Spring 3-26-2021

Large Area Land Cover Mapping Using Deep Neural Networks and Landsat Time-Series Observations

Shahriar Shah Heydari
SUNY College of Environmental Science and Forestry, sshahhey@esf.edu

LARGE AREA LAND COVER MAPPING USING DEEP NEURAL NETWORKS AND LANDSAT TIME-SERIES OBSERVATIONS

by

Shahriar Shah Heydari

A dissertation
submitted in partial fulfillment
of the requirements for the
Doctor of Philosophy Degree
State University of New York
College of Environmental Science and Forestry
Syracuse, New York
March 2021

Department of Environmental Resources Engineering

Approved by:
Giorgos Mountrakis, Major Professor
Eddie Bevilacqua, Chair, Examining Committee
Lindi J. Quackenbush, Department Chair
S. Scott Shannon, Dean, The Graduate School
Gary Scott, Director, Division of Engineering

ACKNOWLEDGEMENTS

I had a great time during my PhD studies in Syracuse. It enabled me to work on some of the most interesting scientific challenges of my field and helped me acquire great knowledge and skills to setup my future career. It was also a very joyful and pleasant life experience living in beautiful Central New York and visiting many other places nearby and in other US states and learning a lot. I am grateful for this opportunity to many people.

It was not possible without having the offer from my professor, Giorgos Mountrakis. His full support in every aspect of our collaboration from scientific advice to spending days and nights to handle our simulations coordination on NASA high-end computers, papers co-authoring and publishing support, and unique long-term financial support was far more than my expectation. I am truly grateful to him.

I also present my special thanks to the team of colleagues who helped creating our huge dataset needed for my research: Atef A. Amriche, Harrison Goldspiel, Rabia Munsaf Khan, Meghan K Medwid, Madusha Sammani, and Zhixin Wang.

I am thankful to my committee and particularly Lindi Quackenbush and Steve Stehman for their helps and very positive attitude in our discussions. I am also thankful to all my officemates, department officials (with special thanks to Mark A. Storrings), Iranian and non-Iranian friends in SUNY ESF or Syracuse University, people in my living neighborhoods and people I met during my numerous travels, and my social media friends. I am sorry for not being able to name everybody but I have all of you in mind at the time of writing this.

Behind every moment for everybody, there is always unconditional love and support from his/her parents and close family. Any success and fortune that I have achieved is because they are there. Thank you.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Sh. Shah Heydari. Large-Area Land Cover Mapping Using Deep Neural Networks And Landsat Time-Series Observations, 186 pages, 19 tables, 27 figures, 2021. Chicago style guide used.

This dissertation focuses on analysis and implementation of deep learning methodologies in the field of remote sensing to enhance land cover classification accuracy, which has important applications in many areas of environmental planning and natural resources management.

The first manuscript conducted a land cover analysis on 26 Landsat scenes in the United States by considering six classifier variants. An extensive grid search was conducted to optimize classifier parameters using only the spectral components of each pixel. Results showed no gain in using deep networks by using only spectral components over conventional classifiers, possibly due to the small reference sample size and richness of features. The effect of changing training data size, class distribution, or scene heterogeneity were also studied and we found all of them having significant effect on classifier accuracy.

The second manuscript reviewed 103 research papers on the application of deep learning methodologies in remote sensing, with emphasis on per-pixel classification of mono-temporal data and utilizing spectral and spatial data dimensions. A meta-analysis quantified deep network architecture improvement over selected convolutional classifiers. The effect of network size, learning methodology, input data dimensionality and training data size were also studied, with deep models providing enhanced performance over conventional one using spectral and spatial data. The analysis found that input dataset was a major limitation and available datasets have already been utilized to their maximum capacity.

The third manuscript described the steps to build the full environment for dataset generation based on Landsat time-series data using spectral, spatial, and temporal information available for each pixel. A large dataset containing one sample block from each of 84 ecoregions in the conterminous United States (CONUS) was created and then processed by a hybrid convolutional+recurrent deep network, and the network structure was optimized with thousands of simulations. The developed model achieved an overall accuracy of 98% on the test dataset. Also, the model was evaluated for its overall and per-class performance under different conditions, including individual blocks, individual or combined Landsat sensors, and different sequence lengths. The analysis found that although the deep model performance per each block is superior to other candidates, the per block performance still varies considerably from block to block. This suggests extending the work by model fine-tuning for local areas. The analysis also found that including more time stamps or combining different Landsat sensor observations in the model input significantly enhances the model performance.

Keywords: land-cover mapping, remote sensing, per-pixel classification, deep neural network, recurrent network, convolutional network, Long Short-Term Memory (LSTM), Grey-Level Co-occurrence Matrix (GLCM), map accuracy assessment

Sh. Shah Heydari
Candidate for the degree of Doctor of Philosophy, March 2021
Giorgos Mountrakis, Ph.D.
Department of Environmental Resources Engineering
State University of New York College of Environmental Science and Forestry,
Syracuse, NY

CHAPTER 1: DISSERTATION INTRODUCTION

1.1. Background and motivation

1.1.1. Land cover mapping and the application of machine learning

Land cover mapping is the process by which a thematic map is generated to delineate features of interest on the ground, typically georeferenced in a spatial coordination system. This task is a foundation for many applications including land and agriculture planning, forestry and wildlife habitat monitoring, and environmental impact evaluation (Vogelmann et al. 2001). It is also the basis for some second-level analysis such as land use mapping and land cover/land use change analysis (LCLUC), and it has been increasingly used as a basic tool for studies on subjects such as climate change and conservation planning, particularly when it requires continuous evaluation (Fry et al. 2011).

Both land cover data collection and processing can be very difficult and time consuming. The use of aerial imagery or drones may be appropriate for a limited local area, but use of satellite imagery quickly becomes the only viable option for increased mapping footprint. The remotely sensed data is typically in the form of raster imagery in multiple electromagnetic bands. But this basic data is commonly complemented with other maps or tabular data to enhance the mapping quality. These additional data may be driven by remotely sensed information such as climate and topographic data or night-time light (for example see Yu et al., 2013; Wulder et al., 2018), or administrative procedures such as natural resources inventory maps or population data (Vogelmann et al. 2001). The mapping process, i.e. converting image and other tabular data to thematic maps, will then be a combination of image processing and data mining tasks with all geoprocessing and photogrammetry considerations in place (for good examples see National

Land cover Dataset – NLCD – reference articles by Vogelmann et al., 2001; Homer et al., 2007; Fry et al., 2011). Launch of the first Landsat satellite in the early 1970s and subsequent missions provided a breakthrough technology that brought a new era of possibilities for data availability. Map generation and image classification was one of the main tasks applied to this data.

As shown in Landsat-based timeline of Figure 1-1, different generations of image classification methods have been practiced over time, from early visual inspection to incorporating machine learning methodologies by unsupervised and supervised classification. Classification was enhanced over time to include advanced methods with data fusions, object-based analysis, and advanced classifiers. The remote sensing field has adopted methods available in computer vision and produced machine learning libraries to deploy powerful algorithm to mine ever increasing available remote sensing data and automatically generate the most accurate maps.

It is needed in many cases to have land cover classification with subtypes of major types (e.g., different types of forest such as coniferous, deciduous, or mixed), with the finest possible spatial resolution, and with greatest accuracy. But all these qualities are limited in practice, especially if we want to go global. First of all, data sources with spatial resolution below 10 meters are still scarce or expensive for global or even regional applications. Depending on the type of investigated land covers, we may have the problem of mixed pixels (one pixel having different land covers types existing in its area) at Landsat 30 meters resolution or higher. There are also many rules and mapping preferences that we may want to impose such as class or feature-specific preferences, and a simple image processing will fail to apply it automatically. Even without giving consideration to mixed pixels, we still have many design and implementation issues with land cover class definition. For example, separating similar land cover types such as different densities of urbanism or similar crops is not trivial. We also have class boundaries that are user-defined, administrative, or dynamic for some classes, for example developed land intensity levels, road network, or wetlands. All these issues demand better fusion of remote sensing and administrative information and more powerful data mining algorithms for pixel and object analysis, for which the deep learning is currently the most promising methodology.

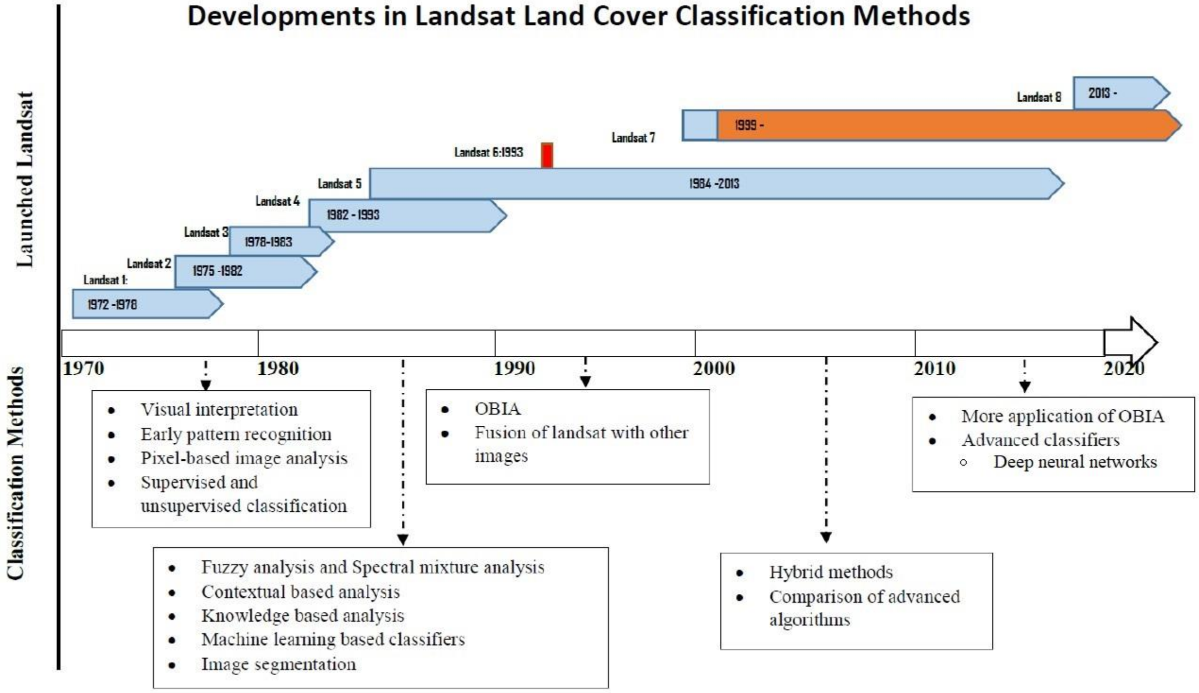


Figure 1-1: Landsat mission launches and land cover classification development over time, adapted from Phiri and Morgenroth (2017). Orange/red bar indicates failure in launching or malfunction in operation. OBIA stands for Object-Based Image Analysis.

Some of the most important and widely used advanced classifiers nowadays are deep networks, which are complex multilayer neural networks that aim to look deeply into feature structures and relationships and extract useful hidden features from input data in successive layers, from simple geographical features (boundaries, corners, etc.) to feature groups and eventually to objects (face components, buildings, etc.). Once the relationships/objects have been hierarchically analyzed, their classification will be potentially easy. Deep learning has been used for more than a decade in many applications such as computer vision, speech recognition, and natural language processing. These approaches have also found their way into the remote sensing area as well, and especially since 2015 we have seen a surge of related papers (X. X. Zhu et al. 2017a). Within various applications of deep learning in remote sensing, scene and pixel classification are the most studied applications and land cover mapping directly falls under this category. This dissertation discusses various methodologies for deep learning application in land cover mapping. The main aim of deep learning methodologies is to automatically generate rich features from raw input data, hence increase final classification performance. Deep learning architectures may look at different dimensions of the data – spectral, spatial, and temporal – and our aim in this research is to move a step forward to employ all these dimensions in a hybrid architecture combining different network types such as standard multilayer, convolutional, and recurrent neural networks.

Feature generation is very important both for spatial dimension, which is the natural focus of computer vision as well, and spectral dimension, which is of particular interest when dealing with hyperspectral imagery with hundreds of spectral bands. Different network types or their combination are employed for hyperspectral image processing such as Mou et al. (2017) or Lakhal et al. (2018). Extraction of features by deep networks is even extended beyond direct use of raw remote sensing values to modeling spectral curves such as the work by Kim et al. (2018) or Lee et al. (2020) which employed convolutional neural network.

In addition to the above, deep learning is being used more and more in applications such as change detection, for example Lyu et al. (2016) or Song and Choi (2020) used two time stamps and recurrent neural network for identifying the land cover change. Deep learning has also been used in forecasting, for which MODIS (Moderate Resolution Imaging Spectroradiometer) data has a special position because of its better temporal resolution than Landsat. Although it has lower spatial resolution than Landsat, they have been used in combination to predict some parameters such as vegetation index (Kong et al., 2018) or regress crop yields (Jiang et al., 2020). In other approaches, Cao et al. (2019) used land cover maps (no remote sensing data directly) to forecast the next period of land cover, and Mu et al. (2019) used Landsat imagery first to determine land cover, then added regional economic, climate, and construction data to it and used a recurrent network to predict the land use in the next years.

1.1.2. Identified research gaps

Phiri and Morgenroth (2017) provide a general discussion of the continuous effort to enhance the mapping quality by applying new methods and technologies. Alhassan et al. (2020) focuses on the application of deep learning and mostly convolutional networks in land cover mapping. Although there is substantial research on application of deep networks in local land cover classification, there is a gap of research to apply this methodology to large area and global land cover mapping. This is a particularly important application because the accuracy of global land cover mapping is still unsatisfactory and there is high hope that deep learning can help to boost this performance.

There are, however, some obstacles ahead. One of the big obstacles in application of deep learning methods is their need for large datasets to reveal their potential for advanced data mining. This is because the deep networks are generally complex and have many parameters in their structure that have to be optimized by training on a dataset of appropriate size. This is really a big issue because most of the prior research on application of deep learning in land cover mapping has been done on limited areas and to the best of our knowledge, there is no global or regional land cover mapping application of deep neural network surpassing traditional methods in achieved accuracy. So, our work is two-fold: generate reference datasets (as an input product) and develop model (the research work as well as end product). We need to focus effort on dataset generation to support model training and testing. The NLCD (National Land Cover Database) reference labels may be a good starting point for US-wide land cover data, but because it is an end product of a classification effort and not matched to ground truth in every point, it is better to be verified manually for those points that are going to be used as the reference labels for another classification. We also need to verify the stability of land cover for reference points over the analysis time period.

The main idea and promise of deep learning methodology is about automatic generation of powerful features from input data to achieve better results than traditional methods. However, as shown in chapter 2 of this dissertation, simple deep networks based on just spectral information may not outperform traditional classifiers. Except for change detection or forecasting applications, where the time stamp matters, the prior research on land cover classification has been focused mostly on spectral and spatial dimensions, particularly through convolutional networks and custom-designed modules and optimizing algorithms, and the network design based on the use of all data dimensions is still a relatively untouched area. One of the reasons is that convolutional networks has been studied extensively for computer vision and big models has been trained and are already available. Further, convolutional networks have been found to be powerful in automatic feature extraction in image processing applications. Deep networks in general have a very flexible structure and there are almost unlimited possibilities to setup the layers, neurons, connections, and network options. However, this flexibility makes it very difficult to find a roadmap in designing and optimizing them. As discussed in chapter 3 of this dissertation, the literature reviews provide mostly general information and do not provide detailed description about and comparison between many new emerging approaches and network enhancements, and they do not provide a quantitative comparison.

Employing temporal dimension along with the others and expanding the work to a global extent is of our special interest in this research. We aim to try different network and experiment design parameters – especially different methods of feature generation – and find the best combination to get the highest possible accuracy.

1.2. Research objectives and questions

The proposed research focuses on enhancing global land cover mapping based on Landsat data. To this end, we worked on the three objectives and their related research questions.

1) Our first objective was to evaluate and compare a deep classifier to some widely used traditional classifiers using spectral data only. It is also of interest to identify their best parameter settings and scene-specific statistics. We studied SVM, Decision Tree ensemble (a close cousin of Random Forest classifier), K-Nearest Neighbor (KNN), Naïve Bayesian (a Maximum-Likelihood type classifier), conventional multilayer Neural Network, and a specific implementation of deep neural networks. This step answered these questions:
    a. What is the classification accuracy of the above methods on Landsat spectral data?
    b. How to choose model and design parameters to reach the best performance?
    c. What are the effects of training sample size, class distribution, and scene complexity on performance?

Answering the above questions provides a good insight into the Landsat spectral-based land cover classification performance and the study of other parameters and scene dependency.

2) Aiming to deploy a state-of-the-art deep architecture in our work, the next objective was to do an extensive literature review on deep learning approaches in remote sensing image classification and do a meta-analysis of the findings. We want to know:

   a. Which techniques have been used in the past and how frequently?

   b. Is there any significant improvement in applying deep learning methods compared to conventional non-deep classifiers?

   c. Which deep architecture gives better results?

The literature review included a quantitative comparison between deep learning and conventional classification methods. This analysis is still very general because of the different datasets and design parameters and hyperparameters applied in each study.

3) The main goal of our research was to investigate the potential of combining different data dimensions, particularly the less frequently applied temporal dimension. To this end, we concentrated on recurrent neural networks, which are inherently fit to analyze time-series data, and constructed a hybrid design to process combined spectral-spatial-temporal Landsat data to generate high accuracy land cover maps in any ecoregion in the conterminous United States. We want to know:

   a. Do deep recurrent network improve classification accuracy compared to non-recurrent neural networks and non-deep conventional classifiers?

   b. Which network configuration and feature selection generate the best result?

   c. What is the performance of the best model on individual blocks and its variation?

   d. Considering use of temporal data dimension is this part of work, what is the model’s response to limited input scenarios (e.g., just one Landsat sensor or limited number of available scenes per year)?

The last question is of special interest from a practical point of view because if the model requires high temporal resolution, its application will be limited to locations with very low cloud cover through the year and also it cannot be used until near the end of the current year.

Based on those objectives and questions, our research hypotheses are:

Hypothesis 1: Deep networks do not provide practical improvement in classification performance over conventional classifiers when dataset is small compared to number of network parameters or only spectral data dimension is used.

Hypothesis 2: Our proposed network architecture for processing temporal-spectral-spatial Landsat data can achieve better accuracy than its companion spectral-spatial or spectral-only variants, and performs better than currently available global land cover products (over conterminous US).

Hypothesis 3: There is considerable improvement in fusion of different Landsat sensors in terms of achieved accuracy and minimum number of requires scenes.

1.3. Dissertation organization and chapters overview

This dissertation is organized in three main chapters after this introduction, each focused on one of the research objectives. After that, we revisit the research questions and hypotheses in the conclusion and provide some insights on future work. Here we briefly introduce the contents and methodology used in each chapter to give the reader a better overall picture of this work.

1.3.1. Chapter 2: Effect of classifier selection, reference sample size, reference class distribution and scene heterogeneity in per-pixel classification accuracy using 26 Landsat sites

For this part, we trained and tested various popular classifiers on a dataset used in a prior research that was based on the USGS Land cover Trends project data (Khatami et al. 2017). This dataset consists of sample 10 km ×10 km areas from 26 selected Landsat scenes along with their associated land cover maps, taken from different states within conterminous US. We included five conventional and one example deep classifier and evaluate performance of each classifier on each block. Each classifier had a set of hyperparameters that we tested their different values in a wide range of settings to tune the model parameters. We also tried changing some other simulation parameters including training data sampling rate and class distribution. This part of our work was published as Heydari and Mountrakis (2018).

1.3.2. Chapter 3: Deep learning in remote sensing: Review and meta-analysis of mono-temporal image classification methods

This part of our research was based on an extensive literature review focused on papers dealing with application of deep learning techniques on per-pixel classification of remote sensing data. It was accompanied by a meta-analysis (an overarching analysis of the results obtained by some other existing analyses on a common topic) to build comparative quantitative results on those cases who compare a popular conventional non-deep classifier to a deep network implementation (we found SVM as the most common basis for our comparison). Such an analysis could give insight into the pros and cons of different deep learning architectures, for which there was no quantitative analysis published. We discussed different design ideas and network configurations focusing on mono-temporal image classification research, and this work was also published (Heydari and Mountrakis 2019).

1.3.3. Chapter 4: Large area land cover mapping using deep neural networks and Landsat time-series observations

This chapter presented the main body and goal of our research. Here we first discussed our work on generating a big feature-rich training dataset of Landsat observations over the conterminous United States. Then, we explained developing a deep recurrent network to extract useful spectral-spatial-temporal patterns in pixel's data and classify them with a high level of accuracy to answer research questions in this section. Each sample land cover map was a 10 km ×10 km block at 30m resolution, carefully checked using Google Earth high resolution imagery to assign proper land cover to its pixels, and different data blocks were aggregated to feed our proposed network models.

We explained how a complex feature-rich time sequence is extracted for each pixel, comprising Landsat band values, additional spectral indices, topography, texture and spatial information, and climate variables. We then reported on the result of our experiments to find the best combination of input features and tune the network structural and other hyperparameters to obtain the best classification performance. We also studied the network performance on individual blocks and under various scenarios of limited sensor or available time stamps, plus visual study and interpretation of results for selected blocks. The corresponding paper for this chapter is in final drafting stages and will be submitted for publication soon.

1.4. Intended Audience

Land cover mapping has a lot of applications in all levels of public and private administration and resources management, therefore a better model for doing this task will be of interest to a diverse audience. Our work was focused on large area CONUS-wide land cover mapping, for which some big organizations such as USGS and NASA are involved, and the results of our work may be of their interest to enhance their models. Because the model is based on public Landsat data, it can be easily used by any other researcher anywhere in the world to use it directly or via fine-tuning and model transfer to study their own area of interest. Any land cover application – particularly at global scope – has a very immediate application in natural resource management and monitoring and change detection, for which the base model accuracy is paramount. Our developed architecture can be applied to those applications directly or with some change. Examples would include forest management, wetland inventory, urban development studies, crop analysis, and climate change and its effects on natural resources.

References

Alhassan, Victor, Christopher Henry, Sheela Ramanna, and Christopher Storie. 2020. “A Deep Learning Framework for Land-Use/Land-Cover Mapping and Analysis Using Multispectral Satellite Imagery.” *Neural Computing and Applications* 32 (12): 8529–44. https://doi.org/abdi.

Cao, Cong, Suzana Dragićević, and Songnian Li. 2019. “Short-Term Forecasting of Land Use Change Using Recurrent Neural Network Models.” *Sustainability* 11 (19): 5376. https://doi.org/10.3390/su11195376.

Fry, Joyce, George Z. Xian, Suming Jin, Jon Dewitz, Collin G. Homer, Limin Yang, Christopher A. Barnes, N.D. Herold, and J.D. Wickham. 2011. “Completion of the 2006 National Land Cover Database for the Conterminous United States.” *Photogrammetric Engineering and Remote Sensing* 77 (9): 858–64.

Heydari, Shahriar S., and Giorgos Mountrakis. 2018. “Effect of Classifier Selection, Reference Sample Size, Reference Class Distribution and Scene Heterogeneity in per-Pixel Classification Accuracy Using 26 Landsat Sites.” *Remote Sensing of Environment* 204 (January): 648–58. https://doi.org/10.1016/j.rse.2017.09.035.

———. 2019. “Meta-Analysis of Deep Neural Networks in Remote Sensing: A Comparative Study of Mono-Temporal Classification to Support Vector Machines.” *ISPRS Journal of Photogrammetry and Remote Sensing* 152 (June): 192–210. https://doi.org/10.1016/j.isprsjprs.2019.04.016.

Homer, Collin, Jon Dewitz, Joyce Fry, Michael Coan, Nazmul Hossain, Charles Larson, Alexa Mckerrow, J. VanDriel, and James Wickham. 2007. “Completion of the 2001 National Land Cover Database for the Conterminous United States.” *Photogrammetric Engineering and Remote Sensing* 73 (April).

Jiang, Hao, Hao Hu, Renhai Zhong, Jinfan Xu, Jialu Xu, Jingfeng Huang, Shaowen Wang, Yibin Ying, and Tao Lin. 2020. “A Deep Learning Approach to Conflating Heterogeneous Geospatial Data for Corn Yield Estimation: A Case Study of the US Corn Belt at the County Level.” *Global Change Biology* 26 (3): 1754–66. https://doi.org/10.1111/gcb.14885.

Khatami, Reza, Giorgos Mountrakis, and Stephen V. Stehman. 2017. “Mapping Per-Pixel Predicted Accuracy of Classified Remote Sensing Images.” *Remote Sensing of Environment* 191 (March): 156–67. https://doi.org/10.1016/j.rse.2017.01.025.

Kim, Miae, Junghee Lee, Daehyeon Han, Minso Shin, Jungho Im, Junghye Lee, Lindi J. Quackenbush, and Zhu Gu. 2018. “Convolutional Neural Network-Based Land Cover Classification Using 2-D Spectral Reflectance Curve Graphs With Multitemporal Satellite Imagery.” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11 (12): 4604–17. https://doi.org/10.1109/JSTARS.2018.2880783.

Kong, Yun-Long, Qingqing Huang, Chengyi Wang, Jingbo Chen, Jiansheng Chen, and Dongxu He. 2018. “Long Short-Term Memory Neural Networks for Online Disturbance Detection in Satellite Image Time Series.” *Remote Sensing* 10 (3): 452. https://doi.org/10.3390/rs10030452.

Lakhal, Mohamed Ilyes, Hakan Çevikalp, Sergio Escalera, and Ferda Ofli. 2018. “Recurrent Neural Networks for Remote Sensing Image Classification.” *IET Computer Vision* 12 (7): 1040–45. https://doi.org/10.1049/iet-cvi.2017.0420.

Lee, Junghee, Daehyeon Han, Minso Shin, Jungho Im, Junghye Lee, and Lindi J. Quackenbush. 2020. “Different Spectral Domain Transformation for Land Cover Classification Using Convolutional Neural Networks with Multi-Temporal Satellite Imagery.” *Remote Sensing* 12 (7): 1097. https://doi.org/10.3390/rs12071097.

Lyu, Haobo, Hui Lu, and Lichao Mou. 2016. “Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection.” *Remote Sensing* 8 (6): 506. https://doi.org/10.3390/rs8060506.

Mou, Lichao, Pedram Ghamisi, and Xiao Xiang Zhu. 2017. “Deep Recurrent Neural Networks for Hyperspectral Image Classification.” *IEEE Transactions on Geoscience and Remote Sensing* 55 (7): 3639–55. https://doi.org/10.1109/TGRS.2016.2636241.

Mu, Lin, Lizhe Wang, Yuewei Wang, Xiaodao Chen, and Wei Han. 2019. “Urban Land Use and Land Cover Change Prediction via Self-Adaptive Cellular Based Deep Learning With Multisourced Data.” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12 (12): 5233–47. https://doi.org/10.1109/JSTARS.2019.2956318.

Phiri, Darius, and Justin Morgenroth. 2017. “Developments in Landsat Land Cover Classification Methods: A Review.” *Remote Sensing* 9 (9): 967. https://doi.org/10.3390/rs9090967.

Song, Ahram, and Jaewan Choi. 2020. “Fully Convolutional Networks with Multiscale 3D Filters and Transfer Learning for Change Detection in High Spatial Resolution Satellite Images.” *Remote Sensing* 12 (5): 799. https://doi.org/10.3390/rs12050799.

Vogelmann, Jim, S. Howard, Limin Yang, C. Larson, Bruce Wylie, and N Driel. 2001. “Completion of the 1990s National Land Cover Data Set for the Conterminous United States From LandSat Thematic Mapper Data and Ancillary Data Sources.” *Photogrammetric Engineering and Remote Sensing* 67 (June): 650–55. https://doi.org/10.1007/978-94-011-4976-1_32.

Wulder, Michael A., Nicholas C. Coops, David P. Roy, Joanne C. White, and Txomin Hermosilla. 2018. “Land Cover 2.0.” *International Journal of Remote Sensing* 39 (12): 4254–84. https://doi.org/10.1080/01431161.2018.1452075.

Yu, Le, Jie Wang, and Peng Gong. 2013. “Improving 30 m Global Land-Cover Map FROM-GLC with Time Series MODIS and Auxiliary Data Sets: A Segmentation-Based Approach.” *International Journal of Remote Sensing* 34 (16): 5851–67. https://doi.org/10.1080/01431161.2013.798055.

Zhu, Xiao Xiang, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. 2017. “Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources.” *IEEE Geoscience and Remote Sensing Magazine* 5 (4): 8–36. https://doi.org/10.1109/MGRS.2017.2762307.

CHAPTER 2 (MANUSCRIPT 1):

Effect of classifier selection, reference sample size, reference class distribution and scene heterogeneity in per-pixel classification accuracy using 26 Landsat sites

Abstract

Land cover mapping is an important and widely used practice that is typically done by converting aerial or satellite imagery to thematic maps. This task is typically done through classification and therefore a major issue in land cover mapping is classifier selection. In this study we investigated classifier performance under different sample sizes, reference class distribution, and scene complexities for twenty six 10 km ×10 km blocks with complete reference information across the continental US. Per-pixel classification was done using the six spectral bands from Landsat imagery. The tested classifiers included Naïve Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Bootstrap-aggregation ensemble of decision trees (BagTE), artificial neural network (ANN) up to 2 hidden layers, and deep neural network (DNN) up to 3 hidden layers. Our accuracy assessment conducted on full blocks extent indicated that all classifiers, with the exception of NB (a Maximum Likelihood variant), performed similarly. However, when we concentrated on the edge pixels –defined as the pixels at the border of adjacent land cover classes- it was clear that the SVM and KNN offer considerable accuracy advantages, especially for larger reference datasets. Coupled with their relatively low execution times we would recommend them for classifications using Landsat’s spectral inputs and Anderson’s 11-level classification scheme. Caution should be exercised though as primarily the SVM and secondarily the KNN demonstrated substantial accuracy

degradation during the parameter grid search, therefore an exhaustive parameter optimization process is suggested. While the ANN and DNN variants did not perform as well, their performance may have been restricted by the lack of rich contextual information in our simple six band per-pixel relatively small input spaces. The effect of class distribution in the training dataset was also evident on the calculated accuracy metric. We also observed gradual accuracy degradation as the edge pixel presence increased. Future work could focus on data-rich classification problems such as change detection using Landsat stacks or expand in high spectral or spatial resolution sensors.

2.1. Introduction

Classification of remotely sensed data is essential in generating thematic maps, which have many applications in environmental management, agricultural planning, health studies, climate and biodiversity monitoring, and land change detection (Khatami et al. 2016). A wide range of regional and global datasets are currently available facilitating studies at unprecedented scales (Grekousis et al. 2015). The classification process, in general, is composed of different tasks, from the selection of data source and sampling design, to classification method selection and classifier performance evaluation (Lu and Weng 2007). Although all of these tasks are important and their successful implementation is dependent on each other, a core task is the selection of the suitable classification method.

Each classification method may be more suitable for a specific target objective, problem condition, or imaging details (see table 1 in Lu and Weng, 2007). The classifiers performance assessment is also highly dependent on data quality, data values distribution, and sampling design (Jin et al. 2014; Li et al. 2014); and it can also be evaluated under various criteria like accuracy, reproducibility and/or robustness (Cihlar et al. 1998). Even for the most widely used assessment criteria, classification accuracy, there are important concerns that limit the ability to properly assess the accuracy of resulting map (see Foody, 2002, for a review). This line of research has been followed by more recent papers discussing the problems arising from increasing accuracy degradation over time in temporal land cover analysis and change detection (Foody, 2010), or stressing the importance of sample size or statistical hypothesis testing when comparing different classifiers or scenarios performance (Foody, 2009).

Therefore, it is difficult to generate a general statement to advise on classifiers ranking and one should always clarify the specific conditions that the classifier performance assessment is based on it. There are good review papers that introduce the classifiers in general and discuss their application conditions, strengths and weaknesses (Lu and Weng, 2007; Li et al. 2014), but they are mostly qualitative without specific quantitative results for best attainable classifiers accuracy. There are also works that go more in depth discussing classifiers for certain problem types. For example, see Weng (2012) for a discussion on classifiers for mapping of impervious surfaces, Mallinis and Koutsias (2012) for a comparison of ten classifiers for burned area mapping, J. He et al. (2015), for comparing four main classifiers in generation of arctic geological maps, or Pelletier et al. (2016) for assessing the robustness of random forest (RF) classifier for a specific area. Another research category is to review the application of a specific classifier in more detail. For example, see Mountrakis et al. (2011) for a review of SVM classifiers; Pal and Mather (2003), for an assessment of decision tree methods for land use classification; or Belgiu and Drăguţ (2016), for an overview of random forest classifier. Additional processing is also of interest, such as making ensemble of classifiers (X. Li et al. 2014), controlling of misclassification by post-processing (Martinez and Baerenklau, 2015), or using ancillary data to aid in classification by field visits (Meddens et al. 2016) or other sources and sensors (Z. Zhu et al. 2016). Based on numerous case studies, one can perform a meta-analysis of previously researched cases and assess the comparative results of case studies at a higher level. This meta-analysis has been done for a single type of classifier such as KNN (Chirici et al. 2016), or more general including pairwise comparisons among many classifiers (Khatami et al. 2016).

While fragmented comparisons between traditional classifiers can be found in existing literature, they are limited in terms of: i) number of case studies incorporated, ii) the search space of the classifier parameters (often resorting to default values), and iii) absence of a promising new classification family based on deep neural networks (DNN). To the best of our knowledge, there are just a few studies that investigated per-pixel classifier accuracy performance over multiple case studies or over a large area. For example, Ballantine et al. (2005) performed mapping for continental North Africa using MODIS data but comparisons were restricted to a few classifiers. In Gong et al. (2013) a global sampling and classification has been done using four different classifiers, but they used a fixed set of parameters for each classifier. Similarly, Lawrence and Moran (2015) tested classification accuracy for multiple classifiers for 30 data sets but they used a fixed set of classifier parameters that may not allow classifiers to reach their best potential. Pelletier et al. (2016) did a grid search on classifier parameters over two large areas in France, focusing on SVM and Random Forest classifiers. Finally, W. Li et al., (2016) employed most important classifiers plus the new autoencoder-based DNN implementations over one composite set sampled through the entire Africa, but they only reported a fixed parameter set (except for DNN).

Therefore, it is important to fill this research gap overcoming the three aforementioned limitations. Our research goals were: i) to compare classifiers' best achievable accuracy, ii) identify the accuracy costs associated with the reduction of the parameter grid and training dataset size and iii) investigate how data distribution and landscape heterogeneity influences classifier performance. We tested six different classifiers in our research: Naïve Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Bootstrap-aggregation ensemble of decision trees (BagTE), artificial neural network (ANN) up to 2 hidden layers, and autoencoder-based deep neural network (DNN) up to 3 hidden layers. We used a dataset of 26 Landsat images for classifiers comparison, and run each classifier with a grid of parameter settings to evaluate its performance.

2.2. Study area

This study uses the same input data reported in Khatami et al. (2017). The data was based on a set of 26 Landsat images (blocks), each covering a 10 km ×10 km area at 30m spatial resolution and represented by a matrix of 333×333 6-band pixel values. It was also accompanied with the entire block reference data on land cover classes for the complete 26 blocks. This set was part of a larger work maintained by US Geological Survey under the Land Cover Trends program. Reference data was created with the help of aerial photography, and over 33,000 geographically referenced field photos with associated keywords capturing existing land cover (the field photo map is available at http://landcovertrends.usgs.gov/fieldphotomap/map.html). Land cover types were represented by a single value for each pixel and coded in 11 different classes according to modified Anderson scheme (Anderson et al. 1976). The selected blocks covered a range of climate and topographic conditions throughout the continental US, and all had the same spatial resolution of 30m (Figure 2-1). The incorporated Landsat images reflected the same acquisition years of the high resolution data. Land cover class composition for every block is provided in Appendix A, Table A-1.

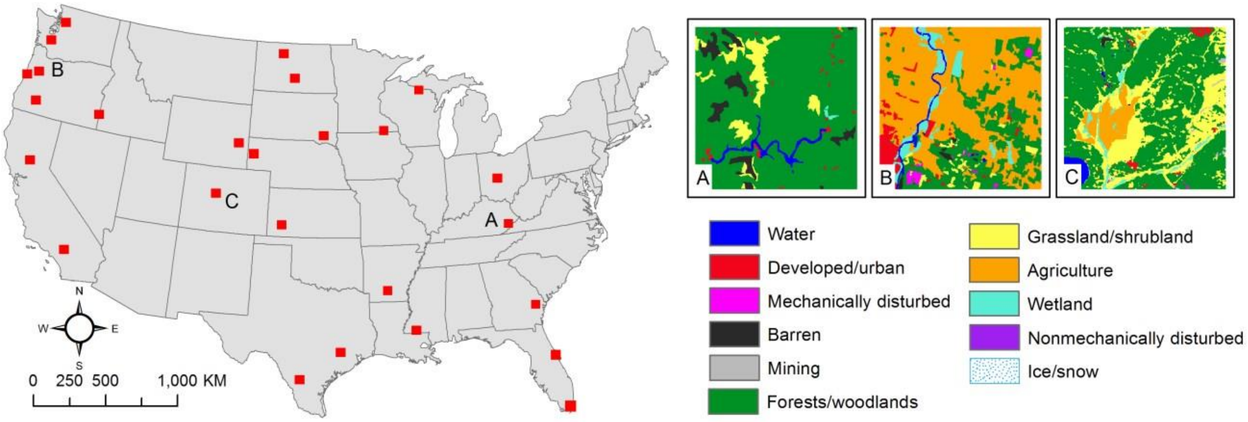


Figure 2-1: Spatial distribution and three samples of the 26 images used in this study (from Khatami et al. 2017)

2.3. Methodology

2.3.1 Sampling design

We used a fixed-rate stratified class-proportional random sampling on each image to train the classifiers and validate them. Having full reference data for each of the 26 blocks, we randomly sampled each land cover type at 2% and 0.2% to assess the effect of sample size. Note that land cover types with less than two sample pixels in the sampled set (less than 122 instances in the entire image) were dropped. To increase the statistical confidence on the performance results, the process was replicated 10 times to create 10 independent sampled data sets (that we name *calibration sets* hereafter) for each image, and the calibration datasets were the same for all classifiers. Each calibration set was further divided into training and validation parts. The training part comprised 82% of calibration set and was used to train a classifier, and the validation part was used to check classifier's generalization capability and pick the best model for accuracy assessment. Final assessed accuracy was reported at the entire block according to the procedure described in section 2.3.3. As the main purpose of this research was to compare the classifiers' performance over a large data set and under an extended parameter grid search, we did not enter the topic of optimizing training sample selection and especially recent advances in active learning area.

2.3.2 Classifier parameterization and training

Six popular classifiers were selected, and their implementation in Mathwork's Matlab was used to run the experiments. Details on these classifiers can be found in multiple sources, for example Domingos and Pazzani (1997) for Bayesian classifiers; Foody and Mathur (2004) for SVM; Calvo-Zaragoza et al. (2015) for KNN; Breiman (1996) and Breiman (2001) for tree ensembles and Random Forests; Mas and Flores (2008) for ANN; and Chen et al. (2014) for DNN. However, we recommend consulting the Matlab documentation (Mathworks Inc. 2016) and Matlab help pages for classifier parameters description, especially for neural network classifiers. Each classifier has a set of tuning parameters; we selected the most important ones (based on past studies) as indicated in Table 2-1. We defined ranges of applicable values for each parameter and tested the classifiers' performance for each possible combination of individual parameters to identify the best performer (i.e., a grid search approach). The range of values for each parameter was chosen to cover the practically important cases. In some cases, a subset of all available parameter settings was used through a quick initial assessment in order to constrain the large number of possible parameter combinations.

Table 2-1: Classifiers' parameters

Classifier	***Parameter***	***Parameter values/range***
Naïve Bayesian (NB)	Probability distribution type	Normal, Kernel
	Smoothing function	Normal, Box, Triangle, Epanechnikov
K-Nearest Neighbor (KNN)	Distance metric	Chebychev, Euclidean, Mahalanobis, Minkowski
	Distance weight	Inverse, squared inverse
	Number of neighbors	1 to 40 (step of 2)
Support Vector Machine (SVM)	Kernel function	Fixed at Gaussian
	Box constraint (C)	0.01, 0.1, 0.5, 1, 2, 5, 10, 25, 50, 100, 300
	Kernel scale (gamma)	0.1, 0.5, 1, 2, 5, 10, 25, 50
Tree ensemble (BagTE)	Ensemble method	Bagging
	Number of trees	50, 100, 200, 500
	Maximum number of tree splits	10, 25, 50, 100, 200
	Minimum tree leaf size	1, 3, 5, 10, 25
	Number of simulation iterations	10
Artificial Neural Network (1 or 2 hidden layers, followed by a softmax classifier)	Training algorithm	Resilient backpropagation (trainrp)
	# of nodes in 1st hidden layer	5 to 15 (step of 1)
	# of nodes in 2nd hidden layer	0 to 8 (step of 1)
	Number of simulation iterations	100
	Training parameters (specific to chosen training algorithm): - Learning rate - Delta0 - Delta_inc - Delta_dec	Changed randomly in each iteration within given range: - 0.01 ~ 1 - 0.01 ~ 0.5 - 1 ~ 5 - 0.1 ~ 1
Deep Neural Network, autoencoder-based (1, 2, or 3 hidden layers, followed by a softmax classifier)	Training algorithm	Standard backpropagation
	# of nodes in 1st hidden layer	5 to 30 (step of 2)
	# of nodes in 2nd hidden layer	0 to 20 (step of 2)
	# of nodes in 3rd hidden layer	0 to 10 (step of 2)
	Number of simulation iterations	100
	Training parameters (specific to chosen training algorithm): - Lambda - Rho - Beta	- 1E-8 ~ 1E-3 - 0.05 ~ 0.7 - 1 ~ 9

Additional considerations pertaining to a specific classifier include the following:

- NB: The smoothing function was used only when the probability distribution type is set to ‘Kernel’.
- KNN: The ‘Minkowski’ metric setting requires an additional parameter named ‘exponent’, it was set at a fixed value of 3 in all simulations of this distance type.
- BagTE: Due to the randomization involved in the bagging algorithm, we repeated each single run of classifier for a number of iterations and picked best result to record. Our experiments showed that the tree ensemble performance result varied marginally between iterations so we limited the number of iterations to 10. Note that the BagTE slightly differs from the Random Forest implementation. The Random Forest preselects the features used to make each tree branch randomly among all the feature sets, but in the BagTE all the features are available at each branching.
- ANN: As with the BagTE parameter initialization values may affect the result. In the ANN this effect is more pronounced than in the BagTE (standard deviation more than 20% accuracy in some cases) therefore we set the number of iterations to 100. For some parameters an exhaustive grid search took place (# of nodes) while for other parameters, random values within the predefined range were assigned for each run to keep the combination choices at a reasonable level.
- DNN: Training followed the same considerations as the ANN. To have control over training parameters, the DNN implementation was based on custom code with the help of the “Unsupervised Feature Learning and Deep Learning” web site at http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial.

After building the training/validation data sets and setting up the classifier parameter grid, the main experiment was done by running each classifier on each image separately, iterating through all parameters grid points and all ten input data sets (repetitions for same parameters) for that image.

2.3.3 Accuracy Assessment

The accuracy assessment replicates the typical algorithmic training procedure with the advantage that entire block accuracy metric generalizations can be extracted for further study. Table 2-2 presents the general steps followed to obtain accuracy estimations for each classifier and block.

Table 2-2: Pseudocode of accuracy calculation for each classifier and each image block

```
Define classifier to use
    For replication =1:10 (10 calibration datasets per block)
        For each parameter combination (dependent on classifier characteristics, see Table 1)
            For iterations =1:n (n=10 for BagTE, n=100 for ANN and DNN, n=1 for others)
                Train classifier using training data
                Estimate accuracy metric using validation data
            End
        End
        Identify optimal parameter set defined as the set with the highest validation accuracy metric for
            given replication
        Calculate the entire block accuracy metric for the selected optimal parameters
    End
Calculate average best entire block accuracy metric (and standard deviation) over the ten calibration datasets
```

The above process was repeated for the two sampling rates, 0.2% and 2% separately (sections 2.4.1 and 2.4.2). We also did the assessment only for the subset of edge pixels (i.e. pixels that lay on the border line of different land cover classes) to assess the influence of landscape heterogeneity, further discussed in section 2.4.4. In some cases, the simulations were also repeated by changing the class distribution in data sets as described in section 2.4.3.

There are many metrics available to assess classifiers performance in the literature. We used overall accuracy (OA) as one of the most widely used metrics with easy interpretation and high practical value. The drawback is that OA hides the class specific performance and as discussed in H. He and Garcia (2009), the OA value can be deceiving when the input dataset is highly imbalanced. In such a case, the OA value mostly reflects the dominant class performance while the rare classes may be classified very poorly. Using other metrics such as Precision/Recall or Receiver Operating Characteristics (ROC) is more favored when performance on rare classes is more important. Picking OA as the assessment metric, the best class distribution is the naturally occurring one H. He and Garcia (2009) and therefore our stratified sampling for training matches the selected metric. Another metric, the Kappa statistic, has also been used in prior literature to reflect the possibility of chance agreement. However, its usage is nowadays less favored and even it is highly criticized to be “useless, misleading and/or flawed for the practical applications in remote sensing that we have seen” (Pontius and Millones, 2011). Therefore, we opted to solely report OA results.

In our presented results we mainly compare classifiers according to their performance (measured by OA) on the same dataset, therefore dependency of OA to class distribution does not bias results. In section 2.4.4 we look at the change in OA over all blocks by change in frequency of edge pixels, which may affect results. We therefore discuss the class distribution issue in section 2.4.3 before presenting the edge pixel analysis results.

2.4. Results

2.4.1 Classifier accuracy comparison for a typical 2% reference dataset

Table 2-3 shows the obtained average best entire block overall accuracy following the procedure of Table 2-2 for the 2% calibration dataset (Table A-2 in Appendix contains the corresponding results for the 0.2% calibration dataset). The coefficient of variation (ratio of standard deviation to mean value) is shown in parenthesis. The SVM was the best classifier in 18 out of 26 cases followed by BagTE (3 cases), ANN (3), and DNN (2). However, with the exception of NB that had significantly lower performance, all other classifiers performed similarly with minor practical variations. This result indicates that when sufficient training data and parameter searches are fed to these popular algorithms, performance does not differ substantially.

We also looked at the best setting of classifier parameters for each image but found that there is no specific parameter value that can be advised for a classifier as the best parameters over all study cases. In fact, raising one configuration as the winner over the others is not justified and everything is dependent on a specific image (and sampling design). The NB method was not considered further in this manuscript due to its considerably lower performance.

Table 2-3: Best average overall accuracies and their coefficient of variation for 2% sample size

Classifier→	***NB***	***SVM***	***KNN***	***BagTE***	***ANN***	***DNN***
ImageNo ↓						
1	95.10% (0.80%)	98.11% (0.32%)	98.08% (0.21%)	98.19% (0.13%)	98.05% (0.16%)	98.10% (0.10%)
2	64.45% (6.76%)	83.84% (0.40%)	83.14% (0.52%)	82.97% (0.42%)	82.42% (0.55%)	83.12% (0.56%)
3	73.01% (1.01%)	92.01% (0.96%)	91.25% (0.26%)	91.32% (0.49%)	91.72% (0.40%)	91.81% (0.37%)
4	74.71% (1.46%)	80.38% (0.50%)	79.28% (0.48%)	80.24% (0.31%)	79.93% (0.63%)	80.35% (0.62%)
5	96.31% (0.41%)	98.37% (0.15%)	98.15% (0.13%)	98.09% (0.06%)	98.31% (0.20%)	98.24% (0.23%)
6	68.71% (4.47%)	77.95% (0.54%)	77.71% (0.72%)	78.18% (0.18%)	78.10% (0.34%)	78.02% (0.51%)
7	87.99% (0.16%)	90.85% (0.25%)	90.57% (0.19%)	90.48% (0.19%)	90.73% (0.26%)	90.80% (0.32%)
8	79.88% (0.85%)	84.66% (0.43%)	84.32% (0.51%)	84.47% (0.31%)	84.58% (0.32%)	84.38% (0.47%)
9	57.26% (1.21%)	65.29% (0.72%)	64.17% (1.50%)	64.76% (0.59%)	64.15% (1.39%)	64.57% (1.05%)
10	58.37% (0.96%)	77.16% (0.32%)	76.26% (0.75%)	75.99% (0.36%)	75.56% (0.73%)	76.43% (0.51%)
11	87.83% (0.62%)	92.26% (0.31%)	92.21% (0.34%)	92.39% (0.13%)	92.18% (0.21%)	92.19% (0.18%)
12	65.22% (0.82%)	76.01% (1.03%)	75.94% (0.37%)	75.95% (0.35%)	75.72% (0.65%)	76.11% (0.33%)
13	80.39% (0.50%)	83.85% (0.33%)	83.75% (0.34%)	83.72% (0.19%)	83.88% (0.34%)	83.99% (0.18%)
14	84.91% (0.46%)	87.66% (0.30%)	87.19% (0.56%)	87.45% (0.20%)	87.70% (0.36%)	87.67% (0.23%)
15	64.13% (4.40%)	86.72% (0.42%)	86.24% (0.26%)	86.20% (0.54%)	86.27% (0.57%)	86.54% (0.51%)
16	79.09% (0.95%)	86.74% (0.28%)	85.79% (0.28%)	85.97% (0.37%)	86.80% (0.29%)	86.56% (0.44%)
17	77.68% (2.96%)	85.83% (0.45%)	84.78% (0.69%)	85.06% (0.55%)	85.21% (0.47%)	85.35% (0.37%)
18	68.95% (7.39%)	85.09% (0.44%)	85.24% (0.30%)	85.12% (0.17%)	85.34% (0.27%)	85.17% (0.26%)
19	72.62% (2.49%)	80.53% (0.41%)	80.26% (0.50%)	80.29% (0.26%)	80.39% (0.17%)	80.33% (0.23%)
20	65.33% (1.73%)	78.70% (0.99%)	77.29% (1.54%)	76.97% (0.63%)	78.05% (0.59%)	77.90% (1.07%)
21	80.20% (0.86%)	87.20% (0.33%)	86.88% (0.32%)	86.55% (0.31%)	87.17% (0.32%)	86.68% (0.55%)
22	58.18% (2.24%)	71.99% (1.00%)	71.23% (0.40%)	70.99% (0.54%)	70.99% (0.64%)	71.79% (0.59%)
23	74.43% (0.40%)	87.74% (0.31%)	86.80% (0.30%)	86.74% (0.37%)	87.36% (0.46%)	87.37% (0.71%)
24	81.86% (1.03%)	89.19% (0.16%)	88.77% (0.32%)	88.63% (0.25%)	88.83% (0.37%)	88.89% (0.30%)
25	76.18% (1.50%)	80.38% (0.54%)	80.26% (0.25%)	80.20% (0.32%)	79.91% (0.44%)	79.84% (0.56%)
26	85.33% (1.04%)	93.71% (0.24%)	93.29% (0.19%)	93.21% (0.28%)	93.60% (0.16%)	93.39% (0.45%)

(for ANN and DNN classifiers, the listed OA is the highest achieved by any number of hidden layers and nodes per layer within the parameter limits)

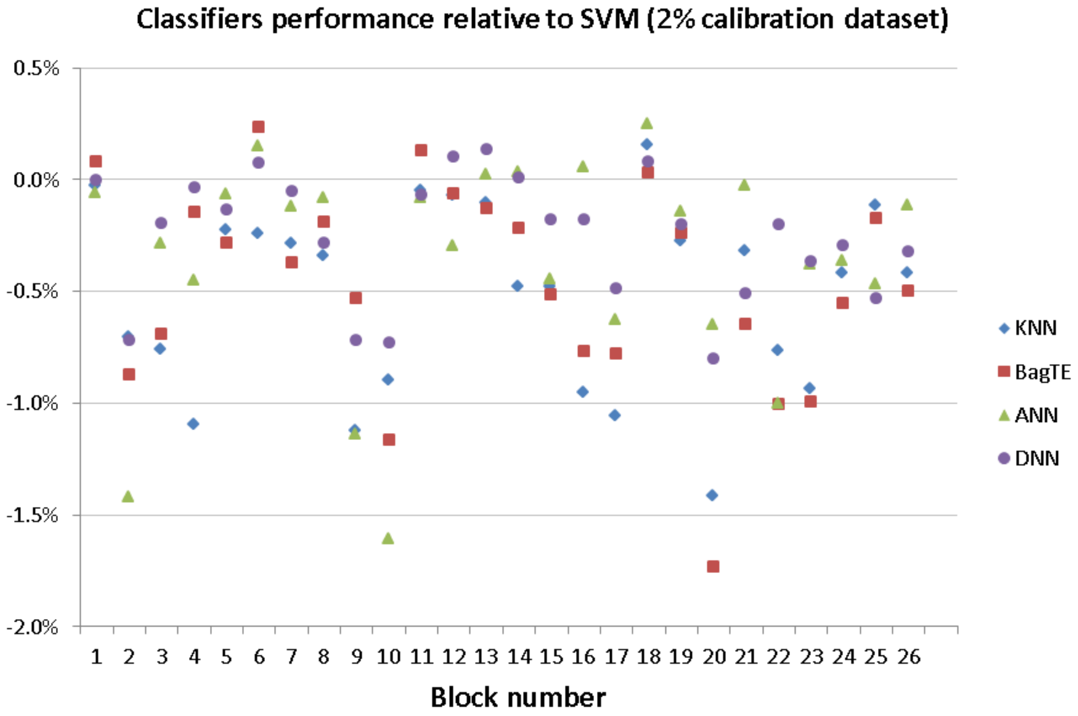
2.4.2 Effect of sample size on classification accuracy

Although it is desirable to have as many training data as possible, accurately located and labelled training samples in a remote sensing application are generally difficult to obtain. Our sampling rate of 2% for each land cover type (which translates to the total of 2218 pixels for each of our 110,889-pixel blocks) reflects a practical upper bound. For completeness, we also tested the classifiers' performance with a larger 5% sampling rate and the results were very close to the 2% sampling ratio, showing a saturation in classifiers' accuracy. We also examined a lower bound by generating a second, considerably smaller calibration dataset (with the same proportional stratified sampling design) at 0.2% of the image size (222 pixels). Table 2-4 shows the best result among all classifiers for 2% and 0.2% sampling scenarios. Detailed accuracy metrics for the 0.2% sampling rate were similar to Table 2-3 and are offered in the Appendix A, Table A-2. As expected, there was a decline in best attainable accuracy with the 0.2% sampling ranging from 0.6% to 5.7%.

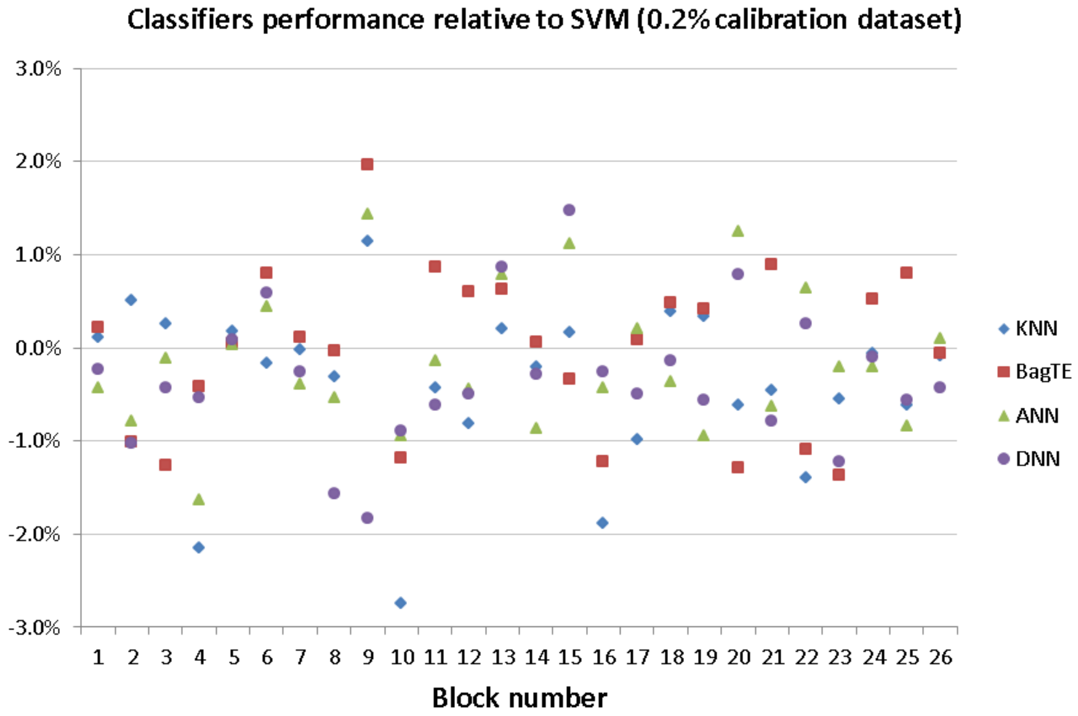
Table 2-4: Best attainable accuracy (over all classifiers) for sampling rates of 2% and 0.2%

Image# →		***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***
Sampling rate	2	98.2	83.8	92.0	80.4	98.4	78.2	90.9	84.7	65.3	77.2	92.4	76.1	84.0
Sampling rate	0.2	97.6	79.5	88.5	77.5	97.6	76.1	88.9	81.8	59.6	71.6	91.5	73.6	82.1
Image# →		***14***	***15***	***16***	***17***	***18***	***19***	***20***	***21***	***22***	***23***	***24***	***25***	***26***
Sampling rate	2	87.7	86.7	86.8	85.8	85.3	80.5	78.7	87.2	72.0	87.7	89.2	80.4	93.7
Sampling rate	0.2	86.0	84.1	84.3	82.7	83.4	79.1	73.3	84.4	66.9	84.5	86.2	78.5	91.2

To investigate further individual classifier performance each classifier's accuracy was contrasted with the SVM accuracy. Figure 2-2 depicts this comparison for the 2% and 0.2% calibration datasets (excluding NB due to low performance).



(a)



(b)

Figure 2-2: Classifiers overall accuracy relative to SVM for (a) 2% and (b) 0.2% calibration dataset

For the 2% case the SVM typically outperforms other classifiers. In the 0.2% case the BagTE is the best classifier in 12 out of 26 cases, and SVM wins in only 5 cases. However, independently of the calibration dataset size the magnitude of the accuracy difference is still small and practically insignificant.

2.4.3 Effect of dataset class imbalance on classification accuracy

In the previous two sections we compared different classifiers given similar training and testing data separately for each image, therefore class distribution did not bias the results. In this section, we specifically assess the effect of class distribution in obtained accuracy. Two training sample scenarios were examined, one with a stratified proportional training set (imbalanced training dataset), and another by training classifiers with randomly selected almost equal class member datasets. In addition to the two training scenarios, performance for each image was assessed using two different testing datasets, one covering the entire image (imbalanced testing dataset) and another constraining equal members per class (balanced testing dataset).

Figure 2-3 shows the result of calculating OA for the imbalanced training dataset and Figure 2-4 shows the same thing for balanced training dataset. In each case, OA values express the maximum attainable value among different classifiers for each image. Due to the low number of pixels in rare classes, the overall dataset size (and therefore the training part) in the balanced scenario was smaller than the original imbalanced case, therefore accuracy comparisons are only applicable within the same training dataset (i.e. Figure 2-3 and Figure 2-4 should not be combined). The OA drops is significant by changing class distribution with the imbalanced training dataset (Figure 2-3), however the difference is limited under the balanced distribution training (Figure 2-4). This finding suggests that stratification for sample selection can significantly impact the obtained accuracy, therefore the sampling design should match the study preferences (highest overall scene accuracy vs. balanced accuracy between classes).

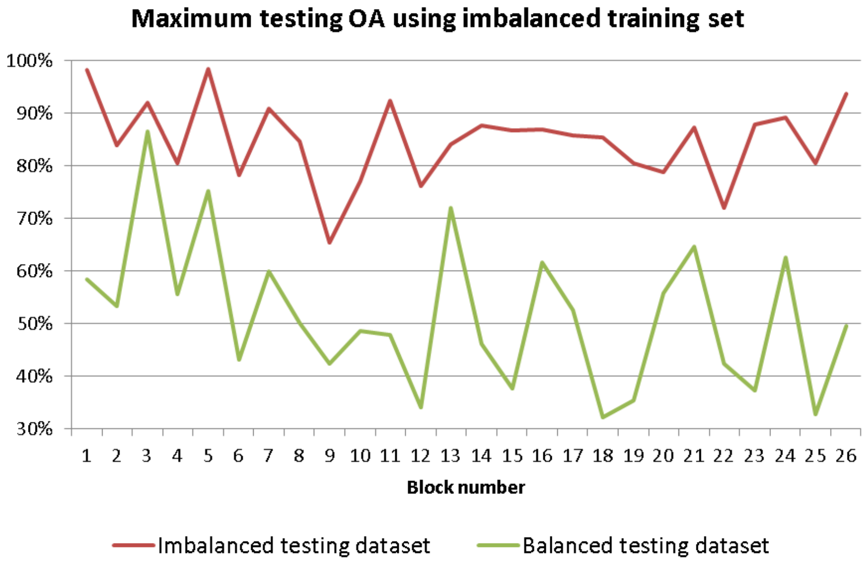


Figure 2-3: Effect of changing class distribution in test sets on best attainable accuracy for imbalanced training set distribution

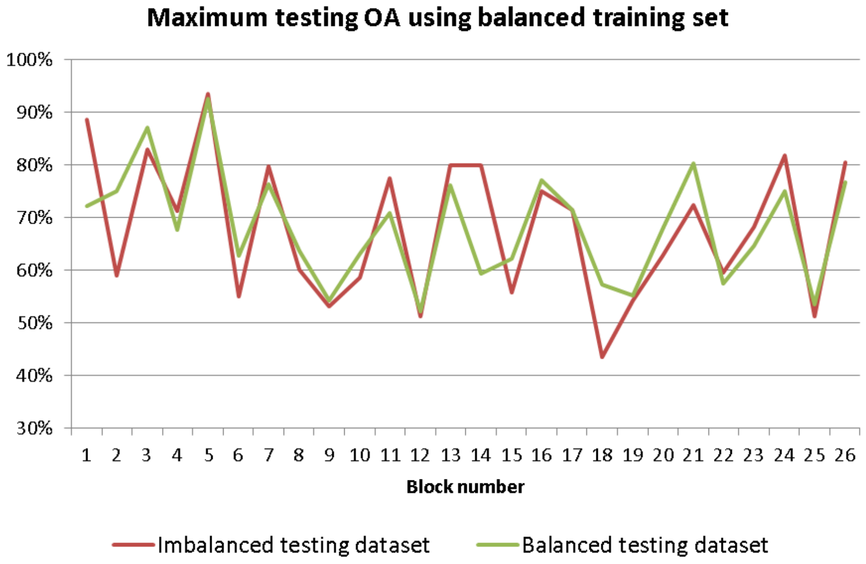


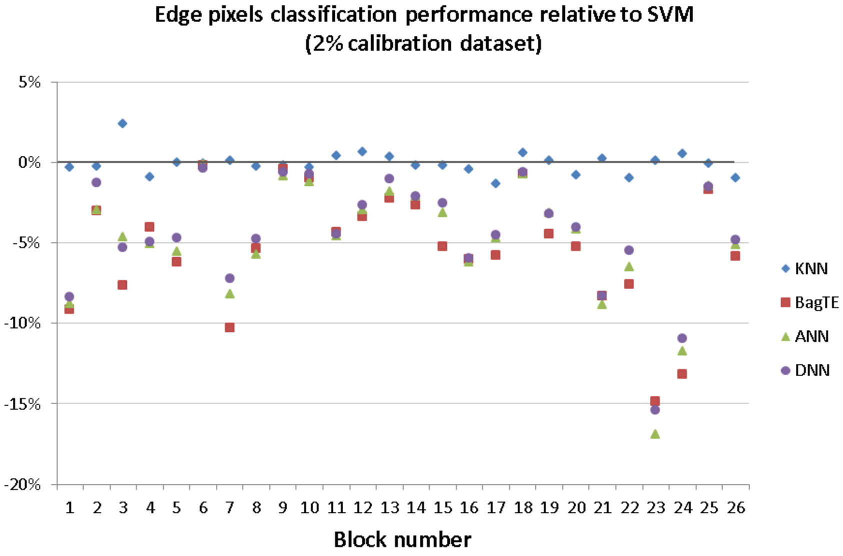
Figure 2-4: Effect of changing class distribution in test sets on best attainable accuracy for balanced training set distribution

2.4.4 Effect of landscape heterogeneity on classification accuracy

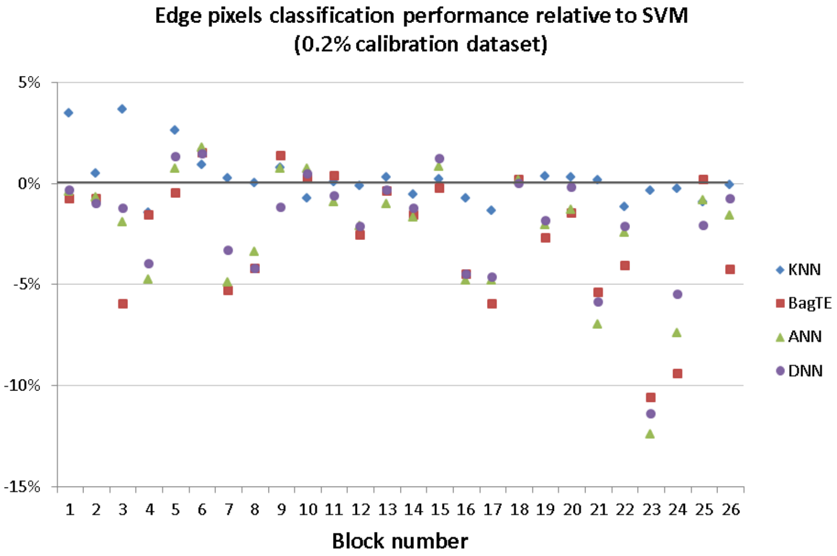
Numerous metrics have been developed over time to characterize and assess the scene and landscape heterogeneity, and software packages are also available for help (for example see Turner, 2005; or Lausch et al. 2015). These metrics can be defined in many ways and vary in scope, considering the number of different landscape classes and/or their spatial distribution. An exhaustive list of tens of landscape metrics can be found in FragStat software's documentation (McGarigal, 2015). Although there is no general rule to pick among them, edge statistics are a good candidate to represent scene heterogeneity as it is affected by both class variety and class spatial arrangement. Here we defined the edge pixels as pixels lying on land cover change boundaries; they were extracted from the ground truth data. This selection was also natural from a remote sensing point of view, because Landsat images are of medium resolution and in the edge pixels, there is a high chance of land cover mixing. Our 26 blocks exhibited a wide range of edge pixel presence ranging from 2% to more than 40% of the overall block pixels. Two separate analyses are presented in the next two sections. First, we isolated each block and examined algorithmic performance on the edge pixels in order to identify best performing classifiers. Second, we combined classifier performance across all blocks to investigate accuracy degradation as scene heterogeneity increases through higher edge pixel presence.

2.4.4.1. Algorithmic accuracy assessment on edge pixels separately for each block

Having previously trained classifiers on stratified proportional samples, we can calculate the test accuracy by limiting the test pixels only to edge pixels for each image. The idea is to investigate how different classifiers perform particularly on these difficult-to-classify pixels. Resulting accuracies are depicted in Figure 2-5 relative to SVM performance. In Figure 2-5(a) the classifiers have been trained on 2% stratified proportional sample and tested on edge pixels (for each image). Figure 2-5(b) shows the same result but the training has been done by 0.2% sampling rate. This figure shows superiority of the SVM and KNN classifiers, especially for the larger 2% calibration sample. BagTE, ANN and DNN performance is not as consistent; it can be close to the SVM/KNN or it could deviate considerably.



(a)



(b)

Figure 2-5: Edge pixels classification accuracy relative to SVM for (a) 2% and (b) 0.2% calibration dataset

2.4.4.2 Effect of edge pixel presence on accuracy across blocks

It is interesting to seek a potential relationship between scene heterogeneity and classification accuracy. Figure 2-6 shows the OA results vs. the ratio of edge pixels in each block. For clarity purposes and guided by results in Figure 2-5 we limited assessment to the two best performing classifiers, SVM and KNN. The presented results are based on the balanced training dataset to limit potential class influence on the obtained results. For testing purposes the entire block dataset was used as accuracy differences between an unbalanced and a balanced testing dataset were minor (see Figure 2-4).

A clear decreasing trend can be identified with approximately 8-9% accuracy reduction for every 10% increase in edge pixel presence for SVM. While the model explained about one third of the variability, it is an important finding considering the multitude of additional factors that may affect classification accuracy in our 26 different sites (e.g., variable spectral signatures and separability of classes).

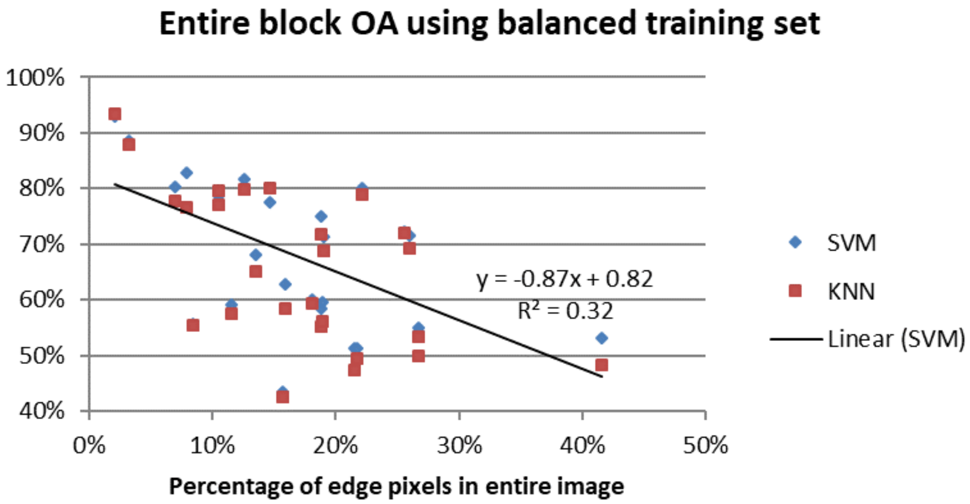


Figure 2-6: Effect of edge pixel presence in classifiers overall accuracy, trained on balanced data set and tested on entire image

2.4.5 Trade-off between execution time and accuracy

Average run-time requirements for the experiment reported in Table 2-3 (2% reference dataset) are presented in Table 2-5. To compensate for the usage of different machine configurations and parallel processing capabilities (i.e., number of CPU cores) the NB classifier ran on all machines and the run times were used as a benchmark to provide a common base for comparison. Our intention is not to provide exact execution times but simply a ballpark figure to guide user decisions. The total run time was directly dependent on the number and range of configuration parameters. We also calculated the average runtime per each parameter setting in the last column, but we still had an arbitrary parameter as the number of iterations some cases, which directly affect total runtime. In our case, NB and SVM were the fastest classifiers per image, but DNN classifiers tend to be the quickest classifiers per each parameter setting (on average).

Table 2-5: Average computer run times per image per CPU core for different classifiers

Classifier	***Average run time per Image per CPU core for all combinations (min.)***	***# of parameter combinations***	***# of iterations per parameter***	***Average single run time per parameter setting (sec.)***
NB	1.5	5	1	18.1
SVM	4.0	88	1	2.7
KNN	22.1	160	1	8.3
BagTE	53.3	100	10	3.2
ANN	464.3	99	100	2.8
DNN (1-Layer)	15.0	13	100	0.7
DNN (2-Layer)	278.2	130	100	1.3
DNN (3-Layer)	1365.8	650	100	1.3

Lacking a specific protocol on how to set the classifier parameters for best performance, our approach was to do a complete set of simulations for each image/classifier over all reasonable parameter combinations. However, building on our experiments we can investigate best attainable accuracy (on average) taking only a subset of the initial parameter values. A smaller set may miss some of the best parameter combinations, resulting in a decrease in best attainable accuracy. Table 2-6 and Table 2-7 show worst case estimates of this gap for different cases of the parameter set contraction for 2% and 0.2% sampling rates respectively. For generating these tables, we assumed that by extracting the accuracy at a given percentile the worst case scenario is obtained. For example in the 100%-75% column, the 75th percentile of the obtained accuracies for each case (image/classifier) was identified. It may be unlikely that by randomly constraining the parameter combinations to 75% of the total possibilities the resulting accuracy will also be bounded by accuracy’s 75th percentile, but this is the worst case. Then we averaged the gap between top and 75th percentile accuracy over all blocks for each classifier and reported the results in 100%-75% column (same procedure for other columns by changing 75th percentile to other percentile values). In the special case of 1-Layer DNN, which has only 13 different configurations, 5th percentile is not meaningful hence table entry is set to N/A. Results indicate that the most tolerant classifier to limiting parameter search space is the BagTE, while the least tolerant is the SVM.

Table 2-6: Percentiles performance gap for 2% reference dataset

Classifier	*100%-75%*	*100%-50%*	*100%-25%*	*100%-10%*	*100%-5%*
SVM	0.9%	3.0%	8.7%	13.4%	13.7%
KNN	0.3%	0.5%	1.0%	2.1%	3.8%
BagTE	0.2%	0.6%	1.1%	1.7%	1.8%
ANN	0.5%	0.7%	1.1%	2.1%	2.6%
DNN (1-Layer)	0.5%	0.7%	1.1%	1.7%	N/A
DNN (2-Layer)	0.7%	1.1%	1.5%	2.0%	2.4%
DNN (3-Layer)	1.2%	1.8%	2.5%	3.1%	3.5%

Table 2-7: Percentiles performance gap for 0.2% reference dataset

Classifier	*100%-75%*	*100%-50%*	*100%-25%*	*100%-10%*	*100%-5%*
SVM	1.9%	5.0%	11.7%	11.9%	12.0%
KNN	0.8%	1.3%	2.3%	3.9%	5.1%
BagTE	0.5%	0.8%	1.4%	2.9%	3.1%
ANN	1.3%	2.0%	2.9%	3.9%	4.7%
DNN (1-Layer)	0.7%	1.4%	2.5%	3.7%	N/A
DNN (2-Layer)	2.6%	3.5%	4.5%	5.4%	6.0%
DNN (3-Layer)	2.9%	3.8%	4.8%	5.7%	6.3%

2.5. Discussion and concluding remarks

Our goal was to investigate classifier performance under different sampling scenarios and landscape complexities. For the entire block, our accuracy assessment indicated that all classifiers, with the exception of NB, performed similarly. The general performance gap with Naïve Bayes compared to the other classifiers can be explained by the high level of band correlation, which invalidates the class conditional independence assumption. For other classifiers, similar results can be obtained assuming sufficient search of optimal parameter identification. This suggests that parameter optimization is a key component in the training process and results using a pre-determined set could be misleading (as presented in Lawrence and Moran, 2015). Unfortunately, optimal parameter values may vary significantly across sites, therefore, an extensive grid search is required. Moving beyond individual classifiers, training data characteristics can be more influential than classifier selection as shown by limiting the test data to edge pixels. A similar conclusion has been made in other studies, for example in (C. Li et al. 2014).

However, when we concentrated on the edge pixels, it was clear that the SVM and KNN offer considerable accuracy advantages. This could be attributed to the right balance between algorithmic and data complexity. Other methods (ANN, DNN) may offer higher modelling capabilities; however, the relatively small training datasets result in unpredictable generalizations in the feature space. SVM and KNN may also work better than decision trees in the presence of imbalanced data and rare classes because decision trees require enough training samples to find optimum branching decisions and divide-and-conquer strategies may fail on imbalanced data sets. Coupled with their relatively low execution times we would recommend SVM or KNN for classifications using Landsat's spectral inputs and Anderson's 11-level classification scheme. We should also caution though that primarily the SVM and secondarily the KNN demonstrated substantial accuracy degradation during the parameter grid search, therefore an exhaustive optimization process is suggested.

Moving into further details and to compare our findings with prior research, we looked at the articles database provided in (Khatami et al. 2016) and selected similar case studies (i.e. analyzing Landsat multispectral images with no ancillary data) along with other recent works. According to Ouyang and Ma (2006), Zhong et al. (2007), Dixon and Candade (2008), Qing et al. (2010), and C.-H. Li et al. (2012), SVM outperformed the Maximum-Likelihood classifier (which is based on the same principle as our NB classifier) by at least 5% in overall accuracy, but the SVM gain was less than 5% compared to multilayer neural networks and less than 3% when compared to KNN classifier. In a different experiment Maximum Likelihood, Neural Network, and SVM achieved overall accuracies with difference less than around 1%, and it was not statistically significant (Mallinis and Koutsias, 2012).

J. He et al. (2015) used Landsat images and reported SVM as the best classifier, followed by Neural Networks, Random Forest, and lastly Maximum Likelihood. The average performance difference between first two classifiers was not statistically significant, also between last two ones, but it was significant (although less than 5%) between the two groups. In another recent study, the Maximum Likelihood classifier was 2% less accurate than the Random Forest, with the latter achieving 86.8% (J. Liu et al. 2016). Lawrence and Moran (2015) reported higher performance for Random Forest than SVM, although their use of a fixed set of parameters may not allow either algorithm to reach their potential. One recent study (Weijia Li et al. 2016) reported a 1.2% increase in overall accuracy of a 3-layer DNN compared to SVM; also RF was 1.8% worse than SVM, with differences being statistically significant. However, it is not clear if an extensive grid search was used in their analysis for SVM and RF. Finally, Pelletier et al. (2016) did a large grid search on SVM and RF parameters and found the RF to perform significantly better than SVM. They also noticed the RF's low sensitivity to parameter changes.

With respect to the BagTE, a Random Forest variant, it showed the highest potential when the parameter search space is minimized, similar to Pelletier et al. (2016). This is attributed to the ensemble nature of this classifier that potentially makes it more tolerant to small or noisy samples. Neural network classifiers (ANN and DNN) did not reach their promising credentials in our study. In other fields, ANNs and particularly DNNs have provided significant advances when fed with large amounts of information. Rich data was not the case in our experiment as we restricted input data to pixel-based multispectral information and we found neural networks generally less promising in our case compared to SVM, KNN, and tree ensembles. This may be the result of insufficient or low quality training samples, or data overfitting because of higher complexity of classification network compared to data structure. In our simulations simpler (1-layer) deep networks worked generally better than deeper ones, and we found in our additional trials that increasing the sampling ratio (we tried up to 5%) or using edge pixels for classifier training (case of active learning) does not make the neural network winner. Therefore, it is more probable that this deficiency comes from low number of features (and their dependence) and data overfitting due to higher complexity of neural networks. As described in Zhang et al. (2016), the main benefit of DNN use will be for processing hyperspectral data, or mix the spectral data with spatial and contextual information and then combine the spectral and other information in a composite per-pixel analysis. Therefore, while neural networks offer limited benefits in our six-dimensional spectral feature space, they still may offer advances when feature space dimensionality increases and spatial relationships are included.

Another finding across classifiers is a reduction of classification accuracy as scene complexity increases. While this has been reported in the past by looking at the class richness or other landscape heterogeneity metrics (for example in Mallinis and Koutsias, 2012; Collin and Planes, 2012; Roelfsema and Phinn, 2010; and Andrefouet et al., 2003), comparing performances over multiple scenes based on ratio of edge pixels has not been done before to our knowledge.

Looking into future work, there are two important areas for further evaluation: selection of performance evaluation metric, and sampling design alternatives. Although overall accuracy is widely used, there are some suggestions that prefer ROC or Precision/Recall curves over overall accuracy for analysis of imbalanced cases (e.g. Jeni et al. 2013). A closer look could also identify land cover classes that exhibit higher confusion and try to at least balance the misclassification errors over different classes (Puertas et al. 2013) or perform a one-class classification and modify the evaluation metric (Wenkai Li and Qinghua Guo, 2014). Another approach is to use accuracy metrics at the individual pixel level (Khatami et al. 2017).

With respect to sampling design alternatives (for training) many approaches have been recently devised especially for learning from imbalanced data, as reviewed in H. He and Garcia (2009) and later presented in a book (H. He and Ma, 2013). Systematic inclusion of difficult-to-classify samples like edge pixels is another option to consider, which has been investigated recently in another research (M. Liu et al. 2016). This approach can be considered as an example of a group of techniques named active learning, which is well known in machine learning and has been used and discussed in remote sensing field as well (see Bachmann, 2003, and Crawford et al. 2013). As discussed and reviewed in Tuia et al. (2011), it "aims at building efficient training sets by iteratively improving the model performance through sampling." In other words, samples used for training are selected interactively. Most of the research in this area is, for now, concentrated on very high spatial/spectral resolution imagery, and Landsat type data is not examined by this approach. There are also cases of unexpected results with active learning (Wuttke et al. 2016), so caution should be exercised.

To summarize, our experiments identified SVM and KNN as the best performing methods for Landsat classifications. Caution should be exercised though as their performance is dependent on a wide search of their parameter space. Furthermore, the selection of the training sample composition (class balance) will have a considerable effect on the obtained accuracy, therefore users should consider accuracy priorities (overall scene vs specific classes) in their sampling design. Finally, edge pixel presence, a heterogeneity metric, was shown to have a considerable effect on the classification accuracy.

Acknowledgments

This work was supported by the USDA McIntire Stennis program, a SUNY ESF Graduate Assistantship and NASA's Land Cover Land Use Change Program (grant # NNX15AD42G). We thank Reza Khatami for sharing the processed datasets and suggestions for the experimental design. We also appreciate the efforts from Kristi Sayler and Mark Drummond (USGS) for providing the Trends data. We thank the reviewers for providing constructive corrections and insights, especially with respect to the influence of the class distribution.

References

Anderson, James R., Ernest E. Hardy, John T. Roach, and Richard E. Witmer. 1976. “A Land Use and Land Cover Classification System for Use with Remote Sensor Data.” Report 964. Professional Paper. USGS Publications Warehouse. http://pubs.er.usgs.gov/publication/pp964.

Andrefouet, Serge, Philip Kramer, Damaris Torres-Pulliza, Karen E Joyce, Eric J Hochberg, Rodrigo Garza-P?rez, Peter J Mumby, et al. 2003. “Multi-Site Evaluation of IKONOS Data for Classification of Tropical Coral Reef Environments.” *Remote Sensing of Environment* 88 (1–2): 128–43. https://doi.org/10.1016/j.rse.2003.04.005.

Bachmann, C.M. 2003. “Improving the Performance of Classifiers in High-Dimensional Remote Sensing Applications: An Adaptive Resampling Strategy for Error-Prone Exemplars (ARESEPE).” *IEEE Transactions on Geoscience and Remote Sensing* 41 (9): 2101–12. https://doi.org/10.1109/TGRS.2003.817207.

Ballantine, John-Andrew C., Gregory S. Okin, Dylan E. Prentiss, and Dar A. Roberts. 2005. “Mapping North African Landforms Using Continental Scale Unmixing of MODIS Imagery.” *Remote Sensing of Environment* 97 (4): 470–83. https://doi.org/10.1016/j.rse.2005.04.023.

Belgiu, Mariana, and Lucian Drăguţ. 2016. “Random Forest in Remote Sensing: A Review of Applications and Future Directions.” *ISPRS Journal of Photogrammetry and Remote Sensing* 114 (April): 24–31. https://doi.org/10.1016/j.isprsjprs.2016.01.011.

Breiman, Leo. 1996. “Bagging Predictors.” *Machine Learning* 24 (2): 123–40. https://doi.org/10.1023/A:1018054314350.

———. 2001. “Random Forests.” *Machine Learning* 45 (1): 5–32. https://doi.org/10.1023/A:1010933404324.

Calvo-Zaragoza, Jorge, Jose J. Valero-Mas, and Juan R. Rico-Juan. 2015. “Improving KNN Multi-Label Classification in Prototype Selection Scenarios Using Class Proposals.” *Pattern Recognition* 48 (5): 1608–22. https://doi.org/10.1016/j.patcog.2014.11.015.

Chen, Y., Z. Lin, X. Zhao, G. Wang, and Y. Gu. 2014. “Deep Learning-Based Classification of Hyperspectral Data.” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7 (6): 2094–2107. https://doi.org/10.1109/JSTARS.2014.2329330.

Chirici, Gherardo, Matteo Mura, Daniel McInerney, Nicolas Py, Erkki O. Tomppo, Lars T. Waser, Davide Travaglini, and Ronald E. McRoberts. 2016. “A Meta-Analysis and Review of the Literature on the k-Nearest Neighbors Technique for Forestry Applications That Use Remotely Sensed Data.” *Remote Sensing of Environment* 176 (April): 282–94. https://doi.org/10.1016/j.rse.2016.02.001.

Cihlar, J., Qinghan Xiao, J. Chen, J. Beaubien, K. Fung, and R. Latifovic. 1998. “Classification by Progressive Generalization: A New Automated Methodology for Remote Sensing Multichannel Data.” *International Journal of Remote Sensing* 19 (14): 2685–2704. https://doi.org/10.1080/014311698214451.

Collin, Antoine, and Serge Planes. 2012. “Enhancing Coral Health Detection Using Spectral Diversity Indices from WorldView-2 Imagery and Machine Learners.” *Remote Sensing* 4 (10): 3244–64. https://doi.org/10.3390/rs4103244.

Crawford, Melba M., Devis Tuia, and Hsiuhan Lexie Yang. 2013. “Active Learning: Any Value for Classification of Remotely Sensed Data?” *Proceedings of the IEEE* 101 (3): 593–608. https://doi.org/10.1109/JPROC.2012.2231951.

Dixon, B., and N. Candade. 2008. “Multispectral Landuse Classification Using Neural Networks and Support Vector Machines: One or the Other, or Both?” *International Journal of Remote Sensing* 29 (4): 1185–1206. https://doi.org/10.1080/01431160701294661.

Domingos, Pedro, and Michael Pazzani. 1997. “On the Optimality of the Simple Bayesian Classifier under Zero-One Loss.” *Machine Learning* 29 (2–3): 103–30. https://doi.org/10.1023/A:1007413511361.

Foody, Giles M. 2002. “Status of Land Cover Classification Accuracy Assessment.” *Remote Sensing of Environment* 80 (1): 185–201. https://doi.org/10.1016/S0034-4257(01)00295-4.

———. 2009. “Classification Accuracy Comparison: Hypothesis Tests and the Use of Confidence Intervals in Evaluations of Difference, Equivalence and Non-Inferiority.” *Remote Sensing of Environment* 113 (8): 1658–63. https://doi.org/10.1016/j.rse.2009.03.014.

———. 2010. “Assessing the Accuracy of Land Cover Change with Imperfect Ground Reference Data.” *Remote Sensing of Environment* 114 (10): 2271–85. https://doi.org/10.1016/j.rse.2010.05.003.

Foody, G.M., and A. Mathur. 2004. “A Relative Evaluation of Multiclass Image Classification by Support Vector Machines.” *IEEE Transactions on Geoscience and Remote Sensing* 42 (6): 1335–43. https://doi.org/10.1109/TGRS.2004.827257.

Gong, Peng, Jie Wang, Le Yu, Yongchao Zhao, Yuanyuan Zhao, Lu Liang, Zhenguo Niu, et al. 2013. “Finer Resolution Observation and Monitoring of Global Land Cover: First Mapping Results with Landsat TM and ETM+ Data.” *International Journal of Remote Sensing* 34 (7): 2607–54. https://doi.org/10.1080/01431161.2012.748992.

Grekousis, George, Giorgos Mountrakis, and Marinos Kavouras. 2015. “An Overview of 21 Global and 43 Regional Land-Cover Mapping Products.” *International Journal of Remote Sensing* 36 (21): 5309–35. https://doi.org/10.1080/01431161.2015.1093195.

He, Haibo, and E.A. Garcia. 2009. “Learning from Imbalanced Data.” *IEEE Transactions on Knowledge and Data Engineering* 21 (9): 1263–84. https://doi.org/10.1109/TKDE.2008.239.

He, Haibo, and Yunqian Ma. 2013. *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley-IEEE Press.

He, J., J.R. Harris, M. Sawada, and P. Behnia. 2015. “A Comparison of Classification Algorithms Using Landsat-7 and Landsat-8 Data for Mapping Lithology in Canada’s Arctic.” *International Journal of Remote Sensing* 36 (8): 2252–76. https://doi.org/10.1080/01431161.2015.1035410.

Jeni, Laszlo A., Jeffrey F. Cohn, and Fernando De La Torre. 2013. “Facing Imbalanced Data--Recommendations for the Use of Performance Metrics.” In , 245–51. IEEE. https://doi.org/10.1109/ACII.2013.47.

Jin, Huiran, Stephen V. Stehman, and Giorgos Mountrakis. 2014. “Assessing the Impact of Training Sample Selection on Accuracy of an Urban Classification: A Case Study in Denver, Colorado.” *International Journal of Remote Sensing* 35 (6): 2067–81. https://doi.org/10.1080/01431161.2014.885152.

Khatami, Reza, Giorgos Mountrakis, and Stephen V. Stehman. 2016. “A Meta-Analysis of Remote Sensing Research on Supervised Pixel-Based Land-Cover Image Classification Processes: General Guidelines for Practitioners and Future Research.” *Remote Sensing of Environment* 177 (May): 89–100. https://doi.org/10.1016/j.rse.2016.02.028.

———. 2017. “Mapping Per-Pixel Predicted Accuracy of Classified Remote Sensing Images.” *Remote Sensing of Environment* 191 (March): 156–67. https://doi.org/10.1016/j.rse.2017.01.025.

Lausch, Angela, Thomas Blaschke, Dagmar Haase, Felix Herzog, Ralf-Uwe Syrbe, Lutz Tischendorf, and Ulrich Walz. 2015. "Understanding and Quantifying Landscape Structure – A Review on Relevant Process Characteristics, Data Models and Landscape Metrics." *Ecological Modelling*, Use of ecological indicators in models, 295 (January): 31–41. https://doi.org/10.1016/j.ecolmodel.2014.08.018.

Lawrence, Rick L., and Christopher J. Moran. 2015. "The AmericaView Classification Methods Accuracy Comparison Project: A Rigorous Approach for Model Selection." *Remote Sensing of Environment* 170 (December): 115–20. https://doi.org/10.1016/j.rse.2015.09.008.

Li, Cheng-Hsuan, Bor-Chen Kuo, Chin-Teng Lin, and Chih-Sheng Huang. 2012. "A Spatial-Contextual Support Vector Machine for Remotely Sensed Image Classification." *IEEE Transactions on Geoscience and Remote Sensing* 50 (3): 784–99. https://doi.org/10.1109/TGRS.2011.2162246.

Li, Congcong, Jie Wang, Lei Wang, Luanyun Hu, and Peng Gong. 2014. "Comparison of Classification Algorithms and Training Sample Sizes in Urban Land Classification with Landsat Thematic Mapper Imagery." *Remote Sensing* 6 (2): 964–83. https://doi.org/10.3390/rs6020964.

Li, Weijia, Haohuan Fu, Le Yu, Peng Gong, Duole Feng, Congcong Li, and Nicholas Clinton. 2016. "Stacked Autoencoder-Based Deep Learning for Remote-Sensing Image Classification: A Case Study of African Land-Cover Mapping." *International Journal of Remote Sensing* 37 (23): 5632–46. https://doi.org/10.1080/01431161.2016.1246775.

Li, Xuecao, Xiaoping Liu, and Le Yu. 2014. "Aggregative Model-Based Classifier Ensemble for Improving Land-Use/Cover Classification of Landsat TM Images." *International Journal of Remote Sensing* 35 (4): 1481–95. https://doi.org/10.1080/01431161.2013.878061.

Liu, Jiantao, Quanlong Feng, Jianhua Gong, Jieping Zhou, and Yi Li. 2016. "Land-Cover Classification of the Yellow River Delta Wetland Based on Multiple End-Member Spectral Mixture Analysis and a Random Forest Classifier." *International Journal of Remote Sensing* 37 (8): 1845–67. https://doi.org/10.1080/01431161.2016.1165888.

Liu, Meng, Xin Cao, Yang Li, Jin Chen, and XueHong Chen. 2016. "Method for Land Cover Classification Accuracy Assessment Considering Edges." *Science China Earth Sciences* 59 (12): 2318–27. https://doi.org/10.1007/s11430-016-5333-5.

Lu, D., and Q. Weng. 2007. "A Survey of Image Classification Methods and Techniques for Improving Classification Performance." *International Journal of Remote Sensing* 28 (5): 823–70. https://doi.org/10.1080/01431160600746456.

Mallinis, Giorgos, and Nikos Koutsias. 2012. "Comparing Ten Classification Methods for Burned Area Mapping in a Mediterranean Environment Using Landsat TM Satellite Data." *International Journal of Remote Sensing* 33 (14): 4408–33. https://doi.org/10.1080/01431161.2011.648284.

Marcos Martinez, Raymundo, and Kenneth A. Baerenklau. 2015. "Controlling for Misclassified Land Use Data: A Post-Classification Latent Multinomial Logit Approach." *Remote Sensing of Environment* 170 (December): 203–15. https://doi.org/10.1016/j.rse.2015.09.025.

Mas, J. F., and J. J. Flores. 2008. "The Application of Artificial Neural Networks to the Analysis of Remotely Sensed Data." *International Journal of Remote Sensing* 29 (3): 617–63. https://doi.org/10.1080/01431160701352154.

Mathworks Inc. 2016. *Matlab Statistics and Machine Learning Toolbox User's Guide R2016a; Matlab Neural Network Toolbox User's Guide R2016a*.

McGarigal, Kevin. 2015. *FRAGSTATS HELP*. University of Massachusetts, Amherst.

Meddens, Arjan J.H., Crystal A. Kolden, and James A. Lutz. 2016. "Detecting Unburned Areas within Wildfire Perimeters Using Landsat and Ancillary Data across the Northwestern United States." *Remote Sensing of Environment* 186 (December): 275–85. https://doi.org/10.1016/j.rse.2016.08.023.

Mountrakis, Giorgos, Jungho Im, and Caesar Ogole. 2011. "Support Vector Machines in Remote Sensing: A Review." *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (3): 247–59. https://doi.org/10.1016/j.isprsjprs.2010.11.001.

Ouyang, Y., and J. Ma. 2006. "Classification of Multi-spectral Remote Sensing Data Using a Local Transfer Function Classifier." *International Journal of Remote Sensing* 27 (24): 5401–8. https://doi.org/10.1080/01431160600823222.

Pal, Mahesh, and Paul M Mather. 2003. "An Assessment of the Effectiveness of Decision Tree Methods for Land Cover Classification." *Remote Sensing of Environment* 86 (4): 554–65. https://doi.org/10.1016/S0034-4257(03)00132-9.

Pelletier, Charlotte, Silvia Valero, Jordi Inglada, Nicolas Champion, and Gérard Dedieu. 2016. "Assessing the Robustness of Random Forests to Map Land Cover with High Resolution Satellite Image Time Series over Large Areas." *Remote Sensing of Environment* 187 (December): 156–68. https://doi.org/10.1016/j.rse.2016.10.010.

Pontius, Robert Gilmore, and Marco Millones. 2011. "Death to Kappa: Birth of Quantity Disagreement and Allocation Disagreement for Accuracy Assessment." *International Journal of Remote Sensing* 32 (15): 4407–29. https://doi.org/10.1080/01431161.2011.552923.

Puertas, Olga Lucia, Alexander Brenning, and Francisco Javier Meza. 2013. "Balancing Misclassification Errors of Land Cover Classification Maps Using Support Vector Machines and Landsat Imagery in the Maipo River Basin (Central Chile, 1975–2010)." *Remote Sensing of Environment* 137 (October): 112–23. https://doi.org/10.1016/j.rse.2013.06.003.

Qing, Jianjun, Hong Huo, and Tao Fang. 2010. "Supervised Land Cover Classification Based on the Locally Reduced Convex Hull Approach." *International Journal of Remote Sensing* 31 (8): 2179–87. https://doi.org/10.1080/01431161003636708.

Roelfsema, Chris, and Stuart Phinn. 2010. "Integrating Field Data with High Spatial Resolution Multispectral Satellite Imagery for Calibration and Validation of Coral Reef Benthic Community Maps." *Journal of Applied Remote Sensing* 4 (1): 043527-043527–28. https://doi.org/10.1117/1.3430107.

Tuia, Devis, Michele Volpi, Loris Copa, Mikhail Kanevski, and Jordi Munoz-Mari. 2011. "A Survey of Active Learning Algorithms for Supervised Remote Sensing Image Classification." *IEEE Journal of Selected Topics in Signal Processing* 5 (3): 606–17. https://doi.org/10.1109/JSTSP.2011.2139193.

Turner, Monica G. 2005. "Landscape Ecology: What Is the State of the Science?" *Annual Review of Ecology, Evolution, and Systematics* 36 (1): 319–44. https://doi.org/10.1146/annurev.ecolsys.36.102003.152614.

Weng, Qihao. 2012. "Remote Sensing of Impervious Surfaces in the Urban Areas: Requirements, Methods, and Trends." *Remote Sensing of Environment*, Remote Sensing of Urban Environments, 117 (February): 34–49. https://doi.org/10.1016/j.rse.2011.02.030.

Wenkai Li and Qinghua Guo. 2014. "A New Accuracy Assessment Method for One-Class Remote Sensing Classification." *IEEE Transactions on Geoscience and Remote Sensing* 52 (8): 4621–32. https://doi.org/10.1109/TGRS.2013.2283082.

Wuttke, S., W. Middelmann, and U. Stilla. 2016. “Active Learning with SVM for Land Cover Classification-What Can Go Wrong?” https://pdfs.semanticscholar.org/6cd0/0aaed7eec8e83611d272345399e99c6e8da2.pdf.

Zhang, L., L. Zhang, and B. Du. 2016. “Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art.” *IEEE Geoscience and Remote Sensing Magazine* 4 (2): 22–40. https://doi.org/10.1109/MGRS.2016.2540798.

Zhong, Yanfei, Liangpei Zhang, Jianya Gong, and Pingxiang Li. 2007. “A Supervised Artificial Immune Classifier for Remote-Sensing Imagery.” *IEEE Transactions on Geoscience and Remote Sensing* 45 (12): 3957–66. https://doi.org/10.1109/TGRS.2007.907739.

Zhu, Zhe, Alisa L. Gallant, Curtis E. Woodcock, Bruce Pengra, Pontus Olofsson, Thomas R. Loveland, Suming Jin, Devendra Dahal, Limin Yang, and Roger F. Auch. 2016. “Optimizing Selection of Training and Auxiliary Data for Operational Land Cover Classification for the LCMAP Initiative.” *ISPRS Journal of Photogrammetry and Remote Sensing* 122 (December): 206–21. https://doi.org/10.1016/j.isprsjprs.2016.11.004.

Appendix

Table A-1: Input images class distribution

Class→ *Image↓*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
1		46	427			107949	403	2064		
2	783	11024		245			14320	84030	487	
3	7			43			18769	92070		
4	214	2019	5025	4211	5	73024	1050	7800	17541	
5	10					94	2735	108024	26	
6	8310	2048	1438	219		77368	2272	1784	17450	
7	2494	1196		6125		93251	7540	48	235	
8	444	2299	10812			79857	13350	1864	2263	
9	767	8406	3246	81		51288	2420	9983	34698	
10	1363	43961	1171	734	17	3107	609	48494	11433	
11	2488	737	3862	30		98369	3842		1561	
12	1662	6086	1286	652	150	28584	2178	64502	5789	
13		22	9255	65		77953	23513	81		
14	88	3696	3021			91143	9853	2940	148	
15	1030	35989	261			981	652	71254	722	
16	3965	1688		206	597	24055	72473	7713	192	
17	3039	1449	53	252	570	67015	28682	7052	2777	
18	686	677				6736	93279	9434	77	
19	471	3617				19139	2833	78919	5910	
20	818			96		415	45060	55931	8569	
21	6				3064	12356	86116	9119	46	182
22	731	40068		1006		3834	17900	45477	1873	
23	1336	1925				727	21572	84802	527	
24	238	11327		345		8949		90030		
25	328	1047	7539	31		76036	6302	18116	1490	
26		769			271	2588	92381	13633	797	

Note: Those classes that have less than 122 instances in an image will result in less than 2 samples in the training set, and thus will be dropped by sampling. Those classes are underlined.

Table A-2: Best average overall accuracies and their coefficient of variation for 0.2% sample size

Classifier →	*NB*	*SVM*	*KNN*	*BagTE*	*ANN*	*DNN*		
ImageNo ↓						*1 Layer*	*2 Layers*	*3 Layers*
1	96.48% (1.19%)	97.38% (0.13%)	97.49% (0.34%)	97.60% (0.22%)	96.95% (0.77%)	97.07% (0.43%)	97.15% (0.48%)	97.02% (0.45%)
2	69.17% (4.54%)	79.02% (1.46%)	79.53% (1.54%)	78.01% (1.36%)	78.23% (2.29%)	77.99% (2.14%)	75.48% (3.11%)	76.42% (2.28%)
3	71.77% (6.13%)	88.19% (1.08%)	88.45% (1.71%)	86.93% (1.21%)	88.08% (1.92%)	87.13% (1.49%)	87.33% (1.82%)	87.76% (1.70%)
4	72.67% (1.70%)	77.54% (2.44%)	75.39% (2.02%)	77.12% (1.36%)	75.92% (2.35%)	77.01% (1.76%)	71.84% (3.36%)	73.46% (2.83%)
5	97.20% (0.37%)	97.46% (0.14%)	97.64% (0.22%)	97.51% (0.13%)	97.49% (0.90%)	97.25% (0.18%)	97.50% (0.57%)	97.55% (0.44%)
6	70.65% (5.77%)	75.28% (1.26%)	75.11% (1.76%)	76.08% (1.02%)	75.71% (1.46%)	75.87% (1.64%)	69.44% (3.68%)	69.87% (2.96%)
7	87.30% (1.29%)	88.80% (1.24%)	88.78% (0.67%)	88.91% (0.76%)	88.41% (1.04%)	88.55% (1.29%)	87.27% (1.35%)	87.52% (1.25%)
8	75.69% (2.97%)	81.79% (1.99%)	81.48% (2.58%)	81.76% (1.13%)	81.25% (1.93%)	80.22% (2.15%)	78.71% (1.28%)	78.28% (1.79%)
9	54.68% (6.71%)	57.58% (7.18%)	58.73% (4.09%)	59.55% (2.84%)	59.02% (2.79%)	55.75% (7.29%)	52.99% (3.81%)	52.58% (4.46%)
10	55.63% (7.61%)	71.57% (1.59%)	68.82% (3.01%)	70.39% (1.52%)	70.63% (1.88%)	70.67% (1.77%)	66.32% (2.80%)	66.26% (3.80%)
11	87.64% (1.81%)	90.65% (1.08%)	90.22% (1.37%)	91.51% (1.03%)	90.52% (0.63%)	90.03% (1.06%)	88.88% (1.40%)	88.60% (1.02%)
12	63.39% (6.86%)	72.95% (3.73%)	72.14% (2.47%)	73.56% (0.70%)	72.51% (2.20%)	72.45% (3.25%)	68.93% (2.84%)	67.41% (3.63%)
13	77.72% (5.11%)	81.26% (3.18%)	81.46% (0.95%)	81.89% (1.48%)	82.05% (0.65%)	82.13% (1.40%)	78.41% (3.06%)	78.80% (2.93%)
14	82.78% (1.10%)	85.94% (0.84%)	85.74% (0.68%)	86.00% (0.45%)	85.07% (1.13%)	85.65% (0.67%)	83.03% (1.52%)	82.54% (1.48%)
15	72.50% (5.98%)	82.64% (2.88%)	82.80% (1.28%)	82.31% (2.15%)	83.76% (1.50%)	84.12% (1.20%)	81.37% (2.20%)	81.36% (2.34%)
16	77.54% (3.30%)	84.25% (1.18%)	82.37% (1.49%)	83.03% (0.95%)	83.83% (2.73%)	83.99% (1.61%)	81.35% (2.41%)	81.06% (1.57%)
17	77.92% (2.97%)	82.47% (0.42%)	81.49% (1.30%)	82.56% (0.33%)	82.67% (1.00%)	81.98% (1.15%)	80.02% (2.08%)	79.43% (2.03%)
18	70.58% (9.85%)	82.92% (3.32%)	83.32% (1.66%)	83.40% (0.66%)	82.56% (1.59%)	82.78% (2.95%)	77.76% (4.54%)	78.55% (3.58%)
19	72.26% (6.21%)	78.66% (1.33%)	78.99% (0.77%)	79.08% (0.69%)	77.71% (1.38%)	78.09% (1.67%)	73.53% (3.34%)	72.53% (2.68%)
20	64.02% (3.71%)	72.01% (4.46%)	71.39% (3.51%)	70.73% (1.94%)	73.27% (2.08%)	72.80% (2.76%)	70.94% (3.54%)	68.92% (2.99%)
21	80.07% (1.43%)	83.47% (1.87%)	83.01% (1.15%)	84.36% (0.88%)	82.84% (2.05%)	82.68% (1.20%)	80.28% (2.38%)	80.71% (1.29%)
22	58.09% (5.66%)	66.27% (3.15%)	64.87% (2.81%)	65.18% (1.41%)	66.91% (2.26%)	66.53% (2.47%)	63.71% (3.26%)	63.10% (1.73%)
23	78.44% (3.39%)	84.48% (2.13%)	83.93% (1.80%)	83.12% (1.65%)	84.27% (2.77%)	83.26% (1.87%)	82.19% (1.87%)	82.28% (2.59%)
24	80.21% (2.85%)	85.64% (1.15%)	85.58% (1.17%)	86.17% (1.01%)	85.44% (1.46%)	85.45% (1.26%)	85.54% (1.40%)	84.57% (1.56%)
25	75.23% (1.84%)	77.69% (1.81%)	77.08% (2.67%)	78.49% (0.55%)	76.85% (1.39%)	77.13% (1.76%)	74.23% (1.89%)	73.71% (1.96%)
26	85.04% (2.23%)	91.09% (1.09%)	91.00% (1.20%)	91.03% (1.18%)	91.19% (0.92%)	90.66% (1.68%)	89.83% (1.53%)	89.19% (1.79%)

CHAPTER 3 (MANUSCRIPT 2):

Meta-analysis of deep neural networks in remote sensing: A comparative study of mono-temporal classification to support vector machine

Abstract

Deep learning methods have recently found widespread adoption for remote sensing tasks, particularly in image or pixel classification. Their flexibility and versatility have enabled researchers to propose many different designs to process remote sensing data in all spectral, spatial, and temporal dimensions. In most of the reported cases they surpass their non-deep rivals in overall classification accuracy. However, there is considerable diversity in implementation details in each case and a systematic quantitative comparison to non-deep classifiers does not exist. In this paper, we look at the major research papers that have studied deep learning image classifiers in recent years and undertake a meta-analysis on their performance compared to the most used non-deep rival, Support Vector Machine (SVM) classifiers. We focus on mono-temporal classification as the time-series image classification did not offer sufficient samples. Our work covered 103 manuscripts and included 92 cases that supported direct accuracy comparisons between deep learners and SVMs.

Our general findings are the following: i) Deep networks have better performance than non-deep spectral SVM implementations, with Convolutional Neural Networks (CNNs) performing better than other deep learners. This advantage, however, diminishes when feeding SVM with richer features extracted from data (e.g., spatial filters) ii) Transfer learning and fine-tuning on pre-trained CNNs are offering promising results over spectral or enhanced SVM, however these

pre-trained networks are currently limited to RGB (Red-Green-Blue or natural color) input data, therefore currently lack applicability in multi/hyperspectral data. iii) There is no strong relationship between network complexity and accuracy gains over SVM; small to medium networks perform similarly to more complex networks. iv) Contrary to the popular belief, there are numerous cases of high deep networks performance with training proportions of 10% or less.

Our study also indicates that the new generation of classifiers is often overperforming existing benchmark datasets, with accuracies surpassing 99%. There is a clear need for new benchmark dataset collections with diverse spectral, spatial and temporal resolutions and coverage that will enable us to study the design generalizations, challenge these new classifiers, and further advance remote sensing science. Our community could also benefit from a coordinated effort to create a large pre-trained network specifically designed for remote sensing images that users could later fine-tune and adjust to their study specifics.

3.1. Introduction

Artificial neural networks (ANNs) first started with cybernetics in the 1940s–1960s and led to the invention of the first single neuron model named perceptron (Rosenblatt, 1958). Being a data-driven model with the ability to simulate arbitary computing functions through optimization, ANNs found a wide range of applications. The next major breakthrough happened in late 80's with the invention of back-propagation and a gradient-based optimization algorithm to train a neural network with one or two hidden layers with any desired number of nodes (Rumelhart et al. 1986). The back-propagation method has worked well for non-deep structures (1-2 hidden layers) but gradient-based training of deep neural networks (DNNs) could get stuck in local minima or plateaus due to the dramatic increase in number of model parameters and vanishing of gradients during backpropagation (Bengio, 2009). There is no standard definition to label a neural network as deep, but it mostly refers to network of two hidden layers or more, used to automatically extract a hierarchical set of features from data. Compared to 1-2 layer structures, DNNs promise to provide more compact models for the same modeling capabilities (Bengio, 2009). However, the high node number of DNNs made it difficult to train and optimize in a practical manner.

The seminal work of Hinton et al. (2006) showed that unsupervised pre-training of each layer, one after another, could considerably improve results. This layer-wise training approach, named greedy algorithm, was the key that opened new avenues to deep neural networks. The greedy algorithm could also be followed by a fine-tuning process, in which the entire network is tuned together using backpropagation, but this time from a much better starting point. Deep network theories and practices have expanded considerably during the last decade. It has resulted in establishment of some major network types (with continuous enhancements) and numerous applications in different domains. In close relationship with image processing and computer vision, remote sensing (RS) is one of many areas that deep learning is targeting.

Generally and following discussion in L. Zhang et al. (2016), we can categorize remote sensing applications of deep learning into four groups: 1) RS image pre-processing, 2) scene classification, 3) pixel-based classification and image segmentation, and 4) target detection. For image pre-processing tasks, we can name pan-sharpening, denoising, and resolution enhancement as major applications. Scene classification is done based on some extracted features from a scene, which the deep networks are assumed to be good at. The non-deep approaches normally use some handcrafted features extracted from the scene to feed the classifier (SVM, KNN, etc.) and predict the scene type. Deep networks have opened the door to direct use of spectral and spatial information together to generate a richer set of features automatically. This automatic extraction increases the potential for good generalization and scalability of this method compared to handcrafted features. Handcrafted features tend to be tailored closely to a specific case and application and possibly perform better than any automatic system, but because of this specificity they cannot be easily or successfully generalized to another cases/studies. This type of work is closely related to image recognition task but for categorization of remote sensing scenes (such as agricultural field, residential area, airport, parking lot, etc.), therefore sharing network configurations between computer vision and remote sensing applications is common here. Pixel classification and segmentation (or semantic labeling) are similar to scene classification but operate at the pixel rather than scene level, and produce a thematic map instead of a single category index. This is perhaps the most studied RS application and deep networks have shown performance benefits due to their ability to co-process spatial and spectral data easily, especially for hyperspectral images. Our main target in this paper is to focus on image or scene classification - we do not address other applications. In addition, we focus on mono-temporal classification as the time-series image classification is still in its infancy. Target or object detection is generally an extension to the three aforementioned groups, where specific objects defined by their shape or boundary are extracted from an image. This field has found many useful but challenging applications in high resolution and real time image/video processing.

Following the explosive growth of new algorithmic developments and case studies in deep learning RS applications in the past 3-4 years, several review manuscripts have been published (Ghamisi et al. 2017; Xia et al. 2017; L. Zhang et al. 2016; or P. Liu et al. 2017). The majority of these reviews are descriptive and do not offer a quantitative assessment of deep learning benefits building on the extensive available comparisons in the literature. The overal goal of this work is to bridge this knowledge gap by undertaking a meta-analysis comparing deep and non-deep classification algorithms through a meta-analysis of published research.

Other meta-analysis works exist but they do not examine explicitly deep learning benefits. For example, Khatami et al. (2016) grouped all neural network types under one category and did not distinguish deep networks from other implementations. Ma et al. (2017) conducted similar meta-analysis focusing on object-based classification (thus excluding pixel-based ones) without separating deep learning methods. There are some other papers that review deep learning architectures in general such as Deng (2014) or W. Liu et al. (2017), or for specific type of data, such as Camps-Valls et al. (2014) on hyperspectral data classification. These works also lack quantitative comparisons using a meta-analysis approach.

The overarching goal here is to provide readers with the “big picture” of current research and build on the collective knowledge of published works to assess deep learning benefits in remote sensing. To undertake the proposed meta-analysis task, we reviewed major research papers and built a database of case studies of deep network applications in the remote sensing field while extracting main network and data characteristics. This database was analyzed to identify deep learning classification performance and its distribution across these network (e.g. network complexity) and data characteristics (e.g. spatial resolution). We expect this analysis to provide a knowledge baseline as the remote sensing community further incorporates deep leaning in related activities.

The structure of the manuscript is as follows. A brief overlook of deep network types is presented in section 3.2 along with key introductory references. A summary table is also provided to describe extracted parameters for each research paper. Section 3.3 starts with introducing a descriptive statistics and summarizing design ideas encountered in the selected resesrch papers and used datasets. Then we provide our main comparative analysis and discuss important research questions about parameters effect on network performance. The last section provides concluding remarks.

3.2. Methods

In this section we first describe three DNN methods that have been popular in Remote Sensing (RS) tasks. Section 3.2.2 contains an explanation on the paper database and associated characteristics and metrics used in the comparative accuracy analysis between DNNs and non-deep methods.

3.2.1. Summary of popular deep neural networks in remote sensing

The deep learning paradigm is concentrated on automated hierarchical feature extraction. Numerous methods and their modifications have been devised along the past years. Here we briefly introduce the three most widely used structures which were used in our identified studies. More detailed descriptions of each structure can be found in many machine learning textbooks, for example Bengio (2009) and Goodfellow et al. (2016), or tutorials such as Le (2015) and Deng (2014). Zhu et al. (2017) and L. Zhang et al. (2016) also provide tutorials for deep learning for remote sensing applications.

Deep networks have been developed to enhance and enrich data representations in an automated and intelligent manner. A good representation is, of course, dependent on the specific application and should be learned from training data. One important deep network category in this class is based on Autoencoders (AEs). The idea behind an autoencoder is basically an encoder-decoder network to regenerate the input as accurately as possible in its output. Under specific conditions, the encoder part works as a good feature extractor and can be stacked to build deep networks named Stacked Auto Encoders or SAEs (the decoder part is not used). The imposed condition on objective function is typically a form of sparsity, but other variants are also studied. To put it simply, AE learns a deterministic representation of the input by minimizing a cost function based on the difference between input and the regenerated one at the decoder output. This learning takes place using gradient-based optimization and standard backpropagation techniques. AEs are well suited to unsupervised learning and can be trained layer-wise, possibly followed by a supervised fine-tuning phase of the entire network. For a good overview of autoencoders with some work examples and executable codes see Andrew Ng's Deep Learning tutorial at http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial. Vincent et al. (2010) also provide more details on autoencoders and unsupervised learning.

Another way of thinking about data representation is to learn the statistical distribution of input, i.e., a probabilistic approach. This approach has led to Generative models or Structured Probabilistic Models. Deep belief network (DBN) based on stacking layers of Restricted Boltzmann Machine (RBM) is the most popular variant for RS applications. Here the aim is to minimize the Boltzmann cost function, to maximize "the similarity (in a probabilistic sense) between the representation and projection of the input" (Singhal et al. 2016). This optimization does not use an assumed output, so a different algorithm (contrastive divergence) is required to train the neurons. However, similarly to the autoencoder, training is unsupervised and, more important, it can be done in a greedy layer-wise approach for a stack of layers. This layer-wise approach was devised by the seminal work of Hinton et al. (2006) and later implemented by both SAEs and DBNs. Therefore SAEs and DBNs are often discussed together in the literature (e.g. Vincent et al. 2010). When trained, the network can provide extracted features for the new data to be classified. Tutorials on RBM and DBN are available through the internet, for example see https://deeplearning4j.org/restrictedboltzmannmachine, which includes executable codes.

The third type, which is the most used structure in recent years, is the convolutional neural network (CNN). Inspired by the human visual system and designed to process images, it has limited connection to only adjacent neurons in each layer, with the same connection weights for each neuron within each layer. It may include down-sampling in each layer, which reduces the processing resolution but adds translation invariance property to the network. Each layer's output is typically named a map and it is generally desired to have multiple maps generated at each layer. Here the filter weights are tuned typically by supervised training, as the limited number of shared parameters in each layer (compared to a fully connected network) allows it. There are also some pre-trained large network structures publicly available for use and fine-tuning them for specific applications is another common approach. For a university course on convolutional neural networks readers are referred to http://cs231n.stanford.edu/. Zeiler and Fergus (2014) also provide a discussion on visualization and understanding of the internal CNN workings.

Working with sequence data is another important type of remote sensing works, particularly on three bases: studying hyperspectral signal variations and analyzing their dependencies; adding the time dimension as another data element to explore land use feature patterns (profiles) and use them in classification; and pursuing detection of changes in land cover or land use by processing time-series data. Neural networks – and specifically Recurrent Neural Networks – are gaining momentum for these applications but the number of published papers is still low. These networks are promising with new modifications such as adding more powerful and deep memory cells (see for example Lyu et al. 2016, Mou et al. 2017, Rußwurm and Körner 2017, Rußwurm and Körner 2018, Ndikumana et al. 2018, Niculescu et al. 2018, or Sharma et al. 2018). However, we did not consider sequence data applications in our paper due to lack of enough data and our focus was only on feed-forward networks and its three main variants: SAE, DBN, and CNN.

3.2.2. Comparative performance database creation

Our overarching goal is to look at the analyzed DNN case studies and compare them together and to a well-known non-deep classifier, Support Vector Machine (SVM). SVM served both as a representative for non-deep classifier to compare with deep networks, and as a baseline to compare different DNN architectures. SVMs were selected as the benchmarking algorithm because: i) they were found to be the best non-deep performing classifiers in an extensive comparison of published work (Khatami et al., 2016), and ii) the majority of DNN papers found in this review chose to include SVM as the main benchmark, thus validating our decision. We also examined accuracy trends across data and method characteristics. Direct comparisons of published works were not feasible due to variances in data types, sampling design, algorithmic details, and test metrics. Therefore, we concentrated on aggregating results from manuscripts where accuracy metrics are reported mutually under common conditions for deep and non-deep implementations. This database was then used to do comparative meta-analysis and other quantitative statistical analyses.

The result was 103 research papers from 2014 until Nov. 2018 covering 183 case studies that include deep learning-based classification, 92 cases of which supported direct comparisons of accuracy to SVM. The main characteristics of these case studies are summarized in Appendix A, Table A-1, with each column of the appendix table defined in Table 3-1 below. These parameters reflect the most important aspects of the research design and we used them to present the discussion of our research questions in the subsequent sections. We treated each data set in a research paper as a separate case, because the output result and possibly the network structure may vary per case in any single paper.

Table 3-1: Parameters collected on each case study

Reference	Citation code for the referenced research paper
Network Type	One of below categories: - Convolutional Neural Network (CNN), - Deep Belief Network (DBN), - Stacked AutoEncoder (SAE)
Learning strategy	One of below categories: - Unsupervised - Unsupervised & fine-tuning - Semisupervised - (fully) Supervised - Transfer learning - Transfer learning & fine-tuning
Number of parameters	Number of trainable network parameters, i.e. weights and biases of network neurons and connections. We manually created this number to approximate network complexity.
Dataset	Name of dataset used for the research, including: - Brazilian coffee, NWPU-RESISC45, RSSCN7, UC Merced, and WHU-RS19: 3-band images used in scene classification, - Indian Pines, Houston, Kennedy Space Center, Pavia University, Pavia City Center, and Salinas: hyperspectral images used in pixel classification, - ISPRS Potsdam and ISPRS Vaihingen: very high resolution images used in image segmentation, - Others: Remaining datasets.
Spatial resolution	Dataset spatial resolution expressed through pixel size.
# of channels	Number of spectral and auxiliary channels.
Training proportion	Proportion of training data size in reference dataset.
Metric type	Metric used for reporting performance in research case, including Overall Accuracy, Average Accuracy, Average Precision, F1, Kappa, etc.
Deep network result	Best reported value of network classification performance
SVM results	Best achieved performance of SVM implementation

One of the most important parameters in network specification is the number of network parameters which reflects network complexity. This is typically a surrogate of network depth and width. It is expected that a bigger network would be more powerful, but the network architecture and way of processing (reflected in other columns of the table) greatly impacts this performance. Therefore, it is not unexpected that a smaller but more elegant network outperforms a larger one in obtained accuracy. For example, in classifying the ISPRS Potsdam and Vaihingen datasets, Maggiori et al. (2016) achieved more than 1% better accuracy than Volpi and Tuia (2017) by a network having around 1/10th of their network size. This number was mostly calculated from network parameters given in the cited paper but in some cases it is given in the cited paper as well. In cases that given information was not sufficient or ambiguity was not cleared by correspondence, the entry was left blank. This number included parameters in as many network branches as implemented, but it did not include parameters associated with additional stages of combination or fusion with other data or algorithms. It also counted the network layers parameters up to the last layer before the final classifier, which was typically a Softmax layer but SVM was also used. In around 70% of our cases the deep network was followed by a Softmax classifier, therefore we dropped the final classifier type from our list of parameters.

The learning strategy column was another important network parameter. It does not point to the final classifier training as it is always supervised, but shows the methodology for determining network parameters. The supervised learning was the most common approach in deep networks. It could also have different variations in the form of cost function or optimization procedure, or being enhanced by data-driven techniques such as active learning. Those advanced cases were designated as supervised+ in our database. The fine-tuning options show the cases when network parameters are fine-tuned after an initial unsupervised learning or transferred from a pre-trained network in transfer learning. Transfer learning is available to CNN only. DBNs were usually limited to unsupervised & fine-tuning type, while SAEs were used with both unsupervised learning techniques. Semi-supervised learning was also used in some cases, which is a strategy for using both labeled and unlabeled data in optimizing the network cost function.

Spatial resolution in our collected research cases varied from 5cm for very high resolution (VHR) imagery to 30m for Landsat, left blank if not provided. The number of channels shows the ones that have been actually used in the experiment (some channels have been set aside for their low quality in some studies but not in the others). Note that in some cases the input channels were processed and dimensionality was reduced (mostly employing principal component analysis) and the result was applied to the network, but we did not mention this dimensionality reduction in Table A-1, although we took it into consideration when calculating the number of parameters and considered the network in its actual tested configuration. There were two cases of using Landsat and one case of MODIS imagery that has been indicated in Table A-1 separately due to importance of these data sources. Although from one hand they are of less attention today because of their inferior spatial resolution, but from the other hand they are of interest for their rich temporal dimension in time-series analysis. Data fusion from different sources is also experiencing growing attention, especially adding height data through Digital Elevation Models (DEM). We discuss this further in the design options (section 3.3.3) but an in-depth analysis of this issue was outside the scope of this work.

Another important factor in network prediction performance was the data training size. More training data typically leads to better network generalization, but in many cases the labeled training data was very limited. The corresponding column in Table A-1 shows the rounded proportion of (labeled) training data samples to the entire reference data set, varying from as low as 0.1% to 90%. We refer to it as “training proportion” hereafter, and consider the proportion in one single run of the network, therefore a cross-validation scheme does not change the value in the table from a similar hold-out.

The reported classification accuracy (overall or average) value was the best performance reported for the reference dataset in each case. It was reported as a number between 0-100 except for the Average Normalized Modified Retrieval Rank (ANMRR) metric. Although overall accuracy is an aggregate metric and cannot show important class-dependent performance values, but it is still the most widely used metric due to its simplicity and general applicability. Even though in some cases more detailed evaluations were provided along with overall accuracy, due to different experimental designs and data structures in our meta-analysis, these detailed metrics were not widely comparable and therefore class-specific measures were not included.

In some cases, an additional pre- or post-processing step complements the deep network to enhance the performance, for example merging the resulting map with an auxiliary segmentation result, adding a conditional random field (CRF) layer for edge enhancement, or object-based processing. These methods differ largely in implementation details and experiment setup so cannot be directly compared to assess the processing gains; we provided more details on them in section 3.3.3.

Although the chosen non-deep methods varied greatly in type and options from paper to paper, there were still numerous cases where DNNs are compared to an SVM-based implementation, with Random Forest and KNN being the next classifier types used by much less frequency in our observed cases. Therefore, we chose those papers reporting on SVM results as the candidates for doing our quantitative analysis (in the next section). SVM is a good choice for benchmarking because it is a well-established and proven classification tool with generally superior performance (Mountrakis et al. 2011; Khatami et al. 2016). Note that in remote sensing image or scene classification tasks, we are generally interested in both feature generation and classification. Neural networks can do both automatically – and deep networks put more stress on the feature extraction task – but SVM classifiers should be fed with features already generated by another algorithm. The SVM implementation itself may vary between processing the raw pixels data or some secondary handcrafted spectral/spatial features derived from data. To ensure a more fair comparison we separated these two cases due to the potential important impact of working with features instead of raw data. Clearly, there are many variations and methods for handcrafting features and each paper may include a different set of methods for comparison, so we could not go into their implementations detail and a detailed comparison. Furthermore, SVM optimization methods varied by hyperparameters or kernel choice. However, we assumed (and it was also stressed in some papers) that the authors reported their best SVM performance after tuning parameters.

We should mention here that although our meta-analysis covers many different cases, each case had almost a unique setting of the above parameters and therefore our analysis is naturally limited in depth and statistical richness. Our objective was to study general trends and for the first time in the literature offer a quantitative meta-analysis of DNNs in remote sensing applications. Our quantitative analysis did not go into a detailed analysis of the effect of every design option due to lack of data.

3.3. Results and discussion

3.3.1. Descriptive statistics

Table 3-2 provides information on case studies distribution by year, network type, spatial resolution and input dimensionality. Note that some manuscripts may contain more than one study, and spatial or input dimensionality information was not always available.

Table 3-2: Basic statistics of collected case studies

Year	2014	2015	2016	2017	2018
Number of publications	4	27	21	22	33

Network Type	CNN	DBN	SAE
Number of cases	150	9	25

Dataset spatial resolution	< 30cm	30cm ~ 3m	> 3m
Number of cases	23	88	48

Spectral and auxiliary bands	1-3	4-10	11-99	> 100
Number of cases	59	48	1	70

There is an increase in research papers on deep networks for remote sensing classification applications after 2014, continuing to date. CNN was the most commonly used network type, then SAE followed by DBN. Most of the datasets were either hyperspectral (>100 spectral channels) or less than 10 channels. Just one case study had spectral channels between 10 and 100. Hyperspectral datasets were of high spatial resolution (around 1m) so sit in the middle group of spatial resolution category. Very high resolution ones (<30cm) were mostly available in RGB with possibly adding Near-Infrared (NI) band and/or DSM data to it, with just one very recent case incorporating a drone-based six band experiment at spatial resolution of 4.7 cm. More information on datasets will be given in the next section.

3.3.2. Datasets in the selected case studies

A wide variety of approximately 60 different datasets were used throughout the selected case studies. They included frequently used datasets along with datasets selected from public sources such as Google Earth, QuickBird, WorldView, Landsat archives and proprietary data sources.

Cases that have been used more than twice in our review are listed in Table 3-3. The table includes both scene and pixel classification applications as indicated in the last column, and the “labelled elements” column should be interpreted accordingly. Among them, the Brazilian Coffee, NWPU-RESISC45, RSSCN7, UC Merced, and WHU-RS19 have been used for scene classification while the others concentrated on pixel classification/image segmentation. It is important to note a significant limitation. While still being extensively used even in papers from 2018, some of the commonly used datasets are old and outdated: the major issue is their small size compared to datasets with millions of elements typically used in computer vision and other artificial intelligence studies. This issue has been partly addressed by some very high resolution datsets such as ISPRS Vaihingen and Potdam datasets, which became a standard test bench for newly arrived (mostly CNN-based) networks. Furthermore, hyperspectral cases are limited to a single scene and some datasets cover a very small geographic area, which limits the generalization ability of the obtained results. Again, there is a new dataset presented through IEEE GRSS contest in 2018 which consists of a relatively big area of 1.4 km^2 covered by both very high resolution (5cm) RGB and high resolution (1m) multispectral data. However, none of our reviewed articles were based on this new dataset (Le Saux et al. 2018).

It seems that there is a still a need to create more large and rich datasets for remote sensing applications in different spatial and spectral resolutions. Preparing datasets for tackling temporal applications is another important issue, which is even more restricted than other applications. However, the decision to pick specific labels and the procedure for creating ground truth maps is very application-specific and it needs more serious discussion in the community. Provision of auxiliary data (commonly DSM based on LiDAR) is also an important enhancement that is available in few datasets and should be encouraged.

Table 3-3: Specifications for most frequently used datasets

Dataset name	*Sensor platform*	*Dataset size*	*Image size (pixels)*	*Labelled elements*	*Spatial res.*	*# of spectral and aux. channels*	*# of classes / classification task*
Brazilian coffee	SPOT	50000	64x64	50000 scenes		RG + NI	3 class, but highly imbalanced / scene
Houston (2013 GRSS)	ITRES-CASI	1	1905x349	15029	2.5 m	144 + LiDAR	15 class / pixel
Indian pines	AVIRIS	1	145x145	10249	20 m	220	16 class / pixel
ISPRS Potsdam	Aerial photo	38	6000x6000	24 full images (of 38)	5 cm	RGB + NI + DSM	6 class / pixel
ISPRS Vaihingen	Aerial photo	33	circa 2500x2000	16 full images (of 33)	9 cm	RG + NI + DSM	6 class / pixel
KSC (Kennedy Space Center)	AVIRIS	1	512x614	5211	18 m	224	13 class / pixel
NWPU-RESISC45	Google Earth images	31500	256x256	31500 scenes	0.2 m ~ 30 m	RGB	45 class, 700 samples per class / scene
Pavia Center	ROSIS	1	512x614	148152	1.3 m	103	9 class / pixel
Pavia University	ROSIS	1	610x340	42776	1.3 m	103	9 class / pixel
RSSCN7	Google Earth images	2800	400x400	2800 scenes		RGB	7 class, 400 samples per class / scene
Salinas	AVIRIS	1	512x217	5348	3.7 m	224	16 class / pixel
UC Merced	USGS satellite imagery	2100	256x256	2100 scenes	1 ft	RGB	21 class, 100 samples per class / scene
WHU-RS19	Google Earth images	950	600x600	950 scenes	0.5 m	RGB	19 class, 50 samples per class / scene

3.3.3. Network design options

In terms of network optimization for deep networks the simplest way is to change the network depth (number of layers) and width (neurons per layer). Additional modifications include changes in the activation function, the type of classifier or the training strategy (supervised/unsupervised). Looking beyond these fairly common adjustments, we presented in Table 3-4 a descriptive summary of the most important design innovations we encountered. The table is organized to titles summarizing the main design point, followed by specific design ideas in each section. The number of papers using each option is provided to suggest popularity. Some design options were not exclusive to a specific network type (e.g. network mixing options), while some others may only be applicable to specific network types (e.g. fully convolutional network). The classification task type may also require special provisions. For example, in image segmentation the objects' boundary alignment is of primary concern, while in scene classification this is not important. This makes edge enhancement techniques more relevant to the former application than the latter. Because each design idea was presented and tested in a unique setting with single or multiple choices of listed options on different datasets and compared with different non-deep rivals, comparison between different design ideas and quantitative analysis of their merit was not possible. However, we discuss general findings on design options below and our intent is that this table act as a preliminary catalog and guide future research, either through gap analysis or through frequently-implemented method identification.

Dense (fully connected) networks: This CNN-type network is the de-facto network of choice for very high resolution classification and almost all of the image segmentation works – particularly experiments with ISPRS Potsdam and Vaihingen datasets. The competition in this field is extensive, and some of the most popular networks have been implemented in this category to win the ISPRS competition. It is always possible to run the entire network and classify the image pixel by pixel, but it means a huge redundancy in calculations and therefore a direct map-to-map conversion (which typically contains chain of downsampling and then upsampling) is preferred. Upsampling design is a hot topic and each paper tried to find a better way to do it. Edge enhancement and additional segmentation techniques have also been examined by different approaches to enhance the result (we will refer to it in another paragraph in this section).

Multiscale capability options: This issue is of a particular interest in CNN networks due to the limited connectivity of their neurons to the previous layer, but other network types may also use it when they use a sliding window mechanism in their input layer. In custom CNN networks the multiscale filters with or without skip links (forwarding features/scores from one layer to another non-adjacent layer) is a promising choice, but this option is not typically available for pre-trained networks.

Network mixing options: There were a variety of practices for this option as listed by categories in Table 3-4. The most frequent option was ensemble of different networks or using a parallel network on different bands (especially when the additional input in form of DSM or LiDAR is provided). Parallel 1-D (spectral) and 2-D (spatial) network was also found in some cases, but other forms of spectral/spatial input combination were more frequent (we discuss it in a later paragraph). As the use of pretrained networks becomes more common, parallel networks are the natural way of overcoming the imported network input limitation to RGB channels.

Training options: Engineering the input data was the most frequent form of enhancing training operations in deep networks, which was implemented in a variety of methods. The simplest case was to crop, rotate and flip the input patches (basic data augmentation) or adding virtual samples to the input data (particularly used for making input set more balanced). Recently, active learning and interactive sample selection approaches are gaining more attention. There were other specially designed algorithms used to enhance data quality, such as salient patch selection; or specialized methods for calculating network parameters, such as calculation of neurons weights by clustering or PCA decomposition instead of training.

Multimodal processing: Deep networks in remote sensing classification started with processing spectral components but quickly evolved to process other dimensions of data as well. Data processing in spatial context is now typical, especially with CNN, and joint spectral-spatial processing in 3-D convolutional filters is popular. Before that, other techniques such as averaging over spatial dimension or PCA compression of spectral dimension were common, but newer 3-D architectures have shown slightly better performance in our case studies. The newest trend in multimode processing was to incorporate sequence/temporal processing, for example by treating spectral component of hyperspectral imagery as a (correlated) sequence, or working on time-series of spectral-spatial data cubes.

Other features: In addition to the above, we identified numerous special algorithms and techniques throughout our survey that are organized in this section. In earlier studies we saw some cases of performance improvement by feeding network with handcrafted features, but it seems to be an obsolete idea now. Object-based classification, image segmentation, and additional MRF/CRF processing have been attractive research areas from the early days and still draw a lot of attention. Parallel to that, developing and applying newer and more complicated network modules (for example residual modules in CNN or LSTM in RNN) in RS applications are increasing trends. In the reviewed cases, newly emerging modules seem to have the upper hand at the expense of much larger and more complicated networks. The other options found are:

- CRF postprocessing of deep network predictions to delineate and enhance object edges.
- Initial segmentation and creation of superpixels to feed deep network.
- Merging of deep network predicted map and segmented or CRF/MRF generated map based on network prediction confidence.
- Other pre/post processing methods (e.g., GLCM/gabor filters).

There was no dominant method among the aforementioned techniques and new methods are continuously emerging.

Table 3-4: Network options and design innovations in collected papers

Option	*Frequency*
Making dense (full resolution) output options (for CNN):	
Fully Convolutional Network (convolution and deconvolution)	13
No down-sampling	1
Multiscale capability options:	
Getting multiscale input	10
Using multiscale kernels (filters)	7
Skip links (forwarding features/scores from one layer to another non-adjacent layer)	8
Network mixing options (fusion/aggregation method varies by case):	
Parallel handcrafted features	3
Parallel 1-D and 2-D convolutional networks	3
Parallel networks on different band combinations or sensors	9
Cascaded networks	3
Parallel (Ensemble) of different deep networks	10
Training options:	
Salient patches selection to train/test network	2
Active learning or iterative feature selection (removing inferior features)	4
Data augmentation or adding virtual samples to the input data	15
Other specialized methods	9
Multimodal processing:	
3-D processing modules	7
Spatial averaging/filtering over a neighborhood for spectral+spatial input generation	2
PCA dimensionality reduction and spectral+spatial input generation	9
Sequence data/temporal processing	4
Other specialized methods	2
Other features:	
Feeding network with handcrafted features (not raw data)	4
Optimizing input band selection with genetic algorithms	1
MRF/CRF processing or boundary detection	7
Denoising SAE implementation	3
Initial and/or final data/feature filtering or segmentation to enhance object discrimination	12
Sparse or other type of coding to create codebook after feature generation and classify the code	4
Emerging network modules (e.g., residual module, inception module, LSTM)	10

3.3.4. DNN vs. SVM classification accuracy comparisons

This section focuses on classification accuracy comparisons between DNN and SVM methods. We focused on SVM comparisons since the majority of the manuscripts we reviewed selected SVM as their benchmark. The SVM choice over other methods (e.g., RF) is further supported by a previously conducted meta-analysis, where SVM was found to outperform other (non-deep) methods (Khatami et al. 2016). For a case study to be considered in this section, both methods should have been tested on the same dataset and results reported in the form of average or overall accuracy.

Deep networks are usually designed to employ high volumes of available spectral and spatial data. However, in many of the selected cases of pixel classification, the authors compare DNNs to simple spectral processing by SVM or other non-deep rivals, thus providing an unfair advantage to DNNs as they also incorporate spatial information. Knowledge of feature generation details may not be a primary concern in deep networks as it is optimized automatically by the network, but finding the best method for feature generation to feed an SVM is not a straightforward task that often requires trial and error for each dataset. On the other hand, designing the best deep networks out of standard basic schemes is not a trivial issue and we see new designs continuously arising. To further inform readers in all figures in this section, we marked the cases with enhanced feature generation for non-deep classifier (SVM) with a black circle in below figures to separate them from cases using exclusively spectral information in their SVM implementation. Initial summary results are depicted in Figure 3-1. As there were just two cases with SVM accuracy below 70%, we set our scale to start from that and omitted those two in display to reduce the congestion on the upper accuracy values in pictures.

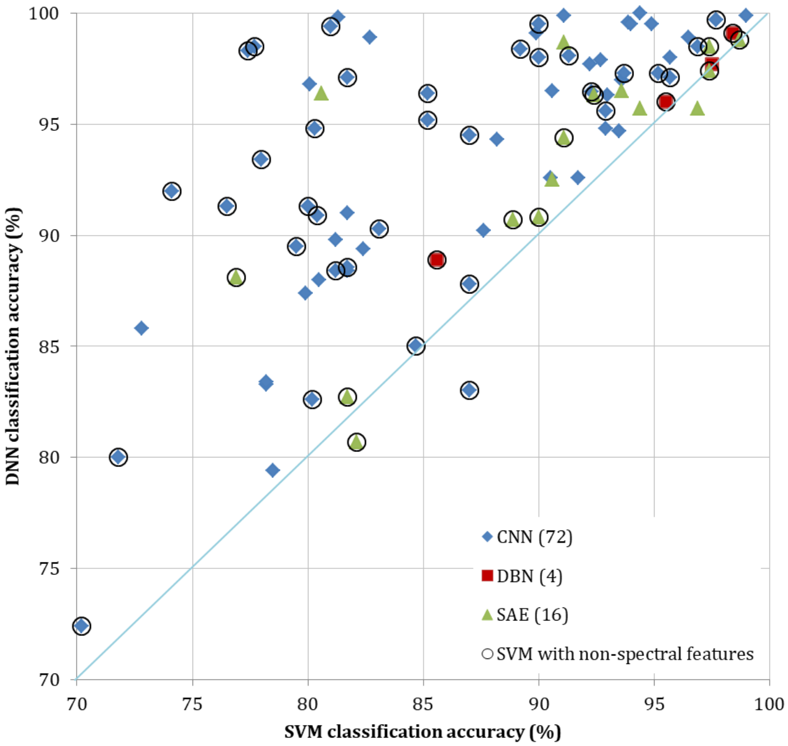


Figure 3-1: Comparative performance distribution of DNN vs SVM

In general, deep learning approaches offer consistently better results than SVM methods. The reported improvement (difference in accuracy value) can be as high as 30% for CNN[1], 16% for SAE and 3% for DBN. The DBN values may not be very representative due to the scarcity of this network type application in remote sensing, but this lower application rate itself can be a sign of its lack of merit and/or underlying complexity. CNN accuracy benefits are often attributed to the integrated processing of spatial and spectral information, while for SAE or DBN benefits involve specific experimental design. As an example, one author used the average values of a neighborhood around each pixel (to be classified) for each band and added it to the central pixel's own data, then fed the SAE or DBN with this composite data vector. It was also

[1] This difference was reported where SVM accuracy were below 70% and therefore was omitted in Figure 3-1.

common in hyperspectral image classification to process input image with dimensionality reduction techniques such as PCA first, and then build the spatial information to be added to the original central pixel for classification.

CNN has the ability to preserve the spatial relationships while processing through different layers, as spatial filtering takes place in each layer without flattening data to a row vector. In SAE or DBN implementations, the spatial information was flattened and concatenated to the spectral data at the network input, and although the spatial information is implicitly included, the spatial relations between vector values are lost (Y. Li et al. 2017; Yue et al. 2015; Basu et al. 2015). However, the CNN spatial coverage is limited to a neighborhood of fixed size at the input, increasing step by step while resolution is reduced accordingly in pooling layers. This issue was restrictive to scale-dependent information, although it could be remedied to some extent by a multiscale structure (for example see X. Chen et al. 2014, Zhao et al. 2015, or Zhao and Du 2016). Other networks are not inherently limited by these rules, though the strength of spatial relationships is generally reduced with the increasing distance from the central point according to Tobler’s law in geography (Tobler, 1970). It is also important to consider that the SAE and DBN methods were trained in an unsupervised fashion while the CNN method followed a supervised approach. Therefore, the CNN implementation might be advantageous due to the incorporation of labeling information (Y. Li et al. 2017; Shi and Pun, 2018). SAE and DBN were also trained in a greedy layer-wise fashion that may limit potential learning opportunities; each layer’s parameters are fixed when tuning the next layer. Joint training of layers for SAE and DBN has been proposed in Zhou et al. (2014) and reported to perform better than typical greedy layer-wise approach, but it was not of common use.

To investigate further DNN accuracy gains, we examined their distribution across five contributing factors, namely the DNN learning method, the network complexity, spatial resolution, input dimensionality and training dataset proportion. Due to the low number of case studies and variation of design parameters and datasets employed in different studies we did not report a multivariable regression model. Instead, we limit our analysis to single factor distribution plots.

Distribution across learning methods. Figure 3-2 and Figure 3-3 present accuracy comparisons for different learning methods for CNN and SAE, respectively. DBNs had a single learning option therefore they were omitted from this analysis. Starting with Figure 3-2 and CNN methods, the majority of cases have used supervised training or its enhanced version shown as supervised+ (cases with different cost function or optimization procedure, or being enhanced by data-driven techniques such as active learning). CNNs using supervised learning was mostly compared to spectral SVM and tend to offer higher relative gains in more complex classifications, where the corresponding SVM accuracy is lower. This result is expected due to integration of spatial data in CNN and lack of it in spectral SVM. As mentioned in Zhao et al. (2017), there are similarities between low-level features in different classes that cannot be resolved solely by the spectral components and integration of spatial data is required (an example is the road and building roof pixels in an aerial image). As seen in the upper right corner of the graph, in cases where the SVM was fed with enhanced features, the performance can be fairly close to the supervised learning DNN cases. One benefit of deep networks is the flexibility to build the features automatically and match them to the specific dataset under study, contrary to handcrafted features that should be selected among many variants for SVM or other non-deep classifiers.

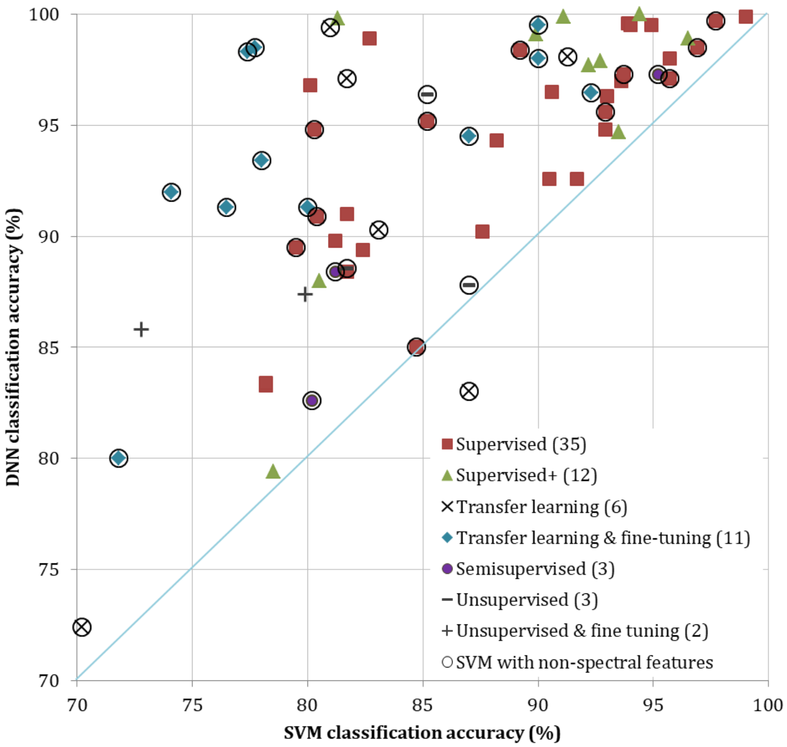


Figure 3-2: Comparative performance distribution across learning methods for CNN

Transfer learning in CNN also offers some improvements visible in Figure 3-2 – and especially when combined with fine-tuning – over enhanced SVM. Base networks are typically taken from models developed and trained in computer vision industry and will not be introduced here. The most widely used model in our reviewed cases was AlexNet (9 cases), followed by VGG-16 (8 cases), VGG-M (5 cases), and GoogLeNet (4 cases). Other networks have also been applied with lower frequency such as ResNet, other VGG-series networks, SegNet, Overfeat, and CaffeNet. Their improved performance over supervised learning CNN cases could be attributed to the fact that supervised CNN cases are usually custom designed and small in size compared to CNN networks used for transfer learning, therefore they may not be much more powerful than a SVM fed with enhanced features. A large fully supervised network may achieve considerable improvement over an enhanced SVM, but the computational budget might be prohibitive and the risk of overfitting is high. Transfer learning, however, uses a proven network architecture that is pre-set using an extensive collection of labelled image data, and reduces user involvement into network design issues. Successful application of this technique suggests that the features generated by those large image collections have a good generalization capability and can be matched to arbitrary datasets assuming a fine-tuning step. Supervised training and transfer learning (with or without fine tuning) for selected CNN architectures are described in detail in Nogueira et al. (2017) and tested for three well-known scene classification datasets, and comparisons of different possibilities for feature extraction is presented. They suggested to use transfer learning and fine-tuning instead of fully supervised training because the pre-trained networks start from a better initialization state in the search space. A significant limitation though is that pre-trained networks are not currently applicable in multispectral/hyperspectral classification tasks because existing pre-trained networks come from computer vision - trained on ImageNet - using RGB images. However, as studied in Huang et al. (2018), we could mix a big pre-trained network fed by the RGB portion of spectrum with a smaller deep network capable of mining the entire spectrum and obtain good results. Such a combination can also be run on limited number of input samples as the large network is pretrained. For the other options of semisupervised and unsupervised learning we could see limited improvement but there were not enough samples for conclusive results.

Looking at Figure 3-3 and the SAE learning methods, fine-tuning of unsupervised methods tends to offer some gains over enhanced SVM, while there was no gain without fine-tuning. An explanation could be that unsupervised learning receives its strength from using much more data (labeled or unlabeled), so the features may be more representative of the data. However, matching them to classes requires an extra step of supervised learning. Therefore, unsupervised learning alone is comparable to enhanced SVM, and fine-tuning further improves results. Semi-supervised learning was used in two cases with better results, but its application detail is case-dependent. There were different methods and also underlying assumptions about actual class distribution for doing semi-supervised learning (for example see Zhu and Goldberg, 2009, and Camps-Valls et al. 2014). Each method and assumption were embedding a specific additional regularization term for unlabeled data in the optimization cost function but there was no standardized way of doing that. This lack of standardization might be a cause for its limited use.

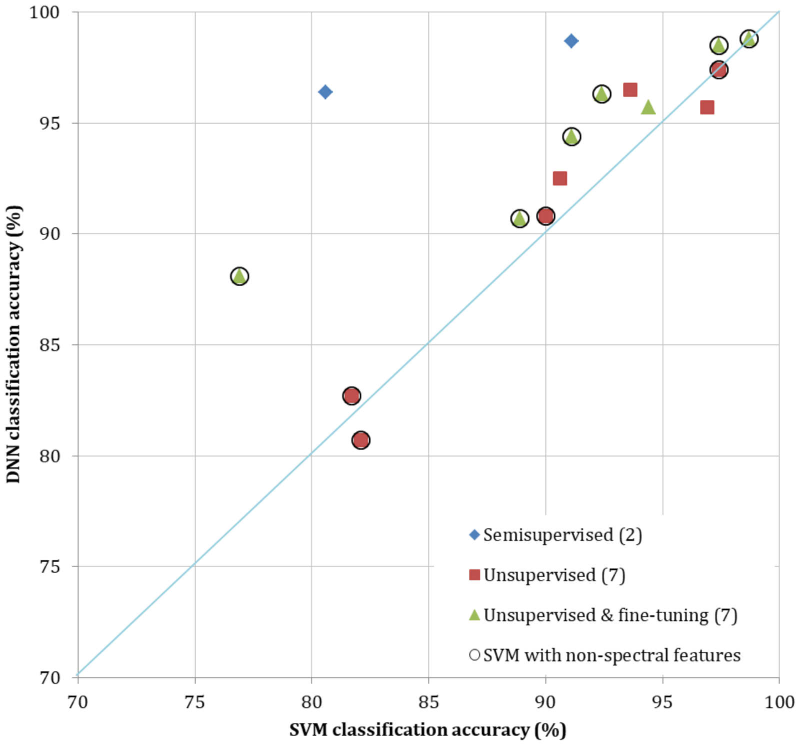


Figure 3-3: Comparative performance distribution across learning methods for SAE

Distribution across network complexity. To examine this, we discretized the number of parameters to six bins from less than 10K (class A) to greater than 100M (class F); the result is shown in Figure 3-4. Extremely low end (class A) cases were rare and do not seem to offer considerable improvements. Class B had the highest frequency (23 cases), then class C, D, E, and F with 16, 11, 10, and 9 cases correspondingly. It could be seen that class B, which is a still relatively small network, was mostly present in the upper right part of the graph, where the performance of spectral SVM is already high. These cases were those mostly associated with supervised learning method mentioned before. But larger networks (especially classes D and F) showed considerable improvements over enhanced SVM. Based on this, we may advise to use larger networks (with fine-tuning) as mentioned before. However, this graph also demonstrates that all network complexity classes have the potential to achieve accuracy of 95% or more, which might be sufficient for many cases, especially considering other data limitations (e.g. registration errors). Note that class F cases were all ImageNet pre-trained networks, which were the largest networks in our study cases.

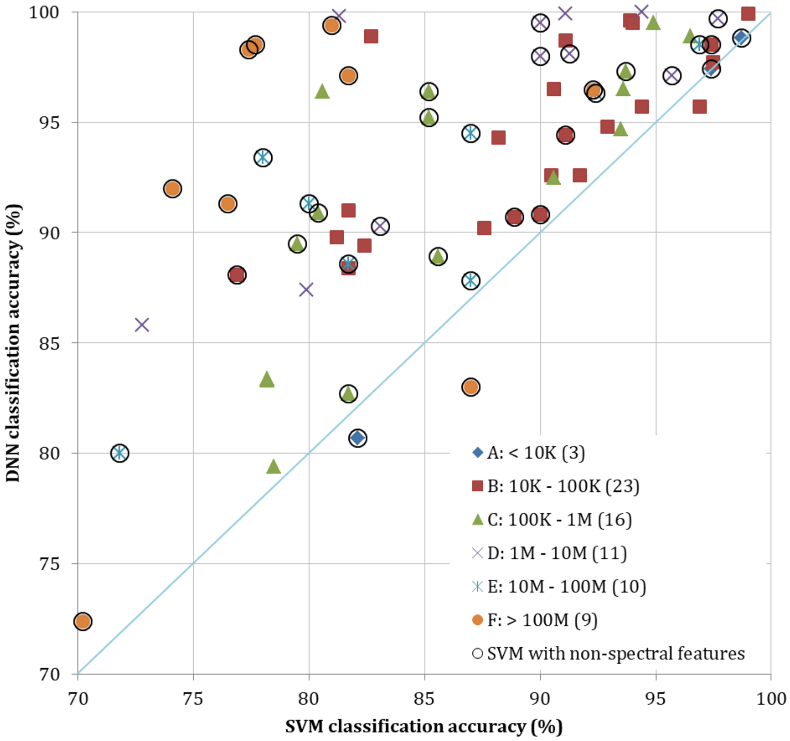


Figure 3-4: Comparative performance distribution across network complexity

Distribution across spatial resolution. The corresponding graph is shown in Figure 3-5. It was difficult to discern a specific pattern with respect to the spatial resolution, therefore no conclusive remarks could be made.

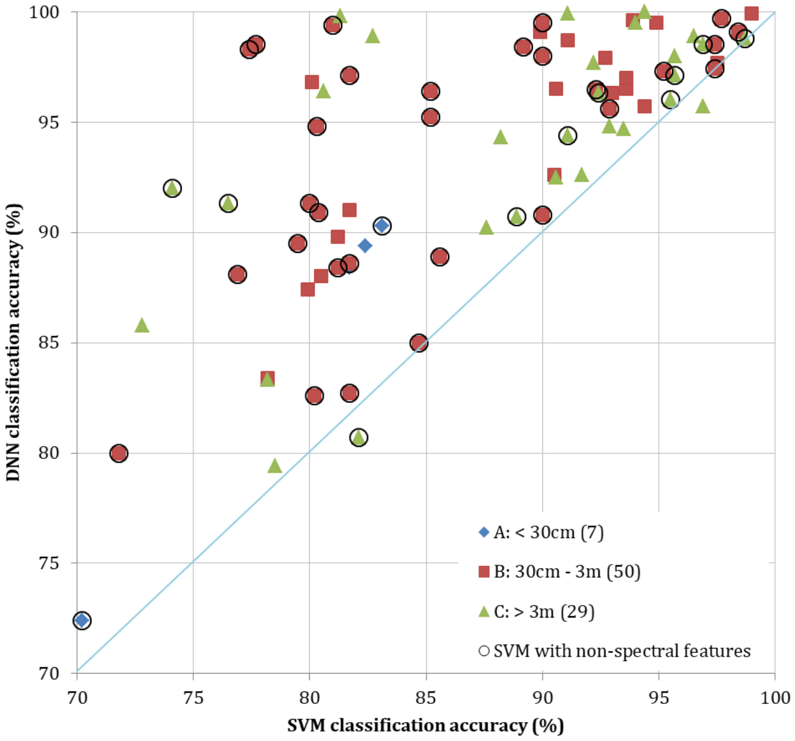


Figure 3-5: Comparative performance distribution across spatial resolution

Distribution across input data dimensionality. Figure 3-6 organizes the results in three general categories, mostly separating RGB (group A) and hyperspectral images (group C), with group B being cases employing additional multispectral components such as NI and/or auxiliary data such as DSM/LiDAR.

Although it seems that multispectral group (B) generally achieves a bit less improvement compared to other two groups, there was no strong evidence and supporting theory for that.

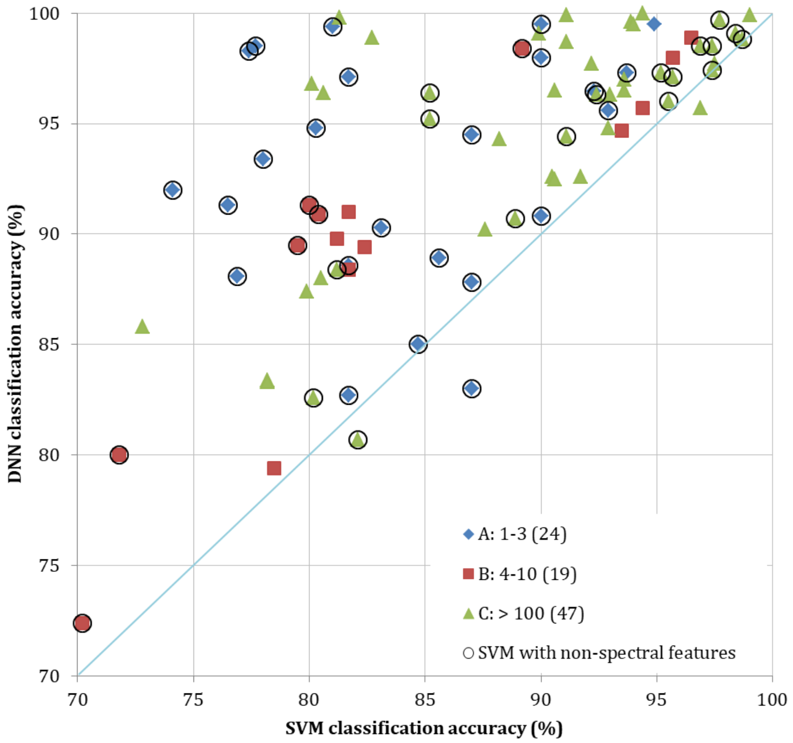


Figure 3-6: Comparative performance distribution across input data dimensionality

Distribution across training size/proportion. In the examined manuscripts the sampling was either a single pixel (for pixel classification or image segmentation applications) or an image patch (for scene classification or target detection applications). Labeled data size was mostly in the order of a few thousands, with some cases with considerably more labelled data. Sampling was done within the labelled dataset, with the proportion varying substantially in different implementations from as low as 0.1% to as high as 90%. We considered two different ways of training data size affecting network simulations. The first issue is the training data size, which should be considered in accordance with the network size and number of parameters. A large network with few training data may experience overfitting and lack of generalization, while a small network may not be powerful enough to model a complex set of training data. The other issue was the training data proportion, which imposes the same underfitting/overfitting scenario.

We compared the absolute number of training data units (pixels or scenes) to the number of network parameters in our database and found that in almost 90% of cases we have less data units than networks parameters to be tuned. The overfitting control mechanisms such as regularization were mostly included in the network design and it alleviate overfitting problem, but there was still a substantial difference between the remote sensing and computer vision fields, as we have very large reference datasets in the latter. Looking at Table 3-3, the only case with millions of samples in remote sensing were ISPRS datasets, but the winners were all CNNs and there was no comparison reported with SVMs (competition is just between different CNN architectures) so we couldn't include them in our SVM-based charts.

Two figures were produced in order to examine how DNN gains were influenced by training data absolute size and relative proportion with respect to the testing data. Figure 3-7 shows the comparative performance categorized in training proportions from A (less than 20%) to E (greater than 80%), and Figure 3-8 groups cases by absolute training dataset size from A (below 1000) to E (over one hundred thousand). The observed variability in the graphs and the lack of a consistent pattern suggest that high training size or proportion are not a general requirement for deep learning algorithms because there were various cases of high (> 95%) overall accuracy from very low to very high sampling ratio or size. A closer examination took place to further investigate training size and proportion with respect to network and learning method type. In cases of DBN, the training proportion was always high (> 50%) but there was no explanation or justification for it in the reviewed articles. In general, the CNN methods with supervised learning have been used in a wide range of training proportions, while CNNs with transfer learning with fine-tuning were run with higher training proportion. This may be attributed to overfitting concerns in transfer learning cases, as the base network is usually large with millions of parameters. Therefore, it is an open question how the transfer learning works in remote sensing cases where low training ratios are predominant.

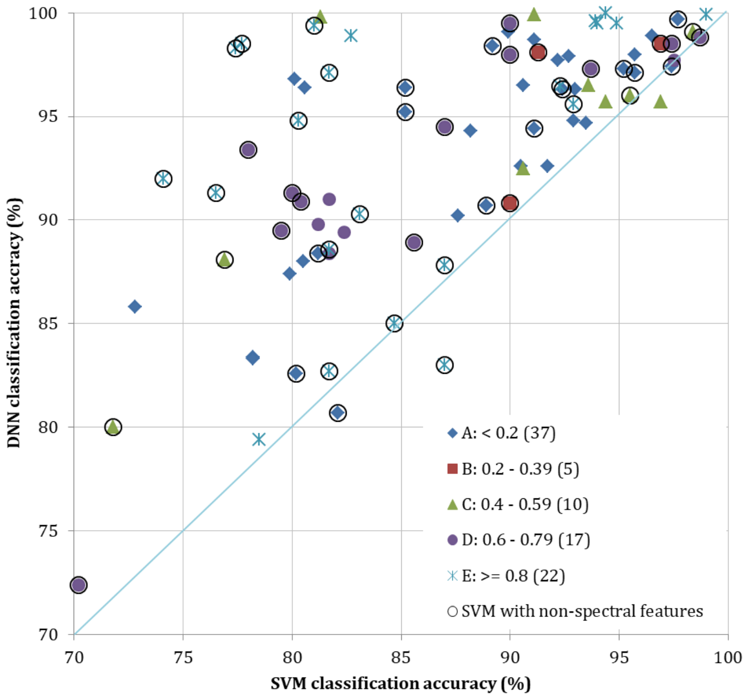


Figure 3-7: Comparative performance distribution across training proportion

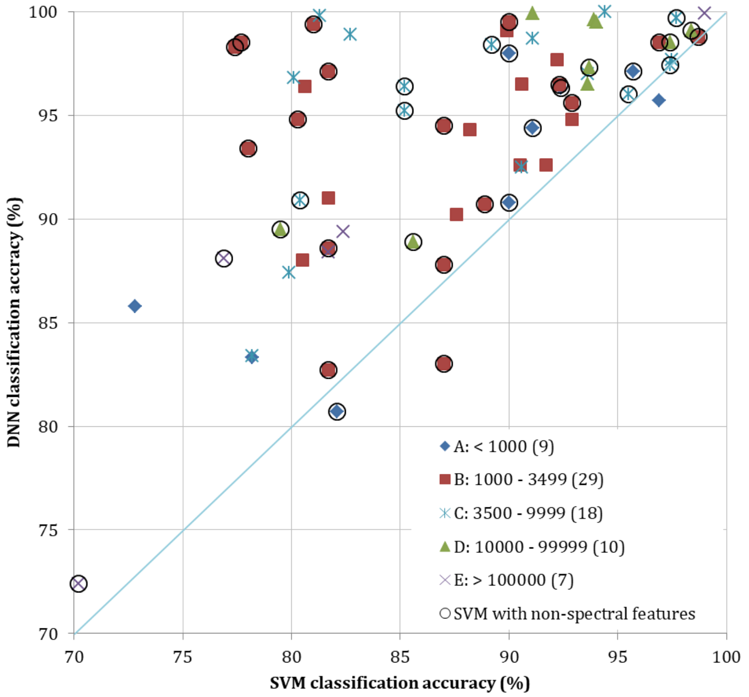


Figure 3-8: Comparative performance distribution across training data size

There is also another concern that has not been discussed in the reviewed papers. About 64% of the cases in our entire database (and about 73% of cases used in the figures) belong to pixel classification category with the rest focusing on scene classification. In scene classification cases we had completely separated train and test scenes, therefore adding spatial data in training phase (which naturally happens in any CNN network) would not affect the testing performance. In pixel classification application the train and test pixels are chosen independently, but if the spatial processing is part of algorithm (that is typical), the training and testing pixels' neighborhoods may overlap and this may violate the basic assumption of independent training/testing samples. The real impact of this issue was not discussed in any of reviewed literature and it seems that the authors didn't consider it critical. It can be also argued that with multiple pooling layers in a CNN network and enlarging scale of pixel influence, there is always some trace of even far pixels on training phase. Therefore, strictly enforcing independency rule to the neighboring pixels may invalidate all of the CNN networks, which is no desire for anybody.

Review of widely-used data sets. In previous sections the objective was to reveal patterns (or lack of any pattern) in different networks comparative performance along important parameters. The main limitation of this analysis is that the comparisons could not be done by varying just one parameter and fixing the others as we could not have such a control in our data collection (hence we used the term 'distribution' instead of 'effect' in our section titles). In this section we went one step further and looked at different cases as applied on the same dataset to extract more information on the competency of different network types. There were some datasets that have been heavily used in various papers and therefore can serve as a benchmark for algorithmic comparison. Except for ISPRS Potsdam and ISPRS Vaihingen (where comparison to SVM was not available), Figure 3-9 shows the result graphically.

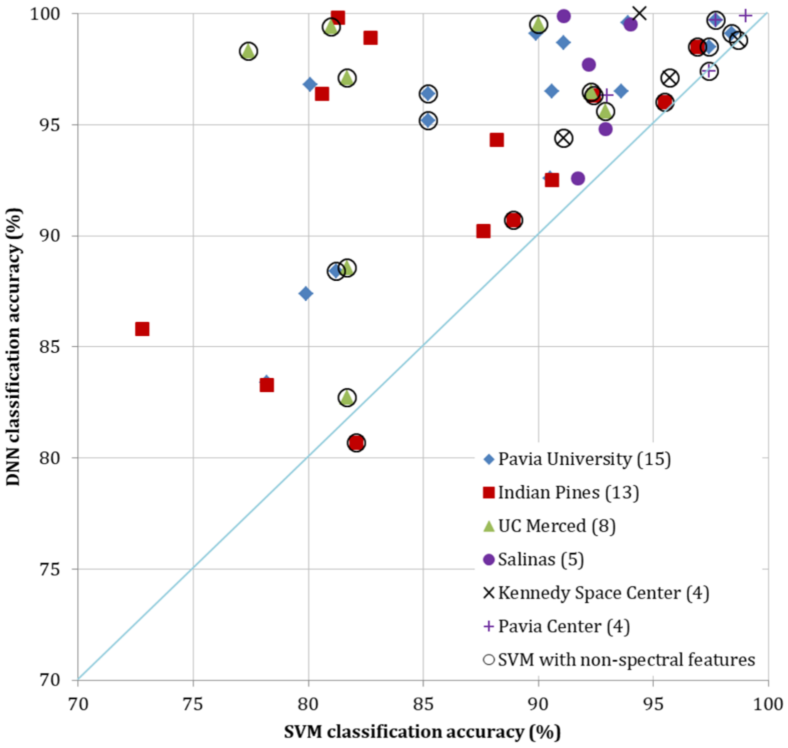


Figure 3-9: Comparative performance distribution across widely used datasets

Here are some observations:

- Indian Pines (a hyperspectral dataset): The highest accuracy here was obtained by CNN at 99.8% overall accuracy, but SAE and CNN have generally the same level of performance. Their improvement over spectral SVM can be as large as 16% for both network types. However, this high gain is reduced to just about 2-4% in comparison to enhanced SVM cases.
- Kennedy Space Center (a hyperspectral dataset): Recently CNN achieved an accuracy of 100% on this dataset (Haut et al. 2018) but with a high training proportion of 85% and 5.6% gain improvement over spectral SVM. Other comparisons were made with CNN and SAE but with

enhanced SVM. The best accuracy of SAE was 98.8%, which was almost the same as a very sophisticated SVM implementation.

- Pavia Center (a hyperspectral dataset): CNN implementations showed a little improvement up to 3.3% with training proportion of10%. The peak achieved accuracy was 99.95% but with a training proportion of 80% with very minor gains over SVM, and in all cases it was compared to the spectral SVM. We have only one SAE implementation for this dataset in our list, which does not improve over the enhanced SVM.
- Pavia University (a hyperspectral dataset): Here CNN here worked better than SAE with a maximum overall accuracy of 99.7% and improvements up to 16.7% over spectral SVM, while for SAE it is at most 7.6% (both with training proportion of 10%). We have about 2% improvement over a very sophisticated SVM implementation for this dataset, but for SAE the gain over enhanced SVM is minor.
- Salinas (a hyperspectral dataset): This dataset was only applied to CNN and the best achieved accuracy was 99.9% with a training proportion of 50%. This was 8% gain in accuracy compared to spectral SVM, and other results showed some other gains. But there was no case of comparison with enhanced SVM.
- UC Merced (an RGB dataset used for scene classification): Here CNN works well with maximum overall accuracy of 99.5% and improvement up to 21% over SVM, while SAE was tested once with improvement of just 1%. In all cases, training proportion was high (60%-80%) and it was compared to enhanced SVM, but SVM was fed with very different features in different cases.

For the ISPRS Potsdam and Vaihingen datasets, the CNNs has been the winner over all of the recent contests, so the race was only between them and there was no research that compare them to a SVM based classification. Therefore, we could not include them in Figure 3-9. In both ISPRS datasets the best results are achieved by transfer learning & fine-tuning in recent years. The best case was based on ResNet-101 with overall accuracy of 91.1% for both cases, followed by VGG-16 and SegNet-based cases with overall accuracy of 90.3%. Training proportion is standardized at 30% for Vaihingen and 45% for Potsdam (except ResNet-101 case, where the training proportion was 47% and 63%, respectively). These implementations are large networks, but a recent paper (C. Zhang et al. 2018) has also achieved accuracies of 89.4% for Potsdam and 88.4% for Vaihingen with a small supervised network with number of parameters much less than above transferred networks (but with increasing training proportion to 70-75%). In almost all cases additional enhancement techniques such as joint segmentation, CRF processing or multiscale blocks has been implemented to boost the performance a bit higher.

The above datasets, while used extensively for classification assessment, should be avoided in the future. They are relatively small to match the generalization capabilities of deep networks and in most cases there are already algorithms that reach 100% accuracy, therefore offering limited opportunities for improvement. It is necessary to develop new, large and multi-sensory datasets for remote sensing image classification, especially for hyperspectral data, to help better investigate the potential of deep networks.

3.4. Concluding remarks

While the number of case studies precluded detailed statistical analysis on the effect of each contributing factors generally we can see that:

- Deep networks have generally better performance than spectral SVM implementations, with CNNs performing better than other deep learners. This advantage, however, diminishes when using SVM over more rich features extracted from data.

- Transfer learning and fine-tuning on pre-trained CNNs offer promising results even when compared to enhanced SVM implementations, and they provide for flexibility and scalability because there is no need to manually engineer the features or use a very large training dataset. However, these pre-trained networks are currently limited to RGB input data, therefore currently lack applicability in multi/hyperspectral data. They have also not been tested in low training proportion scenarios.
- There is no strong relationship between network complexity and accuracy gains over SVM; small to medium networks perform similarly to more complex networks.
- Contrary to popular belief, there are numerous cases of good deep network performance with training proportions of 10% or lower.

As previously noted, deep networks are important due to their ability to extract useful rich features automatically from large data sets without the need for manual feature extraction. For example, automatic feature extraction has been used in Rußwurm and Körner (2018) to automatically detect cloud occlusion in temporal remote sensing data. This automation of feature extraction also has limitations, most notably the difficulty to extract and evaluate these features. The visualizations in deep networks rarely go further than the first two layers, which focus on very basic features like edges and gradients. There have been limited trials to describe and visualize the extracted features and even developing methods for it (for example see Zeiler and Fergus, 2013, or Yosinski et al. 2015), but currently research is lacking in remote sensing tasks.

We compare different studies and reflect on their findings in a collective manner. The possible reasons for deep network strengths in each individual aspect (network type, learning strategy, sampling proportion, etc.) was discussed in previous sections without going into mathematical formulas, due to the nature of meta-analysis. The majority of manuscripts reported that the SVM (or other rivals) parameters have been tuned and optimized for best performance, but there is a lack of consistency in reporting and protocol (e.g., grid search density). Establishing best optimization practices would benefit our community by limiting inconsistencies that could lead to result bias.

Another important conclusion is that algorithms are now outpacing benchmark datasets. We already see accuracy estimations exceeding 99% for some well-known datasets such as Indian Pine, Pavia Center and University, Salinas, and UC Merced. To allow deep learners to reach their full potential, it is paramount that more elaborate benchmark datasets should become available with diverse spectral/spatial/temporal resolution and geographic coverage.

We could not analyze further the processing time because either it was not available in many cases, or it was not specified if it contains the entire time for optimizing meta-parameters or not. It is generally true that deep networks need considerably more processing time for training (though the testing/simulation process is generally quick) but with continuous increases in processing power, deep networks are readily usable particularly by incorporating both CPUs and GPUs together. It would be interesting to evaluate the time saved by using pre-trained networks and just fine-tuning them, but currently there were no statistics reported to extract conclusive guidance.

There are numerous design options currently offered (see Table 3-4). Multiscale input is particularly useful to capture geographic relationships in earth observations. Furthermore, fully convolutional networks are promising for dense semantic labeling (classification of all image pixels at once and producing the same output dense map as the input image size). Other research have added various segmentation techniques, boundary detection and correction methods and CRF/MRF post-processing and showed their benefit to enhance classification of edge pixels.

While existing comparisons suggest the potential of CNN, they do not concretely identify a winning design among different options. For example, at the ISPRS Vaihingen image segmentation contest three CNN methods were within 1.2% of overall accuracy (Sherrah, 2016; Audebert et al. 2016; and Marmanis et al. 2016b). Looking into the future, remote sensing experts will favor 3-D CNN structures from pre-processing, dimensionality reduction methods like PCA or shallow 1-D and 2-D networks. The current state of the art 3-D CNN structures has already offered significant improvements and the training process is becoming easier (see Chen et al. 2016, and Y. Li et al. 2017). Furthermore, our community would significantly benefit from a coordinated investment from large funding institutions to create a pre-trained DNN for remote sensing data (similar to the ImageNet for RBG images). This pre-trained network would harness the power of large data volumes while allowing fine-tuning to specific applications.

Acknowledgements

This work was supported by the USDA McIntire Stennis program, a SUNY ESF Graduate Assistantship and NASA's Land Cover Land Use Change Program (grant # NNX15AD42G).

References

Aptoula, Erchan, Murat Can Ozdemir, and Berrin Yanikoglu. 2016. "Deep Learning With Attribute Profiles for Hyperspectral Image Classification." *IEEE Geoscience and Remote Sensing Letters* 13 (12): 1970–74. https://doi.org/10.1109/LGRS.2016.2619354.

Audebert, Nicolas, Bertrand Le Saux, and Sébastien Lefèvre. 2016. "Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-Scale Deep Networks." In *Asian Conference on Computer Vision*, 180–96. Springer, Cham. http://link.springer.com/chapter/10.1007/978-3-319-54181-5_12.

Basaeed, Essa, Harish Bhaskar, and Mohammed Al-Mualla. 2016. "Supervised Remote Sensing Image Segmentation Using Boosted Convolutional Neural Networks." *Knowledge-Based Systems* 99 (May): 19–27. https://doi.org/10.1016/j.knosys.2016.01.028.

Basu, Saikat, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna Nemani. 2015. "Deepsat: A Learning Framework for Satellite Imagery." In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 37. ACM. http://dl.acm.org/citation.cfm?id=2820816.

Ben Hamida, Amina, Alexandre Benoit, Patrick Lambert, and Chokri Ben Amar. 2018. “3-D Deep Learning Approach for Remote Sensing Image Classification.” *IEEE Transactions on Geoscience and Remote Sensing* 56 (8): 4420–34. https://doi.org/10.1109/TGRS.2018.2818945.

Bengio, Y. 2009. “Learning Deep Architectures for AI.” *Foundations and Trends in Machine Learning* 2 (1): 1–127. https://doi.org/10.1561/2200000006.

Bittner, K., S. Cui, and P. Reinartz. 2017. “Building Extraction from Remote Sensing Data Using Fully Convolutional Networks.” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-1/W1 (May): 481–86. https://doi.org/10.5194/isprs-archives-XLII-1-W1-481-2017.

Camps-Valls, Gustavo, Devis Tuia, Lorenzo Bruzzone, and Jon Atli Benediktsson. 2014. “Advances in Hyperspectral Image Classification: Earth Monitoring with Statistical Learning Methods.” *IEEE Signal Processing Magazine* 31 (1): 45–54. https://doi.org/10.1109/MSP.2013.2279179.

Cao, YuShe, Xin Niu, and Yong Dou. 2016. “Region-Based Convolutional Neural Networks for Object Detection in Very High Resolution Remote Sensing Images.” In , 548–54. IEEE. https://doi.org/10.1109/FSKD.2016.7603232.

Castelluccio, Marco, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. 2015. “Land Use Classification in Remote Sensing Images by Convolutional Neural Networks.” *ArXiv:1508.00092 [Cs]*, August. http://arxiv.org/abs/1508.00092.

Chen, Fen, Ruilong Ren, Tim Van de Voorde, Wenbo Xu, Guiyun Zhou, and Yan Zhou. 2018. “Fast Automatic Airport Detection in Remote Sensing Images Using Convolutional Neural Networks.” *Remote Sensing* 10 (3): 443. https://doi.org/10.3390/rs10030443.

Chen, Xueyun, Shiming Xiang, Cheng-Lin Liu, and Chun-Hong Pan. 2013. “Aircraft Detection by Deep Belief Nets.” In , 54–58. IEEE. https://doi.org/10.1109/ACPR.2013.5.

———. 2014. “Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks.” *IEEE Geoscience and Remote Sensing Letters* 11 (10): 1797–1801. https://doi.org/10.1109/LGRS.2014.2309695.

Chen, Y., Z. Lin, X. Zhao, G. Wang, and Y. Gu. 2014. “Deep Learning-Based Classification of Hyperspectral Data.” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7 (6): 2094–2107. https://doi.org/10.1109/JSTARS.2014.2329330.

Chen, Yushi, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram Ghamisi. 2016. “Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks.” *IEEE Transactions on Geoscience and Remote Sensing* 54 (10): 6232–51. https://doi.org/10.1109/TGRS.2016.2584107.

Chen, Yushi, Xing Zhao, and Xiuping Jia. 2015. “Spectral-Spatial Classification of Hyperspectral Data Based on Deep Belief Network.” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8 (6): 2381–92. https://doi.org/10.1109/JSTARS.2015.2388577.

Cheng, Gong, Junwei Han, and Xiaoqiang Lu. 2017. “Remote Sensing Image Scene Classification: Benchmark and State of the Art.” *Proceedings of the IEEE* 105 (10): 1865–83. https://doi.org/10.1109/JPROC.2017.2675998.

Cheng, Gong, Zhenpeng Li, Xiwen Yao, Lei Guo, and Zhongliang Wei. 2017. “Remote Sensing Image Scene Classification Using Bag of Convolutional Features.” *IEEE Geoscience and Remote Sensing Letters* 14 (10): 1735–39. https://doi.org/10.1109/LGRS.2017.2731997.

Cui, Wei, Zhendong Zheng, Qi Zhou, Jiejun Huang, and Yanbin Yuan. 2018. “Application of a Parallel Spectral–Spatial Convolution Neural Network in Object-Oriented Remote Sensing Land Use Classification.” *Remote Sensing Letters* 9 (4): 334–42. https://doi.org/10.1080/2150704X.2017.1420265.

Deng, Li. 2014. “A Tutorial Survey of Architectures, Algorithms, and Applications for Deep Learning.” *APSIPA Transactions on Signal and Information Processing* 3. https://doi.org/10.1017/atsip.2013.9.

Ding, Chen, Ying Li, Yong Xia, Wei Wei, Lei Zhang, and Yanning Zhang. 2017. “Convolutional Neural Networks Based Hyperspectral Image Classification Method with Adaptive Kernels.” *Remote Sensing* 9 (6): 618. https://doi.org/10.3390/rs9060618.

Fu, Gang, Changjun Liu, Rong Zhou, Tao Sun, and Qijian Zhang. 2017. “Classification for High Resolution Remote Sensing Imagery Using a Fully Convolutional Network.” *Remote Sensing* 9 (6): 498. https://doi.org/10.3390/rs9050498.

Geng, Jie, Jianchao Fan, Hongyu Wang, Xiaorui Ma, Baoming Li, and Fuliang Chen. 2015. “High-Resolution SAR Image Classification via Deep Convolutional Autoencoders.” *IEEE Geoscience and Remote Sensing Letters* 12 (11): 2351–55. https://doi.org/10.1109/LGRS.2015.2478256.

Ghamisi, Pedram, Yushi Chen, and Xiao Xiang Zhu. 2016. “A Self-Improving Convolution Neural Network for the Classification of Hyperspectral Data.” *IEEE Geoscience and Remote Sensing Letters* 13 (10): 1537–41. https://doi.org/10.1109/LGRS.2016.2595108.

Ghamisi, Pedram, Javier Plaza, Yushi Chen, Jun Li, and Antonio J Plaza. 2017. “Advanced Spectral Classifiers for Hyperspectral Images: A Review.” *IEEE Geoscience and Remote Sensing Magazine* 5 (1): 8–32. https://doi.org/10.1109/MGRS.2016.2616418.

Gong, Maoguo, Tao Zhan, Puzhao Zhang, and Qiguang Miao. 2017. “Superpixel-Based Difference Representation Learning for Change Detection in Multispectral Remote Sensing Images.” *IEEE Transactions on Geoscience and Remote Sensing* 55 (5): 2658–73. https://doi.org/10.1109/TGRS.2017.2650198.

Gong, Xi, Zhong Xie, Yuanyuan Liu, Xuguo Shi, and Zhuo Zheng. 2018. “Deep Salient Feature Based Anti-Noise Transfer Network for Scene Classification of Remote Sensing Imagery.” *Remote Sensing* 10 (3): 410. https://doi.org/10.3390/rs10030410.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge, Massachusetts: The MIT Press.

Gu, Xiaowei, Plamen P. Angelov, Ce Zhang, and Peter M. Atkinson. 2018. “A Massively Parallel Deep Rule-Based Ensemble Classifier for Remote Sensing Scenes.” *IEEE Geoscience and Remote Sensing Letters* 15 (3): 345–49. https://doi.org/10.1109/LGRS.2017.2787421.

Han, Wei, Ruyi Feng, Lizhe Wang, and Yafan Cheng. 2018. “A Semi-Supervised Generative Framework with Deep Learning Features for High-Resolution Remote Sensing Image Scene Classification.” *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (November): 23–43. https://doi.org/10.1016/j.isprsjprs.2017.11.004.

Haut, Juan Mario, Mercedes E. Paoletti, Javier Plaza, Jun Li, and Antonio Plaza. 2018. “Active Learning With Convolutional Neural Networks for Hyperspectral Image Classification Using a New Bayesian Approach.” *IEEE Transactions on Geoscience and Remote Sensing* 56 (11): 6440–61. https://doi.org/10.1109/TGRS.2018.2838665.

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. 2006. “A Fast Learning Algorithm for Deep Belief Nets.” *Neural Computation* 18 (7): 1527–54. https://doi.org/10.1162/neco.2006.18.7.1527.

Hu, Fan, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang. 2015. “Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery.” *Remote Sensing* 7 (11): 14680–707. https://doi.org/10.3390/rs71114680.

Hu, Jingliang, Lichao Mou, Andreas Schmitt, and Xiao Xiang Zhu. 2017. “FusioNet: A Two-Stream Convolutional Neural Network for Urban Scene Classification Using PolSAR and Hyperspectral Data.” In , 1–4. IEEE. https://doi.org/10.1109/JURSE.2017.7924565.

Hu, Wei, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li. 2015. “Deep Convolutional Neural Networks for Hyperspectral Image Classification.” *Journal of Sensors* 2015: 1–12. https://doi.org/10.1155/2015/258619.

Huang, Bo, Bei Zhao, and Yimeng Song. 2018. “Urban Land-Use Mapping Using a Deep Convolutional Neural Network with High Spatial Resolution Multispectral Remote Sensing Imagery.” *Remote Sensing of Environment* 214 (September): 73–86. https://doi.org/10.1016/j.rse.2018.04.050.

Ishii, Tomohiro, Ryosuke Nakamura, Hidemoto Nakada, Yoshihiko Mochizuki, and Hiroshi Ishikawa. 2015. “Surface Object Recognition with CNN and SVM in Landsat 8 Images.” In , 341–44. IEEE. https://doi.org/10.1109/MVA.2015.7153200.

Ji, Shunping, Chi Zhang, Anjian Xu, Yun Shi, and Yulin Duan. 2018. “3D Convolutional Neural Networks for Crop Classification with Multi-Temporal Remote Sensing Images.” *Remote Sensing* 10 (2): 75. https://doi.org/10.3390/rs10010075.

Karalas, Konstantinos, Grigorios Tsagkatakis, Michalis Zervakis, and Panagiotis Tsakalides. 2015. “Deep Learning for Multi-Label Land Cover Classification.” In , edited by Lorenzo Bruzzone, 96430Q. https://doi.org/10.1117/12.2195082.

Kemker, Ronald, Carl Salvaggio, and Christopher Kanan. 2018. “Algorithms for Semantic Segmentation of Multispectral Remote Sensing Imagery Using Deep Learning.” *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (November): 60–77. https://doi.org/10.1016/j.isprsjprs.2018.04.014.

Khan, Salman H., Xuming He, Fatih Porikli, and Mohammed Bennamoun. 2017. “Forest Change Detection in Incomplete Satellite Images With Deep Neural Networks.” *IEEE Transactions on Geoscience and Remote Sensing*, 1–17. https://doi.org/10.1109/TGRS.2017.2707528.

Khatami, Reza, Giorgos Mountrakis, and Stephen V. Stehman. 2016. “A Meta-Analysis of Remote Sensing Research on Supervised Pixel-Based Land-Cover Image Classification Processes: General Guidelines for Practitioners and Future Research.” *Remote Sensing of Environment* 177 (May): 89–100. https://doi.org/10.1016/j.rse.2016.02.028.

Lagrange, Adrien, Bertrand Le Saux, Anne Beaupere, Alexandre Boulch, Adrien Chan-Hon-Tong, Stéphane Herbin, Hicham Randrianarivo, and Marin Ferecatu. 2015. “Benchmarking Classification of Earth-Observation Data: From Learning Explicit Features to Convolutional Networks.” In *IGARSS 2015*. http://ieeexplore.ieee.org/abstract/document/7326745/.

Längkvist, Martin, Andrey Kiselev, Marjan Alirezaie, and Amy Loutfi. 2016. “Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks.” *Remote Sensing* 8 (4): 329. https://doi.org/10.3390/rs8040329.

Le, Quoc V. 2015. *A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks*.

Le Saux, Bertrand, Naoto Yokoya, Ronny Hansch, and Saurabh Prasad. 2018. “Advanced Multisource Optical Remote Sensing for Urban Land Use and Land Cover Classification [Technical Committees].” *IEEE Geoscience and Remote Sensing Magazine* 6 (4): 85–89. https://doi.org/10.1109/MGRS.2018.2874328.

Lguensat, Redouane, Miao Sun, Ronan Fablet, Evan Mason, Pierre Tandeo, and Ge Chen. 2017. “EddyNet: A Deep Neural Network For Pixel-Wise Classification of Oceanic Eddies.” *ArXiv:1711.03954 [Physics]*, November. http://arxiv.org/abs/1711.03954.

Li, Jiming, Lorenzo Bruzzone, and Sicong Liu. 2015. “Deep Feature Representation for Hyperspectral Image Classification.” In , 4951–54. IEEE. https://doi.org/10.1109/IGARSS.2015.7326943.

Li, T., J. Zhang, and Y. Zhang. 2014. “Classification of Hyperspectral Image Based on Deep Belief Networks.” In *2014 IEEE International Conference on Image Processing (ICIP),* 5132–36. https://doi.org/10.1109/ICIP.2014.7026039.

Li, Wei, Guodong Wu, Fan Zhang, and Qian Du. 2017. “Hyperspectral Image Classification Using Deep Pixel-Pair Features.” *IEEE Transactions on Geoscience and Remote Sensing* 55 (2): 844–53. https://doi.org/10.1109/TGRS.2016.2616355.

Li, Ying, Haokui Zhang, and Qiang Shen. 2017. “Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network.” *Remote Sensing* 9 (1): 67. https://doi.org/10.3390/rs9010067.

Liu, Peng, Kim-Kwang Raymond Choo, Lizhe Wang, and Fang Huang. 2017. “SVM or Deep Learning? A Comparative Study on Remote Sensing Image Classification.” *Soft Computing* 21 (23): 7053–65. https://doi.org/10.1007/s00500-016-2247-2.

Liu, Shuang, Mei Li, Zhong Zhang, Baihua Xiao, and Xiaozhong Cao. 2018. “Multimodal Ground-Based Cloud Classification Using Joint Fusion Convolutional Neural Network.” *Remote Sensing* 10 (6): 822. https://doi.org/10.3390/rs10060822.

Liu, Weibo, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. 2017. “A Survey of Deep Neural Network Architectures and Their Applications.” *Neurocomputing* 234 (April): 11–26. https://doi.org/10.1016/j.neucom.2016.12.038.

Liu, Yanfei, Yanfei Zhong, Feng Fei, Qiqi Zhu, and Qianqing Qin. 2018. “Scene Classification Based on a Deep Random-Scale Stretched Convolutional Neural Network.” *Remote Sensing* 10 (3): 444. https://doi.org/10.3390/rs10030444.

Liu, Yongcheng, Bin Fan, Lingfeng Wang, Jun Bai, Shiming Xiang, and Chunhong Pan. 2018. “Semantic Labeling in Very High Resolution Images via a Self-Cascaded Convolutional Neural Network.” *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (November): 78–95. https://doi.org/10.1016/j.isprsjprs.2017.12.007.

Luus, F. P. S., B. P. Salmon, F. van den Bergh, and B. T. J. Maharaj. 2015. “Multiview Deep Learning for Land-Use Classification.” *IEEE Geoscience and Remote Sensing Letters* 12 (12): 2448–52. https://doi.org/10.1109/LGRS.2015.2483680.

Lyu, Haobo, Hui Lu, and Lichao Mou. 2016. “Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection.” *Remote Sensing* 8 (6): 506. https://doi.org/10.3390/rs8060506.

Ma, Lei, Manchun Li, Xiaoxue Ma, Liang Cheng, Peijun Du, and Yongxue Liu. 2017. “A Review of Supervised Object-Based Land-Cover Image Classification.” *ISPRS Journal of Photogrammetry and Remote Sensing* 130 (August): 277–93. https://doi.org/10.1016/j.isprsjprs.2017.06.001.

Ma, Xiaorui, Jie Geng, and Hongyu Wang. 2015. “Hyperspectral Image Classification via Contextual Deep Learning.” *EURASIP Journal on Image and Video Processing* 2015 (1). https://doi.org/10.1186/s13640-015-0071-8.

Ma, Zhong, Zhuping Wang, Congxin Liu, and Xiangzeng Liu. 2016. “Satellite Imagery Classification Based on Deep Convolution Network.” *World Academy of Science, Engineering and Technology* 10. http://waset.org/abstracts/44963.

Maggiori, Emmanuel, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. 2016. “High-Resolution Semantic Labeling with Convolutional Neural Networks.” *ArXiv Preprint ArXiv:1611.01962*. https://arxiv.org/abs/1611.01962.

———. 2017. “Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification.” *IEEE Transactions on Geoscience and Remote Sensing* 55 (2): 645–57. https://doi.org/10.1109/TGRS.2016.2612821.

Makantasis, Konstantinos, Konstantinos Karantzalos, Anastasios Doulamis, and Nikolaos Doulamis. 2015. “Deep Supervised Learning for Hyperspectral Data Classification through Convolutional Neural Networks.” In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, 4959–62. IEEE. http://ieeexplore.ieee.org/abstract/document/7326945/.

Marcos, Diego, Michele Volpi, Benjamin Kellenberger, and Devis Tuia. 2018. “Land Cover Mapping at Very High Resolution with Rotation Equivariant CNNs: Towards Small yet Accurate Models.” *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (November): 96–107. https://doi.org/10.1016/j.isprsjprs.2018.01.021.

Marmanis, Dimitrios, Mihai Datcu, Thomas Esch, and Uwe Stilla. 2016. “Deep Learning Earth Observation Classification Using ImageNet Pretrained Networks.” *IEEE Geoscience and Remote Sensing Letters* 13 (1): 105–9. https://doi.org/10.1109/LGRS.2015.2499239.

Marmanis, Dimitrios, Konrad Schindler, Jan Dirk Wegner, Silvano Galliani, Mihai Datcu, and Uwe Stilla. 2016. “Classification with an Edge: Improving Semantic Image Segmentation with Boundary Detection.” *ArXiv Preprint ArXiv:1612.01337*. https://arxiv.org/abs/1612.01337.

Mou, Lichao, Lorenzo Bruzzone, and Xiao Xiang Zhu. 2018. “Learning Spectral-Spatial-Temporal Features via a Recurrent Convolutional Neural Network for Change Detection in Multispectral Imagery.” *ArXiv:1803.02642 [Cs]*, March. http://arxiv.org/abs/1803.02642.

Mou, Lichao, Pedram Ghamisi, and Xiao Xiang Zhu. 2017. “Deep Recurrent Neural Networks for Hyperspectral Image Classification.” *IEEE Transactions on Geoscience and Remote Sensing* 55 (7): 3639–55. https://doi.org/10.1109/TGRS.2016.2636241.

———. 2018. “Unsupervised Spectral–Spatial Feature Learning via Deep Residual Conv–Deconv Network for Hyperspectral Image Classification.” *IEEE Transactions on Geoscience and Remote Sensing* 56 (1): 391–406. https://doi.org/10.1109/TGRS.2017.2748160.

Mountrakis, Giorgos, Jungho Im, and Caesar Ogole. 2011. “Support Vector Machines in Remote Sensing: A Review.” *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (3): 247–59. https://doi.org/10.1016/j.isprsjprs.2010.11.001.

Ndikumana, Emile, Dinh Ho Tong Minh, Nicolas Baghdadi, Dominique Courault, and Laure Hossard. 2018. “Deep Recurrent Neural Network for Agricultural Classification Using Multitemporal SAR Sentinel-1 for Camargue, France.” *Remote Sensing* 10 (8): 1217. https://doi.org/10.3390/rs10081217.

Niculescu, S., D. Ienco, and J. Hanganu. 2018. “APPLICATION OF DEEP LEARNING OF MULTI-TEMPORAL SENTINEL-1 IMAGES FOR THE CLASSIFICATION OF

COASTAL VEGETATION ZONE OF THE DANUBE DELTA." *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII–3 (April): 1311–18. https://doi.org/10.5194/isprs-archives-XLII-3-1311-2018.

Nogueira, Keiller, Otávio A. B. Penatti, and Jefersson A. dos Santos. 2017. "Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification." *Pattern Recognition* 61 (January): 539–56. https://doi.org/10.1016/j.patcog.2016.07.001.

Paisitkriangkrai, S., J. Sherrah, P. Janney, and A. Van-Den Hengel. 2015. "Effective Semantic Pixel Labelling with Convolutional Networks and Conditional Random Fields." In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 36–43. https://doi.org/10.1109/CVPRW.2015.7301381.

Pan, Bin, Zhenwei Shi, and Xia Xu. 2018. "MugNet: Deep Learning for Hyperspectral Image Classification Using Limited Samples." *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (November): 108–19. https://doi.org/10.1016/j.isprsjprs.2017.11.003.

Paoletti, M.E., J.M. Haut, J. Plaza, and A. Plaza. 2018. "A New Deep Convolutional Neural Network for Fast Hyperspectral Image Classification." *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (November): 120–47. https://doi.org/10.1016/j.isprsjprs.2017.11.021.

Penatti, O. A. B., K. Nogueira, and J. A. dos Santos. 2015. "Do Deep Features Generalize from Everyday Objects to Remote Sensing and Aerial Scenes Domains?" In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 44–51. https://doi.org/10.1109/CVPRW.2015.7301382.

Qayyum, Abdul, Aamir Saeed Malik, Naufal M Saad, Mahboob Iqbal, Mohd Faris Abdullah, Waqas Rasheed, Tuan AB Rashid Abdullah, and Mohd Yaqoob Bin Jafaar. 2017. "Scene Classification for Aerial Images Based on CNN Using Sparse Coding Technique." *International Journal of Remote Sensing* 38 (8–10): 2662–85. https://doi.org/10.1080/01431161.2017.1296206.

Rezaee, Mohammad, Masoud Mahdianpari, Yun Zhang, and Bahram Salehi. 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* 11 (9): 3030–39. https://doi.org/10.1109/JSTARS.2018.2846178.

Romero, A., C. Gatta, and G. Camps-Valls. 2016. "Unsupervised Deep Feature Extraction for Remote Sensing Image Classification." *IEEE Transactions on Geoscience and Remote Sensing* 54 (3): 1349–62. https://doi.org/10.1109/TGRS.2015.2478379.

Rosenblatt, F. 1958. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychological Review* 65 (6): 386–408. https://doi.org/10.1037/h0042519.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. 1986. "Learning Representations by Back-Propagating Errors." *Nature* 323 (6088): 533–36. https://doi.org/10.1038/323533a0.

Rußwurm, M., and M. Körner. 2017. "Multi-Temporal Land Cover Classification With Long Short-Term Memory Neural Networks." *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-1/W1 (May): 551–58. https://doi.org/10.5194/isprs-archives-XLII-1-W1-551-2017.

Rußwurm, Marc, and Marco Körner. 2018. "Multi-Temporal Land Cover Classification with Sequential Recurrent Encoders." *ArXiv:1802.02080 [Cs]*, February. http://arxiv.org/abs/1802.02080.

Sharma, Atharva, Xiuwen Liu, and Xiaojun Yang. 2018. “Land Cover Classification from Multi-Temporal, Multi-Spectral Remotely Sensed Imagery Using Patch-Based Recurrent Neural Networks.” *Neural Networks* 105 (September): 346–55. https://doi.org/10.1016/j.neunet.2018.05.019.

Sherrah, Jamie. 2016. “Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery.” *ArXiv Preprint ArXiv:1606.02585*. https://arxiv.org/abs/1606.02585.

Shi, Cheng, and Chi-Man Pun. 2018. “Superpixel-Based 3D Deep Neural Networks for Hyperspectral Image Classification.” *Pattern Recognition* 74 (February): 600–616. https://doi.org/10.1016/j.patcog.2017.09.007.

Singhal, Vanika, Anupriya Gogna, and Angshul Majumdar. 2016. “Deep Dictionary Learning vs Deep Belief Network vs Stacked Autoencoder: An Empirical Analysis.” In *Neural Information Processing*, edited by Akira Hirose, Seiichi Ozawa, Kenji Doya, Kazushi Ikeda, Minho Lee, and Derong Liu, 9950:337–44. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-46681-1_41.

Sun, Xin, Fei Zhou, Junyu Dong, Feng Gao, Quanquan Mu, and Xinhua Wang. 2017. “Encoding Spectral and Spatial Context Information for Hyperspectral Image Classification.” *IEEE Geoscience and Remote Sensing Letters* 14 (12): 2250–54. https://doi.org/10.1109/LGRS.2017.2759168.

Tang, Tianyu, Shilin Zhou, Zhipeng Deng, Huanxin Zou, and Lin Lei. 2017. “Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining.” *Sensors* 17 (2): 336. https://doi.org/10.3390/s17020336.

Tao, Chao, Hongbo Pan, Yansheng Li, and Zhengrou Zou. 2015. “Unsupervised Spectral&#x2013;Spatial Feature Learning With Stacked Sparse Autoencoder for Hyperspectral Imagery Classification.” *IEEE Geoscience and Remote Sensing Letters* 12 (12): 2438–42. https://doi.org/10.1109/LGRS.2015.2482520.

Tao, Yiting, Miaozhong Xu, Zhongyuan Lu, and Yanfei Zhong. 2018. “DenseNet-Based Depth-Width Double Reinforced Deep Learning Neural Network for High-Resolution Remote Sensing Image Per-Pixel Classification.” *Remote Sensing* 10 (5): 779. https://doi.org/10.3390/rs10050779.

Tobler, W. R. 1970. “A Computer Movie Simulating Urban Growth in the Detroit Region.” *Economic Geography* 46 (June): 234. https://doi.org/10.2307/143141.

Tschannen, Michael, Lukas Cavigelli, Fabian Mentzer, Thomas Wiatowski, and Luca Benini. 2016. “Deep Structured Features for Semantic Segmentation.” *ArXiv:1609.07916 [Cs]*, September. http://arxiv.org/abs/1609.07916.

Vakalopoulou, M., K. Karantzalos, N. Komodakis, and N. Paragios. 2015. “Building Detection in Very High Resolution Multispectral Data with Deep Learning Features.” In , 1873–76. IEEE. https://doi.org/10.1109/IGARSS.2015.7326158.

Vincent, Pascal, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion.” *J. Mach. Learn. Res.* 11: 3371–3408.

Volpi, Michele, and Devis Tuia. 2017. “Dense Semantic Labeling of Subdecimeter Resolution Images with Convolutional Neural Networks.” *IEEE Transactions on Geoscience and Remote Sensing* 55 (2): 881–93.

Wang, Jun, Jingwei Song, Mingquan Chen, and Zhi Yang. 2015. “Road Network Extraction: A Neural-Dynamic Framework Based on Deep Learning and a Finite State Machine.”

International Journal of Remote Sensing 36 (12): 3144–69. https://doi.org/10.1080/01431161.2015.1054049.

Wang, Shui-Hua, Junding Sun, Preetha Phillips, Guihu Zhao, and Yu-Dong Zhang. 2018. “Polarimetric Synthetic Aperture Radar Image Segmentation by Convolutional Neural Network Using Graphical Processing Units.” *Journal of Real-Time Image Processing* 15 (3): 631–42. https://doi.org/10.1007/s11554-017-0717-0.

Weng, Qian, Zhengyuan Mao, Jiawen Lin, and Wenzhong Guo. 2017. “Land-Use Classification via Extreme Learning Classifier Based on Deep Convolutional Features.” *IEEE Geoscience and Remote Sensing Letters* 14 (5): 704–8. https://doi.org/10.1109/LGRS.2017.2672643.

Weng, Qian, Zhengyuan Mao, Jiawen Lin, and Xiangwen Liao. 2018. “Land-Use Scene Classification Based on a CNN Using a Constrained Extreme Learning Machine.” *International Journal of Remote Sensing* 39 (19): 6281–99. https://doi.org/10.1080/01431161.2018.1458346.

Wu, Hang, Baozhen Liu, Weihua Su, Wenchang Zhang, and Jinggong Sun. 2016. “Deep Filter Banks for Land-Use Scene Classification.” *IEEE Geoscience and Remote Sensing Letters* 13 (12): 1895–99. https://doi.org/10.1109/LGRS.2016.2616440.

Wu, Hao, and Saurabh Prasad. 2018. “Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification.” *IEEE Transactions on Image Processing* 27 (3): 1259–70. https://doi.org/10.1109/TIP.2017.2772836.

Xia, Gui-Song, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. 2017. “AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification.” *IEEE Transactions on Geoscience and Remote Sensing* 55 (7): 3965–81. https://doi.org/10.1109/TGRS.2017.2685945.

Xing, Chen, Li Ma, and Xiaoquan Yang. 2016. “Stacked Denoise Autoencoder Based Feature Extraction and Classification for Hyperspectral Images.” *Journal of Sensors* 2016: 1–10. https://doi.org/10.1155/2016/3632943.

Xu, Xiaodong, Wei Li, Qiong Ran, Qian Du, Lianru Gao, and Bing Zhang. 2018. “Multisource Remote Sensing Data Classification Based on Convolutional Neural Network.” *IEEE Transactions on Geoscience and Remote Sensing* 56 (2): 937–49. https://doi.org/10.1109/TGRS.2017.2756851.

Yosinski, Jason, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. “Understanding Neural Networks Through Deep Visualization.” *ArXiv:1506.06579 [Cs]*, June. http://arxiv.org/abs/1506.06579.

Yu, Yang, Zhiqiang Gong, Cheng Wang, and Ping Zhong. 2017. “An Unsupervised Convolutional Feature Fusion Network for Deep Representation of Remote Sensing Images.” *IEEE Geoscience and Remote Sensing Letters*, 1–5. https://doi.org/10.1109/LGRS.2017.2767626.

Yu, Yongtao, Haiyan Guan, Dawei Zai, and Zheng Ji. 2016. “Rotation-and-Scale-Invariant Airplane Detection in High-Resolution Satellite Images Based on Deep-Hough-Forests.” *ISPRS Journal of Photogrammetry and Remote Sensing* 112 (February): 50–64. https://doi.org/10.1016/j.isprsjprs.2015.04.014.

Yue, Jun, Wenzhi Zhao, Shanjun Mao, and Hui Liu. 2015. “Spectral–Spatial Classification of Hyperspectral Images Using Deep Convolutional Neural Networks.” *Remote Sensing Letters* 6 (6): 468–77. https://doi.org/10.1080/2150704X.2015.1047045.

Zabalza, Jaime, Jinchang Ren, Jiangbin Zheng, Huimin Zhao, Chunmei Qing, Zhijing Yang, Peijun Du, and Stephen Marshall. 2016. “Novel Segmented Stacked Autoencoder for

Effective Dimensionality Reduction and Feature Extraction in Hyperspectral Imaging." *Neurocomputing* 185 (April): 1–10. https://doi.org/10.1016/j.neucom.2015.11.044.

Zeiler, Matthew D., and Rob Fergus. 2013. "Visualizing and Understanding Convolutional Networks." *ArXiv:1311.2901 [Cs]*, November. http://arxiv.org/abs/1311.2901.

———. 2014. "Visualizing and Understanding Convolutional Networks." In *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, edited by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, 818–33. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-10590-1_53.

Zhang, Ce, Xin Pan, Huapeng Li, Andy Gardiner, Isabel Sargent, Jonathon Hare, and Peter M. Atkinson. 2018. "A Hybrid MLP-CNN Classifier for Very Fine Resolution Remotely Sensed Image Classification." *ISPRS Journal of Photogrammetry and Remote Sensing* 140 (June): 133–44. https://doi.org/10.1016/j.isprsjprs.2017.07.014.

Zhang, Ce, Isabel Sargent, Xin Pan, Andy Gardiner, Jonathon Hare, and Peter M. Atkinson. 2018. "VPRS-Based Regional Decision Fusion of CNN and MRF Classifications for Very Fine Resolution Remotely Sensed Images." *IEEE Transactions on Geoscience and Remote Sensing* 56 (8): 4507–21. https://doi.org/10.1109/TGRS.2018.2822783.

Zhang, Ce, Isabel Sargent, Xin Pan, Huapeng Li, Andy Gardiner, Jonathon Hare, and Peter M. Atkinson. 2018. "An Object-Based Convolutional Neural Network (OCNN) for Urban Land Use Classification." *Remote Sensing of Environment* 216 (October): 57–70. https://doi.org/10.1016/j.rse.2018.06.034.

Zhang, Fan, Bo Du, and Liangpei Zhang. 2015. "Saliency-Guided Unsupervised Feature Learning for Scene Classification." *IEEE Transactions on Geoscience and Remote Sensing* 53 (4): 2175–84. https://doi.org/10.1109/TGRS.2014.2357078.

———. 2016. "Scene Classification via a Gradient Boosting Random Convolutional Network Framework." *IEEE Transactions on Geoscience and Remote Sensing* 54 (3): 1793–1802. https://doi.org/10.1109/TGRS.2015.2488681.

———. 2017. "A Multi-Task Convolutional Neural Network for Mega-City Analysis Using Very High Resolution Satellite Imagery and Geospatial Data." *ArXiv:1702.07985 [Cs]*, February. http://arxiv.org/abs/1702.07985.

Zhang, Haokui, Ying Li, Yuzhu Zhang, and Qiang Shen. 2017. "Spectral-Spatial Classification of Hyperspectral Imagery Using a Dual-Channel Convolutional Neural Network." *Remote Sensing Letters* 8 (5): 438–47. https://doi.org/10.1080/2150704X.2017.1280200.

Zhang, L., Z. Shi, and J. Wu. 2015. "A Hierarchical Oil Tank Detector With Deep Surrounding Features for High-Resolution Optical Satellite Imagery." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8 (10): 4895–4909. https://doi.org/10.1109/JSTARS.2015.2467377.

Zhang, L., L. Zhang, and B. Du. 2016. "Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art." *IEEE Geoscience and Remote Sensing Magazine* 4 (2): 22–40. https://doi.org/10.1109/MGRS.2016.2540798.

Zhang, Liangpei, Lefei Zhang, and Bo Du. 2016. "Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art." *IEEE Geoscience and Remote Sensing Magazine* 4 (2): 22–40. https://doi.org/10.1109/MGRS.2016.2540798.

Zhao, Chunhui, Xiaoqing Wan, Genping Zhao, Bing Cui, Wu Liu, and Bin Qi. 2017. "Spectral-Spatial Classification of Hyperspectral Imagery Based on Stacked Sparse Autoencoder and

Random Forest." *European Journal of Remote Sensing* 50 (1): 47–63. https://doi.org/10.1080/22797254.2017.1274566.

Zhao, Wenzhi, and Shihong Du. 2016. "Learning Multiscale and Deep Representations for Classifying Remotely Sensed Imagery." *ISPRS Journal of Photogrammetry and Remote Sensing* 113 (March): 155–65. https://doi.org/10.1016/j.isprsjprs.2016.01.004.

Zhao, Wenzhi, Shihong Du, and William J. Emery. 2017. "Object-Based Convolutional Neural Network for High-Resolution Imagery Classification." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10 (7): 3386–96. https://doi.org/10.1109/JSTARS.2017.2680324.

Zhao, Wenzhi, Zhou Guo, Jun Yue, Xiuyuan Zhang, and Liqun Luo. 2015. "On Combining Multiscale Deep Learning Features for the Classification of Hyperspectral Remote Sensing Imagery." *International Journal of Remote Sensing* 36 (13): 3368–79. https://doi.org/10.1080/2150704X.2015.1062157.

Zhou, W., Z. Shao, and Q. Cheng. 2016. "Deep Feature Representations for High-Resolution Remote Sensing Scene Classification." In *2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA)*, 338–42. https://doi.org/10.1109/EORSA.2016.7552825.

Zhou, Weixun, Shawn Newsam, Congmin Li, and Zhenfeng Shao. 2017. "Learning Low Dimensional Convolutional Neural Networks for High-Resolution Remote Sensing Image Retrieval." *Remote Sensing* 9 (5): 489. https://doi.org/10.3390/rs9050489.

Zhou, Weixun, Zhenfeng Shao, Chunyuan Diao, and Qimin Cheng. 2015. "High-Resolution Remote-Sensing Imagery Retrieval Using Sparse Features by Auto-Encoder." *Remote Sensing Letters* 6 (10): 775–83. https://doi.org/10.1080/2150704X.2015.1074756.

Zhou, Yingbo, Devansh Arpit, Ifeoma Nwogu, and Venu Govindaraju. 2014. "Is Joint Training Better for Deep Auto-Encoders?" *ArXiv:1405.1380 [Cs, Stat]*, May. http://arxiv.org/abs/1405.1380.

Zhu, Xiao Xiang, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. 2017. "Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources." *IEEE Geoscience and Remote Sensing Magazine* 5 (4): 8–36. https://doi.org/10.1109/MGRS.2017.2762307.

Zhu, Xiaojin, and Andrew B. Goldberg. 2009. "Introduction to Semi-Supervised Learning." *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3 (1): 1–130. https://doi.org/10.2200/S00196ED1V01Y200906AIM006.

Zou, Qin, Lihao Ni, Tong Zhang, and Qian Wang. 2015. "Deep Learning Based Feature Selection for Remote Sensing Scene Classification." *IEEE Geoscience and Remote Sensing Letters* 12 (11): 2321–25. https://doi.org/10.1109/LGRS.2015.2475299.

Appendix

Table A-1: Database of collected deep network application in remote sensing

									Best Reported Performance	
Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
(Penatti, Nogueira, and Santos 2015)	CNN	289M	Transfer learning	Brazilian coffee		3	0.8	Average accuracy	83	87
(Yang Yu et al. 2017)	CNN	24.6M	Unsupervised	Brazilian coffee		3	0.8	Overall accuracy	87.8	87
(Castellucci o et al. 2015)	CNN	5M	Transfer learning	Brazilian coffee		3		Overall accuracy	91.8	
(Nogueira, Penatti, and Santos 2017)	CNN	60M	Transfer learning & fine-tuning	Brazilian coffee		3	0.6	Overall accuracy	94.5	87
(Hao Wu and Prasad 2018)	CNN+RNN		Semisupervised	Houston	2.5 m	144		Overall accuracy	82.6	80.2
(Xu et al. 2018)	CNN		Supervised+	Houston	2.5 m	144+1	0.19	Overall accuracy	88	80.5
(Pan, Shi, and Xu 2018)	CNN			Houston	2.5 m	144		Overall accuracy	90.8	
(T. Li, Zhang, and Zhang 2014)	DBN	14.7K	Unsupervised & fine-tuning	Houston	2.5 m	144		Overall accuracy	97.7	97.5
(Zabalza et al. 2016)	SAE	4.2K	Unsupervised	Indian Pines	20 m	200	0.05	Overall accuracy	80.7	82.1
(Ghamisi, Chen, and Zhu 2016)	CNN	188K	Supervised	Indian Pines	20 m	200	0.05	Overall accuracy	83.3	78.2
(Shi and Pun 2018)	CNN	2.5M	Supervised	Indian Pines	20 m	200	0.01	Overall accuracy	85.2	
(Mou, Ghamisi, and Zhu 2018)	CNN	1.44M	Unsupervised & fine-tuning	Indian Pines	20 m	200	0.05	Overall accuracy	85.8	72.8
(C. Zhao et al. 2017)	SAE	30.2K	Unsupervised & fine-tuning	Indian Pines	20 m	200	0.1	Overall accuracy	89.8	88.9
(W. Hu et al. 2015)	CNN	80.6K	Supervised	Indian Pines	20 m	220	0.2	Overall accuracy	90.2	87.6
(Pan, Shi, and Xu 2018)	CNN			Indian Pines	20 m	200		Overall accuracy	90.7	

Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
(Xing, Ma, and Yang 2016)	SAE	241K	Unsupervised	Indian Pines	20 m	200	0.5	Overall accuracy	92.1	90.6
(Wei Li et al. 2017)	CNN	57.9K	Supervised	Indian Pines	20 m	220	0.2	Overall accuracy	94.3	88.2
(Yushi Chen, Zhao, and Jia 2015)	DBN		Unsupervised & fine-tuning	Indian Pines	20 m	200	0.5	Overall accuracy	96	95.5
(J. Li, Bruzzone, and Liu 2015)	SAE	21.7M	Unsupervised & fine-tuning	Indian Pines	20 m	200	0.05	Overall accuracy	96.3	92.4
(X. Sun et al. 2017)	SAE	107K	Semisupervised	Indian Pines	20 m	200	0.1	Overall accuracy	96.4	80.6
(Ding et al. 2017)	CNN	380K	Unsupervised	Indian Pines	20 m	200	0.5	Overall accuracy	97.8	
(X. Ma, Geng, and Wang 2015)	SAE	14.2K	Unsupervised & fine-tuning	Indian Pines	20 m	200	0.1	Overall accuracy	98.2	
(Paoletti et al. 2018)	CNN	96M	Supervised	Indian Pines	20 m	200	0.24	Overall accuracy	98.4	
(Yushi Chen et al. 2016)	CNN	44.9M	Supervised	Indian Pines	20 m	200	0.2	Overall accuracy	98.5	96.9
(H. Zhang et al. 2017)	CNN		Supervised	Indian Pines	20 m	200	0.1	Overall accuracy	98.8	
(Makantasis et al. 2015)	CNN	97.6K	Supervised	Indian Pines	20 m	224	0.8	Overall accuracy	98.9	82.7
(Y. Li, Zhang, and Shen 2017)	CNN	197K	Supervised	Indian Pines	20 m	200	0.5	Overall accuracy	99.1	
(Haut et al. 2018)	CNN	8.9M	Supervised+	Indian Pines	20 m	200	0.5	Overall accuracy	99.8	81.3
(Sherrah 2016)	CNN	3.26M	Supervised	ISPRS Potsdam	5 cm	5	0.45	Overall accuracy	84.1	
(Volpi and Tuia 2017)	CNN	6.38M	Supervised	ISPRS Potsdam	5 cm	5	0.45	Overall accuracy	85.8	
(Maggiori et al. 2016)	CNN	530K	Supervised	ISPRS Potsdam	5 cm	4	0.45	Overall accuracy	87	
(C. Zhang, Pan, et al. 2018)	CNN	17K	Supervised	ISPRS Potsdam	5 cm	4	0.75	Overall accuracy	89.4	82.4
(Sherrah 2016)	CNN	22.7M	Transfer learning & fine-tuning	ISPRS Potsdam	5 cm	4	0.45	Overall accuracy	90.3	
(Yongcheng Liu et al. 2018)	CNN	481M	Transfer learning & fine-tuning	ISPRS Potsdam	5 cm	4 (DSMs not used)	0.63	Overall accuracy	91.1	
(Tschannen et al. 2016)	CNN	30K	Supervised	ISPRS Vaihingen	9 cm	5	0.3	Overall accuracy	85.5	

Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
(Paisitkriangkrai et al. 2015)	CNN		Supervised	ISPRS Vaihingen	9 cm	5	0.3	Overall accuracy	86.9	
(W. Zhao, Du, and Emery 2017)	CNN		Supervised	ISPRS Vaihingen	9 cm	4	0.1	Overall accuracy	87.1	66.6
(Volpi and Tuia 2017)	CNN	6.38M	Supervised	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	87.3	
(Marcos et al. 2018)	CNN	100K	Supervised	ISPRS Vaihingen	9 cm	4	0.45	Overall accuracy	87.6	
(C. Zhang, Sargent, Pan, Gardiner, et al. 2018)	CNN	17K	Supervised	ISPRS Vaihingen	9 cm	4	0.7	Overall accuracy	88.4	81.7
(Maggiori et al. 2016)	CNN	727K	Supervised	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	88.9	
(Sherrah 2016)	CNN	3.26M	Supervised	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	89.1	
(Audebert, Le Saux, and Lefèvre 2016)	CNN	32M	Transfer learning & fine-tuning	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	89.8	
(Marmanis, Schindler, et al. 2016)	CNN	806M	Transfer learning & fine-tuning	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	90.3	
(Yongcheng Liu et al. 2018)	CNN	481M	Transfer learning & fine-tuning	ISPRS Vaihingen	9 cm	3 (DSMs not used)	0.47	Overall accuracy	91.1	
(C. Zhao et al. 2017)	SAE	20.8K	Unsupervised & fine-tuning	Kennedy Space Center	18 m	224	0.1	Overall accuracy	93.5	91.1
(Yushi Chen et al. 2016)	CNN	5.85M	Supervised	Kennedy Space Center	18 m	224	0.1	Overall accuracy	97.1	95.7
(Y. Chen et al. 2014b)	SAE	8.72K	Unsupervised & fine-tuning	Kennedy Space Center	18 m	176	0.6	Overall accuracy	98.8	98.7
(Haut et al. 2018)	CNN	8.8M	Supervised+	Kennedy Space Center	18 m	224	0.85	Overall accuracy	100	94.4
(Ishii et al. 2015)	CNN	60M	Supervised	Landsat 8	30m	3	0.35	F1	71	37.2
(Mou, Bruzzone, and Zhu 2018)	CNN+RNN		Supervised	Landsat ETM	30 m	6		Overall accuracy	98	95.7
(Karalas et al. 2015)	SAE	155K	Unsupervised & fine-tuning	MODIS	500 sq.m	7		Average precision	62.8	
(Weixun Zhou et al. 2017)	CNN	126M	Transfer learning & fine-tuning	Other	0.5m	3		ANMRR	0.04	
(Kemker, Salvaggio, and Kanan 2018)	CNN	11.9M	Supervised+	Other	4.7 cm	6	0.25	Average accuracy	57.3	29.6

Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
(Kemker, Salvaggio, and Kanan 2018)	CNN	69M	Supervised+	Other	4.7 cm	6	0.25	Average accuracy	59.8	29.6
(Bittner, Cui, and Reinartz 2017)	CNN	134M	Transfer learning & fine-tuning	Other	0.5m	1		F1	70	
(Lagrange et al. 2015)	CNN	141M	Transfer learning	Other	5 cm	4	0.6	Overall accuracy	72.4	70.2
(Y. Cao, Niu, and Dou 2016)	CNN	60M	Transfer learning & fine-tuning	Other		3		F1	72.4	
(Ji et al. 2018)	CNN	102K	Supervised+	Other	15 m	4	0.85	Overall accuracy	79.4	78.5
(Fu et al. 2017)	CNN		Supervised	Other	1m	3	0.9	F1	79.5	61.5
(Tang et al. 2017)	CNN		Transfer learning & fine-tuning	Other		3		Average precision	79.5	
(Huang, Zhao, and Song 2018)	CNN	39M	Transfer learning & fine-tuning	Other	0.5 m	4	0.57	Overall accuracy	80	71.8
(F. Chen et al. 2018)	CNN		Transfer learning & fine-tuning	Other	8 , 16 m	3		Average precision	80	
(X. Chen et al. 2013)	DBN	4.2M	Unsupervised & fine-tuning	Other		3	0.2	F1	81.7	78.4
(Marcos et al. 2018)	CNN	430K	Supervised	Other	5 cm	4	0.7	Overall accuracy	82.6	
(Cheng et al. 2017)	CNN	14.7M	Transfer learning	Other	30m		0.2	Overall accuracy	84.3	
(Yanfei Liu et al. 2018)	CNN		Supervised	Other	4 m (MSI), 1 m (Pan)	3	0.8	Overall accuracy	85	84.7
(Yongcheng Liu et al. 2018)	CNN	481M	Transfer learning & fine-tuning	Other	1 m	3	0.93	F1	85.6	
(Geng et al. 2015)	SAE	28.4K	Unsupervised & fine-tuning	Other	0.38m	1	0.5	Overall accuracy	88.1	76.9
(Lguensat et al. 2017)	CNN	177K	Supervised	Other		1	0.18	Overall accuracy	88.6	
(Han et al. 2018)	CNN	286M	Semisupervised	Other	30m			Overall accuracy	88.6	
(W. Zhao et al. 2015)	DBN	379K	Unsupervised & fine-tuning	Other	0.6m	1	0.7	Overall accuracy	88.9	85.6
(C. Zhang, Sargent, Pan, Li, et al. 2018)	CNN	226K	Supervised	Other	50 cm	4	0.6	Overall accuracy	89.5	79.5
(C. Zhang, Pan, et al. 2018)	CNN	17K	Supervised	Other	50 cm	4	0.5	Overall accuracy	89.6	
(C. Zhang, Sargent, Pan,	CNN	17K	Supervised	Other	50 cm	4	0.7	Overall accuracy	89.8	81.2

Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
Gardiner, et al. 2018)										
(Vakalopoulou et al. 2015)	CNN	60M	Transfer learning	Other	0.6m	4	0.4	Average precision	90	
(Qayyum et al. 2017)	CNN	6.61M	Transfer learning	Other	15cm	3	0.8	Overall accuracy	90.3	83.1
(Cheng, Han, and Lu 2017)	CNN	134M	Transfer learning & fine-tuning	Other	30m		0.8	Overall accuracy	90.3	
(F. Zhang, Du, and Zhang 2015)	SAE	90.3K	Unsupervised	Other	1 m	3	0.25	Overall accuracy	90.8	90
(C. Zhang, Pan, et al. 2018)	CNN	17K	Supervised	Other	50 cm	4	0.5	Overall accuracy	90.9	
(C. Zhang, Sargent, Pan, Li, et al. 2018)	CNN	226K	Supervised	Other	50 cm	4	0.6	Overall accuracy	90.9	80.4
(C. Zhang, Sargent, Pan, Gardiner, et al. 2018)	CNN	17K	Supervised	Other	50 cm	4	0.7	Overall accuracy	91	81.7
(W. Zhao and Du 2016)	CNN		Supervised	Other	1.8m	8	0.15	Overall accuracy	91.1	
(Huang, Zhao, and Song 2018)	CNN	39M	Transfer learning & fine-tuning	Other	1.24 m	8	0.62	Overall accuracy	91.3	80
(Khan et al. 2017)	CNN	151M	Transfer learning & fine-tuning	Other	25m	3	0.9	Overall accuracy	91.3	76.5
(Han et al. 2018)	CNN	286M	Semisupervised	Other				Overall accuracy	91.4	
(X. Chen et al. 2014)	CNN	395K	Supervised	Other		1		F1	91.6	79.3
(L. Zhang, Shi, and Wu 2015)	CNN	44M	Transfer learning	Other		3	0.75	F1	91.8	
(Khan et al. 2017)	CNN	151M	Transfer learning & fine-tuning	Other	25m	3	0.9	Overall accuracy	92	74.1
(F. Zhang, Du, and Zhang 2017)	CNN	266K	Supervised	Other	1.2 m	3	0.8	Overall accuracy	92.4	
(Pan, Shi, and Xu 2018)	CNN			Other	1 m	84		Overall accuracy	93.2	
(S. Liu et al. 2018)	CNN	28.4M	Transfer learning & fine-tuning	Other		3	0.67	Overall accuracy	93.4	78
(Basu et al. 2015)	DBN	3.6K	Unsupervised & fine-tuning	Other		4	0.8	Overall accuracy	93.9	

Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
(Längkvist et al. 2016)	CNN	1.91M	Unsupervised & fine-tuning	Other	0.5 m	6	0.7	Overall accuracy	94.5	
(Ji et al. 2018)	CNN	102K	Supervised+	Other	4 m	4	0.17	Overall accuracy	94.7	93.5
(Rezaee et al. 2018)	CNN	53.9M	Transfer learning & fine-tuning	Other	5 m	5 (3 used for CNN)	0.46	Overall accuracy	94.8	
(Cui et al. 2018)	CNN	9.7K	Supervised	Other	2m (MSI), 0.5m (Pan)	8 (MSI) + Pan	0.8	Overall accuracy	94.8	
(Yanfei Liu et al. 2018)	CNN		Supervised	Other	2 m	3	0.8	Overall accuracy	94.8	80.3
(Xing, Ma, and Yang 2016)	SAE	52.8K	Unsupervised	Other	30 m	224	0.5	Overall accuracy	95.5	96.9
(M. Gong et al. 2017)	SAE	81K	Unsupervised & fine-tuning	Other	2m	4	0.5	Overall accuracy	95.7	94.4
(Z. Ma et al. 2016)	CNN		Supervised	Other		4	0.8	Overall accuracy	96	
(W. Zhao, Du, and Emery 2017)	CNN		Supervised	Other	0.5 m	8	0.1	Overall accuracy	96.3	66.5
(Han et al. 2018)	CNN	286M	Semisupervised	Other		3		Overall accuracy	96.8	
(J. Hu et al. 2017)	CNN		Supervised	Other	1m	161		Overall accuracy	97	93.6
(Yongtao Yu et al. 2016)	DBN	2.43M	Unsupervised & fine-tuning	Other	0.27m	3		F1	97	
(Hao Wu and Prasad 2018)	CNN+RNN		Semisupervised	Other	1 m	360		Overall accuracy	97.3	95.2
(S.-H. Wang et al. 2018)	CNN	252K	Supervised	Other		3	0.74	Overall accuracy	97.3	93.7
(Basu et al. 2015)	DBN	3.6K	Unsupervised & fine-tuning	Other		4	0.8	Overall accuracy	97.9	
(Xu et al. 2018)	CNN		Supervised+	Other	1 m	63+1	0.03	Overall accuracy	97.9	92.7
(Nogueira, Penatti, and Santos 2017)	CNN	5M	Transfer learning & fine-tuning	Other	0.5 m	3	0.6	Overall accuracy	98	90
(Y. Tao et al. 2018)	CNN		Supervised	Other	0.5 ~ 4 m	4	0.008	Overall accuracy	98.4	89.2
(Z. Ma et al. 2016)	CNN		Supervised	Other		4	0.8	Overall accuracy	98.4	
(X. Gong et al. 2018)	CNN	139M	Transfer learning & fine-tuning	Other	2 m	3	0.8	Overall accuracy	98.5	77.7
(Jun Wang et al. 2015)	CNN	438K	Supervised	Other		3	0.6	Overall accuracy	98.7	
(F. Zhang, Du, and Zhang 2016)	CNN		Supervised	Other	1 m	3	0.2	Overall accuracy	98.8	

Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
(Qian Weng et al. 2018)	CNN	3.4M	Transfer learning	Other			0.25	Overall accuracy	98.8	91.3
(X. Gong et al. 2018)	CNN	139M	Transfer learning & fine-tuning	Other		3	0.8	Overall accuracy	98.8	
(Ji et al. 2018)	CNN	107K	Supervised+	Other	4 m	4	0.03	Overall accuracy	98.9	96.5
(Maggiori et al. 2017)	CNN	459K	Supervised	Other	1 m	3	0.9	Overall accuracy	99.5	94.9
(Y. Li, Zhang, and Shen 2017)	CNN	128K	Supervised	Other	30 m	242	0.5	Overall accuracy	99.6	
(Basaeed, Bhaskar, and Al-Mualla 2016)	CNN	56.4K	Supervised	Other	30m	10	0.75	Overall accuracy	99.7	
(Qian Weng et al. 2018)	CNN	3.4M	Transfer learning	Other			0.5	Overall accuracy	99.7	
(W. Zhao, Du, and Emery 2017)	CNN		Supervised	Pavia Center	1.3 m	103	0.1	Overall accuracy	96.3	92.98
(Shi and Pun 2018)	CNN	673K	Supervised	Pavia Center	1.3 m	103	0.001	Overall accuracy	97	
(Aptoula, Ozdemir, and Yanikoglu 2016)	CNN	1.31M	Supervised	Pavia Center	1.3 m	103	0.05	Kappa	97.4	
(Zabalza et al. 2016)	SAE	2.4K	Unsupervised	Pavia Center	1.3 m	103	0.05	Overall accuracy	97.4	97.4
(Ben Hamida et al. 2018)	CNN	3681	Supervised	Pavia Center	1.3 m	103	0.05	Overall accuracy	98.9	
(C. Tao et al. 2015)	SAE		Unsupervised	Pavia Center	1.3 m	103	0.05	Overall accuracy	99.6	
(W. Zhao and Du 2016)	CNN		Supervised	Pavia Center	1.3 m	103	0.05	Overall accuracy	99.7	97.7
(Makantasis et al. 2015)	CNN	10.9K	Supervised	Pavia Center	1.3 m	103	0.8	Overall accuracy	99.9	99
(Ghamisi, Chen, and Zhu 2016)	CNN	188K	Supervised	Pavia University	1.3 m	103	0.1	Overall accuracy	83.4	78.2
(Mou, Ghamisi, and Zhu 2018)	CNN	1.39M	Unsupervised & fine-tuning	Pavia University	1.3 m	103	0.1	Overall accuracy	87.4	79.9
(Hao Wu and Prasad 2018)	CNN+RNN		Semisupervised	Pavia University	1.3 m	103		Overall accuracy	88.4	81.2

Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
(Ding et al. 2017)	CNN	226K	Unsupervised	Pavia University	1.3 m	100	0.5	Overall accuracy	90.6	
(W. Hu et al. 2015)	CNN	80.6K	Supervised	Pavia University	1.3 m	103	0.05	Overall accuracy	92.6	90.5
(Yue et al. 2015)	CNN	182K	Supervised	Pavia University	1.3 m	103		Overall accuracy	95.2	85.2
(Xing, Ma, and Yang 2016)	SAE	212K	Unsupervised	Pavia University	1.3 m	103	0.5	Overall accuracy	96	93.6
(W. Zhao et al. 2015)	CNN	239K	Unsupervised	Pavia University	1.3 m	103	0.1	Overall accuracy	96.4	85.2
(Wei Li et al. 2017)	CNN	57.9K	Supervised	Pavia University	1.3 m	103	0.05	Overall accuracy	96.5	90.6
(W. Zhao and Du 2016)	CNN		Supervised	Pavia University	1.3 m	103	0.1	Overall accuracy	96.8	80.1
(Ben Hamida et al. 2018)	CNN	6862	Supervised	Pavia University	1.3 m	103	0.05	Overall accuracy	97.2	
(Paoletti et al. 2018)	CNN	173M	Supervised	Pavia University	1.3 m	103	0.04	Overall accuracy	97.8	
(Aptoula, Ozdemir, and Yanikoglu 2016)	CNN	1.31M	Supervised	Pavia University	1.3 m	103	0.1	Kappa	97.9	
(Shi and Pun 2018)	CNN	673K	Supervised	Pavia University	1.3 m	103	0.01	Overall accuracy	98.5	
(Y. Chen et al. 2014b)	SAE	29K	Unsupervised & fine-tuning	Pavia University	1.3 m	103	0.6	Overall accuracy	98.5	97.4
(C. Tao et al. 2015)	SAE		Unsupervised	Pavia University	1.3 m	103	0.1	Overall accuracy	98.6	
(X. Ma, Geng, and Wang 2015)	SAE	10K	Unsupervised & fine-tuning	Pavia University	1.3 m	103	0.1	Overall accuracy	98.7	
(X. Sun et al. 2017)	SAE	30.2K	Semisupervised	Pavia University	1.3 m	103	0.1	Overall accuracy	98.7	91.1
(Xu et al. 2018)	CNN		Supervised+	Pavia University	1.3 m	103	0.04	Overall accuracy	99.1	89.9
(Yushi Chen, Zhao, and Jia 2015)	DBN		Unsupervised & fine-tuning	Pavia University	1.3 m	103	0.5	Overall accuracy	99.1	98.4
(Y. Li, Zhang, and Shen 2017)	CNN	110K	Supervised	Pavia University	1.3 m	103	0.5	Overall accuracy	99.4	
(Makantasis et al. 2015)	CNN	10.9K	Supervised	Pavia University	1.3 m	103	0.8	Overall accuracy	99.6	93.9
(H. Zhang et al. 2017)	CNN		Supervised	Pavia University	1.3 m	103	0.05	Overall accuracy	99.7	

Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
(Yushi Chen et al. 2016)	CNN	5.85M	Supervised	Pavia University	1.3 m	103	0.1	Overall accuracy	99.7	97.7
(Weixun Zhou et al. 2017)	CNN	126M	Transfer learning & fine-tuning	RSSCN7		3		ANMRR	0.3	
(Zou et al. 2015)	DBN	3.1M	Unsupervised & fine-tuning	RSSCN7		3	0.5	Average accuracy	77	
(Hang Wu et al. 2016)	SAE	2.53M	Unsupervised	RSSCN7		3	0.5	Overall accuracy	90.4	
(W. Hu et al. 2015)	CNN	80.6K	Supervised	Salinas	3.7 m	220	0.05	Overall accuracy	92.6	91.7
(Wei Li et al. 2017)	CNN	57.9K	Supervised	Salinas	3.7 m	204	0.05	Overall accuracy	94.8	92.9
(Xu et al. 2018)	CNN		Supervised+	Salinas	3.7 m	204	0.06	Overall accuracy	97.7	92.2
(X. Ma, Geng, and Wang 2015)	SAE	37.7K	Unsupervised & fine-tuning	Salinas	3.7 m	204	0.01	Overall accuracy	98.3	
(Makantasis et al. 2015)	CNN	10.9K	Supervised	Salinas	3.7 m	224	0.8	Overall accuracy	99.5	94
(Haut et al. 2018)	CNN	8.9M	Supervised+	Salinas	3.7 m	204	0.5	Overall accuracy	99.9	91.1
(Weixun Zhou et al. 2017)	CNN	126M	Transfer learning & fine-tuning	UC Merced	1 ft	3		ANMRR	0.33	
(Weixun Zhou et al. 2015)	SAE	51.6K	Unsupervised	UC Merced	1 ft	3		Average precision	64.5	
(F. Zhang, Du, and Zhang 2015)	SAE	301K	Unsupervised	UC Merced	1 ft	3	0.8	Overall accuracy	82.7	81.7
(Romero, Gatta, and Camps-Valls 2016)	CNN	49.1M	Unsupervised	UC Merced	1 ft	3	0.8	Overall accuracy	84.5	
(Yang Yu et al. 2017)	CNN	24.6M	Unsupervised	UC Merced	1 ft	3	0.8	Overall accuracy	88.57	81.7
(Marmanis, Datcu, et al. 2016)	CNN	155M	Transfer learning & fine-tuning	UC Merced	1 ft	3	0.7	Overall accuracy	92.4	
(Hang Wu et al. 2016)	SAE	2.53M	Unsupervised	UC Merced	1 ft	3	0.5	Overall accuracy	92.7	
(Qian Weng et al. 2017)	CNN	60M	Transfer learning	UC Merced	1 ft	3	0.7	Overall accuracy	93.4	
(Luus et al. 2015)	CNN	920K	Supervised	UC Merced	1 ft	3	0.8	Overall accuracy	93.5	
(F. Zhang, Du, and	CNN		Supervised	UC Merced	1 ft	3	0.8	Overall accuracy	94.5	

Reference	*Network Type*	*# of parameters*	*Learning type*	*Dataset*	*Spatial resolution*	*# of channels*	*Training proportion*	*Metric type*	*Deep network*	*SVM (Non deep)*
Zhang 2016)										
(Han et al. 2018)	CNN	286M	Semisupervised	UC Merced	1 ft	3		Overall accuracy	94.5	
(Yanfei Liu et al. 2018)	CNN		Supervised	UC Merced	1 ft	3	0.8	Overall accuracy	95.6	92.9
(W. Zhou, Shao, and Cheng 2016)	CNN	126M	Transfer learning & fine-tuning	UC Merced	1 ft	3	0.8	Overall accuracy	96.48	92.3
(F. Hu et al. 2015)	CNN	19.6M	Transfer learning	UC Merced	1 ft	3	0.8	Overall accuracy	96.9	
(Castellucci o et al. 2015)	CNN	5M	Transfer learning	UC Merced	1 ft	3		Overall accuracy	97.1	
(Gu et al. 2018)	CNN	117M	Transfer learning	UC Merced	1 ft	3	0.8	Overall accuracy	97.1	81.7
(X. Gong et al. 2018)	CNN	139M	Transfer learning & fine-tuning	UC Merced	1 ft	3	0.8	Overall accuracy	98.3	77.4
(Penatti, Nogueira, and Santos 2015)	CNN	204M	Transfer learning	UC Merced	1 ft	3	0.8	Average accuracy	99.4	81
(Nogueira, Penatti, and Santos 2017)	CNN	5M	Transfer learning & fine-tuning	UC Merced	1 ft	3	0.6	Overall accuracy	99.5	90
(F. Hu et al. 2015)	CNN	44.1M	Transfer learning	WHU-RS19		3	0.6	Overall accuracy	98.6	

CHAPTER 4 (MANUSCRIPT3):

Large area land cover mapping using deep neural networks and Landsat time-series observations

Abstract

Land cover mapping is an important activity for many applications in high-level planning and for monitoring of natural resources and forestry/agriculture sector. In this paper we present our results on employing deep neural networks for land cover classification over selected regions within all United States ecoregions. Our work is based on Landsat data, but we augmented it by spectral indices, spatial convolutional-based features and texture features generated from Landsat data, plus topography data from another dataset. No other data source was needed. This combination is applied to a hybrid recurrent/multilayer neural network to process all spectral, spatial, and temporal dimensions of data and predict the land cover class among seven principal types. Network optimization was done in multiple stages by first selecting the best combination of input features and then trying many different network configurations and input data sizes. Our best network consisted of 11 different layers of convolutional, recurrent, and dense neural network layers with about 2.7 million parameters. The trained network was then tested on different configurations such as individual ecoregion blocks or limited sensor availability and available scenes per year.

The best achieved overall accuracy over the whole evaluation dataset was 98.0% with average and minimum class F1 of 98.2% and 96.9%. Comparison was also made with non-recurrent neural network and traditional classifiers such as SVM and Random Forest. SVM was not scalable to our dataset size and we found the other two classifiers' performance considerably

below our selected deep network configuration. Although the performance over all blocks is promising, our tests on individual blocks showed lower performance on some of them, which may be due to lack of enough samples or extreme local conditions in some classes. We also found that including two Landsat sensors (5/7 or 7/8) provide gain of about 4.5% in overall accuracy and average F1 over single-sensor experiments, and it also showed better performance with limited number of input scenes per year. We also tested an ensemble of up to 10 selected models and found that ensembling can increase individual block performance, and this gain was more for blocks that experienced lower performance on single model simulations.

4.1. Introduction

4.1.1. Global land cover mapping

Land cover mapping is the process of compiling geographical data and creating thematic maps to delineate different land regions and assign desired labels to them based on features that make up the ground. This task has been under continuous attention for many years and has found many applications from land and agriculture planning and forestry and wildlife habitat monitoring to environmental impact evaluation, fire risk assessment, urban studies, and even human health risks (Vogelmann et al. 2001). It has been more pronounced in recent years as a basic tool for studies on subjects such as climate change and conservation planning, particularly when re-evaluated continuously (Fry et al. 2011).

Our emphasis here is on regional and global studies, for which remote sensing tools and satellite sensors play a vital role. The remotely sensed data is typically in the form of raster imagery in multiple electromagnetic bands, commonly complemented with many other maps or tabular data. The mapping process itself will then be a combination of image processing and data mining tasks with all geoprocessing and photogrammetry considerations in place (for good examples see US National Land Cover Dataset – NLCD – reference articles by Vogelmann et al. 2001, Homer et al. 2007, or Fry et al. 2011).

The review paper of Grekousis et. al (2015) discusses the specifications, pros and cons of 21 global and 43 regional land cover mapping products which covers spatial resolutions from 30 m to 1 km using Landsat, MODIS, MERIS, and other satellite platforms. Pérez-Hoyos et al. (2017) also review seven global land cover maps for cropland classification. In our research we are concentrating on medium-resolution spatial and temporal sensors such as Landsat, because the spatial resolution of 30 m and temporal resolution of 16 days are reasonable selections for global land cover mapping application, also we have Landsat data globally available and free. According to et al. (2015), the best reported global overall accuracy among the above-mentioned products (by the date of their paper) was achieved by GlobeLand30 using Landsat 2010 data at 80.3% (Jun Chen et al. 2015) and FROM-GLC at up to around 70% (L. Yu et al. 2014) which are not very high accuracies. The situation in continental or regional maps is not much better: Assessed accuracy of US-covering NLCD maps is around 80% (Fry et al. 2011) while another product covering South American continent is evaluated at 89% accuracy (Giri and Long, 2014), and another reported accuracy is 71.7% over China (L. Hu et al. 2014). These quantities are not satisfactory and should be improved. Additional work has been done on refining global land cover datasets using additional data sources such as in Yu et al. (2013) by merging Landsat plus MODIS, or fusing several different global land cover maps in Feng and Bai (2019) and producing one enhanced global product, but their overall accuracies did not improve beyond 71%. In addition to the above, there have been some more recent research on global land cover classification such as improving FROM-GLC model accuracy to 72% using 10 m resolution

Sentinel-2 data (Gong et al. 2019). USGS has also made a new product named LCMAP that has achieved an overall accuracy of up to 87% in continental US (Brown et al. 2020).

However, all of the above results have been obtained by unsupervised or supervised classification methods based on decision tree, random forest, or SVM. The LCMAP product is different as it uses a well-known algorithm named Continuous Change Detection and Classification (CCDC) for harmonic modelling of Landsat observation time series (Zhu and Woodcock, 2014) and passes the model parameters to the classifier, which is an enhanced version of classification trees.

Note that there is no universal 'best' land cover class definition as the class definitions are always problem dependent. Classes are normally separated based on user-defined preferences and thresholds for intensity level selection, and decision making for mixed pixels may vary from case to case. The global land cover maps mentioned before correspond to classification systems of as low as 5 classes to as high as 30 classes or more, which, as studied by Yang et al. (2017), turns out to be incompatible and inconsistent and need to be harmonized for terminology, semantic interpretation, and legend translation. Also, there is considerable difference between overall and per class performance, which can vary widely depending on classification system. For example, Alhassan et al. (2020) reported 90% overall accuracy but per-class precision varied from 52 to 97% for the best design of their land cover classifier for an application in lower Manitoba, Canada. Therefore, there is still high demand for more advanced classifiers who can at the same time achieve higher overall accuracy and better and more balanced per-class accuracies.

4.1.2. Deep models for land cover classification

Deep learning has been used for more than a decade in many applications such as computer vision, speech recognition, and natural language processing. Deep learning methods have also found their way into remote sensing applications as well in remote sensing (RS) image pre-processing, scene classification, pixel-based classification and image segmentation, and target detection. Scene and pixel classification are the most studied applications with land cover classification typically falling into the pixel classification category. Phiri and Morgenroth (2017) stated that land cover classification has progressed through several phases from early unsupervised and supervised classification methods in the late 1970s to use of machine learning algorithms, object-based methods, hybrid methods, and recently more advanced classifiers including deep neural networks. However, currently available global land cover maps are still based on conventional unsupervised or supervised classification and none of them have tried new deep learning methods. The aim of using deep networks is to employ their inherent capability to discover useful features in data automatically (contrary to non-deep methods that rely on hand-crafted features). As we discussed in chapter 3 of this dissertation, deep implementations based on convolutional networks has been more popular and achieved better results by generating powerful spatial and spectral features. However, utilizing time-series and temporal dimension is less practiced in the past literature and is gaining more recently. Here we will review the past research on deep learning by emphasizing those cases working on data from medium-resolution sensors such as Landsat and Sentinel and utilize temporal data.

Basically, deep networks based on just spectral information may not have an advantage over traditional classifiers without employing all data dimensions and enough amount of training data. For example Abdi (2020) used Sentinel-2 data for a region in Sweden and showed SVM to be the best classifier compared to their deep learning model and Random Forest. We also used Landsat data from different regions in US as described in chapter 2 of this dissertation, and achieved similar performance for both conventional and deep classifiers with overall accuracies varying from region to region and a worst case regional accuracy of 64%.. Accuracy can be improved by including spatial dimension and particularly by introducing convolutional networks and ensembling. For example X. Zhao et al. (2019) applied deep convolutional network to regions in China and obtained overall accuracies around 74-78% using Landsat data. J. Wang et al. (2017) reported overall accuracy of 75% using fully convolutional network trained on NLCD data over Kansas. Verma and Jana (2020) applied convolutional neural network (CNN), artificial neural network (ANN), Random Forest (RF), and SVM classifiers to Sentinel-2 data and got the best accuracy of 82% on CNN (they enhanced ANN, RF, and SVM classification results with post-processing to include spatio-textural information by majority voting).

One early example of using ensemble of classifiers is Mountrakis et al. (2009), who used a hierarchical structure of nodes and expert decision making to find the best combination of SVM, neural network, and decision tree classifiers to classify impervious land in the Las Vegas area based on Landsat data. Using this hierarchical combination, they achieved overall accuracy of 92.4%. More recently, Kussul et al. (2017) used ensemble of Landsat and Sentinel-2 data to classify crop type in Ukraine and obtained an overall accuracy of 94.6%. Shendryk et al. (2019) used ensemble of convolutional networks on some scenes using Sentinel-2 and PlanetScope sensors data and obtained overall accuracies ranging from 74% to 90%, depending on which test and training sets were used, and this clearly shows limitation of generalizability of classification results on local data. They classified clouds as well, so didn’t need the do pre-processing to eliminate cloudy pixels. Alhassan et al. (2020) used advanced deep learning techniques and combined convolutional/adversarial architecture to reach overall accuracy of 90% on Landsat data for land cover mapping in a region in Canada.

Recurrent-type neural networks (RNNs) are of special interest to utilize the temporal dimension of data because they are designed to process time-series data efficiently (see Salehinejad et al. (2017) for a detailed review of recurrent neural networks). RNN has entered the remote sensing literature in 2016 (see Lyu et al. 2016) for change analysis, and then found applications in scene or land cover classification and forecasting values with good results. The basic form of employing temporal information in classification is to feed the recurrent network directly with time series of imagery bands. This line of research has been tried in the past for crop type classification by Rußwurm and Körner (2017) on Sentinel-2 data with overall accuracy of 84.4%, and by Sun et al. (2019) on Landsat data with overall accuracy about 89% for land cover classification. Temporal dimension and structure of RNN enables us to increase performance by incorporating more time stamps in processing, which is pursued in different forms in Rußwurm and Körner (2017) and Zhao et al. (2019). Campos-Taberner et al. (2020) also report on a bidirectional Long Short-Term Memory (LSTM) implementation on Sentinel-2 data to get overall accuracy of 98.7% for crop type classification, while the best non-deep classifier was RF with accuracy of 94.9%.

Adding spatial information is done in a variety of forms in different articles, such as flattening spatial information (converting a 2-D spatial neighborhood around a pixel to a 1-D array) and feeding a recurrent network with a temporal sequence of flattened spatial-spectral data (Sharma et al. 2018). But the more popular approach is to put a convolutional neural network before the RNN step. This idea has been applied to crop type classification in Pelletier et al. (2019) using a stack of Formosat-2 images. CNN implementation can be integrated in LSTM cell, as Rußwurm and Körner (2018) used to process top of atmosphere Sentinel-2 data for 17-class crop type classification with overall accuracy of 90%, and they showed that their network is even capable of detecting the cloudy scenes by itself. The CNN part can also be put after the recurrent part as demonstrated by Mazzia et al. (2019) who reported overall accuracy of 96.5% using Sentinel-2 data for crop type classification. The CNN part can also be placed in parallel to recurrent network as used by Interdonato et al. (2018), where they aggregated output of either of RNN and CNN branches for all time stamps in one data vector, combined two aggregated feature vectors and classified the result. Using Sentinel-2 data, they reported overall accuracies of 86.1% and 96.8% for two land cover classification case studies. Parallelism has gone further by fusion of Landsat, high-resolution imagery via National Agriculture Imagery Program (NAIP) dataset, climate data via PRISM dataset, and terrain topography data in Chang et al. (2019). In this work for each pixel and its neighborhood, Landsat data were processed by convolutional LSTM subnetwork, NAIP data were processed by a special convolutional-dense network, climate and terrain data were processed by another dense network, and resulting features are all concatenated and classified for the final land cover type. They reported average class producer accuracy of 92% and also conducted additional regression estimations (such as above ground biomass, canopy cover, and two other quantities) using these features.

Fusion of optical and radar data is also a hot topic addressed in some research. For example. Liao et al. (2020) extracted backscatter features from multiple observations by Radarsat-2 and combined it with optical data from Venµs satellites to classify crop type. They tried different deep and conventional classifiers and reached overall accuracies of better than 96% for their best data fusion strategy with all classifiers.

There is also some research based on specialized and custom-designed network architectures or processing algorithms to process temporal dimension in recurrent networks. One example is Ienco et al. (2017) who used a combination of pixel data and spatial object-based data to do land cover and crop type classification by an LSTM network and reached overall accuracy between 75.3% and 84.6% for different datasets. Feng et al. (2019) used a special technique called deformable convolutional networks on each time stamp of imagery taken by Sentinel-1 and Sentinel-2 for their study area, fused the features together in another special neural network and did the final land cover classification to reach overall accuracy of 93.8%. A promising work was also done by Jia et al. (2019) in which a complex architecture based on General Adversarial Network (GAN) principle is presented. The network accepts MODIS and Sentinel-2 data and processes their time series and uses MODIS data to estimate and fill the gaps in Sentinel-2 data, thereby enhancing the crop type classification.

4.1.3. Research objectives and contributions

One of the main issues in most of the research mentioned above is the limitation of geographical or temporal span of the training dataset and subsequently the resulting model. The performance statistics reported are also very dependent on the locality and the land cover classification type, for example, fine crop type classification typically results in lower accuracy than broad and general land cover classification (because of the more similarity between reflection profile of many crops), so the results cannot be generalized. Lack of remote sensing training data is mentioned in many articles but never resolved, and to the best of our knowledge, our work is the first study to build a deep model by training it over many years and covering all ecoregions of the United States. This is a very challenging situation because of the huge diversity in ecoregion specifications and inherent differences between pixels having the same land cover label in different geographic areas. For example, type and density of a “forest” setting in Colorado is naturally very different than Florida or New York state, but we want a model to be trained on all regions with a very high accuracy and be transferable to every individual region with minimal loss in performance. Such a model will be of a very high practical value, either directly on any desired area or by being fine-tuned to other local areas to overcome possible deficiencies. We investigated the potential of deep learning models by employing recurrent neural networks and training them with a big dataset consisting of Landsat observations, climate and topography data, spectral indices, texture metrics, and convolutional-based spatial features to achieve this goal. As discussed in the introduction, global land cover mapping accuracies are not satisfactory and there has been no product available to take benefit of deep learning methodologies in this field. Therefore, the main contribution of this paper is to harvest the extensive Landsat archive and combine it with state-of-the-art deep learners to improve classification accuracy of large area mapping.

After this introduction, the rest of the paper is organized as follows: Section 4.2 presents the description of our data sources, including the reference data and calibration/test data. Section 4.3, Methods, details the steps done to generate the data, setup the simulation environment, architect the system, and assess the results. Section 4.4 provides the results of our experiments and analyzing them from different viewpoints. Last section, section 4.5, will provide the concluding discussion and future related works.

4.2. Data

4.2.1. Study area and reference labels

Our study area was the entire conterminous United States, which is divided into 84 ecoregions at level III classification by EPA as shown in Figure 4-1. As defined in https://www.epa.gov/eco-research/ecoregions, ecoregions are areas where ecosystems (and the type, quality, and quantity of environmental resources) are generally similar. We received land cover samples for 2717 10 km ×10 km blocks for each ecoregion from USGS, which was originally produced for the USGS Land Cover Trends Project (https://www.usgs.gov/centers/wgsc/science/land-cover-trends). The blocks composed of 333×333 pixels (at 30 m ground resolution per pixel) at Albers Conical Equal Area projection and each pixel was labeled according to a modified Anderson classification system (USGS, 2014) to designate the pixel's dominate land cover. The 11 classes were Water, Developed/Urban, Mechanically Disturbed (human-induced distrubances), Barren, Mining, Forests/Woodlands, Grassland/Shrubland, Agriculture, Wetland, Nonmechanically Disturbed (disturbances caused by natural causes such as caused by wind, floods, fire, animals), and Ice/Snow. The data were dated around year 2000. We selected one representative block for each of 84 ecoregions for further investigation and refinement, as explained in the methods section. The selection criteria were high class diversity and balanced distribution of land cover types through the block, which was checked visually. Within each block, we examined possible changes within each pixel's land cover using Google Earth high-resolution imagery and selected a subset of pixels that have stable land cover over a long period. This period was generally considered to be 2005-2019 (including both start and end year) but was adjusted that for each pixel based on availability of high-resolution imagery at that location.

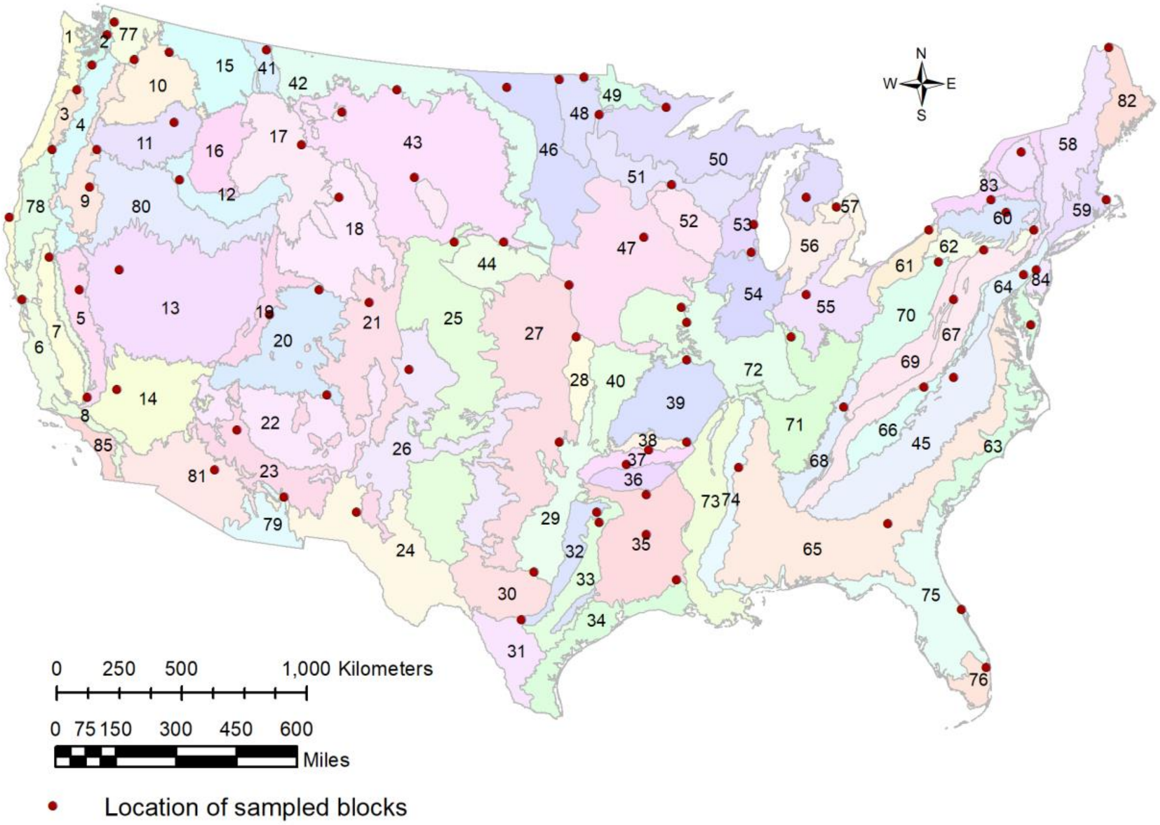


Figure 4-1: Level III ecoregions in the conterminous United States (source: https://www.epa.gov/eco-research/level-iii-and-iv-ecoregions-continental-united-states) and our selected blocks (red circles)

4.2.2. Calibration and test data

After selecting the valid points and their labels for each block in section 4.2.1, the model calibration and test data (which we name it "model data" hereafter) were generated based on Landsat Surface Reflectance data archive, available through Google Earth Engine platform. By calibration data we refer to the data that were used during model training, and the test data were used to evaluate model performance on unseen data. The Landsat data were processed to extract different spectral, spatial, and temporal domain features from it. The data have a spatial resolution of 30 m and temporal resolution of 16 days, which seems appropriate for a global land cover mapping application. But to supply the model with more information to better distinguish different land cover types (especially to mine inter-block and regional differences that are of special importance in our application), we added some climate and topography variables to our dataset. We extracted temperature and rain statistics from the PRISM dataset, which has a spatial resolution of 2.5 arc minutes and is available at temporal resolution of 1 day or 1 month; and topography information from SRTM dataset with spatial resolution of 30 m (it is a static value with no temporal dimension). All of these datasets are available freely under Google Earth Engine platform, which we used for data access and dataset generation. Each dataset may have different geospatial referencing system, but all spatial data was reprojected and resampled automatically to match the reference maps' projection and spatial resolution when processing data under Google Earth Engine.

4.2.2.1. Landsat data

We used Landsat surface reflectance Tier 1 data in our work. Being surface reflectance, these data have already been corrected for atmospheric errors. We also used Landsat radsat_qa and pixel_qa quality bits for each pixel to identify radiometric saturation and cloud or cloud shadow conditions (medium or high confidence) and removed those pixels from our candidates. The Landsat 7 errors due to SLC failure have already been processed by Google Earth Engine and those pixels are masked. Our main motivation in this work was to use deep networks to generate features that reflect spatial-spectral-temporal profiles for each land cover class so they can best be distinguished from each other. If a point has a stable land cover, we can assume that a change in its remotely sensed data is mainly attributed to phenology and therefore the same pattern of change will be repeated each year (a long-term trend can be added using other variables). Based on this assumption, we generated yearly sequences of Landsat data values. Each record of the original sequence had the Landsat Blue, Green, Red, NIR, SWIR1, and SWIR2 band values for a single observation. We incorporated all data available from Landsat 5, 7, and 8 sensors in the same sequence.

4.2.2.2. Climate data

PRISM dataset (AN81m) provides monthly average temperature and total monthly rain statistics per each pixel. PRISM also provides daily statistics, but the land cover is not affected by daily climate variations. The spatial resolution of PRISM is also lower than Landsat, but it is still appropriate for catching the climate trends and inter- and intra-block climate differences that may affect land cover.

We derived two type of variables from PRISM dataset: one set is monthly data that was extracted for the current year (year of generating Landsat sequence as described in section 4.2.2.1) and the other set was the normal long-term data, which is defined for each pixel as the average of climate variables for that pixel over 30 years. The choice of 30 years window is voluntary and we considered it from 1990 till 2020. These monthly and normal data were generated for both temperature and precipitation variables. The monthly values for each variable are grouped by the year of observation and added to the model data. Normal values are also calculated on a monthly basis (by averaging same month value over 30 years) but it is also complemented by three other aggregate variables: minimum, maximum, and average value of normal statistics over year. In other words, we put the monthly normal variables in a 2-D matrix with months in columns and years in rows, then first did an aggregation for each row over columns, then calculated minimum, maximum, and average of the above aggregate value over rows. The row aggregation for temperature was calculated as average value over the columns, but for precipitation it was calculated as the sum of values over columns.

4.2.2.3. Topography data

We also used the Shuttle Radar Topography Mission (SRTM) digital elevation data V3 product as provided in Google Earth Engine catalog and extracted elevation, slope, and aspect fields. These variables are static per pixel and do not change over year, same as climate normal statistics.

4.3. Methods

The methods section presents details for dataset generation (both reference and model data), simulation framework, and model architecture. We also discuss benchmark algorithms, and performance evaluation criteria.

4.3.1. Dataset generation

4.3.1.1. Reference labels generation

As mentioned earlier our reference labels are based on USGS land cover trends maps. However, those maps were only prepared for a single year and our visual inspection revealed many inaccurate points in the generated maps. Therefore, we decided to edit the maps and only keep the points that we are certain of their stable land cover over multiple years. This helped create several different samples of model data (yearly sequences) for one label, thereby enriching our database to include different profiles for the same land cover and make model predictions more robust. Such a revision requires having a much more accurate imagery than Landsat, and we used Google Earth because it provides the best free source of high-resolution (less than one meter) imagery over the earth, while providing historical archive as well.

To make things clearer and more objective, we reduced the land cover types from 11 in original USGS maps to 7, including water, developed, grass/shrub, forest, bare, agriculture, and wetland. No ice/snow was present in our selected points. Details on class definition is given in Appendix A, along with detailed guidelines to decide how to assign labels to the pixels while dealing with all imperfections of low quality or missing data for some years in Google Earth, mixed pixels and transitions (for example between forest and grassland), dynamic boundaries (such as in wetlands), and class priorities. We assigned higher priority for detection of developed areas (anthropogenic impact on nature) and decided to assign that to a pixel if at least 20% of its area is human developed. The next priority was given to the agricultural fields if they occupy at least 20% of the pixel area, then to the other land cover types using a simple majority rule. We also developed many detailed notes on how to distinguish farm from grassland, grass or forest from wetland, wetland from water, bare from grassland, etc. A summary of required steps for data processing and overall workflow is given in Appendix B. Note that we used only high-resolution normal color imagery for reference labels generation and only remote sensing datasets for model data (no administrative information). This would help to keep the process generalizable and applicable to other locations. It is also important to keep in mind that we dropped many pixels within each of the 84 blocks due to uncertainty and/or instability of land cover type, so our reference maps are patchy and not contiguous.

4.3.1.2. Calibration and test data generation

We ended up with about 35,000 up to 100,000 valid pixels in each of our 84 selected blocks after the reference label assignment, but with different stable time spans from 8 to 15 years and with differences in class distributions from block to block. To keep the simulation time reasonable, we selected about 7,000 to 50,000 pixels in each block to reach a total of about 1.6M (million) pixels with a specific class distribution. The detail of block class distribution is given in Appendix C, which shows the very unbalanced class distribution because of the class imbalance in the original labels. Wetland and bare were rare classes compared to the others. Our tests showed that the bare land characteristics were sufficiently different than other land cover classes to make it relatively easy to separate. But we found that the wetland and developed classes were more challenging to classify accurately. Therefore, we selected all of the edited bare and wetland pixels and also selected a bigger proportion of edited developed class pixels compared to other classes. After generating yearly sequences for each available year for each pixel, the final dataset contains approximately 21M sequences with the class distribution presented in Table 4-1.

Table 4-1: Selected final dataset class distribution

Water	Developed	Grass/Shrub	Forest	Bare	Agriculture	Wetland	Total
1,407,689	7,131,290	3,657,994	2,574,143	1,091,321	3,888,984	1,430,939	21,182,360

Each sample of our dataset is related to a specific pixel and year and consists of a sequence of Landsat and Landsat-derived features throughout the year, a vector of monthly climate variables for that year, and the fixed values of normal climate and topography variables (usage of these data will be discussed in section 4.3.3). The length of the Landsat sequence of features varied from pixel to pixel and block to block and ranged from 23 to 98 time stamps (mean of 50). These different sequence lengths required some special treatment (zero-padding) to be done when feding the data into the network. By zero-padding, extra feature records with zero values in their fields were added to the sequence to make all sequences having 100 time stamps.

As with any other neural network implementation strategy, we must divide our dataset in calibration and test partitions, used for various stages in model training and evaluation. Our approach to the partitioning was the same and what we do for each block is:

- Choosing N pair of calibration points in such a way that in each pair the training and validation points are disjoint (i.e., have no point is common),

- Choosing one set of test points in such a way that it is disjoint to all the above N sets.

Our full dataset in our final experiments was partitioned in such a way that 4/35 (about 11.5% or 2.4M sequences) were set aside as test data, and the rest was sampled N times to create N calibration sets in such a way that in each set, 28/35 (80% or 16.9M sequences) was used for training and the rest (8.5% or 1.8M sequences) for validation. Training sets were used to train the neural network during specified training epochs, while its companion validation data was used after each epoch during training to evaluate the model performance on unseen data and stop the training when it is no longer useful for generalization (i.e., prevent overfitting of the model). The reason for generating N different calibration sets was twofold: reducing the neural network performance variance that is caused by inherent randomness in its training, and enhancing our estimation of model performance on unseen data. As we trained many different model configurations with the same sets of calibration data, we needed another independent dataset to evaluate and compare their performance together and test set was fulfilling this purpose.

One approach to have a quantitatively justified decision on the number of N is presented in Iyer and Rhinehart (1999), which is based on principle of choosing the best of N experiments and gives a very general and conservative formula. The best value of N depends on the level of confidence and the acceptable generalization error, but heuristically N=10 is an acceptable norm. We determined it more practically by running the network N times, looking at the average performance, ranking the performances for different configurations, and observing when the ranking is almost steady. In our case we found that N=8 is a good start for training sets of about 500,000 samples but when size of data or network increases, it can be lowered because the model variance is decreased.

4.3.2. Network architecture and optimization

Deep networks bring a lot of network parameters and hyperparameters that need to be optimized. Given the vast availability of data in our research, the network can also go very deep and become more complex, more difficult and time consuming to train and test, and include more hyperparameters to tune. Therefore, we had no other way than proceeding with the parameter selection step by step and a simultaneous exhaustive parameter search was not possible. The most important parameters that we dealt with in our research were: 1) selection of input features, and 2) selection of network structure. Other parameters including network optimization settings (e.g., batch size, optimizer type, learning rate, and simulation stopping criteria) were tuned at the start of the work on a simple model and re-checked from time to time during simulations and also at the end on the final model.

4.3.3. Model candidates

The general schematic of our designed network is shown in Figure 4-2, which is composed of three input data sources, two multilayer dense subnets, one recurrent subnet, and one convolutional subnet, plus the final classifier. Number of neurons/cells in each layer and number of layers were investigated through our simulation framework as described in the previous section. For principles of operation of dense or recurrent neural networks and refer the reader to the standard neural network textbooks and other literature cited in section 4.2.1.

We had different model candidates based on the data dimensions or feature groups we wanted to include. We have considered three types of models that are expected to provide increasing performance and introduce them in subsections 4.3.3.1 to 4.3.3.3. Each model incorporates some of the blocks shown in Figure 4-2. For example, the convolutional processing block is not utilized in models presented in sections 4.3.3.1 and 4.3.3.2.

The base data in all models are sequences of Landsat data and other remote sensing datasets. Each sequence contains several records, and each record contains a number of features, including Landsat band values for a specific observation day plus optional spectral indices or spatial texture metrics as explained later. The whole sequence of features is applied to recurrent network to get an output after processing the last time stamp. There are different cell types proposed as building blocks for recurrent neural networks, for which LSTM and GRU (Gated Recurrent Unit) are the most used types. We tested both cell types and found LSTM performing slightly better, and it has more trainable parameters. Static data (such as topography) are in the form of 1-D vector and for each pixel, there is one vector of static features corresponding to one sequence of Landsat-based features. Parallel to processing of Landsat-based sequence by recurrent network, the static features are processed by a standard multilayer dense network (N_D layers of M_{Di} neurons each, where i is the layer index). Output features from recurrent subnet and the above mentioned dense subnet are concatenated and applied to the second multilayer dense network for further processing, which again has N_P layers of M_{Pi} neurons each. The result is applied to a classifier to determine the assigned land cover to the input pixel.

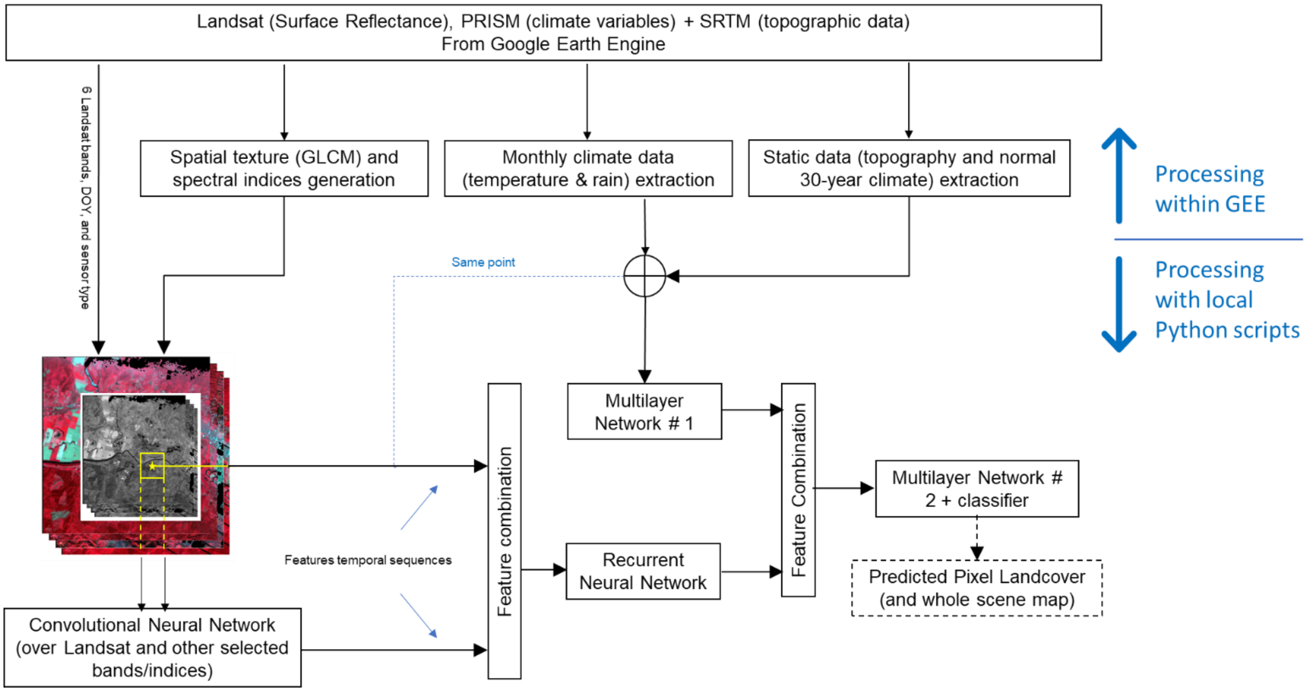

Landsat (Surface Reflectance), PRISM (climate variables) + SRTM (topographic data)
From Google Earth Engine
6 Landsat bands, DOY, and sensor type
Spatial texture (GLCM) and spectral indices generation
Monthly climate data (temperature & rain) extraction
Static data (topography and normal 30-year climate) extraction
Processing within GEE
Processing with local Python scripts
Same point
Multilayer Network # 1
Feature combination
Feature Combination
Multilayer Network # 2 + classifier
Predicted Pixel Landcover (and whole scene map)
Features temporal sequences
Recurrent Neural Network
Convolutional Neural Network (over Landsat and other selected bands/indices)

Figure 4-2: Designed system architecture

In our training and model building phase, input data were random pixels and did not constitute a whole scene. But in the same way, we can feed the trained network by a whole scene pixel by pixel and get a wall-to-wall output land cover map of the region.

4.3.3.1. Temporal LSTM (T-LSTM)

This is the base model that we start from, and it processes spectral and temporal data. We have eleven (11) base variables for each temporal observation that are included in this experiment for each pixel: Day-of-year (DOY), sensor type (Landsat 5/7/8), six Landsat surface reflectance values, and three topography variables (elevation, slope, aspect). In addition, we considered eight different spectral indices reported in different literature to be useful for identification of different ground features such as vegetation, water, built-up area, bare soil, and soil wetness, and we have provided the list of the reviewed indices and their formulas in appendix D. Among all of these indices, our experiments showed ENDISI significantly improved network performance. All spectral index calculations were done in Google Earth Engine while extracting Landsat data. We tried both monthly and normal (long-term) climate variables on our LSTM model as well, but they did not provide any benefit compared to other selected variables, therefore they were dropped from the model data. Note that we used a fixed number of layers and cells in this experiment as we were interested in input feature selection and not the network structure optimization.

4.3.3.2. Spatio-temporal LSTM (ST-LSTM)

Our next model adds spatial data dimension by including texture features and increases the network complexity to reach the performance saturation. We considered two spatial information extraction methods: Gray-Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP). The GLCM method generates a co-occurrence matrix from any image band of interest from which multiple metrics are calculated that are used to describe the texture around the pixel (Hall-Beyer, 2017a). GLCM produces various texture metrics for each point in three main groups: contrast group, orderliness group, and descriptive statistics. Each group contains several metrics, which we tried individually and together. There is no clear best metric, since this depends on the application and GLCM parameters. Hall-Beyer (2017b) looked at this issue for a classification application based on Landsat data, and recommended choosing Mean/Correlation (for general texture identification), Contrast/Dissimilarity (helpful for classes containing edge-like features), and Entropy (for more detailed texture study). We did some trials with our data and picked four metrics of dissimilarity, entropy, mean, and variance. GLCM generation requires specifying two other parameters: GLCM window size and quantization level. We fixed quantization level at 64 but generated the above four GLCM features for two window sizes (radius) of 5 and 15 pixels to serve different spatial scales. GLCM generation also requires picking a base band to do the spatial calculation on it. Looking at the other works, we found that either one of visual bands, NIR band, or an artificial band such as principal component based on visual bands has been used for this purpose. In our work, we initially took NIR band for GLCM generation but then extended this idea and generated GLCM features based on two Landsat bands (blue, NIR) and two spectral indices (DD, ENDISI). Selection of these bands and indices was a result of another experiment in which we tested performance of a sample network when fed by GLCM features generated from all bands and indices, and we picked those that performed better in this experiment. Therefore, we had 4x2x4=32 GLCM features added per pixel, to compare and pick the best combination. For simplicity, we use a three-part name to designate each GLCM feature in our work. For example, ENDISI_ent_15x64 denotes the GLCM entropy metric generated based on ENDISI band using a window size of 15 and quantization level of 64. All GLCM calculations were done in Google Earth Engine with available functions. Based on our experiments, we selected DD_ent_5x64, ENDISI_ent_15x64, and blue_savg_15x64 as the best spatial texture features to include in our model.

The LBP method finds local spatial patterns around a pixel and codifies it with numbers that can tell us if the point is a corner, edge, or middle of a homogeneous area (Ojala et al. 2002). As our initial tests did not show any benefit for LBP over GLCM, we did not use it in our main simulations.

4.3.3.3. CNN+Spatio-Temporal LSTM (CNN+ST-LSTM)

The last and most complete model builds upon the ST-LSTM model by supplementing the manually engineered spatial features (GLCM) with computer-generated convolutional features. In this model we utilized the CNN block in Figure 4-2 and added the spatial features generated by it to the input features of the recurrent network. In other words, they are added to the input features of the previous Spatio-Temporal LSTM model. The input to this convolutional block can be any of the Landsat or other available bands in the input data, and we chose the combination of six Landsat surface reflectance bands plus Elevation band (as a possible factor in building powerful spatial features) as its input. For each training point, the neighborhood data for the chosen bands was extracted. Then we used standard 2-D convolutional filters without padding for doing convolution, because we want the final output of CNN network to be a 1-D vector. For example, for an input neighborhood of size 5, we can deploy a 3x3 filter in the first convolutional layer to reduce the 5x5 input to 3x3 (5-3+1=3), and then a 3x3 filter in the second convolutional layer to reduce it to a spatial size of 1x1. Our convolutional network consists of N_C layers of M_{Ci} filters of size w_i and its output will be a vector of length M_{CNc} (number of filters in the last layer of CNN block). The rest of network is the same as Spatio-Temporal LSTM.

4.3.4. Benchmark algorithms

We compared many different configurations and input features in our simulations based on the network configuration described in section 4.3.3. We also tried the non-recurrent and non-neural network classifiers listed below to see how better the performance of our hybrid convolutional/dense/recurrent network is compared to them:

- Non-recurrent dense multilayer neural network
- SVM
- Random Forest

One important distinction between the baseline models and our proposed deep models is that in our designed network, the data has a built-in temporal dimension and is presented to the network in the form of temporal sequences. This is not the case for the baseline models, which are not recurrent in nature. As each yearly record of our dataset contains on average about 50 time stamps (i.e. Landsat scenes), combining all time stamps of one year together and provide them as simultaneous input of baseline models is not feasible due to the large data volume. Instead, we opted to feed them with just one time stamp at a time. We tried various schemes of choosing individual time stamps from sequences. To have the baseline trained models as general as possible (in terms of both time and space), we chose one time stamp at random from each sequence. This scheme retains a representative from each sequence while distributing the sampling time as uniform as possible throughout the year.

4.3.5. Accuracy assessment

One important consideration in accuracy assessment is to ensure the independency of calibration and test datasets. We guarantee spatial independence by our point selection mechanism. But for the temporal dimension we particularly need all years in calibration and test data to adapt our model to any abnormal year conditions (e.g., extreme climate) and be able to test it. This is the normal practice in other literature that deal with time dimension, for example change detection using two fixed time stamps. As we seek good balanced performance in all classes and overall accuracy is more representative of the performance of dominant class, we decided to use the F1 metric[2] for each class and then calculated the average of this value over all classes to obtain an aggregate performance measure for our models after each simulation. Along with average F1, we also calculated minimum F1 and overall accuracy (reported as test accuracy in our tables) in each simulation. Model selection was based on higher average F1, but if two models had very close average F1 values, the minimum F1 and overall accuracy were also considered to make a decision. The reported assessment is always done on test data unless

[2] F1 metric is defined for each class as the harmonic mean of the class precision and recall. It is calculated as F1 = 2 × precision × recall / (precision + recall).

otherwise specified, which consists of 2.4M sequences for the whole blocks (ecoregions) but each block may have a different share as listed in appendix C.

Note that we run each classifier multiple times to have a higher confidence of its performance. Therefore, we calculated the average of the above performance metrics over these runs and built confidence intervals around it to compare different configurations.

4.3.6. Algorithmic implementation

Development and implementation of our model and data processing steps was done on different platforms but all of the coding was done using python. The Tensorflow environment was used for model development and simulation. As shown in Figure 4-2, extraction of data was done via Google Earth Engine platform and then the data were downloaded to our local machines. It was followed by pixel sampling and calibration/test datasets generation. Although some tests and model evaluation were done on our local resources, most of the model training runs were conducted on a cluster of powerful GPU-enabled nodes available through NASA high-end computing facilities at NASA Ames research center. We used up to 56 single-GPU and 28 4-GPU nodes during different stages of model training, comparison, and selection, which also required corresponding data transfers and job scripting tasks.

We did another round of Google Earth Engine data extraction for year 2015 as a sample year using the same procedure as above for prediction of wall-to-wall maps after model selection. We did the prediction directly after downloading the dataset because no sampling is required in this case. A more detailed description of the algorithmic implementation is given in appendix B.

4.4. Results

4.4.1. Properties of selected deep neural networks

As mentioned before, we did our experiments in successive steps because of the huge number of parameter combinations. We fixed some parameters at the end of each stage, and the specific parameters of the selected models are listed in Table 4-2.

Table 4-2: Selected models specification

Model type	T-LSTM	ST-LSTM	CNN+ST-LSTM
Number of model parameters	52,663	2,297,127	2,685,287
Training parameters			
Batch size	1024	1024	1024
Optimizer	Adadelta	Adam	Adam
Optimizer parameters	Learning rate = 1.0	learning rate=0.001, AMSgrad=True	learning rate=0.001, AMSgrad=True
Final network architecture			
Layer structure:			
CNN layers: (# of filters and neighborhood size) per layer	N/A	N/A	(128,3), (96,3)
LSTM layers: # of cells per layer	48, 48, 48	320, 320. 320	340, 340, 340
Multilayer network#1: # of neurons per layer	16, 16	64,32	32, 32
Multilayer network#2: # of neurons per layer	32, 32	256, 256, 256	128, 128, 128, 128
*Dropout regularization**	0.2, 0.2, 0	0.25, 0.1, 0	0.3,0.25,0.05,0.05
Input features			
T-LSTM	DOY, sensor, Landsat SR 6bands, Topography, ENDISI		
ST-LSTM	DOY, sensor, Landsat SR 6bands, Topography, ENDISI, DD_ent_5x64, ENDISI_ent_15x64, blue_savg_15x64		
CNN+ST-LSTM	*CNN subnet input*: Landsat SR 6 bands + Elevation *Rest of network*: DOY, sensor, Landsat SR 6bands, Topography, ENDISI, DD_ent_5x64, ENDISI_ent_15x64, blue_savg_15x64		

* Dropout ratios are given as a tuple of numbers and each number belongs to one of the blocks mentioned in the Layer Structure row. If not zero, the dropout ratio will be applied to all layers of that block.

4.4.2. Comparison of baseline models and proposed deep models

We chose three other classification methods as baseline for comparison to our models: SVM (Support Vector Machines), RF (Random Forest), and non-recurrent multilayer neural network (also named MultiLayer Perceptron – MLP). Among the baseline methods SVM was quickly unable to scale with input data sizes over a hundred thousand samples and had overall accuracy under 80%, so we didn’t continue with it. RF still takes a few days to train but it is of the same order as our model, and MLP was running in less than a day (due to its simpler architecture). For selection of input features to include, we fed the RF with all spectral, spatial (texture), and climate variables mentioned before and used the implemented possibility to automatically calculate the features relative importance. This analysis showed Landsat bands, topography variables, spectral indices and ENDISI_based GLCM features, and normal climate variables of higher importance. The higher priority of elevation and climate variables for RF (while climate variables were not among selected features in recurrent network) show that recurrent network can efficiently extract the local climate-related data from the data sequence Landsat bands, while RF and MLP are fed with single time stamps and do not have such an opportunity. We tried different feature combinations for a sample classifier and found that including all features gave us the best performance, so we kept all features (101 values) as the input to RF and MLP classifiers.

For both RF and MLP there are some parameters to select: For RF the most important ones are number of trees, tree depth, and minimum leaf size, and for MLP the number of layers and number of neurons per layer. We tried different number of trees, tree depths, and minimum leaf sizes, and found the classifier performance saturated above 100 trees. Allowing for minimum leaf size of 1 and depth of 50 gave us the best performance (both overall accuracy and average

F1). For MLP, we tried networks of up to 5 layers and up to 1024 neurons per layer and found the classifier performance metrics (both overall accuracy and average F1) come close to the saturation with three layers of 1024 neurons each, having about 2.2 million parameters similar to our ST-LSTM model. For the MLP model, we kept the other hyperparameters such as batch size and network optimization algorithm the same as the values we chose for our final LSTM network. We also applied a dropout value of 0.05 to enhance the classifier generalization performance as we found this value working better than the other values.

The results for the best model in each category of classifiers are reported in Table 4-3 for the 2.4M samples test dataset, which shows considerable increase of up to 5.4% in average F1 and up to 6.3% in overall accuracy for our deep models compared to the Random Forest. Increase in individual class F1 values varied from 1.2% for bare class up to 8.8% for agriculture. The increase in performance can be directly related to exploiting more data dimensions. As discussed previously, the baseline models used only spectral data and texture features at their input and they couldn't use full temporal information.

Table 4-3: Classification accuracy of benchmark and proposed models

Classifier	***Test***	***Avg.***	***Class F1 values***						
	acc.	***F1***	***water***	***developed***	***grass***	***forest***	***bare***	***agri.***	***wetland***
RF(100,50,1)	91.7%	92.9%	97.5%	92.0%	91.3%	89.1%	98.2%	89.5%	92.3%
MLP(3x1024)+Dropout(0.05)	91.2%	92.2%	97.3%	91.8%	90.3%	89.5%	97.8%	88.7%	90.1%
T-LSTM	93.2%	93.9%	97.9%	92.5%	91.6%	94.5%	95.9%	92.7%	92.3%
ST-LSTM	97.1%	97.4%	98.9%	97.1%	96.6%	95.6%	99.0%	97.2%	97.6%
CNN+ST-LSTM	**98.0%**	**98.2%**	**99.3%**	**98.1%**	**97.5%**	**96.9%**	**99.4%**	**98.2%**	**98.3%**
Ref. Matching to NLCD*			95.6%	91.9%*	83.6%*	86.3%	76.7%	92.1%*	86.6%

* We did a comparison of our reference labels with NLCD 2016 labels, which is described in more detail in the text. Note that the definition of NLCD for these land cover classes designated with (*) differs from our definition.

T-LSTM model investigated temporal dimension and gave us one step of enhancement. Although T-LSTM did not have access to GLCM texture information, it performed better than baseline models and it can show that temporal pattern exploitation is more beneficial than texture information. In ST-LSTM model we added selected spatial features and it raised the performance considerably. Finally, CNN+ST-LSTM explored full potential spatial-spectral-temporal analysis by letting network decide on spatial features as well. It is also notable that the improvement came in all metrics of overall accuracy, average F1, and minimum F1. This may credit our initial plan for class distribution in training data, where we intentionally favored more difficult classes.

We also did a comparison of our reference labels to NLCD 2016 land cover. NLCD maps are the product of a classification algorithm and we did not have their reference data. As we explained in section 4.3.1.1, we developed our own reference data by carefully examining each pixel manually and determined our definition of classes (as described in Appendix A), which differ slightly from NLCD classification scheme. These differences result in two set of labels that are not 100% compatible and also some differences arise due to application of administrative data or different class decision making thresholds. However, we still compare NLCD classes to our reference labels by setting up a confusion matrix and reporting the results in Table 4-3 (full confusion matrix is given in Appendix E). The differences between NLCD 2016 and our classification scheme are:

- NLCD considers lawn grasses, city parks, golf courses, and vegetation planted in developed settings as part of its developed class, while we consider it as grass similar to natural settings.
- NLCD considers pasture/hay in the agriculture class, while we may consider it as grassland if there is no evidence of intensive agricultural practices like crops.

- NLCD considers grass and shrub vegetation in two different class groups, while we consider both of them in the same class.
- NLCD makes distinction between low, medium, and high density developed areas. We put all of them under developed title.

Investigating the confusion matrix in appendix E, we found moderate to high confusion in below cases:

- NLCD shrub/scrub with our developed, bare, and to a lesser degree with forest
- NLCD grassland with our agriculture, developed, and to a lesser degree with bare
- NLCD forest classes with our developed, grass and bare
- NLCD Pasture/Hay with our developed and grass
- NLCD agriculture with our developed (high confusion)
- NLCD woody wetland with our forest (high confusion)

The last two cases showed considerable confusion, therefore we dropped them and the reported results at the end of Table 4-3 are without them. Most of the confusion with developed class may come from the fact that NLCD has used roads map and other administrative data for mapping the developed class, while we did not do that and just used remote sensing data. We also consider tree canopy areas inside urban land as forest land cover.

4.4.3. Detailed assessment of final deep model

The selected final deep model is the best model we found in our simulations in terms of average F1 value, which is a CNN+ST-LSTM model with configuration shown in Table 4-2 and overall performance metrics shown in Table 4-3. In this section we look closer at the model performance, both quantitatively and qualitatively.

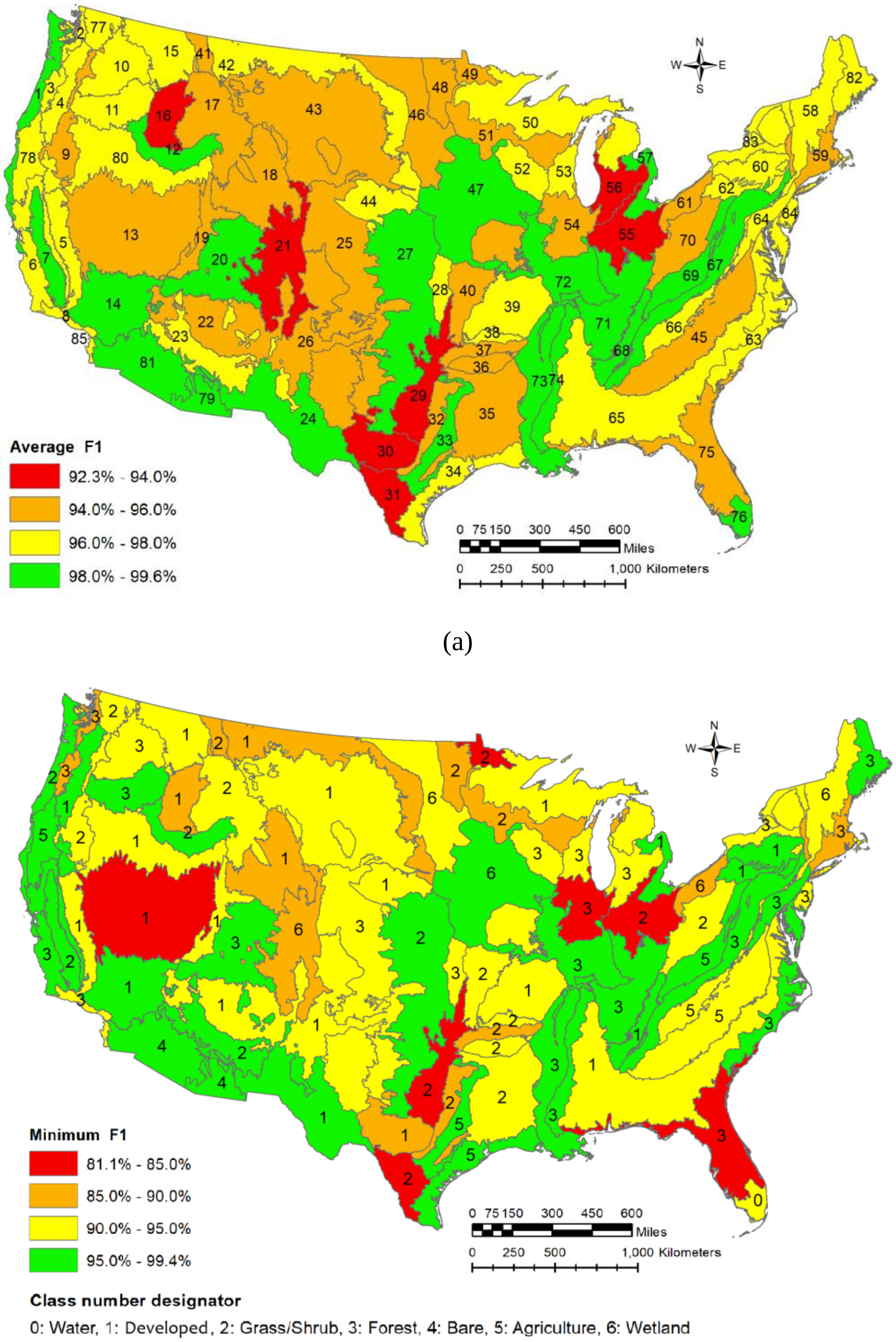
4.4.3.1. Spatial variability of accuracy

Model assessment in section 4.4.2 was based on aggregate test data by combining test data from all blocks. Here we assessed model performance on individual blocks to illustrate its spatial variability. For each block, the overall test accuracy, average F1 over classes, and minimum F1 over classes were calculated and the average value over all blocks is reported in Table 4-4 (we also included the same figures based on MLP model for comparison). One important problem we found when reporting on individual blocks was low F1 performance on rare classes for individual scenes. This is the main tradeoff that exists when mixing local and global targets. We can probably get a better local performance by focusing our training on just one block, but we will lose the general picture and model will not perform well in other local areas. We also dropped classes with less than 500 pixels in any individual block in this section when calculating average or minimum F1 values and considered those cases as not enough samples to extract reliable information for that class.

Table 4-4: Best model and MLP model average and range of performance over all 84 blocks

	Test Accuracy		***Average F1***		***Minimum F1***	
Model	***Average***	***(range)***	***Average***	***(range)***	***Average***	***(range)***
CNN+ST-LSTM	97.6%	(93.5%-99.7%)	96.5%	(92.3%-99.6%)	92.7%	(81.1%-99.4%)
MLP	90.2%	(77.2%-98.8%)	84.7%	(66.7%-98.8%)	70.1%	(28.5%-98.2%)

We also visualized the variation in individual ecoregion classification performance as shown in the maps of Figure 4-3. Note that this visualization extrapolates our results from a single block falling within each ecoregion. Thus, it is not an extensively tested product, however it can guide us roughly on which areas we need additional samples and/or improved algorithms to enhance our classification accuracies.



(a)

(b)

Figure 4-3: Visual representation of a) Average F1, b) Minimum F1 over 84 US ecoregions. Red color shows the worst and green shows the best performance. The numbers shown on the top figure indicates each polygon’s associated ecoregion number, while the numbers shown on the bottom figure indicates the worst class number (in the range of 0-6) by F1 statistics for that ecoregion.

4.4.3.2. Value of multi-sensor Landsat observations

It is also important to know how the selected CNN+ST-LSTM model performs under limited input data due to either sensor availability or missing information (e.g., cloud coverage). This experiment was executed with two main configurations: single-sensor and multi-sensor. For single-sensor case, we removed all observations of the other sensors, fed the model with the single sensor data, and calculated the performance metrics. It is important to note that no retraining took place using the single sensor observations, which could potentially increase classification performance. The results for these simulations are provided in Table 4-5. The range of difference in overall accuracy, average F1, and most of class F1 values is around 1% except wetland class for which this range is 3.6%. Landsat 8 provides the best performance for most of the metrics except F1 for forest and bare classes. Landsat 5 stands next to it by performing better than Landsat 7 except for developed class F1. Note that the input data sequences cover different years from 2005 up to 2019, therefore we did not have all the sensors in all the years but the figures are adjusted to account for this change in reference data.

As the set of pixels used in evaluation are the same for all sensors, we may think of better Landsat 8 performance due to being more recent with more precise and better-quality instruments. But worse performance of Landsat 7 may be a result of its longer mission time, which have increased the number of test sequences substantially and may increase observations variation for the same point due to phenomena such as climate-induced or anthropogenic changes. As for the Landsat 7 SLC failure problem, the invalid data were automatically stripped by Google Earth Engine and they were not included in the downloaded data, so it did not affect our analysis. The last column of the table shows the average number of scenes per year available in each configuration. This figure was made by counting the scenes available in each year in each configuration for each block and averaging it over years and then over all blocks.

Table 4-5: Summary performance statistics for single-sensor simulations

Sensor	*Test acc.*	*Average F1*	*# test sequences*	*# Scenes/year*
Landsat 5	94.2%	94.8%	1,240,935	30
Landsat 7	93.8%	94.1%	2,297,126	34
Landsat 8	94.9%	95.1%	882,002	37

Sensor	*Water_F1*	*Imp_F1*	*Grass_F1*	*Forest_F1*	*Bare_F1*	*Agri_F1*	*Wetland_F1*
Landsat 5	98.5%	95.1%	91.3%	94.6%	98.0%	93.4%	93.1%
Landsat 7	98.1%	95.4%	91.3%	93.4%	97.7%	92.5%	90.6%
Landsat 8	98.6%	96.3%	91.9%	94.4%	97.0%	93.6%	94.3%

For multi-sensor experiments we considered two cases: Fusion of Landsat 5 and 7, or fusion of Landsat 7 and 8. Fusion of Landsat 5 and 8 is impossible due to their non-overlapping mission time. For fusion of Landsat 5/7 we picked those sequences that contain both Landsat 5 and Landsat 7, then generated two other sets of sequences from it by keeping either sole Landsat 5 or sole Landsat 7 observations in each of them. This way, the performance evaluation of individual and fused observations can be directly compared. We followed the same process for Landsat 7/8 assessment. The results are shown in Table 4-6 and Table 4-7, respectively.

The fusion results are very promising and show the power of combining multiple sensors together. Although the difference between individual sensors' performance was mostly about 1%, the fusion of Landsat 5/7 or Landsat 7/8 gives us about 4.5% gain in overall accuracy or average F1 and up to 8.4% gain in class F1 for wetland class. We also see the maximum class F1 improvement in Wetland, agriculture and grass classes and least improvement in bare and water classes, which is good and what we need because bare and water are the easiest and least problematic classes.

Increased performance with multi-sensor data can be attributed to two things: 1) having more observations and more data to do a better prediction, and 2) having our model trained on multi-sensor training data. As not all the sensors are available for all periods of time and our aim was to have our model as general as possible, we opted to train our model on combination of all sensors.

Table 4-6: Summary performance statistics for Landsat 5/7 fusion

Sensor	***Test accuracy***	***Average F1***	***# test sequences***
Landsat 5 (adjusted)	94.2%	94.8%	1,240,603
Landsat 7 (adjusted)	93.5%	93.9%	1,240,603
Landsat 5 and 7	98.0%	98.2%	1,240,603

Sensor	***Water_F1***	***Imp_F1***	***Grass_F1***	***Forest_F1***	***Bare_F1***	***Agri_F1***	***Wetland_F1***
Landsat 5 (adjusted)	98.5%	95.1%	91.3%	94.6%	98.0%	93.4%	93.1%
Landsat 7 (adjusted)	97.8%	95.0%	90.9%	93.5%	97.7%	92.3%	89.9%
Landsat 5 and 7	99.2%	98.1%	97.5%	97.0%	99.4%	98.3%	98.2%

Table 4-7: Summary performance statistics for Landsat 7/8 fusion

Sensor	***Test accuracy***	***Average F1***	***# test sequences***
Landsat 7 (adjusted)	93.5%	93.7%	881,781
Landsat 8 (adjusted)	94.9%	95.1%	881,781
Landsat 7 and 8	98.1%	98.3%	881,781

Sensor	***Water_F1***	***Imp_F1***	***Grass_F1***	***Forest_F1***	***Bare_F1***	***Agri_F1***	***Wetland_F1***
Landsat 7 (adjusted)	98.3%	95.5%	90.5%	92.6%	97.3%	91.5%	90.3%
Landsat 8 (adjusted)	98.6%	96.3%	91.9%	94.4%	97.0%	93.6%	94.3%
Landsat 7 and 8	99.3%	98.3%	97.4%	96.7%	99.4%	98.2%	98.4%

4.4.3.3. Classification accuracy and scene availability

We also looked at the variation of classification performance on annual sequences of different lengths, in essence the number of valid observations per year. The results are shown in Figure

4-4 and Figure 4-5 for different for different combinations of sensors. In each graph, we depicted the distribution of sequence lengths (showing the number of available Landsat scenes for each range of lengths in a histogram) over the entire dataset used, and variation of performance metrics over sequence length. Note that these graphs cannot be compared, because the distribution of points and blocks in each case and in each histogram is different (for example sequences of Landsat 5 and 8 have no time overlap). Also note that we have dropped reporting F1 for classes with less than 50 members in a bin, therefore the average of individual class F1 curves may be truncated. The following remarks could be made:

1- Increasing the sequence length generally enhances the classification performance, which was expected. Combining sensors gives us the better and more stable performance over different sequence lengths. A critical threshold for single-sensor case seems to be around 10-15 observations per year, beyond which the accuracy improvements are phasing out. This problem is less pronounced in multi-sensor configuration and we get more stable performance over all sequence lengths. This better performance in multi-sensor configuration may be a result of training the model with sequences containing multi-sensor input data, which makes it performing better on multi-sensor test sequences as well. Each bin typically contains most of the blocks, but this diversity decreases considerably in the last bins and the sequences in these bins mostly come from a few blocks. This may explain the decrease in performance in Landsat 5 and 7 at the end of their curve, plus some sudden dips for certain curves in the diagrams. However, as noted before, bin-to-bin comparison is not generally valid.

2- Grass and forest classes are the worst from individual class F1 standpoint, which is also confirmed from the full evaluation result for class F1 (table 3). Grass requires longer sequences than any other class to obtain a good F1. It might be because of the high level of confusion between grass and agriculture or bare classes when only a few time stamps of observations are available. Forest performance deteriorates with too many observations per sequence, but we could not identify a clear cause for it.

3- As expected, water has always been easy to predict with any sensor. Developed class performance is also good at all sequence lengths, which is expected because this class typically does not experience temporal changes through time for a stable pixel.

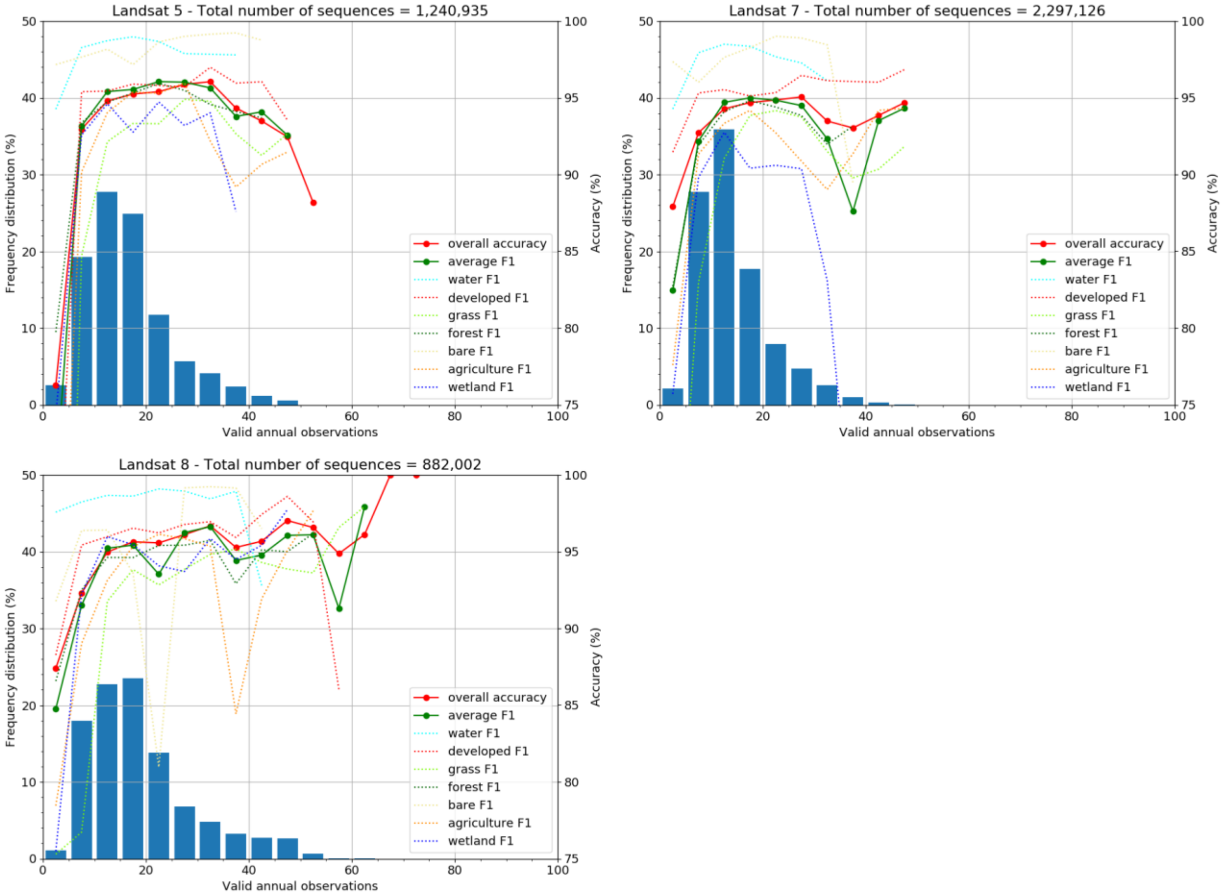


Figure 4-4: Model performance metrics for different length of input sequences with limiting input observations to a single sensor. In each drawing, left axis corresponds to the scene length histogram (blue bars) with the numbers showing the ratio of each bar frequency to the total number of sequences in percentage. Right axis corresponds to performance metrics (detailed in the legend at bottom right of each drawing)

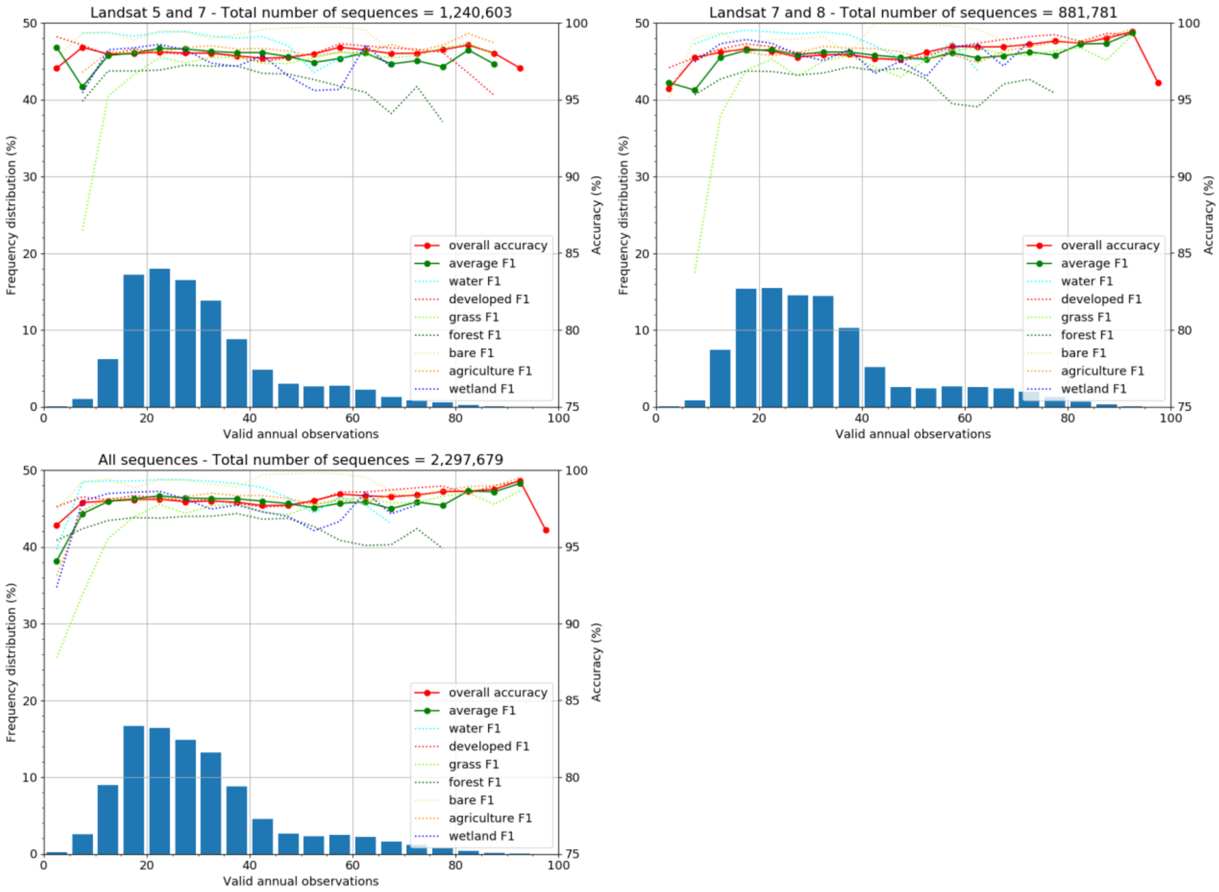


Figure 4-5: Model performance metrics for different length of input sequences by inclusion of multiple sensors. Note that the fusion of Landsat 5/7 and Landsat 7/8 involves the dataset adjusting procedure described before in this section, but the last drawing imposes no limitation and simply considers all of the sequences and draws the histogram and curves.

4.4.4. Visual assessment

We follow the quantitative analysis of our model performance in the previous section with visual inspections. We generated wall-to-wall maps for two sample blocks, here we discuss details on limited areas (we did it for all blocks for year 2015 and the results for some more blocks are provided in appendix F). A wall-to-wall map was generated for a given year, as the model input sequences cover maximum one year of Landsat observations. Also, there is no sampling and all the block points were used for training. As the training data was just samples over each block’s points within full time range of our study period, only a very small portion (on average about 1%) of the above mentioned one-year sequences might be used in the training process and the block map was almost completely model generalization. The first block is the sample used to represent ecoregion 03, located northwest of Portland, OR. This area is a mixture all seven land covers, and the predicted map for full block is shown in Figure 4-6. For this block, the test accuracy was 98.6% and average F1 was 95.2%.

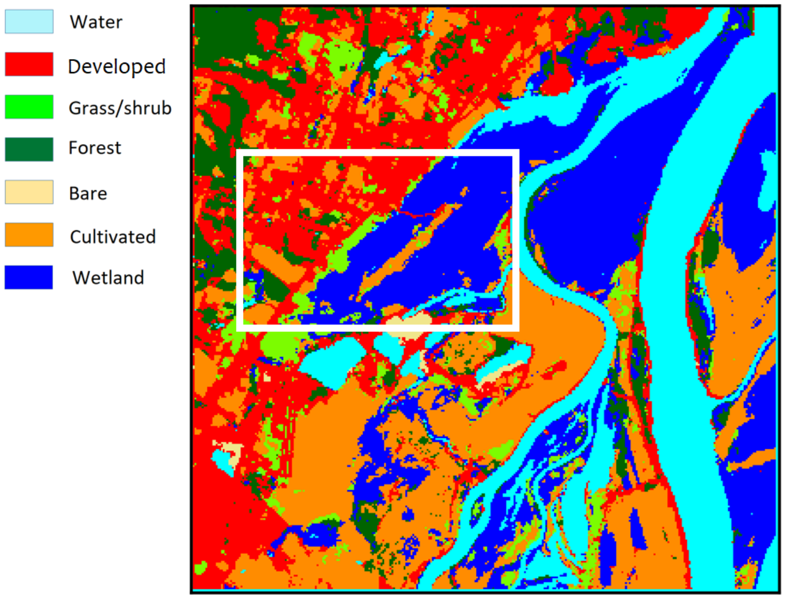


Figure 4-6: Block 03 classified map

We also take a closer look by zooming into the area marked with white rectangle. In Figure 4-7, we provided both high-resolution imagery of this area (provided by Google Earth) and the classified map, overlaid on imagery with some transparency.

Comparing high-resolution imagery and classified map, you can see how different land covers delineated and it is mostly in accordance with the actual image. Although it may not be evident from high-resolution imagery (as it is just a one-time snapshot), the big blue area in the center and right side of the image was a wetland (woody or herbaceous) and we could confirm it by looking at the other images in Google Earth, also by NLCD 2016 land cover map. However, the ‘island’ identified in the center of image (polygon #1) showed some evidence of agricultural activities and our classifier could separate it well from wetland, while NLCD map showed this island as some small, fragmented areas. Evidence of agricultural activities was also the base for assignment of undeveloped land inside or around city to grass or agriculture class (some examples are marked by polygons #2). We also see how the narrow road connecting island to the city has been identified (arrow #3), but there was another narrow road below that one, which is missed by the classifier (arrow #4).

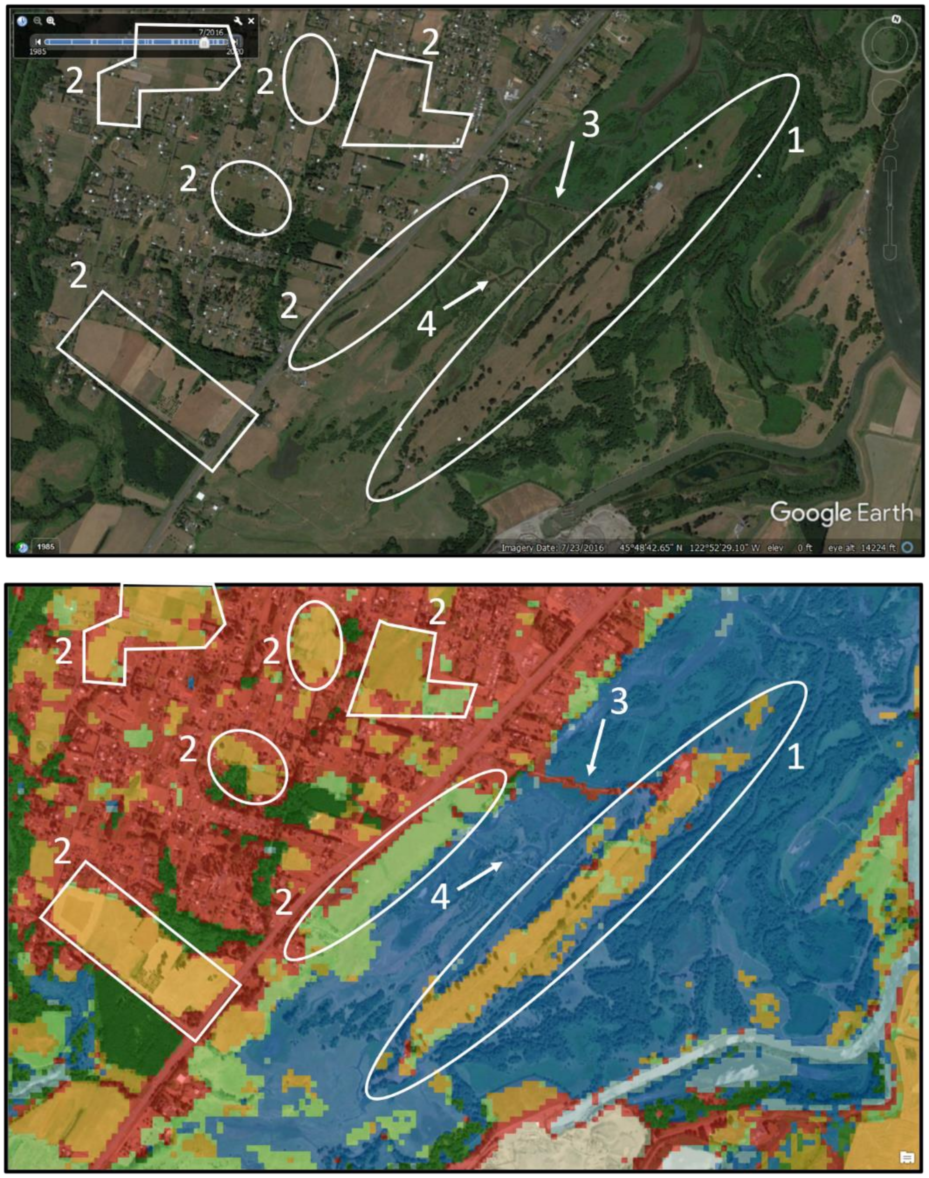

7/2016
1985
2020
2
2
2
2
2
2
3
1
4
Google Earth
1985
Imagery Date: 7/23/2016 45°48'42.65" N 122°52'29.10" W elev 0 ft eye alt 14224 ft
2
2
2
2
2
2
3
1
4

Figure 4-7: Zoomed area in Figure 4-6, high-resolution imagery (top) and classified map (bottom)

Another case that we show in this section is the block representing ecoregion 23, located west of Flagstaff, AZ. The area is home to National Guard camp Navajo, and is dominated by forest and grass land cover but a lot of camp access roads, as shown in Figure 4-8. For this block, the test accuracy was 96.9% and average F1 was 96.6%. We also took a closer look at the marked area in this figure with a snapshot of its high-resolution imagery via Google Earth, as shown in Figure 4-9. Here we have the roads network extracted very well, without any external information. We can also see in Figure 4-9 that the forest and grass/low vegetation areas are accurately delineated in the area marked with #1 (mostly forested), area marked #2 (mostly grass or very low vegetation), and the two areas marked # 3 (mixed forest/grass).

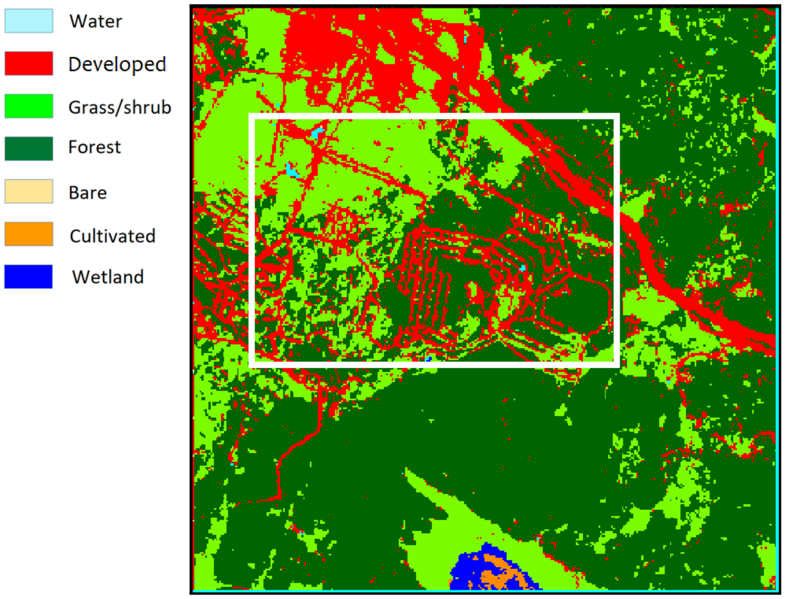


Figure 4-8: Block 23 classified map

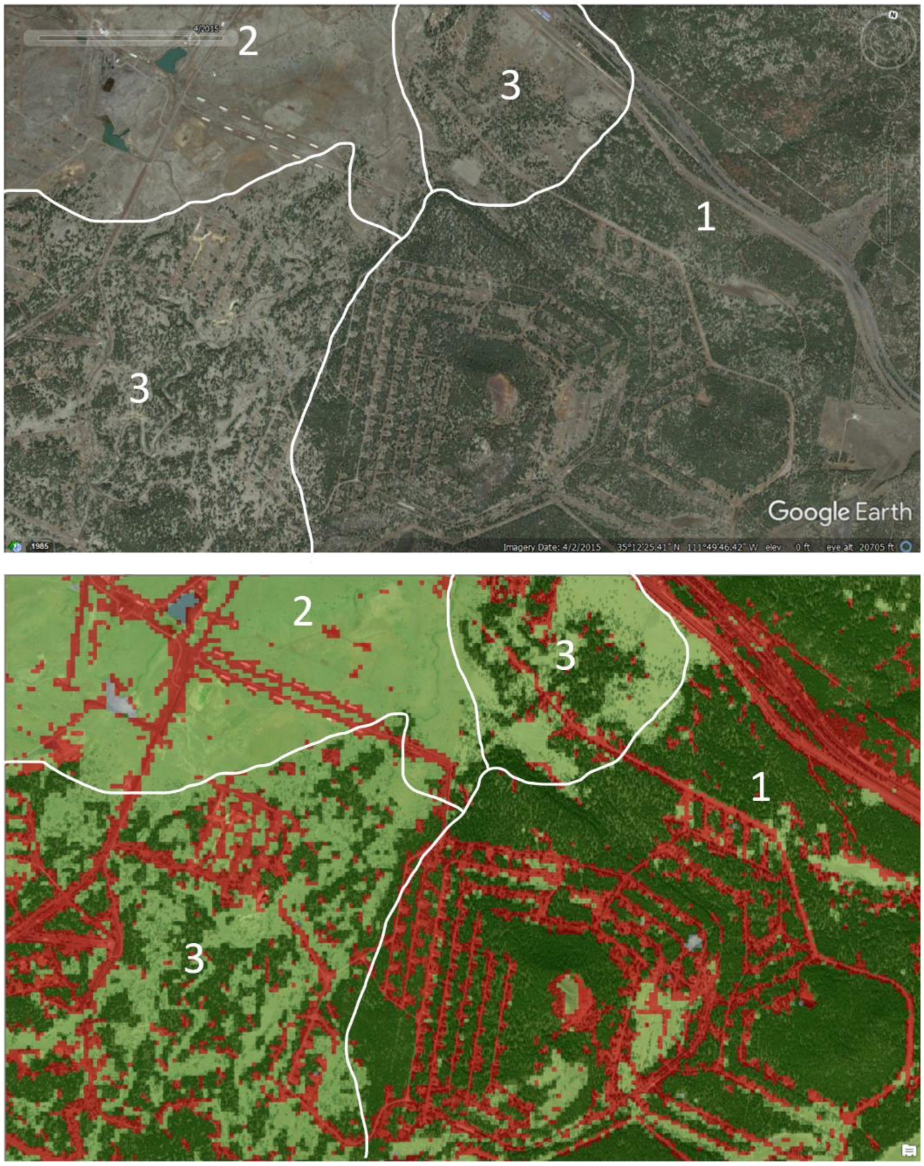

2
3
1
3
Google Earth
Imagery Date: 4/2/2015 35°12'25.41" N 111°49'46.42" W elev 0 ft eye alt 20705 ft
2
3
1
3

Figure 4-9: Zoomed area in Figure 4-8, high-resolution imagery (top) and classified map (bottom)

4.4.5. Ensemble of models

Ensembling is a widely used method of improving model performance in machine learning and is based on combination of a set of trained models to get their collective best vote, which should be more accurate than any one of them (Random Forest classifier is a well-known example of applying the ensembling technique). As our training procedure included a wide range of the network parameters, we had many additional models at the end of this procedure which were similar to the selected model in performance but with minor differences. Therefore, it was worth looking at their ensemble.

To do this, we created a pool of 40 best models from our latest simulations. The best selected single model was the one that we introduced before, which gave us the best average F1 performance over the whole evaluation dataset. Then in an iterative loop, we identified the next best model to add to it as the next ensemble member to achieve the best collective performance on average F1 metric, and continued it up to an ensemble of 10 models. Table 4-8 shows the performance statistics for ensembles of 2 to 10 models, generated by testing the models on the 2.4M test dataset that has been used in the prior single-model experiments in Table 4-3.

The collective voting can simply be based on majority of classes that are predicted per pixel by participating models (i.e., hard-decision making). But we chose a soft-decision approach and selected the class that provides the greatest average softmax value over participating models for each pixel. Our analysis showed that a soft-decision approach gives better performance than hard-decision making.

Table 4-8: Performance statistics for ensembles of two to ten CNN+ST-LSTM models. Last two columns are the difference between Avg F1 or Min F1 of each row and the original CNN+ST-LSTM

# of models	*Accuracy*	*Average F1*	*Minimum F1*
CNN+ST-LSTM	98.0%	98.2%	96.9%
2	98.4%	98.6%	97.4%
3	98.5%	98.6%	97.5%
4	98.5%	98.7%	97.6%
5	98.6%	98.7%	97.7%
6	98.6%	98.7%	97.7%
7	98.6%	98.8%	97.7%
8	98.6%	98.8%	97.7%
9	98.6%	98.8%	97.8%
10	98.6%	98.8%	97.8%

As the result shows us, the overall performance improvement is very small as the model performance is already very high. One interesting finding was that having the first best model (the model used in the previous section) as the best model when measured by average F1, the second-best model (used for ensemble of two models) was the model with the best minimum F1. It makes sense that these two be the best first and second models. There was no apparent specification that we could attach to the next models.

It would be also interesting to look at the performance of ensemble models on individual blocks quantitatively and visually. We calculated the achieved gain in average and minimum F1 by ensembling of 2, 3, 4, 5, and 10 models and the results are shown in Figure 4-10 and Figure 4-11. In these figures, the horizontal axis shows base model performance (average F1 or minimum F1) and the vertical axis measures ensembling difference compared to it. Each block in each ensemble is depicted by a symbol and the regression line for each ensemble is also drawn. Both diagrams show a similar trend and the most important feature is that due to negative slope of regression line, the ensembling gain is more pronounced for blocks where the single model is not performing very well. This is good news, as the ensemble gives us better gain where we need it the most.

As seen in Figure 4-10, the maximum ensembling gain is about 2.5% for ensemble of 10 models but it can also be as high as about 2% for just two models. Therefore, the improvement on average F1 is marginal. The calculated R^2 value for the ensemble regression lines in figure 10 is between 0.6 (for 2 model) to 0.75 (for 10 models), which is not very high but shows moderate linear relationship. For minimum F1 as you see in Figure 4-11, we have the maximum ensembling gain about 6.5%, achieved by ensemble of 4, 5, or 10 models. In this case the two-model ensemble will give us improvement less than 3.5%. The regression line, however, is almost identical for ensemble of 5 and 10 models, suggesting there is no benefit for going further than 5 models. The calculated R^2 value for the ensemble regression lines in Figure 4-11 are similar to Figure 4-10, ranging between 0.57 (for 2 model) to 0.70 (for 4 models).

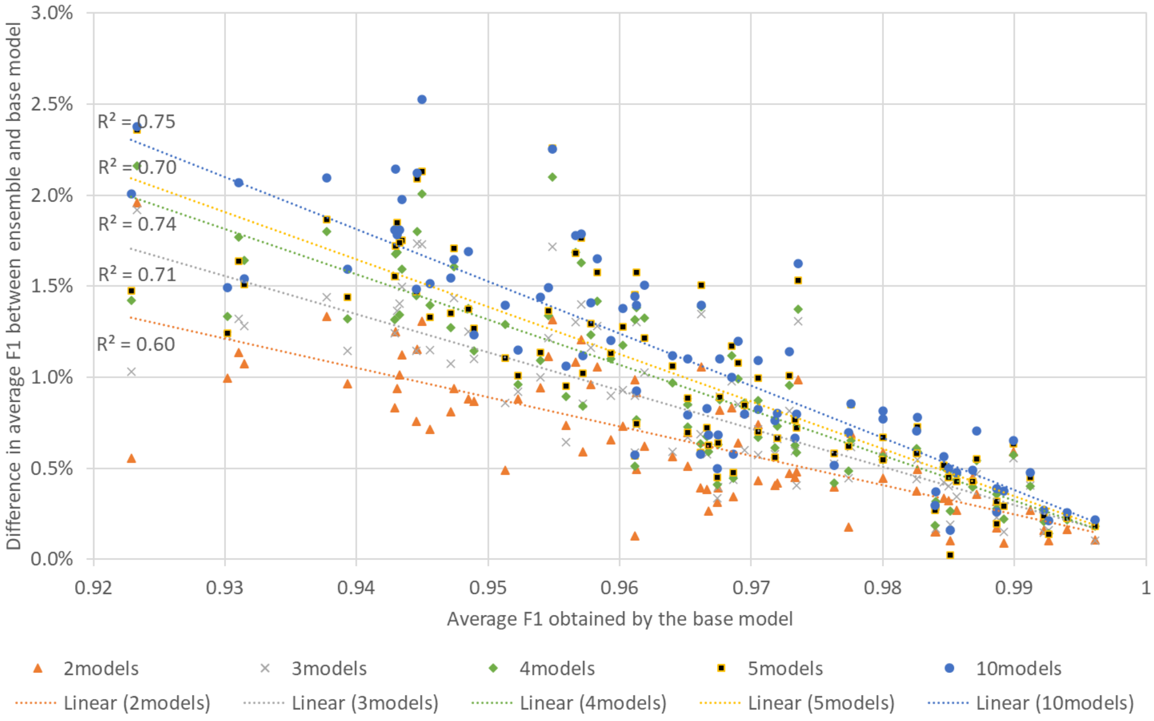


Figure 4-10: Comparison of ensemble of 2, 3, 5, and 10 models to the base model for individual 84 blocks on average F1

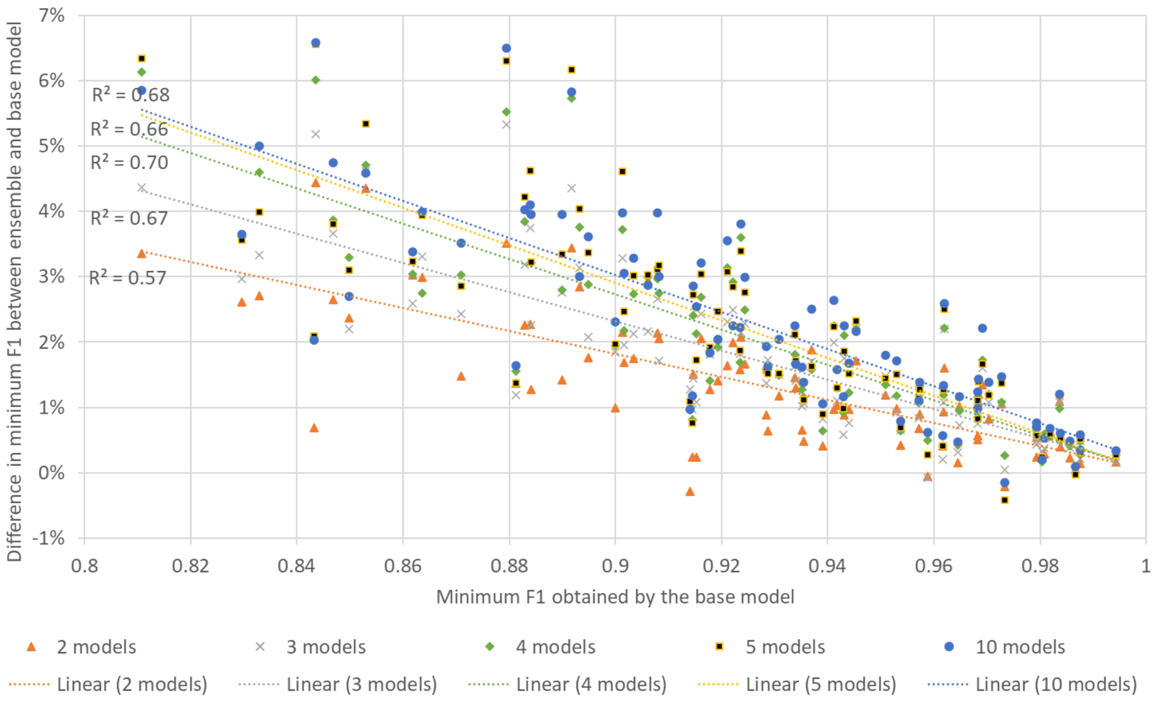


Figure 4-11: Comparison of ensemble of 2, 3, 5, and 10 models to the base model for individual 84 blocks on minimum F1

Note that the above improvement in block classification accuracy may not be evident from visual examination as the improvement is marginal. We have provided an example of predicted ensemble map in appendix G using 1, 2, and 3 models.

4.5. Conclusion

In this paper we applied state-of-the-art deep learning methods to global land cover classification and presented the steps that we designed and implemented for this purpose. In our approach we used LSTM-type recurrent networks complemented with additional multilayer perceptron and convolutional networks. The model is fed and trained with rich spectral-spatial-temporal features by combining different globally free available medium-resolution datasets. To the best of our knowledge, this is the first application of recurrent deep learning methodology in large-area land cover classification, which was typically done by conventional classifiers and with additional input data (e.g., road vectors). We tested three different network configurations and tuned many hyperparameters and feature combinations through numerous simulations and achieved outstanding performance and very high accuracy of above 98% (for both overall accuracy and average F1) for our most complete convolutional spatio-temporal LSTM model based on a selected number of input features. These values were limited to around 93% (overall accuracy) and 94% (average F1) when not employing spatial features (temporal LSTM model). It was also around 92% (overall accuracy) and 93% (average F1) with the best random forest classifier in our tests even though we used all of the spectral and spatial features as their input. We also showed that although providing more time stamps is beneficial, the best model will maintain a stable performance with only about 10 time stamps per year. Although the limited resolution of Landsat data to a medium level of 30 meters hides fine details from the observing model, we showed that our model can employ temporal profile of data to distinguish similar land covers such as grassland and cultivation. As another enhancement, ensembling some of the best achieved models increased the performance statistics, particularly minimum F1, substantially for some blocks.

The wide geographical coverage of our sample data is another strength of our study and provided us with good insight on the pros and cons of our model. As we discussed, there is a tradeoff between local and global performance and we were more interested in having a global model, only based on remote sensing data and no auxiliary administrative information and therefore easily deployable in any other location in USA or abroad.

There are still some limitations and caveats in the model, and we have ideas to enhance the work. For example, we need to assess the model performance by a complete random sampling over whole CONUS, and to evaluate more blocks in each ecoregion to identify local performance issues. The class imbalance and its effect on the model performance is also a well-known issue that is inevitable unless some special measures are taken to combat it. There are well-known approaches such as increasing the weight of rare classes in the classifier optimization algorithm or generating synthetic data to increase rare class presence. Preliminary efforts applying these ideas did not enhance the performance in our experiments, but this remains an open area for further exploration.

There are some immediate next-step enhancements to the model architecture as well. One of the most useful enhancements – especially for reducing the required computing budget – is to transform the model processing to a fully convolutional architecture. The current problem is that the reference data is not available for all pixels in any block, therefore there are a lot of "invalid" pixels in the reference map that makes fully convolutional processing problematic. Another drawback of the current patchy reference data and the subsequent pixel-based classification is that we cannot perform any object-based analysis or employ segmentation techniques to make the output map smoother and less noisy. As the model is already a very high performance one with accuracy of over 98% on aggregate dataset, modifying its architecture to use proprietary modules, optimizers, or layer connections does not seem to be a vital need. However, improving the local block-level performance is a major issue that needs further investigation. Migrating to finer resolution data such as Sentinel-2 may help, and as it is available on the same platform as we used for this work, the transition can be seamless. And last but not least, adding some pre-processing steps such as spectral unmixing might be helpful to reduce class confusion.

Acknowledgements

This work was supported by the NASA's Land Cover Land Use Change Program (grant # NNX15AD42G) and NASA High-End Computing facility at NASA Ames Research Center, CA.

We are grateful for excellent technical and administrative NASA support, in particular the assistance from Blaise Hartman and Samson Cheung among others. We also appreciate the efforts from Kristi Sayler and Mark Drummond (USGS) for providing the Trends data. We thank our reference data editor team of current and former SUNY ESF students: Atef Alla Eddin Amriche, Harrison Goldspiel , Madusha Sammani, Megan Kate Medwid, Rabia Munsaf Khan, and Zhixin Wang.

References

Abdi, Abdulhakim Mohamed. 2020. “Land Cover and Land Use Classification Performance of Machine Learning Algorithms in a Boreal Landscape Using Sentinel-2 Data.” *GIScience & Remote Sensing* 57 (1): 1–20. https://doi.org/10.1080/15481603.2019.1650447.

Alhassan, Victor, Christopher Henry, Sheela Ramanna, and Christopher Storie. 2020. “A Deep Learning Framework for Land-Use/Land-Cover Mapping and Analysis Using Multispectral Satellite Imagery.” *Neural Computing and Applications* 32 (12): 8529–44. https://doi.org/abdi.

Brown, Jesslyn F., Heather J. Tollerud, Christopher P. Barber, Qiang Zhou, John L. Dwyer, James E. Vogelmann, Thomas R. Loveland, et al. 2020. “Lessons Learned Implementing an Operational Continuous United States National Land Change Monitoring Capability: The Land Change Monitoring, Assessment, and Projection (LCMAP) Approach.” *Remote Sensing of Environment* 238 (March): 111356. https://doi.org/10.1016/j.rse.2019.111356.

Campos-Taberner, Manuel, Francisco Javier García-Haro, Beatriz Martínez, Emma Izquierdo-Verdiguier, Clement Atzberger, Gustau Camps-Valls, and María Amparo Gilabert. 2020. “Understanding Deep Learning in Land Use Classification Based on Sentinel-2 Time Series.” *Scientific Reports* 10 (1): 17188. https://doi.org/10.1038/s41598-020-74215-5.

Chang, Tony, Brandon Rasmussen, Brett Dickson, and Luke Zachmann. 2019. “Chimera: A Multi-Task Recurrent Convolutional Neural Network for Forest Classification and Structural Estimation.” *Remote Sensing* 11 (7): 768. https://doi.org/10.3390/rs11070768.

Chen, Jun, Jin Chen, Anping Liao, Xin Cao, Lijun Chen, Xuehong Chen, Chaoying He, et al. 2015. “Global Land Cover Mapping at 30m Resolution: A POK-Based Operational Approach.” *ISPRS Journal of Photogrammetry and Remote Sensing* 103 (May): 7–27. https://doi.org/10.1016/j.isprsjprs.2014.09.002.

Feng, Min, and Yan Bai. 2019. “A Global Land Cover Map Produced through Integrating Multi-Source Datasets.” *Big Earth Data* 3 (3): 191–219. https://doi.org/10.1080/20964471.2019.1663627.

Feng, Quanlong, Jianyu Yang, Dehai Zhu, Jiantao Liu, Hao Guo, Batsaikhan Bayartungalag, and Baoguo Li. 2019. “Integrating Multitemporal Sentinel-1/2 Data for Coastal Land Cover Classification Using a Multibranch Convolutional Neural Network: A Case of the Yellow River Delta.” *Remote Sensing* 11 (9): 1006. https://doi.org/10.3390/rs11091006.

Fry, Joyce, George Z. Xian, Suming Jin, Jon Dewitz, Collin G. Homer, Limin Yang, Christopher A. Barnes, N.D. Herold, and J.D. Wickham. 2011. “Completion of the 2006 National Land Cover Database for the Conterminous United States.” *Photogrammetric Engineering and Remote Sensing* 77 (9): 858–64.

Giri, Chandra, and Jordan Long. 2014. “Land Cover Characterization and Mapping of South America for the Year 2010 Using Landsat 30 m Satellite Data.” *Remote Sensing* 6 (10): 9494–9510. https://doi.org/10.3390/rs6109494.

Gong, Peng, Han Liu, Meinan Zhang, Congcong Li, Jie Wang, Huabing Huang, Nicholas Clinton, et al. 2019. “Stable Classification with Limited Sample: Transferring a 30-m Resolution Sample Set Collected in 2015 to Mapping 10-m Resolution Global Land Cover in 2017.” *Science Bulletin* 64 (6): 370–73. https://doi.org/10.1016/j.scib.2019.03.002.

Grekousis, George, Giorgos Mountrakis, and Marinos Kavouras. 2015. “An Overview of 21 Global and 43 Regional Land-Cover Mapping Products.” *International Journal of Remote Sensing* 36 (21): 5309–35. https://doi.org/10.1080/01431161.2015.1093195.

Hall-Beyer, Mryka. 2017a. “GLCM Texture: A Tutorial v. 3.0 March 2017,” March. https://doi.org/10.11575/PRISM/33280.

———. 2017b. “Practical Guidelines for Choosing GLCM Textures to Use in Landscape Classification Tasks over a Range of Moderate Spatial Scales.” *International Journal of Remote Sensing* 38 (5): 1312–38. https://doi.org/10.1080/01431161.2016.1278314.

Homer, Collin, Jon Dewitz, Joyce Fry, Michael Coan, Nazmul Hossain, Charles Larson, Alexa Mckerrow, J. VanDriel, and James Wickham. 2007. “Completion of the 2001 National Land Cover Database for the Conterminous United States.” *Photogrammetric Engineering and Remote Sensing* 73 (April).

Hu, LuanYun, YanLei Chen, Yue Xu, YuanYuan Zhao, Le Yu, Jie Wang, and Peng Gong. 2014. “A 30 Meter Land Cover Mapping of China with an Efficient Clustering Algorithm CBEST.” *Science China Earth Sciences* 57 (10): 2293–2304. https://doi.org/10.1007/s11430-014-4917-1.

Ienco, Dino, Raffaele Gaetano, Claire Dupaquier, and Pierre Maurel. 2017. “Land Cover Classification via Multitemporal Spatial Data by Deep Recurrent Neural Networks.” *IEEE Geoscience and Remote Sensing Letters* 14 (10): 1685–89. https://doi.org/10.1109/LGRS.2017.2728698.

Interdonato, Roberto, Dino Ienco, Raffaele Gaetano, and Kenji Ose. 2018. “DuPLO: A DUal View Point Deep Learning Architecture for Time Series ClassificatiOn.” *ArXiv:1809.07589 [Cs]*, September. http://arxiv.org/abs/1809.07589.

Iyer, M.S., and R.R. Rhinehart. 1999. “A Method to Determine the Required Number of Neural-Network Training Repetitions.” *IEEE Transactions on Neural Networks* 10 (2): 427–32. https://doi.org/10.1109/72.750573.

Jia, Xiaowei, Mengdie Wang, Ankush Khandelwal, Anuj Karpatne, and Vipin Kumar. 2019. “Recurrent Generative Networks for Multi-Resolution Satellite Data: An Application in Cropland Monitoring.” In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2628–34. Macao, China: International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2019/365.

Kussul, Nataliia, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. 2017. “Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data.” *IEEE Geoscience and Remote Sensing Letters* 14 (5): 778–82. https://doi.org/10.1109/LGRS.2017.2681128.

"Land Cover Trends Dataset, 1973-2000." 2014. Data Series. Data Series. United States Geological Survey.

Liao, Chunhua, Jinfei Wang, Qinghua Xie, Ayman Al Baz, Xiaodong Huang, Jiali Shang, and Yongjun He. 2020. "Synergistic Use of Multi-Temporal RADARSAT-2 and VENµS Data for Crop Classification Based on 1D Convolutional Neural Network." *Remote Sensing* 12 (5): 832. https://doi.org/10.3390/rs12050832.

Lyu, Haobo, Hui Lu, and Lichao Mou. 2016. "Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection." *Remote Sensing* 8 (6): 506. https://doi.org/10.3390/rs8060506.

Mazzia, Vittorio, Aleem Khaliq, and Marcello Chiaberge. 2019. "Improvement in Land Cover and Crop Classification Based on Temporal Features Learning from Sentinel-2 Data Using Recurrent-Convolutional Neural Network (R-CNN)." *Applied Sciences* 10 (1): 238. https://doi.org/10.3390/app10010238.

Mountrakis, Giorgos, Raymond Watts, Lori Luo, and Jida Wang. 2009. "Developing Collaborative Classifiers Using an Expert-Based Model." *Photogrammetric Engineering & Remote Sensing* 75 (7): 831–43. https://doi.org/10.14358/PERS.75.7.831.

Ojala, T., M. Pietikainen, and T. Maenpaa. 2002. "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (7): 971–87. https://doi.org/10.1109/TPAMI.2002.1017623.

Pelletier, Charlotte, Geoffrey Webb, and François Petitjean. 2019. "Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series." *Remote Sensing* 11 (5): 523. https://doi.org/10.3390/rs11050523.

Pérez-Hoyos, Ana, Felix Rembold, Hervé Kerdiles, and Javier Gallego. 2017. "Comparison of Global Land Cover Datasets for Cropland Monitoring." *Remote Sensing* 9 (11): 1118. https://doi.org/10.3390/rs9111118.

Phiri, Darius, and Justin Morgenroth. 2017. "Developments in Landsat Land Cover Classification Methods: A Review." *Remote Sensing* 9 (9): 967. https://doi.org/10.3390/rs9090967.

Rußwurm, M., and M. Körner. 2017. "Multi-Temporal Land Cover Classification With Long Short-Term Memory Neural Networks." *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-1/W1 (May): 551–58. https://doi.org/10.5194/isprs-archives-XLII-1-W1-551-2017.

Rußwurm, Marc, and Marco Körner. 2018. "Multi-Temporal Land Cover Classification with Sequential Recurrent Encoders." *ArXiv:1802.02080 [Cs]*, February. http://arxiv.org/abs/1802.02080.

Salehinejad, Hojjat, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. 2017. "Recent Advances in Recurrent Neural Networks," December.

Sharma, Atharva, Xiuwen Liu, and Xiaojun Yang. 2018. "Land Cover Classification from Multi-Temporal, Multi-Spectral Remotely Sensed Imagery Using Patch-Based Recurrent Neural Networks." *Neural Networks* 105 (September): 346–55. https://doi.org/10.1016/j.neunet.2018.05.019.

Shendryk, Yuri, Yannik Rist, Catherine Ticehurst, and Peter Thorburn. 2019. "Deep Learning for Multi-Modal Classification of Cloud, Shadow and Land Cover Scenes in PlanetScope and Sentinel-2 Imagery." *ISPRS Journal of Photogrammetry and Remote Sensing* 157 (November): 124–36. https://doi.org/10.1016/j.isprsjprs.2019.08.018.

Sun, Ziheng, Liping Di, and Hui Fang. 2019. “Using Long Short-Term Memory Recurrent Neural Network in Land Cover Classification on Landsat and Cropland Data Layer Time Series.” *International Journal of Remote Sensing* 40 (2): 593–614. https://doi.org/10.1080/01431161.2018.1516313.

Verma, Deepank, and Arnab Jana. 2020. “LULC Classification Methodology Based on Simple Convolutional Neural Network to Map Complex Urban Forms at Finer Scale: Evidence from Mumbai.” *ArXiv:1909.09774 [Cs]*, May. http://arxiv.org/abs/1909.09774.

Vogelmann, Jim, S. Howard, Limin Yang, C. Larson, Bruce Wylie, and N Driel. 2001. “Completion of the 1990s National Land Cover Data Set for the Conterminous United States From LandSat Thematic Mapper Data and Ancillary Data Sources.” *Photogrammetric Engineering and Remote Sensing* 67 (June): 650–55. https://doi.org/10.1007/978-94-011-4976-1_32.

Wang, J., X. Li, S. Zhou, and J. Tang. 2017. “Landcover Classification Using Deep Fully Convolutional Neural Networks.” In *AGU Fall Meeting Abstracts*, 2017:IN11E-02.

Yang, Hui, Songnian Li, Jun Chen, Xiaolu Zhang, and Shishuo Xu. 2017. “The Standardization and Harmonization of Land Cover Classification Systems towards Harmonized Datasets: A Review.” *ISPRS International Journal of Geo-Information* 6 (5): 154. https://doi.org/10.3390/ijgi6050154.

Yu, Le, Jie Wang, and Peng Gong. 2013. “Improving 30 m Global Land-Cover Map FROM-GLC with Time Series MODIS and Auxiliary Data Sets: A Segmentation-Based Approach.” *International Journal of Remote Sensing* 34 (16): 5851–67. https://doi.org/10.1080/01431161.2013.798055.

Yu, Le, Jie Wang, XueCao Li, CongCong Li, YuanYuan Zhao, and Peng Gong. 2014. “A Multi-Resolution Global Land Cover Dataset through Multisource Data Aggregation.” *Science China Earth Sciences* 57 (10): 2317–29. https://doi.org/10.1007/s11430-014-4919-z.

Zhao, Hongwei, Zhongxin Chen, Hao Jiang, Wenlong Jing, Liang Sun, and Min Feng. 2019. “Evaluation of Three Deep Learning Models for Early Crop Classification Using Sentinel-1A Imagery Time Series—A Case Study in Zhanjiang, China.” *Remote Sensing* 11 (22): 2673. https://doi.org/10.3390/rs11222673.

Zhao, Xuemei, Lianru Gao, Zhengchao Chen, Bing Zhang, and Wenzhi Liao. 2019. “Large-Scale Landsat Image Classification Based on Deep Learning Methods.” *APSIPA Transactions on Signal and Information Processing* 8: e26. https://doi.org/10.1017/ATSIP.2019.18.

Appendices

Appendix A: Land cover labels inspection and editing procedure and rules

The land cover class definition and general rules governing the class assignment are summarized in table A-1. Although we did our best to minimize mistakes and confusion, subjective decision making is inevitable in such a process and it will make the dataset not 100% error-free (for example judgement on majority rule or distinguishing farmland from pasture by eye). General considerations are:

- Only pixels that could be confidently labeled from high resolution Google Earth imagery were selected.
- The labeling process first prioritized the developed class. This approach was taken because of the importance of the effects of anthropogenic changes to the environment, and it is also the approach taken in NLCD class specification. If not labeled developed, the cropland class and water continuity (i.e., rivers) were considered. Otherwise, the class was assigned based on the majority class of the pixel area.
- For a pixel to be assigned to a given class it had to remain unchanged for the 2005-2015 period. For developed, water, and forest classes this temporal consistency was easily enforced as exceptions were rare (e.g., a forest is clear cut, a water body dried up, a building was abandoned). For grass, barren, cropland and wetland temporal consistency was challenging due to the expected within class dynamics (e.g., expanding/shrinking wetland, postponing cultivation for some period of time, drought conditions). For these cases, a class

was assigned when it was present for 2/3 of the available observations through the 2005-2015 period.[3]

- Class expected dynamics such as leaf dropping, grow/decline of grassy areas in prairies and agriculture were considered natural changes keeping the class label unchanged. However, other unexpected changes, for example clear cutting within the above time period, disqualified pixels from inclusion.
- Wetland is by far the trickiest class because technically we should know the soil moisture to label it correctly, which we cannot measure from the aerial optical natural color imagery. So, for this class, reference assignment was done very conservatively and almost always only the inner parts of the area designated in USGS trends map as wetland was preserved and the rest was removed.

[3] The reason to keep land cover unchanged over many years was mostly to have a good temporal diversity and be able to employ all three Landsat satellites (5/7/8). If we had chosen a shorter period such as 2008-2012, we might have more pixels included but less temporal diversity. Actually we didn't even limit ourselves in practice to 2015 and included more recent years if we were sure of land cover stability.

Table A-1: Land cover class definitions and class-specific considerations in reference assignment

Water **(class 0)**	*Ponds, lakes, rivers, oceans that are persistently filled by water.* For lakes, ponds, and oceans, only the pixels demonstrating a persistent water presence were included. For streams/rivers, a pixel was assigned to the water class when deemed necessary to preserve the spatial continuity (i.e., avoid river breaks), even if the water occupied less pixel area than other classes). Water presence was required to be persistent through time, therefore seasonal water presence (e.g., seasonal streams, limited flooding) did not qualify a pixel for the water class. Presence of algae on the water surface did not disqualify a pixel from water class assignment.
Developed **(class 1)**	*Built-up area including houses, factories, barns, parking lots, roads (paved or dirt), railroads.* The developed class was prioritized over all other classes. When 20% or higher of the pixel area was deemed developed the pixel was assigned to the developed class. Road pixels, even when occupying less than 20% of pixel area were still assigned as developed when it was necessary to preserve their spatial continuity. Dirt roads were considered falling in the developed class, but irregular/temporary tracks and trails/footpaths were not.
Grass/Shrub **(class 2)**	*Low vegetation that is not cultivated, including natural patches, pasture and grazing land, and man-made patches.* Man-made patches including yard lawns, city parks, golf courses, and soccer fields were assigned to this class. Pasture and Grazing lands that are not intensely cultivated were also included here.
Forest **(class 3)**	*Tall vegetation (taller than typical grass/shrubs) that is not intensely cultivated.* All tree types, including forest plantations, were assigned as forest. Tree orchards were not assigned as forest.
Bare **(class 4)**	*Soil, rocks, mining land, or land with very limited vegetation.* If vegetation was identified as majority for a pixel area even for a short time period it was labeled as grassland and not barren. Sand dunes and dry sandy areas were assigned in the barren land. This class includes Barren, mechanically disturbed, and nonmechanically disturbed classes of the original 11-level Anderson classification scheme.
Agriculture **(class 5)**	*Cultivated areas demonstrating distinct agricultural parcel shapes and tilling lines, including orchards and vineyards.* For designation as cropland, these characteristics were sought for at least 20% of the pixel area: 1) row pattern of tilling/cultivation, 2) temporal high contrast color transition from green to yellow, 3) regular rectangular shape with clear farm edges.
Wetland **(class 6)**	*A typically vegetated area that is periodically saturated or covered with water.* For designation as a wetland (that may have low plants or high trees mixed with water) water should be present mixed with vegetation most of the time. There should be no clear water boundary as the boundary may change every year (unlike a lake or a pond). For woody wetlands, where water was difficult to identify under thick canopy, wetlands were assigned when high vegetation turnover was present. Examination of these challenging pixels during winter months was a critical decision component. Muddy, vegetation-free areas in lake borderlines or seashores were assigned as wetland, not barren land.
Ice/Snow	*Permanent coverage by ice/snow. This class was not present in our dataset.*

Appendix B: General workflow and reference maps generation procedure

General steps in dataset and simulation tasks in our work is depicted in figure B-1. The inspection and editing of land cover labels were a very labor-intensive work that was done by a team and the work was checked by different people in various stages to minimize the human error. We used Google Earth high-resolution imagery to review pixels by executing below steps for each block:

1. Converting the original USGS land cover map to several layers per class and creating a 2D mesh over the area to designate 30mx30m squares.
2. Loading the above set of layers on Google Earth and enable its historical imagery.
3. Sliding time back and forth and decide on each square class correctness and keep it (if we were certain) or drop it (if it was incorrect or we were uncertain).
4. We also assign a time tag to each pixel to designate the time period in which the pixel had its stable land cover. This feature is necessary because we do not have full 2005-2015 high-resolution imagery for every block available on Google Earth.
5. Each block is edited by one person and then reviewed by another person and the final inspector.
6. The final layers are merged to make the final corrected map and saved in the repository for the next stage (sampling and building training dataset).

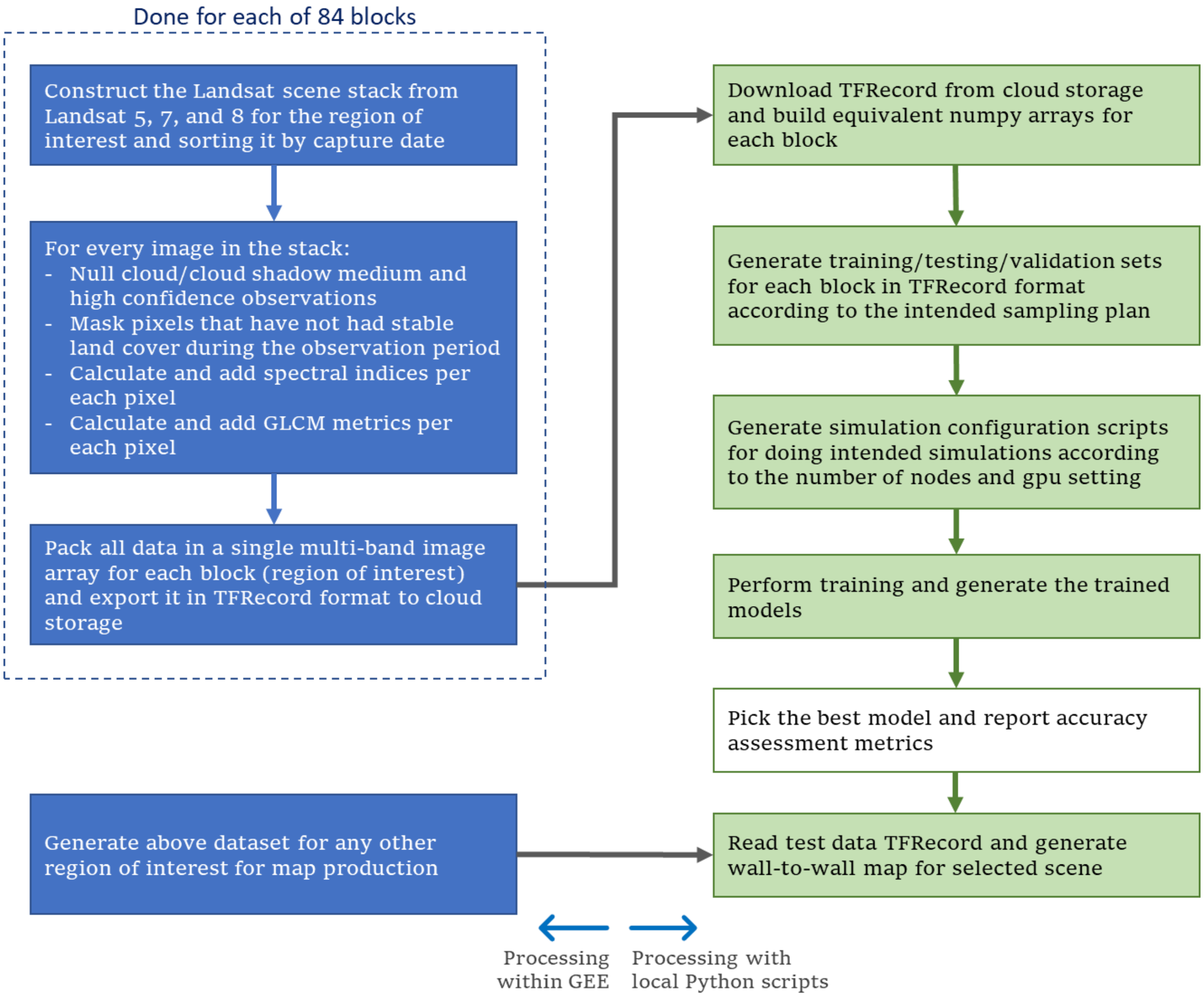

Done for each of 84 blocks
Construct the Landsat scene stack from Landsat 5, 7, and 8 for the region of interest and sorting it by capture date
For every image in the stack:
- Null cloud/cloud shadow medium and high confidence observations
- Mask pixels that have not had stable land cover during the observation period
- Calculate and add spectral indices per each pixel
- Calculate and add GLCM metrics per each pixel
Pack all data in a single multi-band image array for each block (region of interest) and export it in TFRecord format to cloud storage
Generate above dataset for any other region of interest for map production
Download TFRecord from cloud storage and build equivalent numpy arrays for each block
Generate training/testing/validation sets for each block in TFRecord format according to the intended sampling plan
Generate simulation configuration scripts for doing intended simulations according to the number of nodes and gpu setting
Perform training and generate the trained models
Pick the best model and report accuracy assessment metrics
Read test data TFRecord and generate wall-to-wall map for selected scene
Processing within GEE
Processing with local Python scripts

Figure B-1: Flow diagram of the processing steps in data generation and simulation/testing in our project

Appendix C: Details of land cover class distribution for each of 84 selected blocks

For each sample block we chose the pixels that we confirmed as stable for our reference labels. Based on that, we picked all available developed, bare, and wetland pixels (as they were more difficult to classify or rare) and a subset of the pixels for other labels to keep their frequency between the other classes. The final result is shown in table C-1.

Table C-1: Individual blocks class final distribution after adjusting class distribution

Block#	*Water*	*developed*	*Grass/Shrub*	*Forest*	*Bare*	*Agriculture*	*Wetland*	*Total points*
01_0438	8972	3653	1133	1005	938	138	698	16537
02_0049	1851	19794	74	1308	3	756	4058	27844
03_0003	3595	8452	294	423	212	3178	10127	26281
04_0030	241	1914	249	3302	37	352	893	6988
05_0247	2243	544	1513	1914	896	0	290	7400
06_0258	140	25263	3851	2699	0	577	845	33375
07_0016	0	37250	2467	37	0	10346	0	50100
08_0017	0	2552	9837	858	0	0	0	13247
09_0275	151	2121	2014	2407	569	0	617	7879
10_0096	296	17452	13057	520	86	1133	0	32544
11_0083	53	2056	4087	1475	25	2135	0	9831
12_0116	4564	5841	553	147	85	12508	1226	24924
13_1185	0	766	17385	0	21961	0	8149	48261
14_0493	803	5206	11440	0	29891	0	0	47340
15_0266	4449	1242	10573	2002	135	278	0	18679
16_0061	0	1276	11712	2424	0	553	219	16184
17_0013	38	5681	2762	5237	46	3146	14	16924
18_0143	0	4193	21215	99	177	595	639	26918
19_0276	56	1722	3440	4299	0	0	0	9517
20_0031	145	432	9454	3083	640	0	0	13754
21_0235	132	2639	8878	4978	0	344	466	17437
22_0303	0	4313	4195	1167	2145	0	0	11820
23_0135	9	3967	2088	2823	0	0	0	8887
24_0315	76	10036	11590	0	1211	0	232	23145
25_0049	0	1013	10029	2812	15	1551	0	15420
26_0265	46	2100	12902	5	14	5117	0	20184
27_0101	211	7678	1369	270	99	18632	0	28259
28_0005	36	1960	9127	749	0	10839	534	23245
29_0124	509	9330	1211	1755	0	3580	0	16385
30_0080	73	1212	4885	1123	42	728	0	8063
31_0021	48	2411	966	2629	0	7906	0	13960
32_0075	114	1829	1292	2112	0	2460	6435	14242

Block#	*Water*	*developed*	*Grass/Shrub*	*Forest*	*Bare*	*Agriculture*	*Wetland*	*Total points*
33_0005	5346	2596	280	1472	4	1648	842	12188
34_0002	269	3071	309	2651	0	1507	236	8043
35_0171	37	3801	640	2405	2	1605	567	9057
36_0009	4	1478	780	5830	0	566	9	8667
37_0015	291	5536	620	3190	108	1598	0	11343
38_0088	624	2483	682	5379	0	2541	54	11763
39_0058	1354	2802	79	3411	37	10539	20	18242
40_0153	400	1211	663	3114	0	11497	0	16885
41_0034	1367	518	1326	5467	21943	0	8	30629
42_0787	2098	2591	5636	68	0	12885	395	23673
43_2156	49	588	14807	919	73	4060	537	21033
44_0040	142	715	16466	1361	0	976	0	19660
45_0059	133	2412	316	4476	0	1403	0	8740
46_0205	42	2061	8329	310	0	5852	3156	19750
47_0572	3408	10611	260	152	0	16358	679	31468
48_0017	164	4023	316	1547	25	11292	881	18248
49_0030	49	993	1331	1911	0	3899	17466	25649
50_0140	3026	1212	57	5964	0	145	4355	14759
51_0016	325	3947	522	2732	0	3287	5411	16224
52_0005	421	5267	153	2405	0	11437	1911	21594
53_0161	2925	16296	120	533	40	5075	149	25138
54_0017	2430	18215	345	569	149	2496	3720	27924
55_0195	194	3848	421	1453	13	18690	169	24788
56_0005	34	6126	84	1795	0	10610	2523	21172
57_0051	14509	3250	78	85	0	6811	2043	26776
58_0576	250	2988	409	7361	113	648	1494	13263
59_0140	8661	9143	132	548	330	83	1124	20021
60_0059	252	3458	0	6233	0	4785	0	14728
61_0005	153	5433	0	5621	0	4808	653	16668
62_0007	1420	5554	0	8133	0	0	453	15560
63_0031	68	4808	0	4380	0	5033	0	14289
64_0018	464	26194	0	1721	0	405	59	28843
65_0451	279	1083	137	1745	0	6456	0	9700
66_0021	283	2396	16	5529	0	697	0	8921
67_0029	426	25079	116	3272	0	3352	0	32245
68_0034	158	3323	26	6161	0	438	0	10106
69_0111	10	3511	341	6924	0	1940	0	12726
70_0015	91	1669	1008	4452	0	3347	0	10567
71_0042	29	11848	130	2946	320	2891	0	18164
72_0060	2957	3246	272	1468	0	12188	161	20292
73_0786	882	20353	160	1989	0	742	141	24267
74_0176	54	4215	251	3546	28	3738	0	11832
75_0747	10738	1923	97	369	0	603	3727	17457

Block#	***Water***	***developed***	***Grass/Shrub***	***Forest***	***Bare***	***Agriculture***	***Wetland***	***Total points***
76_0047	759	13767	132	52	0	4218	13416	32344
77_0057	365	1845	318	3975	60	763	340	7666
78_0007	24	1871	2396	2406	0	501	0	7198
79_0018	255	0	21039	150	2893	0	0	24337
80_0039	0	1278	11879	1194	806	0	0	15157
81_0469	5	4768	10129	35	918	337	0	16192
82_0004	289	2192	407	2688	35	5691	1044	12346
83_0346	217	18532	223	3712	900	2503	840	26927
84_0024	2379	40745	80	486	0	162	2798	46650
sum	99,960	522,771	299,960	199,957	88,024	299,958	106,823	1,617,453

Appendix D Definition of spectral indices used in this research

Below you will find formulas for the eight spectral indices we used in this work, divided into 4 categories. Unless otherwise stated, the index definitions are taken from L3Harris Geospatial Alphabetical List of Spectral Indices[4].

➢ To better delineate vegetation, we looked at below three indices:

1. NDVI: This is the basic and most widely used vegetation index.

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

2. Modified Soil Adjusted Vegetation Index 2 (MSAVI2): This index is also well known and has more discrimination power to highlight vegetation and as the name suggests corrects some of NDVI dependency to the soil type and inclusion of bare soil in pixel.

$$MSAVI2 = \frac{2NIR + 1 - \sqrt{(2NIR + 1)^2 - 8(NIR - Red)}}{2}$$

3. Non-Linear Index (NLI): This index aims to help model nonlinear relationships between vegetation indices and surface parameters, and defined as:

$$NLI = \frac{NIR^2 - Red}{NIR^2 + Red}$$

➢ To better delineate bare soil and wetland class, we looked at below three indices:

4. Bare Soil Index (Sahana, Sajjad, and Ahmed 2015): The higher the BSI, there will be more bare areas and less vegetation.

$$BSI = \frac{(SWIR1 + Red) - (NIR + Blue)}{(SWIR1 + Res) + (NIR + Blue)}$$

[4] https://www.l3harrisgeospatial.com/docs/alphabeticallistspectralindices.html

5. Drought Distance index (Sadeghi et al. 2017): This index measures the distance of pixel to the origin in NIR-Red space and normalizes it w.r.t NDVI. The lower the index, the wetter or less vegetation canopy is the area. I dropped the denominator and only used the numerator in my calculations because for wetland I am more concerned about wetness (being close to the origin) than vegetation.

$$DD = \frac{\sqrt{Red^2 + NIR^2}}{1 + NDVI}$$

6. Visible and Shortwave infrared Drought Index (Sadeghi et al. 2017): This index is theoretically based on the difference between moisture-sensitive bands (SWIR and red) and moisture reference band (blue) to account for different sources of moisture (soil or vegetation). The higher the VSDI, the wetter the soil or vegetation.

$$VSDI = 1 - [(SWIR1 - blie) + (Red - Blue)]$$

➢ And for delineating water bodies, we used below index:

7. Modified Normalized Difference Water Index (Guo et al. 2017): It is reported that this modified water index picks water bodies more precisely than its former variants.

$$MNDWI = \frac{Green - SWIR1}{Green + SWIR1}$$

➢ To help identify built-up and impervious land cover, we used below index:

8. Enhanced Normalized Difference Impervious Surfaces Index (Junyi Chen et al. 2019) has been introduced recently and it is reported to have better performance compared to other available index, NDBI (normalized difference built-up index).

$$ENDISI = \frac{Blue - \alpha\left[\frac{SWIR1}{SWIR2} + MNDWI^2\right]}{Blue + \alpha\left[\frac{SWIR1}{SWIR2} + MNDWI^2\right]}, \alpha = \frac{2Blue_{Mean}}{\left(\frac{SWIR1}{SWIR2}\right)_{Mean} + MNDWI^2_{Mean}}$$

Appendix E: Comparing our reference data to NLCD 2016 data

We did a comparison of the labels for the same points (our evaluation dataset) and the resulting confusion matrix is shown in table E-1. Class titles are:

Our land cover classes	*NLCD classes (Alaska-specific classes not included)*	
0: Water	11: Water	42: Evergreen forest
1: Developed	21: Developed, open space	43: Mixed forest
2: Grass/shrub	22: Developed, low intensity	52: Shrub/scrub
3: Forest	23: Developed, medium intensity	71: Grassland
4: Bare	24: Developed, high density	81: Pasture/Hay
5: Agriculture	31: Barren land	82: Agriculture
6: Wetland	41: Deciduous forest	90: Woody wetland
		95: Herbaceous wetland

Classes that cross to each other are shown by diagonal red-font numbers and major confused classes are highlighted yellow. We tried different options of including or dropping the confused classes in our calculations and found that the best performance (on overall accuracy and average F1) by dropping two cases marked by red rectangles.

Table E-1: comparison of our labels and NLCD labels extracted for points in our evaluation dataset

		Our class						
		0	***1***	***2***	***3***	***4***	***5***	***6***
NLCD class	***11***	11034	59	5	11	14	13	532
	21	41	14084	267	196	2	383	47
	22	10	19324	39	17	10	119	27
	23	5	12464	6	2	11	18	5
	24	1	3795	0	0	6	0	0
	52	50	2336	18739	712	1914	378	152
	71	25	1249	11568	130	896	1565	287
	41	44	1028	442	8174	1	280	210
	42	41	786	744	7469	502	166	54
	43	9	564	60	3850	0	96	24
	31	6	291	324	2	6660	29	119
	81	8	2229	1354	103	4	4518	194
	82	0	1094	157	39	15	26364	91
	90	61	219	161	2031	0	65	2204
	95	43	169	379	76	5	257	8241

Appendix F: Selected blocks imagery and land cover maps

We provide in this appendix some selected blocks high-resolution imagery (screenshot taken from ArcMap's imagery base map), NLCD 2016 map (snapshot taken from ArcMap's NLCD 2016 base map), and our CNN+ST-LSTM model prediction. According to the ESRI web site, the ArcMap's imagery base map is updated to 2021, but we cannot get a historical map of 2015/2016 in ArcMap. We also left NLCD map with its original colormap and provided below the color legend for different classes, plus our own legend.

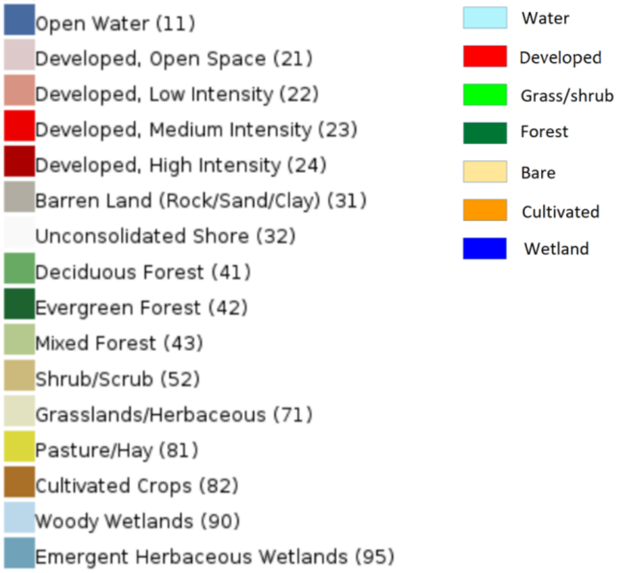

Open Water (11)
Developed, Open Space (21)
Developed, Low Intensity (22)
Developed, Medium Intensity (23)
Developed, High Intensity (24)
Barren Land (Rock/Sand/Clay) (31)
Unconsolidated Shore (32)
Deciduous Forest (41)
Evergreen Forest (42)
Mixed Forest (43)
Shrub/Scrub (52)
Grasslands/Herbaceous (71)
Pasture/Hay (81)
Cultivated Crops (82)
Woody Wetlands (90)
Emergent Herbaceous Wetlands (95)
Water
Developed
Grass/shrub
Forest
Bare
Cultivated
Wetland

Figure F-1: NLCD 2016 legend (left) and our predicted map legend (right)

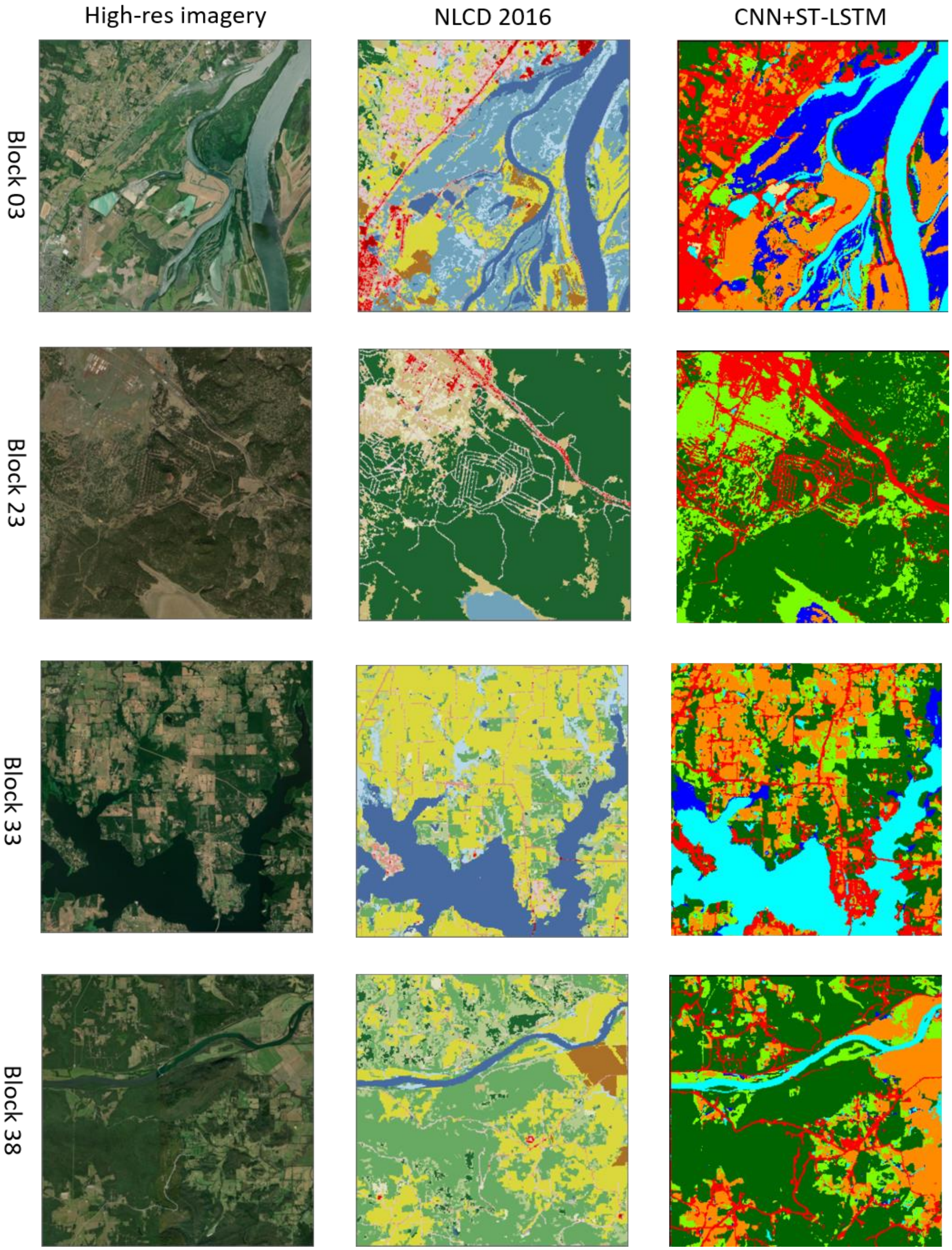


Figure F-2: Selected blocks high-resolution imagery, NLCD map, and our predicted map

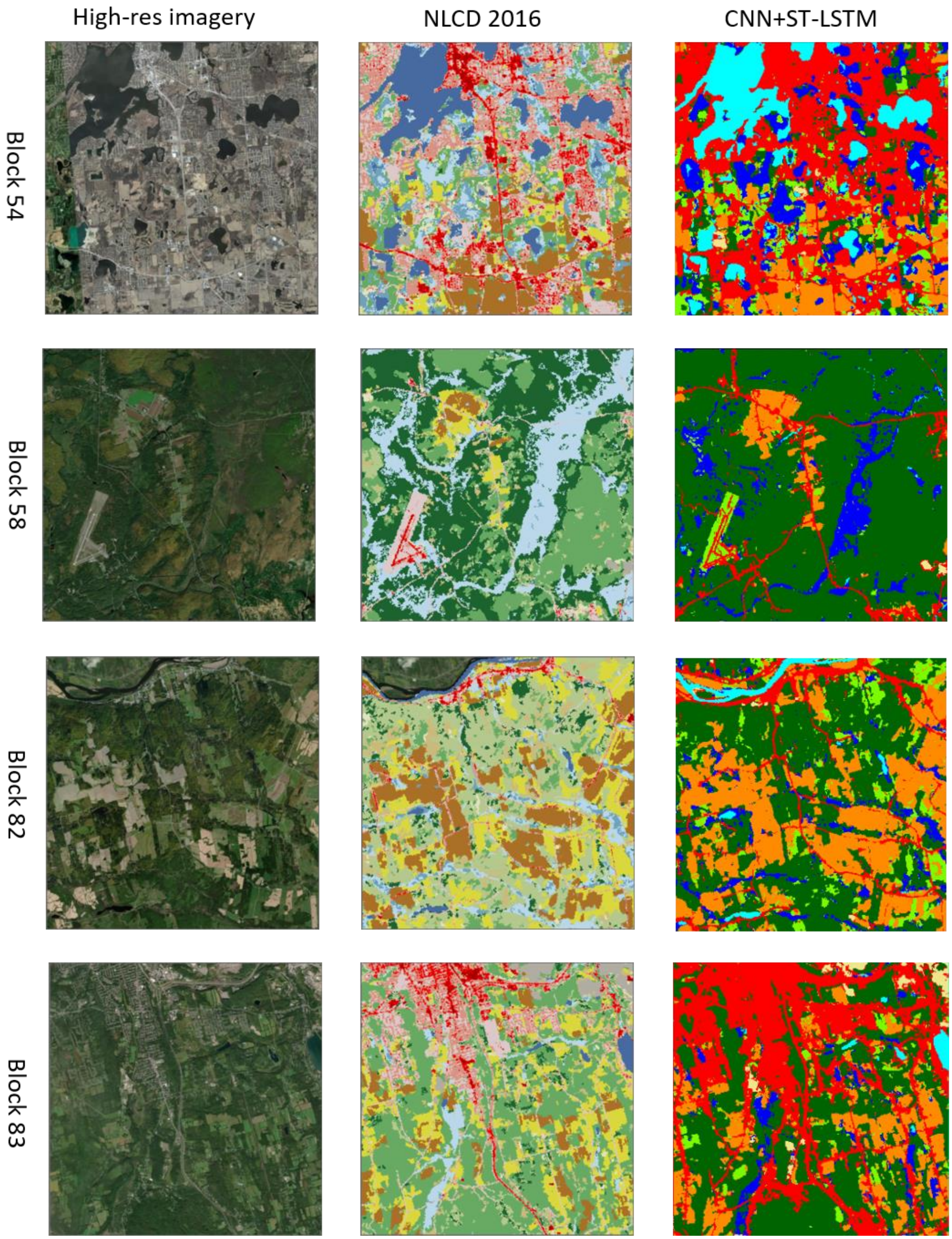


Figure F-2 (continued): Selected blocks high-resolution imagery, NLCD map, and our predicted map

Appendix G: Visual inspection of ensemble models

In this appendix we provide visualization of ensemble models performance by showing full and zoomed image of the prediction map for block representing ecoregion 37 (Arkansas Valley). This block is located north of the city of Clarksville, AK, and the predicted map for the whole block and a zoomed rectangle in its lower right part are shown in figure G-1. As you see, very little change is visible by looking at the predicted map for base model and two ensemble models and the change actually happens in very fine details. The only distinct change that we found in this figure is the area marked with white polygon in the zoomed image, which shows the correction of some pixels that has been incorrectly classified as agriculture and return it back to the forest class. The high-resolution imagery of the zoomed area is also provided in figure G-2 for reference.

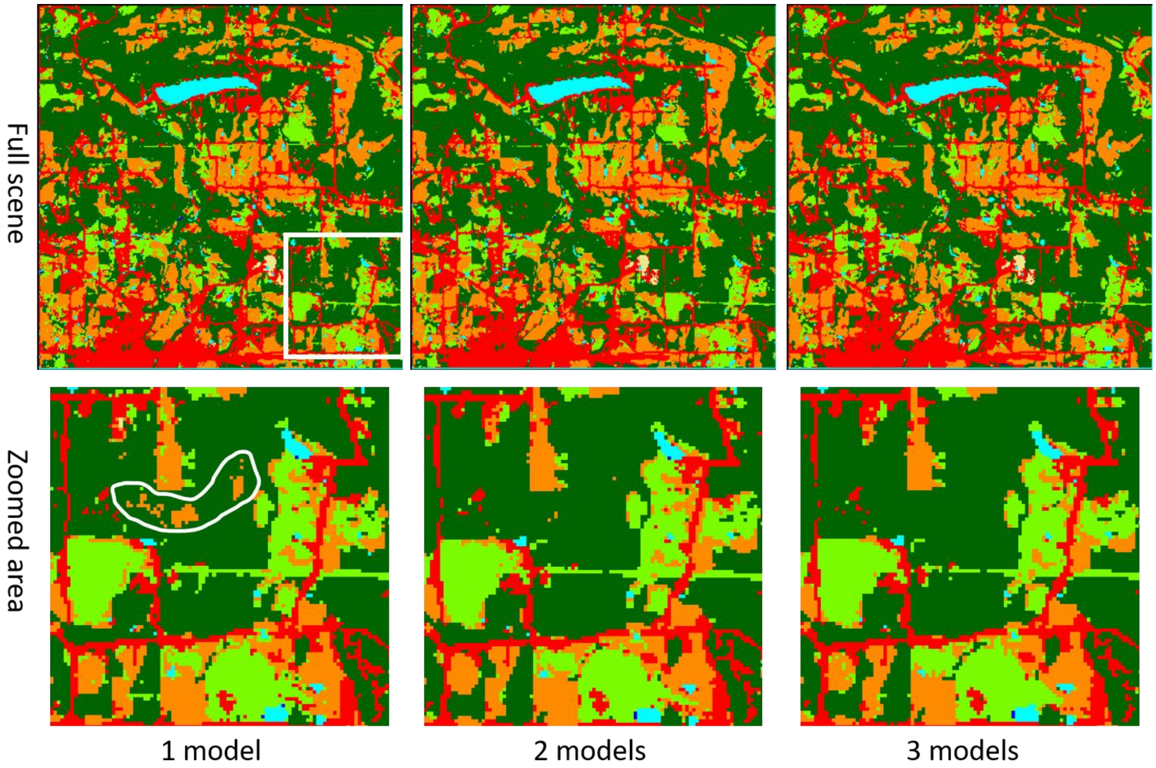


Figure G-1: Full (top) and zoomed rectangle (bottom) of block representing ecoregion 37 for one model and ensemble of two and three top models.

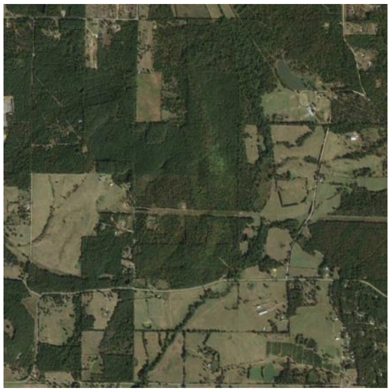


Figure G-2: High-resolution imagery of zoomed block in figure G-1

CHAPTER 5: CONCLUSION

5.1. Summary

This work advances the integration of deep networks in remote sensing. To this end, we developed the full chain of remote sensing data extraction, reference data generation, model optimization, final model evaluation, and additional model enhancements, all in the scope of mapping blocks from each ecoregion in the conterminous United States. Our model's very high achieved accuracy shows the great potential of deep learning models for other applications in any geographical scope.

As described in the introduction chapter, we considered three research objectives and formulated the three main hypotheses as listed below:

Hypothesis 1: Deep networks do not provide practical improvement in classification performance over conventional classifiers when datasets are small or only spectral data dimension is used.

This hypothesis was discussed in the second chapter and our results on 26 Landsat scenes (tested separately and then the results aggregated) showed no gain in using neural networks compared to established conventional classifiers such as SVM, KNN, or tree ensembles. We did an extensive grid search of the most important parameters in our simulations to obtain their best performance. We also tried increasing the neural network layers to test the promised capability of deep learning methodology but it did not improve classification performance. As discussed in the paper, we think this can be attributed to the small datasets with low feature dimension that prohibit full deployment of deep networks potential. As the results of chapter 4 show, deeper networks with big and rich input data will pass conventional methods with considerable margin.

We also showed in the first chapter how overall performance metric is dependent on class distribution and how either calibration or test data class distribution affects the simulation results. This suggested that we better to try another metric to assess our networks during optimization in chapter 4, which led to reliance on F1 instead of overall accuracy for model comparison and selection.

Hypothesis 2: Our proposed network architecture for processing temporal-spectral-spatial Landsat data can achieve better accuracy than its companion spectral-spatial or spectral-only variants, and performs better than currently available global land cover products (over conterminous US).

This was the main hypothesis in our study and we answered it through analysis in chapters 3 and 4. We first conducted a review on current applications of mono-temporal deep networks over more than 100 papers in chapter 3 and showed the existing problems with lack of big datasets, and possible close competition from conventional methods when they are fed with feature-rich datasets. In other words, with mono-temporal implementation and existing datasets, the deep networks still may not provide significant gain. However, we showed that using our large area dataset containing samples from all ecoregions in conterminous US and employing full spectral-spatial-temporal dimensions of data, processed by our complex hybrid deep network structure, we can achieve considerable gain relative to both our baseline benchmarks and other available global land cover products. Although the developed network was highly accurate, we showed that the test performance in local areas still have significant variations and needs to be improved.

Hypothesis 3: There is considerable improvement is fusion of different Landsat sensors in terms of achieved accuracy and *minimum number of requires scenes.*

This hypothesis was also covered in analyses conducted in chapter 4, and we showed that fusion of Landsat 5 and 7 or Landsat 7 and 8 (over their overlapping mission times) considerably improves the overall performance metrics, compared to single-sensor data. We attribute this to increase in available data by combining observations by multiple sensors in the same input sequence. It may also be due to the fact that the model has been trained on fused sequences. Interestingly, our study showed that the multi-sensor response is more stable and keeps its good performance in shorter sequences compared to single-sensor input data. Our study also showed that comparing single sensors, Landsat 8 provides the best performance, then Landsat 5, and then Landsat 7. The performance of model under single-sensor or fused data varies between different land cover classes, but the worst class under fused data has still better performance than the best class under single-sensor scenario.

5.2. Future work

Our proposed network with 2.6 million parameters is a complex model with very good performance, but there are still many other possibilities for continuing this study in different aspects. First of all, model assessment over whole conterminous United States should be executed. Improving model performance on local scope and improving its generalization is another important extension. The idea of transfer learning and fine-tuning a pre-trained model seems promising in this regard and these ideas have been presented in recent studies.

The current model has been developed over a simplified general land cover scheme, suitable for regional and global studies. However, given the model high power, we can proceed to higher levels of detail and develop our model based on a more detailed land cover / land use classification scheme or set of models for different applications, same as a coarse/fine approach. Both of these maps are much needed in any resource management administration and remote sensing technology is the only viable approach to solve these sort of problems, particularly over large areas.

Apart from model application, there are many areas of further work and improvement with the model itself. For example, other possibilities for network design and feature combination strategies can be considered.

The most important improvement in the model, however, may be transition to a fully convolutional model. Such a change may greatly improve model computing budget in terms of storage, memory, and simulation time. But it requires our reference labels to be available wall-to-wall in training areas. It might be possible to adapt the fully convolutional network to ignore a few pixels without reference labels but our current dataset is too patchy, and our limited tries for implementing fully convolutional network resulted in performance drop.

Along with transition to fully convolutional model, there is a strong desire for integration of object-based methods with pixel-based classifiers in the remote sensing field. For example, other researchers have tried to combine the classifier output with another output generated by segmentation algorithms to enhance the object boundaries. Being itself a rich field of study, object-based classification can offer many possibilities to join and enhance our model output.

Some pre- or post-processing can also be automated and done by network which we did it ourselves. A simple example is cloud filtering, which we did separately and masked pixels covered by cloud or cloud shadow (based on Landsat quality bit setting). As some other research showed, this task can also be handled by deep networks themselves and may provide a higher level of accuracy when done by the model itself. Fusion of other sensors in addition to Landsat – particularly Sentinel, due to its free global data availability – may be very helpful.

One of the direct benefits of this research was to employ advanced hardware resources such as GPUs, cloud-based resources and parallel computing, and latest machine learning scripting platforms. This will serve as a benchmark to have a better estimate on needed resources for research in this field and to enrich the skills on using the more advanced technology in the department. Deep learning provides many opportunities for processing of different data dimensions in specialized structures, and it will allow scientists to concentrate on the network design instead of dealing with custom feature extraction methods. Our hybrid network is a simple example in this regard, and the possibilities ahead are unlimited.

VITA

Shahriar Shah Heydari

321 Baker Labs, 1 Forestry Dr., (315)395-1471
Syracuse, NY 13210 sshahhey@esf.edu

EDUCATION

PhD in Environmental Resources Engineering / Geospatial Analysis **May 2021**
State University of New York - College of Environmental Science and Forestry,
Syracuse, NY, USA
Master of Science in Environmental Resources Management **Dec. 2014**
University of Southern Denmark, Esbjerg, Denmark
Master of Science in Electrical Engineering **Apr. 1995**
Sharif University of Technology, Tehran, Iran
Bachelor of Science in Electrical Engineering **Apr. 1992**
Sharif University of Technology, Tehran, Iran

PUBLICATIONS

- Shahriar S. Heydari, Giorgos Mountrakis, “Large area land cover mapping using deep neural networks and Landsat time-series observations” (in final drafting)
- Shahriar S. Heydari, Giorgos Mountrakis, “Meta-analysis of deep neural networks in remote sensing: A comparative study of mono-temporal classification to support vector machines”, *ISPRS Journal of Photogrammetry and Remote Sensing*,152 (2019), https://doi.org/10.1016/j.isprsjprs.2019.04.016.
- Shahriar S. Heydari, Giorgos Mountrakis, “Effect of classifier selection, reference sample size, reference class distribution and scene heterogeneity in per-pixel classification accuracy using 26 Landsat sites”, *Remote Sensing of Environment*, 204 (2018), https://doi.org/10.1016/j.rse.2017.09.035.
- Shah Heydari Sh., Vestergaard N., "Alternate solutions in mixing energy tax/subsidy and emission control policies", University of Southern Denmark, Department of Environmental and Business Economics, Working Paper No. 119/2015, ISSN 1399-3224

PROFESSIONAL EXPERIENCE

Graduate Research Assistant **2015-2021**
State University of New York - College of Environmental Science and Forestry,
Syracuse, NY, USA
GIS specialist (Intern) **Summer 2018**
New York State Department of Environmental Conservation, Albany, NY, USA
Research Scientist **Summer 2015**
Wegener Center for Climate and Global Change, Graz, Austria
Senior Electrical Engineer and Engineering Advisor **2010-2012**
Rahgozin Rayaneh, Tehran, Iran
Electrical Engineer and Project Manager **1997-2010**
Various companies, Tehran, Iran