DEVELOPING A DEEP LEARNING NETWORK SUITABLE FOR AUTOMATED CLASSIFICATION OF HETEROGENEOUS LAND COVERS IN HIGH SPATIAL RESOLUTION IMAGERY

MOHAMMAD REZAEE

February 2019

TECHNICAL REPORT NO. 317

DEVELOPING A DEEP LEARNING NETWORK SUITABLE FOR AUTOMATED CLASSIFICATION OF HETEROGENEOUS LAND COVERS IN HIGH SPATIAL RESOLUTION IMAGERY

Mohammad Rezaee

Department of Geodesy and Geomatics Engineering
University of New Brunswick
P.O. Box 4400
Fredericton, N.B.
Canada
E3B 5A3

February 2019

PREFACE

This technical report is a reproduction of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Geodesy and Geomatics Engineering, February 2019. The research was supervised by Dr. Yun Zhang, and support was provided by the New Brunswick Innovation Foundation, Canada Research Chairs, and the Natural Sciences and Engineering Research Council of Canada.

As with any copyrighted material, permission to reprint or quote extensively from this report must be received from the author. The citation to this work should appear as follows:

Rezaee, Mohammad (2019). *Developing a Deep Learning Network Suitable for Automated Classification of Heterogeneous Land Covers in High Spatial Resolution Imagery*. Ph.D. dissertation, Department of Geodesy and Geomatics Engineering Technical Report No. 317, University of New Brunswick, Fredericton, New Brunswick, Canada, 133 pp.

ABSTRACT

The incorporation of spatial and spectral information within multispectral satellite images is the key for accurate land cover mapping, specifically for discrimination of heterogeneous land covers. Traditional methods only use basic features, either spatial features (e.g. edges or gradients) or spectral features (e.g. mean value of Digital Numbers or Normalized Difference Vegetation Index (NDVI)) for land cover classification. These features are called low level features and are generated manually (through so-called *feature engineering*). Since feature engineering is manual, the design of proper features is time-consuming, only low-level features in the information hierarchy can usually be extracted, and the feature extraction is application-based (i.e., different applications need to extract different features).

In contrast to traditional land-cover classification methods, Deep Learning (DL), adapting the artificial neural network (ANN) into a deep structure, can automatically generate the necessary high-level features for improving classification without being limited to low-level features. The higher-level features (e.g. complex shapes and textures) can be generated by combining low-level features through different level of processing.

However, despite recent advances of DL for various computer vision tasks, especially for convolutional neural networks (CNNs) models, the potential of using DL for land-cover classification of multispectral remote sensing (RS) images have not yet been thoroughly explored. The main reason is that a DL network needs to be trained using a huge number of images from a large scale of datasets. Such training datasets are not usually available in RS. The only few available training datasets are either for object

detection in an urban area, or for scene labeling. In addition, the available datasets are mostly used for land-cover classification based on spatial features. Therefore, the incorporation of the spectral and spatial features has not been studied comprehensively yet.

This PhD research aims to mitigate challenges in using DL for RS land cover mapping/object detection by (1) decreasing the dependency of DL to the large training datasets, (2) adapting and improving the efficiency and accuracy of deep CNNs for heterogeneous classification, (3) incorporating all of the spectral bands in satellite multispectral images into the processing, and (4) designing a specific CNN network that can be used for a faster and more accurate detection of heterogeneous land covers with fewer amount of training datasets.

The new developments are evaluated in two case studies, i.e. wetland detection and tree species detection, where high resolution multispectral satellite images are used. Such land-cover classifications are considered as challenging tasks in the literature. The results show that our new solution works reliably under a wide variety of conditions. Furthermore, we are releasing the two large-scale wetland and tree species detection datasets to the public in order to facilitate future research, and to compare with other methods.

TO MY WIFE,
FOR HER SUPPORT, ENCOURAGEMENT, AND INSPIRATION

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend sincere thanks to those people who made this work achievable. First, I would like to thank my supervisor, Prof. Yun Zhang, for his constant trust, guidance, and encouragement throughout my Ph.D. program, and other supports that he provided. Second, I want to thank Dr. Fan-Rui Meng and his student, Joan Grau, for their help to collect data for identifying individual trees using images acquired by their UAV and local observations. I also would like to show my gratitude to the biology department at the Memorial University for collecting training data for Wetland Detection project. Advices given by Dr. Shabnam Jabari and all the assistance provided by David Fraser and other members of CRC-Lab in Advanced Geomatics Image Processing have been a great help during my Ph.D. My special thanks are extended to the faculty members and staffs of Geodesy and Geomatics Engineering department in the University of New Brunswick, especially Dr. David Coleman, for all their support. Finally, I would like to thank my wife, Fatemeh Zahra, for her unconditional love, support, patience, and understanding.

Table of Contents

List of Tables

List of Figures

List of Symbols, Nomenclature or Abbreviations

DL	- Deep learning
OD	- Object detection
ANN	- Artificial neural networks
SVM	- Support vector machine
RF	- Random forest
GB	- Gradient boosting
MRF	- Markov random field
CRF	- Conditional random field
CK	- Composite kernel methods
ML	- Machine learning
DBN	- Deep belief net
CNN	- Convolutional neural network
ReLU	- Rectified linear unit
ITSD	- Individual tree species detection
CWCS	- Canadian Wetland Classification System
GPS	- Global Positioning System
PBIL	- Patch-based image labeling
Conv	- Convolution layer

ILSVRC - ImageNet Large Scale Visual Recognition Challenge

NDVI - Normalized difference vegetation index

NDWI - Normalized difference water index

ReNDVI - Red Edge Normalized Difference Vegetation Index

SGD - Stochastic Gradient Descent

OA - Overall accuracy

UAV - Unmanned aerial vehicle

DITDN - Deep individual tree detection network

VGG16 - Visual Geometry Group

DSM - Digital-surface model

LiDAR - Light detection and ranging

SWIR - Short-wave-infrared

VNIR - Visible-and-near-infrared

Pan - Panchromatic

TPE - Tree-structure parzen estimator

1 Chapter 1: Introduction

This PhD dissertation reviews, examines, and improves state-of-the-art deep learning (DL) techniques for remote sensing (RS) data analysis. This article-based dissertation presents the following papers:

1. Rezaee, M., Mahdianpari, M., Zhang, Y., & Salehi, B. (2018). Deep Convolutional Neural Network for Complex Wetland Classification Using Optical Remote Sensing Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, (99).
2. Rezaee, M., Mahdianpari, M., Zhang, Y., & Salehi, B. (2018). Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sensing of Environment*. Under review
3. Rezaee, M., Tong, F., Mishra, R., Zhang, Y. (2018). Detection of Individual Tree Species Using an Optimized Deep CNN in an Object-Based Approach. *Remote Sensing of Environment*. Under review

1.1 Dissertation Structure

This article-based dissertation consists of five chapters: chapter 1 provides an introduction to the research, chapters 2 through 4 present three peer-reviewed journal papers, published or under review; and chapter 5 presents the summary of the work and conclusions.

1.2 Background

Satellite images are being increasingly used to provide high levels of information and detail in various applications, such as urban monitoring and geographical information system updating. The improvement of spatial and spectral resolution in satellite images has undeniably increased usage of these images in different applications. However, having high spatial and spectral resolution has also generated complexities to the processing of satellite images for object detection (OD) or land cover mapping. These complexities in the processing can be described as high intra-class spectral variations due to the images fine detail, shadow and occlusion, and the increase in data volume that is supposed to be processed because of the high spatial, spectral, and radiometric resolutions of the data (Shu, 2014). These challenges have accelerated the need to develop new methods for processing satellite images (Cai & Liu, 2013; Li, 2014). As a result of these attempts, different trends for improving the satellite image processing can be observed.

The most important trend for improving the satellite image processing is the increase of classifiers' power to discriminate the objects. The increase in classification power started in the 1990s with the use of machine learning (ML) method with nonlinear boundaries for discrimination of objects, including the introduction of artificial neural networks (ANN; Benediktsson, Swain, & Ersoy, 1990), support vector machines (SVM; Song, Fan, & Rao, 2005), random forest (RF; Ball, Anderson, & Chan, 2017), and gradient boosting (GB; Zhang, Du, & Zhang, 2016). While the accuracy of detection and classification using these new methods has increased, different studies mention that the original images are not sufficient to conquer the data's complexity (Mnih & Hinton, 2010). This could be

related to the spectral similarity of the objects/land-covers in the satellite images, which makes the raw spectral information insufficient for the classification and detection of heterogeneous objects/land covers.

The spectral data insufficiency is responsible for the second trend in dealing with the challenges of high-spatial-resolution satellite images: incorporating both spectral and spatial information for land cover mapping. The studies in the second trend were based on the Markov random field (MRF) model (Q Jackson & Landgrebe, 2002), the conditional random field (CRF) model (Zhong & Wang, 2010), and composite kernel methods (CK; Camps-Valls, Gomez-Chova, Muñoz-Marí, Vila-Francés, & Calpe-Maravilla, 2006). These studies tried to augment spectral information with spatial features. These attempts, however, needed an extraction of a large number of spectral/spatial features, the so-called feature engineering, which is time-consuming and expert-knowledge dependant (about the relationship between the data and the desired objects) since it is manually designed (Chollet, 2017; LeCun, Bengio, & Hinton, 2015). Most generated features are dependent on location, image type, and desired objects for detection (Zhao & Du, 2016). In addition, the extracted features rely mostly on either spatial (e.g. edges) or spectral features (e.g. mean value of Digital Numbers or Normalized Difference Vegetation Index (NDVI)) that are called low-level features (Jiang, Hauptmann, & Xiang, 2012), which for complex objects and land covers are insufficient (Zhao & Du, 2016).

Recently, DL algorithms have become state-of-the-art for image classification and object detection in the machine learning field due to their ability to learn discriminative and representative hierarchical features (L. Zhang, Zhang, & Kumar, 2016). DL is

characterized by an ANN with (usually) more than two hidden layers (Zhu et al., 2017). The term *deep* is assigned to these ANNs due to the number of hidden layers they have. Like other shallow ANNs, which have existed since the 1980s, deep ANNs try to extract feature representations that are learned directly from data in a supervised manner (using training data). They can exploit hierarchical feature representations of different levels, while shallow ANNs are limited.

DL is inspired by the human brain's high efficiency for object recognition in its hierarchical learning of multi-level features in the primate visual system (Serre et al., 2007). The human brain, in this way, motivated researchers to alter shallow ANNs by finding a solution for the vanishing gradient problem (Pascanu, Mikolov, & Bengio, 2012)—specifically, the modification of the ANN's depth (increasing its number of hidden layers). Recent advances in the application of DL to various applications have proven this method's capability in different fields, including machine learning (Musterie, Chuang, & Chan, 2016; Zheng et al., 2016), medical imagary (Dou et al., 2017), and remote-sensing fields (Mnih, 2013; Wang, Zhang, Liu, Choo, & Huang, 2017).

Different DL models have been used for visual-data processing since 2006 (Zhu et al., 2017). Deep belief net (DBN; Hinton, Osindero, & Teh, 2006), stacked auto-encoders (SAE; Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010), and deep convolutional neural network (CNN; Krizhevsky, Sutskever, & Hinton, 2012; Szegedy et al., 2015) are deep-learning models currently in use. All these models follow a general workflow (Figure 1.1) that includes three main parts: input data, the core DL model, and the output map. The input-output pair can be different based on their application. After

determination of the input-output pair and the DL network, the relationship between the input-output pair will be established through the DL network.

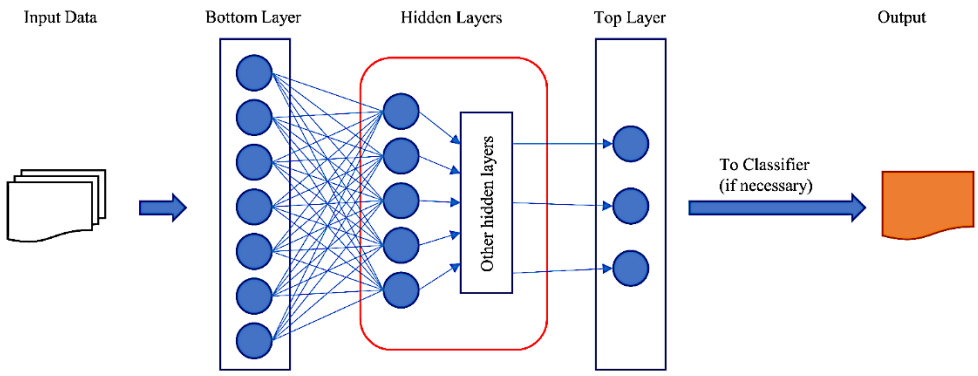


Figure 1-1 A general workflow of DL for image analysis

Convolutional neural network (CNN), the most well-known network amongst DL networks, is a multilayer architecture composed of multiple trainable stages that act as feature extractors. Each stage may be composed of three types of layers: a convolutional layer, a nonlinearity layer, and a pooling layer. The network then connects to a typical fully connected ANN that receives the features and behaves as a classifier. Because of the convolution layer's designation, CNN can take advantage of the image dimensions it receives as input. Each layer type can be described as follows:

- Convolutional layer: This layer consists of a rectangular grid. Each of its neurons inside the grid picks data from the preceding layer. The selected data are then multiplied by corresponding weights to generate new values. The new values are fed to the next layer as input. In fact, the rectangular weights act as a kernel, and the whole process is like convolution.

- Nonlinearity layer: In this layer, a nonlinear function applies to the result of the convolutional layer in a pointwise manner and generates the final feature maps. The rectified linear unit (ReLU) function is commonly used in this layer.
- Pooling layer: This layer selects a small rectangular part of the previous layer and subsamples it to one output unit. Different methods are used for subsampling, such as averaging (Avg. pooling), selecting maximum value (max pooling), or using a learned linear combination. In CNN, the max pooling is usually used.

With the availability of training data and computation resources for RS, use of DL in the RS field has increased. However, the inherent statistical differences between machine-learning images and remote-sensing images raises new concerns that lead to new challenges in applying DL to RS on a large scale. These new challenges are as follows:

- The amount of available data with known RS-data labels is limited. This limitation relates to the high price of RS images, the time-consuming and expensive process of preparing labels for data, and the application-specific nature of RS-data labeling. During the last decades, a few datasets have been prepared and released publicly. However, most focus on urban areas. Since DL needs a large dataset for training, the unavailability of training data becomes the bottleneck for applying DL to RS images.
- RS data is geolocated—every pixel in satellite images is assigned to a specific location on Earth. Some DL processing, such as pooling or subsampling, can distort the pixel's position. This distortion can lead to a situation where a pixel is assigned to an incorrect position within a specific confidence ellipse, which can then distort the detected object's shape.

- Continuous image acquisition in the RS field raises the possibility of considering time factor in the processing of satellite images. In this case, the processing unit moves from individual images to multiple images that are taken from a location in different times. To accommodate this ability, the network should be extendable, and the generated features should be general, so the filters can be applied to the time-series images all at once, while obtaining promising results. The dynamic changes of some objects in the RS scenes render the extendibility and generality a new challenge for RS data.
- DL, specifically CNN, works based on the spatial distribution of data. Different studies have reported the importance of spatial information in the data for object detection. Some studies even tried to remove the colour variation in images to decrease the sensitivity of objects to spectral information. However, in most RS applications, such as wetland detection, the spectral information is the main information for the discrimination of objects.
- High spatial and spectral RS data represents a big data challenge. With recent satellites like Worldview-3, the size of the images is very large. If time-series analysis is necessary, the size of images can grow even more. With this type of data, the speed of application is a challenge in real-world applications.
- Machine-learning images are usually composed of three bands related to the area of wavelength spectrum that we know as red, green, and blue. However, in most RS images, the number of bands can range from 3 to more than 200. Each band is related to a specific range in wavelength spectrum, so the digital number of the

image has a physical meaning (the reflectance of the object on that wavelength). All DL libraries currently work with 3 bands.

1.3 Research Questions

These aforementioned challenges raise questions regarding the application of DL to RS images:

1. Can CNN be used with limited RS training data?
2. Can CNN detect different objects mostly based on their spectral information? In other words, is CNN sensitive to spectral information as much as spatial information for detection? Can CNN incorporate spectral and spatial features at a higher level of information hierarchy?
3. Is there any specific network architecture more reliable for processing RS images?
4. Considering the heavy computational process of DL and the large image sizes in RS, can DL reach a processing speed that can be used in real-world application? If not, is there any way to reach a processing speed comparable to other in-use methods of land cover mapping?
5. Since training a DL network is expensive in terms of the amount of data and time it needs, can a CNN be reused on other images for the same purpose? In other words, are CNN features general enough to be applied to other images they are not trained with?

1.4 Research Objectives

The general objectives of this thesis can be defined based on the above questions. These objectives are as follows:

- Determine the suitability of CNNs for classification and object detection of RS images
- Examine the power of deep CNNs for the classification of spectrally similar classes
- Compare the efficiency of the most well-known deep CNNs
- Generate an appropriate model for classifying spectrally similar classes based on comparison by optimization of the network and its parameters to better fit the data
- Explore the use of transfer learning and fine-tuning for working with limited training data. Explore whether full training or fine-tuning for exploiting the pre-existing CNN model is the optimal strategy for classification and object detection of RS images
- Take advantage of the object-based approach for speeding up CNN deployment
- Eliminate the limitation of the number of input bands by developing a pipeline in Python with the capacity to operate with multi-layer remote sensing imagery
- Investigate the generalization capacity of existing CNNs for the classification of multispectral satellite imagery (i.e., a different dataset than those they were trained for)
- Compare results of existing CNNs and the designed CNN model with state-of-the-art classifiers RF and GB

1.5 Structure of the Thesis

Chapter 1 is an introduction to this dissertation, presenting background, problem statements, research questions and objectives, and an overview of each chapter.

Chapter 2 to 4 comprise this dissertation's 3 journal papers and its main contribution of research:

- Chapter 2 evaluates the capability of the state-of-the-art classification tool deep CNN for classifying spectrally similar land objects (Wetland classes) using limited training data. This chapter tries to answer the first and second research questions.
- Chapter 3 compares the most state-of-the-art CNN networks and their accuracy in the consideration of different scenarios related to the availability of the training data. It also addresses the limitation of the CNN input image layer by taking all the multispectral bands into account for processing. This chapter studies the first, second, and fifth research questions.
- Chapter 4 presents a solution to individual tree species detection (ITSD), using CNN and an object-based approach. This chapter addresses the third, fourth, and fifth research questions. We introduce a new architecture for ITSD that is fast, reliable, and independent of large-scale training data.

Chapter 5 presents the summary of the work accomplished in this research, concluding remarks, contribution of research, and recommendation for future work.

References

Benediktsson, J. a, Swain, P. H., & Ersoy, O. K. (1990). Neural Network Approaches Versus Statistical Methods In Classification Of Multisource Remote Sensing Data. IEEE Transactions on Geoscience and Remote Sensing, 28(4), 540–552. https://doi.org/10.1109/TGRS.1990.572944

Bengio, Y. (2009). Learning deep architectures for AI. Foundations and Trends® in Machine Learning, 2(1), 1–127.

Brisco, B., Ahern, F., Murnaghan, K., White, L., Canisus, F., & Lancaster, P. (2017). Seasonal Change in Wetland Coherence as an Aid to Wetland Monitoring. Remote Sensing, 9(2), 158–176.

Cai, S., & Liu, D. (2013). A comparison of object-based and contextual pixel-based classifications using high and medium spatial resolution images. Remote Sensing Letters, 4(10), 998–1007.

Camps-Valls, G., Gomez-Chova, L., Muñoz-Marí, J., Vila-Francés, J., & Calpe-Maravilla, J. (2006). Composite kernels for hyperspectral image classification. IEEE Geoscience and Remote Sensing Letters, 3(1), 93–97.

Chen, X., Xiang, S., Liu, C.-L., & Pan, C.-H. (2014). Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geoscience and Remote Sensing Letters, 11(10), 1797–1801.

Chen, Y., Lin, Z., Zhao, X., Wang, G., & Gu, Y. (2014). Deep learning-based classification of hyperspectral data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7(6), 2094–2107.

Chollet, F. (2017). Deep Learning with Python. Manning Publications Company. Retrieved from https://books.google.ca/books?id=Yo3CAQAACAAJ

DiCarlo, J. J., Zoccolan, D., & Rust, N. C. (2012). How does the brain solve visual object recognition? Neuron, 73(3), 415–434.

Dou, Q., Yu, L., Chen, H., Jin, Y., Yang, X., Qin, J., & Heng, P.-A. (2017). 3D deeply supervised network for automated segmentation of volumetric medical images. Medical Image Analysis. https://doi.org/10.1016/j.media.2017.05.001

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18(7), 1527–1554.

Jackson, Q., & Landgrebe, D. A. (2002). Adaptive Bayesian contextual classification based on Markov random fields. IEEE Transactions on Geoscience and Remote Sensing, 40(11), 2454–2463. https://doi.org/10.1109/TGRS.2002.805087

Jackson, Q., & Landgrebe, D. A. (2002). Adaptive Bayesian contextual classification based on Markov random fields. IEEE Transactions on Geoscience and Remote Sensing, 40(11), 2454–2463.

Jiang, L., Hauptmann, A. G., & Xiang, G. (2012, October). Leveraging high-level and low-level features for multimedia event detection. In Proceedings of the 20th ACM international conference on Multimedia (pp. 449-458). ACM.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012a). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097–1105).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012b). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25 (pp. 1097–1105). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Landgrebe, D. A. (2005). Signal theory methods in multispectral remote sensing (Vol. 29). John Wiley & Sons.

Längkvist, M., Kiselev, A., Alirezaie, M., & Loutfi, A. (2016). Classification and segmentation of satellite orthoimagery using convolutional neural networks. Remote Sensing, 8(4), 329–349.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. Retrieved from http://dx.doi.org/10.1038/nature14539

Li, M. (2014). A Review of Remote Sensing Image Classification Techniques: the Role of Spatio-contextual Information. European Journal of Remote Sensing, 389–411. https://doi.org/10.5721/EuJRS20144723

Loosvelt, L., Peters, J., Skriver, H., De Baets, B., & Verhoest, N. E. C. (2012). Impact of reducing polarimetric SAR input on the uncertainty of crop classifications based on the random forests algorithm. IEEE Transactions on Geoscience and Remote Sensing, 50(10), 4185–4200.

Mahdianpari, M., Salehi, B., Mohammadimanesh, F., Brisco, B., Mahdavi, S., Amani, M., & Granger, J. E. (2018). Fisher Linear Discriminant Analysis of coherency

matrix for wetland classification using PolSAR imagery. Remote Sensing of Environment, 206, 300–317.

Mahdianpari, M., Salehi, B., Mohammadimanesh, F., & Motagh, M. (2017). Random forest wetland classification using ALOS-2 L-band, RADARSAT-2 C-band, and TerraSAR-X imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 130, 13–31.

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

Mnih, V. (2013). Machine Learning for Aerial Image Labeling, 109.

Mnih, V., & Hinton, G. E. (2010). Learning to detect roads in high-resolution aerial images. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6316 LNCS(PART 6), 210–223. https://doi.org/10.1007/978-3-642-15567-3_16

Mohammadimanesh, F., Salehi, B., Mahdianpari, M., Brisco, B., & Motagh, M. (2018). Multi-temporal, multi-frequency, and multi-polarization coherence and SAR backscatter analysis of wetlands. ISPRS Journal of Photogrammetry and Remote Sensing, 142, 78–93. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2018.05.009

Musterie, P., Chuang, Y., & Chan, C. (2016). Berkeley Deep Drive Project An Approach to Pedestrian Detection for Autonomous Driving Using Deep Learning, 1–2.

Nogueira, K., Penatti, O. A. B., & dos Santos, J. A. (2017). Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification. Pattern Recognition, 61, 539–556.

Pascanu, R., Mikolov, T., & Bengio, Y. (2012). On the difficulty of training Recurrent Neural Networks. https://doi.org/10.1109/72.279181

Scholkopf, B., & Smola, A. J. (2001). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, MA, USA: MIT Press.

Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., & Poggio, T. (2007). A quantitative theory of immediate visual recognition. Progress in Brain Research, 165, 33–56.

Shu, Y. (2014). Deep Convolutional Neural Networks for Object Extraction from High Spatial Resolution Remotely Sensed Imagery. University of Waterloo.

Song, X., Fan, G., & Rao, M. (2005). Automatic CRP Mapping Using Machine Learning Approaches. IEEE Transactions on Geoscience and Remote Sensing, 43(4), 888–897.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1–9).

Tiner, R. W., Lang, M. W., & Klemas, V. V. (2015). Remote sensing of wetlands: applications and advances. CRC Press.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(Dec), 3371–3408.

Wang, L., Zhang, J., Liu, P., Choo, K.-K. R., & Huang, F. (2017). Spectral–spatial multi-feature-based deep learning for hyperspectral remote sensing image classification. Soft Computing, 21(1), 213–221. https://doi.org/10.1007/s00500-016-2246-3

Wdowinski, S., Kim, S.-W., Amelung, F., Dixon, T. H., Miralles-Wilhelm, F., & Sonenshein, R. (2008). Space-based detection of wetlands' surface water level changes from L-band SAR interferometry. Remote Sensing of Environment, 112(3), 681–696.

Zhang, F., Du, B., & Zhang, L. (2016). Scene classification via a gradient boosting random convolutional network framework. IEEE Transactions on Geoscience and Remote Sensing, 54(3), 1793–1802.

Zhang, L., Zhang, L., & Kumar, V. (2016). Deep learning for Remote Sensing Data. IEEE Geoscience and Remote Sensing Magazine, 4(2), 22–40. https://doi.org/10.1155/2016/7954154

Zhao, W., & Du, S. (2016). Learning multiscale and deep representations for classifying remotely sensed imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 113, 155–165.

Zheng, Y., Zhu, C., Luu, K., Bhagavatula, C., Le, T. H. N., & Savvides, M. (2016). Towards a deep learning framework for unconstrained face detection. 2016 IEEE 8th

International Conference on Biometrics Theory, Applications and Systems (BTAS). https://doi.org/10.1109/BTAS.2016.7791203

Zhong, P., & Wang, R. (2010). Learning conditional random fields for classification of hyperspectral images. IEEE Transactions on Image Processing, 19(7), 1890–1907.

Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. IEEE Geoscience and Remote Sensing Magazine. https://doi.org/10.1109/MGRS.2017.2762307

2 Chapter 2: Deep Convolutional Neural Network for Complex Wetland Classification Using Optical Satellite Imagery

Abstract

The synergistic use of spatial features with spectral properties of satellite images enhances thematic land cover information, which is of great significance for complex land cover mapping. Incorporating spatial features within the classification scheme, which has previously been mainly carried out by applying only low-level features, has shown improvement in the classification results. However, the application of high-level spatial features for classification of satellite imagery has been underrepresented. This study aims to address the neglect of high-level features by proposing a classification framework based on convolutional neural network (CNN) to learn deep spatial features for wetland mapping using optical remote sensing data. Designing a fully trained new convolutional network is infeasible due to the limited amount of training data in most remote sensing studies. Thus, we applied fine-tuning of a pre-existing CNN. Specifically, AlexNet was used for this purpose. The classification results obtained by the deep CNN were compared with those based on well-known ensemble classifiers, namely Random Forest (RF), to evaluate the efficiency of CNN. Experimental results demonstrated that CNN was superior to RF for complex wetland mapping even by incorporating the small number of input features (i.e., 3 features) for CNN compared to RF (i.e., 8 features). The proposed classification scheme is the first attempt to investigate the potential of fine-tuning pre-existing CNN for land cover mapping. It also serves as a baseline framework

to facilitate further scientific research using the latest state-of-art machine learning tools for processing remote sensing data.

2.1 Introduction

Wetlands are transitional zones between a water body and dry land, which may experience wet conditions permanently or at least periodically during high water seasons (Tiner, Lang, & Klemas, 2015). Wetlands support several environmental services, including flood storage, carbon sequestration, shoreline stabilization, and water-quality renovation, and provide a favorable habitat for several unique aquatic vegetation and animal species. Despite their high contributions to the ecosystem, they have been threatened by the anthropogenic and natural processes during past decades. In particular, human activities progressively converted wetlands to non-wetland areas due to agriculture irrigation, road construction, and pollution. Global warming, flooding, shoreline erosion, and sea level rise further expedite wetland loss through natural processes (Tiner et al., 2015).

Given the numerous benefits of wetlands for the ecosystems, their restoration and preservation of wetlands are critical. Thanks to the advancement of remote sensing techniques, their extend and condition can be monitored globally. In particular, remote sensing tools have significantly contributed to the wetland mapping and monitoring in a variety of aspects, including classification (Mahdianpari et al., 2018; Mahdianpari, Salehi, Mohammadimanesh, & Motagh, 2017; Mohammadimanesh, Salehi, Mahdianpari, Brisco, & Motagh, 2018), change detection (Brisco et al., 2017), and water level monitoring (Wdowinski et al., 2008).

Despite significant improvements in remote sensing tools in both satellite image and applied techniques, classification of complex heterogeneous land cover such as wetland is challenging. This is because, in a highly fragmented landscape such as wetland, there are several small classes without clear-cut borders between them, which in turn increases the within-class variability and decreases between class separability. Furthermore, some of these classes may have very similar spectral characteristics, which further complicates the matter. Thus, combining spatial feature with spectral information may contribute to differentiating complex land cover (Zhao & Du, 2016). Several approaches have been proposed in order to evaluate the efficiency of integrating spectral-spatial features for classification, including kernel methods (Scholkopf & Smola, 2001), Bayesian models (Landgrebe, 2005), Markov Random Field (MRF) (Qiong Jackson & Landgrebe, 2002), and Conditional Random Field (CRF) (Zhong & Wang, 2010). However, these methods apply to low-level features such as spectral information within neighboring pixels or morphological properties. Thus, the main disadvantage associated with these techniques is setting the proper parameters in order to produce suitable features for the different image objects (Zhao & Du, 2016). Furthermore, most of these algorithms are only fitted to the particular problem (e.g., specific case study and datasets), while they are inappropriate in other cases (Längkvist, Kiselev, Alirezaie, & Loutfi, 2016).

Inspired by high efficiency of the human brain in object recognition, high-level spatial features produced by hierarchical learning have attracted substantial interest in several applications, such as object recognition, scene labeling, and document analysis (DiCarlo, Zoccolan, & Rust, 2012). In particular, Deep Learning is one of the most well-known approaches to obtaining high-level spatial features (Jiang, et.al., 2012), using a

hierarchical learning framework. It works based on a multi-layer interconnected neural network framework that learns features and classifiers simultaneously (Nogueira, Penatti, & dos Santos, 2017). Specifically, a single network with multiple layers may be utilized to learn features and classifiers and exploit the parameters depending on the problem and accuracy demanded. This is also known as an end-to-end feature learning framework, wherein the image pixels and semantic labels are input and output of the algorithm, respectively (LeCun et al., 2015).

Convolutional Neural Network (CNN) is one of the most efficient approaches among all deep learning based frameworks that does not require prior feature extraction and thereby has a greater generalization capability (LeCun et al., 2015). This is because a multi-layer-based classifier has a high capacity to exploit abstract and invariable features. In particular, a deep CNN extracts varying levels of abstraction for the data in different layers, for example, low-level (e.g., edges), intermediate level (e.g., object fragment), and high-level information (e.g., full object) obtained in the initial, intermediate, and last layers, respectively (Nogueira et al., 2017).

There are three main strategies to use CNN, including full training, fine-tuning, and pre-training CNN (Nogueira et al., 2017). In a fully trained CNN, a network is built from scratch in order to extract particular visual features based on the applied dataset. Despite the great efficiency and robustness of this method, which provides a full control over the parameters and architecture, this is inappropriate for remote sensing applications. This is because building a network from scratch requires a large amount of training data (Bengio, 2009). The other two approaches are represented as more promising for remote sensing applications since they utilize the pre-trained model, which has been previously trained

using different data. This is possible because the initial layers of CNN are typically general filters (i.e., low-level features such as edges); so, they need little or no updating during the fine-tuning process.

The efficiency of deep CNN has been demonstrated in object detection (Krizhevsky, Sutskever, & Hinton, 2012b), recognition of handwritten characters and traffic signs (X. Chen, Xiang, Liu, & Pan, 2014), and classification (X. Chen et al., 2014). Although the application of CNN was employed in a number of remote sensing studies for classification of different land cover types using hyperspectral imagery (Y. Chen, Lin, Zhao, Wang, & Gu, 2014; Makantasis, Karantzalos, Doulamis, & Doulamis, 2015), its efficiency was not examined for complex land cover classification (e.g., wetland and sea ice). Currently, the classification of the complex land cover is performed by incorporating a large number of input features to address the difficulty of discriminating land cover classes with very similar spectral signatures. However, extracting a large number of input features is not time efficient, while their manipulation could be challenging. Furthermore, some of these input features are highly correlated, which means no improvement in the information content of input data. Accordingly, a large number of feature selection algorithms have been proposed to determine optimum features for different applications (Loosvelt, Peters, Skriver, De Baets, & Verhoest, 2012). Given the main drawbacks of current approaches for classification of complex land cover types, this study aims (i) to evaluate the generalization capacity of pre-trained CNN in the classification of multi-spectral satellite imagery; (ii) to determine the suitability of CNN for complex wetland classification, and (iii) to generate an appropriate model for further wetland mapping studies. In particular, a pre-trained CNN framework was utilized to classify a wetland

ecosystem using a RapidEye multi-spectral imagery in a case study located in Newfoundland and Labrador, Canada. The results of this study are the first attempt to show the potential of CNN for complex land cover mapping with very similar spectral signatures, which facilitates the application of CNN for wetland classification. Given the similarity of wetland species across Canada and the generality of this approach, the results of this study progress towards utilization of CNN for complex wetland classification.

2.2 Method

2.2.1 Study area and dataset

The study site is located in Newfoundland and Labrador, Canada, covering an area of approximately 700 km^2. This province comprises a number of ecoregions with different characteristics depending on hydrology, ecology, and geomorphology. Particularly, this study is carried out in the northeast of this province in the Maritime Barren ecoregion, which is identified by an oceanic climate, foggy/cool summers, and relatively mild winters (Marshall & Schut, 1999). Different wetland classes specified by Canadian Wetland Classification System (CWCS), including bog, fen, marsh, swamp, and shallow-water, are found within the study region (Tiner et al., 2015). However, bog and fen are the dominant wetland types due to the province's climate aiding extensive peatland formation. Other land cover types are upland, urban, and deep-water in this ecoregion. Field data were acquired for 191 sample sites in summers and falls of 2015 and 2016. Spatial distribution of different wetland types, as well as other land cover classes, as

Table 2-1 Testing and training pixel counts for the Avalon reference data

Class	Class Description	#Training Pixels	#Testing Pixels	Total
Bog	Peatland dominated by Sphagnum species	20650	19565	40215
Fen	Peatland dominated by graminoid species	11183	8794	19977
Swamp	Mineral wetlands dominated by woody vegetation	3197	9491	12688
Marsh	Mineral wetlands dominated by emergent graminoid species	10869	5238	16107
Shallow Water	Mineral wetlands dominated by submerged and floating vegetation	6205	5679	11884
Urban	Human-made structures	66339	67125	133464
Deep Water	Deep water areas	62927	89194	152121
Upland	Dry forested upland	73458	89878	163336
Total		254828	294964	549792

determined and recorded along with photographs and field notes to facilitate the wetland boundary delineation. Furthermore, Global Positioning System (GPS) locations were recorded for each visited land cover types in order to both train the classifier and assess the classification accuracy (see Table 2.1).

For classification, two RapidEye optical images in Level 3A (radiometrically and geometrically corrected) that were acquired on June 18 and October 22, 2015, were used. This satellite was launched on August 29th, 2008. Its sensors have 5 *m* spatial resolution with 5 different spectral bands: Blue, Green, Red, Red-edge, and near-infrared (see Figure 2.1.

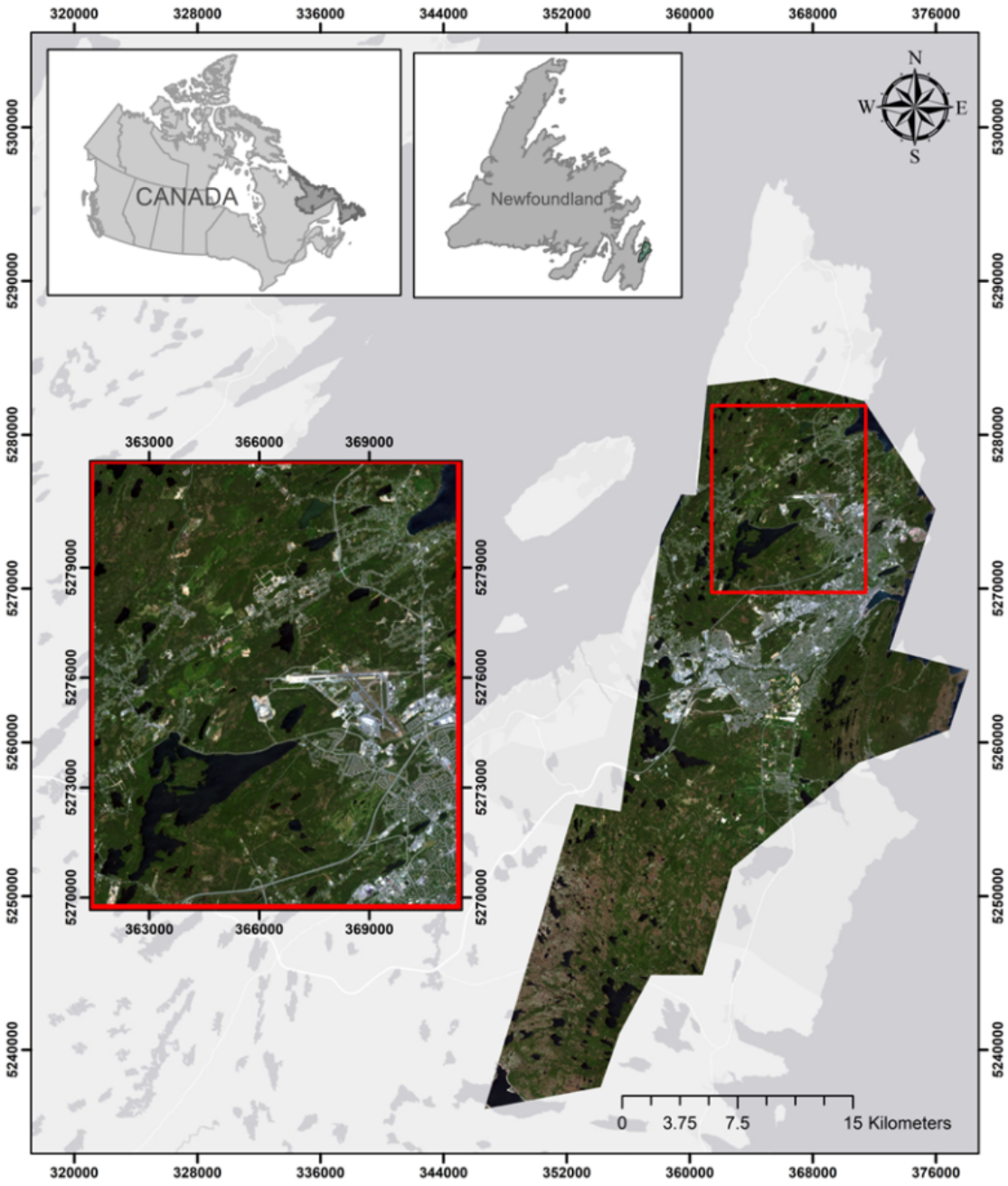


Figure 2-1 A true colour composite of RapidEye optical imagery (bands 3, 2, and 1) acquired on June 18, 2015, illustrating the geographic location of the study area. The red rectangular was selected to display the classified maps obtained from different approaches.

2.2.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is one of the most well-known deep learning algorithms, and has gained interest for image processing in recent years (Zhu et al., 2017). CNN is superior to other deep network algorithms due to its ability to preserve the geometry of the image (i.e., the 2D format). Particularly, it maintains the interconnection between pixels and accordingly, preserves the spatial information. A typical CNN network consists of three types of layers, namely the convolution layer, pooling layer, and fully connected layer (Zhu et al., 2017). The convolution layer extracts information from previous layers and acts as a filter in the image domain. The filter's values also determine the type of information to be extracted. This filter is sensitive to the spatial information and is defined as a rectangular grid inside the layer. This layer is formulated as a simple convolution:

$$feature\ map = input{*}kernel = \sum_{y=0}^{columns}\left(\sum_{x=0}^{rows} input(x-a, y-b)kernel(x,y)\right). \quad (2\text{-}1)$$

Where x and y are concerned with the image matrices while those of a and b deal with that of the kernel. The second layer, the so-called pooling layer, reduces the size of data, and it preserves the most important information and the geometry of the input data. In each pooling layer, a particular number is determined by sub-sampling of a small selected rectangle. There are different methods for subsampling, such as using maximum value or a linear combination (Lee, Gallagher, & Tu, 2016).

The last layer, namely the fully connected layer, is the reasoning part of the network, which determines the final label of the input data. Particularly, each neuron receives the information from all neurons in the previous layers to make the final decision. However, it does not have a role to preserve the geometry as well as turning the input layer into a vector.

2.2.3 Patch-Based Image Labeling (PBIL)

The main goal of a typical convolutional network is to find a unique label for an input image. Although this is a perfect approach to categorize the images, it is far from our goal in remote sensing data processing. Particularly, in the most remote sensing applications (e.g., segmentation and classification), a specific label should be determined for each imaging pixel. Thus, patch-based image labeling methods were introduced to convert categorization problem to classification in order to make CNN compatible with remote sensing applications (Mnih, Heess, & Graves, 2014). In these approaches, an input image is divided into several patches and a label is assigned to the center of each patch. Given the image patches, *S*, and the corresponding target, *M*, the whole problem is defined as a probability approach, wherein the distribution of the image patch over the label is represented as follows (Mnih, 2013):

$$P\big(n(M,i,w_m)\big|n(S,i,w_s)\big). \tag{2-2}$$

where $n(I,i,w)$ indicates a patch with the size of $w \times w$ from the image *I,* which is centered on pixel *i*. The problem is also rewritten in a function form. Given a path from pixel *i* in the input image to the output unit *l* in a multi-class classification, the problem is formulated as:

$$f_{il}(s) = \frac{exp(a_{il}(s))}{Z} = P(m_i = l|s). \tag{2-3}$$

where a_{il} is the total input for the l^{th} output and f_{il} shows the predicted probability mapping pixel i to label l. The network should be trained to find the function, which is typically employed by minimizing the residuals of a predefined function.

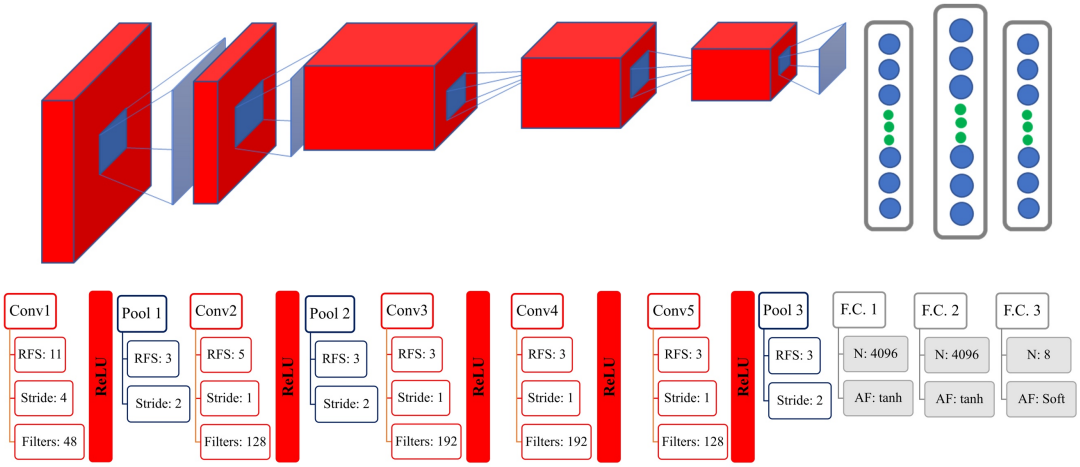


Figure 2.2 The Architecture of AlexNet employed in this study (Conv: Convolution layer, Pool: Pooling layer, F.C.: Fully Connected layer, RFS: Receptive Field Size, N: number of neurons in fully connected layer, AF: Activation Function, Soft: Softmax).

The negative log-likelihood is used for the training procedure in this study, which is formulated as follows:

$$\mathrm{L(s,m)} = \sum_{all\ patches} \sum_{i=1}^{w_m^2} \left(m_i \ln\left(f_i(s)\right) + (1 - m_i) \ln\left(1 - f_i(s)\right)\right). \tag{2-4}$$

The optimization is performed using stochastic gradient descent with mini-batches (Le, Coates, Prochnow, & Ng, 2011). The speed of the optimization is also enhanced through

tuning some hyper-parameters, such as momentum, learning rate, and weight decay. AlexNet, which is the winner of 2012 ImageNet Large Scale Visual Recognition Challenge, is utilized for object detection (ILSVRC) (Krizhevsky et al., 2012). The architecture of this network is shown in Figure 2.2.

As seen in Figure 2.2, this network has eight hidden layers, including five convolution layers and three fully connected layers (Krizhevsky et al., 2012). The adjustment of this network needs less effort due to the relatively small number of layers and accordingly, a smaller number of input parameters relative to other deep networks. Thus, a smaller amount of training data is required to train the network compared to other commonly used networks. This is advantageous for remote sensing data processing given the small number of training samples available in most studies.

2.2.4 Preprocessing step

A comparison of spectral characteristics of four wetland classes was carried out by plotting their signature using 1000 samples (see Figure 2.3).

Error! Bookmark not defined.A band selection technique was employed to reduce the dimensionality of the input data. It is necessary since the AlexNet is designed to receive just 3 bands as input. It also speeds up the training and prediction processes and overcomes the GPU memory limitation. For this purpose, the correlation of different bands of input data was obtained. The highest correlation was associated with the blue and red bands (0.95) and also red-edge and near-infrared bands (0.81). Therefore, the blue and red-edge bands were removed, and other processing steps were implemented on green, red, and near-infrared bands.

Other pre-processing steps employed in this study were the preparation of both the network and imagery. In the PBIL approach, the image should be patched into small tiles, and then a label is assigned to the center pixel of each patch.

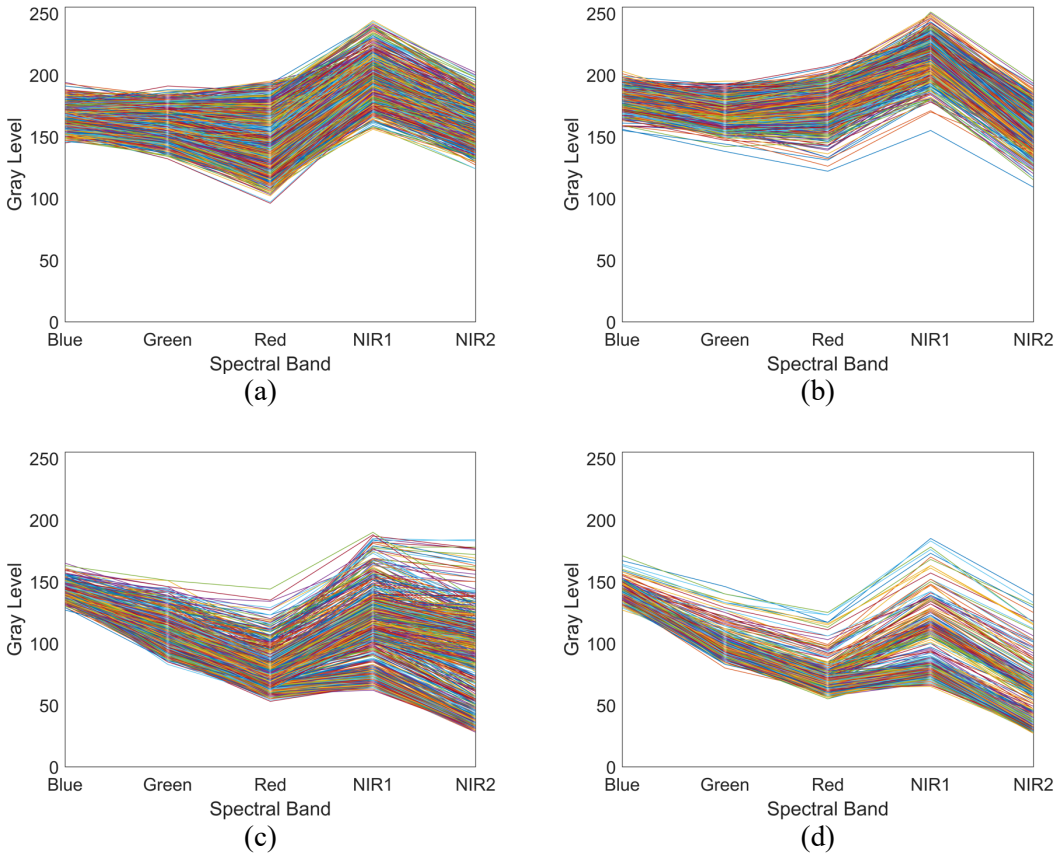


Figure 2-3 The spectral signature of four wetland classes, namely (a) bog, (b) fen, (c) marsh, and (d) swamp obtained using 1000 samples from each class in five bands.

There was a high degree of overlap between patches since the step parameter was adjusted to 1 pixel. It is worth noting that the AlexNet is designed to receive normalized images. Thus, the mean of each patch was set to zero by subtracting the mean value of each patch from the image. Finally, the last pre-processing step was cloud masking in one

of our datasets. This is because the CNN was not trained to classify the cloud and, importantly, it could not have an unclassified label. This means that the network would assign a wrong label to the cloud class if it was not masked out in this step.

2.2.5 Training step

The main challenges associated with the network training were the limited number of training samples and determining an optimum patch size to be utilized in PBIL. Instead of full training a network from scratch, a pre-trained network can be utilized, which addresses the former problem. In this approach, the parameters of the last layers are

Figure 2-4 Sample patches (i) and field surveying images (ii) of four wetland classes, namely (a) bog, (b) fen, (c) marsh, and (d) swamp.

mostly updated, not those of all layers. Furthermore, the update's values are small since the updating is carried out on a pre-trained network. Due to the limited number of *in-situ* data in this study, the last four layers of the network were updated to ensure a sufficient amount of sampling data for both training and testing the network. However, primitive information from the image, such as the size of objects of interest, and the network architecture, is required in order to address the latter problem. In particular, each patch should have enough information to generate a distinct distribution for the specific object

within the image. Different patch-sizes were tested according to spatial resolution of the data and object size of interest (i.e., different wetland classes) from 10 to 40. A patch size of 30 pixels (i.e., 150 *m* on the ground) was found to be an optimum value in this study because a small patch size resulted in overfitting of the model and on the other hand, under segmentation was observed in the case of a large patch size. The CNN network was trained using the Caffe library (Jia et al., 2014). Specifically, the training was carried out on a computer with an Intel® Xenon 2.80GHz CPU (16GB memory) and a Nvidia Quadro k2200 GPU (4GB memory).

2.2.6 Testing step

In order to evaluate the robustness of the CNN result and to prevent information leak from the testing dataset to the model, two strategies were considered during test data preparation. First, to make sure that the testing dataset is independent of the training dataset and both groups had roughly comparable pixel counts, reference polygons in each class were sorted based on their size and alternatingly assigned to testing and training groups. This procedure ensured that both the testing and training groups are selected independently from different parts of the image. Second, to make sure that the network is not over-fitted, the model was trained over first satellite imagery (June 18th) and accuracy indices were determined based on classified map, which was obtained by applying the trained model over the second satellite imagery (October 22). Since the second image was acquired on a different date and was spectrally different from the first image, this procedure illustrates the reliability of results on the testing dataset. We also compared the result of CNN with a state-of-the-art classifier, Random Forest (RF). More specifically, the two main parameters of RF, which should be adjusted are the number of trees (Ntree)

and the number of variables (Mtry) (Belgiu & Drăguţ, 2016). In this study, a total of 500 trees were selected in the classification model. Moreover, the square root of the number of input variables was considered as the number of variables. This is because it decreased both the computational complexity of the model and the correlation between trees (Gislason, Benediktsson, & Sveinsson, 2006).

However, a feature extraction step was an additional step, which is required in such a classifier before the classification step. Therefore, results of applying the CNN model over three original bands of the second image are compared to results of the RF classifier over extracted features in the next section.

2.3 Results and Discussion

The training step was completed using 30,000 iterations in

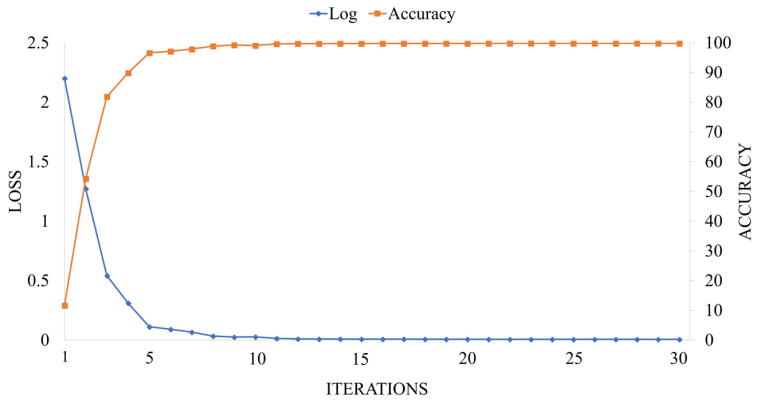


Figure 2-5 The value of validation accuracy and loss as a function of epochs.

approximately 12 hours. Figure 2.5 illustrates the loss and accuracy curve for the validation set. As seen in Figure 2.5, the speed of convergence is high in the initial epochs since the training is actually a fine-tuning of a pre-trained network. To have a

better understanding of the features that were generated and used during the training phase, features of some random patches were extracted. Figure 2.6 depicts the first convolution layer, its corresponding kernels and features in AlexNet.

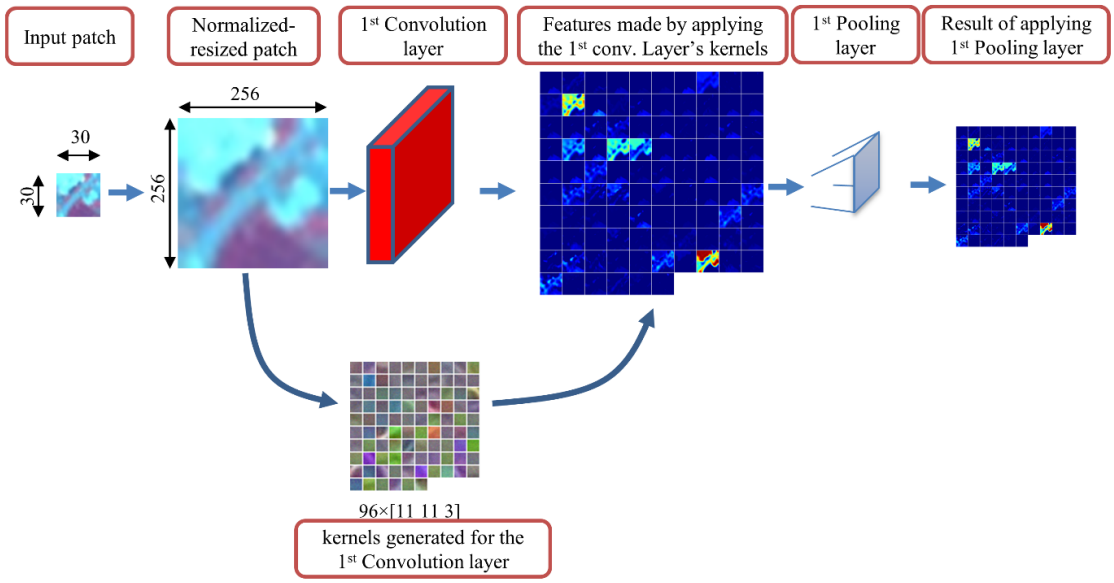


Figure 2-6 The first convolution layer, its designed kernels, and generated features.

Every patch is normalized and resized to feed the network. The designed convolution kernels are then applied to the input patch to generate some normalized features. Each feature highlights a group of similar objects. The kernels are designed to highlight the group of pixels that decrease the loss function. The first layers tend to extract primitive features like edges, and the last layers extract high-level features, such as the pattern.

These high-level features mostly rely on the spatial information in the patches. Incorporating the spatial information into the classification scheme is crucial due to the spectral similarity of wetland classes, which causes a great degree of mixture between them. Figure 2.7 shows some kernels and extracted features for sample patches. These features highlight the area that is related to the corresponding class.

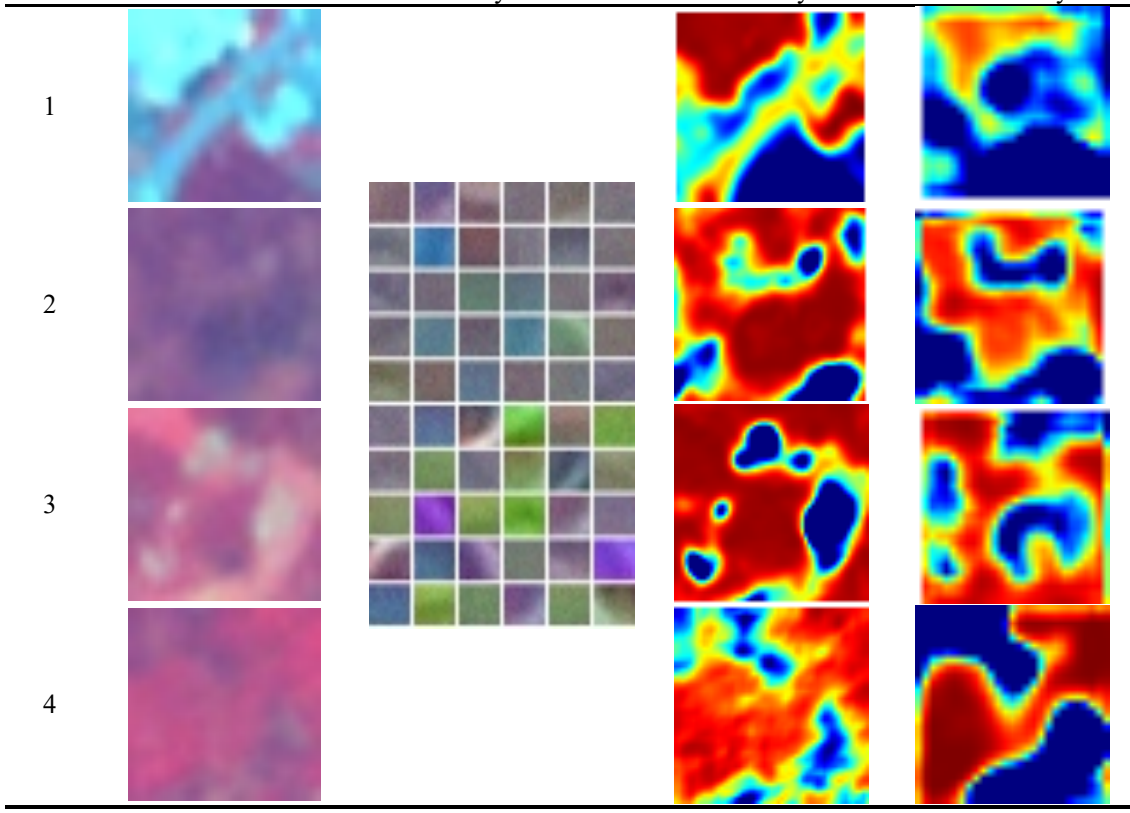


Figure 2-7 Visualization of features related to the first and second convolution layer for four sample patches

To evaluate the efficiency of CNN for wetland mapping, the classification results of CNN were compared with the Random Forest (RF) classifier. RF is an ensembles classifier and has shown good results for several lands cover mappings, such as wetland (Mahdianpari, Salehi, Mohammadimanesh, & Brisco, 2017). For classification based on the RF classifier, a total number of eight features, namely normalized difference vegetation index (NDVI), normalized difference water index (NDWI), Red Edge Normalized Difference Vegetation Index (ReNDVI), as well as all original spectral bands of the RapidEye image were used. However, for CNN only three original spectral bands of the

RapidEye image, including red, green, and near-infrared were applied. The classification maps obtained by RF and CNN are depicted in Figure 2.8

As seen, there is a significant degree of disagreement between two classified maps for all wetland classes. For example, the dominant wetland classes obtained by RF are swamp and marsh wetlands. Whereas, the dominant wetland classes for CNN are bog and fen. As reported by field biologists participating during field data collection, bog and fen wetlands are dominant classes. This is attributed to the oceanic climate of the Avalon area facilitating extensive peatland formation (i.e., bog and fen) (Ecological Stratification Working Group (ESWG), 1995). The dominant non- wetland class is upland, which is defined as forested dry land. Notably, the classified map produced by CNN is realistic and demonstrates the detailed spatial distribution of all land cover classes presented in the study area. For example, the classified map shows the predominance of bog and upland classes, while marsh and swamp are less prevalent. These observations agree well with field notes recorded during field data collection. Confusion matrices for these classified maps are presented in Tables 2.2 and 2.3.

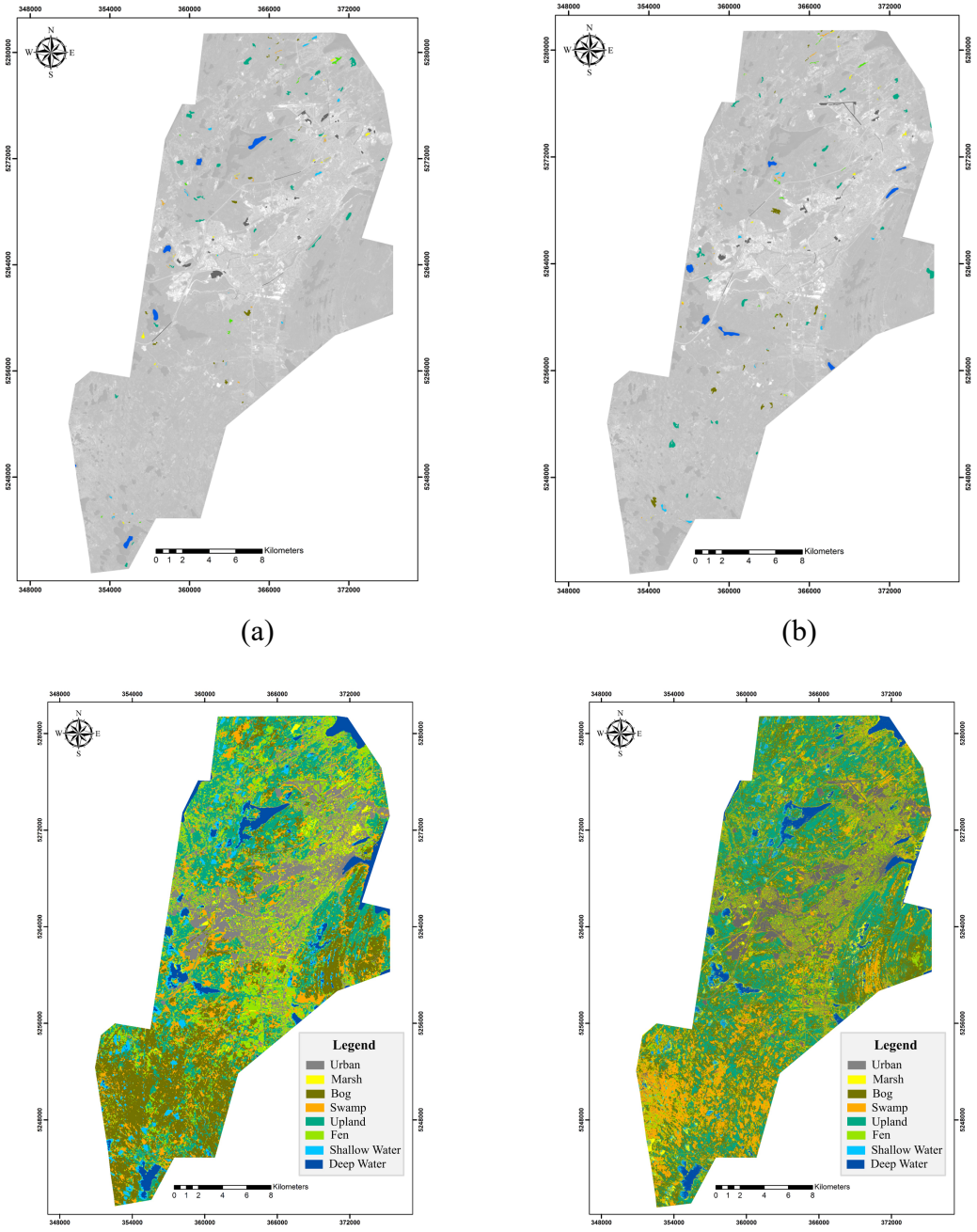


Figure 2-8 (a) The training and (b) testing polygons followed by the classification maps obtained by (c) CNN using three input features and (d) RF using eight input features.

As seen, the classification overall accuracy of about 95% is achieved using CNN by incorporating three input features. This is of great significance taking into account the complexity of similar wetland classes and the large number of pixels, which were correctly classified. In particular, all land cover classes have high producer's accuracies of greater than 77%, excluding the fen class. More precisely, bog is correctly classified in 89% of cases, fen in 62% of cases, swamp in 78% of cases, marsh in 77% of cases, and shallow-water in 95% of cases. As seen, there is a small degree of confusion between different land cover classes indicating small omission and commission errors in most cases. However, the fen class had the lowest producer's accuracy meaning the largest omission error for this class. Particularly, the fen wetland was erroneously classified as bog and upland classes in some cases. This could be due to the relatively small amount of training samples for the fen class and similar characteristics of bog and fen, both of which caused a great degree of confusion. In particular, bog and fen classes are both peatlands dominated by *Sphagnum* and *graminoid* species, respectively, with very similar ecological characteristics. Furthermore, these classes were found to be difficult to distinguish by ecological experts familiar with wetlands in the study area.

Although the classification overall accuracy of about 79% was obtained using RF, all wetland classes had relatively low producer's accuracies. In particular, bog is only correctly classified in 50% of cases, fen in 44% of cases, swamp in 60% of cases, marsh in 64% of cases, and shallow-water in only 31% of cases. Thus, the overall accuracy of 79% is because of the high classification accuracy for non-wetland classes, such as deep-water and urban classes. The confusion matrix illustrates a high degree of confusion between wetland and non-wetland classes. In particular, there is a high degree of

confusion between upland and all wetland classes using the RF classifier. Confusion also occurs between herbaceous wetland classes (i.e., bog, fen, and marsh) indicating a high degree of omission for these classes. Other wetland studies reported the great capacity of RF for wetland mapping using a large number of input features such as multi-temporal polarimetric and optical features (Mahdianpari et al., 2017). Thus, it was concluded that the efficiency of the RF classifier for complex land cover mapping greatly depends on the number of input features. A comparison between two confusion matrices demonstrated a significant improvement by applying CNN relative to RF. Specifically, CNN was about 39%, 18%, 18%, 13%, and 63% more accurate than RF to classify bog, fen, swamp, marsh, and shallow-water classes, respectively. CNN also outperformed RF in discriminating non-wetland classes, wherein all classes were correctly classified in more than 95% of cases.

Table 2-2 Confusion matrix of CNN: overall accuracy: 94.82%, kappa coefficient: 0.93

		Reference Data									
	Class	Bog	Fen	Swamp	Marsh	Upland	Urban	Shallow-water	Deep-water	Tot.	User Acc.
Classified Data	Bog	15237	1810	4	11	1320	1183	0	0	19565	77.88
	Fen	256	7094	26	920	436	8	54	0	8794	80.67
	Swamp	203	128	7623	156	978	403	0	0	9491	80.32
	Marsh	125	71	773	4015	168	86	0	0	5238	76.65
	Upland	1259	2187	1259	59	85114	0	0	0	89878	94.70
	Urban	0	21	0	0	931	66173	0	0	67125	98.58
	Shallow-Water	0	0	0	0	0	0	5461	218	5679	96.16
	Deep-water	0	0	0	0	0	0	228	88966	89194	99.74
	Total	17080	11311	9685	5161	88947	67853	5743	89184	294964	
	Prod. Acc.	89.21	62.72	78.71	77.80	95.69	97.52	95.09	99.76		

Table 2-3 Confusion matrix of RF: overall accuracy: 79.11%, kappa coefficient: 0.73

		Reference Data									
	Class	Bog	Fen	Swamp	Marsh	Upland	Urban	Shallow-water	Deep-water	Tot.	User Acc.
Classified Data	Bog	11620	2457	249	691	4337	17	194	0	19565	59.39
	Fen	1950	5907	108	649	180	0	0	0	8794	67.17
	Swamp	749	57	6234	376	1996	79	0	0	9491	65.68
	Marsh	188	192	51	4392	291	17	107	0	5238	83.85
	Upland	7913	3768	3106	39	59981	15071	0	0	89878	66.74
	Urban	775	818	591	97	4916	59928	0	0	67125	89.28
	Shallow-Water	11	8	0	516	0	0	3661	1483	5679	64.47
	Deep-water	0	0	0	24	0	0	7559	81611	89194	91.50
	Total	23206	13207	10339	6784	71701	75112	11521	83094	294964	
	Prod. Acc.	50.07	44.73	60.30	64.74	83.65	79.78	31.78	98.22		

2.4 Conclusions

In this study, the capability of a state-of-the-art classification tool, deep convolutional neural network (CNN), was investigated for wetland classification. In particular, we examined the potential of a pre-existing convolutional neural network, namely AlexNet, for mapping wetland complexes using RapidEye optical imagery in a study area located in the Avalon Peninsula, Newfoundland and Labrador, Canada. The overall classification accuracy obtained by CNN was compared with Random Forest (RF), suggesting the superiority of CNN relative to RF even by incorporating a smaller number of input features. In particular, an overall classification accuracy of 94.82% was achieved using CNN, demonstrating an improvement of about 16% compared to the RF classifier for all land cover types. Moreover, an average improvement of about 30% was attained for wetland classes when CNN was employed. The latter observation suggests the significance of incorporating high-level spatial features into the classification scheme to reduce confusion between spectrally similar wetland classes.

The novel classification framework employed in this study, along with the fine spatial resolution map, obtained by CNN, can be used as baseline information and tool for wetland mapping while significantly facilitating the application of CNN for classification of satellite remote sensing data. Furthermore, the use of other very deep CNNs, such as DenseNet, VGG, Xception, and InceptionResNet for classifying wetland complexes offers a potential avenue for further research. This will be particularly feasible by employing advanced cloud processing tools to accelerate the feature learning process.

References

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. ISPRS Journal of Photogrammetry and Remote Sensing, 114, 24–31.

Bengio, Y. (2009). Learning deep architectures for AI. Foundations and Trends® in Machine Learning, 2(1), 1–127.

Brisco, B., Ahern, F., Murnaghan, K., White, L., Canisus, F., & Lancaster, P. (2017). Seasonal Change in Wetland Coherence as an Aid to Wetland Monitoring. Remote Sensing, 9(2), 158–176.

Brisco, B., Kapfer, M., Hirose, T., Tedford, B., & Liu, J. (2011). Evaluation of C-band polarization diversity and polarimetry for wetland mapping. Canadian Journal of Remote Sensing, 37(1), 82–92.

Chen, X., Xiang, S., Liu, C.-L., & Pan, C.-H. (2014). Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geoscience and Remote Sensing Letters, 11(10), 1797–1801.

Chen, Y., Lin, Z., Zhao, X., Wang, G., & Gu, Y. (2014). Deep learning-based classification of hyperspectral data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7(6), 2094–2107.

DiCarlo, J. J., Zoccolan, D., & Rust, N. C. (2012). How does the brain solve visual object recognition? Neuron, 73(3), 415–434.

Ecological Stratification Working Group (ESWG). (1995). A National Ecological Framework for Canada. https://doi.org/Cat. No. A42-65/1996E; ISBN 0-662-24107-X

Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random forests for land cover classification. Pattern Recognition Letters, 27(4), 294–300.

Jackson, Q., & Landgrebe, D. A. (2002). Adaptive Bayesian contextual classification based on Markov random fields. IEEE Transactions on Geoscience and Remote Sensing, 40(11), 2454–2463.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014). Caffe: Convolutional Architecture for Fast Feature Embedding. ArXiv Preprint ArXiv:1408.5093.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25 (pp. 1097–1105). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Landgrebe, D. A. (2005). Signal theory methods in multispectral remote sensing (Vol. 29). John Wiley & Sons.

Längkvist, M., Kiselev, A., Alirezaie, M., & Loutfi, A. (2016). Classification and segmentation of satellite orthoimagery using convolutional neural networks. Remote Sensing, 8(4), 329–349.

Le, Q. V, Coates, A., Prochnow, B., & Ng, A. Y. (2011). On Optimization Methods for Deep Learning. Proceedings of The 28th International Conference on Machine Learning (ICML), 265–272.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. Retrieved from http://dx.doi.org/10.1038/nature14539

Lee, C.-Y., Gallagher, P. W., & Tu, Z. (2016). Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In Artificial Intelligence and Statistics (pp. 464–472).

Loosvelt, L., Peters, J., Skriver, H., De Baets, B., & Verhoest, N. E. C. (2012). Impact of reducing polarimetric SAR input on the uncertainty of crop classifications based on the random forests algorithm. IEEE Transactions on Geoscience and Remote Sensing, 50(10), 4185–4200.

Mahdianpari, M., Salehi, B., Mohammadimanesh, F., & Brisco, B. (2017). An Assessment of Simulated Compact Polarimetric SAR Data for Wetland Classification Using Random Forest Algorithm. Canadian Journal of Remote Sensing, 43(5). https://doi.org/10.1080/07038992.2017.1381550

Mahdianpari, M., Salehi, B., Mohammadimanesh, F., Brisco, B., Mahdavi, S., Amani, M., & Granger, J. E. (2018). Fisher Linear Discriminant Analysis of coherency matrix for wetland classification using PolSAR imagery. Remote Sensing of Environment, 206, 300–317.

Mahdianpari, M., Salehi, B., Mohammadimanesh, F., & Motagh, M. (2017). Random forest wetland classification using ALOS-2 L-band, RADARSAT-2 C-band, and

TerraSAR-X imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 130, 13–31.

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

Marshall, I. B., & Schut, P. (1999). A national ecological framework for Canada. Eastern Cereal and Oilseed Research Centre (ECORC), Research Branch, Agriculture and Agri-Food Canada.

Mnih, V. (2013). Machine Learning for Aerial Image Labeling, 109.

Mnih, V., Heess, N., & Graves, A. (2014). Recurrent models of visual attention. In Advances in neural information processing systems (pp. 2204–2212).

Mohammadimanesh, F., Salehi, B., Mahdianpari, M., Brisco, B., & Motagh, M. (2018). Multi-temporal, multi-frequency, and multi-polarization coherence and SAR backscatter analysis of wetlands. ISPRS Journal of Photogrammetry and Remote Sensing, 142, 78–93. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2018.05.009

Nogueira, K., Penatti, O. A. B., & dos Santos, J. A. (2017). Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification. Pattern Recognition, 61, 539–556.

Scholkopf, B., & Smola, A. J. (2001). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, MA, USA: MIT Press.

Tiner, R. W., Lang, M. W., & Klemas, V. V. (2015). Remote sensing of wetlands: applications and advances. CRC Press.

Wdowinski, S., Kim, S.-W., Amelung, F., Dixon, T. H., Miralles-Wilhelm, F., & Sonenshein, R. (2008). Space-based detection of wetlands' surface water level changes from L-band SAR interferometry. Remote Sensing of Environment, 112(3), 681–696.

Zhao, W., & Du, S. (2016). Learning multiscale and deep representations for classifying remotely sensed imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 113, 155–165.

Zhong, P., & Wang, R. (2010). Learning conditional random fields for classification of hyperspectral images. IEEE Transactions on Image Processing, 19(7), 1890–1907.

Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. IEEE Geoscience and Remote Sensing Magazine. https://doi.org/10.1109/MGRS.2017.2762307

3 Chapter 3: Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery

Abstract

Despite recent advances in deep Convolutional Neural Networks in various computer vision tasks, their potential for classification of multispectral remote sensing images has not been thoroughly explored. In particular, the applications of deep CNNs using optical remote sensing data have focused on the classification of very high-resolution aerial and satellite data, owing to the similarity of these data to the large datasets in computer vision. Accordingly, this study presents a detailed investigation of state-of-the-art machine learning tools for classification of complex wetland classes using RapidEye multispectral imagery. Specifically, we examine the capacity of seven well-known deep convnets, namely DenseNet121, InceptionV3, VGG16, VGG19, Xception, ResNet50, and InceptionResNetV2 for wetland mapping in Canada. In addition, the classification results obtained from deep CNNs are compared with those based on conventional machine learning tools, including Random Forest and Support Vector Machine, to further evaluate the efficiency of the former to classify wetlands. The results illustrate that the full-training of convnets using five spectral bands outperforms the other strategies for all convnets. InceptionResNetV2, ResNet50, and Xception are distinguished as the top three convnets providing state-of-the-art classification accuracies of 96.17%, 94.81%, and 93.57%, respectively. The classification accuracies obtained using SVM and RF are 74.89% and 76.08%, respectively, considerably inferior relative to CNNs. Importantly, InceptionResNetV2 is consistently found to be superior compared to all other convnets,

suggesting the integration of Inception and ResNet modules is an efficient architecture for classifying complex remote sensing scenes such as wetlands.

3.1 Introduction

Wetlands are transitional zones between terrestrial and aquatic systems that support a natural ecosystem of a variety of plant and animal species, adapted to wet conditions (Tiner, Lang, & Klemas, 2015). Flood- and storm-damage protection, water quality improvement and renovation, greenhouse gas reduction, shoreline stabilization, and aquatic productivity are only a handful of the advantages associated with wetlands. Unfortunately, wetlands have undergone changes due to natural processes, such as changes in temperature and precipitation caused by climate change, coastal plain subsidence and erosion, as well as human-induced disturbances such as industrial and residential development, agricultural activities, and runoff from lawns and farms (Tiner et al., 2015).

Knowledge of the spatial distribution of these valuable ecosystems is crucial in order to characterize ecosystem processes and to monitor the subsequent changes over time (Mahdianpari, Salehi, Mohammadimanesh, & Motagh, 2017). However, the remoteness, vastness, seasonally dynamic nature, and inaccessibility of most wetland ecosystems make conventional methods of data acquisition (e.g., surveying) labor-intensive and costly (Evans & Costa, 2013). Fortunately, remote sensing, as a cost- and time-efficient tool, addresses the limitations of conventional techniques by providing valuable ecological data to characterize wetland ecosystems and to monitor land cover changes (Mahdianpari et al., 2018). Optical remote sensing data have shown to be promising tools

for wetland mapping and monitoring. This is because biomass concentration, leaf water content, and vegetation chlorophyll —all important characteristics of wetland vegetation— can be determined using optical satellite images (Adam, Mutanga, & Rugege, 2010). In particular, optical remote sensing sensors collect spectral information of ground targets at various points of the electromagnetic spectrum, such as visible and infrared, which is of great benefit for wetland vegetation mapping (Adam et al., 2010). Therefore, several studies reported the success of wetland mapping using optical satellite imagery (Friedl & Brodley, 1997; Frohn, Autrey, Lane, & Reif, 2011; Hestir et al., 2008; Mahdianpari, Salehi, Mohammadimanesh, & Brisco, 2017).

Despite the latest advances in remote sensing tools, such as the availability of high spatial and temporal resolution satellite data and object-based image analysis tools (Blaschke, 2010), the classification accuracy of complex land cover, such as wetland ecosystems, is insufficient (Mahdianpari et al., 2018). This could be attributed to the spectral similarity of wetland vegetation types, making the exclusive use of spectral information insufficient for the classification of heterogeneous land cover. In addition, several studies reported the significance of incorporating both spectral and spatial information for land cover mapping (Tiner et al., 2015; Zhao & Du, 2016). Thus, spatial features may augment spectral information and thereby contribute to the success of complex land cover mapping. Accordingly, several experiments were carried out to incorporate both spectral and spatial features into a classification scheme. These studies were based on the Markov Random Field (MRF) model (Jackson & Landgrebe, 2002), the Conditional Random Field (CRF) model (Zhong & Wang, 2010), and Composite Kernel (CK) methods (Zhong & Wang, 2010). However, in most cases, the process of extracting a large number of

features, called the feature engineering process (Chollet, 2017), for the purpose of supervised classification is time intensive, and requires broad and profound knowledge to extract amenable features (LeCun, Bengio, & Hinton, 2015). Furthermore, classification based on hand-crafted spatial features primarily relies on low-level features, resulting in inadequate classification results in most cases and a poor capacity for generalization (Zhao & Du, 2016).

Most recently, Deep Learning (DL), a state-of-the-art machine learning tool, has been placed in the spotlight in the field of computer vision and, subsequently, in remote sensing (LeCun et al., 2015). This is because these advanced machine learning algorithms address the primary limitations of the conventional shallow-structured machine learning tools, such as Support Vector Machine (SVM) and Random Forest (RF; Ball, Anderson, & Chan, 2017). Deep Belief Net (DBN; Hinton, Osindero, & Teh, 2006), Stacked Auto-Encoder (SAE; Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010), and deep Convolutional Neural Network (CNN; Evans & Costa, 2013; Krizhevsky, Sutskever, & Hinton, 2012; Mahdianpari et al., 2017; Szegedy et al., 2014) are current deep learning models, of which the latter is most well-known (Patterson & Gibson, 2017). Importantly, CNN has led to a series of breakthroughs in several remote sensing applications, such as classification (Zhong & Wang, 2010), segmentation (Zhong & Wang, 2010), and object detection (X. Chen, Xiang, Liu, & Pan, 2014), due to its superior performance in a variety of applications relative to shallow-structured machine learning tools. CNNs are characterized by multi-layered interconnected channels, with a high capacity for learning the features and classifiers from data spontaneously given their deep architecture, their capacity to adjust parameters jointly, and this ability to classify simultaneously

(Nogueira, Penatti, & dos Santos, 2017). One of the ubiquitous characteristics of such a configuration is its potential to encode both spectral and spatial information into the classification scheme in a completely automated workflow (Nogueira et al., 2017). Accordingly, the complicated, brittle, and multistage feature engineering procedure is replaced with a simple end-to-end deep learning workflow (Chollet, 2017).

Notably, there is a different degree of abstraction for the data within multiple convolutional layers, wherein low-, mid-, and high-level information is extracted in a hierarchical learning framework at the initial, intermediate, and final layers, respectively (Nogueira et al., 2017). This configuration omits the training process from scratch in several applications since the features in the initial layers are generic filters (e.g., edge) and, accordingly, are less dependent on the application. However, the latest layers are related to the final application and should be trained according to the given data and classification problem. This also addresses the poor generalization capacity of shallow-structured machine learning tools, which are site- and data-dependent, suggesting the versatility of CNNs (Chollet, 2017).

Although the advent of CNN dates back to as early as the 1980s, when LeCun designed a primary convolutional neural network known as LeNet to classify handwritten digits, it gained recognition and was increasingly applied around 2010 (LeCun, Bottou, Bengio, & Haffner, 1998). This is attributable to the advent of more powerful hardware, larger datasets (e.g., ImageNet; Deng et al., 2009), and new ideas, which consequently improved network architecture (Szegedy et al., 2014). The original idea of deep CNNs (LeCun et al., 1998) has been further developed by Krizhevsky and his colleagues, who designed a breakthrough CNN, known as AlexNet, a pioneer of modern deep CNNs with

multiple convolutional and max-pooling layers, which provides deeper feature-learning at different spatial scales (Krizhevsky et al., 2012a). Subsequent successes have been achieved since 2014, when VGG (Simonyan & Zisserman, 2014), GoogLeNet (i.e., Inception network; Szegedy et al., 2014), ResNet (He, Zhang, Ren, & Sun, 2016), and Xception (François Chollet, 2016) were introduced in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC).

The intricate tuning process, heavy computational burden, high tendency of overfitting, and the empirical nature of model establishment are the main limitations associated with deep CNNs (Nogueira et al., 2017). Although some research has argued that all deep learning methods have a black-box nature, it is not completely true for CNN (Chollet, 2017). This is because the features learned by CNNs can be visualized, thereby providing illustrations of concepts. There are three different strategies for employing current CNNs: a full-training network, a pre-trained network as a feature extractor, and fine-tuning of a pre-trained network. In the first case, a network is trained from scratch with random weights and biases to extract particular features for the dataset of interest. However, the limited amount of training samples constrains the efficiency of this technique due to the overfitting problem. The other two strategies are more useful when a limited amount of training samples is available (Hu, Xia, Hu, & Zhang, 2015; Nogueira et al., 2017).

In the case of limited amount of training data, a stacked autoencoder (SAE) is also useful to learn the features from a given dataset using an unsupervised learning network (G. Hinton & Salakhutdinov, 2006). In such a network, the deconstruction error between the input data at the encoding layer and its reconstruction at the decoding layer is minimized (Zhang, Zhang, & Kumar, 2016). SAE networks are characterized by a relatively simple

structure relative to deep CNNs and they have a great capacity for fast image interpretation. In particular, they convert raw data into more abstract representation using a simple non-linear model and integrate features using optimization algorithm. This results in a substantial decrease of the redundant information between the features while achieving a strong generalization capacity (Kang, Ji, Leng, Xing, & Zou, 2017).

Despite recent advances in deep CNNs, their applications in remote sensing have been substantially limited to the classification of very high spatial resolution aerial and satellite imagery from a limited number of well-known datasets, owing to the similar characteristics of these data to those used in object recognition in computer vision. However, acquiring high spatial resolution imagery may be difficult, especially on a large scale. Accordingly, less research has been carried out on the classification of medium and high spatial resolution satellite imagery in different study areas. Furthermore, the capacity of CNNs has been primarily investigated for the classification of urban areas, whereas there is limited research examining the potential of state-of-the-art classification tools for complex land cover mapping. Complex land cover units, such as wetland vegetation, are characterized by high intra- and low inter-class variance, resulting in difficulties in their discrimination relative to typical land cover classes. Thus, an environment with such highly heterogeneous land cover is beneficial for evaluating the capacity of CNNs for the classification of remote sensing data. Finally, the minimal application of well-known deep CNNs in remote sensing may be due to the limitation of input bands. Specifically, these convnets are designed to work with three input bands (e.g., Red, Green, and Blue), making them inappropriate for most remote sensing data.

This indicates the significance of developing a pipeline compatible with multi-channel satellite imagery.

The main goals of this study were, therefore, to: (1) eliminate the limitation of the number of input bands by developing a pipeline in Python with the capacity to operate with multi-layer remote sensing imagery; (2) examine the power of deep CNNs for the classification of spectrally similar wetland classes; (3) investigate the generalization capacity of existing CNNs for the classification of multispectral satellite imagery (i.e., a different dataset than those they were trained for); (4) explore whether full-training or fine-tuning is the optimal strategy for exploiting the pre-existing convnets for wetland mapping; and (5) compare the efficiency of the most well-known deep CNNs, including DenseNet121, InceptionV3, VGG16, VGG19, Xception, ResNet50, and InceptionResNetV2 for wetland mapping in a comprehensive and elaborate analysis. Thus, this study contributes to the use of the state-of-the-art classification tools for complex land cover mapping using multispectral remote sensing data.

3.2 Materials and Methods

3.2.1 Deep Convolutional Neural Network

CNNs are constructed by multi-layer interconnected neural networks, wherein powerful low-, intermediate-, and high-level features are hierarchically extracted. A typical CNN framework has two main layers –the convolutional and pooling layers– that, together, are called the convolutional base of the network (Chollet, 2017). Some networks, such as AlexNet and VGG, also have fully connected layers. The convolutional layer has a filtering function and extracts spatial features from the images. Generally, the first

convolutional layers extract low-level features or small local patterns, such as edges and corners, while the last convolutional layers extract high-level features, such as image structures. This suggests the high efficiency of CNNs for learning spatial hierarchical patterns. Convolutional layers are usually defined using two components: the convolution patch size (e.g., 3x3 or 5x5) and the depth of the output feature map, which is the number of filters (e.g., 32 filters). In particular, a rectangular sliding window with a fixed-size and a pre-defined stride is employed to produce convoluted feature maps using a dot product between the weights of the kernel and a small region of the input volume (i.e., the receptive field). A stride is defined as a distance between two consecutive convolutional windows. A stride of one is usually applied in convolutional layers since larger stride values result in down-sampling in feature maps (Chollet, 2017). A feature map is a new image generated by this simple convolution operation and is a visual representation of the extracted features. Given the weight-sharing property of CNNs, the number of parameters is significantly reduced compared to those of a fully connected layer, since all the neurons across a particular feature map share the same parameters (i.e., weights and biases).

A non-linearity function, such as the Rectified Linear Unit (ReLU; Krizhevsky, Sutskever, & Hinton, 2012b), is usually applied as an elementwise nonlinear activation function to each component in the feature map. The ReLU function is advantageous relative to conventional activation functions used in traditional neural networks, such as the tanh or sigmoid functions, for adding non-linearity to the network (Krizhevsky, Sutskever, & Hinton, 2012b). The ReLU significantly accelerates the training phase relative to the conventional functions with gradient descent. This is because of the so-

called vanishing gradient problem, wherein the derivatives of earlier functions (e.g., sigmoid) are extremely low in the saturating region and, accordingly, the updates for the weights nearly vanish.

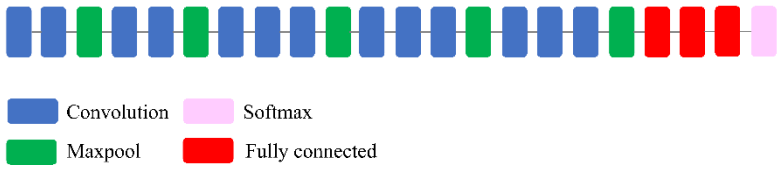
Due to the presence of common pixels in each window, several feature maps may be produced that are very similar, suggesting redundant information. Therefore, pooling layers are used after each convolutional layer to decrease the variance of the extracted features using simple operations such as the maximizing or averaging operations. The max- and average-pooling layer determine the maximum or mean values, respectively, using a fixed-size sliding window and a pre-defined stride over the feature maps and, therefore, are conceptually similar to the convolutional layer. In contrast to convolutional layers, a stride of two or larger is applied in the pooling layers to down-sample the feature maps. Notably, the pooling layer, or the sub-sampling layer, generalizes the output of the convolutional layer into a higher level and selects the more robust and abstract features for the next layers. Thus, the pooling layer decreases computational complexity during the training stage by shrinking the feature maps.

As mentioned, some networks may have fully connected layers before the classifier layer that connects the output of several stacked convolutional and pooling layers to the classifier layer. Overfitting may arise in the fully connected layer because it occupies a large number of parameters. Thus, the dropout technique, an efficient regularization technique, is useful to mitigate or decrease problems associated with overfitting. During training, this technique randomly drops some neurons and their connections across the network, which prevents neurons from excess co-adaptation and contributes to developing more meaningful independent features (Krizhevsky et al., 2012a). The last

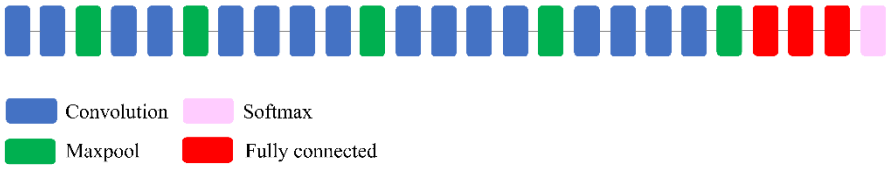
layer is a classification layer, which determines the posterior probabilities for each category. A Softmax classifier, also known as a normalized exponential, is the most commonly used classifier layer among the deep learning community in the image field. Stochastic Gradient Descent (SGD) optimization in a backpropagation workflow is usually used to train CNNs and to compute adjusting weights. This is an end-to-end learning process, from the raw data (i.e., original pixels) to the final label, using a deep CNN.

3.2.1.1 VGG

VGG network (Simonyan & Zisserman, 2014), the runner-up of the localization and classification tracks of the ILSVRC-2014 competition, is characterized by a deep network structure with a small convolutional filter of 3x3 compared to its predecessor, AlexNet (Krizhevsky et al., 2012a). VGG-VD group introduced six deep CNNs in the competition, among which two were more successful than the others, namely VGG16 and VGG19. The VGG16 consists of 13 convolutional layers and three fully connected layers, while the VGG19 has 16 convolutional layers and three fully connected layers. Both networks use a stack of small convolutional filters of 3x3 with stride 1, which are followed by multiple non-linearity layers (see Figure 3.1). This increases the depth of the network and contributes to learning more complex features. The impressive results of VGG revealed that the network depth is an important factor in obtaining high classification accuracy (Hu et al., 2015).



(**a**)



(**b**)

Figure 3-1 Schematic diagram of (**a**) VGG16 and (**b**) VGG19 models.

3.2.1.2 Inception

GoogLeNet, the winner of the classification and detection tracks of the ILSVRC-2014 competition, is among the first generation of non-sequential CNNs. In this network, both depth (i.e., the number of levels) and width (i.e., the number of units at each level), were increased without causing computational strain (Szegedy et al., 2014). GoogLeNet is developed based on the idea that several connections between layers are ineffective and have redundant information due to the correlation between them. Accordingly, it uses an "Inception module", a sparse CNN, with 22 layers in a parallel processing workflow and benefits from several auxiliary classifiers within the intermediate layers to improve the discrimination capacity in the lower layers. In contrast to conventional CNNs such as AlexNet and VGG, wherein either a convolutional or a pooling operation can be used at each level, the Inception module could benefit from both at each layer. Furthermore,

filters (convolutions) with varying sizes are used at the same layer, providing more detailed information and extracting patterns with different sizes. Importantly, a 1x1 convolutional layer, the so-called bottleneck layer, was employed to decrease both the computational complexity and the number of parameters. To be more precise, 1x1 convolutional layers were used just before a larger kernel convolutional filter (e.g., 3x3 and 5x5 convolutional layers) to decrease the number of parameters to be determined at each level (i.e., the pooling feature process). In addition, 1x1 convolutional layers make the network deeper and add more non-linearity by using ReLU after each 1x1 convolutional layer. In this network, the fully connected layers are replaced with an average pooling layer. This significantly decreases the number of parameters since the fully connected layers include a large number of parameters. Thus, this network is able to learn deeper representations of features with fewer parameters relative to AlexNet while it is much faster than VGG (François Chollet, 2016). Figure 3.2 illustrates a compressed view of InceptionV3 employed in this study.

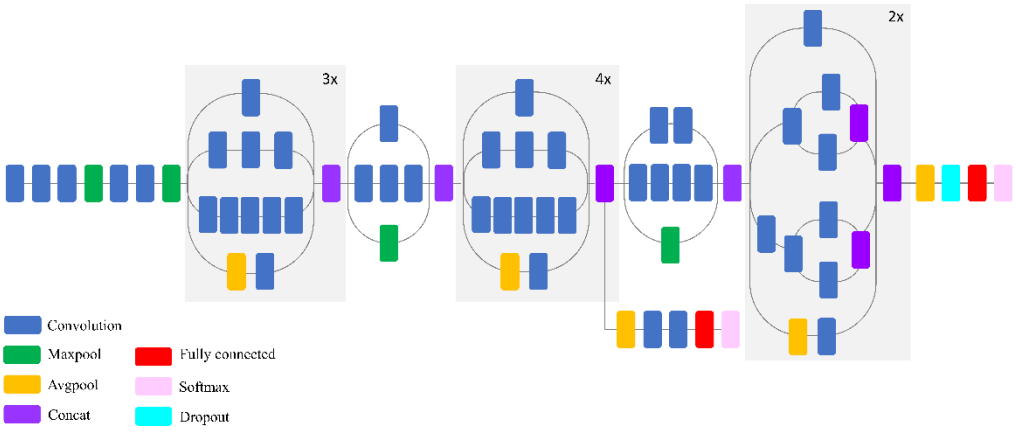


Figure 3-2 Schematic diagram of InceptionV3 model (compressed view).

3.2.1.3 ResNet

ResNet, the winner of the classification task in the ILSVRC-2015 competition, is characterized by a very deep network with 152 layers (He et al., 2016). However, the main problems associated with the deep network are difficulty in training, high training error, and the vanishing gradient that causes learning to be negligible at the initial layers in the backpropagation step. The deep ResNet configuration addresses the vanishing gradient problem by employing a deep residual learning module via additive identity transformations. Specifically, the residual module uses a direct path between the input and output and each stacked layer fits a residual mapping rather than directly fitting a desired underlying mapping (He et al., 2016). Notably, the optimization is much easier on the residual map relative to the original, unreferenced map. Similar to VGG, 3x3 filters were mostly employed in this network; however, ResNet has fewer filters and less complexity relative to the VGG network (He et al., 2016; Mahdianpari et al., 2018). Figure 3.3 illustrates a compressed view of ResNet, which used in this study.

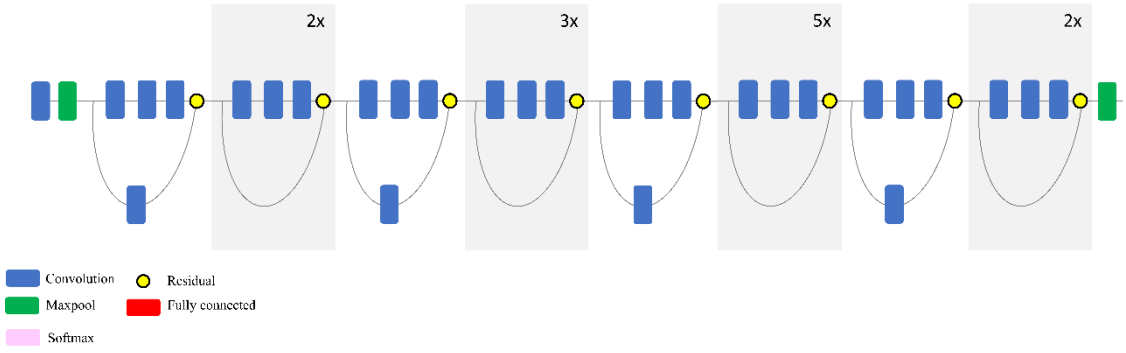


Figure 3-3 Schematic diagram of ResNet model (compressed view).

3.2.1.4 Xception

Xception network is similar to inception (GoogLeNet), wherein the inception module has been substituted with depth-wise separable convolutional layers (François Chollet, 2016). Specifically, Xception's architecture is constructed based on a linear stack of a depth-wise separable convolution layer (i.e., 36 convolutional layers) with linear residual connections (see Figure 3.4). There are two important convolutional layers in this configuration: a depth-wise convolutional layer (Sifre & Mallat, 2013), where a spatial convolution is carried out independently in each channel of input data, and a pointwise convolutional layer, where a 1x1 convolutional layer maps the output channels to a new channel space using a depth-wise convolution.

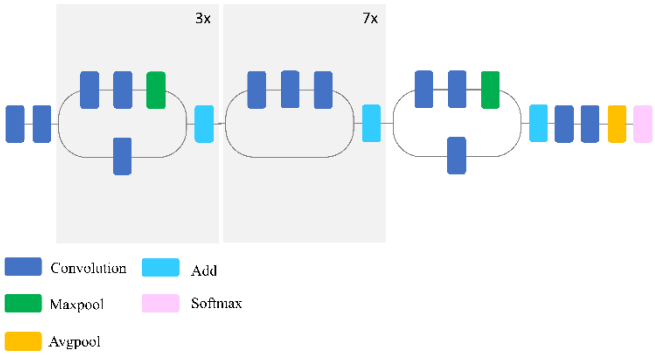


Figure 3-4 Schematic diagram of Xception model (compressed view).

3.2.1.5 InceptionResNetV2

This network is constructed by integrating the two most successful deep CNNs, ResNet (He et al., 2016) and Inception (Szegedy et al., 2014), wherein batch-normalization is used only on top of the traditional layers rather than on top of the summations. In particular, the residual modules are employed in order to allow an increase in the number

of Inception blocks and, accordingly, an increase of network depth. As mentioned earlier, the most pronounced problem associated with very deep networks is the training phase, which can be addressed using the residual connections (He et al., 2016). The network scales down the residual as an efficient approach to address the training problem when a large number of filters (greater than 1000 filters) is used in the network. Specifically, the residual variants experience instabilities and the network cannot be trained when the number of filters exceeds 1000. Therefore, scaling the residual contributes to stabilizing network training (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017). Figure 3.5 illustrates a compressed view of InceptionResNetV2 used in this study.

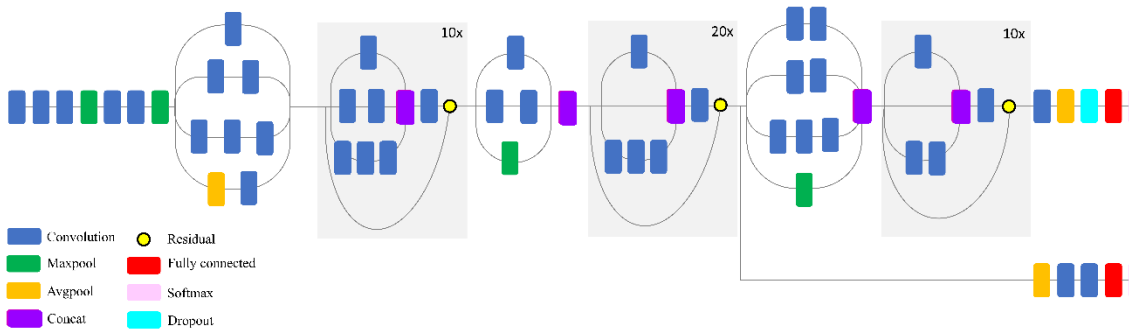


Figure 3-5 Schematic diagram of InceptionResNetV2model (compressed view).

3.2.1.6 DenseNet

This network is also designed to address the vanishing gradient problem arising from the network depth. Specifically, all layers' connection architectures are employed to ensure maximum flow of information between layers (Huang, Liu, Van Der Maaten, & Weinberger, 2017). In this configuration, each layer acquires inputs from all previous layers and conveys its own feature-maps to all subsequent layers. The feature maps are

concatenated at each layer to pass information from preceding layers to the subsequent layers. This network architecture removes the necessity to learn redundant information and accordingly, the number of parameters is significantly reduced (i.e., parameter efficiency). It is also efficient for preserving information owing to its all layers connection property. Huang et al. reported that the network performed very well for classifications with a small training data set and the overfitting is not a problem when DenseNet121 is employed (Huang et al., 2017). Figure 3.6 illustrates a compressed view of DenseNet employed in this study.

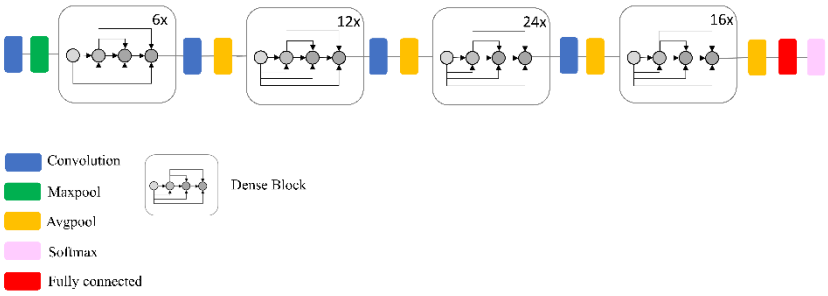


Figure 3-6 Schematic diagram of DenseNet model (compressed view).

3.2.2 Training

3.2.2.1 Fine-tuning of a pre-trained network

Fine-tuning of a pre-trained network is an optimal solution when a limited number of training samples are available. In this case, a fine adjustment is performed on the parameters of the top layers in a pre-trained network, while the first layers, representing general features, are frozen. Freezing is defined when weights for a layer or a set of layers are not updated during the training stage. Importantly, this approach benefits from

the parameters learned from a network that has been previously trained using a specific dataset and, subsequently, adjusts the parameters for the dataset of interest. Accordingly, fine-tuning adjusts the parameters of the reused model, making it more relevant to the dataset of interest. Fine-tuning can be performed for either all layers or the top layers of a pre-trained network; however, the latter approach is preferred (Chollet, 2017). This is because the first layers in convnets encode generic, reusable features, whereas the last layers encode more specific features. Thus, it is more efficient to fine-tune those specific features. Furthermore, fine-tuning of all layers causes overfitting due to the large number of parameters, which should be determined during such a processing (Chollet, 2017). As such, in this study, fine-tuning of pre-existing convnets was carried out only on the top three layers. These may be either only fully connected layers (e.g., VGG) or both fully connected and convolutional layers (e.g., Xception). Accordingly, the fine-tuning of the top three layers allowed us to compare the efficiency of fine-tuning for both fully connected and convolutional layers.

Notably, the number of input bands for these CNNs is limited to three because they have been trained using the ImageNet dataset; however, RapidEye imagery has five bands. Therefore, a band selection technique was pursued to determine three uncorrelated bands of RapidEye imagery most appropriate for use in CNNs. The results of this analysis demonstrated that green, red, and near-infrared bands contain the least redundant information and thus, they were selected for fine-tuning of CNNs in this study.

3.2.2.2 Full-training

Full-training is feasible when a large number of training samples is available to aid in converging the network (Nogueira et al., 2017). In this case, there is a full control on the network parameters and, additionally, more relevant features are produced since the network is specifically tuned with the dataset of interest. However, the full-training of a network from scratch is challenging due to computational and data strains, leading to overfitting problems. Some techniques, such as dropout layers and data augmentation and normalization, are useful for mitigating the problems that arise from overfitting. In particular, data augmentation, introduced by Krizhevsky in 2012, is a process that produces more training samples from existing training data using a number of random transformations (e.g., image translation and horizontal reflection; Krizhevsky et al., 2012a). The main goal is that the model will never look at the same image twice. In particular, the model explores more aspects of the data, which contributes to a better generalization (Chollet, 2017).

Notably, there are two different categories in the case of full-training of convnets. In the first category, a new CNN architecture is fully designed and trained from scratch. In this case, the number of convolutional, and pooling layers, neurons, the type of activation function, the learning rate, and the number of iterations should be determined. Conversely, the second strategy benefits from a pre-existing architecture and full-training is only employed using a given dataset. In the latter case, the network architecture and the number of parameters remain unchanged.

In this study, the second strategy was employed. In particular, we examined the potential of a number of pre-existing networks (e.g., VGG, Inception, and etc.,) for classification

of complex land cover when they are trained from scratch using a new dataset substantially different from those (e.g., ImageNet) for which it was originally trained. Notably, full-training was employed for both three and five bands of RapidEye imagery. The full-training of three bands was performed to make the results comparable with those of the fine-tuning strategy.

3.2.3 Study area and satellite data

The study area is located in the northeast portion of the Avalon Peninsula, Newfoundland and Labrador, Canada. Figure 3.7 shows the geographic location of the study area. Land cover in the study area comprises a wide variety of wetland classes categorized by the Canadian Wetland Classification System (CWCS), including bog, fen, marsh, swamp, and shallow water (Tiner et al., 2015). Wetlands are characterized as complex species with high intra-class variance and low inter-class variance. Additionally, these classes are extremely different from typical objects found in the ImageNet dataset. Such a diverse ecological ecosystem is an ideal setting in which the efficiency and robustness of the state-of-the-art classification algorithms in a comprehensive and comparative study may be examined. Other land-cover classes found in the study area include urban, upland, and deep water classes.

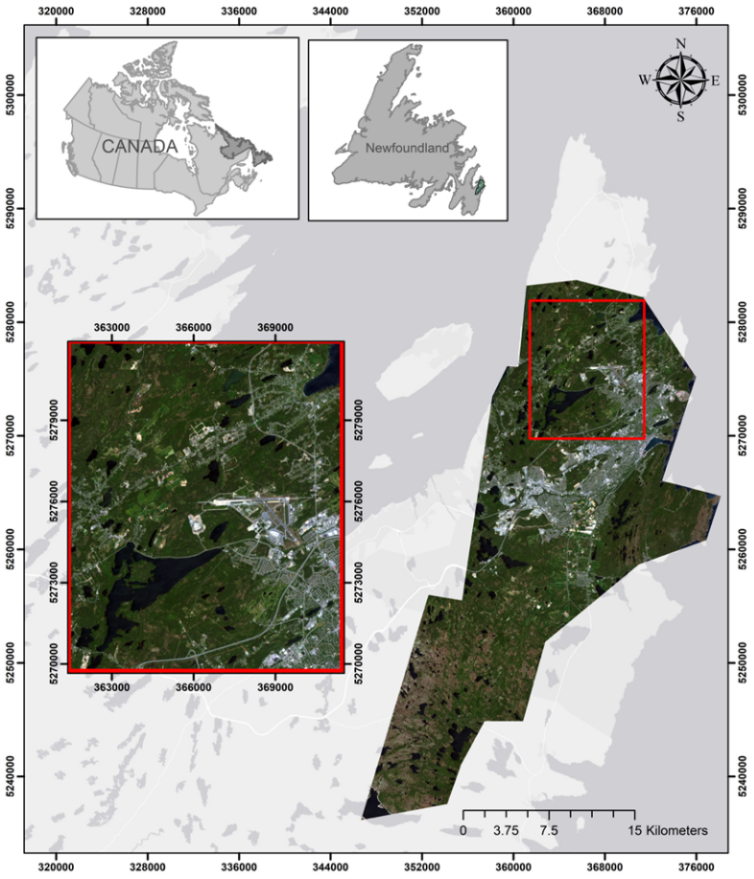
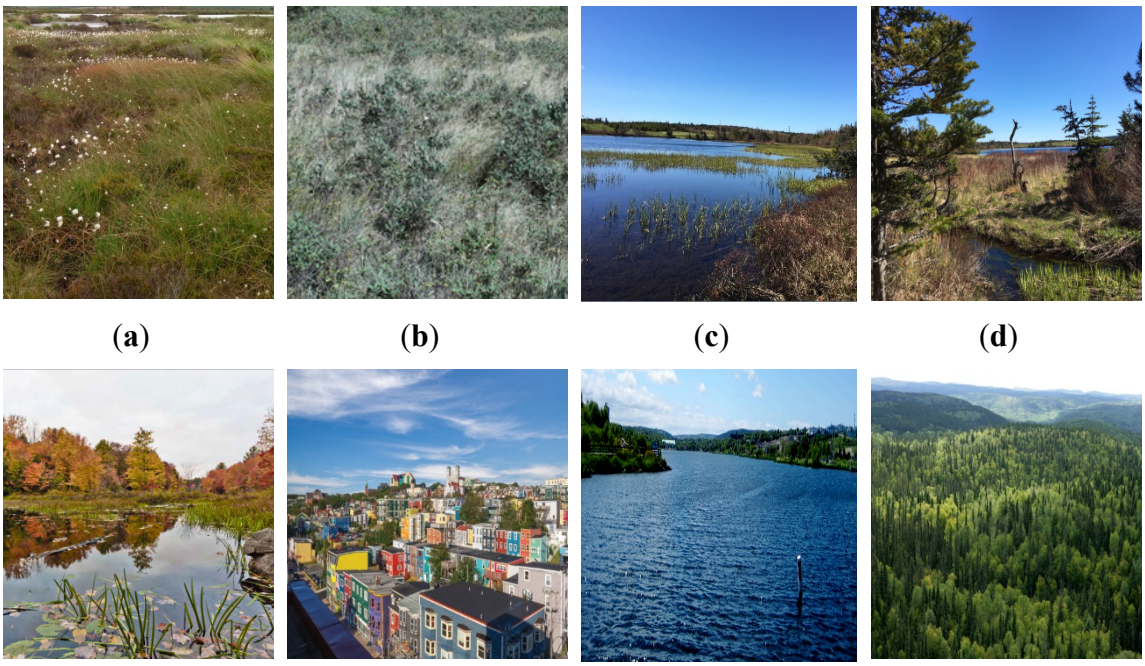


Figure 3-7 A true colour composite of RapidEye optical imagery (bands 3, 2, and 1) acquired on June 18, 2015, illustrating the geographic location of the study area. The red rectangle, the so-called test-zone, was selected to display the classified maps obtained from different approaches. Note that the training samples within the rectangle were excluded during the training stage for deep CNNs.

Figure 3.8 illustrates ground photo examples of land cover classes in this study. The characteristics of all land cover classes in this test site are presented in Table 3.1.



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

Figure 3-8 Ground reference photos showing land cover classes in the study area: (**a**) bog, (**b**) fen, (**c**) marsh, (**d**) swamp, (**e**) shallow water, (**f**) urban, (**g**) deep water, and (**h**) upland.

Two level 3A multispectral RapidEye images with a spatial resolution of five meters, acquired on June 18 and October 22, 2015, were used for classification in this study. This imagery has five spectral bands, namely blue (440 – 510 nm), green (520 – 590 nm), red (630 – 685 nm), red edge (690 – 730 nm), and near-infrared (760 – 850 nm).

3.2.4 Training, validation, and testing data

Field data were acquired for 257 ground sites in the summer and fall of 2015, 2016, and 2017 by collecting Global Positioning System (GPS) points at each site. For reference data preparation, reference polygons were sorted by size and alternately assigned to testing and training groups. This resulted in both the testing and training containing equal

numbers of small and large wetlands polygons to allow for similar pixel counts and to account for the high variation of intra-wetland size.

Importantly, five tiles of RapidEye optical images were mosaicked to cover the whole study region (see Figure 3.7). The training polygons within the red rectangle (i.e., one tile of the RapidEye optical imagery), the so-called test-zone, were removed for training of deep CNNs. In particular, all patches within the test-zone were only used for testing (i.e., accuracy assessment) of CNNs. Of the training sample data, 80% and 20% were used for training and validation, respectively. Notably, both the training and validation were carried out using the first RapidEye image (June 18, 2015); however, the testing was applied only to the second RapidEye image (October 22, 2015), within the test-zone (see Figure 3.7, the red rectangle), to perform the robust classification accuracy assessment.

Accordingly, the training and testing samples were obtained from independent polygons from distinct geographic regions using satellite imagery acquired at different times. This procedure prevents information leak from the testing dataset to the model by employing two spatially and geographically independent samples for training and testing.

3.2.5 Experiment setup

A multispectral satellite image in three dimensions is represented as m×n×h, a 3D tensor, where m and n indicate the height and width of the image, respectively, and h corresponds to the number of channels. On the other hand, convnets require a 3D tensor as input and, accordingly, a patch-based labeling method was used in this study to be aligned with the inherent of CNNs. Using this approach, the multispectral image was decomposed into patches, which have both spectral and spatial information for a given

pixel, and a class label is assigned to the center of each patch (Makantasis, Karantzalos, Doulamis, & Doulamis, 2015).

An optimal patch size was determined using a trial-and-error procedure by taking into account the spatial resolution of 5m for the input image and the contextual relationship of the objects (Huang et al., 2017). In particular, different patch sizes of 5, 10, 15, 20, 25, 30, 35, and 40 were examined and the patch size of 30 was found to be an optimal value that extracts spatial local correlation within a given neighborhood and contains sufficient information to generate a specific distribution for each object in the image. Thus, we obtained 3D tensors with dimensions of either 30x30x5 (in the case of using 5 multispectral bands) or 30x30x3 (in the case of using 3 multispectral bands), which have both spatial and spectral information at a given location.

In the patch-based CNN, a particular class label is assigned to the given patch when a small rectangle in the center of that patch completely covers a single object (Mnih, 2013). In this study, the training polygons were not rectangular, causing challenges during labeling when a patch contained more than one class. Within a given patch size of 30x30 in this study, if an 8x8 rectangular covered only a single class (e.g., bog) the label of this patch was assigned to that class (bog). Conversely, when this small rectangular window (i.e., 8x8) covered more than one class (e.g., both bog and fen), this patch was removed and excluded from further processing. Thus, the selected patches for the training of convnets cover more than 50% of the object of interest and overcame the problem of edges that arise from multiple objects within a single patch.

The convnets used in this study include VGG16, VGG19, InceptionV3, Xception, DenseNet121, ResNet50, and InceptionResNetV2. The parameters of original deep

architecture were maintained during both fine-tuning and full-training. However, a learning rate of 0.01 and decay rate of 10-4 were selected for full-training and fine-tuning experiments. The number of iterations was set to be 30,000 and 100,000 for fine-tuning and full-training, respectively. Cross-entropy and Stochastic Gradient Descent (SGD) were selected as the loss function and the optimization algorithm, respectively, during processing. As mentioned earlier, the patch size of 30 was selected and the images were resized to 224x224 for VGG16, VGG19, DenseNet121, and ResNet50, as well as 229x299 for InceptionV3, Xception, InceptionResNetV2. All these experiments were implemented using Google's library TensorFlow (Girija, 2016). Table 3.1 presents the parameter settings and the characteristics of the deep convnets examined in this study.

In terms of computational complexity, the full-training strategy was more time intensive relative to the fine-tuning. This is because in the former, the network should be trained from scratch, wherein weights and biases are randomly initialized and, accordingly, more time and resources are required for the model to be convergent. Table 3.1 (last column) represents the processing time when full-training of five bands (from scratch) was carried out. In order to determine the most accurate processing time, each network was fed by 800 images (100 images for each class) and the training time was measured. This procedure was repeated ten times and the average processing time for each network is presented in Table 3.1.

All experiments were carried out on an Intel CPU i7 4790 k machine with a clock speed of 3.6 GHz and 32 GB RAM memory. A Nvidia GeForce GTX 1080 Ti GPU with 11 GB of memory under CUDA version 8.0 was also used in this study.

Table 3-1 The characteristics of deep convnets examined in this study.

ConvNet models	Parameters (millions)	Depth	Processing time[1] (s)
VGG16	138	23	18
VGG19	144	26	21
InceptionV3	24	159	10
ResNet50	26	168	12
Xception	23	126	16
InceptionResNetV2	56	572	19
DenseNet121	8	121	14

[1] Note that the processing time is calculated for training of 800 images (100 images for each class).

3.2.6 Evaluation metrices

Three metrics, namely overall accuracy, Kappa coefficient, and F1-score were used to evaluate the quantitative performance of different classifiers. Overall accuracy represents the correctly classified areas for the whole image and is calculated by dividing the number of correctly classified pixels to the total number of pixels in the confusion matrix. The kappa coefficient determines the degree of agreement between reference data and classified map. F1-scoreis a quantitative metric useful for the imbalanced training data and determines the balance between precision and recall. Precision, also known as positive predictive values, illustrates how many detected pixels for each category are true. Recall, also known as sensitivity, indicates how many actual pixels in each category are detected. Accordingly, F1-score is formulated as follows (Banko, 1998):

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3\text{-}1)$$

Where:

$$Precision = \frac{True\ positives}{True\ positive + False\ positive} \quad (3\text{-}2)$$

$$Recall = \frac{True\ positives}{True\ positive + False\ negative} \quad (3\text{-}3)$$

3.3 Results and discussion

In this study, fine-tuning was employed for pre-existing, well-known convnets, which were trained based on the ImageNet dataset. Figure 3.9 demonstrates the validation and training accuracies and loss in the case of fine-tuning of convnets using the three selected bands of the RapidEye images.

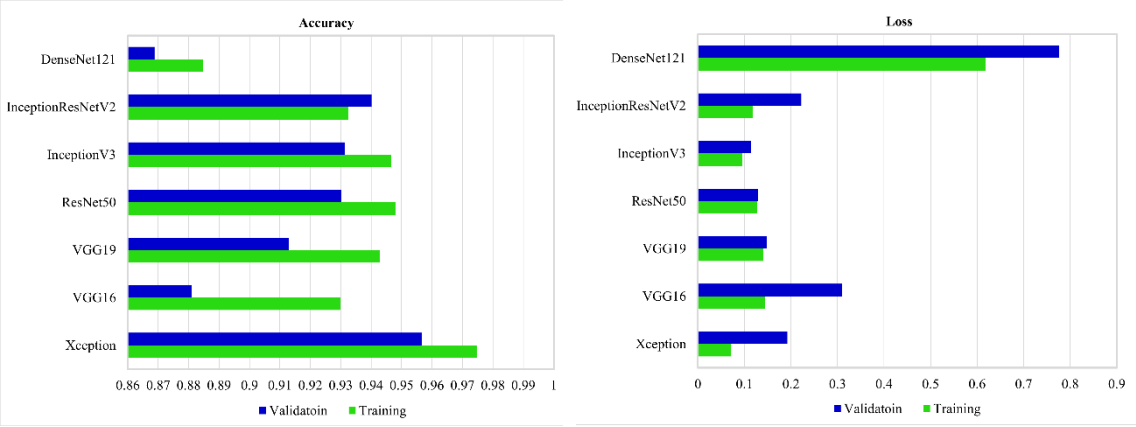


Figure 3-9 Comparing well-known convnets in terms of training and validation accuracies and loss when fine-tuning strategy of three bands (i.e., Green, Red, and NIR) was employed for complex wetland mapping.

As shown, DenseNet121 has the lowest validation accuracy, followed by VGG16. Conversely, Xception network has the highest validation accuracy, followed by InceptioResNetV2. The two convnets, namely InceptionV3 and ResNet50, show relatively equal validation accuracies. Figure 3.10 shows the validation and training accuracies and loss in the case of training convnets from scratch when three bands of the RapidEye images were employed.

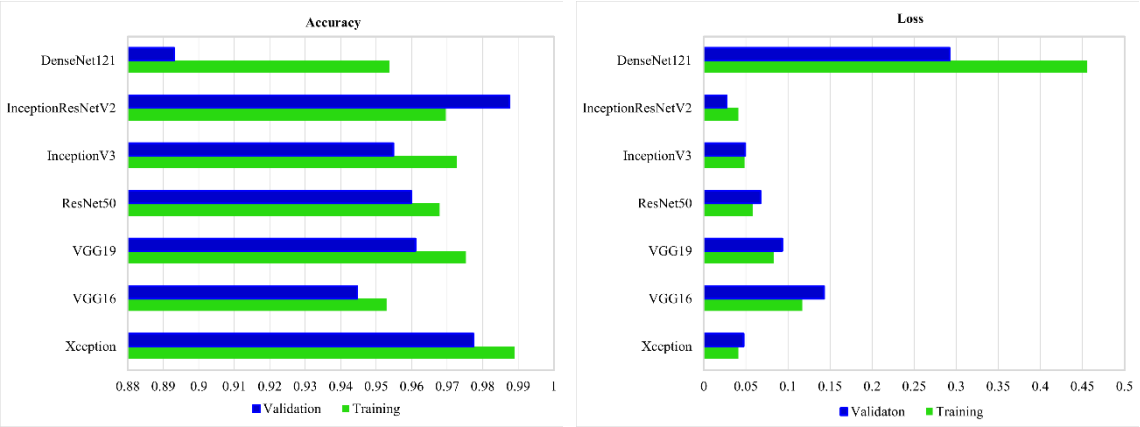


Figure 3-10 Comparing well-known convnets in terms of training and validation accuracies and loss when networks were trained from scratch using three bands (i.e., Green, Red, and NIR) for complex wetland mapping.

As shown, all convnets, excluding DensNet121, performed very well for wetland classification when validation accuracies are compared. In particular, three convnets, including InceptionResNetV2, Xception, and VGG19, have higher training and validation accuracies relative to the other well-known convnets. Conversely, DenseNet121 has the lowest validation accuracy, suggesting this network is less suitable for complex land cover mapping relative to the other convnets. Figure 3.11 shows the validation and

training accuracies and loss in the case of training convnets from scratch when five bands of the RapidEye images were employed.

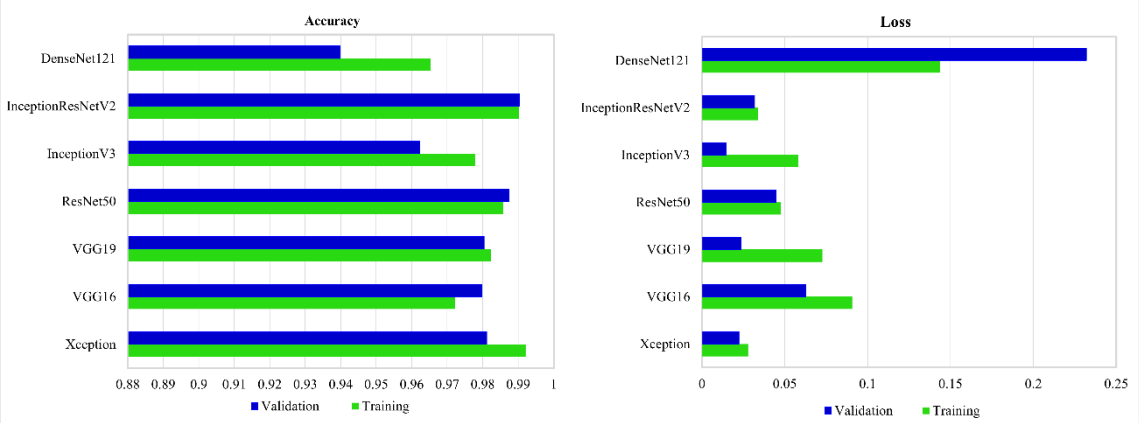


Figure 3-11 Comparing well-known convnets in terms of training and validation accuracies and loss when networks were trained from scratch using five bands for complex wetland mapping.

The influence of increasing the number of bands is readily apparent by comparing Figures 3.10 and 3.11. Specifically, an increase of the number of bands improves the classification accuracy in all cases. For example, the validation accuracy for DenseNet121 was lower than 90% when only three bands were employed. However, by increasing the number of bands, the validation accuracy increased to 94% for DenseNet121. InceptionResNetV2 again exhibited the highest validation accuracy followed by ResNet50, Xception, and VGG19. Thus, the results indicate the significance of incorporating more spectral information for the classification of spectrally similar wetland classes (see Figures 3.10 and 3.11).

One of the most interesting aspects of the results obtained in this study is that the full-training strategy had better classification results relative to the fine-tuning in all cases.

Previous studies reported the superiority of fine-tuning relative to full-training for classification of very high resolution aerial imagery, although full-training was found to be more accurate relative to fine-tuning for classification of multi-spectral satellite data (Castelluccio, Poggi, Sansone, & Verdoliva, 2015). In particular, Nogueira et al. (2017) evaluated the efficiency of fine- and full-training strategies of some well-known deep CNNs (e.g., AlexNet and GoogLeNet) for classification of three well-known datasets, including UCMerced land-use (Castelluccio et al., 2015), RS19 dataset (Xia et al., 2010), and Brazilian Coffee Scenes (Penatti, Nogueira, & dos Santos, 2015). The ص strategy yielded a higher accuracy for the first two datasets, likely due to their similarity with the ImageNet dataset, which was originally used for training these deep CNNs. However, the full-training strategy had similar (Nogueira et al., 2017) or better results (Castelluccio et al., 2015) relative to the fine-tuning for the Brazilian Coffee Scenes. This is because the latter dataset is multi-spectral (SPOT), containing finer and more homogeneous textures, wherein the patterns are substantially visually overlapping and, importantly, different than everyday objects found within the ImageNet dataset (Nogueira et al., 2017). The results obtained from the latter dataset were similar to those found in our study. In particular, there is a significant difference between the original training datasets of these convnets and our dataset. Fine-tuning is an optimal solution when the edges and local structures within the dataset of interest are similar to those for which the networks were trained. However, the texture, colour, edges, and local structures are very different between wetland classes and typical objects found in the ImageNet dataset. Moreover, our dataset is intrinsically different than those used in the ImageNet dataset used for the pre-training. In particular, our dataset has five spectral bands, including red, green, blue,

red-edge, and near-infrared; all of them are essential for classifying spectrally similar wetland classes. However, the ImageNet dataset has only red, green, and blue bands (Castelluccio et al., 2015). This could explain the difference between validation accuracies obtained in the case of full-training and fine-tuning (see Figures 3.9 and 3.10). Nevertheless, the results obtained from fine-tuning are still very promising, taking into account the complexity of wetland classes and the high classification accuracy obtained in most cases. In particular, an average validation accuracy of greater than 86% was achieved in all cases (see Figure 3.9), suggesting the generality and versatility of pre-trained deep convnets for the classification of various land cover types. It is also worth noting that the fine-tuning was employed on the top three layers of convnets in this study. However, the results could be different upon the inclusion of more layers in the fine-tuning procedure.

Having obtained the higher accuracies via full-training of five bands, the classification results obtained from this strategy were selected for further analysis. These classification results were also compared with results obtained from two conventional machine learning tools (i.e., SVM and RF). For this purpose, a total number of eight features, namely normalized difference vegetation index (NDVI), normalized difference water index (NDWI), Red-edge Normalized Difference Vegetation Index (ReNDVI), and all original spectral bands of the RapidEye image, were used as input features for both the SVM and RF classifiers. Table 3.2 represents the overall accuracy, Kappa coefficient, and F1-score using different CNNs (full-training of five bands), RF, and SVM for wetland classification in this study.

Table 3-2 Overall accuracies (%), Kappa coefficients, and F1-score (%) for wetland classification using different deep convnets (full-training of five bands), RF, and SVM.

Methods	Overall Accuracy	Kappa coefficient	F1
SVM	74.89	0.68	53.58
RF	76.08	0.70	58.87
DenseNet121	84.78	0.80	72.61
InceptionV3	86.14	0.82	75.09
VGG16	87.77	0.84	78.13
VGG19	90.94	0.88	84.20
Xception	93.57	0.92	89.55
ResNet50	94.81	0.93	91.39
InceptionResNetV2	96.17	0.95	93.66

As seen in Table 3.2, SVM and RF have the lowest classification accuracies and F1-score relative to all deep convnets in this study. Among deep convnets, InceptionResNetV2 has the highest classification accuracy of 96.17%, as well as F1-score of 93.66%, followed by ResNet50 and Xception with overall accuracies of 94.81% and 93.57%, as well as F1-score of 91.39% and 89.55%, respectively. Conversely, DenseNet121 and InceptionV3 have the lowest overall accuracies of 84.78% and 86.14%, as well as F1-score of 72.61% and 75.09%, respectively. VGG19 was found to be more accurate than VGG16 by about 3% (OA), presumably due to the deeper structure of the former convnet. These results are in general agreement with Han, Feng, Wang, and Cheng (2017), which reported the superiority of ResNet relative to GoogLeNet (Inception), VGG16, and VGG19 for the classification of four public remote sensing datasets (e.g., UCM, WHU-RS19). InceptionResNetV2 benefits from an integration of two well-known deep convnets,

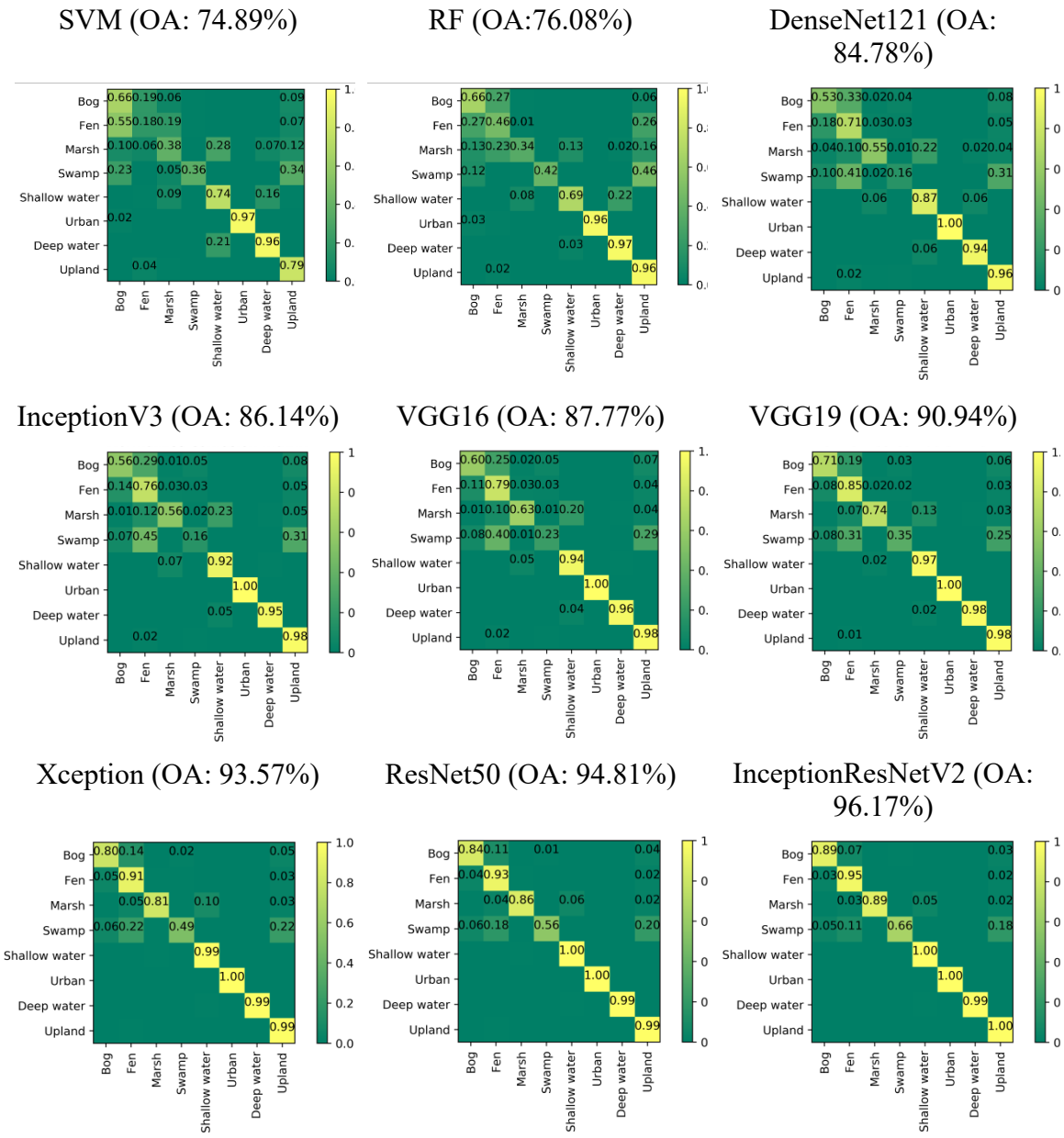


Figure 3-12 Normalized confusion matrix of the wetland classification for different networks in this study (full-training of five optical bands), RF, and SVM.

Inception and ResNet, which positively contribute to the most accurate result in this study. This also suggests that the extracted features from different convnets are supplementary and improve the model's classification efficiency. The results

demonstrated that a greater efficiency of deeper networks (e.g., InceptionResNetV2) in extracting varying degrees of abstraction and representation within the hierarchical learning scheme (Chen, Lin, Zhao, Wang, & Gu, 2014). In particular, they are more efficient for separating the input space into more detailed regions, owing to their deeper architecture, that contributes to a better separation of complex wetland classes.

As shown in Figure 3.12, all deep networks were successful in classifying non-wetland classes, including urban, deep water, and upland classes, with an accuracy greater than 94% in all cases. SVM and RF also correctly classified the non-wetland classes with an accuracy exceeding 96% in most cases (excluding upland). Interestingly, all deep networks correctly classified the urban class with an accuracy of 100%, suggesting the robustness of the deep learning features for classification of complex human-made structures (e.g., buildings and roads). This observation fits well with (Zhao & Du, 2016). However, the accuracy of the urban class did not exceed 97% when either RF or SVM employed.

The confusion matrices demonstrate that by using the last three networks, a significant improvement was achieved in the accuracy of both overall and individual classes. In particular, InceptionResNetV2 correctly classified non-wetland classes with an accuracy of 99% for deep water and 100% for both urban and upland classes. ResNet50 and Xception were also successful in distinguishing non-wetland classes with an accuracy of 100% for urban and 99% for both deep water and upland. One possible explanation for why the highest accuracies were obtained for these classes is the availability of larger amounts of training samples for non-wetland classes relative to the wetland classes.

Although RF and SVM, as well as all convnets, performed very well in distinguishing non-wetland classes, the difference in accuracies between those two groups (i.e., conventional classifiers versus deep networks) was significant for wetland classes. This was particularly true for the last three convnets compared to SVM and RF. Specifically, the three networks of InceptionResNetV2, ResNet50, and Xception were successful in classifying all wetland classes with accuracies exceeding 80%, excluding the swamp wetland. This contrasts with results obtained from SVM and RF, wherein the accuracies were lower than 74% for all wetland classes. Overall, the swamp wetland had the lowest accuracy among all classes using the deep convnets. As the effectiveness of these networks largely depends on the amount of training samples, the lowest accuracy of the swamp wetland could be attributable to the lower quantity of training samples for this class.

A large degree of confusion was observed between herbaceous wetlands (namely marsh, bog, and fen, and especially between bog and fen), when DenseNet121, InceptionV3, and VGG16 were employed. The largest confusion between bog and fen is possibly due to the very similar visual features of these classes (see Figure 3.8). These two classes are both peatland dominated with different species of Sphagnum in bogs and Graminoid in fens. According to field biologist reports, these two classes were adjacent successional classes with a heterogeneous nature and were hardly distinguished from each other during the in-situ field data collection.

Overall, this confusion problem was more pronounced among the first four deep networks, whereas it was significantly reduced when the last three networks were employed (see Figure 3.12).

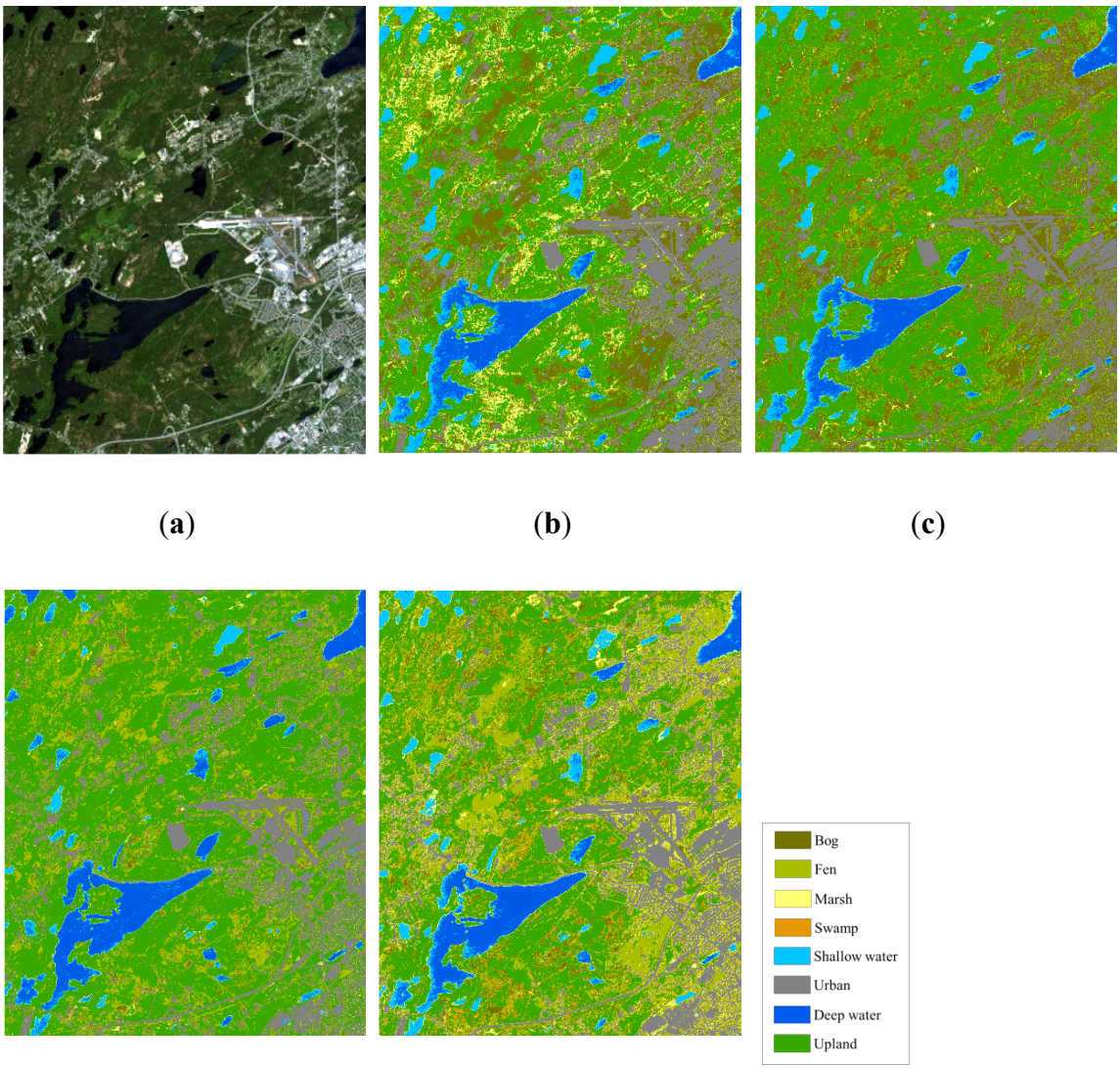
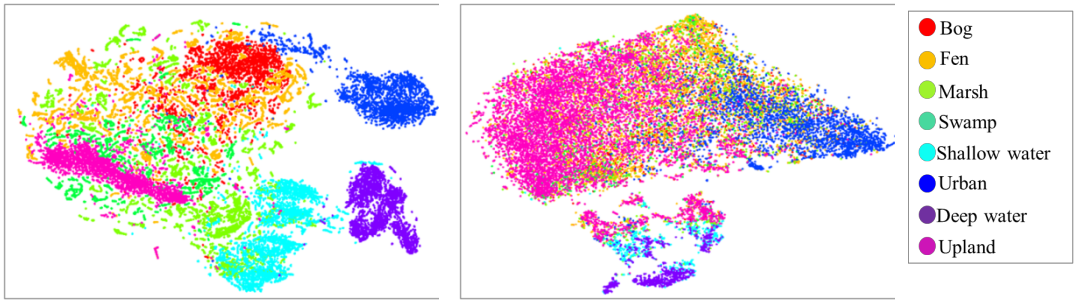


Figure 3-13 (**a**) True colour composite of RapidEye optical image (bands 3, 2, and 1). A crop of the classified maps obtained from (**b**) SVM, (**c**) RF, (**d**) DenseNet121, and (**e**) InceptionResNetV2.

This observation suggests that the last three networks and, especially, InceptioResNetV2, are superior for distinguishing confusing wetland classes relative to the other convnets. For example, the classes of bog and fen were correctly classified with accuracies of

greater than 89% when InceptionResNetV2 was used. Both Xception and ResNet50 were also found to successfully classify these two classes with accuracies of higher than 80%. Overall, the wetland classification accuracies obtained from these three networks were strongly positive for several spectrally and spatially similar wetland classes (e.g., bog, fen, and marsh), and these accuracies demonstrate the large number of correctly classified pixels.

A crop of the classified maps obtained from SVM, RF, DenseNet121, and InceptionResNetV2 are depicted in Figure 3.13. As shown, the classified maps obtained from convnets better resemble the real ground features. Both classified maps, obtained from convnets (Figures 3.13 (**d**) and (**e**)) show a detailed distribution of all land cover classes; however, the classified map obtained from InceptionResNetV2 (Figure 3.13 (**e**)) is more accurate when it is compared with optical imagery (Figure 3.13 (**a**)). For example, in some cases in the classified map obtained from DenseNet121, the fen class was misclassified as bog and upland classes (Figure 3.13 (**d**)). This, too, occurred between shallow water and deep water; however, it was not the case when InceptionResNetV2 was employed. In particular, most land cover classes obtained from InceptionResNetV2 are accurate representations of ground features. This conclusion was based on the confusion matrix (see Figure 3.12) and further supported by a comparison between the classified map and the optical data (Figure 3.13 (**a**) and (**e**)).



(**a**) (**b**)

Figure 3-14 2-D feature visualization of image global representation of the wetland classes using t-SNE algorithm for the last layer of (**a**) InceptionResNetV2 and (**b**) DenseNet121. Each colour illustrates a different class in the dataset.

Figure 3.14 shows two-dimensional features extracted from the last layer of the InceptionResNetV2 (**a**) and DenseNet121 (**b**) and using the two-dimensional t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm (Chen et al., 2014). The features from InceptionResNetV2 demonstrate a clear semantic clustering. In particular, most classes are clearly separated from each other; however, the feature clusters of bog and fen show some degree of confusion. Conversely, the features from DenseNet121 only generate a few visible clusters (e.g., upland and urban), while other features corresponding to wetland classes overlap with each other, suggesting a large degree of confusion.

3.4 Conclusions

Wetlands are characterized as complex land cover with high within-class variability and low between-class disparity, posing several challenges to conventional machine learning tools in classification tasks. To date, the discrimination of such complex land cover using

conventional classifiers, heavily relies on the large number of hand-crafted features incorporated into the classification scheme. In this research, we used state-of-the-art machine learning tools, deep Convolutional Neural Networks, for classification of such a heterogeneous environment to address the problem of extracting a large number of hand-crafted features. Two different strategies of employing pre-existing convnets were investigated: full-training and fine-tuning. The potential of the most well-known deep convnets, while currently are being employed for several computer vision tasks, including DenseNet121, InceptionV3, VGG16, VGG19, Xception, ResNet50, and InceptionResNetV2, was examined in a comprehensive and elaborate framework using multispectral RapidEye optical data for wetland classification.

The results of this study revealed that the incorporation of high-level features, learned by a hierarchical deep framework, is very efficient for the classification of complex wetland classes. Specifically, the results illustrate that the full-training of pre-existing convnets using five bands is more accurate than both full-training and fine-tuning using three bands, suggesting that the extra multispectral bands provide complementary information. In this study, InceptionResNetV2 consistently outperformed all other convnets for the classification of wetland and non-wetland classes, with a state-of-the-art overall classification accuracy of about 96%, followed by ResNet50 and Xception with an accuracy of about 94% and 93%, respectively. The impressive performance of InceptionResNetV2 suggests that an integration of Inception and ResNet modules is an effective architecture for complex land cover mapping using multispectral remote sensing images. The individual class accuracy illustrated that confusion occurred between wetland classes (herbaceous wetlands), although it was less pronounced when

InceptionResNetV2, ResNet50, and Xception were employed. The swamp wetland had the lowest accuracy in all cases, potentially because the lowest number of training samples were available for this class. It is also worth noting that all deep convnets were very successful in classifying non-wetland classes in this study.

The results of this study demonstrate the potential for the full exploitation of pre-existing deep convnets for classification of multispectral remote sensing data, which are significantly different than the large datasets (e.g., ImageNet) currently employed in computer vision. Given the similarity of wetland classes across Canada, the deep trained networks in this study provide valuable baseline information and tools. They will substantially contribute to the success of wetland mapping in this country, using state-of-the-art remote sensing tools and data.

References

Adam, E., Mutanga, O., & Rugege, D. (2010). Multispectral and hyperspectral remote sensing for identification and mapping of wetland vegetation: a review. *Wetlands Ecology and Management*, *18*(3), 281–296.

Ball, J. E., Anderson, D. T., & Chan, C. S. (2017). Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of Applied Remote Sensing*, *11*(4), 42609.

Banko, G. (1998). A review of assessing the accuracy of classifications of remotely sensed data and of methods including remote sensing data in forest inventory.

Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*(1), 2–16. Retrieved from http://www.sciencedirect.com/science/article/pii/S0924271609000884

Castelluccio, M., Poggi, G., Sansone, C., & Verdoliva, L. (2015). Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. *CoRR*, *abs/1508.0*. Retrieved from http://arxiv.org/abs/1508.00092

Chen, X., Xiang, S., Liu, C.-L., & Pan, C.-H. (2014). Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, *11*(10), 1797–1801.

Chen, Y., Lin, Z., Zhao, X., Wang, G., & Gu, Y. (2014). Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(6), 2094–2107.

Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. *ArXiv Preprint*.

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Company. Retrieved from https://books.google.ca/books?id=Yo3CAQAACAAJ

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 248–255). Ieee.

Evans, T. L., & Costa, M. (2013). Landcover classification of the Lower Nhecolândia subregion of the Brazilian Pantanal Wetlands using ALOS/PALSAR, RADARSAT-2 and ENVISAT/ASAR imagery. *Remote Sensing of Environment*, *128*(Supplement C), 118–137. https://doi.org/https://doi.org/10.1016/j.rse.2012.09.022

Friedl, M. A., & Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, *61*(3), 399–409.

Frohn, R. C., Autrey, B. C., Lane, C. R., & Reif, M. (2011). Segmentation and object-oriented classification of wetlands in a karst Florida landscape using multi-season Landsat-7 ETM+ imagery. *International Journal of Remote Sensing*, *32*(5), 1471–1489.

Girija, S. S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Han, W., Feng, R., Wang, L., & Cheng, Y. (2017). A semi-supervised generative framework with deep learning features for high-resolution remote sensing image scene classification. *ISPRS Journal of Photogrammetry and Remote Sensing*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hestir, E. L., Khanna, S., Andrew, M. E., Santos, M. J., Viers, J. H., Greenberg, J. A., … Ustin, S. L. (2008). Identification of invasive vegetation using hyperspectral remote sensing in the California Delta ecosystem. *Remote Sensing of Environment*, *112*(11), 4034–4047.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hinton, G., & Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(July), 504–507. Retrieved from http://www.sciencemag.org/content/313/5786/504.short

Hu, F., Xia, G.-S., Hu, J., & Zhang, L. (2015). Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, *7*(11), 14680–14707.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In *CVPR* (Vol. 1, p. 3).

Jackson, Q., & Landgrebe, D. A. (2002). Adaptive Bayesian contextual classification based on Markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, *40*(11), 2454–2463. https://doi.org/10.1109/TGRS.2002.805087

Kang, M., Ji, K., Leng, X., Xing, X., & Zou, H. (2017). Synthetic aperture radar target recognition with feature fusion based on a stacked autoencoder. *Sensors*, *17*(1), 192.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012a). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012b). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. Retrieved from http://dx.doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Mahdianpari, M., Salehi, B., Mohammadimanesh, F., & Brisco, B. (2017). An Assessment of Simulated Compact Polarimetric SAR Data for Wetland Classification Using Random Forest Algorithm. *Canadian Journal of Remote Sensing*, *43*(5). https://doi.org/10.1080/07038992.2017.1381550

Mahdianpari, M., Salehi, B., Mohammadimanesh, F., Brisco, B., Mahdavi, S., Amani, M., & Granger, J. E. (2018). Fisher Linear Discriminant Analysis of coherency matrix for wetland classification using PolSAR imagery. *Remote Sensing of Environment*, *206*, 300–317. https://doi.org/10.1016/j.rse.2017.11.005

Mahdianpari, M., Salehi, B., Mohammadimanesh, F., & Motagh, M. (2017). Random forest wetland classification using ALOS-2 L-band, RADARSAT-2 C-band, and TerraSAR-X imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *130*, 13–31.

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

Mnih, V. (2013). Machine Learning for Aerial Image Labeling, 109.

Nogueira, K., Penatti, O. A. B., & dos Santos, J. A. (2017). Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification. *Pattern Recognition*, *61*, 539–556.

Patterson, J., & Gibson, A. (2017). Deep learning: A practitioner's approach. " O'Reilly Media, Inc.".

Penatti, O. A. B., Nogueira, K., & dos Santos, J. A. (2015). Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 44–51).

Sifre, L., & Mallat, S. (2013). Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1233–1240).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI* (Vol. 4, p. 12).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2014). Going Deeper with Convolutions. *ArXiv Preprint ArXiv:1409.4842*, 1–12. Retrieved from http://arxiv.org/abs/1409.4842v1

Tiner, R. W., Lang, M. W., & Klemas, V. V. (2015). *Remote sensing of wetlands: applications and advances*. CRC Press.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*(Dec), 3371–3408.

Xia, G.-S., Yang, W., Delon, J., Gousseau, Y., Sun, H., & Maître, H. (2010). Structural high-resolution satellite image indexing. In *ISPRS TC VII Symposium-100 Years ISPRS* (Vol. 38, pp. 298–303).

Zhang, L., Zhang, L., & Kumar, V. (2016). Deep learning for Remote Sensing Data. *IEEE Geoscience and Remote Sensing Magazine*, *4*(2), 22–40. https://doi.org/10.1155/2016/7954154

Zhao, W., & Du, S. (2016). Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *113*, 155–165.

Zhong, P., & Wang, R. (2010). Learning conditional random fields for classification of hyperspectral images. *IEEE Transactions on Image Processing*, *19*(7), 1890–1907.

4 Chapter 4: Detection of Individual Tree Species Using an Optimized Deep CNN in an Object-Based Approach

Abstract

Acquiring information about individual tree species such as types and sizes is a crucial element of forests monitoring. Because of the inefficiency of the single-sensor data, such information is practically assessed using multi-sensor data with heavy manual intervention (or with manual interpretation), which is subjective, time and cost consuming, and prone to error. The current availability of higher spatial- or spectral-resolution satellite/aerial/unmanned aerial vehicle (UAV) data has opened new possibilities for single-sensor-based individual tree species detection (ITSD); however, extracting information from such sources is still challenging. In this paper, we introduced a deep convolutional neural network for ITSD using a single-sensor WorldView-3 image. The reason for using deep learning (DL) is its capability to automatically generating sufficient features to extract information for object detection in machine learning field. We call our network deep individual tree detection network (DITDN), which is developed by optimizing Oxford's renowned Visual Geometry Group (VGG-16) network using a tree-structure parzen estimator. The network receives image objects (segments) with any number of spectral bands and detects the type of the trees. Results demonstrate that using the test data, our method reaches an overall accuracy of about 92%, which outperformed two state-of-the-art ensemble classifiers, random forest (RF) and gradient

boosting (GB; 80% and 83% respectively). In addition, to show the generality of the method and to evaluate our method's accuracy with uncorrelated test data, we apply DITDN to another WorldView-3 image, reaching an overall accuracy of 89%. This study demonstrates the potential of using deep learning in an object-based approach for ITSD using Worldview-3 images and the alike.

4.1 Introduction

Information about forests, such as individual tree species present in the forest, are important for the environment and the forestry industry. Forests cover almost one-third of the earth's surface and are considered one of the most important elements of the natural environment. The importance of forests to the natural environment includes the provision of animal habitats, medicinal value to humans, and raw materials to sustain human life. Forests also support ecosystems, protect watersheds, purify air, stabilize climate, enrich the soil, and regulate the water cycle (Amatya, Williams, Bren, & de Jong, 2016). Therefore, constant monitoring of tree species and their health helps preserve the environment. However, human interpreters currently assess such information using aerial imagery or unmanned aerial vehicles (UAVs), a process that is subjective, time and cost consuming, and prone to errors (Yu et al., 2017).

Use of images acquired by remote-sensing for extracting forest information demonstrates potential for fast and cost-effective extraction of forest information. Some studies, using handheld hyper spectrometers, have proven that tree species can be distinguished from each other based on their leaves' spectral characteristics (Asner, Martin, Ford, Metcalee, & Liddell, 2009; Féret & Asner, 2011). Given this, many researchers have attempted to

detect tree species using satellite/aerial images. Developed methods for this effort can be divided into two groups: the pixel-based approach for making pixel-based detection maps and the object-based approach for individual tree species detection (ITSD).

The first group utilized pixel-based approaches to produce a tree-species cover map using low-resolution hyperspectral imagery (Féret & Asner, 2013). A few studies also applied both low-resolution hyperspectral imagery and digital-surface models (DSMs) generated by light detection and ranging (LiDAR) to improve detection accuracy (Cho et al., 2010; Dalponte, Bruzzone, Vescovo, & Gianelle, 2009; Feret & Asner, 2013). However, because the used hyperspectral imagery was low-resolution, the pixel-based detection map was far from a precise species map.

The second group applied higher resolution imagery to ITSD. This group used LiDAR-generated DSMs in its tree delineation to find individual trees and then hyperspectral images to detect tree species. For instance, Colgan et al. (2012) used LiDAR and airborne hyperspectral images separately for classification and then fused the results in the level of class (Colgan, Baldeck, Féret, & Asner, 2012). Dinuls et al. (2012) used the same approach to find tree canopies using LiDAR data and then performed the detection of each canopy using multispectral images (Dinuls, Erins, Lorencs, Mednieks, & Sinica-Sinavskis, 2012). Heinzel and Koch (2012) improved this work by under-segmenting the LiDAR data and then classifying each segment based on the multispectral data (Heinzel & Koch, 2012). However, although different spectral features were used as a supplementary data, because the multi-spectral imagery used does not provide necessary spectral information to detect tree species, these studies' accuracy lacked promise. In addition, use of multi-sensor images is costly, specifically when airborne hyperspectral

imagery and LiDAR data are used. Finally, alignment between hyperspectral images and LiDAR DSMs remains a major technical issue.

Despite new advances in remote sensing tools and methods, the accuracy of ITSD methods based on single-sensor data is insufficient (Yu et al., 2017), especially given the similar spectral characteristics of most tree species; thus, most studies used multi-sensor data for ITSD (Zhen, Quackenbush, & Zhang, 2016). The main reason for using multi-sensor data even with high spectral and spatial sensor is the insufficiency of the generated features to incorporate both spatial and spectral information together. Therefore, the main efforts for tree species detection focused on the feature-engineering process (Chollet, 2017) to generate both spectral features (based on the hyper/multi-spectral image) and spatial features (based on the LiDAR point cloud/DSM) which required expert knowledge about the data. Furthermore, feature engineering is limited to low-medium level features, which, in the case of similar spectral/spatial characteristics, would remain insufficient (Zhao & Du, 2016).

The 2014 launch of the WorldView-3 satellite, as the first high-resolution satellite to simultaneously collect 17 spectral bands, including 8 short-wave-infrared (SWIR) spectral bands, 8 visible-and-near-infrared (VNIR) spectral bands, and 1 panchromatic (Pan) band presented a new opportunity for forest information extraction (DigitalGlobe, 2014). The commercially available resolution of SWIR, VNIR, and Pan bands are 7.5 m, 1.2 m, and 0.3m, respectively (Digitalglobe, 2014). This satellite's high spatial/spectral resolution can provide the necessary spatial information for tree-crown delineating; however, compared to a hyperspectral image, the spectral information it provides, alone, is insufficient for ITSD, meaning feature engineering is still necessary.

The machine learning field (LeCun, Bengio, & Hinton, 2015) has, of late, noted deep learning (DL) for addressing the limitations of other state-of-the-art methods, including support vector machines (SVMs) and random forest (RF; Ball, Anderson, & Chan, 2017). Among different DL networks, the convolutional neural network (CNN; LeCun et al., 2015) has been used widely in remote sensing, such as classification (Zhao & Du, 2016), segmentation (Marmanis et al., 2018), object detection (X. Chen, Xiang, Liu, & Pan, 2014), and change detection (De, Pirrone, Bovolo, Bruzzone, & Bhattacharya, 2017).

CNN is constructed by connecting multilayers and is characterized by the ability to generate features at different levels (Nogueira, Penatti, & dos Santos, 2017). Therefore, it can substitute manual feature engineering with an end-to-end workflow (Chollet, 2017). CNN generates different features in a hierarchical framework and at different levels (low, mid, high), combining spectral and spatial information (Nogueira et al., 2017). Since 2010, differently designed deep CNNs have been introduced. Some gained attention for their high accuracy and efficiency for detection and localization. These include LeNet, for hand digit recognition (LeCun, Bottou, Bengio, & Haffner, 1998), and object-detection networks, such as AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), VGG (Simonyan & Zisserman, 2014), GoogLenet (Szegedy et al., 2015), ResNet (He, Zhang, Ren, & Sun, 2016), and Xception (François Chollet, 2016).

Despite advances that DL networks represent in machine learning, their application to remote sensing has not been investigated and is mostly limited to classification in chiefly urban areas (Tian, Li, Xu, & Ma, 2018) and scene labeling (Cheng, Yang, Yao, Guo, & Han, 2018). In addition, because of the statistical inherent differences between machine-learning images and remote-sensing images, no available studies relate to the process of

fitting predesigned networks to remote-sensing data. Therefore, this state-of-the-art method should be investigated for detection and discrimination of objects in remote-sensing images in a more complex situation, where objects have similar spectral and spatial information—like ITSD.

This study chiefly aims to (1) examine the power of deep CNN in identifying individual tree species, (2) take advantage of a deep CNN for ITSD by using high spatial- and spectral-resolution images, (3) optimize the network and its parameters to better fit the data, (4) train and apply the deep CNN in an object-based approach, (5) compare the designed network results with the state-of-the-art classifiers RF, gradient boosting (GB), and the unoptimized deep CNN, and (6) examine the generality of the designed network by applying it to uncorrelated data from different areas. In summary, this study aims to benefit from the state-of-the-art segmentation and detection methods for ITSD and improve them with the help of deep networks and by using the WorldView-3 satellite images.

4.2 Methodology

4.2.1 Study area and dataset

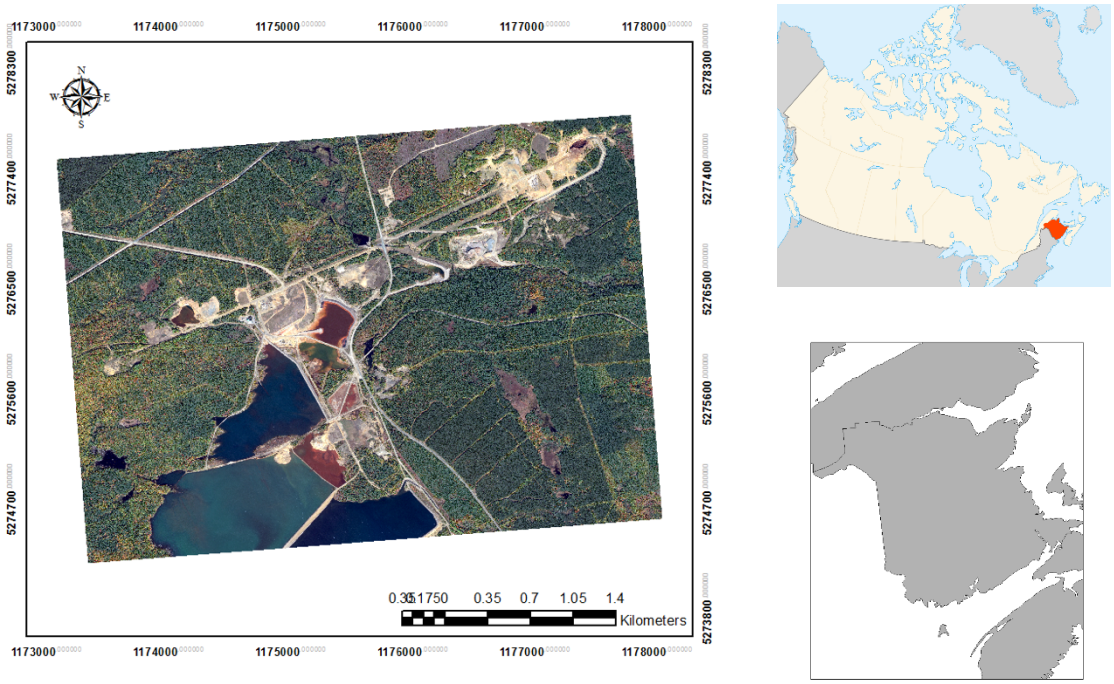
The study area (Heath Steele Mines) is in northern New Brunswick, Canada, and covers approximately 27 km^2. This area is covered mostly with four different tree species: spruce (mostly black), pine (mostly jack), birch (mostly white), and maple (mostly red). Because of the areas cold climate, the softwood population (spruce and pine) is greater than the hardwood (birch and maple).

For classification, the study used two WorldView-3 images, acquired on October 10, 2015, and July 27, 2016. To acquire the field data, a team from the University of New Brunswick departments of Forestry & Environmental Management and Geodesy and Geomatics Engineering collected samples from 7 different locations within the study area at two different times (August and September 2017). Since the trees' resistance to shaking varies, similar types of trees (mostly birch in our area) grow on the forest edge beside the roads. This fact forced us to collect samples from inside the forests as well. To access these interior areas, we used a UAV (DJI phantom 3 professional) to fly at a lower height and take pictures.

We identified 120 polygons in the satellite images along with some photographs with a Canon camera. We selected 70% of the data for training and testing. To split the collected data into training and testing, we sorted the polygons for each class based on their size

Table 4-1 Testing and training pixel counts for the heath Steele mines reference data

Class	Class Description	#Training Pixels	#Testing Pixels	Total
Spruce	Containing black spruce and few red spruce	10196	4371	14567
Pine	Dominated by jack pine, containing few eastern white pine	16323	6996	23319
Birch	Mostly white birch, few yellow birch observed	17810	7633	25443
Maple	Only red maple observed	10621	4552	15173
Other	Any types of ground containing roads (paved and unpaved), bare ground, and rocks. Any type of vegetation that is not included in the tree type is also covered here	35039	15018	50057
Total		89989	38570	128559



A true-colour composite of WorldView-3 optical imagery (bands 5, 3, and 2) acquired on July 27, 2016, illustrating the geographic location of the study area.

UAV image of spruce

UAV image of birch

Sample image of spruce

Figure 4-1 Working area, the original WorldView-3 image, some UAV and sample images

and alternatingly assigned them to testing and training groups. Table 4.1 shows the testing and training pixel counts we collected for our area. Figure 4.1 also demonstrates the satellite images, as well as some sample and UAV images.

4.2.2 Convolutional neural network (CNN)

This type of artificial neural network consists mainly of three different layers (Gibiansky, 2014):

- Convolutional layer: This layer consists of a rectangular grid. Each of its neurons inside the grid picks data from the preceding layer. The selected data are then multiplied by the corresponding weights to generate new values. The new values will be fed to the next layer as input. In fact, the rectangular weights act as a kernel and the whole process is like convolution. This process can be formulated as a simple convolution, as follows:

$$feature\ map = input * kernel = \sum_{y=0}^{columns} \left(\sum_{x=0}^{rows} input(x-a, y-b) kernel(x, y) \right). \quad (4\text{-}1)$$

- Pooling layer: This layer selects a small rectangular part of the previous layer and subsamples it to one output unit. There are different methods for doing subsampling, such as average, maximum, or a learned linear combination. In CNN, the maximum value is usually used.
- Fully connected layer: the network's high-level reasoning takes place in this layer. The neurons in this layer are fully connected to the all the neurons in the previous layer. This layer changes the 2D structures of the input (if there is any), and

change it to 1D; therefore, the spatial relationship between the pixels in the input will be lost in this layer.

4.2.3 Patch-based image labeling (PBIL)

Based on CNN, Mnih (2013) proposed a patch-based system for processing aerial images. The proposed method includes patches of aerial images with their known labels. The input images are called $S = (S^{(1)}, \dots, S^{(N)})$ and the corresponding map images are $M = (M^{(1)}, \dots, M^{(N)})$. The goal here is to develop a method to map *S* to *M* using training data and to predict the label for the rest of the patches without labels. This problem can be modeled as a probabilistic approach by learning a model of distribution over labels. It can be expressed as

$$P(n(M, i, w_m)|n(S, i, w_s)). \tag{4-2}$$

Where $n(I, i, w)$ shows a patch, which is focused on pixel *i,* with the size of $w \times w$ from image *I*. In this method, w_s is selected to be bigger than w_m to extract more contextual information from the patch for the pixel. To show its functional form, function *f* is defined. It maps the input patches to a distribution over the label patches. Considering the binary form as the easiest way of formulating the problem, function *f* can be defined as follows:

$$f_i(s) = \sigma\big(a_i(s)\big) = P(m_i = 1|s). \tag{4-3}$$

Where a_i is the total input for the *i*th output and f_i is the value of the *i*th output unit. $\sigma(x)$ is also a logistic function that is expressed as follows:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{4-4}$$

For multiclass labeling, artificial neural networks (ANN) should be accompanied by a softmax output unit. The output of the softmax would be a vector with a size equal to *L*, which shows the distribution over possible labels for pixel *i*. Thus, if we consider the path from pixel *i* to output unit *l*, the equation for multiclass labeling can be re-written as follows:

$$f_{il}(s) = \frac{exp\big(a_{il}(s)\big)}{Z} = P(m_i = l|s). \tag{4-5}$$

Where f_{il} shows the predicted probability that maps pixel *i* to label *j*. The proposed method uses ANN to approach its goals due to its advantages. These advantages can be summarized as follows: First, ANN can handle a large amount of label data in different domains. Secondly, ANN can be easily paralleled on a graphics processing unit (GPU) to make it faster. Thus, ANN can be extended to many images. Minimizing negative log-likelihood for training data performs the learning procedure in the proposed method. For a binary problem, the negative log-likelihood can be expressed as

$$\mathrm{L(s, m)} = \sum_{all\ patches} \sum_{i=1}^{w_m^2} \big(m_i \ln\big(f_i(s)\big) + (1 - m_i) \ln\big(1 - f_i(s)\big)\big). \tag{4-6}$$

The first sigma is over all the patches of training data. Because the number of training patches is large, the optimization problem is complex. To perform it, we used a stochastic gradient descent (SGD) with mini-batches (Le, Coates, Prochnow, & Ng, 2011). Some hyper-parameters should be tuned in SGD for a faster convergence. Mnih's (2013) study includes a sensitivity analysis for hyper-parameters to tune them more accurately.

4.2.4 Optimizing the network

Every convolutional neural network consists of different hyperparameters that define its architecture—from the number of hidden layers, as the network's major block, to the number of kernels in each convolution layer, size of pooling layers, loss-function, optimization, and regularization parameter. Every combination of these hyperparameters can generate a unique network like some predefined networks, including VGG-16, GoogLenet, etc. These hyperparameters can be adjusted for different datasets and applications. Most studies limit their adjustment to optimizing a few parameters (e.g., learning rate and loss function) to obtain a lower loss and higher accuracy (Mnih, 2013). However, other hyperparameters can also play an important role in generating a model that can better fit the data. One method to optimize hyperparameters for a CNN is the tree-structure parzen estimator (TPE; Bergstra, Bardenet, Bengio, & Kégl, 2011).

4.2.4.1 Tree-structure parzen estimator

This type of optimization follows the Bayes theory. Compared to the Gaussian-process-based approach, which models the conditional probability directly ($p(y|x)$), the Bayes optimization tries to model the $p(y|x)$ and $p(y)$ separately.

The process starts by applying the network to some selected data, then the TPE divides the evaluated results of selected data into two categories of what it calls high and low observations. In other words, using different observations $\{x^1, \ldots, x^k\}$, the TPE uses a learning algorithm that produces a different distribution of the observation over the configuration space χ. The TPE defines two sets of observations with two different densities as follows:

$$p(x|y) = \begin{cases} l(x) & if\ y < y^*. \\ g(x) & if\ y \geq y^*. \end{cases} \tag{4-7}$$

Where $l(x)$ is the distribution generated using the series of observation $\{x^{(i)}\}$ in a way that the corresponding loss function $f(x^{(i)})$ is less than y^*, $g(x)$ is the distribution of the remaining observation, and y^* is a given threshold. The TPE does not consider the lowest loss as y^* but does consider y^* in such a way that it contains some observations to form the distribution. The TPE considers y^* to be a portion of the whole observations (γ). Intuitively, the process creates a probability-distribution estimator that highlights both the set of hyperparameters that perform well $(l(x))$ and another set that act poorly $(g(x))$.

The optimization function in the TPE is considered as

$$\begin{aligned} f_{y^*}^{opt}(x) &= \int_{-\infty}^{y^*} (y^* - y)p(y|x)dy. \\ &= \int_{-\infty}^{y^*} (y^* - y)\frac{p(x|y)p(y)}{p(x)}dy, \end{aligned} \tag{4-8}$$

which can be simplified as

$$\propto \frac{1}{(\gamma + \frac{g(x)}{l(x)}(1-\gamma)}. \tag{4-9}$$

Equation 9 shows that, to maximize the $f_{y^*}^{opt}$, it is necessary to maximize $l(x)$, which is the high observations, and minimize $g(x)$ as the low observations. The TPE will generate a tree based on *l* and *g* distribution for all the candidates to find the best observation. $g(x)$ and $l(x)$ have a hierarchy structure covering all the continuous, discrete, and

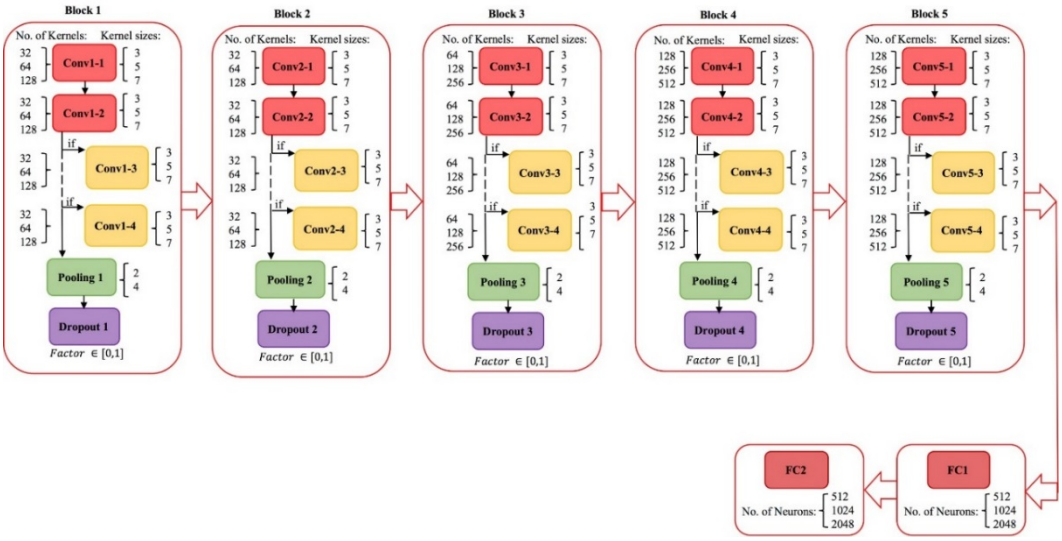
conditional variables. The method generates different nodes in a hierarchy; for each node a 1D TPE will be implemented and the hyperparameters updated, based on the loss of *l(x)* and *g(x)*. At the end, the TPE will start to track the hyperparameters from the roots of the tree to the leaves and follow the path that uses the active hyperparameters. It combines all the 1D results in the hierarchy to find the best set of hyperparameters.

4.2.4.2 The designated and optimized network

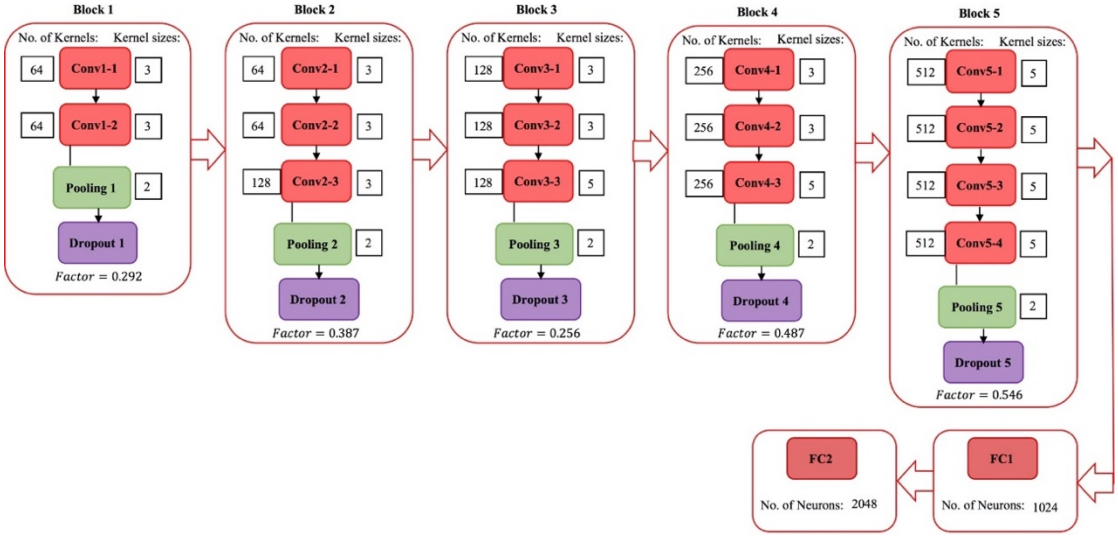
For this study, we optimized a VGG-16 network for ITSD and called it a deep individual tree detection network (DITDN). Optimized parameters contain the kernel size, number of kernels, size of max-pooling, number of hidden layers (a limited range), the dropout factors for regularization, batch size, loss function, and the learning rate. We performed the optimization using a Hyperas wrapper package (https://github.com/maxpumperla/hyperas). We designed 80 evaluations with separate training and validation data to find the best optimization parameters. Figure 4.2 demonstrates different considered options and the selected parameters.

4.2.5 Experiment setup

We fused the panchromatic and multispectral images using Fuze Go software (Scene Sharp Technologies Inc., 2012). We did the segmentation by using eCognition 9.1 (Trimble, 2017). Considering the circular shape of the hardwood crown and the softwoods' cone shape, the shape parameter in the multiresolution segmentation method received a higher weight than the colour. For the scale, we tested 10, 15, 20, and 25 to



The VGG-16 network with all the possible options for the hyperparameters. The yellow rectangles show the convolution layers that can be added to the network while the red ones are the fixed convolution layers. Other hyperparameters are shown in the brackets.



The DITDN network, based on the TPE optimizer.

Figure 4-2 The designated VGG network (a) and the optimized network (DITDN) (b).

The best learning rate, loss function, and batch size, with respect to our computer, is defined as 10^{-2}, stochastic gradient descent optimizer (SGD), and 250 respectively.

find the best parameter for our purpose. We considered the scale in a way to avoid under-segmentation. We selected the final value for the scale, shape, and compactness as 15, 0.7, and 0.5. We then detected and excluded the segments related to shadow and road from the processing in this step using their spectral information and the normalize difference vegetation index (NDVI).

Training the network with the PBIL approach requires the cropping of the original data into patches. Originally, PBIL used a universal patch size for all objects in the training areas. However, because the sizes of the objects (tree crowns) are dissimilar, finding a universal patch size would be difficult. Since the performed segmentation contains the information related to the objects' sizes, these sizes can be used as the patches' sizes. Therefore, each patch would be defined as a bounding box containing the segment. We also applied a buffer around the patch with the size of 5 pixels (1.5 meters) around the bounding box to ensure enough contextual information inside the patch. To prevent making a small patch in this way, we defined a threshold size of 20 pixels. We substituted any smaller segments with a bounding box that covered 20 pixels by 20 pixels on the image (6×6 meters).

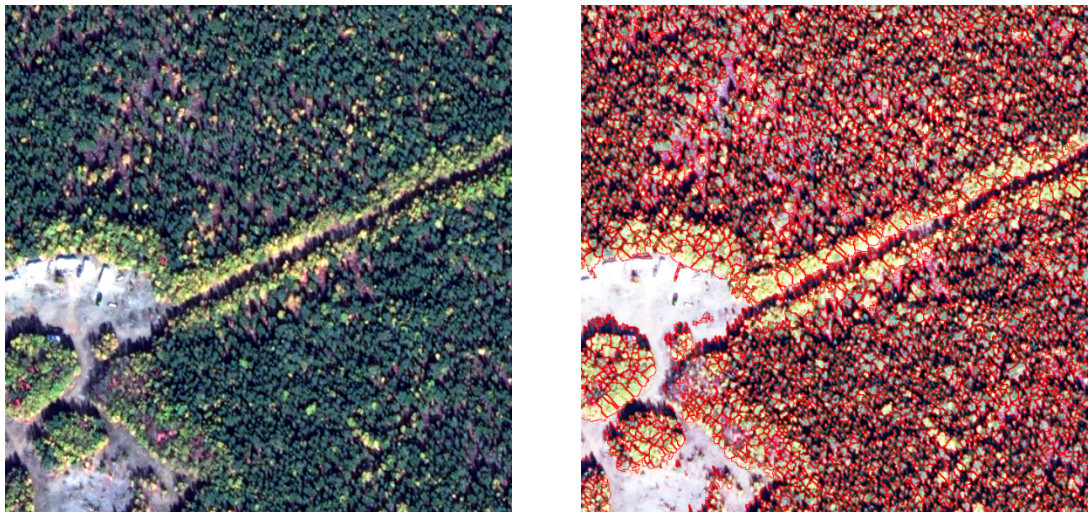
To train and test the CNN, we used the Keras library (François Chollet, 2015). However, this library is not designed to work with geo-tiff images. In other words, it does not recognize the TIFF format and, consequently, cannot read images with more than three spectral bands. To access the GeoTIFF images in this library, we designed a pipeline to read our specific data and handle the CPU memory to feed the patches to the library in

the Python 3.6. We carried out the implementation through support provided by Compute Canada on Cedar cluster using one node with 128 GB RAM memory and one GPU (NVIDIA P100-PCIE-12 GB).

4.3 Results

In this study, we developed a two-step process ITSD using single-sensor data. We first segmented the image to delineate the tree crowns, and then we applied our designed DITDN to detect tree types. We based DITDN's design on a VGG-16 network. To evaluate the DITDN's generality and its corresponding features, we used two datasets to evaluate two different images captured from two different areas at two different times.

Figure 4.3 shows the result of segmentation for the first test image.



(a) True-colour composite of the first test image.

(b) Segmentation image.

Figure 4-3 Result of the segmentation on the first test image, (a) showing the original image and (b) showing the corresponding segmentation image.

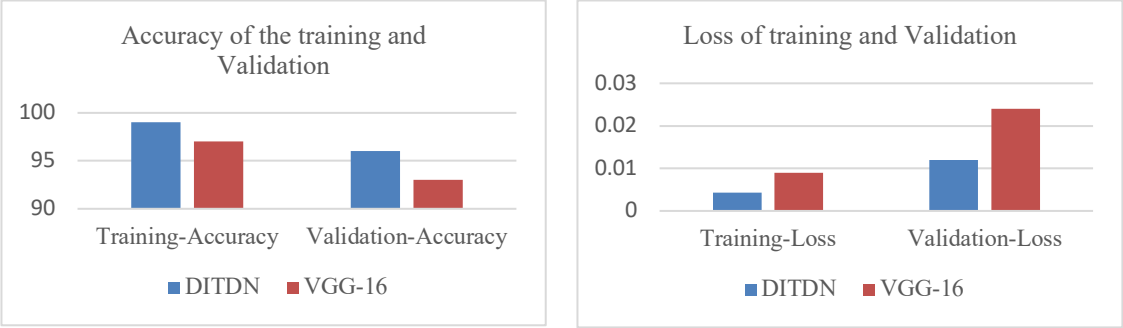


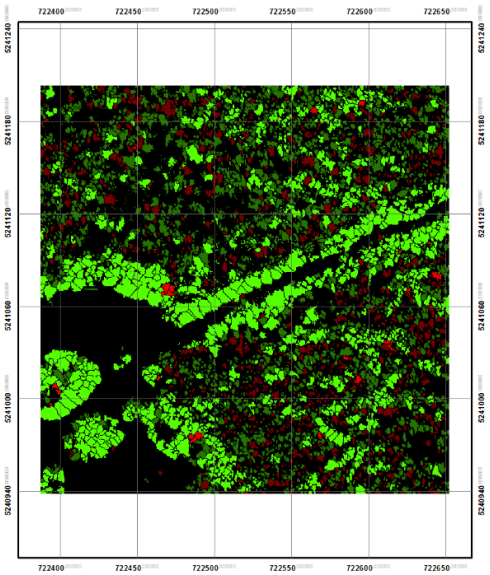
Figure 4-4 Accuracy (left) and loss (right) of the training and validation phase for the DITDN and VGG-16

To show DITDN fits the data better than VGG-16, we compared the accuracy of training and value of loss for both networks, as well as the test's accuracy. Figure 4.4 shows the accuracy and loss of training and validation for DITDN and VGG-16. Clearly, both networks are well trained for our dataset; however, DITDN had a higher accuracy than VGG-16 and less loss of validation data, indicating that the optimized network better fits our dataset, which relates to the network architecture and the optimization parameters such as network depth, optimization space, and the kernel sizes. Firstly, network depth can determine the nonlinearity of boundaries between different classes; a lower or higher number of layers can create boundaries unsuitable for the data. Secondly, defining optimization space can lead to a lower loss value and better training, while nonoptimized parameters can lead to under/over optimization. Finally, the kernels' sizes and numbers directly affect the generated features and, thereby, the network's accuracy.

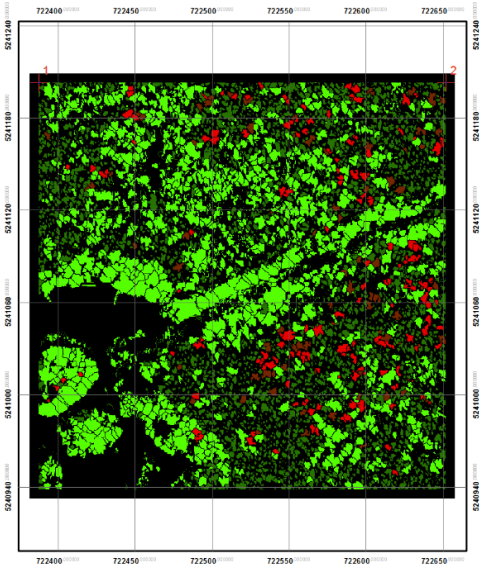
To better understand the CNN's classifier's power, we applied two state-of-the-art classifiers, RF and GB, to both test areas (Ball, Anderson, & Chan, 2017). The two main parameters of RF, which should be adjusted are the number of trees (Ntree) and the

number of variables (Mtry) (Belgiu & Drăguţ, 2016). In this study, a total number of 500 trees were selected in classification model. Moreover, the square root of the number of input variables was considered as Mtry. This is because it decreased both the computational complexity of the model and the correlation between trees (Gislason, Benediktsson, & Sveinsson, 2006). For GB, the performance is influenced by four parameters: the number of iterations, learning rate, the depth of the tree, and the sampling rate (Yang et al., 2018). In this study, these parameters are selected as 300, 0.8, 4, and 2 respectively. For applying RF and GB, spectral features should be manually designed and fed to the algorithm since no feature generation is ensembled to these two classifiers. Therefore, we used a total number of 10 features—normalized difference vegetation index (NDVI), normalized difference soil index (NDSI), and all the WorldView3 image's original spectral bands (8 bands). Figure 4.5 shows the four classification maps related to DITDN, VGG-16, RF, and GB.

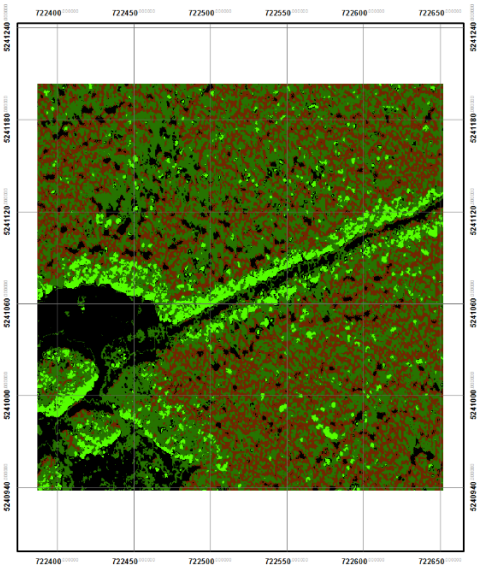
Although all classification maps show a detailed distribution of all tree species, DITDN obtained the most accurate map when compared to the field data. For example, the collected data shows few red maple trees in the field, while VGG-16 and GB detected



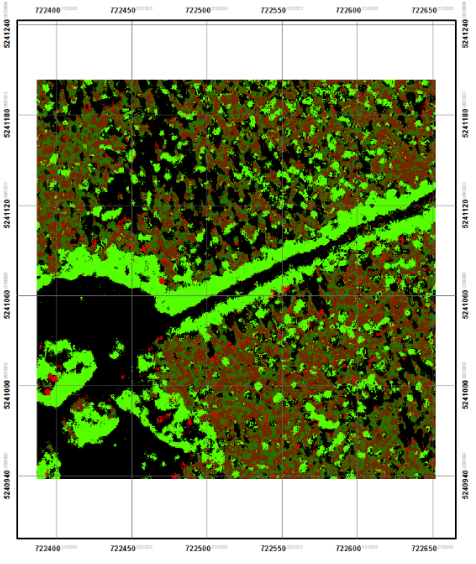
(a) Generated map based on DITDN



(b) Generated map based on VGG-16



(c) Generated map based on RF



(d) Generated map based on GB

Figure 4-5 The obtained maps from DITDN, VGG-16, RF, and GB

many and RF detected none. This false alarm in VGG-16 and GB relates to the spectral similarity between red maple and the ground area covered with red leaf bushes. Birches

also charted differently on the maps. Birches are shake resistant and therefore grow around the roads, as each map shows. However, birch's interior appearance frequency is low. DITDN, RF, and GB all seem to have the same pattern for this type, while VGG-16 falsely detected more trees as birch. This false alarm relates to the fact that birch tree leaves contain different shades of green, which in some areas is like spruce or pine, such as where it is under shadow.

Table 4.2 represents the overall accuracy of ITSD using two different deep networks and two ensembled classifiers that are evaluated using the test data for the aforementioned area.

Table 4-2 Classification overall accuracies and Kappa coefficients for ITSD using two different deep networks and two ensembled classifiers.

detection method	Overall Accuracy	Kappa coefficient
DITDN	92.13%	0.90
VGG-16	87.58%	0.84
Gradient Boosting (GB)	83.57%	0.80
Random Forest (RF)	80.12%	0.77

As seen in Table 4.2, DITDN has the highest classification accuracy (92.13%), followed by VGG-16, GB, and RF with overall accuracies of 87.58%, 83.57%, and 80.12%, respectively. The obtained accuracies for the deep networks suggest that the generated features are supplementary and that they improved the detection model's efficiency. Compared, the two deep networks follow the general rule for correlation between network depth and accuracy of the model (Y. Chen, Lin, Zhao, Wang, & Gu, 2014), which states that the level of abstraction in the features is related to the network's depth. However, the rule does not mean that deeper networks will consequently generate more

accurate results but relates to the nature of the data, which is why an optimization from dataset to dataset is necessary. Table 4.3 and 4.4 show the confusion matrix for DITDN and VGG-16.

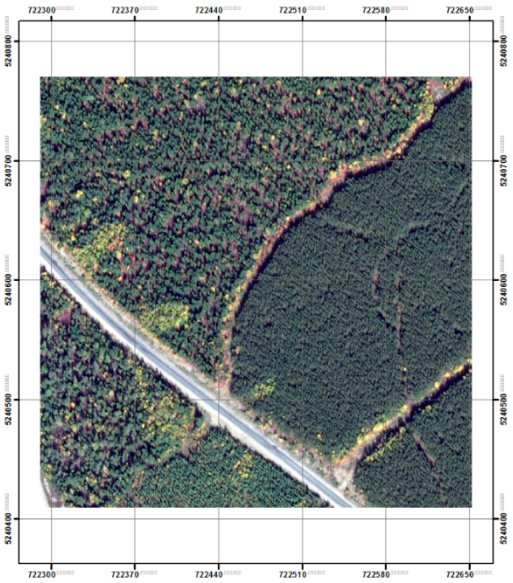
Table 4-3 Confusion matrix of DITDN, overall accuracy is 92.13%, kappa coefficient is 0.90

		Reference Data						
	Class	Red Maple	Birch	Spruce	Pine	Other	Tot.	User Acc.
Classified Data	Red Maple	4323	19	4	11	12	4369	98.94
	Birch	34	4098	26	123	78	4359	94.01
	Spruce	22	128	7623	956	43	8772	86.90
	Pine	12	71	773	7094	23	7973	88.97
	Other	234	22	51	59	8511	8877	95.87
	Total	4625	4338	8477	8243	8667	34350	
	Prod. Acc.	93.47	94.46	89.92	86.06	98.20		

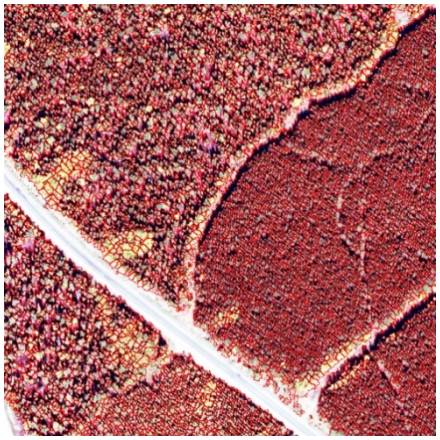
Table 4-4 Confusion matrix of VGG-16; overall accuracy is 87.58%, kappa coefficient is 0.84

		Reference Data						
	Class	Red Maple	Birch	Spruce	Pine	Other	Tot.	User Acc.
Classified Data	Red Maple	3645	13	121	234	356	4369	83.42
	Birch	34	3659	234	342	90	4359	83.94
	Spruce	26	128	7678	884	56	8772	87.52
	Pine	19	150	987	6693	124	7973	83.94
	Other	343	12	67	45	8410	8877	94.73
	Total	4067	3962	9087	8198	9036	34350	
	Prod. Acc.	89.62	92.35	84.49	81.64	93.07		

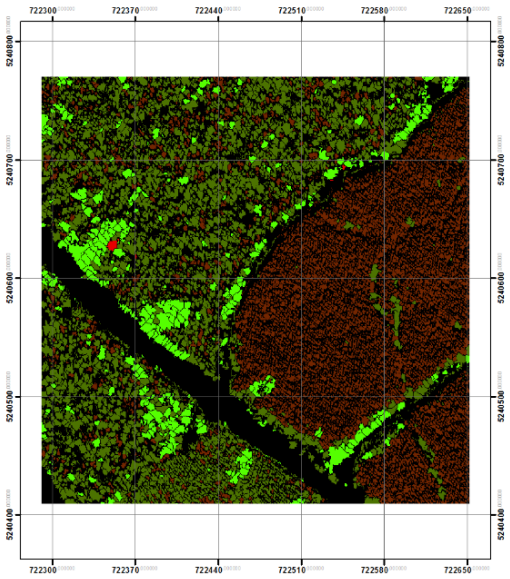
Table 4.3 and 44 show the confusion matrix for two deep networks. The spruce and pine class have the biggest misclassification, resultant from their similarity of spectral and spatial characteristics (87% and 89% user accuracy for DITDN). In addition, while the birch has different spectral information than spruce and pine, these trees can also be misclassified since, in some areas, birch grow amidst spruce or pine and because of shadow.



(a) True-colour composite of second test image



(b) The segmentation image



(c) The corresponding detection map

Figure 4-6 The original second test image (a), its segmentation (b), and corresponding detection map (c)

To evaluate if DITDN is reusable on other images (the generality of the network), we applied DITDN to a different image. Since the network was not trained with this data, the

result's accuracy demonstrates whether the generated features are general enough to detect tree species in this image as well. Figure 4.6 shows the second original image and the corresponding map generated with DITDN.

The right part of the forest in Figure 4.6 demonstrates where clearcutting was done. Our data collection included the observation that only pine was planted here. The generated map clearly shows that the same, except near the road where several tree species exist. Table 4.5 shows the confusion matrix for this map.

Table 4-5 Confusion matrix of DITDN for the second image; overall accuracy is 89.06%, kappa coefficient is 0.85

		Reference Data						
Classified Data	Class	Red Maple	Birch	Spruce	Pine	Other	Tot.	User Acc.
	Red Maple	634	13	31	10	22	710	89.30
	Birch	5	2456	156	234	78	2949	83.28
	Spruce	0	134	5432	544	33	6177	87.94
	Pine	0	164	456	4532	223	5398	83.96
	Other	12	23	98	63	5673	6081	93.29
	Total	651	2790	6173	5383	6029	21315	
	Prod. Acc.	97.39	88.03	88.00	84.19	94.10		

Considering Table 4.5, the obtained map's overall accuracy is 89.06% (4% drop compared to the first image), which shows both a promising result and the network's potential for application to other datasets. The confusion matrix, again, significantly misclassified pine and spruce, due to their similar spectral and spatial characteristics. Birch was also misclassified with all other classes due to its different shades of greenness. All this information further confirms previous results for the first dataset.

4.4 Conclusions

This study proposes a method to use the single-source WorldView-3 satellite images to detect and identify individual tree species. We developed a two-step processing approach. The first step applies a segmentation to delineate tree crowns followed by the second step, in which a deep CNN detects tree species. For the deep CNN, we designed a DITDN with the ability to receive all 8 visible and near-infrared spectral bands and trained it to detect four tree species (pine, spruce, red maple, and birch). DITDN incorporates the high-level features learned by a hierarchical deep framework with low-level features and can generate abstract features that can highlight each tree species and discriminate between them.

In this study, our DITDN outperformed the results of other investigated machine learning detection methods (RF and GB) reaching an overall accuracy of about 92.13%, whereas GB, and RF reached 83.57% and 80.12%, respectively. We also apply VGG16 network directly to see the effect of optimization. VGG16 reached 87.58%, showing that the optimization improved the accuracy by around 5%. Our DITDN's accuracy suggests that network optimization in terms of depth, parameters, and optimization space can generate a network that better fits the data and, thus, produces a more accurate detection map. In addition, the 89% detection accuracy obtained from the second test image's detection map demonstrates that the DITDN network's generated features are not dependent on the image the network trained with, demonstrating further that the network can be applied to other images and that its features to distinguish the same classes in new images taken in the same season.

This study's results demonstrate the potential of using deep learning networks in an object-based approach for ITSD applying to single-source high spatial/spectral satellite images. Considering central and eastern Canada have similar climates, DITDN can be applied to other forests in these areas. This work can be further improved via data collection of other tree species in other parts of New Brunswick and Canada and by training the network to identify them. We hope this study provides valuable baseline information and tools and will substantially contribute to the success of tree species mapping and forest monitoring using state-of-the-art remote sensing tools and data.

References

Amatya, D., Williams, T., Bren, L., & de Jong, C. (2016). *Forest Hydrology: Processes, Management and Assessment*. CABI.

Asner, G. P., Martin, R. E., Ford, A. J., Metcalee, D. J., & Liddell, M. J. (2009). Leaf chemical and spectral diversity in Australian tropical forests. *Ecological Applications*, *19*(1), 236–253. https://doi.org/10.1890/08-0023.1

Ball, J. E., Anderson, D. T., & Chan, C. S. (2017). Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of Applied Remote Sensing*, *11*(4), 42609.

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, *114*, 24–31.

Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems* (pp. 2546–2554).

Chen, X., Xiang, S., Liu, C.-L., & Pan, C.-H. (2014). Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, *11*(10), 1797–1801.

Chen, Y., Lin, Z., Zhao, X., Wang, G., & Gu, Y. (2014). Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(6), 2094–2107.

Cheng, G., Yang, C., Yao, X., Guo, L., & Han, J. (2018). When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs. *IEEE Transactions on Geoscience and Remote Sensing*.

Cho, M. A., Debba, P., Mathieu, R., Naidoo, L., Aardt, J. van, & Asner, G. P. (2010). Improving Discrimination of Savanna Tree Species Through a Multiple-Endmember Spectral Angle Mapper Approach: Canopy-Level Analysis. *IEEE Transactions on Geoscience and Remote Sensing*, *48*(11), 4133–4142. https://doi.org/10.1109/TGRS.2010.2058579

Chollet, F. (2015). Keras.

Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. *ArXiv Preprint*.

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Company. Retrieved from https://books.google.ca/books?id=Yo3CAQAACAAJ

Colgan, M. S., Baldeck, C. A., Féret, J. baptiste, & Asner, G. P. (2012). Mapping savanna tree species at ecosystem scales using support vector machine classification and BRDF correction on airborne hyperspectral and LiDAR data. *Remote Sensing*, *4*(11), 3462–3480. https://doi.org/10.3390/rs4113462

Dalponte, M., Bruzzone, L., Vescovo, L., & Gianelle, D. (2009). The role of spectral resolution and classifier complexity in the analysis of hyperspectral images of forest areas. *Remote Sensing of Environment*, *113*(11), 2345–2355. https://doi.org/10.1016/J.RSE.2009.06.013

De, S., Pirrone, D., Bovolo, F., Bruzzone, L., & Bhattacharya, A. (2017). A novel change detection framework based on deep learning for the analysis of multi-temporal polarimetric SAR images. In *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International* (pp. 5193–5196). IEEE.

Digitalglobe. (2014). *World View- 3 Design and Specifications*. https://doi.org/DS-WV3 Rev 01/13

DigitalGlobe. (2014). DigitalGlobe WorldView-3. Retrieved from http://www.digitalglobe.com/sites/default/files/DG_WorldView3_DS_2014.pdf

Dinuls, R., Erins, G., Lorencs, A., Mednieks, I., & Sinica-Sinavskis, J. (2012). Tree Species Identification in Mixed Baltic Forest Using LiDAR and Multispectral Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *5*(2), 594–603. https://doi.org/10.1109/JSTARS.2012.2196978

Féret, J.-B., & Asner, G. P. (2011). Spectroscopic classification of tropical forest species using radiative transfer modeling. *Remote Sensing of Environment*, *115*(9), 2415–2422. https://doi.org/https://doi.org/10.1016/j.rse.2011.05.004

Féret, J.-B., & Asner, G. P. (2013). Tree species discrimination in tropical forests using airborne imaging spectroscopy. *IEEE Transactions on Geoscience and Remote Sensing*, *51*(1), 73–84.

Feret, J. B., & Asner, G. P. (2013). Tree Species Discrimination in Tropical Forests Using Airborne Imaging Spectroscopy. *IEEE Transactions on Geoscience and Remote Sensing*, *51*(1), 73–84. https://doi.org/10.1109/TGRS.2012.2199323

Gibiansky, A. (2014). Fully Connected Neural Network Algorithms - Andrew Gibiansky. Retrieved April 23, 2015, from http://andrew.gibiansky.com/blog/machine-learning/fully-connected-neural-networks/

Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random forests for land cover classification. *Pattern Recognition Letters*, *27*(4), 294–300.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Heinzel, J., & Koch, B. (2012). Investigating multiple data sources for tree species classification in temperate forest and use for single tree delineation. *International Journal of Applied Earth Observation and Geoinformation*, *18*, 101–110. https://doi.org/https://doi.org/10.1016/j.jag.2012.01.025

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Le, Q. V, Coates, A., Prochnow, B., & Ng, A. Y. (2011). On Optimization Methods for Deep Learning. *Proceedings of The 28th International Conference on Machine Learning (ICML)*, 265–272.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. Retrieved from http://dx.doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Marmanis, D., Schindler, K., Wegner, J. D., Galliani, S., Datcu, M., & Stilla, U. (2018). Classification with an edge: improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, *135*, 158–172.

Mnih, V. (2013). Machine Learning for Aerial Image Labeling, 109.

Nogueira, K., Penatti, O. A. B., & dos Santos, J. A. (2017). Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification. *Pattern Recognition*, *61*, 539–556.

Scene Sharp Technologies Inc. (2012). fuze go™ MS Sharp – Scene Sharp. Retrieved April 17, 2018, from http://scenesharp.com/fuze-go-ms-sharp/

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–9).

Tian, T., Li, C., Xu, J., & Ma, J. (2018). Urban Area Detection in Very High Resolution Remote Sensing Images Using Deep Convolutional Neural Networks. *Sensors*, *18*(3), 904.

Trimble. (2017). eCognition. Retrieved May 17, 2018, from http://www.ecognition.com/

Yang, L., Zhang, X., Liang, S., Yao, Y., Jia, K., & Jia, A. (2018). Estimating Surface Downward Shortwave Radiation over China Based on the Gradient Boosting Decision Tree Method. *Remote Sensing* . https://doi.org/10.3390/rs10020185

Yu, X., Hyyppä, J., Litkey, P., Kaartinen, H., Vastaranta, M., & Holopainen, M. (2017). Single-Sensor Solution to Tree Species Classification Using Multispectral Airborne Laser Scanning. *Remote Sensing*, *9*(2), 108. https://doi.org/10.3390/rs9020108

Zhao, W., & Du, S. (2016). Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *113*, 155–165.

Zhen, Z., Quackenbush, L. J., & Zhang, L. (2016). Trends in Automatic Individual Tree Crown Detection and Delineation—Evolution of LiDAR Data. *Remote Sensing*, *8*(4).

5 Chapter 5: Summary and Conclusions

This chapter summarizes the research conducted for this dissertation. It also outlines the research conducted in Chapters 2 through 4, as well as the contributions of this research. Finally, it provides some suggestions for future work.

5.1 Summary of the research

This dissertation reviewed, examined, and improved state-of-the-art DL techniques for RS data analysis. DL is rooted in the ML field for object detection and classification and has just recently been applied to geoscience and the RS field. During this time, DL, specifically CNN, has had significant success in the RS field. Nevertheless, due to differences between ML and RS image datasets, DL's applications have mostly focused on urban areas and aerial-scene labeling. These differences are the limited available training data, large sizes of satellite images, the importance of spectral information in RS data, the higher number of spectral bands in RS images, and the importance of reusing trained networks on uncorrelated images in this field.

Considering these differences, new questions arise regarding the use of DL in RS image analysis. This dissertation focused on three aspects of these questions:

- It adopted a convolutional neural network for mapping wetland complexes that have spectrally similar classes. This application also faced a lack of training data. Therefore, the dissertation proposed that a pretrained network be used for initializing the weights of CNN to reduce the necessary training data. It also investigated the generality of the generated features for reuse on other datasets.

- It proposed a solution for feeding all spectral bands to CNN. It also proposed a robust solution for comparing different CNN networks with different scenarios, including fully training versus fine-tuning and using 3 bands versus all spectral bands. It also investigated the generality of the features.
- It proposed a method for organizing CNN to fit the network to the data. Based on that, it introduced a new network, based on VGG-16, for individual tree-species detection using Worldview-3 images. It has proposed the use of segments instead of pixels to speed up processing.

This research used, investigated, and improved deep CNNs for classification of the heterogeneous environment. Chapter 2 investigated CNN's capability for wetland classification, in particular examining the potential of a pre-existing CNN for mapping wetland complexes using RapidEye optical imagery in a study area located in the Avalon Peninsula, Newfoundland and Labrador. This chapter addresses the problems of extracting many hand-crafted features, the sensitivity of CNN to spectrally similar classes, lack of training data, and generality of the generated features. The CNN achieved an overall classification accuracy of 94.82%, demonstrating an improvement of about 16% compared to the RF classifier for all land-cover types. Moreover, an average improvement of about 30% was attained for wetland classes when CNN was employed. The latter observation suggests the significance of incorporating high-level spatial features into the classification scheme to reduce confusion between spectrally similar wetland classes.

Chapter 3 used CNN again for classification of complex wetlands. It investigated two different strategies of employing pre-existing CNN: full-training (in case of having

enough training data), and fine-tuning (in case of limited training data). The chapter also examined potential of the most well-known deep convnets currently employed for several computer vision tasks, including DenseNet121, InceptionV3, VGG16, VGG19, Xception, ResNet50, and InceptionResNetV2 in a comprehensive and elaborate framework using multispectral-RapidEye optical data for wetland classification. It also introduced a solution for using all spectral bands to feed CNN and compared the classification results with 3 and 5 spectral bands. The results illustrate that the full-training of pre-existing convnets using five bands is more accurate than both full training and finetuning using three bands, suggesting that the extra multispectral bands provide complementary information. In this study, InceptionResNetV2 consistently outperformed all other convnets for the classification of wetland and nonwetland classes with the state-of-the-art overall classification accuracy of about 96%, followed by ResNet50 and Xception, with respective accuracies of about 94% and 93. The impressive performance of InceptionResNetV2 suggests that an integration of Inception and ResNet modules is an effective architecture for complex land-cover mapping using multispectral remote-sensing images. The individual class accuracy illustrated that confusion occurred between wetland classes (herbaceous wetlands), although it was less pronounced when InceptionResNetV2, ResNet50, and Xception were employed. The swamp wetland had the lowest accuracy in all cases, potentially because the fewest training samples were available for this class.

Chapter 4 proposed a method to use single-source WorldView-3 satellite images to detect and identify individual tree species. It developed a two-step processing approach. The first step applies a segmentation to delineate tree crowns, which is followed by the

second step, wherein a deep CNN detects tree species. For the deep CNN, we designed a DITDN, based on VGG16, with the ability to receive all 8 visible and near-infrared spectral bands and trained it to detect four tree species (pine, spruce, red maple, and birch). This chapter addressed problems of network optimization, processing speed, and the generality of the designated network for ITSD. In this study, our DITDN outperformed the results of other investigated ML detection methods (RF and GB), reaching an accuracy of about 92.13%, whereas GB and RF reached 83.57% and 80.12%, respectively. We also applied the VGG16 network directly to measure the effect of optimization, which reached 87.58% accuracy. Our DITDN's accuracy suggests that network optimization in terms of depth, parameters, and optimization space, can generate a network that better fits the data and, thus, produces a more accurate detection map. In addition, the 89% detection accuracy obtained from the second test image's detection map demonstrates that the DITDN network's generated features are not dependent on the image the network trained with, demonstrating further that the network can be applied to other images and that its features are general enough to distinguish the same classes in new images taken in the same season.

5.2 Achievements of the research

1Based on Chapters 2 to 4 of this dissertation, the summary of overall contributions follows:

5.2.1 Adopting a CNN network for classifying heterogeneous wetland environments in high resolution satellite multispectral imagery

CNN is mostly sensitive to spatial information. Therefore, it has been used in the RS field mostly for classifying land covers or scenes with high discriminant spatial information. We adopt a CNN network that can incorporate the spectral and spatial information and discriminate spectrally similar classes.

5.2.2 Incorporating fine-tuning of different layers of a pre-trained CNN network to deal with limited training data

Since the training data available for training a classifier is limited, we examined different scenarios to use the weights of a pretrained network. Based on the training data, the type of images that the pretrained network was trained for, and the fact that the primitive layers generate more general features, the weights of specific primary layers can be used as the initializer for training the new network (fine-tuning). In this way, the needed training data will be reduced.

5.2.3 Comparing seven selected CNN networks in terms of the accuracy, training, and the number of used bands

Considering a heterogeneous environment with spectrally similar classes, we examined well-known CNN networks with different architectures. We compared the seven selected networks' accuracy, training, and number of used spectral bands to determine the best suitable structure for the task. This comparison can be a baseline for selecting a CNN network for classification and detection. It also can serve as a reference to design new CNN architecture for similar tasks.

5.2.4 Moving from the pixel-based approach to object-based to speed up the DL processing

The Patch-based CNN works as a pixel-based process. Since each pixel should be processed separately, and each process needs more processing time than the other pixel-based classification and detection methods, this method is considered slow. Grouping similar pixels into one can decrease processing time since it runs the method only once per segment. The segmentation can also reduce the effect of a noisy classification map and can preserve the shape of the object if under-segmentation is avoided in the segmentation.

5.2.5 Incorporating the multispectral resolution of the satellite images into CNN for classification

Recent DL libraries are designed to process RGB images, which limits the number of spectral bands a CNN network can receive and process. This limitation is crucial in the RS field since almost all multispectral images have more than three bands. We developed a pipeline that can read the GeoTiff format and connect it to the Tensorflow library. We also demonstrate the importance of using more bands by generating the classification maps for three-band and five-band situations and comparing their accuracies.

5.2.6 Optimizing a CNN network in order to design a new architecture to better fit the network to data

Most of the CNN networks are designed for image dataset that is used in ML. The network architecture, kernels, and optimization are designed for those data. Therefore, given new types of images, the network might not perfectly fit the data, which leads to

incomplete optimization. In other words, there would be room left to increase the accuracy of the trained network by optimizing its architecture. The optimization also can serve as a method for designing a new network for a specific task with particular types of image.

5.3 Suggestions for future works

Based on this research, this dissertation proposes the following suggestions: This research focused on applying CNN for classification of heterogeneous environments. We examined the sensitivity of CNN to spectral information. We found CNN is more sensitive to spatial information than spectral information, which is clearly shown by the difference in accuracy of the wetland and nonwetland classes. This fact can be related to the nature of convolutional layers. Thus, as the first suggestion, there is potentiality to design a specific layer for DL that can generate features that are more spectral-based than spatial-based, and then, incorporate them on higher levels. This idea is of more interest when hyperspectral images are being used.

Another limitation we worked on is the limited available training data. Although fine-tuning of a pretrained network is a solution, decreasing reliance on training data is preferable. Therefore, as the second suggestion, use of combined unsupervised-supervised DL methods might be desirable.

This dissertation focused just on multispectral satellite images. Theoretically, CNN can classify different types of images and fuse them at the class level. This is the third suggestion we are currently working on. Lastly, working on the visualization of the DL features is an open topic that can help us understand the mechanism of the DL algorithm.

Besides, since CNN networks are computationally time-consuming when run on a large area, it would be helpful to identify the most important features and remove the kernels associated with the least important features. In this way, the network can be compressed, so the implementation would be faster, and the network would have fewer parameters, so it would need less training data for fine-tuning.

Curriculum Vitae

Candidate's full name: Mohammad Rezaee

Universities attended:

- 2012: M.Sc., Remote Sensing, Department of Geomatics Engineering, University of Tehran, Iran
- 2009: B.Sc., Geomatics Engineering, Department of Geomatics Engineering, University of Tehran, Iran

Publications:

<u>Peer Reviewed Journal Papers</u>:

1. Jabari S., Rezaee M., Fathollahi F., Zhang Y. "Change Detection Using Multivariate Kullback-Leibler Distance", *ISPRS journal of photogrammetry and remote*, 2018
2. Rezaee, M., Mahdianpari, M., Zhang, Y., & Salehi, B. (2018). Deep Convolutional Neural Network for Complex Wetland Classification Using Optical Remote Sensing Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, (99).
3. Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., & Zhang, Y. (2018). Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. Remote Sensing, 10, 1119. (Feature paper)
4. Rezaee, M., Mahdianpari, M., Zhang, Y., & Salehi, B. (2018). Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using

Multispectral Remote Sensing Imagery. *Remote Sensing of Environment*. Under review

5. Rezaee, M., Tong, F., Mishra, R., Zhang, Y. (2018). Detection of Individual Tree Species Using an Optimized Deep CNN in an Object-Based Approach. *Remote Sensing of Environment*. Under review
6. Rezaee, M., Zhang Y. "Using Locality-Constrained Linear Coding in Automatic Target Detection of HSR Imageries", AIMS Geosciences, 2017.
7. Rezaee, M., Samadzadegan, F. and Homayouni, S. (2012). "Clustering Evaluation Using Worldview-2 Imagery in Urban Area" Journal of Geomatics Science and Technology. Vol. 1, No. 6

Conference Presentations:

1. Rezaee, M., Tong F., Zhang Y., Mishra, R., Tong, H., "Using a VGG-16 Network for Individual Tree Species Detection with an Object-Based Approach". 10th International Workshop on Pattern Recognition in Remote Sensing (PRRS 2018)
2. Rezaee, M., Zhang Y. "Complex Wetland Classification in Optical Remote Sensing Imagery Using Deep Convolutional Neural Network". Geoscience and Remote Sensing Symposium (IGARSS), 2018 IEEE International
3. Rezaee, M., Zhang Y. "Detecting Road and Building using a fully convolutional network in aerial images", The Imaging and Geospatial Information Society (IGTF) 2017, ASPRS

4. Rezaee, M., Zhang Y. “Road and Building detection using a patch-based deep network for aerial images”, The Imaging and Geospatial Information Society (IGTF) 2017, ASPRS
5. Rezaee, M., Zhang Y. “Road Detection Using Deep Neural Network In High Spatial Resolution Images”, Joint Urban Remote Sensing Event (JURSE) 2017, IEEE
6. Rezaee, M., Zhang Y. “Processing Large Scale Data for Urban Road Detection with Deep Networks”, Geomatics Atlantic, 2016
7. Rezaee, M., Mirikharaji, Z., Zhang, Y. “Using Locality-constrained Linear Coding in Automatic Target Detection of HRS Imageries”. International symposium on digital earth, 2015.
8. Rezaee, M., Abouhamzeh, A., Zhang Y. “A heuristic land-cover based image enhancement for satellite imageries”. Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International