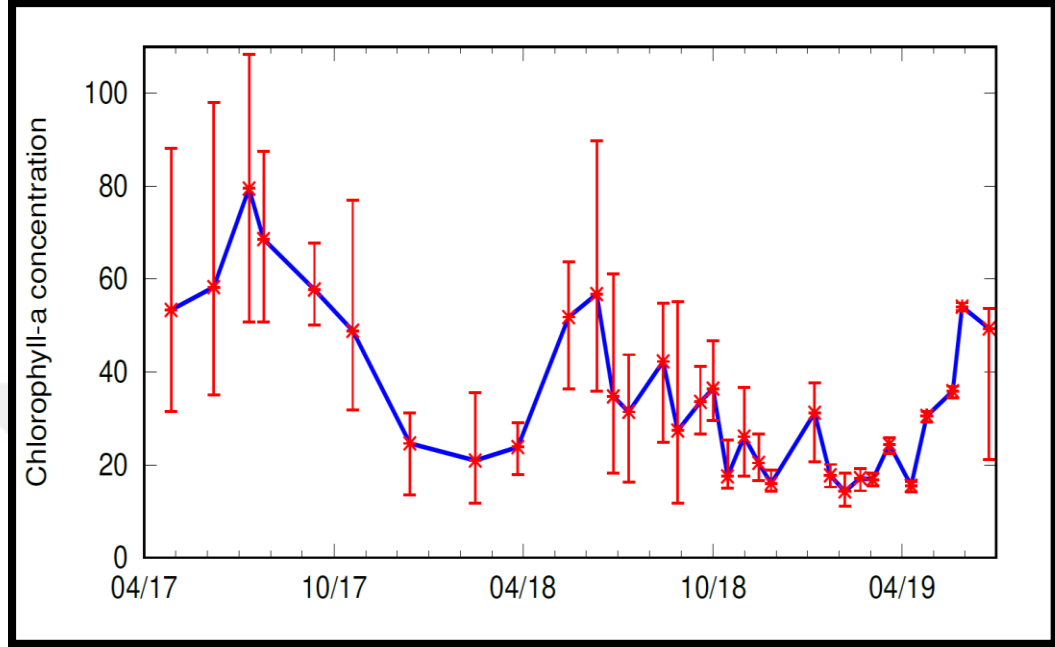


Figure 3.2: The distribution of Chl-a concentration in Lake Balik over time. Each point in the graph represents the average of 10 monitoring sites, while the vertical bar indicates the range of values across sites, where mean = 35.82 $\mu\text{g/L}$, standard deviation = 19.43 $\mu\text{g/L}$, minimum = 11.04 $\mu\text{g/L}$, and maximum = 108.35 $\mu\text{g/L}$.

To capture Chl-a distribution, 10 sites in the lake were selected using a spatial sampling scheme to form a grid structure, each 500 meters apart and 250-300 meters from shore, as displayed in Figure 3.1. At each of these locations in the lake, in-situ measurements were conducted once or twice per month between April 2017 and June 2019 on 32 different dates (7 in 2017, 14 in 2018, and 11 in 2019), depending on weather conditions. Sampling dates were arranged to coincide with the times of the 3-hour satellite overflights. This results in a dataset of $10 \times 32 = 320$ samples.

The results of their laboratory analysis can be found in Figure 3.2, which indicates that the lake is in a hypereutrophic state [49]. Level-1C data from the European Space Agency's Sentinel-2 multispectral imaging instrument are used to acquire the images. For their atmospheric correction, the POLYMER algorithm is used, which works by finding the best fit to the observed reflectance using a model for

ocean water reflectance [50] and has been reported to work well for optically complex waters [51]. Bilinear interpolation is then applied to each band, resulting in a spatial resolution of 10 m/pixel.

It is assumed that the acquisition month of the remote sensing images are available for the proposed approach. Thus, the dataset consisting of 320 samples for which both the in-situ measurements and the acquisition months are known constitutes the labeled dataset. To form the unlabeled samples, in-situ measurement sites and pixels near the lakeshore were excluded, since the former have already been used and . It should be noted that all samples contain the month of acquisition information regardless of the dataset.

3.3.2. Training Procedure

Training with the unlabeled and labeled samples can be done in several ways. One approach might be to first train the model with the unlabeled samples for the month prediction with a higher learning rate until it reaches a meaningful error threshold, and then move on to train the model for the Chl-a estimation task with the labeled dataset, with increasing the learning rate. Although this approach aims to take advantage of the complementary information conveyed by the former dataset, similar to fine-tuning with transfer learning, it could also lead the model to focus on regions in the search space that are far from the optima of the Chl-a estimation problem.

A better approach to this problem would be to train the model simultaneously with both datasets using a shared feature extractor, as a multi-task learning model, with Chl-a estimation as the main task and month prediction as the auxiliary task, as displayed in . The feature extractor shared between these two tasks can help the model learn features that are useful for both tasks, eventually directing them to reinforce each other. In addition, this approach can alleviate the data hunger of the DL model using the entire pixels of the lake surface through the unlabeled dataset, introducing 2.3 million samples, and it can take advantage of the regularization effect that may come along with these additional samples, making the model robust to overfitting.

3.3.3. Input Shape

It is well known that providing the spatial context for the pixel under study has a positive effect on the model's performance by exploiting the correlation between the neighbors of the pixel and their spectral information [34, 35]. Therefore, 3D patches of form $b \times k \times k$, centred on each input pixel are used as samples, where b is the number of multi-spectral bands of the image stacked on top of each other and k is the spatial neighbourhood size.

3.3.4. Network Architecture

The 2D convolutional neural network (CNN) architecture of [35] was adapted for the shared feature extractor part for the proposed approach. It includes four convolutional layers without pooling layers, since the input patch size is already small at 3×3 . In addition, two fully connected (FC) layers for each task are added on top the convolutional layers to produce their outputs.

A 1-pixel padding is applied before each convolution layer to maintain the size of 3×3 patches. Tanh is used as the activation function for all convolutional layers. Then, the output of the last convolutional layer is passed to each task-specific branch as a shared feature map. These task-specific branches, which consist of two FC layers, are identical except for the number of neurons in the last layer. In the last layer, the regression branch contains 1 neuron to estimate the Chl-a value, while the classification branch contains 12 neurons to predict the acquisition month.

The proposed network architecture is presented in . The number of shared and task-specific layers and their learning rates are determined by the ablation studies reported in the next section.

3.3.5. Loss Functions

More formally, given a labeled dataset $\mathcal{D}_L = \{(x_i, y_i, m_i)\}$ where each patch x is labeled with its month of acquisition m for $x \in \mathbb{R}^{b \times k \times k}$ and $m \in \{1, 2, \dots, 12\}$ and has a Chl-a concentration $y \in \mathbb{R}$. In addition, an unlabeled dataset $\mathcal{D}_U = \{(x_i, m_i)\}$ is formed such that each patch x is labeled only with its acquisition month m for $x \in \mathbb{R}^{b \times k \times k}$ and $m \in \{1, 2, \dots, 12\}$. The labeled dataset $\mathcal{D}_L$ was divided into two parts to

form the training set $\mathcal{T}$ and the test set $\mathcal{H}$, where $\mathcal{T}, \mathcal{H} \in \mathcal{D}_L$ and $|\mathcal{T}| + |\mathcal{H}| = |\mathcal{D}_L|$ to test the proposed multi-task learning approach.

The goal is to learn a model Φ^θ with parameters θ to correctly estimate the Chl-a concentration of a sample x as $y = \Phi^\theta(x)$. Unlike existing approaches in the literature that use a single-task learning neural network architecture [34–38, 52], the proposed approach employs a multi-task learning architecture to test the hypothesis that a network trained to predict the acquisition month can provide better Chl-a estimation performance. The proposed architecture consists of a feature extractor Φ_{fe}^θ shared by the tasks and their task-specific branches $\Phi_{reg}^{\theta, FC}$ and $\Phi_{cls}^{\theta, FC}$, so that the model uses them to produce a Chl-a estimation, $\Phi_{reg}^\theta = (\Phi_{fe}^\theta, \Phi_{reg}^{\theta, FC})$, for the main task regression and a month prediction, $\Phi_{cls}^\theta = (\Phi_{fe}^\theta, \Phi_{cls}^{\theta, FC})$, for the auxiliary task classification.

3.3.5.1. Mean Squared Error Loss

The mean square error (MSE) loss function is used to be consistent with water quality estimation studies [35, 36, 38]. To evaluate this loss, the training set $\mathcal{T}$ of the labeled dataset $\mathcal{D}_L$ is used, as only this contains the observed Chl-a values, as follows:

$$\mathcal{L}_{reg}(x_{(i)}, y_{(i)}; \Phi_{reg}^\theta) = \frac{1}{|\mathcal{B}_{reg}|} \sum_i \left(y_{(i)} - \Phi_{reg}^\theta(x_{(i)}) \right)^2 \quad (3.1)$$

where (i) denotes the i_{th} batch of the training set $\mathcal{T}$, $|\mathcal{B}_{reg}|$ is the batch size of the main regression task, and $y_{(i)}$ and $\Phi_{reg}^\theta(x_{(i)})$ denote the observed and estimated Chl-a concentrations.

3.3.5.2. Cross-Entropy Loss

For the auxiliary month prediction task, a standard classification loss is used to exploit the unlabeled image locations. This loss is evaluated for both the training set

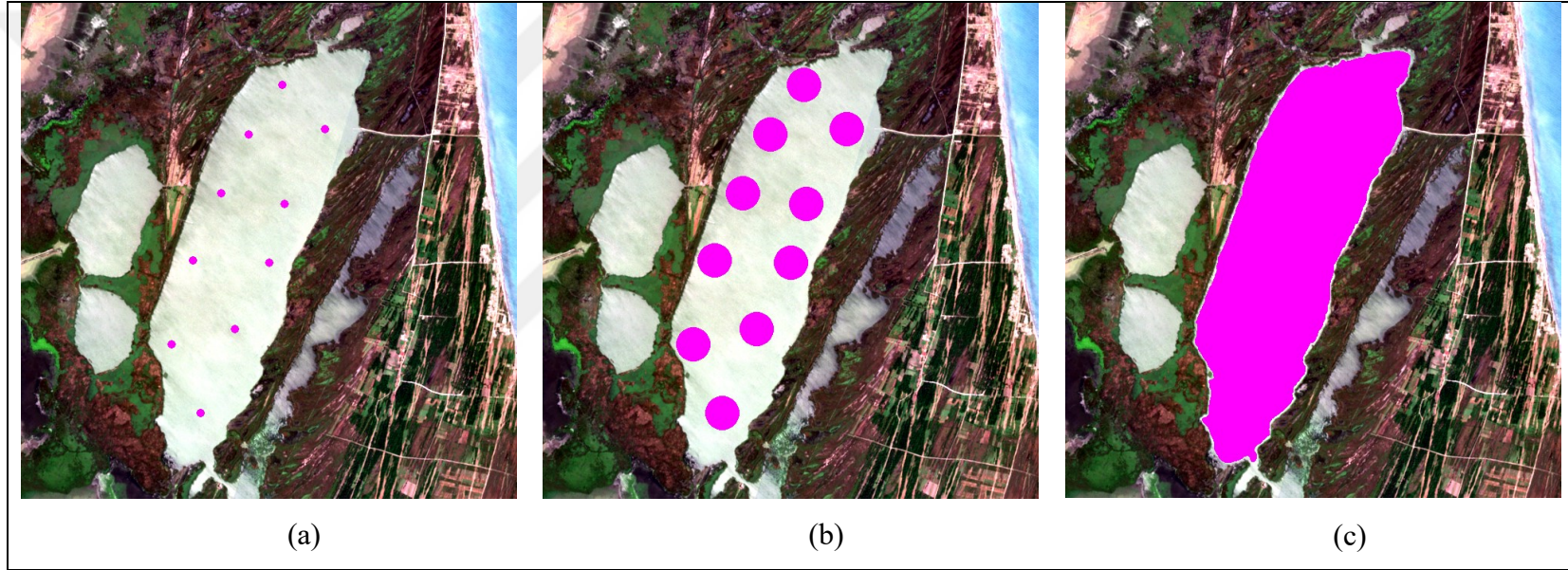


Figure 3.3: The pink areas show the lake masks used to employ unlabeled samples in the ablation studies centered on each Chl-a monitoring site with (a) an 11 pixel diameter circle, (b) a 45 pixel diameter circle, and (c) the entire lake surface, yielding 28,160, 503,040, 2,364,928 unlabeled samples in 32 images, respectively.

Table 3.1: The number of samples per month for each dataset.

	Month												
	1	2	3	4	5	6	7	8	9	10	11	12	Total
Unlabeled	148K	222K	222K	222K	222K	296K	222K	148K	148K	296K	148K	74K	2365K
Labeled	20	30	30	30	30	40	30	20	20	40	20	10	320

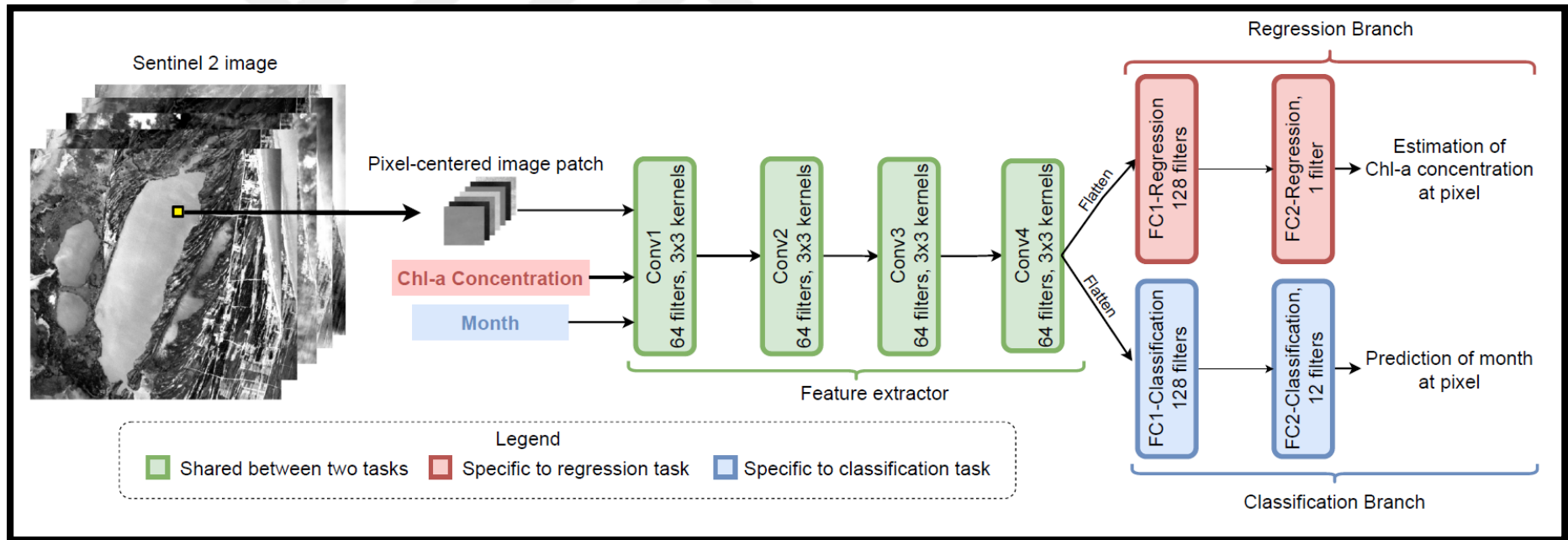


Figure 3.4: The outline of the proposed multi-task learning network architecture, where Conv and FC denote the convolutional layer and the fully connected layer, respectively.

$\mathcal{T}$ of the labeled dataset $\mathcal{D}_L$ and the unlabeled dataset $\mathcal{D}_U$, since the samples of both datasets contain the date of image acquisition, as follows:

$$\mathcal{L}_{cls}(x_{(i)}, m_{(i)}; \Phi_{cls}^\theta) = -\frac{1}{|\mathcal{B}_{cls}|} \sum_i m_{(i)} \log(\Phi_{cls}^\theta(x_{(i)})) \quad (3.2)$$

where (i) denotes the i_{th} batch of $\mathcal{D}$ for $\mathcal{D} = \mathcal{T} \cup \mathcal{D}_U$, $|\mathcal{B}_{cls}|$ is the batch size of the auxiliary classification task, $m_{(i)}$ and $\Phi_{cls}^\theta(x_{(i)})$ denote the observed and predicted acquisition months.

3.3.6. Multi-Task Loss

Multi-task learning consists of optimizing the model with respect to the objectives of multiple tasks. Therefore, the design of the multi-task loss is of paramount importance to the model's performance. A naive attempt would be to simply add the task losses to obtain a final multi-task loss: $\mathcal{L}_{MT} = \mathcal{L}_{reg} + \mathcal{L}_{cls}$. Although this is straightforward, the values produced by $\mathcal{L}_{reg}$ and $\mathcal{L}_{cls}$ may vary significantly since they belong to different loss families. This would result in the model learning the task whose loss value is dominant in the sum but underperforming in the other task.

A better attempt would be formulating the multi-task loss as a weighted combination of task losses as $\mathcal{L}_{MT} = c_{Reg}\mathcal{L}_{Reg} + c_{Cls}\mathcal{L}_{Cls}$. [53] observed that with the selection of optimal values for these coefficients, the shared network achieved a better performance for both tasks compared to the case where each task is independently trained on the network. So, while it also assures the optimization of the task whose loss value is not non-dominant in the final loss compared to the prior approach, it leads to two problems. First, performance of the model is tightly bounded by the selection of the coefficients c_{Reg} and c_{Cls} , and finding the optimal values of these coefficients is computationally expensive, as it includes training the model with different set of coefficients and comparing their performances. Second, it does not take into account the changes in loss values during the training phase. These changes in loss values over epochs may require updating the coefficients to ensure a gradually decreasing total loss.

This leads to the idea of learning the loss weights of the tasks through the model. This can be done by adding these loss coefficients c_τ to the trainable parameters of the network θ , as $\omega_\tau = (\theta, c_\tau)$. In this proposed approach, an implementation [54] of [55] is used to define the multi-task objective function to learn the task coefficients c_{Reg} and c_{Cls} , as follows,

$$\mathcal{L}_{MT}(x, y_\tau; \omega_\tau) = \sum_{\tau \in T} \frac{1}{2c_\tau^2} \mathcal{L}_\tau(x, y_\tau; \omega_\tau) + \ln(1 + c_\tau^2) \quad (3.3)$$

where $\tau \in T$ and $T = \{Reg, Cls\}$, $y_\tau = y$ for $\tau = Reg$ and $y_\tau = m$ for $\tau = Cls$.

In the next section, numerical results of the proposed approach and its alternatives are presented. In addition, the effects of multi-task learning and the details of the architectural design decisions, such as the size of the input patch, the length of the feature extractor shared by the tasks, the method of weighting the task losses, and finally the effect of the number of unlabeled samples used in the training on the performance of Chl-a estimation are examined based on the ablation studies.

3.4. Experiments & Results

This section presents the design of the experiments and their results to test the hypothesis of whether a network trained to predict the acquisition month in a multi-task learning setting for a given sample without its Chl-a value can provide better Chl-a estimation performance.

3.4.1. Data Split

In the water quality estimation literature, single random splits are commonly used to measure model performance [31, 33, 37, 38, 56, 57]. However, because these datasets are relatively small (e.g., hundreds of samples) and are further divided into training and test sets, training with these datasets could lead to overfitting the model to the data and thus deceptively high estimation performance. Therefore, this problem was addressed by employing a 10-fold cross-validation strategy.

In order to tune the network's hyperparameters, the training set is split into a validation set that has the same size with the test set. Therefore, in each fold, 256

samples (80%) are used for training, 32 samples (10%) for validation and 32 samples (10%) for testing. After the hyperparameters of the network are selected, the proposed model is trained without generating the validation set as the dataset size is relatively small. This yields of a final split with 288 samples (90%) for training and 32 samples (10%) for testing in a 10-fold setting.

3.4.2. Alternative Methods

The performance of the proposed multi-task learning approach is compared with the different alternatives, including its single-task version [35], the support vector regression (SVR) with a radial basis function [44], the multilayer perceptron (MLP) with a different number of hidden layers with the same number of trainable parameters as the proposed approach, a morphology-based spectral-spatial approach (EFAL) [58], a CNN architecture adapted for regression proposed in [34], and finally a spectral band ratio approach specialized to Chl-a estimation [31]:

$$\frac{R_{705}}{R_{665}}, (R_{665}^{-1} - R_{705}^{-1}) \times R_{740}, \frac{R_{490}}{R_{443}} \quad (3.4)$$

The results are reported in Table 3.2.

3.4.3. Hyperparameters

Since the training sets of the labeled and unlabeled datasets are very different in size, 320 and 2.3 million samples, respectively, it is reasonable neither to use the same batch size for them nor to adjust their batch sizes to have the same number of iterations on their datasets, since this means that the batch size for the labeled dataset will be as low as 1.

Instead, batches of the training set of the labeled dataset are reused to obtain the same number of iterations for both datasets, since the labeled dataset contains the essential Chl-a information. The learning rate of the network was set to 10^{-4} . The root mean square propagation (RMSprop) algorithm is employed to minimize multi-task learning loss. Training was continued for 200 epochs, which is determined by experiments. The batch sizes are set to 64 and 12,318 for the labeled and unlabeled

datasets, respectively. The same parameters and hyperparameters are used in all experiments, except for the ablation studies, whose ablation values are reported in the corresponding tables. Finally, the coefficient of determination (R^2), root mean square error (RMSE), and mean absolute error (MAE) are used to measure the performance of Chl-a estimation as commonly found in the water quality estimation literature [31, 41, 52, 58].

3.4.4. Results

According to the results reported in the Table 3.2, the band ratio approach achieved performance close to the DL models as derived for estimating Chl-a concentrations of inland and coastal waters from Sentinel-2 images [31]. SVR [44] is outperformed by all DL models, as expected, consistent with the observed paradigm shift from ML models to DL models. EFAL [58] provided the second best results thanks to its ability to describe pixels with a hierarchical representation that also conveys their spatial and spectral information. The CNN architecture of [34] adapted for regression achieves similar performance to MLP and single-task CNN models, as their performance is bounded by the limited number of labeled samples. The performance of MLP models [36] increases up to 4 hidden layers and then starts to decrease, although all MLP models have approximately the same number of trainable parameters as the proposed multi-task learning model. A possible explanation would be that the model starts to overfit after 4 hidden layers. The proposed multi-task learning model trained with both labeled and unlabeled data outperforms its single-task learning regression version [35]. Thus, although the unlabeled dataset on which the model was trained does not contain Chl-a values for its samples, the use of seasonal variations through the auxiliary acquisition month prediction task appears to improve the performance of the main Chl-a estimation task, which increased by almost 8 percentage points.

In addition to the quantitative results, the qualitative results of the proposed multi-task learning approach are provided by creating the Chl-a estimation maps (one for each season) by training the model with the samples of the other years and presented in Figure 3.5.

Table 3.2: Mean Chl-a estimation results across 10 folds of cross-validation with the best results in bold where, MAE, RMSE, and * denote the coefficient of determination, mean absolute error, root mean square error, and the proposed approach, respectively.

Method	Dataset	$R^2 \times 100$	MAE	RMSE	Model Parameters
Band ratio	Labeled	71.37	0.147	0.199	-
SVR	Labeled	51.82	0.193	0.27	-
EFAL	Labeled	77.12	0.151	0.221	-
CNN	Labeled	73.06	0.146	0.197	340 K
MLP (2 layers)	Labeled	64.99	0.176	0.233	267 K
MLP (3 layers)	Labeled	72.47	0.152	0.204	267 K
MLP (4 layers)	Labeled	75.18	0.139	0.194	267 K
MLP (5 layers)	Labeled	72.11	0.144	0.206	267 K
Single-task CNN	Labeled	76.09	0.142	0.19	192 K
Multi-task CNN*	Labeled+unlabeled	83.89	0.107	0.154	267 K

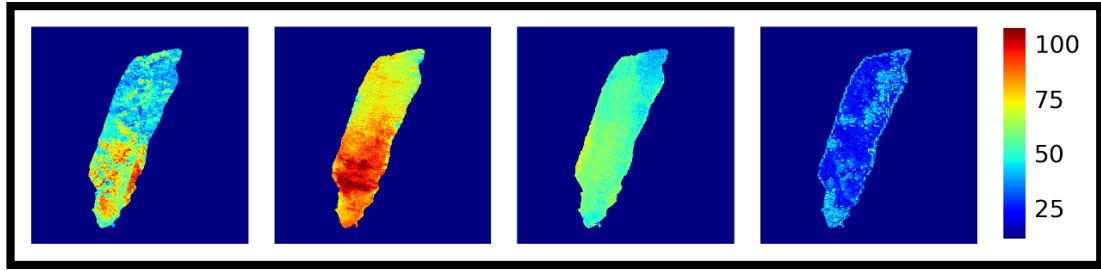


Figure 3.5: Chl-a estimation maps in $\mu\text{g/L}$ created by the proposed multi-task learning model for spring, summer, fall, and winter 2017.

Finally, to understand the relationship between the observed and estimated Chl-a values, it is visualized and presented in Figure 3.6. It shows that the estimated and observed values exhibit a linear relationship.

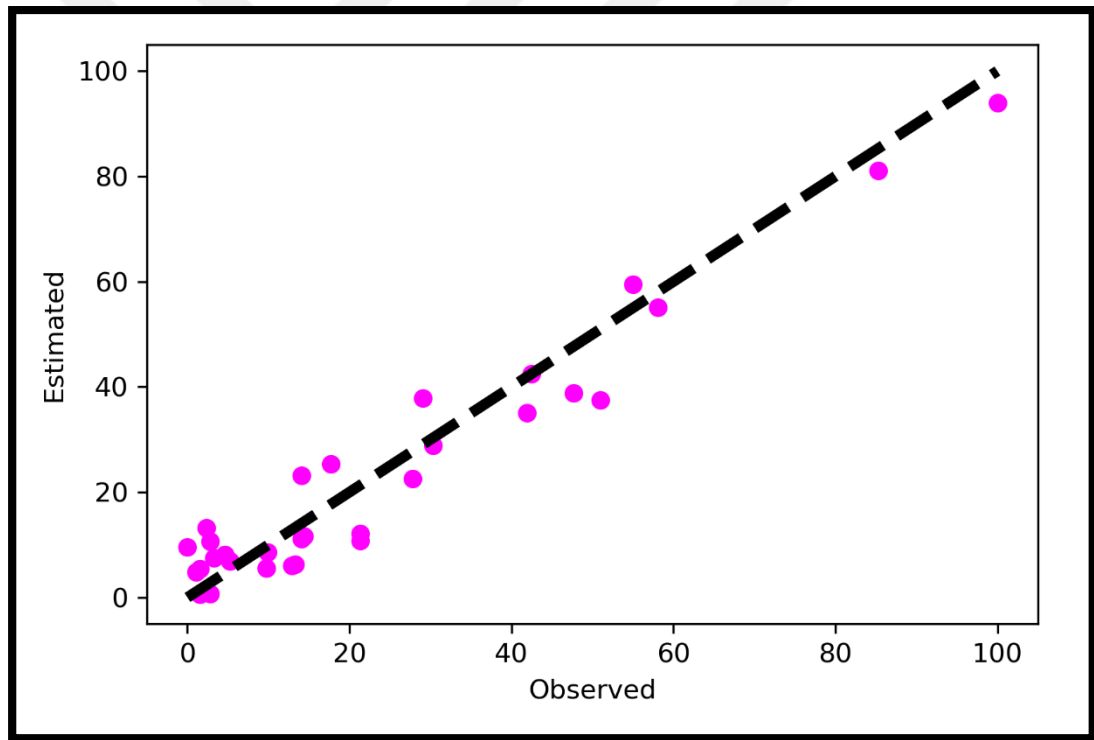


Figure 3.6: The relationship between observed Chl-a values and those estimated by the proposed multi-task learning model.

3.4.5. Ablation Studies

This section investigates and reports the effect of the following design parameters on the performance of Chl-a estimation: the number of shared layers within the architecture, the shape of the input patch, the number of unlabeled samples, and the task loss weighting scheme.

Since the adapted network architecture consists of 6 layers, 5 multi-task learning models were created, experimenting from 1 to 5 layers for the shared feature extractor, starting with the first layer. The results are reported in Table 3.3. According to these results, the best performance is obtained with the architecture consisting of 4 shared hidden layers. This implies that the model benefits from sharing features obtained from the convolutional layers, as they produce a relatively lower level of abstraction that can be useful for the main task of Chl-a estimation, however separate fully connected layers are needed for each task.

Table 3.3: Chl-a estimation performance of the proposed multi-task learning model under 10-fold cross validation with different number of shared layers, where R^2 , MAE, and RMSE denote the coefficient of determination, mean absolute error, and root mean square error, respectively. The best results are indicated in bold (patch size = 3×3).

Shared Layers	1	2	3	4	5
$R^2 \times 100$	83.57	82.81	83.38	83.89	83
RMSE	0.156	0.159	0.158	0.154	0.16
MAE	0.108	0.11	0.109	0.107	0.113

Experiments are also conducted to determine the best performing input patch size and update the first layer of the network accordingly. Patch sizes from 1×1 to 9×9 are considered to observe the effects of spatial neighborhood on performance. Patches larger than 9×9 are not studied because they result in an area of 8.1 km^2 , which is larger than half of the lake area studied, 13.9 km^2 . According to Table 3.4, the best results are obtained with a patch size of 3×3 . This means that up to a certain point the model benefits from the spatial information conveyed in the patches, but after that the performance does not increase because it is a small lake and larger patches can introduce more variation than is needed to represent a sample.

Table 3.4: Chl-a estimation performance of the proposed multi-task learning model under 10-fold cross validation with different patch sizes, where R^2 , MAE, and RMSE denote the coefficient of determination, mean absolute error, and root mean square error, respectively. The best results are indicated in bold (number of shared layers=4).

Patch Size	1×1	3×3	5×5	7×7	9×9
$R^2 \times 100$	75.44	83.89	81.48	82.19	83.45
RMSE	0.194	0.154	0.167	0.163	0.158
MAE	0.134	0.107	0.114	0.112	0.105

In addition, to observe the contribution of the number of unlabeled samples to the performance of the main Chl-a estimation task, 4 multi-task learning models were trained with a gradually increasing number of unlabeled samples, starting with none and ending with the entire surface of the lake. The lake masks used to employ these unlabeled samples are presented in Figure 3.3 and their results are listed in Table 3.5. Accordingly, the model trained with the highest number of samples, namely the entire lake surface, achieved the best estimation performance, as one would expect given the data hunger of the DL models. This finding is particularly promising for lakes that are geographically difficult to access.

Table 3.5: Chl-a estimation performance of the proposed multi-task learning model under 10-fold cross validation with different number of unlabeled samples, where R^2 , MAE, and RMSE denote the coefficient of determination, mean absolute error, and root mean square error, respectively. The best results are indicated in bold (number of shared layers = 4, patch size 3×3).

Unlabeled Samples	0	28160	503040	2364928
$R^2 \times 100$	76.63	82.41	83.03	83.89
RMSE	0.187	0.161	0.159	0.154
MAE	0.137	0.118	0.118	0.107

Similarly, the effect of the number of unlabeled samples used on the performance of the auxiliary classification task is also examined. The same lake masks from Figure 3.3 are employed to train 4 multi-task learning models, whose results and confusion matrices are reported in Figure 3.7. As before, the best classification performance was also obtained by using the whole lake surface.

Lastly, two experiments are conducted to evaluate the contribution of the employed task loss weight learning scheme by comparing it with the direct sum of task losses. The former approach resulted in a better Chl-a estimation performance compared to the latter one as presented Figure 3.6, however it was only a small gain. The next section presents the discussion and limitations of the proposed approach.

Table 3.6: Chl-a estimation performance of the proposed multi-task learning model under 10-fold cross validation with different task loss weighting schemes, where R^2 , MAE, and RMSE denote the coefficient of determination, mean absolute error, and root mean square error, respectively. The best results are indicated in bold (number of shared layers=4, patch size 3×3).

Task Loss Weight	Sum	Weight Learning
$R^2 \times 100$	83.45	83.89
RMSE	0.156	0.154
MAE	0.113	0.107

3.5. Discussion & Limitations

The experiments presented in the previous sections demonstrate that the proposed approach is a promising alternative to its counterparts due to its practicality and performance. The proposed multi-task learning model achieves better performance in Chl-a estimation compared to its single-task learning regression version. This is made possible by using an additional 2.3 million samples, which in turn leads to such an increase. The use of these additional samples as input can introduce a regularization effect into the model that can help it learn a better representation of the data, which can explain this increase in performance. However, both the study area and the proposed approach have their own limitations that need to be considered.

First, even though the precautions are taken to ensure the reliability of the proposed approach and to minimize the effect of overfitting, namely using cross-validation on training for the former and employing of additional samples for the latter, these results are limited with the samples of a single water body. Therefore, it is unknown that this estimation performance of the proposed model can be reproduced

with the samples of lakes from different geographical locations, such as ones from high latitudes and altitudes.

Second, although the Chl-a measurements of the in-situ samples of Lake Balik show a wide range of variation between 11.04 $\mu\text{g/L}$ and 108.35 $\mu\text{g/L}$, the lake is in a hypereutrophic state. Therefore, the validity of the obtained results for the lakes with different characteristics such as mesotrophic and oligotrophic is unknown.

3.6. Conclusion

In this chapter, a semi-supervised multi-task learning CNN architecture is proposed for water quality estimation exploiting unlabeled satellite images. It addresses the limited availability of labelled data by introducing image acquisition month classification as an auxiliary task, leading to the utilization of 2.3 million samples in addition to the main Chl-a estimation task.

The proposed approach was tested with the Lake Balik dataset and resulted in a significant improvement in Chl-a estimation compared to its alternatives. This is possible because the seasonal variations of Chl-a can be exploited with the auxiliary month classification task.

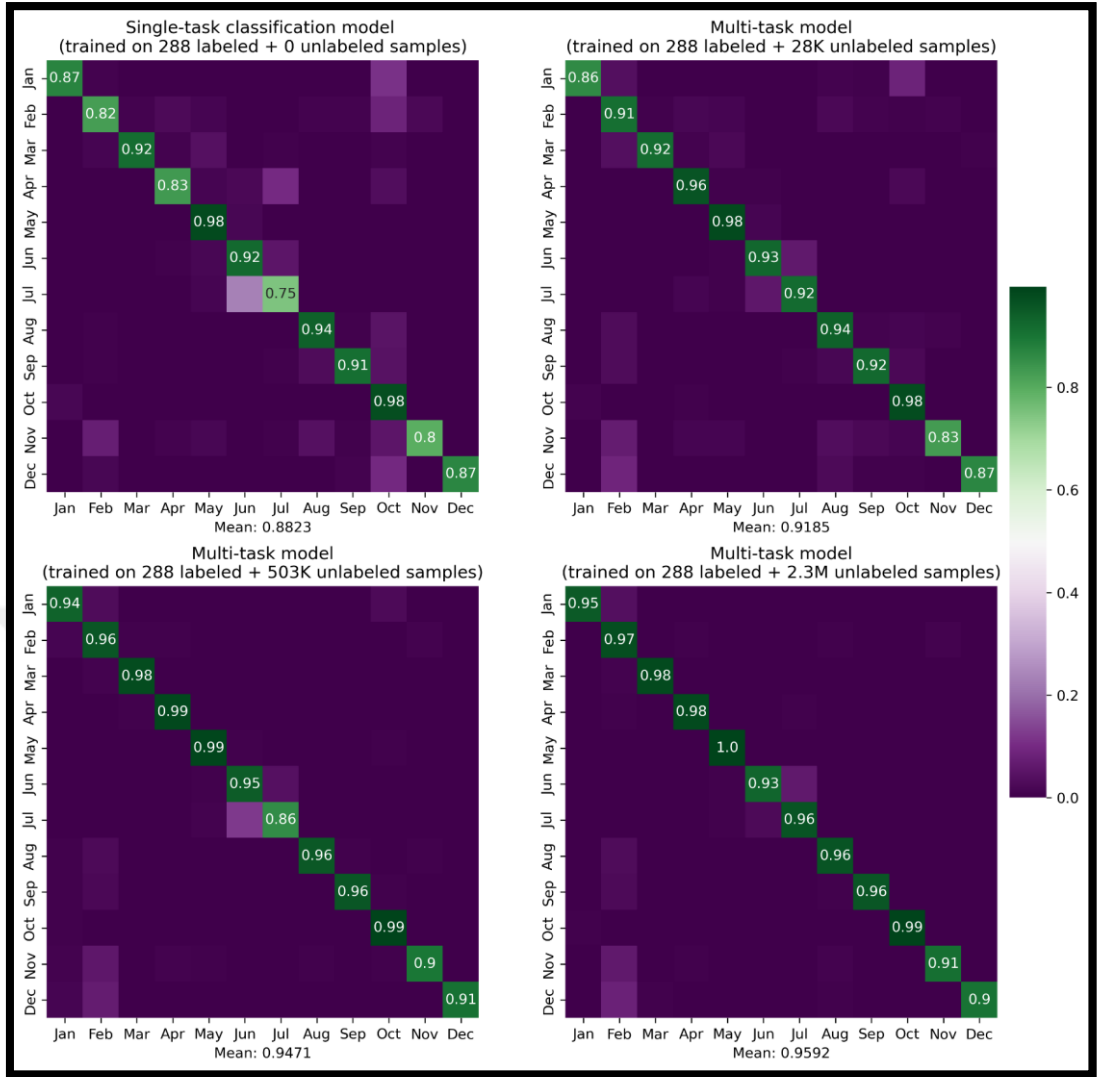


Figure 3.7: Acquisition month prediction performance of the proposed multi-task learning model with different number of unlabeled samples.

4. PROCESSING ATTRIBUTE PROFILES AS SCALE SERIES FOR REMOTE SENSING IMAGE CLASSIFICATION

4.1. Problem Definition

Pixel-level classification of RS images is still a challenging problem as advances in image acquisition platforms lead to higher resolutions in the spatial, temporal, and spectral domains, as discussed in more detail in Chapter 1. These problems necessitate powerful spectral and spatial descriptors for the classification of RS images.

In Chapter 1, it was discussed that DL is known for its state-of-the-art performance in several domains, making it a prominent choice for this task. However, the performance of a DL model is tightly bounded by its need for a large number of labeled samples and the reliability of the labeling. Therefore, shallow strategies are still widely used in the RS literature for pixel-level classification [59]. Morphological attribute profiles (APs) are the apparent spatial-spectral descriptors that perform well even with a limited number of samples.

APs have been extensively studied since their introduction by Dalla Mura [12] in 2010. Some examples of these studies include combining APs with dimension reduction techniques [60, 61], techniques to prevent APs' need of manual thresholds [15, 62, 63], their implementation with various tree representations [47–51].

However, APs have their issues, such as threshold and attribute selection, which is not straight-forward to accomplish and can directly affect their performance. To get the best of both worlds, the combination of APs with DLs has been explored as an alternative approach to shallow or deep methods. [69] studied the combination of Gabor filters and CNNs, while [70] explored the combination of attribute filtered images through CNNs. From this perspective, APs can be considered as mid-level features as they contain more abstract information about the data compared to the raw input. Hence, training a DL model with these mid-level features can alleviate the model's data hunger by providing it with an already higher-level representation of the data than if it were trained from scratch with the same limited amount of data, but in raw form.

This chapter takes a similar approach by exploring the combination of shallow and deep techniques for pixel-level description. The idea is that the values of a pixel

in AP form a numerical sequence, as it is calculated by thresholding the image with different values to represent different scales, and therefore can be considered as a scale-series. Thus, to exploit this sequential characteristic of APs, a long-short term memory (LSTM) network was chosen, an DL architecture specifically designed to process sequential data [71].

In addition, this chapter highlights a common validation error found in the AP literature, where only one tree is calculated to represent the entire scene. The drawback of this approach is that the training and test samples are described by the same tree and some of the samples -pixels- of these two sets may end up in the same connected component, which means that the same features are used both for developing the model and for testing it. As might be expected, this leads to a deceptively high classification performance and a poor indication of generalization performance [72].

To overcome this issue, the proposed model was evaluated on two RS datasets with two additional setups that ensure that separate tree representations are calculated for the training and test sets, in addition to evaluating the model with the original overlapping versions of the datasets. The results show that classification performance indeed decreases, indicating poor generalization capacity when the non-overlapping setups of the datasets are employed. However, the proposed AP-based LSTM model is not affected by this as much as the other models and achieves a significant increase in classification performance.

Finally, this chapter addresses three previously mentioned problems of RS image sequence analysis, namely dealing with the multi-scale appearance of phenomena, the heterogeneity of RS images, and the high number of spectral bands, by proposing an adapted DL model that can exploit the characteristics of such images.

4.2. Background

This section provides the background information necessary to understand the proposed AP-based LSTM model approach. It starts by recalling the hierarchical tree representation, then expounds how APs are calculated using this representation, and finally introduces the original LSTM network.

4.2.1. Hierarchical Tree Representation

Depending on the domain of application, different types of image representations exist. In order to identify the most suitable one for the particular application, factors such as the accuracy and size of the representation, the number of elements and the relationships between them must be taken into account [16].

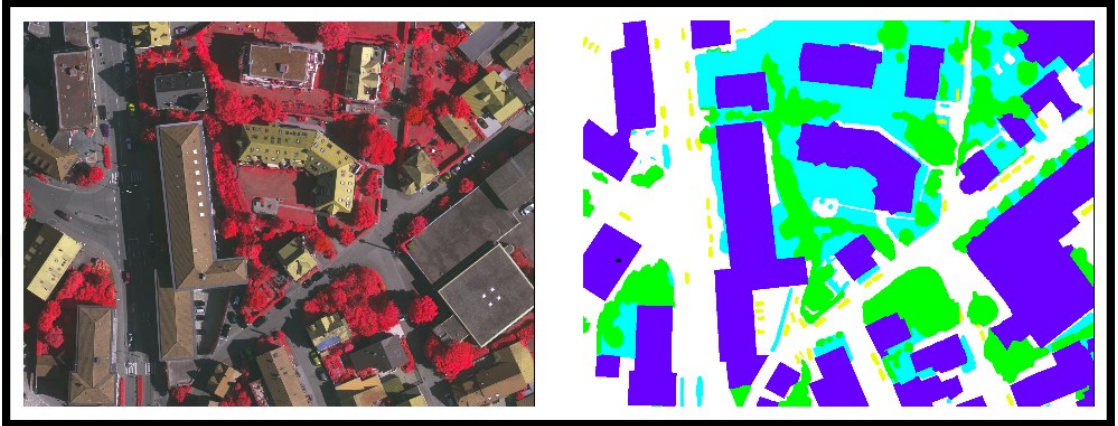


Figure 4.1: The scale difference in size between the instances of cars (yellow) and vegetation (green) classes can be observed, with both images coming from the Vaihingen Dataset.

The hierarchical representation lends itself to analysis of RS images as it can represent the image as a composition for each of the various scales allowing both the coarse (e.g., instances of the vegetation class in Figure 4.1) and fine (e.g., instances of the car class in Figure 4.1) levels of analysis. This is very convenient for analyzing RS images due to their multi-scale nature, as phenomena often appear at very different scales within the same scene, having been captured from high altitude, as displayed in Figure 4.1.

More specifically, the hierarchical representation of the image consists of the stacking of the unions of image regions in different scales, hence the name hierarchical [16]. The component tree structure is a prominent choice for creating the hierarchical representation because each node of the tree represents a region in the image, and the edges between tree nodes correspond to inclusion relations between image regions of different scales. Consequently, the leaves of the tree represent the finest regions, and moving towards the root node gives coarser representations as the regions accumulate

while the root corresponds to the whole image. Depending on their construction, component trees are divided into two families [16]:

- Inclusion Trees

The leaves of the tree consist of the local extrema values (minima or maxima) of the image (displayed for the min-tree in Figure 4.2-b and the max-tree in Figure 4.2-c). The tree grows toward the root, where new nodes are formed by adding one or more same intensity pixels (flat zones) to the regions represented by the existing nodes. This inclusion creates a parent-child relationship between the nodes, resulting the root node to cover the entire image domain. As the tree grows in an accumulative fashion, the leaves do not form a full partition of the image.

- Partition Trees

The leaves and any level of the tree form a full partition of the image, where the leaves yield to a segmentation in the finest detail (displayed for α -tree in Figure 4.2). New nodes are formed by merging two or more child nodes whose regions are adjacent, again creating a parent-child relationship between them. Similar to the inclusion tree, the root node includes all regions and thus covers the entire image pixels.

The main difference between these two tree families lies in the generation of leaves and the formation of parent nodes. The leaves of a partition tree form an over-segmentation of the image, while the leaves of an inclusion tree contain only the local extreme values and do not cover the entire image domain. Similarly, a partition tree grows from the leaves to the root by merging two or more adjacent nodes to form a parent. On the other hand, an inclusion tree grows by adding flat zones to existing nodes. To represent these two tree families, one member is selected from each of them in this chapter, namely the min/max tree and the α -tree for inclusion trees and partition trees, respectively, the details of which are explained below:

- Min-Max Tree

The min-tree is the dual of the max-tree, both belonging to the family of inclusion trees [73, 74]. The min-tree aims to represent the dark regions in the image, hence its leaves correspond to the local minima of the image. The intermediate nodes

are formed by merging existing nodes or adding new connected components to existing nodes, as displayed in Figure 4.2-b. At the root node of the tree, all connected components are merged into one that covers the entire image. Similar to the min-tree, the max-tree represents the bright structures in the images whose leaves correspond to the local maxima Figure 4.2-c. The calculation of the max-tree is equivalent to taking the dual of the input image and calculating its min-tree.

- α -Tree

The α -tree belongs to the family of partition trees. Unlike the members of the inclusion tree, which aim to represent the extrema values of the image, they correspond to the regions with all kinds of intensities. The initial partition of the image is created using flat zones and corresponds to $\alpha = 0$ regions which means that the difference between the intensities of neighboring pixels within the same regions is at most $\alpha=0$ for the leaves, as displayed in Figure 4.2. Similarly, the intermediate nodes are formed by merging the regions of nodes whose neighboring pixels have an intensity difference of at most α . The value of α increases toward the root of the tree and is largest at the root, since it covers all image pixels. The highest value of α can be at most the difference of the range values of the image intensities.

4.2.2. Attribute Profiles

Attribute profiles (APs) are efficient multi-scale image descriptors. They are an extension of morphological profiles (MPs) and have been proposed to overcome the limitations of MPs due to the use of fixed-shape structuring elements, which cannot describe the intensity-related features and lead to poor representation of phenomena due to strong shape constraints [12].

APs overcome the above problems by using attribute filters (AFs), which are flexible and can adapt to phenomena in different shapes. AFs can describe image regions based on the selected attribute(s) by evaluating them against a criterion with different attribute thresholds, which allows for multi-level analysis [75].

More precisely, AFs are connected morphological filters that operate on connected components (CCs), which correspond to tree nodes in the hierarchical tree representation, by evaluating each CC against a criterion [12]. The criterion is a

2	2	2	2	1	1	1
2	4	4	2	1	0	1
2	4	4	2	1	0	1
2	2	2	2	1	1	1
3	3	3	3	3	3	3
1	1	2	2	2	0	0
0	0	0	0	0	0	0

(a)

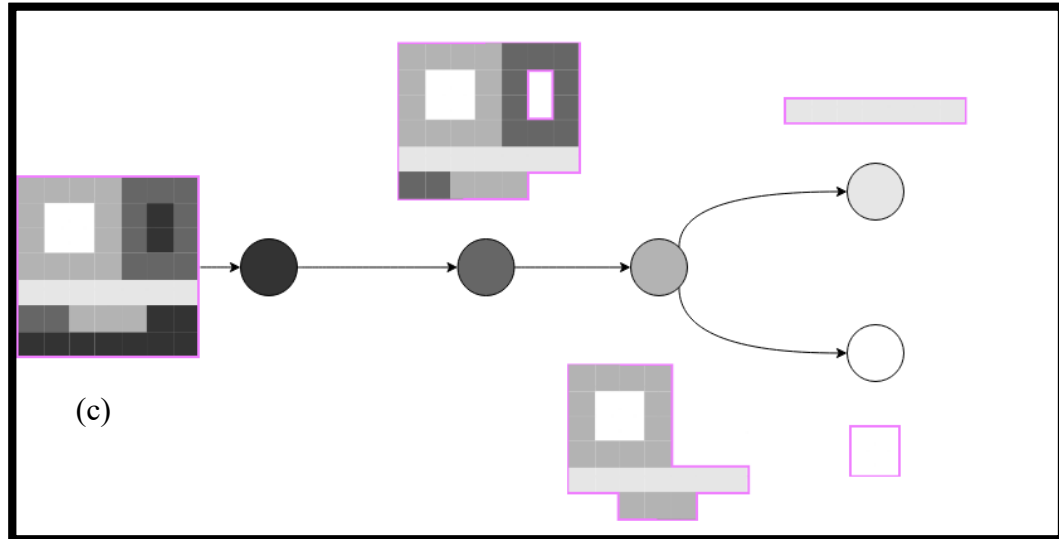
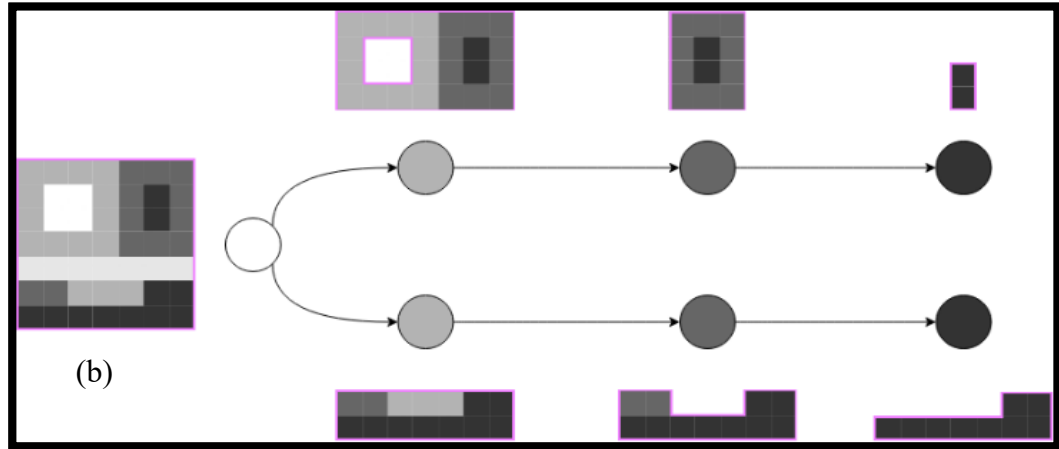


Figure 4.2: The input grayscale image (a), its min-tree (b) and max-tree (c) representations.

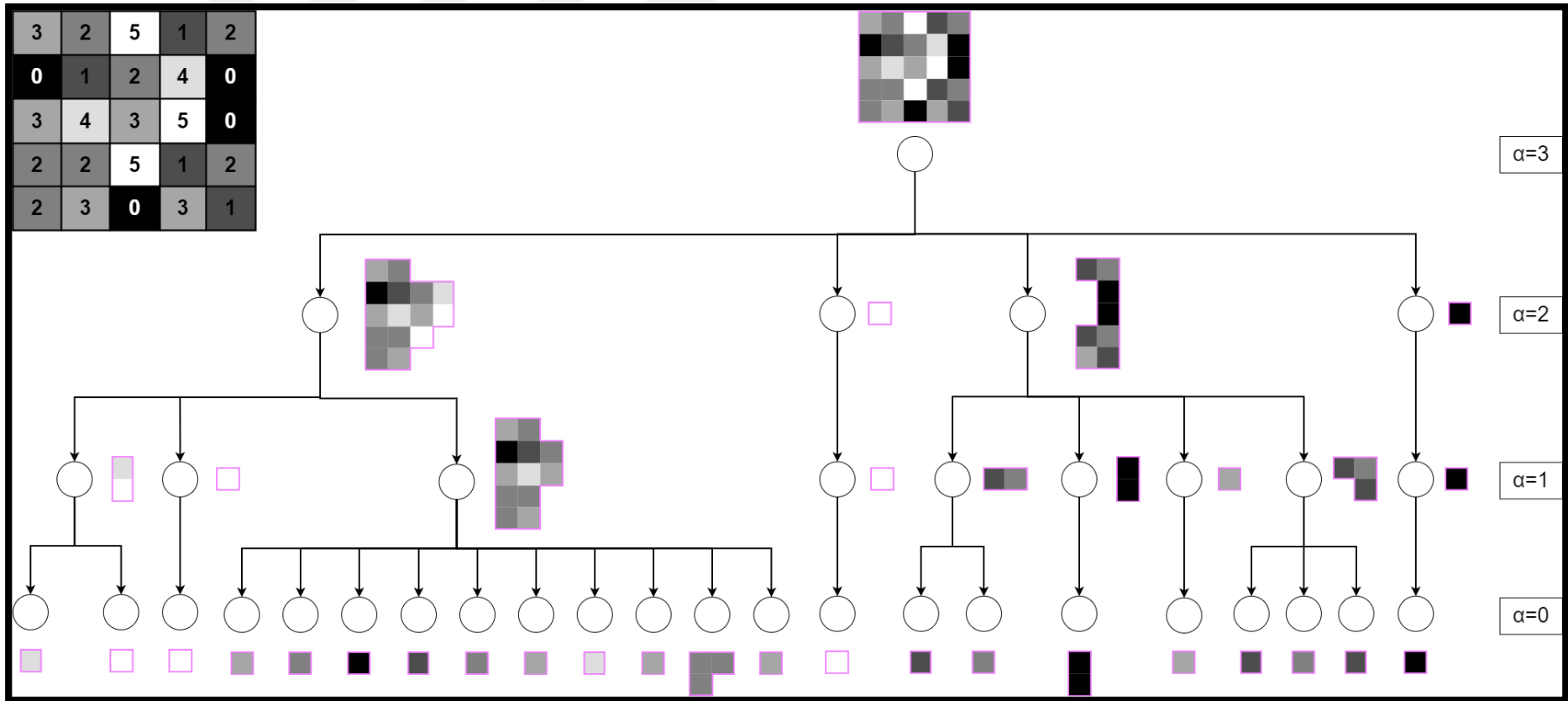


Figure 4.3: The input grayscale image (top left) and its α -tree representation (right).

predicate obtained by comparing a selected threshold with the computed CC attribute, which can be textural (e.g., range, mean, etc.) or geometric (e.g., area, diameter, etc.). An example criterion would be "Is the area of CC greater than 50 pixels?". The CCs that do not meet the criterion are removed, which corresponds to pruning their nodes in the tree representation. Then, the APs are obtained by applying a set of attribute thresholds with increasing values to the AFs and stacking their output images on top of each other.

Formally, let $f: \mathbb{Z}^2 \rightarrow \mathbb{Z}$ be a grayscale image and $\{\kappa_i\}_{1 \leq i \leq L}$ be a set of L attribute thresholds. Let γ^{κ_i} and ϕ^{κ_i} be morphological opening and closing filters with attribute threshold κ_i . An AP of the image f is constructed by applying these opening and closing filters γ^{κ_i} and ϕ^{κ_i} using the set of attribute thresholds κ_i as follows [16],

$$AP(f) = \{\phi^{\kappa_L}(f), \phi^{\kappa_{L-1}}(f), \dots, \phi^{\kappa_1}(f), f, \gamma^{\kappa_1}(f), \dots, \gamma^{\kappa_{L-1}}(f), \gamma^{\kappa_L}(f)\} \quad (4.1)$$

Extended attribute profiles (EAPs) are introduced to use APs with hyperspectral images [76]. Since the number of bands is high, a dimensionality reduction technique is first applied to the hyperspectral image $g: \mathbb{Z}^2 \rightarrow \mathbb{Z}^n$, resulting in $g': \mathbb{Z}^2 \rightarrow \mathbb{Z}^m$ for $m \ll n$. Then AP is calculated for each band and stacked together. More formally [76],

$$EAP(g) = \{AP(band_1), AP(band_2), \dots, AP(band_m)\} \quad (4.2)$$

For both APs and EAPs, each pixel is described by the sequence of values it acquires from the set of filtered images. As the threshold values of the attribute gradually increase, this leads to a coarser-level representation of the input image, resulting in a numerical sequence that decreases by the means of the spatial details for the pixel. Therefore, an AP can be considered as a scale-series from the pixel-level.

Lastly, although AFs were proposed long ago [77], their use is limited because their computation is computationally intensive. This problem was addressed by hierarchical tree representation [74], where tree nodes represent CCs. It made efficient computation of CCs in multi-levels possible, as well as enabling to operate on CCs with the selected attributes and to perform object-level analysis with tree pruning.

4.2.3. Long-Short Term Memory Network

A long-short term memory (LSTM) is a deep neural network designed to process sequential data [71]. It is an extension of the recurrent neural network (RNN) architecture, both of which consist of a chain-like structure. As adopting input from successive time steps, both networks enable employing information from previous events to make inferences for the current event and for future events, while retaining past knowledge. Despite the capability that this structure brings, RNNs suffer from handling long-term dependencies due to vanishing gradients when the gap between related time steps within the input is large. LSTMs were introduced specifically to address this problem [71].

More precisely, an LSTM cell takes a sequential input sample $x \in \mathbb{R}^n$, a hidden state h_{t-1} , and the cell state C_{t-1} from the previous cell, and returns the current cell state C_t and the hidden state h_t at time step t through its forget, input, and output gates f_t , i_t , and o_t , respectively. The forget gate f_t decides how much of the information from the previous cell will be retained. The input gate i_t decides what information is relevant to update the current cell state C_t . The output gate o_t decides the current hidden state h_t , which is passed to the next cell. Formally, the output of an LSTM cell at time step t is calculated as follows,

$$\begin{aligned}
f_t &= \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \\
i_t &= \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \\
o_t &= \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \\
\tilde{C}_t &= \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \\
h_t &= o_t \circ \tanh(C_t)
\end{aligned} \tag{4.3}$$

where σ , $\cdot$ and $\circ$ denote the sigmoid function, matrix multiplication, and dot product, W_* and U_* represent weight matrices, and b_* is the bias term specific to the gate or cell state [71]. Now that the necessary background information has been given, the next section will elaborate the proposed method.

4.3. Methodology

This section presents the AP-based pixel description strategy, where APs can be exploited with an LSTM network, similar to time series, since they also form a scale series due to their nature. Furthermore, training the model with powerful descriptors such as APs can help it learn more advanced features compared to training it with spectral signatures directly. To this end, this section first details how the APs are adopted as input to the LSTM network, and then elaborates the network architecture employed.

4.3.1. APs as Input to LSTM

As elaborated in the background section, the AP-based description of a pixel consists of a numerical sequence. Each member of this sequence represents the pixel by the selected attribute (area, elongation) computed from the CC containing the pixel, at a different spatial scale obtained by filtering the image with the given attribute threshold. Since all filtered images are obtained by applying different attribute thresholds to the same image, there is no reason for the values within an AP to be independent from each other. Hence, employing a deep neural network architecture that can capture the eventual dependencies within this scale series can contribute to classification performance.

The convolutional neural network is a deep neural network architecture known for its success in various classification tasks [6]. However, since it does not adopt the data as a sequence and treats each step independently, it cannot exploit the eventual dependencies mentioned above. Instead, LSTM comes forward with its capability of adopting the sequential data and capture dependencies in it. Moreover, coupling APs with LSTM can alleviate the latter's demand for large labeled dataset by feeding it with the APs that can already be considered as mid-level features and mitigate its training process by not starting from scratch [69, 70].

4.3.2. Network Architecture

The LSTM model from [78] is adapted to test the proposed AP-based pixel description strategy. It predicts the label of the sequence in a many-to-one manner,

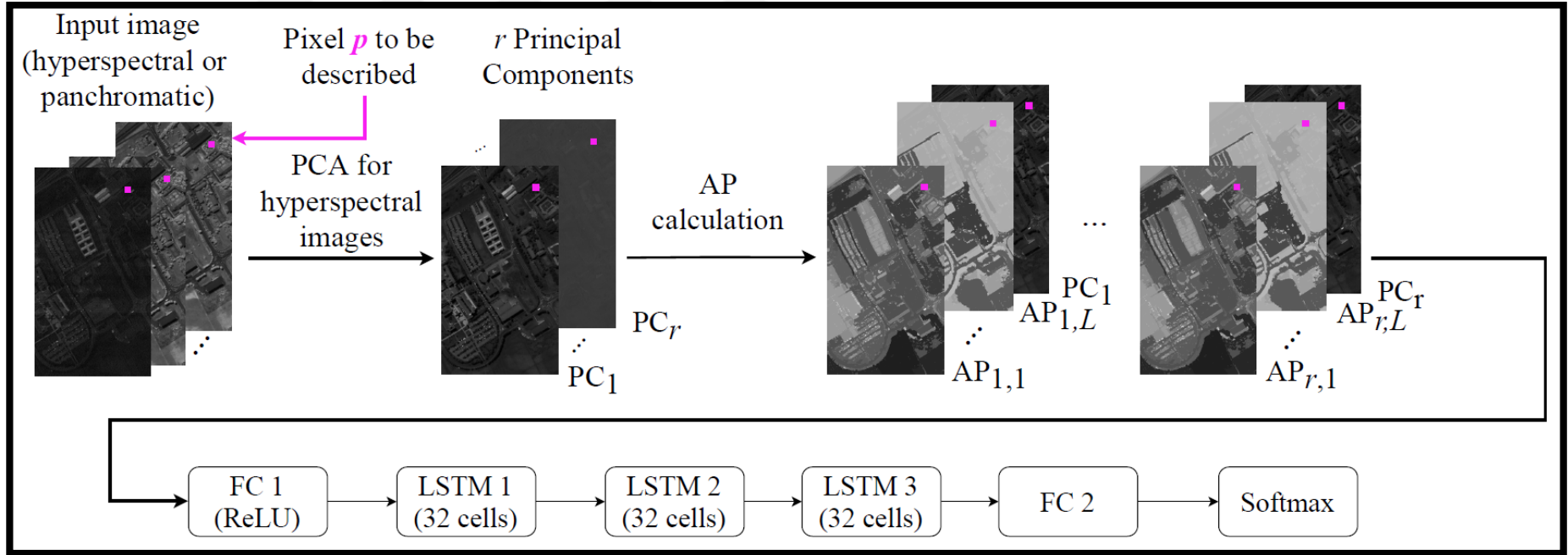


Figure 4.4: Outline of the proposed pixel description strategy and architecture of the model, where PCA, PC, FC, and ReLU denote principal component analysis, principal component image, fully connected layer, and rectified linear unit, respectively.

similar to the proposed AP-based pixel-level classification, where each AP scale series has a pixel label. The proposed architecture consists of 3 LSTM layers, each with 32 LSTM cells. The selection of the network hyperparameters is detailed in the following experiments section. Since the goal is to demonstrate that APs can be used as scale-series with LSTM, the model was not fine-tuned. The outline of the proposed model and AP-based pixel description strategy is presented in Figure 4.4.

4.4. Experiments

This subsection aims to test the hypothesis that exploiting the sequential characteristics of APs by coupling them to an LSTM network leads to better generalization performance in pixel-level classification, since APs also form a scale-series. To this end, the employed RS datasets are detailed, then the validation error common in the AP literature is discussed, and how the experiments were designed to overcome this problem is explained. Afterwards, the implementation parameters of the proposed approach for computing the hierarchical tree representation, generating APs, and training the model are elaborated. Finally, the alternative methods are presented, and the results are reported.

4.4.1. Datasets

The proposed approach is tested on two real RS datasets with different spectral properties, namely the hyperspectral Pavia University dataset and the panchromatic Reykjavik dataset, to demonstrate that it works for both types:

4.4.1.1. Pavia University Dataset

This dataset Figure 4.5 was obtained by the German Space Agency using the ROSIS-03 (Reflective Optics System Imaging Spectrometer) sensor over the flight of the Engineering School at University of Pavia in Pavia, Italy. It consists of 103 spectral bands ranging from 0.43 to 0.86 μm . The imaged area is 610×340 pixels and has a spatial resolution of 1.3 metres. The scene is an urban area with 9 groundtruth classes whose names and number of samples per class are given in Figure 4.5 and Table 4.1, respectively. Before calculating its tree representation, principal component analysis

(PCA) was applied to the dataset, resulting in 4 principal component images covering 99% of the variance. The dataset was provided by Prof. Paolo Gambala from the Telecommunications and Remote Sensing Laboratory of the University of Pavia.

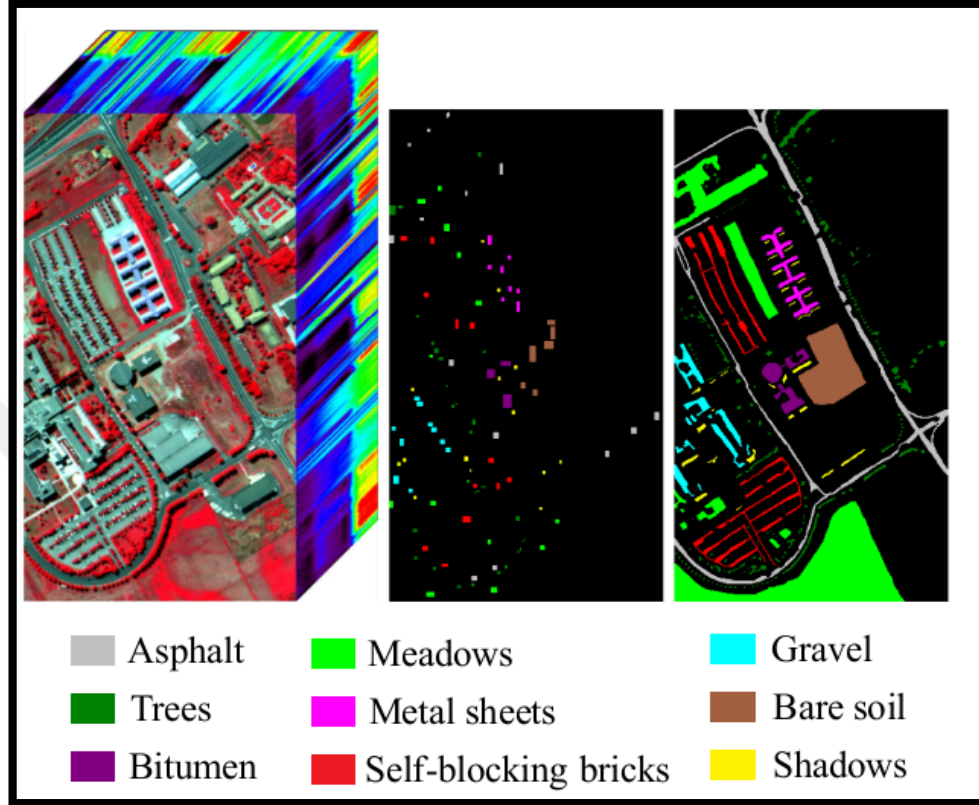


Figure 4.5: The Pavia University dataset, the training set, and the ground truth, as commonly encountered in the state-of-the-art, from left to right.

Table 4.1: The number of samples per class in the standard version of the Pavia University Dataset, and its horizontal and vertical splits.

Class	Labeled Samples per Split					
	Standard		Horizontal		Vertical	
	Train	Test	Train	Test	Train	Test
1	548	6631	210	3499	392	3094
2	540	18649	240	13433	479	9035
3	392	2099	0	0	0	0
4	524	3064	214	1653	524	1176
5	265	1345	0	0	265	43
6	532	5029	372	3530	48	4634
7	375	1330	0	0	0	0
8	514	3682	225	1973	0	0
9	231	947	28	732	191	153

4.4.1.2. Reykjavik Dataset

The second dataset Figure 4.6 was acquired via the IKONOS-2 satellite over Reykjavik, Iceland. It is a high-resolution panchromatic image with a spectral range of 0.45 to 0.9 μm . The image has a size of 628×700 pixels with a spatial resolution of 1 meter. The scene is an urban area with 6 classes whose names and number of samples per class are given in Figure 4.6 and Table 4.2, respectively.

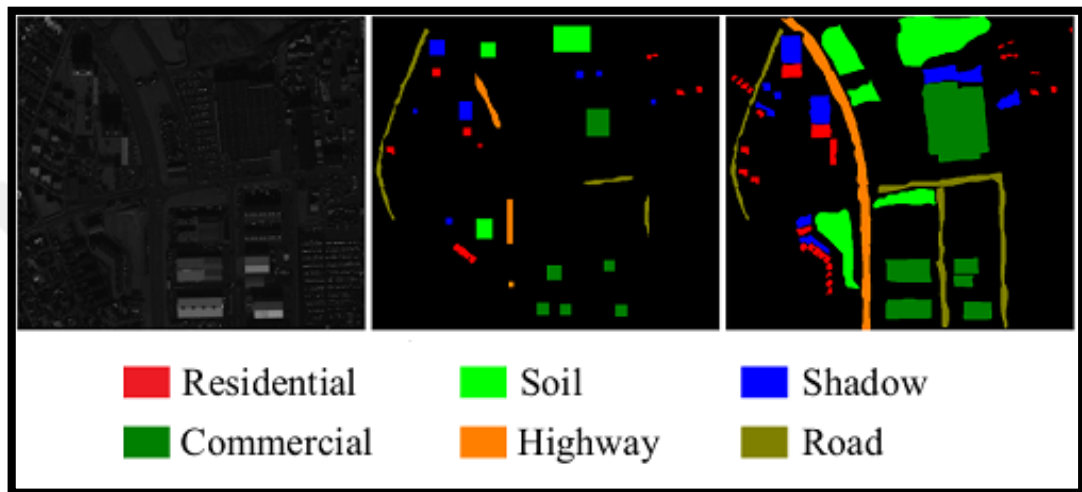


Figure 4.6: The Reykjavik dataset, the training set, and the ground truth, as commonly encountered in the state-of-the-art, from left to right.

Table 4.2: The number of samples per class in the standard version of the Reykjavik Dataset, and its horizontal and vertical splits.

Class	Labeled Samples per Split					
	Standard		Horizontal		Vertical	
	Train	Test	Train	Test	Train	Test
1	1863	6213	1195	1875	1446	669
2	6068	28144	4748	8885	2220	14419
3	2619	10610	2450	1278	2173	4078
4	5599	29768	2530	11893	450	27465
5	2489	12051	1300	5509	0	0
6	4103	11940	2147	9784	2828	8558

4.4.2. A Common Validation Misconduct

The use of standard training and test sets without further action is common in the relevant state-of-the-art remote sensing literature, as [72] notes. However, this leads to computing a single tree representation for the entire scene, resulting in CCs that contain both training and test samples. In contrast, models are anticipated to perform with a distinct tree representation computed from the scene to be classified in actual deployment.

To overcome this problem and properly test the proposed approach, both datasets are split in half horizontally and vertically, which are used to generate separate tree representations for training and test sets. In both cases, the first part is used for training and the second for testing. The resulting split datasets of the Reykjavik Dataset and the Pavia University Dataset are provided in Figure 4.7 and Figure 4.8, respectively. The use of these non-overlapping tree representations ensures that the model does not result in deceptively high performance. These non-overlapping (horizontal and vertical) versions are used along with the standard (overlapping) versions of the two datasets to train the model and evaluate its results.

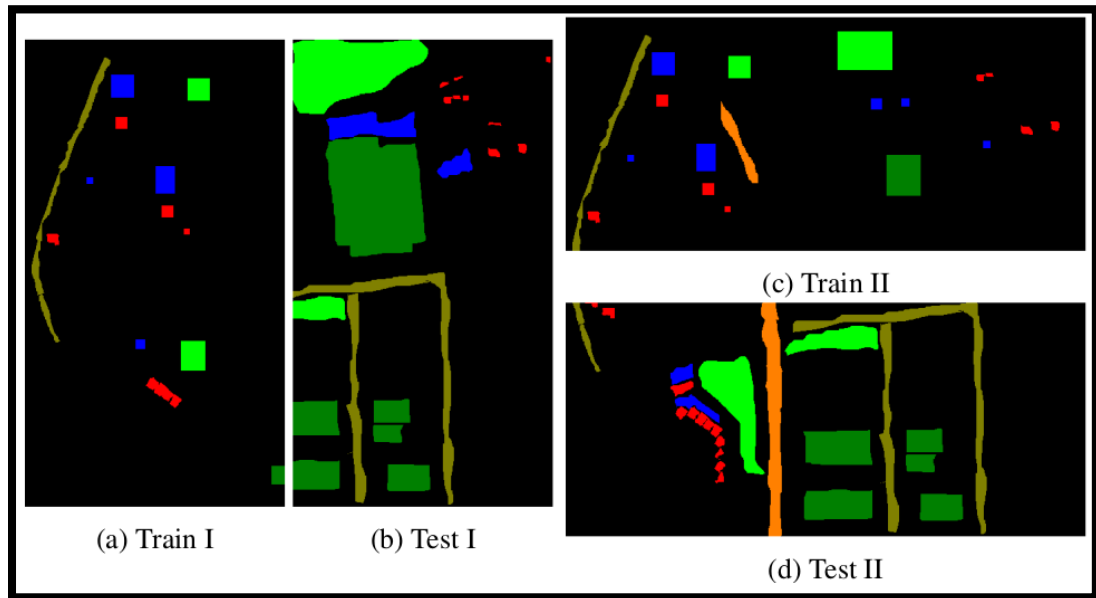


Figure 4.7: The Reykjavik dataset train and test sets, horizontal split (left), and vertical split (right).

It should be noted that this split operation may result in an insufficient number of samples in the training or test set for some of the classes. To solve this problem, the classes for which the ratio between the number of samples in the training or test set and the total number of samples is less than 0.1 are removed. This resulted in the removal of samples from the gravel, bitumen, and self-blocking bricks classes in the vertical split and samples from the gravel, metal, sheets and bitumen classes in the horizontal split for the Pavia University dataset. Similarly, samples of the highway class were removed in the vertical split of the Reykjavik dataset. Table 4.1 and Table 4.2 show the resulting number of samples per class depending on the split type for the Pavia University and Reykjavik Datasets, respectively.

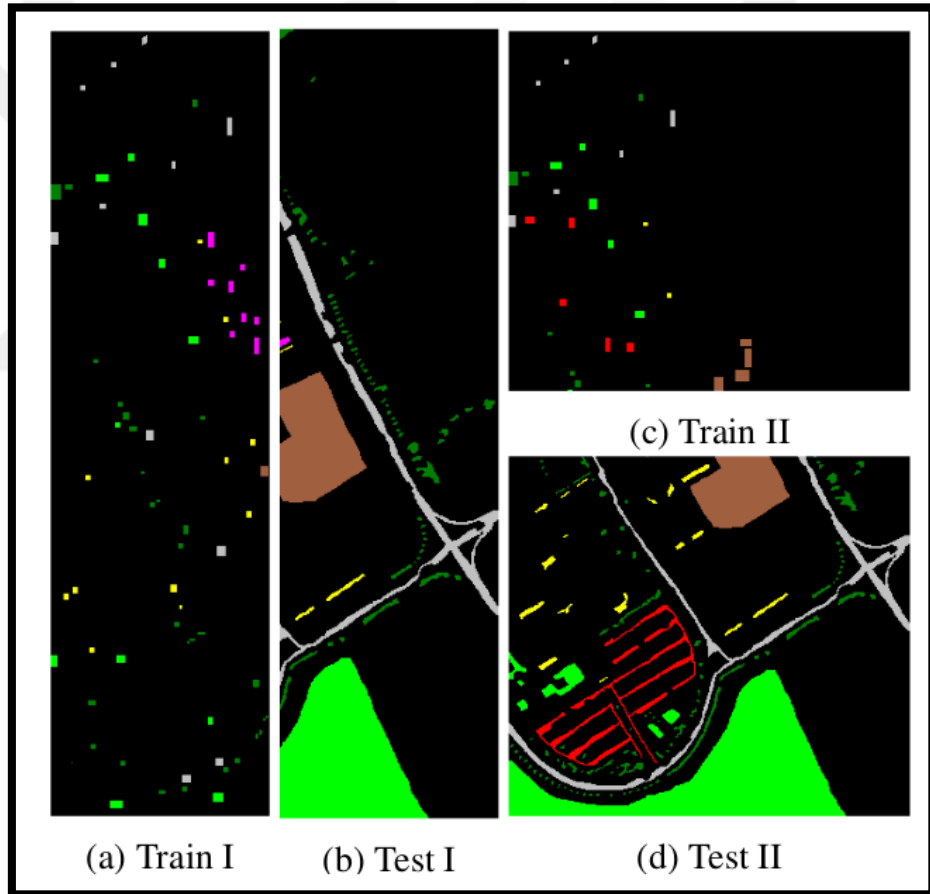


Figure 4.8: The Pavia University dataset train and test sets, horizontal split (left), and vertical split (right).

4.4.3. Implementation Details

4.4.3.1. Parameters for the Construction of Hierarchical Trees and APs

The area (pixel size of the region) was chosen to be comparable with the related literature [12, 62, 63, 65, 66, 79]. It is an increasing attribute that belongs to the family of geometric attributes. The CCs are computed using members of the two different tree families, namely the min-max tree from inclusion trees and the alpha-tree from partition trees. The following automatically calculated attribute thresholds are used for the Pavia University Dataset and the Reykjavik Dataset, which were recommended by [80] and [65], respectively:

$$\lambda_{Pav} = \{770, 1538, 2307, 3076, 3846, 4615, 5384, 6153, 6153, 6923, 8461, 9230, 10000, 10769\} \quad (4.4)$$

$$\lambda_{Rey} = \{25, 100, 500, 1000, 5000, 10000, 20000, 50000, 100000, 150000\}$$

With these threshold values, the length of the resulting APs is calculated as follows,

$$(number\ of\ thresholds \times number\ of\ trees + 1 \times number\ of\ bands) \quad (4.5)$$

where 1 is added because the original input image is used along with the AP images to describe the pixels. The number of trees is 2 for the min-max tree, since it contains CCs calculated for both the min tree and the max tree, and 1 for the alpha tree.

Thus, with the 4 PC images, the APs of the Pavia University Dataset have a length of $(14 \times 2 + 1) \times 4 = 116$ when the min-max tree is used to create the CCs. Similarly, the APs of the Pavia University Dataset have a length of $(14 \times 1 + 1) \times 4 = 60$ when the alpha-tree based CC calculation is used.

Likewise, the APs of the Reykjavik Dataset have a length of $(10 \times 2 + 1) \times 1 = 21$ when the min-max tree is used to calculate the CCs, as the number of bands is 1

since it is a panchromatic dataset. Similarly, the APs of the Reykjavik Dataset have a length of $(10 \times 1 + 1) \times 1 = 11$ in the alpha-tree based CC calculation.

4.4.3.2. Hyperparameters of the Network

The RF was trained with 100 trees. To allow a fair comparison of results, the same LSTM hyperparameters are used in all experiments, namely a learning rate of 0.0025, a batch size of 1500, and a number of 9440 epochs, all determined using grid search. The time step of the LSTM model is chosen to be 1 to be fair, since the length of the APs varies depending on the tree representation and the number of bands of datasets.

4.4.4. Alternative Methods

The proposed approach is compared with three of the following alternatives to demonstrate its performance and the results are presented in Table 4.3 for the Pavia University and Reykjavik Datasets:

- **AP + Random Forest**
APs are classified using a shallow random forest (RF) to see the effect of LSTM as widely used in the state-of-the-art literature.
- **Spectral Signature + LSTM**
Instead of using the mid-level APs as input, the spectral signatures are used as input samples for the LSTM model to observe the contribution of the APs.
- **AP + CNN**
To measure the contribution of LSTM to classification performance, the CNN architecture of [70] was adapted using a patch-based approach. The pixel under study was described as a 3D patch centered around the pixel by stacking APs. 5 CNN models are trained with different patch sizes, namely $\{3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9\}$, to ensure that the CNN model exploits the spatial information. The goal is to observe the effect of exploiting the sequential nature of APs by LSTM.

4.5. Results & Discussion

4.5.1. Results

The following observations can be derived from the Table 4.3: The RF classifier achieves better classification performance on standard datasets (3 out of 5) compared to the AP+LSTM and AP+CNN approaches, unless a spectral signature is used. This implies that using the same tree representation to encode samples of both training and test sets will result in some samples from these sets ending up in the same CC and therefore being described with the same values. This might also yield a worse classification performance for LSTM and CNN because, as DL models, they are more prone to overfitting than RF.

On the other hand, for the non-overlapping training and test sets, LSTM performs better (8 out of 15) than RF and CNN, where all classification performances drop drastically due to the above-mentioned validation misconduct. However, these performances are a better indicator of the generalization performance of their models. The reason why the AP+LSTM approach suffers less from this performance drop compared to the other methods can be explained by the ability of LSTM to deal with the sequential nature of the data. When two instances of the same class end up in different splits, they are less similar because they are described by APs computed from different trees. Therefore, RF and CNN classifiers process these APs as independent vectors, while LSTM treats them as scale-series and leverages the inherent information, which contributes to the performance difference. Therefore, the proposed AP+LSTM approach achieves the best performance for 3 out of 4 split cases (2 datasets $\times$ 2 splits) and for 5 out of 8 cases (2 datasets $\times$ 2 splits $\times$ 2 trees).

4.5.2. Discussion

Specifically, the AP+LSTM model with alpha-tree based APs performs better in the split cases than its min-max tree version for the Reykjavik Dataset. A possible explanation for this could be that Reykjavik is a single-band panchromatic dataset

Table 4.3: Pixel classification results in the form of Kappa statistics $\kappa \times 100$ with the best results in bold for the Pavia University and Reykjavik Datasets, where RF, CNN, LSTM, and * denote random forest, convolutional neural network, long-short term memory network, and the proposed approach, respectively.

Input	Dataset	Tree	Split	RF	CNN; kernel sizes				LSTM*
					3×3	5×5	7×7	9×9	
AP	Reykjavik	Alpha	Overlapping	71.56	40.41	57.06	66.87	71.54	65.61
	Reykjavik	Alpha	Vertical	7.17	5.71	5.74	9	2.81	22.97
	Reykjavik	Alpha	Horizontal	25.31	6.83	9.23	13.26	15.3	28.28
	Pavia	Alpha	Overlapping	85.04	55.32	64.17	73.23	73.99	75.88
	Pavia	Alpha	Vertical	15.69	36.41	40.14	40.94	38.06	33.49
	Pavia	Alpha	Horizontal	45.61	27.27	28.49	32.77	43.23	60
	Reykjavik	MinMax	Overlapping	76.46	62.95	72.15	75.19	76.99	58.26
	Reykjavik	MinMax	Vertical	15.73	23.71	25.13	24.26	24.78	17.45
	Reykjavik	MinMax	Horizontal	1.38	19.9	18.33	16.87	19.55	13.07
	Pavia	MinMax	Overlapping	87.79	60.85	63.39	47.18	59.02	83.94
	Pavia	MinMax	Vertical	17.22	36.46	36.15	28.42	21.31	49.7
	Pavia	MinMax	Horizontal	25.08	31.66	32.27	32.11	18.66	59.89
Spectral Signature	Pavia		Overlapping	65.6	49.76	56.84	54.61	56.05	71.67
	Pavia		Vertical	36.83	37.27	37.5	39.61	22.73	44.28
	Pavia		Horizontal	33.57	29.63	30.16	28.41	36.05	42.07

whose ground truth samples generally do not have extreme values. Figure 4.6 and the alpha-tree aims to represent the objects with low, intermediate and high intensities compared to the min-max tree, which is oriented toward the extrema values. This could also be the reason why the performance of the RF classifier changes drastically between the alpha-tree and min-max tree-based representations for the horizontal split of the Reykjavik dataset.

However, in the case of the Pavia University dataset, it is difficult to make the same suggestion, as the results of min-max tree and alpha-tree are on par for the horizontal case and min-max tree performs better for the vertical case. This could be due to several reasons: First, since the Pavia University is a hyperspectral dataset and these results were obtained from its PC images covering 99% of the variance, using min-max tree-based APs aimed at representing extreme values may lead to a better description. Second, the number of classes removed due to the split operation is higher in the case of the Pavia University Dataset, 3 for both splits, compared to the Reykjavik Dataset, where only 1 class is removed in the vertical split. In addition, there is a drastic jump in the horizontal split of the Pavia University Dataset with alpha-tree based representation. This could also be due to the ability of the LSTM to exploit sequence information within the flat-zone oriented alpha-tree based APs.

Another finding from these results is that the proposed AP+LSTM model achieves the best performance for all setups with the spectral signatures on the Pavia University Dataset. This suggests that the spectral signatures can also be exploited by an LSTM model. This is not surprising since the same scene is captured in RS images at different wavelengths, making the values within a pixel description correlated and thus forming a sequence that is an ideal input to an LSTM model, similar to APs and time series. However, the results also suggest that spectral signatures are not as powerful descriptors as APs, which was to be expected given the multi-scale description capability of the APs.

4.6. Conclusion

In this chapter, the combination of APs, which have a sequential character due to their ability to describe phenomena at multi-scale, with an LSTM network, that is specifically designed for the problems with sequential data, is investigated. It also

highlights a common flaw which found in the AP literature that leads to deceptively high generalization performance.

The proposed approach is tested on two remote sensing datasets using members of two different hierarchical tree representations. In all experiments where training and test sets do not overlap, significantly higher classification performance is obtained when APs and LSTM are used together than when APs or LSTM are used alone.



5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

This thesis focuses on the analysis of remote sensing image sequences in the context of deep learning. Although the DL methods achieve state-of-the-art performance in various fields, the main characteristics of RS images make the direct application of DL techniques for their analysis impossible. To this end, four DL approaches adapted for the RS images through exploiting their characteristics are proposed.

The first approach aims to exploit the spatial and spectral relationship between phenomena in remote sensing images and use the unlabeled data for pixel-level classification via a convolutional graphical neural network. It employs a graph-based approach with using connected components as nodes instead of regular pixel-based grids, which allows contextual relationships to be formed between phenomena even if they appear in different regions of the image. It also allows the use of unlabeled data through propagation of label information from neighboring nodes. It was tested with the Houston dataset, a hyperspectral remote sensing dataset. A node-based technique is used for classification. In order to adapt the problem to the graph structure, the nodes and the edges between them need to be established. Therefore, two different techniques are studied to form the nodes representing the connected components of the image, namely the SLIC algorithm and a selected scale of the α -tree representation. Then, different approaches are investigated to create edges between the nodes, including immediate and hopping spatial neighborhoods of the components and connecting each node to the rest to form a complete graph. The first two approaches resulted in worse classification performance than the random guess that led to the complete graph approach. However, the dense edge connections of this approach resulted in high computational complexity that made training the model impossible.

The second approach aims at proposing an alternative to the trade-off introduced by pooling layers, which are the essential component of semantic segmentation networks. The use of pooling layers leads the model to learn more and more abstract features, but also leads to a loss of contour information due to downsampling. This trade-off is particularly important in RS images because the phenomena in them occur

at multi-scales. For example, instances of car class may be represented by a small number of pixels which may get lost due to downsampling during pooling and may result in poor model performance. The proposed approach provides a solution to this problem with a custom pooling mechanism that uses a multi-scale hierarchical representation. This custom pooling layer uses partition maps obtained from the different scales of the hierarchical representation of the phenomenon. Unlike the widely used pooling layers that operate on regular grid-based windows, this custom pooling layer preserves the contour information with maintaining the multi-scale information, thus preserving the appearance of the small phenomena. This approach is applied to the Vaihingen dataset, which is a high-resolution remote sensing dataset. To generate the hierarchical representations, the α -tree representation is used and to demonstrate the performance, the pooling and unpooling layers of the SegNet architecture are replaced by their custom versions. The initial implementation of custom pooling resulted in impractical training time due to index-based access in GPU operations. It was re-implemented in a more mature way without using index-based access to GPU tensors, making the training time feasible. However, this also resulted in massive GPU memory usage which extends beyond the limits of the available GPU.

The third proposed approach leverages APs with LSTMs for pixel-level classification by combining the multi-scale description ability of APs leading to scale sequences with the capability of LSTMs to exploit sequential data. It also emphasizes a common validation error in the AP literature that leads to misleadingly high generalization performance. It is demonstrated on two remote sensing datasets by examining two hierarchical tree representations for AP construction. In all experiments where the training and test datasets did not overlap, the proposed approach achieved significantly higher performance than using APs or LSTMs alone.

The last approach proposes a water quality estimation technique using the unlabeled satellite images. To this end, a multi-task learning CNN architecture is presented, whose main task is Chl-a estimation and whose auxiliary task is to predict the month of image acquisition in order to exploit seasonal variations in Chl-a levels. This approach is demonstrated at Lake Balik using in-situ measurements and accompanying satellite images. The proposed multi-task learning model achieved 7.8 percent higher estimation performance compared to the single-task version of the model.

5.2. Future Work

As for future research, techniques that can achieve good classification performance without introducing high computational complexity are currently being investigated for the GCN approach. In addition, instead of manually generating the connections between the nodes, the learning of the adjacency matrix of the graph by the model is also under study.

For the pooling approach, techniques to reduce GPU memory usage are being investigated. In addition, alternatives to available GPU, such as cloud services that can offer high GPU memory, are also being explored.

For the AP and LSTM-based pixel-level classification approach, techniques without thresholds are investigated because the LSTM network can handle variable-length input samples.

Finally, the multi-task learning model will be explored for estimating water quality in water bodies with different characteristics, e.g., at different latitudes and/or with different trophic states. In addition, the model will be used to study land use/land cover problems.

REFERENCES

- [1] Avery T. E., (1968), “Interpretation of aerial photographs”, 2nd Edition, Burgess Publishing Company.
- [2] Moore G. K., (1979), “What is a picture worth? A history of remote sensing”, *Hydrological Sciences Bulletin*, 24 (4), 477–485.
- [3] Cheng G., Xie X., Han J., Guo L., Xia G. S., (2020), “Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities”, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 3735–3756.
- [4] Web 1, (2013), <https://www.usgs.gov/landsat-missions/landsat-data-access>, (Accessed on: 28/01/2022).
- [5] Moss B., (2012), “Cogs in the endless machine: Lakes, climate change and nutrient cycles: A review”, *Science of the Total Environment*, 434, 130–142.
- [6] Krizhevsky A., Sutskever I., Hinton G. E., (2012), “ImageNet classification with deep convolutional neural networks”, *Advances in Neural Information Processing Systems*, 1097–1105, Lake Tahoe, USA, 3-8 December.
- [7] Schmidhuber J., (2015), “Deep Learning in neural networks: An overview”, *Neural Networks*, 61, 85–117.
- [8] Bengio Y., Courville A., Vincent P., (2013), “Representation learning: A review and new perspectives”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (8), 1798–1828.
- [9] Alom M. Z., Taha T. M., Yakopcic C., Westberg S., Sidike P., Nasrin M. S., Hasan M., van Essen B. C., Awwal A. A. S., Asari V. K., (2019), “A state-of-the-art survey on deep learning theory and architectures”, *Electronics*, 8 (3), 292.
- [10] Reichstein M., Camps-Valls G., Stevens B., Jung M., Denzler J., Carvalhais N., Prabhat, (2019), “Deep learning and process understanding for data-driven Earth system science”, *Nature*, 566 (7743), 195–204.
- [11] Karpatne A., Jiang Z., Vatsavai R. R., Shekhar S., Kumar V., (2016), “Monitoring land-cover changes: A machine-learning perspective”, *IEEE Geoscience and Remote Sensing Magazine*, 4 (2), 8–21.
- [12] Dalla Mura M., Benediktsson J. A., Waske B., Bruzzone L., (2010), “Morphological attribute profiles for the analysis of very high resolution images”, *IEEE Transactions on Geoscience and Remote Sensing*, 48 (10), 3747–3762.

- [13] Vilaplana V., Marques F., Salembier P., (2008), “Binary partition trees for object detection”, *IEEE Transactions on Image Processing*, 17 (11), 2201–2216.
- [14] Zhang J., Jia X., Hu J., Tan K., (2021), “Moving Vehicle Detection for Remote Sensing Video Surveillance with Nonstationary Satellite Platform”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.
- [15] Aptoula E., Koc S. G., (2018), “Attribute profiles without thresholds”, *International Geoscience and Remote Sensing Symposium (IGARSS)*, 4507–4510, Valencia, Spain, 22-27 July.
- [16] Bosilj P., Kijak E., Lefevre S., (2018), “Partition and inclusion hierarchies of images: A comprehensive survey”, *Journal of Imaging*, 4 (2), 33.
- [17] Tochon G., Dalla Mura M., Veganzones M. A., Géraud T., Chanussot J., (2019), “Braids of partitions for the hierarchical representation and segmentation of multimodal images”, *Pattern Recognition*, 95, 162–172.
- [18] Kipf T. N., Welling M., (2017), “Semi-supervised classification with graph convolutional networks”, *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Toulon, France, April 24-26.
- [19] Zhang M., Luo H., Song W., Mei H., Su C., (2021), “Spectral-spatial offset graph convolutional networks for hyperspectral image classification”, *Remote Sensing*, 13 (21), 241–245.
- [20] Wan S., Gong C., Zhong P., Pan S., Li G., Yang J., (2020), “Hyperspectral Image Classification with Context-Aware Dynamic Graph Convolutional Network”, *IEEE Transactions on Geoscience and Remote Sensing*, 59 (1), 597–612.
- [21] Pahlevan N., Smith B., Schalles J., Binding C., Cao Z., Ma R., Alikas K., Kangro K., Gurlin D., Hà N., Matsushita B., Moses W., Greb S., Lehmann M. K., Ondrusek M., Oppelt N., et al., (2020), “Seamless retrievals of chlorophyll-a from Sentinel-2 (MSI) and Sentinel-3 (OLCI) in inland and coastal waters: A machine-learning approach”, *Remote Sensing of Environment*, 240, 111604.
- [22] Dörnhöfer K., Oppelt N., (2016), “Remote sensing for lake research and monitoring - Recent advances”, *Ecological Indicators*, 64, 105–122.
- [23] Huang C., Chen X., Li Y., Yang H., Sun D., Li J., Le C., Zhou L., Zhang M., Xu L., (2015), “Specific inherent optical properties of highly turbid productive water for retrieval of water quality after optical classification”, *Environmental Earth Sciences*, 73 (5), 1961–1973.
- [24] Lan X., Guo Z., Tian Y., Lei X., Wang J., (2015), “Retrieval of water quality parameters by neural network and analytical algorithm in Guanting Reservoir in Hebei Province in China”, *International Geoscience and Remote Sensing Symposium (IGARSS)*, 723–726, Milan, Italy, July 26-31.

- [25] Wang X., Fu L., Ma L., (2011), “Semi-supervised support vector regression model for remote sensing water quality retrieving”, *Chinese Geographical Science*, 21 (1), 57–64.
- [26] González Vilas L., Spyarakos E., Torres Palenzuela J. M., (2011), “Neural network estimation of chlorophyll a from MERIS full resolution data for the coastal waters of Galician rias (NW Spain)”, *Remote Sensing of Environment*, 115 (2), 524–535.
- [27] Verrelst J., Rivera J. P., Gitelson A., Delegido J., Moreno J., Camps-Valls G., (2016), “Spectral band selection for vegetation properties retrieval using Gaussian processes regression”, *International Journal of Applied Earth Observation and Geoinformation*, 52, 554–567.
- [28] O’Reilly J. E., Maritorena S., Mitchell B. G., Siegel D. A., Carder K. L., Garver S. A., Kahru M., McClain C., (1998), “Ocean color chlorophyll algorithms for SeaWiFS”, *Journal of Geophysical Research: Oceans*, 103 (C11), 24937–24953.
- [29] Henderikx Freitas F., Dierssen H. M., (2019), “Evaluating the seasonal and decadal performance of red band difference algorithms for chlorophyll in an optically complex estuary with winter and summer blooms”, *Remote Sensing of Environment*, 231, 111228.
- [30] Matthews M. W., Bernard S., Robertson L., (2012), “An algorithm for detecting trophic status (chlorophyll-a), cyanobacterial-dominance, surface scums and floating vegetation in inland and coastal waters”, *Remote Sensing of Environment*, 124, 637–652.
- [31] Ansper A., Alikas K., (2018), “Retrieval of chlorophyll a from Sentinel-2 MSI data for the European Union water framework directive reporting purposes”, *Remote Sensing*, 11 (1), 64.
- [32] Hinton G. E., Salakhutdinov R. R., (2006), “Reducing the dimensionality of data with neural networks”, *Science*, 313 (5786), 504–507.
- [33] Peterson K. T., Sagan V., Sloan J. J., (2020), “Deep learning-based water quality estimation and anomaly detection using Landsat-8/Sentinel-2 virtual constellation and cloud computing”, *GIScience and Remote Sensing*, 57 (4), 510–525.
- [34] Pu F., Ding C., Chao Z., Yu Y., Xu X., (2019), “Water-quality classification of inland lakes using Landsat8 images by convolutional neural networks”, *Remote Sensing*, 11 (14).
- [35] Aptoula E., Ariman S., (2021), “Chlorophyll-a Retrieval from Sentinel-2 Images Using Convolutional Neural Network Regression”, *IEEE Geoscience and Remote Sensing Letters*, 19, 1–5.

- [36] Syariz M. A., Lin C. H., van Nguyen M., Jaelani L. M., Blanco A. C., (2020), "WaterNet: A convolutional neural network for chlorophyll-a concentration retrieval", *Remote Sensing*, 12 (12).
- [37] Cho H., Choi U. J., Park H., (2018), "Deep learning application to time-series prediction of daily chlorophyll-a concentration", *WIT Transactions on Ecology and the Environment*, 215, 157–163.
- [38] Barzegar R., Aalami M. T., Adamowski J., (2020), "Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model", *Stochastic Environmental Research and Risk Assessment*, 34 (2), 415–433.
- [39] Reyjol Y., Argillier C., Bonne W., Borja A., Buijse A. D., Cardoso A. C., Daufresne M., Kernan M., Ferreira M. T., Poikane S., Prat N., Solheim A. L., Stroffek S., Usseglio-Polatera P., Villeneuve B., van de Bund W., (2014), "Assessing the ecological status in the context of the European Water Framework Directive: Where do we go now?", *Science of the Total Environment*, 497–498, 332–344.
- [40] Ritchie J. C., Zimba P. v., Everitt J. H., (2003), "Remote sensing techniques to assess water quality", *Photogrammetric Engineering and Remote Sensing*, 69 (6), 695–704.
- [41] Xu M., Liu H., Beck R., Lekki J., Yang B., Shu S., Liu Y., Benko T., Anderson R., Tokars R., Johansen R., Emery E., Reif M., (2019), "Regionally and Locally Adaptive Models for Retrieving Chlorophyll-a Concentration in Inland Waters From Remotely Sensed Multispectral and Hyperspectral Imagery", *IEEE Transactions on Geoscience and Remote Sensing*, 57 (7), 4758–4774.
- [42] Crippen R. E., (1988), "The dangers of underestimating the importance of data adjustments in band ratioing", *International Journal of Remote Sensing*, 9 (4), 767–776.
- [43] Ligi M., Kutser T., Kallio K., Attila J., Koponen S., Paavel B., Soomets T., Reinart A., (2017), "Testing the performance of empirical remote sensing algorithms in the Baltic Sea waters with modelled and in situ reflectance data", *Oceanologia*, 59 (1), 57–68.
- [44] Cortes C., Vapnik V., (1995), "Support-Vector Networks", *Machine Learning*, 20 (3), 273–297.
- [45] Cheng B., Titterton D. M., (1994), "Neural networks: A review from a statistical perspective", *Statistical Science*, 9 (1), 2–30.
- [46] Rumelhart D. E., Hinton G. E., Williams R. J., (1986), "Learning representations by back-propagating errors", *Nature*, 323 (6088), 533–536.
- [47] Son S. H., Wang M., (2012), "Water properties in Chesapeake Bay from MODIS-Aqua measurements", *Remote Sensing of Environment*, 123, 163–174.

- [48] Ariman S., Koyuncu S., (2019), “Su Kirliliği Açısından Hassas Alanların İzlenmesi: Kızılırmak Deltası-Balık Gölü”, Mühendislik Bilimleri ve Tasarım Dergisi, 7 (4), 705–714.
- [49] Arar E. J., Collins G. B., (1997), “Method 445.0: In vitro determination of chlorophyll a and pheophytin a in marine and freshwater algae by fluorescence, In Vitro Determination of Chlorophyll a and Pheophytin a in Marine and Freshwater Algae by Fluorescence,. United States Environmental Protection Agency, Office of Research and Development, National Exposure Research Laboratory”.
- [50] Steinmetz F., Deschamps P.Y., Ramon D., (2011), “Atmospheric correction in presence of sun glint: application to MERIS”, Optics Express, 19 (10), 9783.
- [51] Warren M. A., Simis S. G. H., Martinez-Vicente V., Poser K., Bresciani M., Alikas K., Spyarakos E., Giardino C., Ansper A., (2019), “Assessment of atmospheric correction algorithms for the Sentinel-2A MultiSpectral Imager over coastal and inland waters”, Remote Sensing of Environment, 225, 267–289.
- [52] Choi J.-H., Kim J., Won J., Min O., (2019), “Modelling chlorophyll-a concentration using deep neural networks considering extreme data imbalance and skewness”, 21st International Conference on Advanced Communication Technology (ICACT), 631–634, PyeongChang, South Korea, 17-20 February.
- [53] Cipolla R., Gal Y., Kendall A., (2018), “Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics”, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 7482–7491, Utah, USA, 18-22 June.
- [54] Web 2, (2020), <https://github.com/Mikoto10032/AutomaticWeightedLoss>, (Accessed on: 25/01/2022).
- [55] Liebel L., Körner M., (2018), “Auxiliary Tasks in Multi-task Learning”, arXiv preprint arXiv:1805.06334,.
- [56] Niroumand-Jadidi M., Bovolo F., Bruzzone L., (2019), “Novel spectra-derived features for empirical retrieval of water quality parameters: Demonstrations for OLI, MSI, and OLCI sensors”, IEEE Transactions on Geoscience and Remote Sensing, 57 (12), 10285–10300.
- [57] Batur E., Maktav D., (2018), “Assessment of surface water quality by using satellite images fusion based on PCA method in the Lake Gala, Turkey”, IEEE Transactions on Geoscience and Remote Sensing, 57 (5), 2983–2989.
- [58] Aptoula E., Ariman S., (2020), “Hierarchical Spatial-Spectral Features for the Chlorophyll-a Estimation of Lake Balık, Turkey”, IEEE Geoscience and Remote Sensing Letters, 19, 1–5.

- [59] Yokoya N., Ghamisi P., Hänsch R., Schmitt M., (2020), “2020 IEEE GRSS data fusion contest: Global land cover mapping with weak supervision”, IEEE Geoscience and Remote Sensing Magazine, 8 (1), 154–157.
- [60] Bernabe S., Marpu P. R., Plaza A., Mura M. D., Benediktsson J. A., (2013), “Spectral-spatial classification of multispectral images using kernel feature space representation”, IEEE Geoscience and Remote Sensing Letters, 11 (1), 288–292.
- [61] Dalla Mura M., Villa A., Benediktsson J. A., Chanussot J., Bruzzone L., (2010), “Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis”, IEEE Geoscience and Remote Sensing Letters, 8 (3), 542–546.
- [62] Derbashi U., Aptoula E., (2020), “Attribute Profiles Without Thresholds Through Histogram Based Tree Path Description”, 2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS), 29–32, Tunis, Tunisia, 9-11 March.
- [63] Bhardwaj K., Patra S., Bruzzone L., (2019), “Threshold-Free Attribute Profile for Classification of Hyperspectral Images”, IEEE Transactions on Geoscience and Remote Sensing, 57 (10), 7731–7742.
- [64] Dalla Mura M., Benediktsson J. A., Bruzzone L., (2011), “Self-dual attribute profiles for the analysis of remote sensing images”, International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, 320–330, Verbania-Intra, Italy, 6-8 July.
- [65] Cavallaro G., Mura M. D., Benediktsson J. A., Plaza A., (2016), “Remote Sensing Image Classification Using Attribute Filters Defined over the Tree of Shapes”, IEEE Transactions on Geoscience and Remote Sensing, 54 (7), 3899–3911.
- [66] Bosilj P., Damodaran B. B., Aptoula E., Dalla Mura M., Lefèvre S., (2017), “Attribute profiles from partitioning trees”, International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, 381–392, Fontainebleau, France, 15-17 May.
- [67] Aptoula E., Dalla Mura M., Lefevre S., (2016), “Vector Attribute Profiles for Hyperspectral Image Classification”, IEEE Transactions on Geoscience and Remote Sensing, 54 (6), 3208–3220.
- [68] Aptoula E., (2015), “Hyperspectral Image Classification with Multidimensional Attribute Profiles”, IEEE Geoscience and Remote Sensing Letters, 12 (10), 2031–2035.
- [69] Chen Y., Zhu L., Ghamisi P., Jia X., Li G., Tang L., (2017), “Hyperspectral Images Classification with Gabor Filtering and Convolutional Neural Network”, IEEE Geoscience and Remote Sensing Letters, 14 (12), 2355–2359.

- [70] Aptoula E., Ozdemir M. C., Yanikoglu B., (2016), “Deep Learning with Attribute Profiles for Hyperspectral Image Classification”, *IEEE Geoscience and Remote Sensing Letters*, 13 (12), 1970–1974.
- [71] Hochreiter S., Schmidhuber J., (1997), “Long Short-Term Memory”, *Neural Computation*, 9 (8), 1735–1780.
- [72] Audebert N., le Saux B., Lefevre S., (2019), “Deep learning for classification of hyperspectral data: A comparative review”, *IEEE Geoscience and Remote Sensing Magazine*, 7 (2), 159–173.
- [73] Jones R., (1997), “Component trees for image filtering and segmentation”, *IEEE Workshop on Nonlinear Signal and Image Processing*, Michigan, USA, 8-10 September.
- [74] Salembier P., Oliveras A., Garrido L., (1998), “Antiextensive connected operators for image and sequence processing”, *IEEE Transactions on Image Processing*, 7 (4), 555–570.
- [75] Vincent L., (1993), “Grayscale area openings and closings: their applications and efficient implementation”, *EURASIP Workshop on Mathematical Morphology and its Applications to Signal Processing*, 22–27, Barcelona, Spain, May.
- [76] Dalla Mura M., Benediktsson J. A., Waske B., Bruzzone L., (2010), “Extended profiles with morphological attribute filters for the analysis of hyperspectral data”, *International Journal of Remote Sensing*, 31 (22), 5975–5991.
- [77] Breen E. J., Jones R., (1996), “Attribute Openings, Thinnings, and Granulometries”, *Computer Vision and Image Understanding*, 64 (3), 377–389.
- [78] Web 3, (2016), <https://github.com/guillaume-chevalier/LSTM-Human-Activity-Recognition>, (Accessed on: 15/01/2022).
- [79] Pham M. T., Lefevre S., Aptoula E., (2017), “Local Feature-Based Attribute Profiles for Optical Remote Sensing Image Classification”, *IEEE Transactions on Geoscience and Remote Sensing*, 56 (2), 1199–1212.
- [80] Ghamisi P., Benediktsson J. A., Sveinsson J. R., (2013), “Automatic spectral--spatial classification framework based on attribute profiles and supervised feature extraction”, *IEEE Transactions on Geoscience and Remote Sensing*, 52 (9), 5771–5782.
- [81] Achanta R., Shaji A., Smith K., Lucchi A., Fua P., Süsstrunk S., (2012), “SLIC superpixels compared to state-of-the-art superpixel methods”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (11), 2274–2281.

- [82] Deng Z., Sun H., Lei L., Zhou S., Zou H., (2018), “Object Detection in Remote Sensing Imagery with Multi-scale Deformable Convolutional Networks”, *Acta Geodaetica et Cartographica Sinica*, 47 (9), 1216–1227.



BIOGRAPHY

Melike İteralp took her Bachelor's degree from the Computer Science Department of Istanbul Bilgi University in June 2014. She started her Master's degree at the Computer Engineering Department of Gebze Technical University in September 2019.



APPENDICES

Appendix A: The Publications about the Thesis

İlteralp M., Arıman S., Aptoula E., (2022), “A Deep Multitask Semisupervised Learning Approach for Chlorophyll-a Retrieval from Remote Sensing Images”, *Remote Sensing*, 14 (1), 18.

İlteralp M., Aptoula E., (2021), “Processing Attribute Profiles as Scale-series for Remote Sensing Image Classification.”, *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021)*, 4: VISAPP, 558–565.

Appendix B: Explanation

This thesis was supported by a TUBITAK grant (Project No: 118E258)

Appendix C: Remote Sensing Image Classification with Graph Convolutional Neural Networks

This chapter focuses on pixel-level classification using graph convolutional networks (GCN) with hierarchical tree representation. It is known that GCNs perform well when the labeled data is limited [18] which is also being one of the issues of RS image analysis. Moreover, the proposed approach also addresses two previously mentioned problems in the analysis of RS images, namely, how to deal with the multi-scale appearance of the phenomena and how to exploit the spatial context by proposing an adapted DL model that benefits from these properties of such images.

This approach consists of four parts, namely generating a tree representation from the image, creating its graph, extracting features, and classification. It starts with computing a tree representation of the RS image. For this purpose, a member of the families of partition trees and component trees is used, namely α -tree and max-tree. A hierarchical tree representation consists of a set of nodes, each representing a connected component, and each node is connected to its parent by an edge.

- Graph Construction

Since each node of the tree corresponds to a connected component of the image, these nodes are used to generate a graph representation of the image. A graph $G = (V, E)$ consists of a vertex set and an edge set. The superpixels computed from the

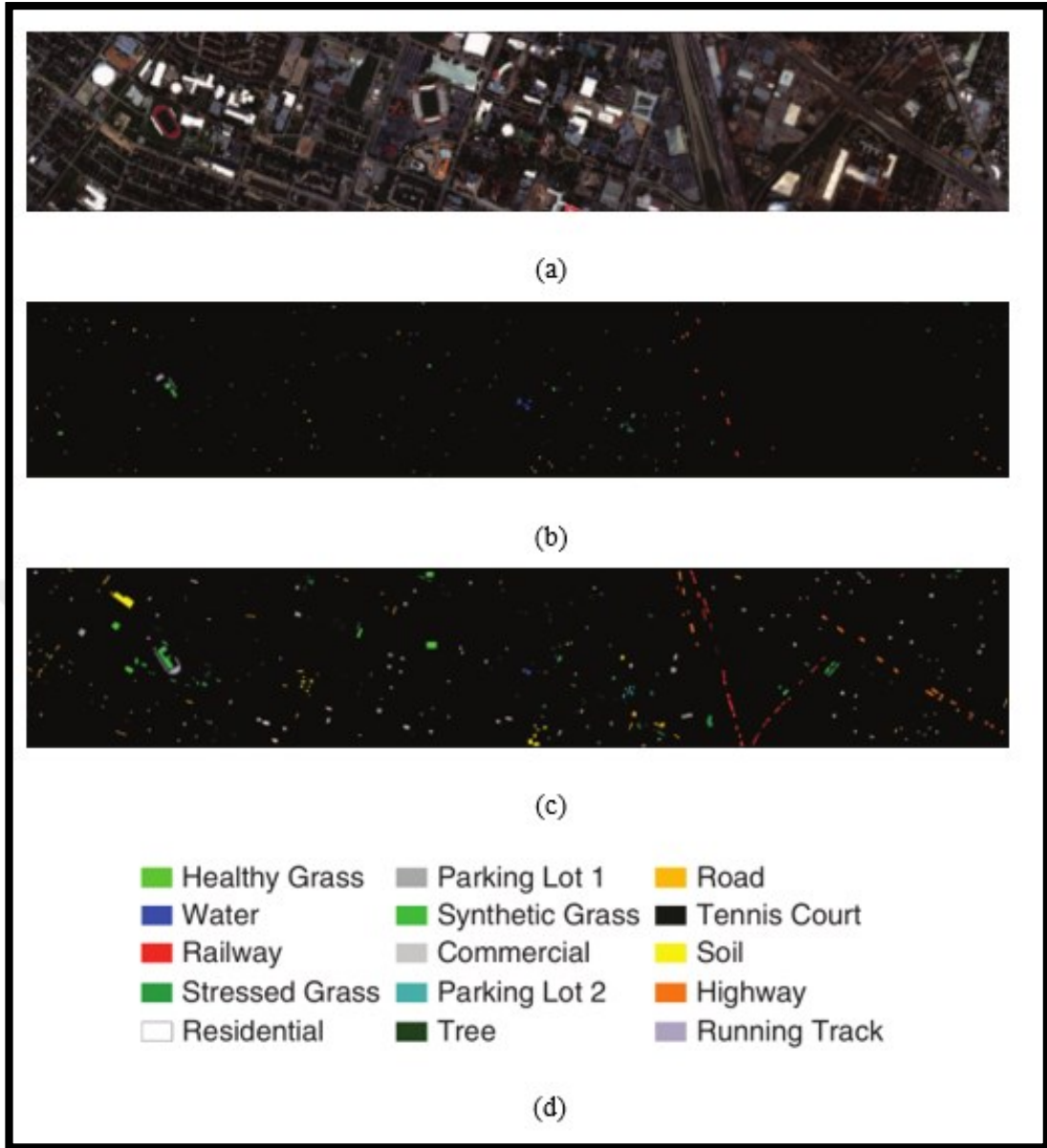


Figure C.1.1: The Houston University dataset, (a) a color composite representation, (b) the training set, (c) the test set, and (d) the ground truth labels, as commonly encountered in the state-of-the-art, from top to bottom.

tree are used as vertices. To represent the relationships between these vertices, different adjacency types are explored. In min/max-tree approach, the leaf nodes of the tree are used as vertices of the graph. Two nodes are connected by the number of hops required to travel from one node to another, forming the edge weight. In this way, a complete graph is created. Using the α -tree, two approaches to constructing edges are explored. In both approaches, $\alpha = 0$ components of the tree are used as graph nodes. In the first approach, coarse components with a manually chosen threshold ($\alpha = 3$) are selected from the tree. Then, for each coarse component, its $\alpha = 0$ children nodes are

connected in the resulting graph. In the second method, spatially adjacent $\alpha = 0$ nodes are connected to each other in the graph. In addition, the SLIC algorithm [81] is used to generate superpixels from the image, which are used as another way to represent graph nodes. In this case, spatially adjacent superpixels are treated as neighbors in the graph. In all three approaches, majority voting is used to determine the label of the resulting graph node. All graphs created using the three aforementioned approaches are undirected graphs.

Area is used as an attribute in graphs generated from tree representations using the feature extraction method of [15]. For the graph generated by the SLIC algorithm, the mean intensity within the superpixel and its area are used as node features. The architecture used is the graph convolutional network architecture of [15] with 2 graph convolution layers.

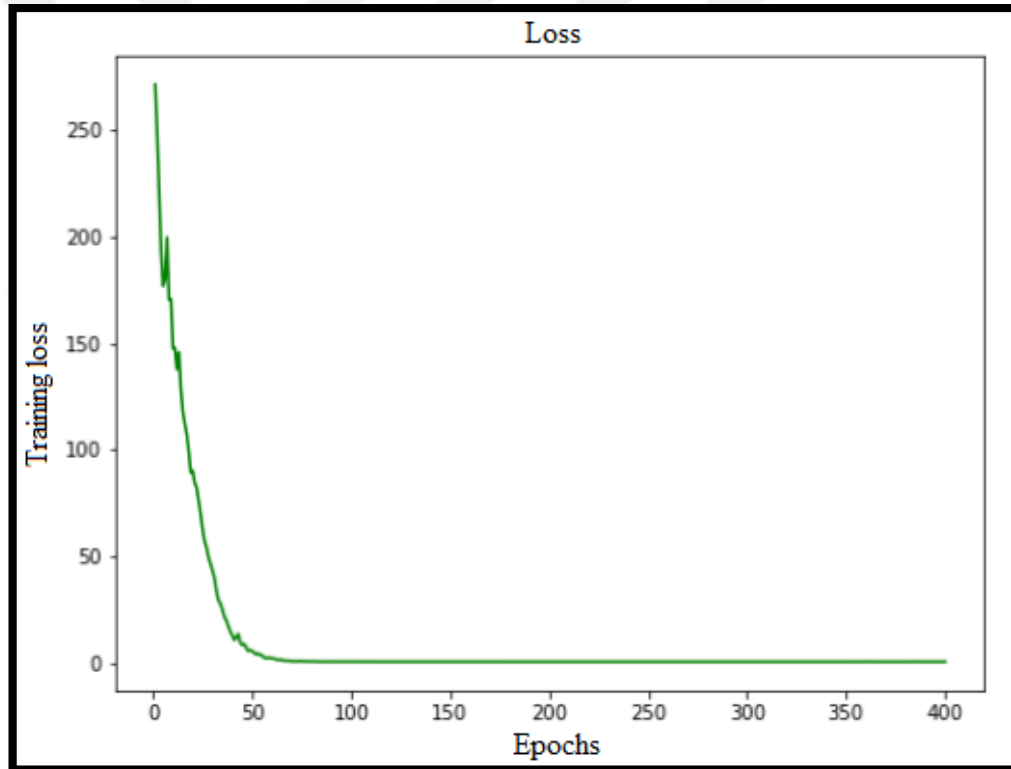


Figure C.1.2: Loss on the training set across epochs for the proposed approach.

- Experiments

The approach is tested with the Houston University Dataset, a hyperspectral dataset consisting of 144 spectral bands with 2.5 m spatial resolution Figure C.1.1. The model trained using the leaf nodes of the max-tree and the α -tree is computationally

infeasible because there are too many graph nodes and connections between them. Using higher level nodes in the tree may seem like a solution to this problem. But there is another problem: as the number of pixels covered by the higher-level nodes increases, it is no longer possible to assign them the proper labels in the resulting graph. This problem is still under investigation.

In addition to the tree representations, graphs generated using the SLIC algorithm are also used. Although the GCN model trained with this graph appeared to learn, as indicated by the training loss in Figure C.1.2, its kappa value fluctuates in both training and test sets and does not exceed 0.17, as displayed in Figure C.1.3. Based on experiments conducted with random forest on the same data where kappa 0.71 was obtained, it is known that the problem is not with the data. It is also known from the literature that the GCN architectures achieve state-of-the-art results on different graph datasets. This indicates that the graph adjacency matrix used does not yield to baseline classification result for this problem. Therefore, learning the optimal adjacency matrix using given node features for the second problem is also currently under investigation.

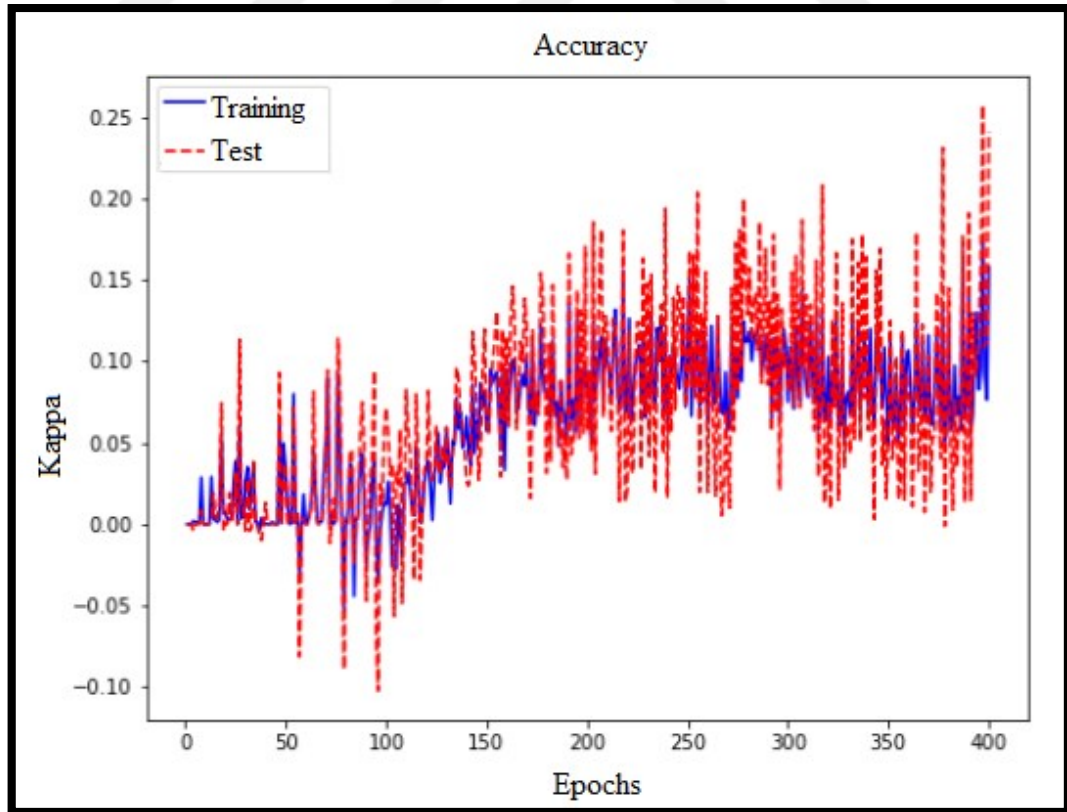


Figure C.1.3: Kappa values of training and test sets across epochs for the proposed approach.

- Conclusion

This chapter focuses on the pixel-level classification of RS images using graph convolutional networks with hierarchical tree representation. It is well known that graph convolutional networks work well when the labeled data is limited [18] which is also one of the problems of RS image analysis. Since it was not possible to find an optimal adjacency matrix for the given node features and node labeling problem, it was not possible to obtain the baseline classification results in this chapter. Two of the stated problems of this approach are currently under investigation.

Appendix D: Hierarchical Tree Representation-Based Pooling Mechanism for Remote Sensing Image Classification

Semantic segmentation is one of the primary tasks in RS image analysis, and deep semantic segmentation networks constitute one of the state-of-the-art approaches to this task. Thanks to their end-to-end architecture, they can efficiently and effectively create pixel-level classification maps for large scenes. In particular, spatial pooling layers constitute integral parts of the aforementioned networks as they can reduce computational complexity by reducing the number of parameters through downsampling of feature maps. However, there is a well-known trade-off between the number of pooling layers (resulting in more abstract features encoding higher level concepts) and the loss of contour precision due to downsampling. This is especially important for phenomena represented with a small number of pixels (e.g., cars) in the RS images, as they will be eliminated in this repetitive pooling process, leading to poor performance of the learning algorithm [72, 82].

In this section, the above problems are addressed using an efficiently computed hierarchical multi-scale representation. More precisely, instead of using the regular grid-shaped pooling/unpooling layers, partition maps obtained from the different levels (i.e., scales) of an α -tree are used to implement custom pooling and unpooling layers, namely TreePooling and TreeUnpooling, respectively. In this way, contour information and small phenomena are preserved in the pooling phase. Moreover, in the upsampling phase, the feature maps are reinforced with the accurate contour information from the different levels of hierarchical representation of the input.

Finally, this chapter addresses two previously mentioned problems of RS image analysis, namely, dealing with the multi-scale appearance of the phenomena and exploiting the spatial context by proposing an adapted DL model that benefits from these properties of such images.

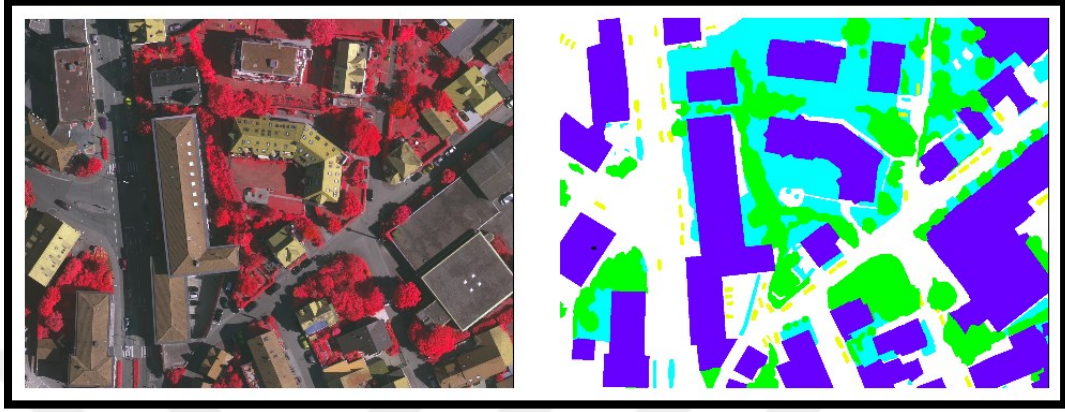


Figure D.1.1: A sample from Vaihingen Dataset and its ground truth.

The hypothesis is that a network trained with this hierarchical multi-scale pooling mechanism will perform a better classification performance compared to the standard pooling version of the network, since the proposed approach inherently preserves contour information and does not eliminate small phenomena. This hypothesis is tested by replacing the pooling and unpooling layers of the SegNet architecture, a semantic segmentation network, with the proposed custom pooling and unpooling layers. The Vaihingen Dataset, a high-resolution RS dataset whose sample image is provided in Figure D.1.1, is used to demonstrate the performance of the proposed approach.

TreePooling and TreeUnpooling layers and their forward and backward propagation operations are implemented using the PyTorch framework. Since the Vaihingen Dataset has a very high resolution and its images are not the same size, its images are padded with their mirrored versions so that they are equal to the closest factor of 320. Then, 320×320 patches are cropped from each of these images to be used as samples for training and testing the model with custom pooling.

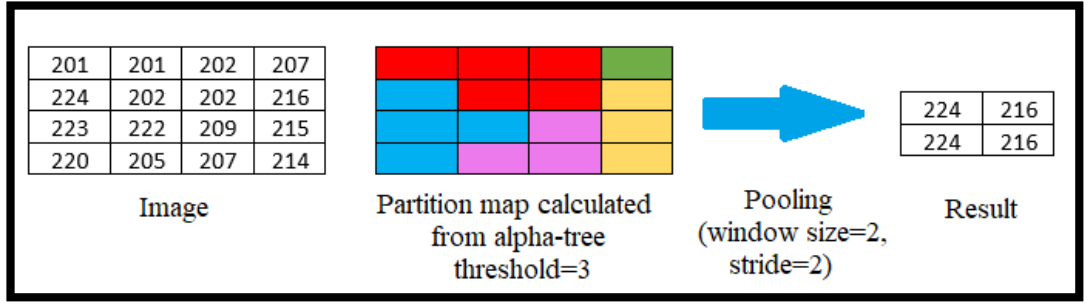


Figure D.1.2: An example of the proposed multi-scale-based pooling mechanism.

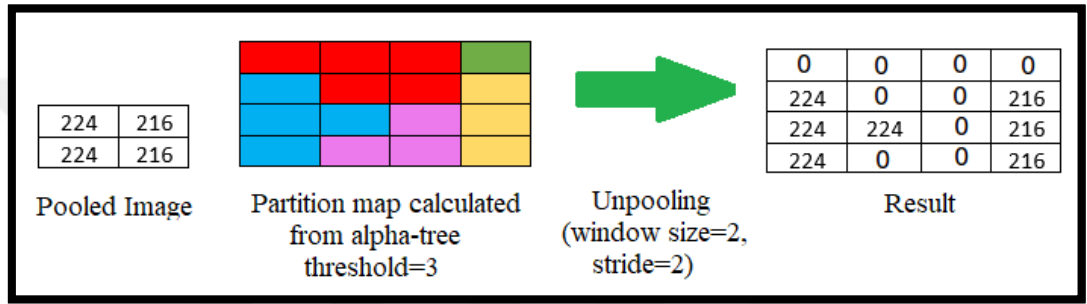


Figure D.1.3: An example of unpooling with the proposed multi-scale-based mechanism.

Since the proposed approach is demonstrated on the SegNet architecture, its 5 pooling and 5 unpooling layers are replaced by their custom versions. Then, to imitate the effect of pooling on image size but without losing contour information, each sample image is resized with gradually decreasing order to form the input to these custom pooling layers. This means that for the SegNet architecture with pooling parameters $stride = 2$ and $window\ size = 2$, the downsized images are $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ of the original input size, respectively. The α -tree of each resized image is then computed to create partition maps with using coarse-to-fine thresholds of 11, 9, 7, 5, and 3, respectively. Each of these partition maps is then passed to the corresponding pooling layer along with its input. An example of this custom pooling is given in Figure D.1.2 and its unpooling is given in Figure D.1.3. A sample image with its 3 partition maps is provided in Figure D.1.4.

- Experiments

The first implementation of the custom pooling and unpooling layers was too slow to train the model because it used index-based access with GPU operations. Then these layers were re-implemented using PyTorch functions, which drastically reduced the forward and backward propagation times for a batch from seconds to milliseconds. However, this resulted in an increase in GPU memory usage and training was not possible again. At this point, techniques to reduce GPU memory usage are currently being investigated. In addition, online alternatives that allow high GPU memory usage are being tested, such as Paperspace¹, which allows free access to GPU memory up to 32 GB for a limited amount of time. As an attempt to reduce memory usage, the calculation of partition maps was changed from fine-to-coarse to coarse-to-fine manner. Since the first partition map is the largest in size and uses the finest threshold from its α -tree, this resulted in too many connected components to fit in memory. This approach reduced memory allocation, but still requires more space. This problem is still under investigation.

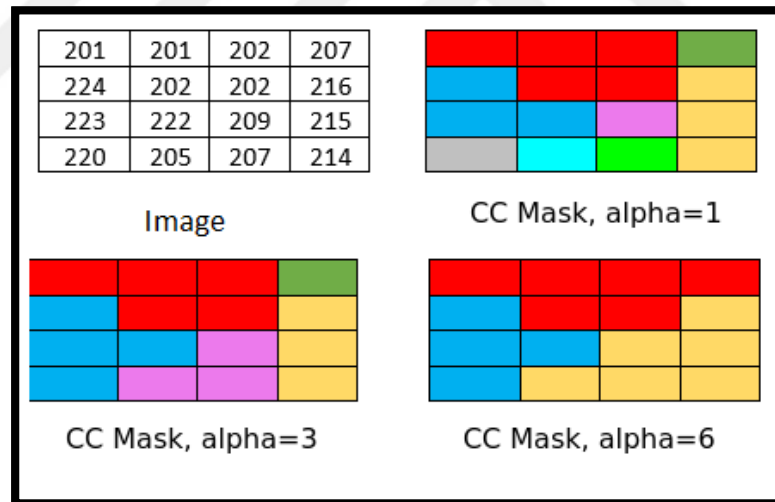


Figure D.1.4: An example of thresholding the α -tree of the image with different values.

In this chapter, a custom pooling for semantic segmentation is proposed. Pooling layers are an essential part of CNNs because they produce effective feature maps from large scenes by downsampling the input. However, there is a trade-off between the number of pooling layers, which introduce increasingly abstract features along with

¹ www.paperspace.com, accessed on February 28th, 2022

convolutional layers, and the loss of contour precision due to downsampling of the input.

This loss of precision is particularly important in analysis of RS images due to the multi-scale appearance of phenomena. Instances of phenomena represented with a small number of pixels (e.g., cars) may be lost during the progressive downsampling inherent in pooling, resulting in poor classification performance for that class and thus a poor performing model.

The proposed pooling mechanism addresses this problem using an efficiently computed multiple-scale hierarchical representation. Instead of using regular grid-based pooling windows, it uses the partition maps obtained from different hierarchies (scales) during downsampling. In the upsampling phase, these hierarchies, which represent the input at different scales, force the model to retain contour information, contributing to the model's performance.

To test this approach, the pooling and unpooling layers of the SegNet architecture are replaced with the proposed pooling and unpooling layers. The approach has been implemented on a high-resolution remote sensing dataset, but due to GPU memory issues, the results cannot be obtained.