

# **Convolutional Neural Networks for Land-cover Classification Using Multispectral Airborne Laser Scanning Data**

by

Zhuo Chen

A thesis  
presented to the University of Waterloo  
in fulfillment of the  
thesis requirement for the degree of  
Master of Science  
in  
Geography

Waterloo, Ontario, Canada, 2018

©Zhuo Chen 2018

## **AUTHOR'S DECLARATION**

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

With the spread of urban culture, urbanisation is progressing rapidly and globally. Accurate and update land cover (LC) information becomes increasingly critical for protecting ecosystems, climate change studies and sustainable human-environment development. It has been verified that combining spectral information from remotely sensed imagery and 3D spatial information from airborne laser scanning (ALS) point clouds has achieved better LC classification accuracy than that obtained by using either of them solely. However, data fusions can introduce multiple errors. To solve this problem, multispectral ALS developed recently is able to acquire point cloud data with multiple spectral channels simultaneously. Moreover, deep neural networks have been proved to be a better option for LC classification than those statistical classification approaches.

This study aims to develop a workflow for automated pixel-wise LC classification from multispectral ALS data using deep-learning methods. A total of six input datasets with a multi-tiered architecture and three deep-learning classification networks (i.e. 1D CNN, 2D CNN, and 3D CNN) have been established to seek the optimal scheme that lead to highest classification accuracy. The highest overall classification accuracy of 97.2% has been achieved using the proposed 3D CNN and the designed input dataset. In regard to the proposed CNNs, the overall accuracy (OA) of the 2D and 3D CNNs was, on average, 8.4% higher than that of the 1D CNN. Although the OA of the 2D CNN was at most 0.3% lower than that of the 3D CNN, the run time of the 3D CNN was five times longer than the 2D CNN. Thus, the 2D CNN was the best choice for the multispectral ALS LC classification when considering efficiency. For different input datasets, the OA of the designed input datasets was, on average, 3.8% higher than that of the classic input datasets. Results also showed that the multispectral ALS data is superior to both multispectral optical imagery and single-wavelength ALS data for LC classification. In conclusion, this thesis suggests that LC classification can be improved with the use of multispectral ALS data and deep-learning methods.

## Acknowledgements

First and foremost, I would like to express my deepest appreciation to my supervisor, Professor Dr. Jonathan Li, whose contribution in stimulating suggestions and encouragement helped me to coordinate my graduate studies especially in completing this thesis. I feel very fortunate that he provided me the opportunity to get involved in and work on state-of-the-art technologies, especially the multispectral LiDAR and deep learning. I am deeply grateful for his guidance and support throughout my graduate studies.

I would like to thank my thesis committee members, Dr. Michael Chapman, Professor at the Department of Civil Engineering, Ryerson University, Dr. Peter Deadman, Associate Professor at the Department of Geography and Environmental Management, University of Waterloo, and Dr. John Zelek, Associate Professor at the Department of Systems Design Engineering, University of Waterloo, for reviewing my thesis, serving as my thesis examining committee, and providing me the constructive comments and valuable suggestions.

Furthermore, I would like to acknowledge Teledyne Optech for providing me the datasets acquired by their multispectral ALS system TITAN to support my study. I would like to thank staffs in the department of Geography and Environmental Management, especially Alan Anthony and Susie Castela, for their help. My sincere thanks also go to Zilong Zhong, Lingfei Ma, Ying Li, Ming Liu, Weiya Ye, Yue Gu, and Mengge Chen, all the members in Mobile Sensing and Geodata Analytics Lab, who shared their experiences during the group meetings and created a delightful environment in the office. A special thanks goes to Yuhao Xie for providing me advice on the studies of Python and deep learning.

Last but not least, I would like to express deepest gratitude to my dear parents for their unconditional love, support and emotional encouragement throughout my studies. I could not be more grateful for having such wonderful parents. Thanks to my boyfriend, Hao Wu, and my friends, Liuyi Guo, Bo Sun, Michael Guo, Yuhao Xie, and Hongjing Chen, who accomplish me and offer me with their help, understanding, and encouragement.

# Table of Contents

AUTHOR'S DECLARATION .....	ii
Abstract .....	iii
Acknowledgements .....	iv
Table of Content .....	v
List of Figures .....	viii
List of Tables .....	x
List of Abbreviations .....	xii
Chapter 1 Introduction.....	1
1.1 Motivation .....	1
1.2 Objectives of the Study .....	4
1.3 Structure of the Thesis.....	5
Chapter 2 Background and Related Studies .....	6
2.1 Multispectral Airborne Laser Scanning System.....	6
2.1.1 Components of a Multispectral ALS System.....	7
2.1.2 Direct Geo-referencing .....	10
2.1.3 Basic Principles of Multispectral ALS System.....	11
2.1.4 Multi-wavelength Intensity Maps .....	12
2.2 LC Classification for Multispectral ALS Datasets .....	13
2.2.1 Potential of Multispectral ALS technique in LC classification .....	13
2.2.2 Classification Methods Used for Multispectral ALS Datasets .....	14
2.3 Deep learning .....	18
2.3.1 Relationship between Machine Learning and Deep Learning .....	20
2.3.2 Deep Learning Algorithms .....	22
2.3.3 Deep Learning in LC Classification .....	24
2.4 Chapter Summary.....	28
Chapter 3 Deep Learning for Multispectral ALS LC Classification .....	29
3.1 Study Area and Datasets.....	29
3.1.1 Study Area .....	29
3.1.2 Datasets.....	31

3.2	Workflow of the Methodology.....	33
3.3	Multispectral ALS Data Pre-processing.....	35
3.3.1	Multispectral ALS Data De-noising and Intensity Normalization .....	35
3.3.2	Multispectral ALS-derived Intensity Imagery .....	36
3.3.3	Multispectral ALS-derived Height Imagery .....	38
3.3.4	Establishment of Input Datasets.....	39
3.4	Labelling.....	40
3.5	Selection of Deep-learning Networks .....	41
3.6	Proposed CNNs .....	42
3.6.1	Convolutional Layers.....	43
3.6.2	Pooling Layers .....	44
3.6.3	Fully Connected Layers .....	44
3.6.4	Other Functional Layers .....	45
3.6.5	Involved Hyper-parameters .....	45
3.7	Implementation of the Proposed CNNs.....	45
3.7.1	Programming Language and Libraries.....	46
3.7.2	Data Import .....	47
3.7.3	Separation of Training, Validation, and Testing Data .....	48
3.7.4	Training Process.....	48
3.7.5	Predict Process .....	50
3.7.6	Involved Hyper-parameters .....	50
3.8	Methods of Accuracy Assessment .....	50
3.8.1	Validation for Labelling.....	50
3.8.2	Accuracy Assessment for Classification.....	50
3.9	Chapter Summary .....	52
Chapter 4	Results and Discussion .....	54
4.1	Labelled Dataset .....	54
4.1.1	Result of Labelling.....	54
4.1.2	Validation of Labelling.....	57
4.2	Hyper-parameters .....	57
4.2.1	Shape of Each Input Unit.....	58

4.2.2	The Number of Kernels .....	59
4.2.3	Size of Kernels .....	59
4.2.4	Size of Pooling Windows.....	60
4.2.5	Units of Dense.....	61
4.2.6	Rate of Training, Validation and Testing Data.....	62
4.2.7	Learning Rate.....	63
4.2.8	Summary of Hyper-parameters.....	63
4.3	Analysis of LC Classification.....	64
4.3.1	Performances of Different Input Data Combinations .....	65
4.3.2	Performances of Different CNNs.....	74
4.3.3	Efficiency of Different CNNs.....	84
4.4	Comparison of LC Classifications for Multispectral ALS Data .....	84
4.5	Chapter Summary.....	86
Chapter 5	Conclusions and Recommendations .....	88
5.1	Conclusions and Contributions .....	88
5.2	Limitations and Recommendations .....	90
REFERENCES	.....	91

## List of Figures

<b>Figure 2.1</b> Titan laser channels with spectral signatures for selected objects (Teledyne Optech Titan, 2015) .....	7
<b>Figure 2.2</b> Optech Titan system (Teledyne Optech Titan, 2015) .....	8
<b>Figure 2.3</b> Cooperation principle of GNSS and IMU (Chen et al, 2018).....	10
<b>Figure 2.4</b> Structure of Standard Deep Neural Network .....	19
<b>Figure 2.5</b> the Relationship among Deep Learning, Machine Learning, Representation Learning, and Artificial Intelligence (Goodfellow et al., 2016) .....	20
<b>Figure 2.6</b> the Relationship among Deep Learning, Machine Learning, Representation Learning, and Rule-based Systems (Goodfellow et al., 2016) .....	21
<b>Figure 2.7</b> Category and Representative Examples of Deep Learning Algorithms .....	22
<b>Figure 2.8</b> Comparisons among Four Categories of Deep Learning Algorithms.....	24
<b>Figure 3.1</b> A map of the study area .....	30
<b>Figure 3.2</b> Workflow of the methodology .....	34
<b>Figure 3.3</b> Multispectral ALS on the study area.....	37
<b>Figure 3.4</b> Multispectral ALS height models on the study area .....	38
<b>Figure 3.5</b> Multi-tiered architecture of input datasets .....	40
<b>Figure 3.6</b> Structure of CNNs.....	42
<b>Figure 3.7</b> Establishment of CNNs.....	43
<b>Figure 3.8</b> Workflow of model implementation.....	46
<b>Figure 3.9</b> Imported libraries .....	47
<b>Figure 3.10</b> Importing data pixel by pixel .....	47
<b>Figure 3.11</b> Selection of valid pixels .....	48
<b>Figure 3.12</b> Separation of Training and Testing Data .....	48
<b>Figure 3.13</b> A forward step and a backward step of training process .....	49
<b>Figure 3.14</b> Training process .....	49
<b>Figure 3.15</b> Predict process .....	50
<b>Figure 4.1</b> Labelled LC map of the study area .....	55
<b>Figure 4.2</b> Results of different shape of input unit in 2D and 3D CNNs .....	58
<b>Figure 4.3</b> Results of different number of kernels in the 1D, 2D and 3D CNNs .....	59



**Figure 4.4** Results of different size of kernels in the 1D, 2D and 3D CNNs ..... 60  
**Figure 4.5** Results of different size of kernels in the 1D, 2D and 3D CNNs ..... 61  
**Figure 4.6** Results of different units of dense in the 1D, 2D and 3D CNNs ..... 62  
**Figure 4.7** Results of different units of dense in the 1D, 2D and 3D CNNs ..... 63  
**Figure 4.8** Predicted maps of different input data combinations with the 3D CNN ..... 66  
**Figure 4.9** Predicted maps of different CNNs with Combination 4 ..... 76

## List of Tables

<b>Table 2.1</b>	Specifications of Titan.....	7
<b>Table 2.2</b>	Studies Related to Multispectral ALS Data Classification .....	13
<b>Table 3.1</b>	Summary of data collection parameters .....	32
<b>Table 3.2</b>	Summary of the cropped, pre-processed and merged data.....	32
<b>Table 3.3</b>	Content of input combinations .....	40
<b>Table 3.4</b>	LC types and examples.....	41
<b>Table 3.5</b>	Hyper-parameters involved in the establishment of models .....	45
<b>Table 3.6</b>	Hyper-parameters involved in the implementation of CNNs.....	50
<b>Table 3.7</b>	An example table of a confusion matrix with UA and PA.....	51
<b>Table 4.1</b>	Detailed examples of each LC class in labelled dataset.....	56
<b>Table 4.2</b>	Statistics of the labelled dataset .....	56
<b>Table 4.3</b>	Confusion matrix of first-labelled dataset and relabelled dataset .....	57
<b>Table 4.4</b>	Summary of hyper-parameters .....	63
<b>Table 4.5</b>	OA and kappa coefficient of each model .....	65
<b>Table 4.6</b>	Confusion matrix for Combination 1 with the 3D CNN .....	67
<b>Table 4.7</b>	Confusion matrix for Combination 2 with the 3D CNN .....	67
<b>Table 4.8</b>	Confusion matrix for Combination 3 with the 3D CNN .....	67
<b>Table 4.9</b>	Confusion matrix for Combination 4 with the 3D CNN .....	68
<b>Table 4.10</b>	Confusion matrix for Combination 5 with the 3D CNN .....	68
<b>Table 4.11</b>	Confusion matrix for Combination 6 with the 3D CNN .....	68
<b>Table 4.12</b>	UA of the predicted WAT for Combination 4 and each CNN.....	77
<b>Table 4.13</b>	PA of the actual WAT for Combination 4 and each CNN.....	78
<b>Table 4.14</b>	UA of the predicted TRE for Combination 4 and each CNN .....	78
<b>Table 4.15</b>	PA of the actual TRE for Combination 4 and each CNN .....	79
<b>Table 4.16</b>	UA of the predicted ROD for Combination 4 and each CNN.....	80
<b>Table 4.17</b>	PA of the actual ROD for Combination 4 and each CNN.....	80
<b>Table 4.18</b>	UA of the predicted BAL for Combination 4 and each CNN .....	81
<b>Table 4.19</b>	PA of the actual BAL for Combination 4 and each CNN .....	81
<b>Table 4.20</b>	UA of the predicted BUD for Combination 4 and each CNN.....	82

**Table 4.21** PA of the actual BUD for Combination 4 and each CNN..... 82  
**Table 4.22** UA of the predicted OIS for Combination 4 and each CNN..... 83  
**Table 4.23** PA of the actual OIS for Combination 4 and each CNN..... 83  
**Table 4.24** Total model parameters and running time of each CNN with Combination 4 ..... 84  
**Table 4.25** Studies of LC Classification Methods for Multispectral ALS Data..... 85

## List of Abbreviations

1D	One-dimensional
2D	Two-dimension
3D	Three-dimensional
ALS	Airborne laser scanning
BAL	Bare land
BUD	Buildings
CE	Commission errors
CNN	Convolutional Neural Networks
CV-CNN	Complex-valued CNN
CWNN	Convolutional-wavelet neural networks
DBN	Deep belief networks
FOV	Field of view
G	Green
GNSS	Global navigation satellite system
HR	High-resolution
IMU	Inertial measurement unit
LC	Land cover
LiDAR	Light detection and ranging
LR	Logistic regression
MLC	Maximum likelihood classification
NIR	Near infrared
OE	Omission errors
OIS	Other impervious surfaces
OA	Overall accuracy
PolSAR	Particular polarimetric SAR
PA	Producer's accuracy
PRF	Pulse repetition frequency
RF	Random forest
ReLU	Rectified linear unit

RBM	Restricted Boltzmann machine
ROD	Roads
SWIR	Shortwave infrared
SWOOP	Southwestern Ontario orthophotography project
SAE	Sacked auto-encoders
SDAE	Sacked de-noising auto-encoders
SVM	Support vector machine
SAR	Synthetic aperture radar
TOF	Time-of-flight
TRE	Trees
UAV	Unmanned aerial vehicles
UA	User's accuracy
VHR	Very-high-resolution
WAT	Water

# Chapter 1

## Introduction

### 1.1 Motivation

With the development of society, urban culture is gradually taking the place of rural culture. Meanwhile urbanisation, a modern phenomenon, is spreading rapidly and globally (the United Nations, 2015). According to an assessment completed by the United Nations (2015), global urbanisation will increase to 66% by 2050. Although the rapid global urbanisation brings social and economic opportunities, it affects stability and sustainability of the environment, accelerates the variation of land cover (LC), and consequentially brings challenges to the supervision of LC (Pugh, 2014).

Defined as the physical composition and features of objects at the surface of the Earth (Cihlar, 2000), LC is a vital parameter than can be used to supervise the changing world. Monitoring the type, scope, and distribution of LC is crucial for the supervision of ecosystems (Lunetta et al., 2002), the Earth's radiation balancing (National Research Council, 2005), and climate change (Feddema et al., 2005). According to Lunetta et al. (2002), accurate LC maps are required for the monitoring of the ecosystem and the study of ecosystem processes such as the functions of wetland, the suitability of habitat, and the potential of soil erosion and sedimentation. Inadequate analysis and supervision of LC can lead to many problems for the ecosystem such as the loss, destruction and degradation of the habitat for various species (Guida-Johnson & Zuleta, 2013). Furthermore, LC change significantly affects the evaporation, transpiration, and heat flux on the ground surface, which further impacts the radiation balance on the Earth (National Research Council, 2005). The variation of the radiation balance on the Earth can lead to serious climate change (National Research Council, 2005). Moreover, the global climate can be impacted by LC change from both biogeochemical and biogeophysical aspects (Steffen et al., 2006). With regard to the biogeochemical aspect, the alteration of LC affects the biogeochemical cycles and consequently changes the chemical composition of the atmosphere (Feddema et al., 2005). With respect to the biogeophysical aspect, the change of LC directly impacts the physical composition and features of the Earth, which thereby affects the energy availability at the Earth's surface (Feddema et al., 2005). The change of climate (e.g. continually increased temperature and changes in precipitation patterns)

can cause serious problems such as the rise of the sea level and the increase of the ice-free arctic (Feddema et al., 2005). Thus, it is highly important to supervise climate change with the help of precise LC maps. In addition, a LC map plays a significant role in many fields such as policy-making since inaccurate LC maps may lead to inappropriate policies (e.g. Ittersum et. al., 1998). As such, it can be concluded that precise and efficient mapping of LC is essential to ensure an accurate representation of LC change, to protect the Earth and to ensure a sustainable human-environment development.

Traditionally, multispectral images are used to capture information on the surface of the Earth. Since different LC features have diverse spectral reflectance in various wavelengths, LC classes can be mapped via analysing spectral information recorded on multispectral images (Wilkinson, 2005). With the improvement of spatial resolution, LC classification with multispectral images should theoretically achieve higher precision. However, according to Wilkinson (2005), the LC classification accuracy of multispectral images did not show a noteworthy improvement in the last 15 years. The main problem is that the separability among different LC features can be degraded by the between-class spectral confusion and within-class spectral variation (Yan et al., 2015). Additionally, aerial photos and satellite images are often affected by cloud coverage and weather conditions. Perhaps, the accuracy of LC mapping using only multispectral images has reached its limit; therefore, to increase the accuracy of LC classification, other information in addition to spectral information is needed (Yan et al., 2015).

During the last 20 years, airborne mapping light detection and ranging (LiDAR), also known as airborne laser scanning (ALS), has become one of the primary remote-sensing technologies for analysing the surface of the Earth due to its good capability of three-dimensional (3D) information acquisition (Glennie et. al., 2013). LiDAR is a gauging technique that surveys distance to an object, which can record a set of points that describe the target object. Compared with two-dimension (2D) images, the LiDAR data have the advantages of acquiring more accurate topographic information from the Earth's surface without problems resulting from cloud coverage, weather conditions, and relief displacement (Glennie et. al., 2013). Previous studies have well demonstrated the capability of ALS data in LC classification (e.g. Antonarakis, 2008; Lodha, 2006). Using its 3D spatial information, the ALS data can separate objects that have similar spectral signatures such as parking lots and buildings (Glennie et. al., 2013). Nevertheless, most of the LiDAR sensors record only one channel of pulse. Thus, the fact that single-wavelength ALS data lack spectral information

limits its accuracy for classifying similarly shaped objects in complicated environments. To overcome this limit, the 3D data obtained by ALS are often integrated with spectral information provided by multispectral images.

Combining spectral information of multispectral imageries and 3D spatial information of ALS point clouds has achieved better results of LC classification than using either of them individually. For example, a study, which fused WorldView-2 images with ALS data to classify urban LC, reached an overall accuracy of 91% (Kim & Kim, 2014). Although the multi-sensor fusion technique is a feasible approach to increase LC classification, it requires the multi-sensor datasets to be registered to the same coordinate system with the same spatial resolution and the same collection time (Yan et al., 2015). However, datasets acquired by different systems often have different data formats, projections, spatial resolutions, and collection times, which can introduce errors to the data fusion process. To deal with these problems, additional data pre-processing and calibration steps must be performed to alleviate those problems even though they may introduce additional errors (Yan et al., 2015). However, some errors still cannot be solved by these steps (e.g., errors introduced by different data collection time; Yan et al., 2015).

To solve these problems of data fusions, multispectral LiDAR techniques, which can acquire LiDAR data with multiple channels simultaneously, have been recently developed. The Optech Titan, which contains three active imaging channels at different wavelengths, is the first commercial multispectral airborne active imaging LIDAR sensor in the world (Bakula, 2015). Even though only a few related studies have been conducted (e.g. Teo and Wu, 2017; Zou et al., 2016), the potential of using multispectral ALS technique to map the Earth's surface has been identified. The multispectral ALS data has been proven to be superior to both traditional multispectral optical imagery and typical single-wavelength ALS data for LC classification (Bakula et. al., 2016; Teo and Wu, 2017; Morsy et al., 2017). Thus, it is necessary to seek optimal classification methods for taking full advantages of this new technique.

Recognized by Massachusetts Institute of Technology as one of the ten breakthrough technologies of 2013 (MIT Technology Review, 2013), deep learning has a powerful capability of learning. Recently, it has been widely applied in the fields of artificial intelligence because of the notably reduced cost of computing hardware, improved chip processing capability, and the significant development of the learning algorithms (Deng, 2014). Since deep learning has been shown to be a highly successful tool, whose learning ability sometimes even exceeds humans' (e.g.



AlphaGo; Chen, 2016), it has become the model of choice in many fields including remote sensing (e.g. Papadomanolaki et al., 2016; Zhang et al., 2017a; Tran et al., 2015). As an evolution version of classic machine learning, deep learning has been applied in different kinds of datasets for LC classification such as hyperspectral images (e.g. Kussul et al., 2017; Li et al., 2017). Moreover, deep-learning classification methods are able to acquire higher accuracy than other conventional classification approaches such as the support vector machine (SVM) (Zhong et al., 2018). However, no published research has attempted to use deep-learning methods and multispectral ALS data in combination to improve LC classification accuracy prior to this thesis being written.

## **1.2 Objectives of the Study**

Since both multispectral ALS technique and deep learning networks have shown their superiority in LC classification, this research has proposed an approach that uses deep learning networks with multispectral ALS data to improve LC classification. However, to the best of author's knowledge, since there is no similar research, it is very challenging to build an eligible workflow to train, validate and test deep learning networks using multispectral ALS data with an appropriate data structure. Thus, this thesis mainly aims to establish a workflow for automated pixel-wise classification using multispectral ALS data with a compatible data structure as input and deep learning networks as the employed classification method. In addition, it is desired to test if deep learning networks and multispectral ALS data can improve LC classification accuracy. Some of the specific objectives are:

- (1) to extract appropriate information from the multispectral ALS data and form input data with the most suitable data structure for deep learning networks;
  - (2) to establish and implement deep learning networks that are appropriate for multispectral ALS data classification;
  - (3) to seek an optimal scheme that leads to the highest classification accuracy by assessing and comparing the classification results of the proposed inputs and deep learning networks;
  - (4) to analyse how different information extracted from the multispectral ALS data impacts classification results;
- and (5) to assess how different deep-learning networks can affect the classification results of multispectral ALS data.

### **1.3 Structure of the Thesis**

This thesis consists of six chapters:

Chapter 1 introduces motivations, challenges, objectives and structure of the study.

Chapter 2 presents the multispectral ALS systems' operating principles and components and the deep learning' principles and categories. It also reviews studies related to the multispectral ALS LC classification and related to the deep-learning LC classification.

Chapter 3 provides a description of the study area and datasets used. It also describes the proposed workflow which consists of multispectral ALS data pre-processing, construction of input datasets, data labelling, selection of deep-learning networks, establishment of CNNs, implementation of CNNs, and an accuracy assessment.

Chapter 4 shows results of the study including validation of labelled dataset, determination of hyper-parameters, and the accuracy assessment of the LC classification. The key findings are also discussed in this chapter.

Chapter 5 offers conclusions of the deliverables of the thesis, analyses the limitations and offers recommendations for future studies.

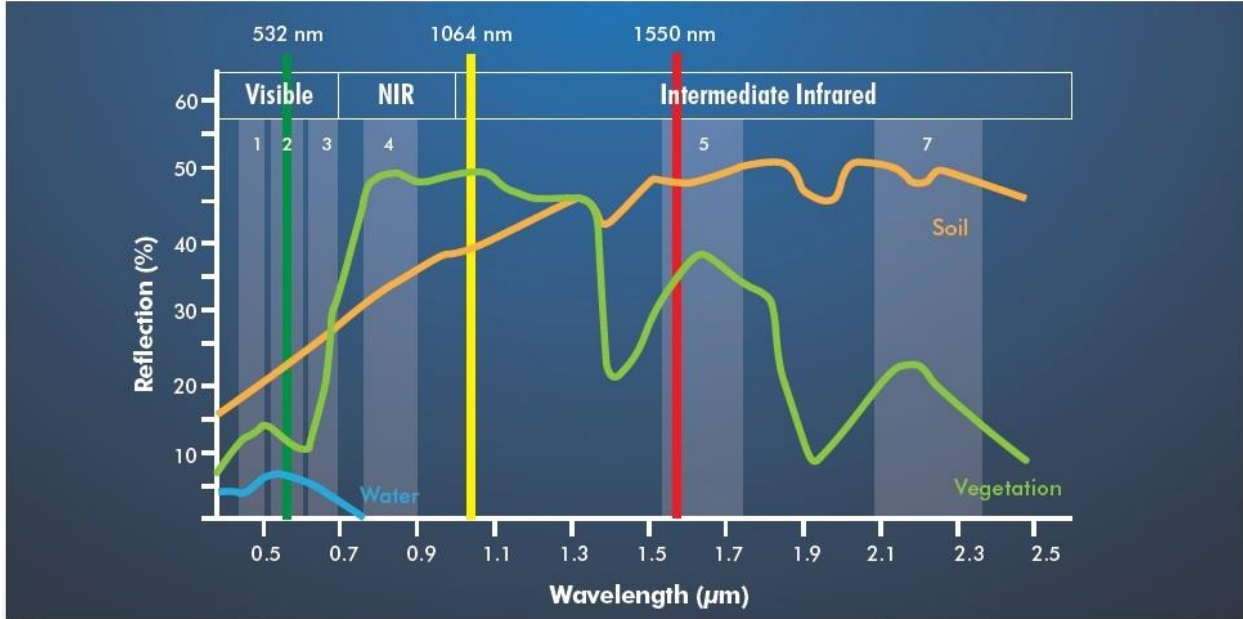
## Chapter 2

### Background and Related Studies

This chapter firstly introduces essential operating principles and components of the typical multispectral ALS system by taking the Teledyne Optech Titan multispectral ALS system as an example. Since this study is pioneering, there is no similar research. Therefore, studies related to LC classifications for multispectral ALS data using other methods and research related to deep-learning LC classifications applied for other datasets are reviewed.

#### 2.1 Multispectral Airborne Laser Scanning System

In addition to a typical ALS system mentioned in Chapter 1, a multispectral ALS system is capable of gathering discrete and full-waveform data from several different active imaging channels of different wavelengths, which may provide a better mapping ability of complicated environments. The first commercial multispectral airborne active imaging LIDAR sensor in the world is the Teledyne Optech Titan multispectral ALS system, which contains three active imaging channels of different wavelengths: 1550 nm (shortwave infrared, SWIR), 1064 nm (near infrared, NIR), and 532 nm (green, G), respectively. The three channels generate laser pulses with separate forward angles to produce independent scan lines. As shown in Figure 2.1, green vegetation is strongly reflective in the NIR spectrum, and slightly reflective in the visible G spectrum. Soil tends to reflect most at the SWIR band but lowest at the green band. Electromagnetic waves are mostly absorbed at the water surface in the NIR and SWIR spectrum. Thus, the three scanning frequencies provided by the Teledyne Optech Titan make it possible to acquire various spectral responses of different materials and to obtain diverse information about the surface of the Earth (Bakuła, 2015). Detailed specifications of the Teledyne Optech Titan are listed in Table 2.1. This section introduces the multispectral ALS System in terms of its components, direct geo-referencing theory, basic principles, and multi-wavelength intensity maps.



**Figure 2.1** Titan laser channels with spectral signatures for selected objects (Teledyne Optech Titan, 2015)

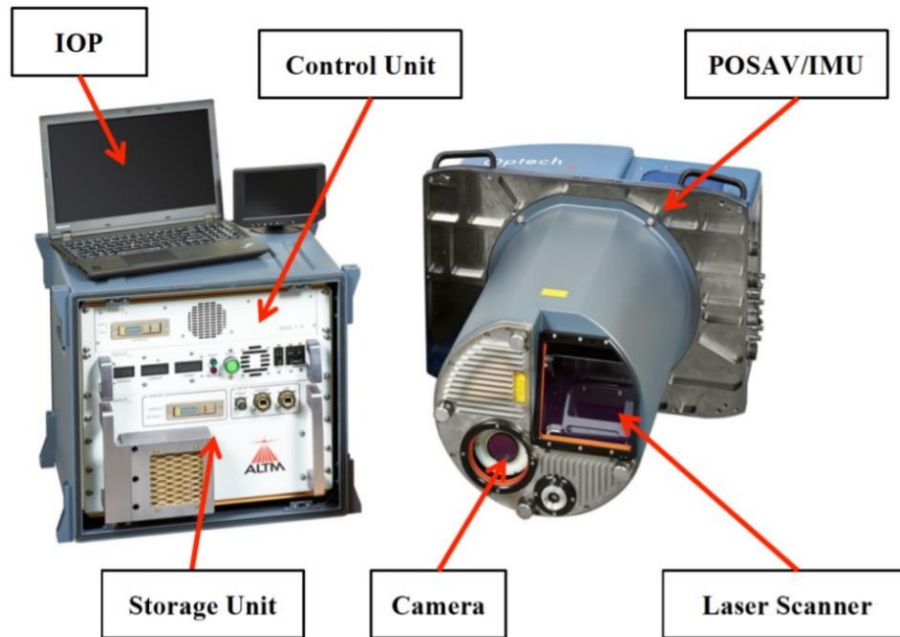
**Table 2.1** Specifications of Titan

Parameter	Specification
Wavelengths	Channel 1: 1550 nm (shortwave infrared, SWIR) Channel 2: 1064 nm (near infrared, NIR) Channel 3: 532 nm (green, G)
Forward Angles	Channel 1: 3.5° Channel 2: 0° Channel 3: 7°
Pulse repetition frequency (PRF)	Programmable; 50 - 300 kHz per channel; 900 kHz in total
Scan Frequency	Programmable; 0 - 210 Hz
Point density	Bathymetric: >15 pts/m <sup>2</sup> Topographic: >45 pts/m <sup>2</sup>
Accuracy	Horizontal: 1/7, 500 × altitude, 1σ Vertical: < 5 - 10 cm, 1σ
Laser range precision	5 < 0.008 m; 1σ

### 2.1.1 Components of a Multispectral ALS System

The components of the Teledyne Optech Titan multispectral ALS system are shown in Figure 2.2. A flight management system, an operator laptop, a digital camera, a laser scanner assembly, a

Global Navigation Satellite System (GNSS), an Inertial Measurement Unit (IMU), and a control and data recording unit are essential parts of a Multispectral ALS system.



**Figure 2.2** Optech Titan system (Teledyne Optech Titan, 2015)

### (1) Flight Management System

A flight management system provides pre-planned flight lines, real-time point display, and real-time survey conditions, which guarantee uncomplicated operation and consistent point distribution. Titan offers a Teledyne Optech's comprehensive flight management system to operators.

### (2) Operator Laptop

The operator laptop builds communications between operators and the control and data recording unit in order to allow operators to set up parameters. Operators can control a multispectral ALS system and monitor the system performance using an operator laptop.

### (3) Digital Camera

A digital camera is often implemented in the fuselage exposed to the ground to provide ancillary information via taking colour images or videos of the study area concurrently with laser scanning. For example, the true-colour information of these images or videos can be used to colorize the points collected by laser scanners to achieve a better visualization. The Titan system also provides a digital camera. However, the digital photos collected by the Titan system are not

available for this study.

#### (4) Laser Scanner

The laser scanner assembly releases continuous laser beams towards the target to capture surfaces of objects and measure the distances to objects. A laser scanner in a multispectral ALS system is set to work in a 2D planar-scanning mode; the third dimension of the collected 2D data can be achieved by moving the aircraft. Different parameters of laser scanners such as field-of-view, range, and scan frequency lead to different quality of collected data. For example, the Titan can achieve a point density of 25 points/m<sup>2</sup> with a flying height of 1000 m, Pulse repetition frequency (PRF) of 900 kHz, field of view (FOV) of 30, and a cruising speed of 60 m/s.

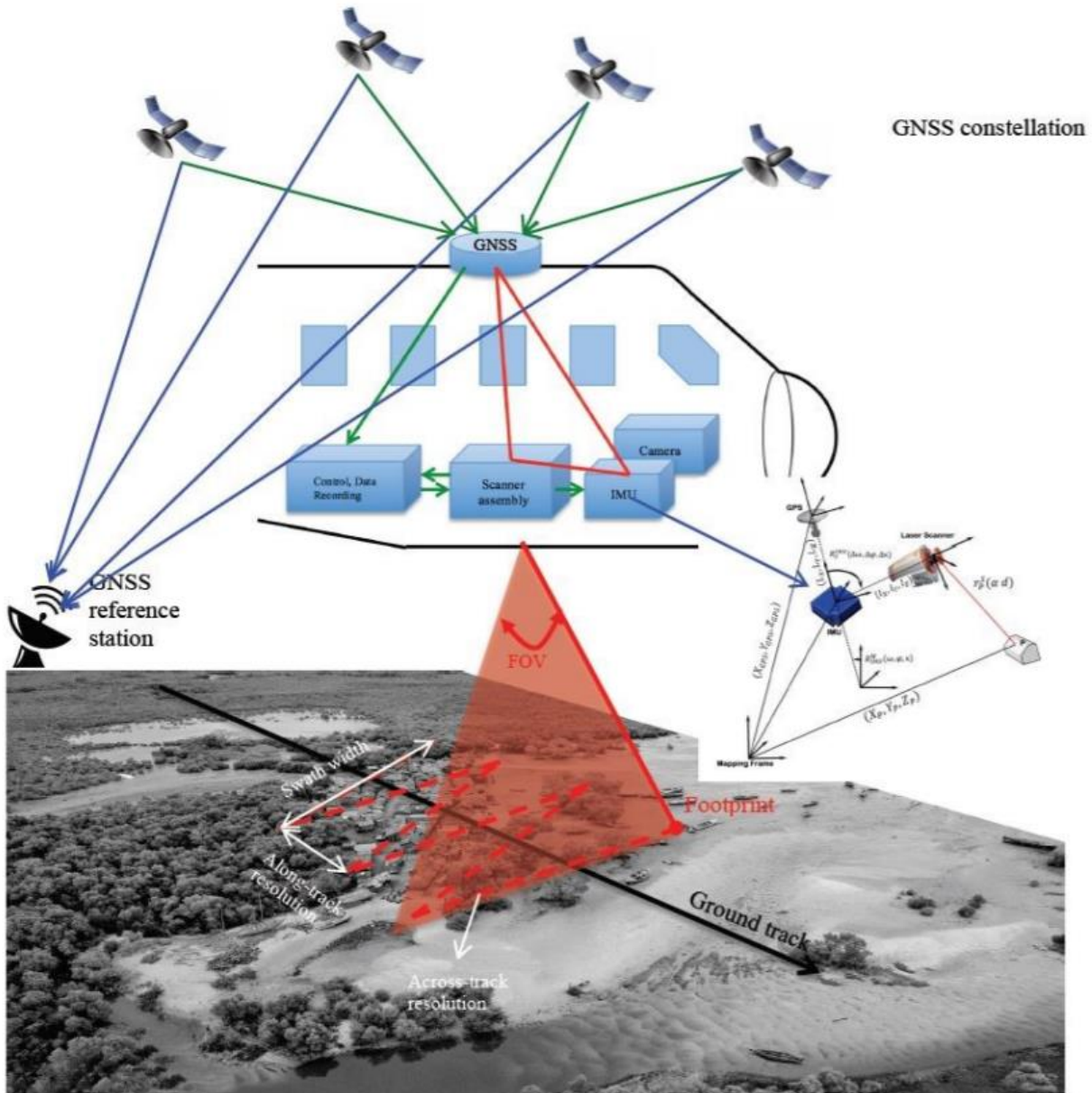
#### (5) GNSS

A GNSS, which is fixed at the top of the aircraft, is the fundamental module of a positioning system. It has the ability to provide centimetre accuracy positions. Integrated with an IMU (see Figure 2.3), a GNSS can provide precise orientation measurement and position information for an airborne sensor. The multispectral ALS system implements these two navigation sensors rather than using one of them alone to exploit the complementary nature of these sensors. In this integrated navigation system, GNSS offers three prime surveys: position, time, and velocity that includes speed and direction. Though GNSS receivers can deliver exceedingly precise position measurements in an open environment, it is practically impossible to receive the signal during a complete survey due to multipath effects and the GNSS outage periods. This limitation can be surmounted by combining the GNSS and IMU data streams.

#### (6) IMU

An IMU sensor, which comprises a microcomputer unit and a set of gyroscopes and accelerometers, can calculate the updated positions and velocities for an initial position and attitude provided by GNSS. Acceleration information provided by IMU facilitates the interpolation of the aircraft position along the GNSS trajectory. Rotation rates recorded by IMU are utilized to determine the orientation of the aircraft. The velocities, positions, and orientations calculation of IMU is autonomous and safe from blockage since no external information is required. Attitude information such as heading, pitch, and roll can be provided by IMU without the assistance of satellite signals; nevertheless, the accuracy of the orientation and position degrades with the time. Thus, the IMU sensor is inappropriate to be used alone but can complement a GNSS; GNSS offers updated location information to the IMU while the IMU provides complementary position

information when the GNSS satellite conditions are poor.



**Figure 2.3** Cooperation principle of GNSS and IMU (Chen et al, 2018)

(7) Control and Data Recording Unit

This unit controls the whole system and records ranging and positioning data collected by the laser scanner, GNSS, and IMU.

**2.1.2 Direct Geo-referencing**

Direct geo-referencing is the determination of time-variable position and orientation parameters for a multispectral ALS system, which produces corresponding coordinate information

for each recorded point. The position ( $p(t)$ ) of a recorded point at a time can be calculated using the following well-defined geo-referencing formula:

$$p(t) = p_N(t) + R_N(t) * R_S(t) * r_L(t) \quad (2.1)$$

where  $p_N(t)$ , in a 3D Cartesian geographic coordinate system, is the location of the scanner at time  $t$ ; the rotation matrix  $R_N(t)$  describes the orientation of the laser channel; the rotation matrix  $R_S(t)$  defines the orientation of the scanning mechanism and process; and  $r_L(t)$  refers to the range measured by the laser scanner at time  $t$  (Hebel and Stilla, 2012).

### 2.1.3 Basic Principles of Multispectral ALS System

The laser scanner mentioned above measures the range of its surroundings. There are two primary methods utilized for range measurements in current ALS systems: time-of-flight (TOF) and phase shift (Vosselman and Maas, 2010). A TOF scanner emits a short laser pulse to the object, and records the time difference between the sent and received pulses to measure the range according to the propagation velocity of light in a medium (Vosselman and Maas, 2010). The following formula can be used to calculate the range:

$$R = \frac{c}{n} * \frac{t}{2} \quad (2.2)$$

where  $c$  is the speed of light;  $t$  is the return time difference; and  $n$  is the refractive index of the medium ( $n \approx 1.00025$  in air) (Vosselman and Maas, 2010). The TOF range measurement approach is applied in the Titan.

Phase-based range measurement in continuous wave modulation is regarded as an indirect form of TOF-based range measurement (Guan et. al., 2016). Phase-based laser scanners record the phase difference between the sent and received backscattered signals of an amplitude modulated continuous wave to measure the range (Puente, 2013). Commonly, the phase-based mode has higher accuracy ranging from sub-millimetre to sub-centimetre and has extremely high data rates, but shorter measuring ranges (Vosselman and Maas, 2010). The following formula can be used to calculate the range for phase-based ranging systems:

$$R = \frac{\varphi}{2\pi} * \frac{\lambda}{2} + \frac{\lambda}{2} * n \quad (2.3)$$

where  $\varphi$  is the phase shift;  $\lambda$  is the modulation wavelength; and  $n$  is the unknown number of full wavelengths between the sensor system and the reflecting object (Puente, 2013).

The swath width  $sw$  of a laser scanner can be specified by the following formula:

$$sw = 2h * \tan\left(\frac{\theta}{2}\right) \quad (2.4)$$



where  $\theta$  represents the scan angle and  $h$  is the height of the aircraft above ground. The scan angle of Titan is programmable between  $0^\circ$  and  $60^\circ$ . This is a nominal formula for nadir scanning over flat terrain.

The width of a laser beam varies with the distance from the laser scanner. Assuming the spot shape (i.e. footprint) to be a circle, the diameter  $D_s$  of the illuminated footprint on the ground can be described by:

$$D_s = 2h * \tan\left(\frac{\gamma}{2}\right) \quad (2.5)$$

where  $h$  is the height of the aircraft above ground and  $\gamma$  is the beam divergence. In Titan, the beam divergence of both Channel 1 and Channel 2 is about 0.35 mrad; and this parameter of Channel 3 is around 0.7 mrad. Thus, the footprint (i.e. diameter on the ground) of a laser beam in Channel 1 and Channel 2 is about 16 cm, and about 32 cm in Channel 3 for a 457m flying height. In addition, the irradiance of a laser beam decreases progressively away from the centre of the beam. In Titan, the irradiance declines to 1/e times of the total irradiance.

#### 2.1.4 Multi-wavelength Intensity Maps

For each measured point, an ALS system also records the strength of the backscattered echo, typically referred to as intensity, besides the spatial information based on TOF measurements. Recorded intensity not only is related with target's reflectance at the given laser wavelength, but also depends on several other factors, such as wetness and roughness of the target surface, environmental effects, data acquisition geometry parameters, and instruments (Bakula, 2015; Ahokas et. al., 2016). Fortunately, intensity calibration and normalization can be applied to reduce the effects of other factors and therefore improve the quality of intensity information (Yan et. al., 2012; Kashani et. al., 2015).

However, since common LiDAR only measures the backscatter at a single and narrow laser wavelength band, the utility of its intensity information has been essentially limited. To break this limitation, a multispectral ALS system provides the intensity information at multiple laser wavelength. For example, Titan can record the backscatter at three wavelengths (i.e. bands): SWIR, NIR, and G. In order to analyse and utilize the intensity information collected at different bands more easily, for each band, the intensity information of 3D point cloud is often converted into a 2D raster imagery which allocates the intensity as the cell value (e.g. Matikainen et. al.,

2016; Bakula et. al., 2016). The selection of the cell size usually relies on the point density of a dataset.

## 2.2 LC Classification for Multispectral ALS Datasets

Since the multispectral ALS technique is still an emerging technology, there are limited studies exploring its feasibility for the LC classification. All studies which tested the accuracy of LC classification using different multispectral ALS datasets are summarized in Table 2.2. Details of these studies are analysed and compared in the following sections.

**Table 2.2** Studies Related to Multispectral ALS Data Classification

Authors	Type	Main Algorithm	Classes	Overall Accuracy	Kappa
Bakula et. al., (2016)	Raster-based	Maximum Likelihood	6	90.9%	0.88
Fernandez-Diaz et. al., (2016)	Raster-based	Maximum Likelihood	5	90.2%	0.87
Morsy et. al., (2017a)	Raster-based	Maximum Likelihood	4	89.9%	0.86
Teo and Wu, (2017)	Object-based	Support Vector Machine	5	96%	0.95
Matikainen et. al. (2016; 2017a; 2017b)	Object-based	Random Forest	6	95.9	0.95
Zou et al. (2016)	Object-based	Decision Tree	9	91.6%	0.89
Wichmann et al. (2015)	Point-based	Progressive TIN Densification; RANSAC-based Segmentation	5	99%	N/A
Morsy et. al., (2017a)	Point-based	Skewness Balancing; Jenks natural breaks optimization	4	92.7%	0.90
Morsy et. al., (2017b)	Point-based	Skewness Balancing; Gaussian Decomposition; Maximum Likelihood	4	95.1%	0.93

### 2.2.1 Potential of Multispectral ALS technique in LC classification

Even though just a few related studies have been published, the potential of using multispectral ALS technique to map the surface of the Earth has been illustrated.

On one hand, it has been proven that multispectral ALS techniques are superior to traditional multispectral optical techniques for LC classification. To explore the superiority of multispectral ALS data compared to typical multispectral optical images for LC classification, Bakula et. al. (2016) selected different inputs of classification: only spectral information recorded on the laser reflectance intensity images, spectral information with elevation data derived from 3D coordinate

of laser points, and spectral information with elevation and textural data derived from granulometric analysis of the point cloud. The result indicates the utilization of elevation information could significantly improve the classification output, especially in the situation that separating objects with distinctive height was required (Bakuła et. al., 2016). Similarly, after comparing classification results when using only spectral information from the three Titan channels with classification results from a combination of the structural images and the intensity images, both Fernandez-Diaz et al. (2016) and Morsy et al. (2017) found that the addition of structural images derived from multispectral ALS data increased the classification accuracy by more than 15%. Furthermore, Teo and Wu (2017) proposed that using a combination of spectral and geometrical features extracted from multispectral ALS point clouds improved the “road” class extraction in urban area by 15.2% when compared to using only spectral features. In addition, compared to passive aerial images, the intensity images have interesting advantages such as a lack of shadows. Thus, it can be concluded that the multispectral ALS point clouds have higher potential for LC classification than typical multispectral optical images

On the other hand, multispectral ALS techniques have been shown to attain better accuracies of LC classification compared to typical single-wavelength ALS techniques. Teo and Wu (2017) concluded the improvement of classification completeness and overall accuracy from single-wavelength to multi-wavelength ALS technique ranged from 1.7% to 42.3% and from 4% to 14%, respectively. The most significant accuracy improvement brought by the multispectral information occurred in the “Soil” class with an improvement of 35.8% (Teo & Wu, 2017). Similarly, in a comparative study, Matikainen et. al. (2017a) stated that using intensity information of only Channel 1 to replace intensity information provided by all three channels resulted in a marked reduction of classification accuracy.

Since multispectral ALS techniques have such a high potential for LC mapping, it is necessary to seek optimal classification methods in order to take full advantages of the dataset.

## **2.2.2 Classification Methods Used for Multispectral ALS Datasets**

It is well known that the accuracy of classification highly relies on the classification algorithms and the information provided by the input; more reliable classification algorithms and more useful input information lead to better classification results. Since multispectral ALS datasets can produce similar data product derived from both typical multispectral optical images (e.g. multi-wavelength intensity images) and ALS point clouds (e.g. height model), most classification algorithms

designed for either remote sensing imagery or ALS point clouds can also be used for multispectral ALS datasets. The problem is how to extract useful information from multispectral ALS point clouds and how to input the appropriate information with an acceptable format into appropriate classification algorithms.

To standardize an acceptable format of multispectral-ALS-derived data products for classification algorithms, a specific classification model type should be identified for each LC classification experiment. Generally, classification models can be categorized into two types: raster-based classification models and object-based classification model. The former classifies the Earth's surface based on the information in each raster cell; while the latter is based on information related to each object, which is a set of similar pixels related to a measure of spectral properties, shape, size, texture, context, and relationship with neighbours as well as super-, and sub-pixels (Weng, 2012). Object-based models usually involve two steps: data segmentation to generate objects and classification of the segmented objects. With regard to LiDAR point clouds, there is a third type of classification model, which classifies the Earth's surface based on points. Different classification model types may lead to different reorganization procedures for points such as rasterization and, therefore, result in different information loss. Furthermore, the type of classification model may limit the use of information extracted from the multispectral ALS point clouds and the selection of classification algorithms. The model types, algorithms, and inputs together affect the accuracy of a LC classification.

#### (1) Maximum Likelihood Classification (MLC)

One of the most widely used classification algorithms is the MLC algorithm, which assigns cells a LC class based on the measure of the highest likelihood. The MLC algorithm is generally implemented to classify LC based on the information in each raster cell. Bakula et. al. (2016) presented an experiment using the raster-based MLC to classify a multispectral ALS point cloud into six classes, achieving an overall accuracy of 91% in the best test. This best attempt integrated multi-wavelength intensity images, elevation data, and textural data as the input. In this attempt, raster cells that belonged to water, trees, and buildings were classified accurately; however, cells which belonged to the classes "sand and gravel" and "asphalt and concrete" were misclassified because it was difficult for the MLC algorithm to distinguish two similar classes from each other when there was a shortage of distinctive features (Bakula et. al. 2016). The raster-based MLC method was also applied by Morsy et al. (2017). Integrating the three raster intensity images with

the DSM raster image, this research obtained an overall accuracy of 89.9%. Similarly, Fernandez-Diaz et al. (2016) implemented a supervised raster-based MLC to categorize a multispectral ALS dataset into five LC classes with best overall accuracy of 90.2%. The best overall accuracy was obtained when structural images and only two intensity images derived from Channel 2 and Channel 3 were used. Addition of an intensity image of Channel 1 decreased the classification accuracy unexpectedly. The authors (Fernandez-Diaz et al., 2016) explained that spectral information provided by Channel 1 increased the within-class variance of the commercial buildings significantly, and therefore increased the correlation between commercial and residential building classes, leading to misclassification of the two classes. To conclude, although all of the above studies used the same raster-based MLC method and acquired satisfactory results, classification accuracy of different classes varied with different inputs. Furthermore, as a parametric classifier, the maximum likelihood classifier assumes that a training sample is normally distributed, which is often not the case. This incorrect assumption can introduce errors when classifying urban landscapes. Therefore, non-parametric methods are preferred for urban LC classification.

## (2) SVM and Random Forest (RF)

Among a number of non-parametric methods, SVM and RF have been proven to be effective for LC classification.

SVM applies optimization algorithms to determine the location of ideal boundaries that can most effectively distinguish between classes (Huang et al., 2002). Although SVM was initially developed for handling binary class problems, it has been extended for multi-class problems (Pal and Mather 2005). In principle, the SVM technique aims to reduce the misclassification errors by locating a hyperplane which splits the dataset into a number of discrete classes (Luque et al, 2013). An object-based SVM classification method was tested by Teo and Wu (2017) to categorize multispectral ALS points into 5 classes. The points were firstly segmented to objects according to heterogeneity index, which combined both attribute and shape factors. Then, a supervised SVM was applied to classify the objects, attaining an overall accuracy of 96%. Although the overall accuracy was remarkable, SVM classification still had a major limitation related to the selection of the kernel function and the setting of proper parameter values since they were decided subjectively by the user; few studies have been conducted concerning the optimal choice of a kernel function and proper settings for corresponding parameters (Petropoulos et al, 2012).

The RF algorithm proposed by Breiman (2001) is based on the random selection of input training data. The RF method is a collection of Decision Trees. A decision tree, which is the predictive model that uses a set of binary rules as nodes to acquire a best solution, can also be used as a LC classification algorithm. An object-based decision tree model was implemented with the multispectral ALS data by Zou et al. (2016) to accomplish a 9-class LC classification. In this study, they produced a pseudo normalized difference vegetation index to improve identification of vegetation classes, reaching an overall accuracy of 91.6%. However, the decision tree algorithm tends to overfit training data, especially when a tree is particularly deep. Random forests mitigate this problem well without substantially increasing errors. To constitute a RF, firstly decision trees are formed by randomly sampling a subset (usually 2/3) of the training data and variables with constant replacement. For each tree, a set of user-defined input features for each subset of training sample are selected to determine the decision criteria at the node. Then, the best split at each node is determined by sampling this subset of features via creating a binary rule (Breiman, 2001). Since this algorithm does not necessitate separate feature selection or feature values normalization, it is appropriate for classifying data with a large number of features (Matikainen et. al., 2017a). Matikainen et al. published several articles (2016; 2017a; 2017b) to discuss the performance of an object-based RF LC classification method. They indicated that this method could achieve an outstanding classification result using multispectral ALS datasets, especially in terms of “Building”, “Tree”, and “Asphalt” classes (accuracy of “Building” = 100%, “Tree” = 97.9%, and “Asphalt” = 97.4%). However, this method led to low completeness and correctness values for “Gravel” class as the number of gravel points was relatively small and the in-class variation was comparatively large (Matikainen et al., 2017a). Furthermore, the large number of trees in this method may make the classification process slow, especially when applied to a large dataset such as a dense multispectral ALS point cloud in a large area.

In the context of classic machine learning algorithms, both SVM and RF provide precise and reliable classification results for multispectral ALS data. Nevertheless, both of them require manually designed features which significantly impact the classification accuracy. This characteristic of classic machine learning algorithms makes them highly user-dependent.

### (3) Other Multi-phase Methods

To reach higher classification results using multispectral ALS datasets, point-based multi-algorithm and multi-phase methods were generated. Wichmann et al. (2015) firstly applied a

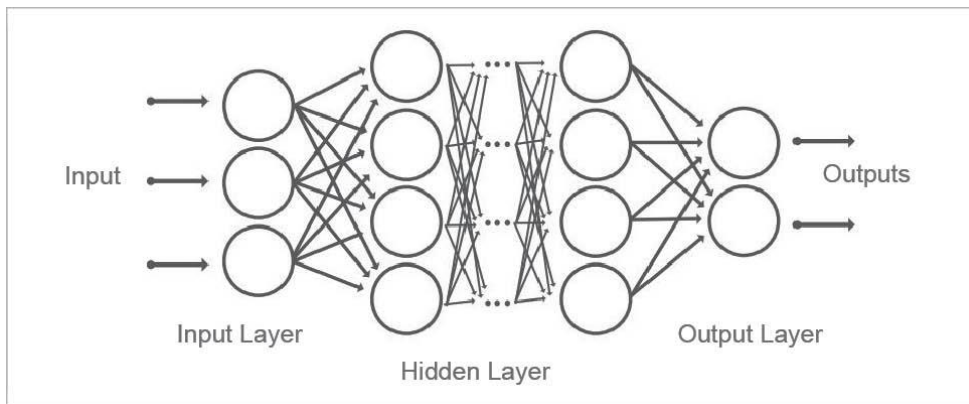
hybrid approach of progressive TIN densification to separate points that belonged to “Ground” class; and then, they combined RANSAC-based point cloud segmentation algorithm and other point cloud features such as eigenvalue-based omnivariance to classify the remaining non-ground points into building and vegetation classes. Even through this automatic multi-phase method achieved an extremely high overall accuracy of about 99%, it did not classify ground points into sub-classes, which made this LC classification not detailed enough. The point-based multi-algorithm and multi-phase methods were also used in two LC classification studies (Morsy et al. 2017a; 2017b). In the first research, Morsy et al. (2017a) initially divided points into non-ground points and ground points based on the skewness balancing algorithm; and then, they applied the Jenks natural breaks optimization method to define threshold of NDVI values and cluster both non-ground points and ground points into detailed classes. The best overall accuracy obtained by this point-based method was 92.7%. To achieve a better classification results, Morsy et al. (2017b) designed another point-based multi-algorithm method for the same multispectral ALS dataset. In this new method, instead of the Jenks natural breaks optimization algorithm, the maximum likelihood algorithm was used based on Gaussian components decomposed by the Gaussian decomposition algorithm to cluster points into detailed classes. This new method obtained an overall accuracy of 95.1%. Although the multi-algorithm and multi-phase methods achieved relatively high accuracy, they were usually designed according to features of a particular multispectral ALS point cloud. The classification accuracy of these methods would significantly vary significantly with the data features such as the type, content, and distribution of LC, which rendered them inappropriate for widespread application.

To conclude, none of the existing classification methods represents an optimal method for classifying multispectral ALS data; most of these methods have serious weaknesses and cannot achieve accuracies higher than 96% in terms of multispectral ALS data classification. With an increasing demand for extremely high accuracy of LC classification, new classification methods should be proposed for multispectral ALS datasets that have high potential on LC mapping.

### **2.3 Deep learning**

Rewarded as one of the ten breakthrough technologies of 2013 by Massachusetts Institute of Technology (MIT Technology Review, 2013), deep learning has been widely applied in the fields of artificial intelligence because of the notably reduced cost of computing hardware, the remarkably improved chip processing capabilities, and the significant developments of the

learning algorithms (Deng, 2014). The theory of deep learning builds on neural networks. A standard neural network consists of numerous linked processors named neurons; each neuron can be activated by either an input environment or weighted connections from previously active neurons (Schmidhuber, 2015). Formed by a mass of neurons with more than two hidden layers (Figure 2.4), deep learning algorithms explore feature representations from data by themselves to learn high-level abstractions in data using hierarchical architectures (Guo et. al., 2016). More specifically, each neuron represents an input value called activation in the input layer and a function in the hidden layer. Each neuron in a hidden layer receives the activations provided by the prior layer, operates the activations based on the function and the given weights, and determines the activations that will be transferred to the next layer. To limit the activations within a specific range, before transferring the activations to the next layer, activation functions such as Sigmoid function and the rectified linear unit (ReLU) function are frequently applied to taper the activations into a specific range. After the last hidden layer yields activations to the output layer, a cost function is implemented to calculate the cost of the method. The aim of the learning process is to find optimal weights which make the neural network show a desired performance by minimizing the cost.



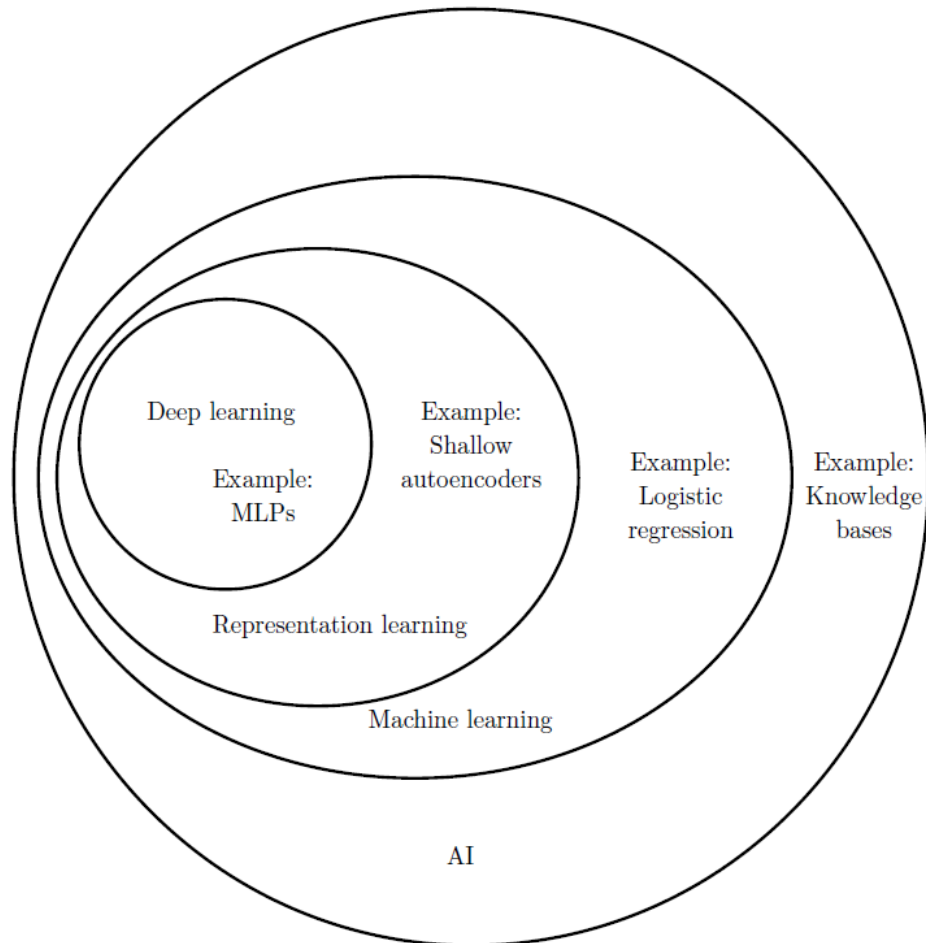
**Figure 2.4** Structure of Standard Deep Neural Network

To allow readers to better understand deep learning and its application in LC classification, this section introduces deep learning through explaining the relationship between deep learning and machine learning, categorizing common deep learning algorithms into four types, and summarizing representative studies that classify remote sensing data using deep learning algorithms.



### 2.3.1 Relationship between Machine Learning and Deep Learning

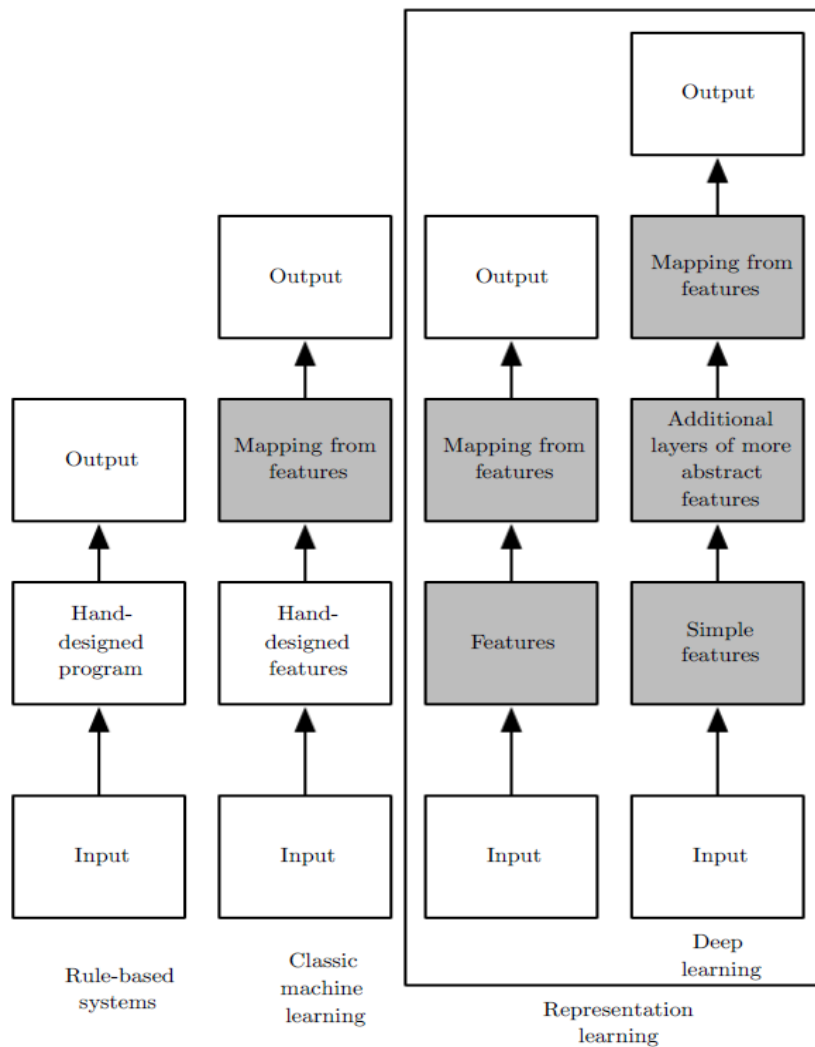
Generally, deep learning is a subfield of representation learning that is a subfield of machine learning. Figure 2.5 describes the relationship among them.



**Figure 2.5** the Relationship among Deep Learning, Machine Learning, Representation Learning, and Artificial Intelligence (Goodfellow et al., 2016)

Deep learning also can be considered as a product of advance and evolution of classic machine learning (see Figure 2.6). Machine learning has the ability to acquire knowledge by extracting patterns from raw data, which allows computers to have the capability of handling problems that involve knowledge of the real world and to make decisions based on human-defined representation of data (Goodfellow et. al., 2016). Thus, for each machine learning algorithm, an appropriate set of features needs to be extracted and provided by designers. The human-defined representation highly impacts the performance of machine learning methods; the optimal set of features will lead to the best result of a machine learning algorithm. Nevertheless, in most instances, it is challenging to determine what features should be extracted. To solve this problem, representation learning has

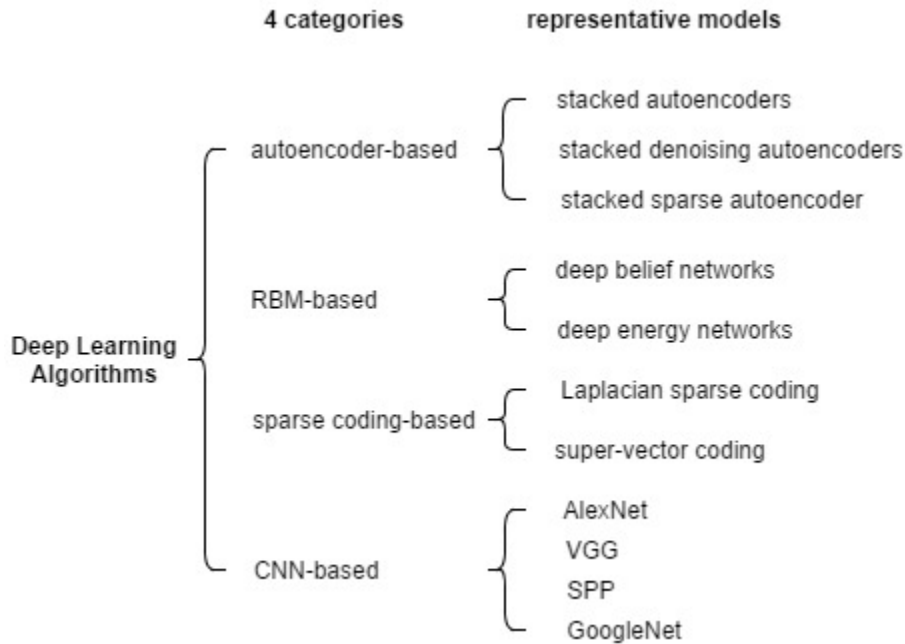
been proposed. Representation learning utilizes the ability of machine learning to “discover not only the mapping from representation to output but also the representation itself” (Goodfellow et al., 2016). The computer-designed representations of representation learning usually lead to significantly better performance than hand-designed representations. A primary target of representation learning is to extract the factors of variation that can explain the data. However, it is also challenging to extract such high-level abstract features from raw data. Thus, deep learning is raised to solve this crucial difficulty of representation learning via introducing simpler representations that can be used to build complex representations (Goodfellow et al., 2016). It decomposes the desired complex mapping to multiple simple mappings described by multiple different layers of the model.



**Figure 2.6** the Relationship among Deep Learning, Machine Learning, Representation Learning, and Rule-based Systems (Goodfellow et al., 2016)

### 2.3.2 Deep Learning Algorithms

Commonly, deep learning algorithms can be divided into four groups based on the elementary method that they apply (Figure 2.7): auto-encoder, Restricted Boltzmann Machine (RBM), sparse coding, and Convolutional Neural Networks (CNN).



**Figure 2.7** Category and Representative Examples of Deep Learning Algorithms

The auto-encoder is often implemented for learning efficient encodings through reconstructing its own inputs rather than predicting a target value for the provided input (Liou et. al., 2014). A typical auto-encoder contains an encoder as well as a decoder to accomplish the reconstruction of inputs. A deep auto-encoder is formed by a series of typical auto-encoders with deterministic network architectures, in which the code learnt from the previous auto-encoder is transferred to the next auto-encoder. Trained with variants of back-propagation, the deep auto-encoder can efficiently obtain more discriminative and characteristic features than the typical auto-encoder (Zhang et. al., 2014). However, deep auto-encoder sometimes can be ineffective especially when first few layers of this model yield errors and deliver them to posterior layers (Guo et. al., 2016).

Different from auto-encoders, which have deterministic network architectures, a RBM is a generative stochastic directionless graphical model designed by Hinton (1986). A classic RBM has a visible layer and a hidden layer; there is no connection within the hidden layer or the input layer (Zhu et. al., 2017). Although the feature representation ability of a single RBM is limited,

efficient deep models such as Deep Belief Networks (DBN) can be composed using RBMs as learning modules. A DBN is a probabilistic generative model applying a greedy learning approach. Nonetheless, as training a DBN involves the training of numerous RBMs, the implementation of the DBN model is computationally costly (Bengio et. al., 2013).

The sparse coding aims to describe the input data through learning a series of elementary functions (Olshausen and Field, 1997). This model has various benefits. Firstly, since the sparse coding utilizes multiple bases, it can positively restructure the descriptor and establish the relationships between similar descriptors identified by sharing bases. Also, noticeable characteristics of the data can be well described due to the sparsity of the sparse coding. Furthermore, data with sparse features can be more linearly separated.

The CNN, the most famous and commonly used deep learning method especially in the field of computer vision, is composed by convolutional layers, pooling layers, and fully connected layers. The CNN has a hierarchical architecture where convolutional layers alternate with pooling layers followed by fully connected layers. The convolutional layer is the main calculating part of a CNN, utilizing numerous kernels to convolve both the input image and the intermediate feature maps to produce multiple feature maps (Guo et. al., 2016). The convolution operation of CNN benefits deep learning process considerably especially in computer vision domains. It reduces the number of parameters due to the weight sharing mechanism, improves the understanding of correlations among neighbouring pixels, and does not change the location of the object (Zeiler, 2013). The pooling layer can be simply regarded to a down-sampling process which combines the outputs of neuron clusters at one layer into a single neuron in the next layer (Ciresan et. al., 2011). To gradually diminish the spatial size of the representation, decrease the number of parameters and calculation, and consequently control overfitting, a pooling layer is usually inserted in-between successive convolutional layers. The fully-connected layer transforms the 2D feature maps into a one-dimensional (1D) feature vector which is similar to a layer in the traditional neural network with about 90% of the parameters in the CNN (Guo et. al., 2016). The vector can be considered as a feature vector for further processing (Girshick et. al., 2014). The training process of CNN consists of forward steps and backward steps. The forward step aims to generate feature maps in each layer based on the current parameters such as weights and bias. The prediction output of this forward process and the given ground truth labels are utilized to calculate the loss cost. Then, a backward step is applied to calculate the gradient of each parameter. These gradients are

used to update all the parameters such as weights and bias in each layer. With these updated parameters the system can proceed to the next forward calculation. The circulation can be stopped when the loss cost of the model or the number of iterations of the forward and backward stages reaches a specified threshold.

Figure 2.8, established by Guo et. al. (2016), summarized the benefits and drawbacks in terms of various properties. To be more specific, ‘Generalization’ indicates whether the method has fine performance in various media and applications. ‘Real-time’ emphasizes the efficiency of the approach. ‘Invariance’ evaluates if the method has robustness in transformation such as scale, rotation, and translation (Guo et. al., 2016). In conclusion, the CNN performs best in automatic feature learning; the auto-encoder and the sparse coding are more efficient in training especially when the training datasets are small; the sparse coding is more suitable for biological studies and more invariant towards transformation.

Properties	CNNs	RBNs	AutoEncoder	Sparse coding
Generalization	Yes	Yes	Yes	Yes
Unsupervised learning	No	Yes	Yes	Yes
Feature learning	Yes	Yes*	Yes*	No
Real-time training	No	No	Yes	Yes
Real-time prediction	Yes	Yes	Yes	Yes
Biological understanding	No	No	No	Yes
Theoretical justification	Yes*	Yes	Yes	Yes
Invariance	Yes*	No	No	Yes
Small training set	Yes*	Yes*	Yes	Yes

*Note: ‘Yes’ indicates that the category does well in the property; otherwise, they will be marked by ‘No’. The ‘Yes\*’ refers to a preliminary or weak ability.*

**Figure 2.8** Comparisons among Four Categories of Deep Learning Algorithms.

### 2.3.3 Deep Learning in LC Classification

Since deep learning has been verified to be a highly successful tool that sometimes its learning ability even exceeds humans’ (e.g. AlphaGo; Chen, 2016), it becomes the model of choice in many fields including remote sensing. As an evolution version of classic machine learning, deep learning

has been applied in different kinds of datasets for LC classification. To better understand how deep learning can improve LC classifications and which type of deep learning algorithms performs best for LC classifications, studies that apply deep learning to classify LC types for other remote-sensing data are referred to as there is no research has studied the feasibility of using deep learning to classify LC types for multispectral ALS data.

#### (1) High-Resolution (HR) Remote Sensing images

In recent years, HR remote sensing images especially very-high-resolution (VHR) images collected from satellites, planes, and unmanned aerial vehicles (UAV) have been widely used for LC classifications. In terms of the HR and VHR images, it has been proven that deep learning methods can achieve remarkable accuracies for LC classification.

Zhang et al. (2017a) proposed two object-based deep learning classification methods involving stacked auto-encoders (SAE) and stacked de-noising auto-encoders (SDAE), respectively. In this study, all the spectral, spatial, and texture features of each object segmented by graph-based minimal-spanning-tree segmentation algorithm were put into either SAE or SDAE network to accomplish classification of the objects (Zhang et al., 2017a). According to the research, the highest accuracy of both SAE-based method and SDAE-based method reached 97% when classifying the VHR images into five classes, which was about 6% higher than the overall accuracy of SVM (Zhang et al., 2017a). Furthermore, according to experiments completed by Papadomanolaki et al. (2016), deep CNN models such as AlexNet and VGG networks achieved better results of LC classification than other deep learning models including SDAE and DBN. For VHR images collected by SAT-4, both AlexNet and VGG networks reached an overall classification accuracy of 99.9%, while the overall accuracy of SDAE and DBN was 80.0% and 81.8%, respectively. Furthermore, Romero et al. (2016) stated that CNN usually performed better in LC classification for VHR images with more training samples and more hidden layers.

#### (2) Hyperspectral Images

The hundreds of narrow spectral bands provided by hyperspectral images facilitate the identification of LC type of each pixel via spectroscopic analysis. Since the hyperspectral imaging procedure is inherently nonlinear (Ghamisi et al., 2016) and deep learning architectures are normally more robust towards the nonlinear processing, the fact that deep learning networks can benefit LC classification of hyperspectral images has been verified recently.

Firstly, auto-encoders were tested for Hyperspectral data classification. The first attempt of

introducing deep learning method into hyperspectral data classification was completed by Chen et al. (2014). They designed a deep learning-based framework which integrated SAE and logistic regression (LR) together to classify hyperspectral images. The authors indicated that the SAE-extracted features were more helpful for LC classification, compared to other traditional methods of feature extraction such as principle component analysis and nonnegative matrix factorization. Since it was a supervised classification method, data labelling was needed, which increased the difficulty of extensive use. To avoid data labelling, Tao et al. (2015) proposed an unsupervised classification method that used the stacked sparse auto-encoder (SSAE) to learn features from unlabelled data. Their experiments demonstrated that features learned by SSAE were more robust for hyperspectral data classification compared to the traditional handcraft features. Both of the auto-encoder-based methods achieved a good overall accuracy of above 97% for hyperspectral data classification.

RBM was also applied in hyperspectral data classification. Chen et al. (2015) presented a new hyperspectral image classification framework based on DBN. Diminishing the feature dimension and presenting a good reconstruction, DBN was an effective method for hyperspectral data classification. Using the DBN-based method, the overall accuracy of hyperspectral image classification reached 99%.

Moreover, CNNs were widely used for hyperspectral data classification. Since the availability of labelled hyperspectral data, supervised CNN has been well studied. Hu et al. (2015) designed a simple 1D CNN with only one convolutional layer, one max-pooling layer, and one fully connected layer to directly classify hyperspectral images. Their experimental results validated that the CNN-based method could achieve higher classification accuracy than traditional methods like SVM. Makantasis et al. (2015) proposed a 2D CNN for hyperspectral data classification which also performed as well as the 1D CNN. To compare the 1D and 2D CNN, Kussul et al. (2017) applied them separately to classify the same dataset. They concluded that the 2D CNN was superior to the 1D CNN in terms of overall accuracy; however, the 2D CNN was more likely to misclassify small objects and lead to overfitting. To solve this problem, Ghamisi et al. (2016) suggested a self-improving 2D CNN-based classification model to iteratively select the most informative bands based on the fractional-order Darwinian particle swarm optimization algorithm. This model was also useful to solve the so-called curse of dimensionality. The challenges of dimensionality could also be avoided by an end-to-end deep learning network proposed by Santara et al. (2017) by

extracting band specific spectral-spatial features. With the increasing importance of spatiotemporal feature, 3D CNN was proposed (Tran et al. 2015). Li et al. (2017) designed a 3D CNN framework to abstract the deep spectral and spatial features effectively, requiring fewer parameters than 1D and 2D CNN methods. The authors indicated that this 3D CNN performed better than other state-of-the-art methods such as SAE, DBN, and 2D CNN. To decrease the dependence on labelling data, studies of unsupervised CNN were also significant. Romero et al. (2016) developed an unsupervised CNN based on a greedy layer-wise fashion; this model also achieved a good classification result.

### (3) Synthetic Aperture Radar (SAR) Images

Deep learning models were also combined with SAR images in particular polarimetric SAR (PolSAR) to analyse LC. Hou et al. (2016) proposed a PolSAR image classification method based on multilayer auto-encoders and super-pixel. In this method, multilayer auto-encoders were utilized to capture the features that could represent abstract concepts contained in PolSAR data. The implementation of multilayer auto-encoders reduced the number of parameters and improved the capability of feature representation and discrimination. The auto-encoder was also applied by Zhang et al. (2016). The authors used a SSAE in their unsupervised classification method to automatically acquire useful features. This method attained good visual coherence.

Besides, the RBM could also benefit LC classification of SAR images. Qin et al. (2017) adopted an RBM as the element to build an adaptive boosting model in order to accomplish object-oriented classification for PolSAR imagery. The experimental results revealed that this RBM-AdaBoost could make full use of the polarimetric information of objects and perform better than the standard RBM and stacked RBM. Formed by RBMs, the DBN was also useful for SAR or PolSAR data classification. Lv et al. (2015) proposed a DBN-based classification approach for PolSAR data, combining the benefits of unsupervised and supervised learning. The authors indicated that the DBN was able to automatically abstract effective contextual features from the PolSAR data to increase the classification accuracy. Zhao et al. (2017) designed discriminant DBN to learn abstract features for SAR data, in which the discriminant features were extracted by implementing ensemble learning.

Furthermore, being the most widely used deep learning network in computer vision, the CNN was also tested for SAR images. Zhou et al. (2016) tailored a four-layer CNN for PolSAR classification to automatically capture hierarchical polarimetric features, achieving an overall



accuracy of 92% in classifying 15 classes. Inspired by the standard CNN, Duan et al. (2017) designed a fresh SAR data classification approach based on convolutional-wavelet neural networks (CWNN) and Markov Random Field to segment the data by patch-by-patch scanning. In the CWNN, a wavelet constrained pooling layer was applied instead of the conventional pooling layer. This novel approach reduced the noise and improved the classification performance. Another novel CNN-based method was proposed by Zhang et al. (2017b). They developed a complex-valued CNN (CV-CNN) for SAR image classification. The experimental results indicated that the CV-CNN could significantly reduce classification error and therefore result in higher overall classification accuracy.

From the above summary of related studies, there are several noticeable points that can be concluded. First of all, generally, deep learning-based classification methods are able to acquire higher accuracy than other conventional classification approaches such as SVM. Secondly, none research has studied on applying deep learning for multispectral ALS data classification to date. Moreover, in terms of either HR remote sensing images, hyperspectral images, or SAR Images, CNNs seem to be most applicable and most commonly used deep-learning models for LC classification.

## **2.4 Chapter Summary**

This chapter summarizes the background and related studies of both multispectral ALS system and deep learning with special emphasis on the LC classification. The essential operating principle and components of the typical multispectral ALS system were introduced via taking the Optech Titan multispectral ALS system as an example. The historical development of multispectral ALS data classification and deep learning networks, studies related to multispectral ALS data classification using other methods, and research related to deep learning methods applied for other datasets were also reviewed. From this chapter, it can be concluded that both multispectral ALS techniques and deep learning networks, especially the CNNs, are theoretically and practically promising in LC classification. However, there is a considerable gap in the cooperation of multispectral ALS technique and deep learning networks for LC classification.

## Chapter 3

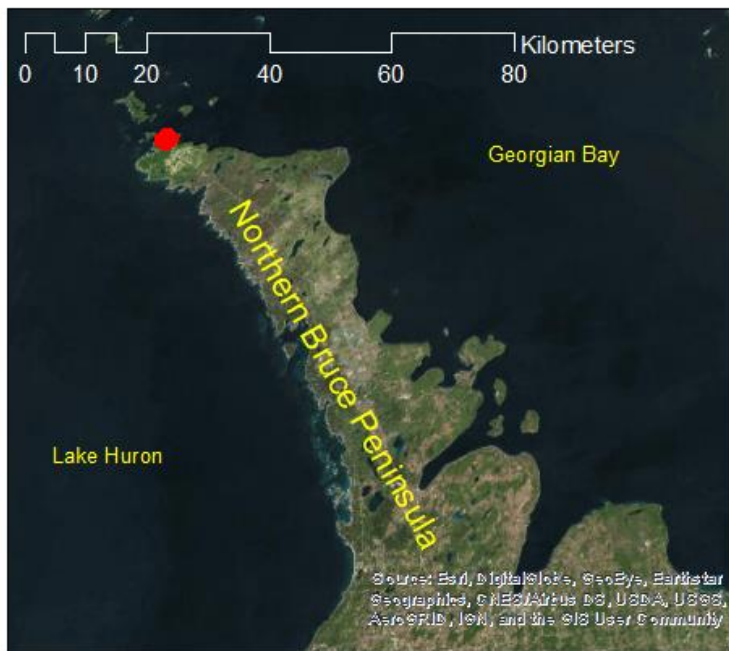
### Deep Learning for Multispectral ALS LC Classification

This chapter provides a description of the study area and the involved datasets. The proposed workflow, which consists of multispectral ALS data pre-processing, construction of input datasets, data labelling, selection of deep-learning networks, establishment of CNNs, implementation of CNNs, and the accuracy assessment, is also described in this chapter.


#### 3.1 Study Area and Datasets

##### 3.1.1 Study Area

The study area is around 1.99 km<sup>2</sup> and is located at the northern tip of Tobermory, Ontario, Canada (see Figure 3.1). Tobermory is a town that is located at the northern tip of the municipality of Northern Bruce Peninsula and can be found between Lake Huron and Georgian Bay. Known as the "freshwater scuba diving capital of the world" (Scuba & H2O Adventure Magazine, 2017), Tobermory is a popular vacation destination. The study area exhibits the heart of Tobermory: the Little Tub Harbour, the harbour village, and the surroundings, which mainly consists of water regions, forest regions, commercial regions, and residential regions. Water regions are primarily located in the northwest of the study area, including the Little Tub Harbour, the Tobermory Harbour, and a part of Big Tub Harbour. Forest regions are at east and southwest of the study area, interspersed with small grasses and bare lands. Commercial regions and residential regions mostly appear in the centre of the study area, surrounding the Little Tub Harbour. Visual inspection of the study area reveals that there are six main LC classes: Water (WAT), Trees (TRE), Bare Land (BAL), Roads (ROD), Buildings (BUD), and Other Impervious Surfaces (OIS) such as parking lots and concrete docks.



**Legend**

 Boundary of Study Area

Coordinate System: NAD 1983 UTM Zone 17N  
 Projection: Transverse Mercator  
 Datum: North American 1983

**Figure 3.1** A map of the study area

### 3.1.2 Datasets

Two datasets were used in the study: a multispectral ALS dataset and an orthophoto dataset.

The multispectral ALS point clouds were collected by Teledyne Optech Titan multispectral ALS system in April of 2015. The Titan system has three spectral channels that each of them collected data simultaneously. The wavelengths of Channels 1, 2 and 3 are 1550nm, 1046 nm, and 564nm, respectively. Each channel has different characteristics, resulting in a rich topographic and bathymetric dataset. The multispectral ALS dataset consisted of ten flight lines. All recorded points were stored separately in thirty LAS files based on the channel and the strip that they belonged to. Each LAS file stored seven attributes of points: the point source ID, scan angles, the flight line edge, the scan direction, the number of returns, return numbers and intensity values. The Titan sensor was installed in an Optech's aircraft that flew at an altitude of about 457 m above ground level during the data collection. The data collection parameters are detailed in Table 3.1. After collection, the data were first pre-processed by Optech. All points were calibrated and all three channels were automatically aligned using the Optech's Lidar Mapping Suite software. To better analyse the study area, the original strips were first cropped according to the boundary of the study area in this thesis; then data pre-processing was done for each strip separately before merging them together. Detailed information of the cropped, pre-processed and merged data was listed in Table 3.2. The average point spacing is defined as the average distance between two adjacent points within a single point cloud while the average point density describes the number of points per m<sup>2</sup> in the study area.

**Table 3.1** Summary of data collection parameters

Parameter	Specification
Wavelengths	Channel 1: 1550 nm Channel 2: 1064 nm Channel 3: 532 nm
PRF	625 kHz in total
FOV	40°
Flight Height	1500 feet (about 457 m)
Flight Speed	140 knots (about 72 m/s)
Number of ALS Strips	10
Number of Returns	4
Number of Points	Channel 1: 310,056,389 Channel 2: 333,541,767 Channel 3: 411,688,529 Total: 1,055,286,685

**Table 3.2** Summary of the cropped, pre-processed and merged data

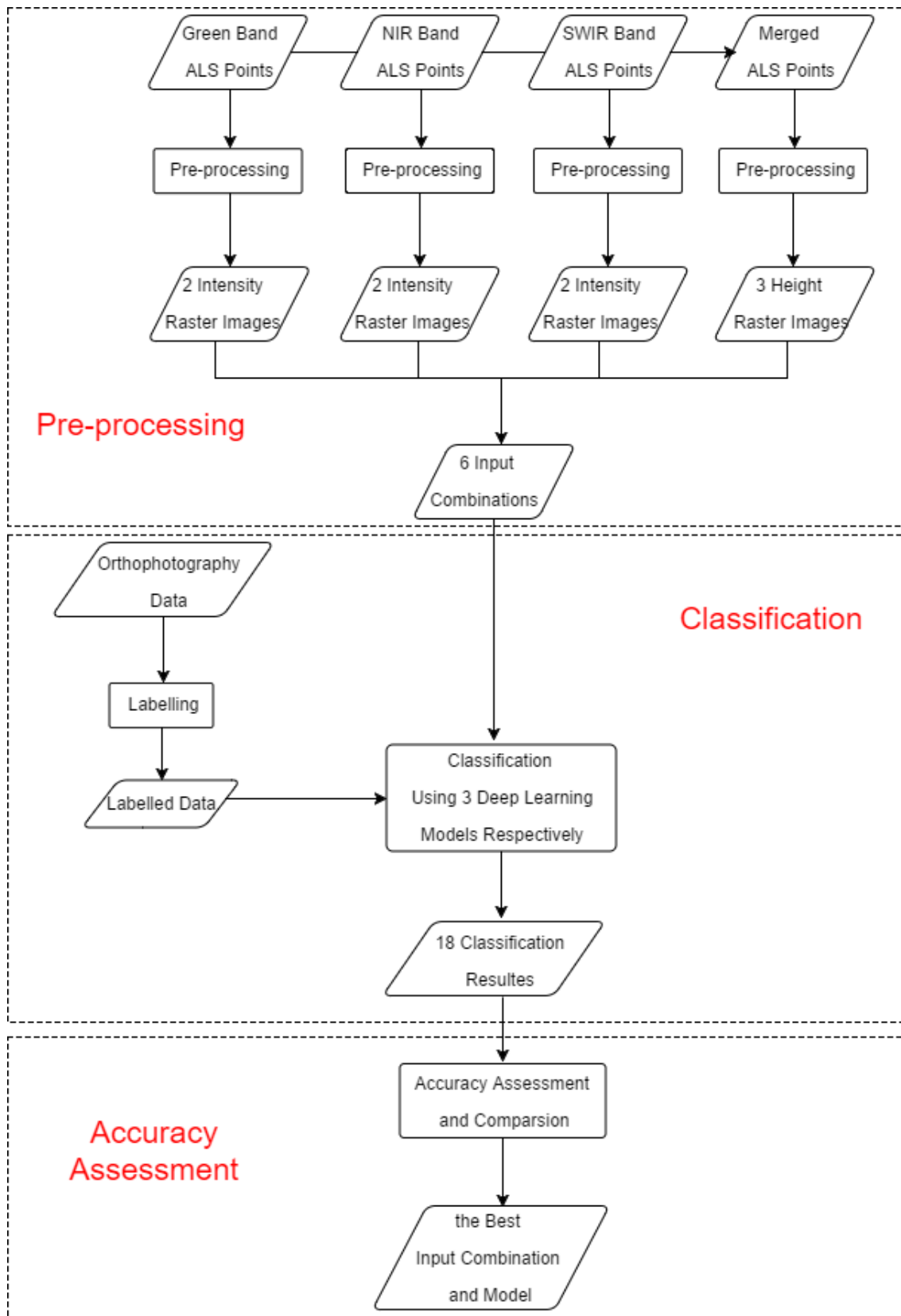
Parameter	Specification
Number of Points	Channel 1: 30,650,352 Channel 2: 32,669,245 Channel 3: 31,015,213 Total: 94,334,810
Average Point Spacing	Channel 1: 0.45 m per point Channel 2: 0.43 m per point Channel 3: 0.44 m per point Total: 0.28 m per point
Average Point Density	Channel 1: 15.4 points per m <sup>2</sup> Channel 2: 16.4 points per m <sup>2</sup> Channel 3: 15.6 points per m <sup>2</sup> Total: 47,4 points per m <sup>2</sup>

The orthophotos were provided by the Southwestern Ontario Orthophotography Project (SWOOP). The SWOOP images were collected using a Leica ADS100 airborne digital sensor between April 12 and May 23, 2015. The collection dates of the orthophotos corresponds well with the collection dates of the multispectral ALS dataset, which ensures that fewer errors were introduced when using these two datasets together. The entire dataset covers an area of approximately 49,167 km<sup>2</sup> in the Southwestern Ontario. This thesis only takes advantage of the SWOOP images that are within the boundary of the study area. The SWOOP images were collected

at 2,377 m above mean terrain to produce 20 cm-resolution orthophotos with four channels (i.e. red, green, blue, and near-infrared). Because of its high spatial resolution and multi-wavelength, this dataset can clearly and precisely depict the Earth's surface of the study area. Thus, the orthophotos were considered as the ground truth for labelling.

### **3.2 Workflow of the Methodology**

Figure 3.2 shows a general workflow of the whole methodology which contained three main parts: the pre-processing, the classification, and the accuracy assessment. The multispectral ALS point clouds were pre-processed at first. A total of nine raster images with different information were generated from the pre-processed point clouds. These images were assembled into six input data combinations. Meanwhile, the labelled dataset was created using the orthophotos as the ground truth. Also, three deep-learning networks were established. Then, each input data combination was used to train and validate each network. This step developed eighteen LC classification models with different parameters to predict LC types for pixels. Therefore, a total of eighteen classification results were produced. Finally, accuracy assessments and comparisons were done for the eighteen classification results to seek an optimal scheme. Details of each part were described in the following sections.



**Figure 3.2** Workflow of the methodology

### 3.3 Multispectral ALS Data Pre-processing

In this study, pre-processing steps could be divided into two groups: normalization steps of data values and establishment steps of input data structure. The normalization steps focused on the value of each LiDAR point; the de-noising and correction processes were applied to make sure the data values could reflect the real surface of the Earth as veritably as possible in Section 3.3.1. The establishment steps aimed to establish appropriate inputs that should have a multi-tiered architecture and abundant information. In order to accomplish the establishment steps, the point clouds were rasterized to images with different information.

#### 3.3.1 Multispectral ALS Data De-noising and Intensity Normalization

The data was firstly pre-processed by Optech. All points were calibrated through geoid correction; and all three channels were automatically aligned using the Optech's Lidar Mapping Suite software. In this study, the original survey strips were cropped according to the boundary of the study area. Then data pre-processing was done for each of the thirty LAS file separately before merging them together.

The further pre-processing of data involved two de-noising steps that were applied separately for each of the thirty LAS files. Firstly, the 99.7% of intensity values (values within three standard deviations) was calculated as a cut-off threshold; points, of which intensity values exceed the threshold, were eliminated. This intensity filtering step removed outliers which had extremely high intensity in each flight strip of each channel. These outliers usually arose from moving objects such as cars and other artificial objects such as building facades (Wichmann et al., 2015). Secondly, a statistical outlier removal filter, provided by CloudCompare v2.6.2 software, was utilized to eliminate isolated outliers which were away from all other points. The filter firstly computed the average distance of a point to its six neighbours, and then compared the average distance of this point with the standard distance which equals to the sum of the total average distances and one standard deviation of the distance. A point was discarded if its average distance was longer than the standard distance.

As mentioned in Section 2.1.4, the values of intensity could be influenced by multiple factors, which reduced the quality of intensity information. Thus, for each LAS file, an intensity range normalization process defined by the following equation was implemented (Matikainen et. al., 2017a).

$$i_{corr} = i_{raw} * \frac{R_i^2}{R_{ref}^2} \quad (3.1)$$



where  $i_{corr}$  is the range-corrected intensity,  $i_{raw}$  is the original intensity,  $R_i$  is the flight-to-target range of the point, and  $R_{ref}$  is the average flying height ( $R_{ref}=457$  m).

### 3.3.2 Multispectral ALS-derived Intensity Imagery

After the outlier removal and intensity correction steps, the thirty LAS files were merged into three point clouds based on the channel that they belonged to. Then, the three point clouds at different laser wavelength bands were projected to the 2D horizontal plane and rasterized into three intensity images. The size of raster cells should be set up to a value that is several times larger than the average point spacing to fill most of voids in the data but is small enough to identify details (ArcGIS Desktop Help, n.d.). According to experience, a reasonable size is two to four times the point spacing (ArcGIS Desktop Help, n.d.). Based on the average point spacing of the dataset calculated in Section 3.1.2 (Channel 1: 0.45 m per point; Channel 2: 0.43 m per point; Channel 3: 0.44 m per point), 0.8m, 1m, 1.2m, and 1.5m was tested as the cell size, respectively. After comparing voids of the generated result maps, ground resolution of the raster images was established at 1 m.

In an intensity image, a cell held an intensity value. With regard to a cell that contained more than one points, the cell value was defined as the distance-weighted average intensity value of all points within the cell. The distance-weighted average value was calculated using the following formula:

$$u(x) = \begin{cases} \frac{\sum_{i=1}^N w_i(x)u_i}{\sum_{i=1}^N w_i(x)}, & \text{if } d(x, x_i) \neq 0 \text{ for all } i \\ u_i, & \text{if } d(x, x_i) = 0 \text{ for some } i \end{cases} \quad (3.2)$$

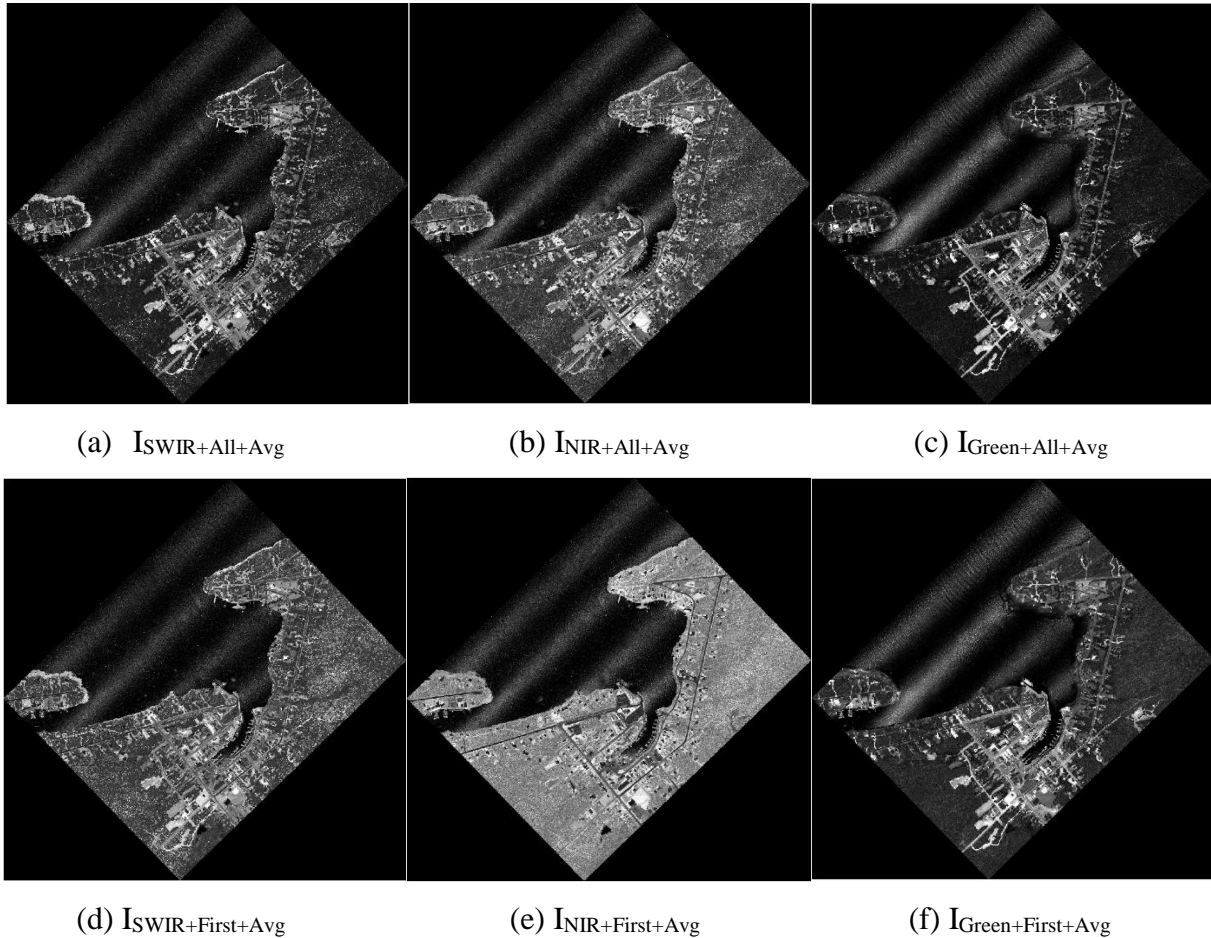
$$w_i(x) = \frac{1}{d(x, x_i)} \quad (3.3)$$

where  $u(x)$  is the final value of a given pixel at the central point  $x$  based on values of all points within the pixel samples  $u_i = u(x_i)$  for  $i = 1, 2, \dots, N$ ;  $d$  is the given distance from the known point  $x_i$  to the unknown point  $x$ ;  $w_i$  is the weight for a point  $x_i$ .

For a cell that contained no points, linear interpolation was used because of its high computational efficiency to compute the average of intensity values of eight neighbouring cells. This simple linear interpolation could provide very good results in the presence of small voids; however, it could be less realistic for big voids (ArcGIS Desktop Help, n.d.). Therefore, only small voids were filled by linear interpolation; the remaining empty areas were filled by a user-defined value. Since visual inspection of these big voids revealed that such voids occur on extensive water

surface, the user-defined value was set up to the average intensity value of water.

To explore the most appropriate input dataset for multispectral ALS classification using deep learning methods, the steps of generating intensity images were executed twice. In the first execution, all returns of a single pulse from a laser were used to produce three intensity images at different laser wavelength bands (named as  $I_{\text{Green+All+Avg}}$ ,  $I_{\text{NIR+All+Avg}}$  and  $I_{\text{SWIR+All+Avg}}$ , respectively). In the second execution, only the first return of each pulse was considered to produce three intensity images (named as  $I_{\text{Green+First+Avg}}$ ,  $I_{\text{NIR+First+Avg}}$  and  $I_{\text{SWIR+First+Avg}}$ , respectively). The average intensity values of all returns reflected the content of objects while the average intensity values of the first returns described the top surface of these objects. Therefore, total six intensity images were generated (see Figure 3.3).

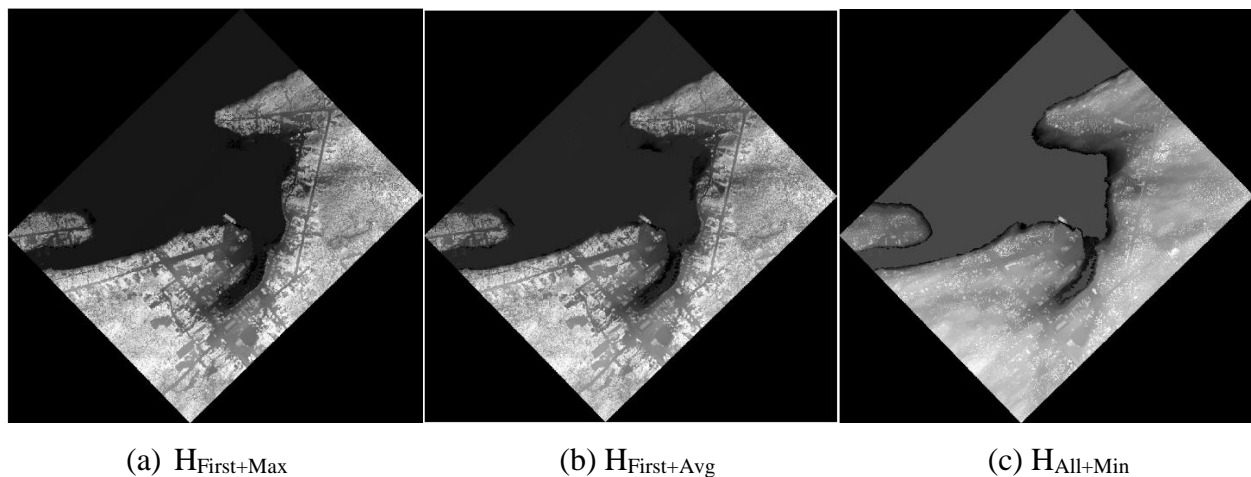


**Figure 3.3** Multispectral ALS on the study area

### 3.3.3 Multispectral ALS-derived Height Imagery

To generate a height image, three point clouds at different laser wavelength bands were merged together to produce a complete point cloud of the study area. This point cloud contains all non-noise points and describes the study area in its entirety. Similar to generating the intensity images, a height image was produced by projecting and rasterizing. Since the height model primarily concerned the top surface of the multispectral ALS point cloud instead of the content, only the first return of each pulse was used to produce the height imagery. The ground resolution of the height imagery was set up to 1 m to match the resolution of the intensity images. In the height imagery, each cell held a height value. For a cell that contained more than one points, the cell value was assigned as the highest height value of all points within the cell. For a cell that contained no point, linear interpolation was used to compute the average of height values of eight neighbouring cells. Similar to the linear interpolation used for the creating intensity images, it was only used to filled small voids; a user-defined value was specified to fill the big voids. Since visual inspection of these big voids revealed that such voids occur on extensive water surface, the user-defined value was set up to the average height value of water pixels.

To provide more information related to height, the steps for generating a height image were repeated twice for the average height of first returns and the minimum height of all returns, respectively. Thus, three height images were generated in total, named as  $H_{\text{First+Max}}$ ,  $H_{\text{First+Avg}}$ , and  $H_{\text{All+Min}}$ , respectively (see Figure 3.4).

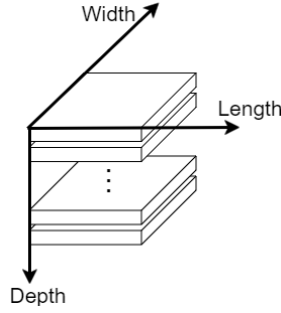


**Figure 3.4** Multispectral ALS height models on the study area

### 3.3.4 Establishment of Input Datasets

Since there was no documentation about how to apply deep learning algorithms to achieve multispectral ALS classification, an appropriate input data structure needed to be identified in this study to not only could reserve adequate information of the original multispectral ALS point clouds but also could facilitate deep-learning computation. Under these circumstances, a multi-tiered raster-based architecture was considered to be the best choice due to the following reasons. Firstly, raster-based images could retain relative position information of raster cells to the maximum extend, which would benefit the LC classification. Moreover, a raster image could store a type of information extracted from the original point clouds. A multi-tiered architecture not only ensured independence of each layer but also make vertical computation among different layers at the same 2D position possible. Therefore, the multi-tiered raster-based architecture was selected in this study.

As shown in Figure 3.5, the input dataset was a stack of several raster images with same length, width and cell size. The length and width were the same as those of the study area. The depth depended on the number of raster images that the input dataset had. In this research, a total of nine raster images were generated from the original multispectral ALS dataset: three all-return average-value intensity images of the three channels, respectively, three first-return average-value intensity images of three Channels, respectively, a first-return maximum-value height image, an all-return average-value height image and an all-return minimum-value height image. These layers were assembled into a variety of combinations as listed in Table 3.3. The Combination 1 was the most classic input dataset of rasterized multispectral ALS data, offering comprehensive spectral information of each channel and general information of the height model. This combination was set as a standard that would be used to compare with other combinations. Removing the height information from Combination 1, Combination 2 provided the spectral information that could be extracted from traditional multispectral optical images. Deleting the spectral information of Green and SWIR bands from Combination 1, Combination 3 can be acquired to simulate the typical ALS data. The Combination 4 was generated by adding the spectral information of first returns, which described the top surface of land objects, to Combination .1 In the Combination 5, more height information was added to Combination 1. Combination 6 combined all of the nine extracted information layers. All of these six combinations would be used as input datasets to train, validate, and test deep learning networks.



**Figure 3.5** Multi-tiered architecture of input datasets

**Table 3.3** Content of input combinations

Combination	Content
1	$I_{\text{Green+All+Avg}} + I_{\text{NIR+All+Avg}} + I_{\text{SWIR+All+Avg}} + H_{\text{First+Max}}$
2	$I_{\text{Green+All+Avg}} + I_{\text{NIR+All+Avg}} + I_{\text{SWIR+All+Avg}}$
3	$I_{\text{NIR+All+Avg}} + H_{\text{First+Max}}$
4	$I_{\text{Green+All+Avg}} + I_{\text{NIR+All+Avg}} + I_{\text{SWIR+All+Avg}} + I_{\text{Green+First+Avg}} + I_{\text{NIR+First+Avg}} + I_{\text{SWIR+First+Avg}} + H_{\text{First+Max}}$
5	$I_{\text{Green+All+Avg}} + I_{\text{NIR+All+Avg}} + I_{\text{SWIR+All+Avg}} + H_{\text{First+Max}} + H_{\text{First+Avg}} + H_{\text{All+Min}}$
6	$I_{\text{Green+All+Avg}} + I_{\text{NIR+All+Avg}} + I_{\text{SWIR+All+Avg}} + I_{\text{Green+First+Avg}} + I_{\text{NIR+First+Avg}} + I_{\text{SWIR+First+Avg}} + H_{\text{First+Max}} + H_{\text{First+Avg}} + H_{\text{All+Min}}$

















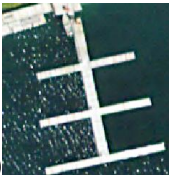

### 3.4 Labelling

To train a network and use the trained model to predict a LC class for each pixel in testing data, a label for each pixel in the training data is needed as ground truth. To calculate the accuracy of predict results, a label of each pixel in testing data is also needed as truth value. Therefore, a labelled dataset that explained the LC type for each pixel within the study area was required. In this study, the labelled dataset of the study area was created. In the labelling process, a blank raster image whose length, width, cell size and coordinate were exactly the same as those of previously generated intensity images and height images was firstly created. Each cell of the blank raster was then given a LC class manually according to the reference map, which was based on the orthophotos of the study area.

As introduced in Section 3.3.1, the study area contains water and various LC features on the ground. To define LC classes in this study, the primary LC classes in the study area and their availability to be labelled were considered. Finally, six major types were selected: WAT, TRE, BAL, ROD, BUD, and OIS. Detailed examples of each class are listed in Table 3.4. When a cell

contained more than one type of LC, this cell was cut to 25 sub-cells based on the 20cm-resolution reference map. Each sub-cell was labelled with a LC type separately. LC type of this cell was identified as the type that occurred most often within the cell.

**Table 3.4** LC types and examples

LC Types	Examples	Illustrations
WAT	(1) Open water (2) Harbours (3) Small lake	(1)  (2)  (3) 
TRE	(1) Multiple (2) Single	(1)  (2) 
BUD	(1) Commercial (2) Residential (3) Small shed	(1)  (2)  (3) 
ROD	(1) Straight road (2) Crossroad	(1)  (2) 
BAL	(1) Sand (2) Rocky area (3) Grass	(1)  (2)  (3) 
OIS	(1) Parking lot (2) Concrete open area (3) Pathway (4) Concrete docks (5) Boats	(1)  (2)  (3)  (4)  (5) 

### 3.5 Selection of Deep-learning Networks

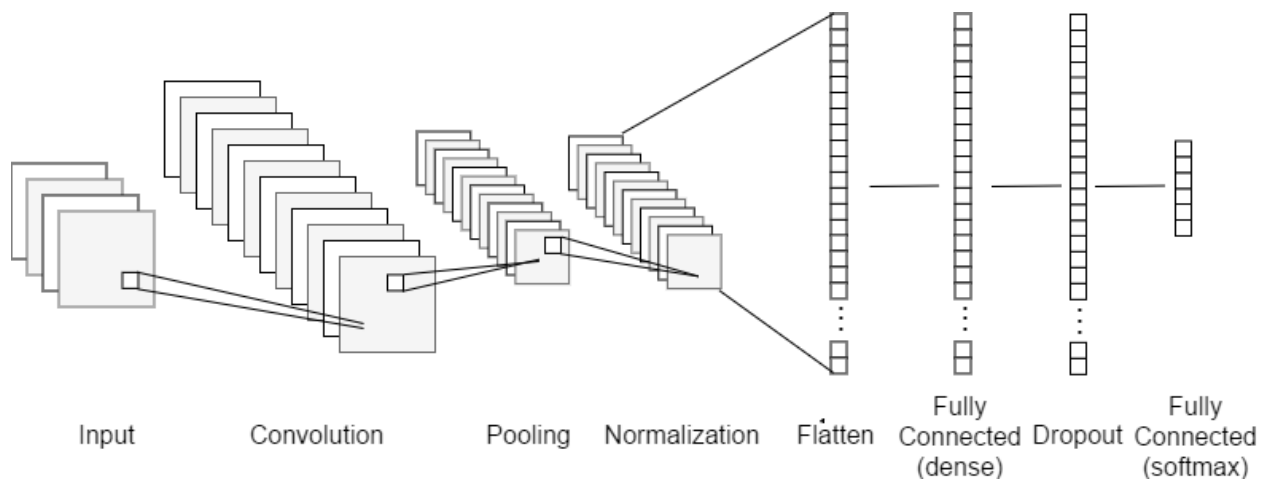
According to Section 2.3.2, CNN-based networks are the most appropriate for LC classification. Firstly, the CNN-based networks have smaller number of parameters, leading to

higher efficiency of computation and training due to the weight sharing mechanism. Secondly, CNN-based networks have higher capability of automatic feature learning compared to auto-encoder-based, RBM-based, and sparse-coding-based networks. Moreover, in CNNs the utilization of kernels improves the understanding of correlations among neighbouring pixels. Furthermore, according to Section 2.3.3, CNN-based networks generally perform best for LC classification, no matter which kind of input data are used (e.g. VHR remote sensing images, hyperspectral images, and SAR data). Thus, CNNs are selected for this thesis.

CNNs can be simply divided into three categories based on the number of the dimension of the convolutional layers, which is the main calculating portion of CNNs. They are 1D CNNs, 2D CNNs, and 3D CNNs. To better understand CNNs and analyse differences of these three types of CNNs, three networks would be established: a 1D CNN, a 2D CNN, and a 3D CNN.

### 3.6 Proposed CNNs

As introduced in Section 2.3.2, CNNs are generally composed by convolutional layers, pooling layers, and fully connected layers with a hierarchical architecture where convolutional layers alternate with pooling layers followed by fully connected layers. In this study, each proposed CNN has seven hidden layers including a convolutional layer, a pooling layer, two fully connected layers, and three other functional layers (see Figure 3.6). These CNNs were established using the scripts shown in Figure 3.7.



**Figure 3.6** Structure of CNNs

```

model = Sequential()
model.add(Conv1D(num_kernels, kernel_size=kernel_size,
                activation='relu',
                padding='same',
                input_shape=input_shape))
model.add(MaxPooling1D(pool_size=pool_size,
                       input_shape=input_shape))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(num_dense, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

```

(a) 1D CNN

```

model = Sequential()
model.add(Conv2D(num_kernels, kernel_size=kernel_size,
                activation='relu',
                padding='same',
                input_shape=input_shape))
model.add(MaxPooling2D(pool_size=pool_size,
                       input_shape=input_shape))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(num_dense, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

```

(b) 2D CNN

```

model = Sequential()
model.add(Conv3D(num_kernels, kernel_size=kernel_size,
                activation='relu',
                padding='same',
                input_shape=input_shape))
model.add(MaxPooling3D(pool_size=pool_size,
                       input_shape=input_shape))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(num_dense, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

```

(c) 3D CNN

**Figure 3.7** Establishment of CNNs

### 3.6.1 Convolutional Layers

Each proposed CNN has one convolutional layer. A convolutional layer, which uses kernels to convolve input data to produce multiple feature maps, is the main calculating part of a CNN. Convolutional calculations are carried out based on the following formula:

$$y(n_1, n_2, \dots, n_m) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \dots \sum_{k_m=-\infty}^{\infty} x(k_1, k_2, \dots, k_m) f(n_1 - k_1, n_2 - k_2, \dots, n_m - k_m) \quad (3.4)$$



where  $n$  is the serial number of output  $y$ ;  $k$  is the serial number of input  $x$ ;  $(n - k)$  is the serial number of filter  $f$ ;  $m$  is the dimension of the convolution.

In the 1D CNN,  $m$  equals to one, meaning that kernels in this model are 1D filters that move only along one direction. To predict a LC class for each pixel in this study, the depth direction was selected as the convolution direction. Thus, this convolution only focused on learning the relationships among provided information layers of each pixel. In the 2D CNN,  $m$  equals to two, signifying that kernels in this model are 2D filters that move along two directions: the width and the length directions. Therefore, the 2D convolution paid more attention on correlations among pixels. In the 3D CNN,  $m$  equals to three, indicating that kernels in this model are 3D filters. These kernels move along all of the three directions, so both relationships among provided information layers of each pixel and correlations among pixels were well considered in this convolution.

An activation function was used in each convolution layer to ensure the nonlinearity. In each established CNN, ReLU was selected as the activation function because it has fewer vanishing gradient and allows for a faster and more effective training compared to other activation functions such as Sigmoid and TanH (Xu et. al., 2015). To ensure the output has the same length as the original input, 'Same' padding strategy was also implemented in each convolutional calculation.

### **3.6.2 Pooling Layers**

A pooling layer was contained in each proposed CNN. The pooling layer is a down-sampling process which combines output pixels of the convolutional layer into a single pixel using a specified window. The window is 1D, 2D and 3D in 1D, 2D and 3D CNNs, respectively. To reserve the strongest features, max pooling layers which choose the maximum value for the cluster of pixels were applied.

### **3.6.3 Fully Connected Layers**

A fully-connected layer generates a 1D feature vector for further processing. Two fully connected layers were applied in each CNN. The former is a dense layer, which performs coarse classification on the features extracted by the convolutional layer and down-sampled by the pooling layer. The latter, a logits layer, is the final layer in each model, returning the raw values for predictions. Since there were six LC classes, six nodes were designed for this layer. The output value of each node describes the possibility of a LC type that the input pixel belongs to. Thus, this layer provides relative measurements of how likely it is that the input pixel falls into each target class.

### 3.6.4 Other Functional Layers

In each model, three functional layers including a batch normalization layer, a flatten layer, and a dropout layer were added. The batch normalization layer normalizes activations of the previous layer at each batch to keep the mean activation close to 0 and the activation standard deviation close to 1. Adding this layer to a CNN can increase the speed of convergence, reduce overfitting, decrease the insensitivity of initial weights, and allow for higher learning rates (Ioffe and Szegedy, 2015). To prepare for calculations in the following fully connected layers, the flatten layer was added to flatten the input from multi-dimension to one-dimension. The Dropout layer randomly sets a fraction rate of input units to 0 to prevent complex co-adaptations on training data. In such a way, the overfitting can be effectively avoided (Srivastava et. al., 2014).

### 3.6.5 Involved Hyper-parameters

In each deep-learning network, hyper-parameters, whose values cannot be estimated from data, are used to help estimate model parameters. Values of hyper-parameters which significantly impact values of model parameters derived via training process are usually specified by the developer. Since the best values of hyper-parameters are unknown, it is important to test various values and to select the most appropriate values, which can achieve relative higher accuracy and efficiency. In this research, each hyper-parameter was tested separately, keeping all other hyper-parameters constant. As listed in Table 3.5, there were five key hyper-parameters involved in the establishment of each CNN. As the control variate method was used, each hyper-parameter is required to be initialized.

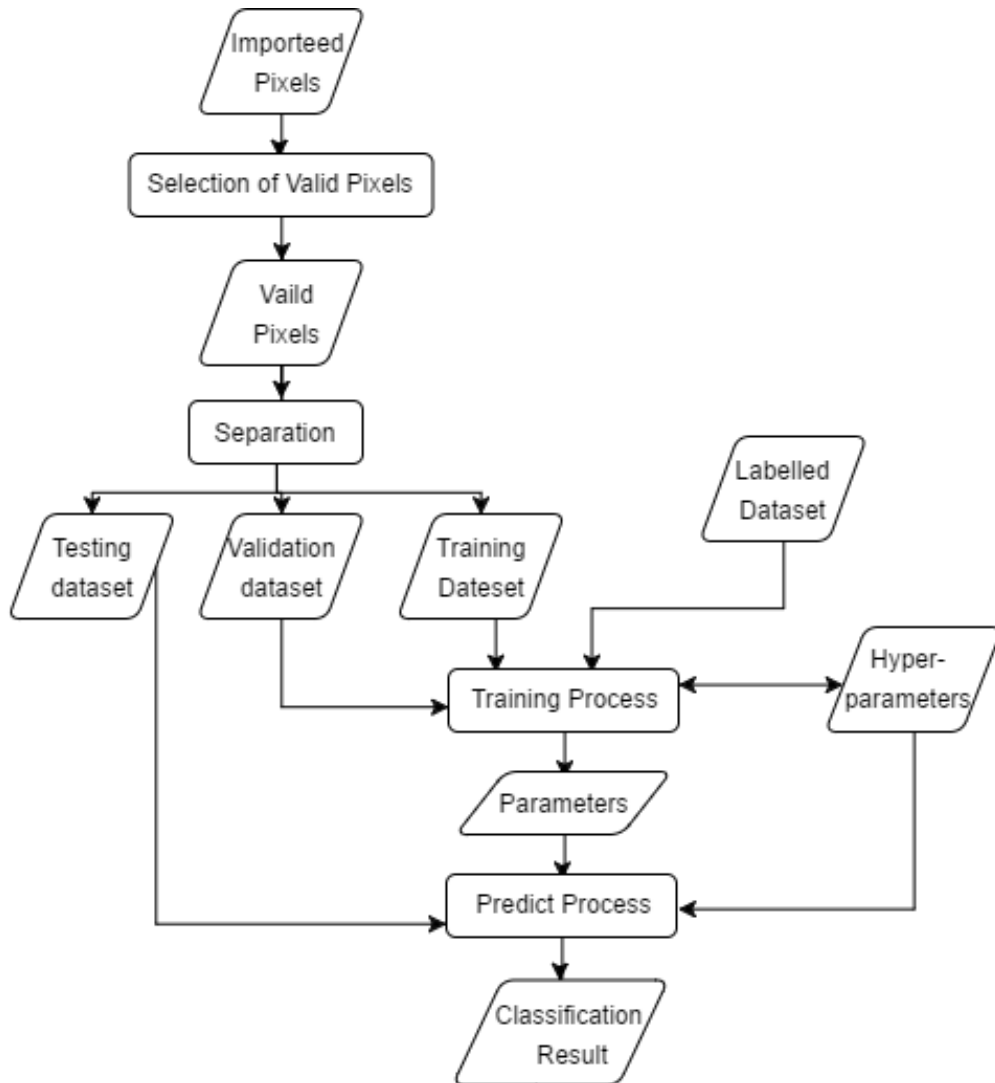
**Table 3.5** Hyper-parameters involved in the establishment of models

Hyper-Parameters	Implemented Layers	Initial Value		
		1D CNN	2D CNN	3D CNN
Shape of Input Unit	Convolutional Layers	(1, 1, depth)	(7, 7, depth)	(7, 7, depth)
Number of Kernels	Convolutional Layers	128	128	128
Size of Kernels	Convolutional Layers	3	(3, 3)	(3, 3, 3)
Size of Pooling Windows	Pooling Layers	2	(2, 2)	(2, 2, 2)
Units of Dense	the First Fully Connected Layers	1024	1024	1024

## 3.7 Implementation of the Proposed CNNs

The implementation procedures, shown in Figure, were used for all of the eighteen classification models. From the very beginning, input datasets, introduced in Section 3.3.4, were

imported into the proposed networks and separated into training, validating and testing data, respectively. To predict LC type for each pixel, every CNN was called twice for two core processes: a training process and a predict process. Detailed descriptions of these implementation procedures were presented in the following sections. The implementation procedures were run using a NVIDIA Tesla P100 16GB GPU computing processor.



**Figure 3.8** Workflow of model implementation

### 3.7.1 Programming Language and Libraries

The CNNs were established and implemented using Python 3 based on the Tensorflow and Keras libraries. This step was achieved for each CNN by the script shown in Figure 3.9.

```

import time
import gc

import numpy as np
%matplotlib inline
from matplotlib import pyplot as plt

import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv3D, MaxPooling3D, BatchNormalization
from keras import backend as K
from keras.utils import to_categorical

from sklearn.model_selection import train_test_split

```

**Figure 3.9** Imported libraries

### 3.7.2 Data Import

The first step of the data import was to simply specify which the input dataset and the labelled dataset were for the model with their formats. Since this study aimed to conduct the pixel-wise classification, input and labelled datasets were imported pixel by pixel with information of relative pixel position. This step was achieved by the script shown in Figure 3.10. The second step was to find valid pixels that were not empty. This was important because the study area was not an upright rectangle and consequently empty pixels existed in import datasets. This step eliminated the impacts from empty pixels and allowed the LC classification for study areas in any shape. This step was achieved by the script shown in Figure 3.11.

```

try:
    import tifffile as tiff
except ImportError:
    !pip install -q tifffile
    import tifffile as tiff

cursor = drive.CreateFile({'id': raster_id})
cursor.GetContentFile('raster')
raster = tiff.imread('raster')

cursor = drive.CreateFile({'id': label_id})
cursor.GetContentFile('label')
label = tiff.imread('label')

```

**Figure 3.10** Importing data pixel by pixel

```

X, y = [], []
for i in range(raster.shape[0]):
    for j in range(raster.shape[1]):
        if not np.any(np.isnan(raster[i, j, :])):
            X.append(raster[i, j, :])
            y.append(label[i, j])

X, y = np.array(X), np.array(y)

```

(a) 1D CNN

```

assert input_width % 2 == 1
assert input_height % 2 == 1

w_offset, h_offset = (input_width - 1) // 2, (input_height - 1) // 2

raster = np.pad(raster, ((w_offset, w_offset), (h_offset, h_offset), (0, 0)), 'constant', constant_values = 0)

X, y = [], []
for i in range(w_offset, raster.shape[0] - w_offset):
    for j in range(h_offset, raster.shape[1] - h_offset):
        if (not np.any(np.isnan(raster[i - w_offset : i + w_offset + 1, j - h_offset : j + w_offset + 1, :]))):
            X.append(raster[i - w_offset : i + w_offset + 1, j - h_offset : j + w_offset + 1, :])
            y.append(label[i, j])

X, y = np.array(X), np.array(y)

time.sleep(10)
gc.collect()

```

(b) 2D and 3D CNNs

**Figure 3.11** Selection of valid pixels

### 3.7.3 Separation of Training, Validation, and Testing Data

The imported dataset was randomly separated into training data, validation data and testing data based on a rate in this step. This rate would be discussed and determined in Section 4.2.6. To ensure testing data were constant in each prediction, a random state was set when split the entire dataset into testing data and other data. Non-testing data were then divided into training data and validation data. This step was achieved by the script shown in Figure 3.12.

```

y = to_categorical(y)
y = y[:, 1:]

time.sleep(5)
gc.collect()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size, random_state=13)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size)

```

**Figure 3.12** Separation of Training and Testing Data

### 3.7.4 Training Process

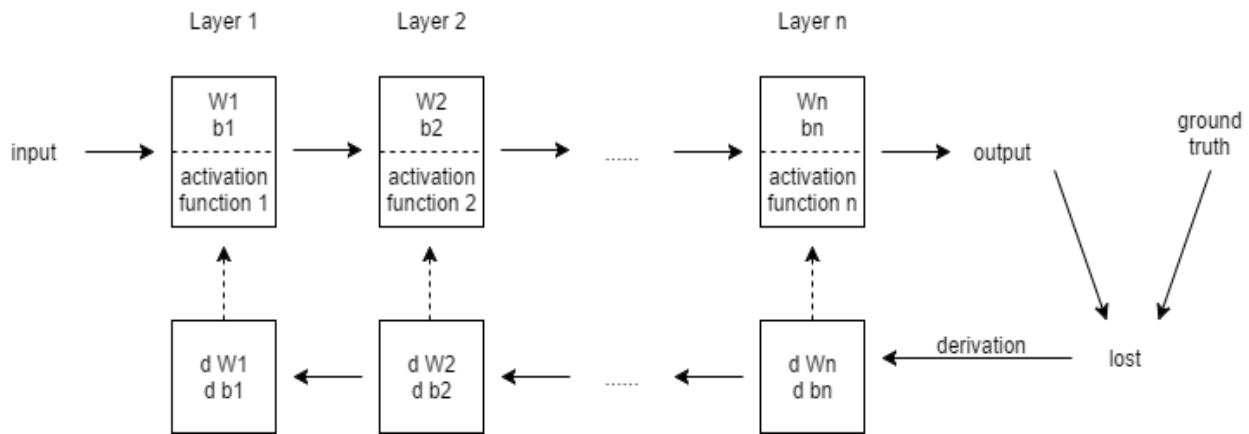
In the training process, training data and corresponding labelled data were used to determine parameters (i.e. weights and bias) and optimize the classifier. This process contained forward steps and backward steps (see Figure 3.13). Forward steps aimed to generate feature maps in each layer based on current weights and bias. Outputs of this forward step and the given ground truth labels were utilized to calculate the loss cost according to the loss function:

$$H(p, q) = - \sum_x p(x) \log q(x) \quad (3.5)$$

where  $H(p, q)$  is the cross entropy of  $p$  and  $q$ ;  $p(x)$  is the actual possibility of an event  $x$ ;  $q(x)$  is the predicted possibility of an event  $x$ .

Then, a backward step was applied to calculate the gradient of each parameter based on a learning rate which would be tested and determined in Section 4.2.7. These gradients were used to update all parameters in each layer. With these updated parameters, the system could proceed the next forward step. The circulation of a forward step and a backward step could be stopped when the loss cost of the model or the number of iterations reached a provided threshold. When an early stop was set, where the difference between the loss costs of two successive iteration was less than 0.0001, all models were stopped between the twentieth and the thirtieth epochs. Additionally, it improved the comparability of the results if the number of iterations was the same of all models. Therefore, thirty epochs were run for each model in this study. An entire training process would be stopped after the thirty iterations of forward and backward steps. After thirty epochs, the epoch with the lowest loss cost was found, and the current used values of parameters was considered as values of parameters derived from this training process.

The validation dataset was also utilized in this process to validate the currently used values of hyper-parameters and to help the determination of the optimal value of each hyper-parameter for the model. To do so, the entire training process was repeated many times with different values of hyper-parameters. After determining the optimal values of hyper-parameter, final values of parameters could be derived from the training process using these hyper-parameters. The training process was achieved by the script in Figure 3.14.



**Figure 3.13** A forward step and a backward step of training process

```
model.fit(X_train, y_train,
         epochs=epochs,
         batch_size=batch_size,
         verbose=1,
         validation_data=(X_val, y_val))
```

**Figure 3.14** Training process

### 3.7.5 Predict Process

The predict process was considered as a forward step in the training process, the aim of which was to predict LC types for testing data using parameters that were determined during the training process. Hyper-parameter values used in predict process should be the optimal values decided in the last step. The step was achieved by the script in Figure 3.15.

```
y = np.argmax(y, axis=1) + 1
y_pred = np.argmax(model.predict(X), axis=1) + 1
y_iter, y_pred_iter = iter(y), iter(y_pred)

time.sleep(5)
gc.collect()
```

**Figure 3.15** Predict process

### 3.7.6 Involved Hyper-parameters

As listed in Table 3.6, there were two key hyper-parameters involved in the implementation of each CNN. An initial value of each hyper-parameter was set for the control variate method.

**Table 3.6** Hyper-parameters involved in the implementation of CNNs

Hyper-parameters	Implemented steps	1D CNN	2D CNN	3D CNN
Rate of training, validation, and testing data	Separation of training, validation, and testing data	70%, 10%, 20%	70%, 10%, 20%	70%, 10%, 20%
Learning rate	Training process	0.001	0.001	0.001

## 3.8 Methods of Accuracy Assessment

### 3.8.1 Validation for Labelling

Since the labelling work was conducted manually, errors might be introduced. Thus, it was necessary to validate the labelled dataset. To do so, a 10 x10 pixels window was created to randomly capture 1000 sampling areas after labelling. After three months, a total of 100,000 pixels within these sampling areas were relabelled and compared with the pervious labelling results. The proportion of the pixels that were relabelled uniformly to the quantity of total tested pixels was calculated.

### 3.8.2 Accuracy Assessment for Classification

To evaluate the proposed methods, the confusion matrix, commission errors (CE), user's accuracy (UA), omission errors (OE), producer's accuracy (PA), overall accuracy (OA), and kappa coefficient were calculated.

### (1) Confusion Matrix

Also known as an error matrix, a confusion matrix is a table that describes the performance of a supervised model (see Table 3.7). In this table, each row represents the statistical data in a predicted class while each column represents the statistical data in actual class. In such a way, it is easy to find out if the model is confusing two classes. The ‘Total’ row indicates the number of pixels that should belong to a given class according to the reference data while the ‘Total’ column states the number of pixels that were identified as a given class based on the classified result.

**Table 3.7** An example table of a confusion matrix with UA and PA

	Class 1	Class 2	Class 3	Total	UA
Class 1	$A_1$	$A_2$	$A_3$	$T_a=A_1+A_2+A_3$	$U_a= A_1/T_a$
Class 2	$B_1$	$B_2$	$B_3$	$T_b=B_1+B_2+B_3$	$U_b= B_2/T_b$
Class 3	$C_1$	$C_2$	$C_3$	$T_c=C_1+C_2+C_3$	$U_c= C_3/T_c$
Total	$T_1=A_1+B_1+C_1$	$T_2=A_2+B_2+C_2$	$T_3=A_3+B_3+C_3$	$T= T_1+T_2+T_3$	N/A
PA	$U_1= A_1/T_1$	$U_1= B_2/T_1$	$U_1= C_3/T_1$	N/A	N/A

### (2) CE and UA

CE and UA were defined from the point of view of a map user but not the map maker. CE shows a proportion of the number of pixels that have been predicted to a given class but should not belong to this class to the total number of pixels of the predicted class. It is calculated by adding the number of incorrect classifications in a row of the confusion matrix together and dividing it by the value in the ‘Total’ of this row. UA is the complement of the commission errors. It is calculated by subtracting errors of commission in a row from a hundred percent. Both of these two evaluation approaches essentially indicate how often a predict class is realistically presented on the ground, which is referred to as the reliability of this predicted class.

### (3) OE and PA

OEs and PA were defined from the view of the map maker. OE shows a proportion of the number of pixels that should belong an actual class but have been predicted to other classes to the total number of pixels of the actual class. It is calculated by summing the number of incorrect classifications in a column of the confusion matrix and dividing them by the value in the ‘Total’ of the column. UP is the complement of the omission errors. It is calculated by subtracting errors of omission in a column from a hundred percent. Both of these two evaluation methods essentially indicate how often real features on the ground are correctly shown on the predicted classification map.



#### (4) OA

OA is usually expressed as percentages, where 100% accuracy represented a perfect classification that all pixels have been classified correctly. OA is the proportion of the amount of correctly predicted pixels to all pixels, which offered general accuracy information to both map users and producers.

#### (5) Kappa Coefficient

Kappa coefficient is a statistic that measures inter-rater agreement for categorical items. Since it considers the possibility of the agreement occurring by chance, kappa is a more robust measure than simple percent agreement calculation (Cohen, J., 1960). It essentially evaluates how better the classification performed compared to randomly assigning values. This coefficient could range from -1 to 1. A value close to 1 indicated that the classification was significantly better than random while a negative number reflected the classification was worse than random. However, the significance of Kappa coefficient is controversial especially for purposes of accuracy assessment and map comparison. Pontius and Millones (2011) summarized more than ten years of studies on the Kappa coefficient and suggested that researchers should cease using Kappa coefficients for purposes of accuracy assessment and map comparison. The first reason is that the Kappa coefficient tries to compare accuracy to the randomness, which is not reasonable for map construction (Pontius and Millones, 2011). Moreover, it also has fundamental conceptual faults, for example, it has no useful interpretation (Pontius and Millones, 2011). Although Kappa coefficient has problems about interpreting classification accuracy, many recent studies still use it as one of the accuracy assessment methods for LC classifications (e.g. Morsy et. al., 2017a; Teo and Wu, 2017; Matikainen et. al., 2017a). Therefore, this study calculates Kappa coefficients for the benefit of researchers who may be curious about it, but would not use it to interpret the classification accuracy.

### **3.9 Chapter Summary**

This chapter described the methodology of the thesis in details. It introduced the stepwise processes in the multispectral ALS data pre-processing, the construction of input datasets, data labelling, the selection of models, the establishment of CNNs, the implementation of CNNs, and the accuracy assessment. The multispectral ALS point clouds were pre-processed at first. A total of nine raster images with different information were generated from the pre-processed point clouds. These images were assembled into six input data combinations. Meanwhile, the labelled

dataset was created using the orthophotos as the ground truth. Also, three deep-learning networks were established. Then, each input data combination was used to train and validate each network. This step developed eighteen LC classification models with different parameters to predict LC types for pixels. Therefore, a total of eighteen classification results were produced. Finally, accuracy assessments and comparisons were done for the eighteen classification results to seek an optimal scheme. Results of the eighteen models are presented and compared in the next chapter.

# Chapter 4

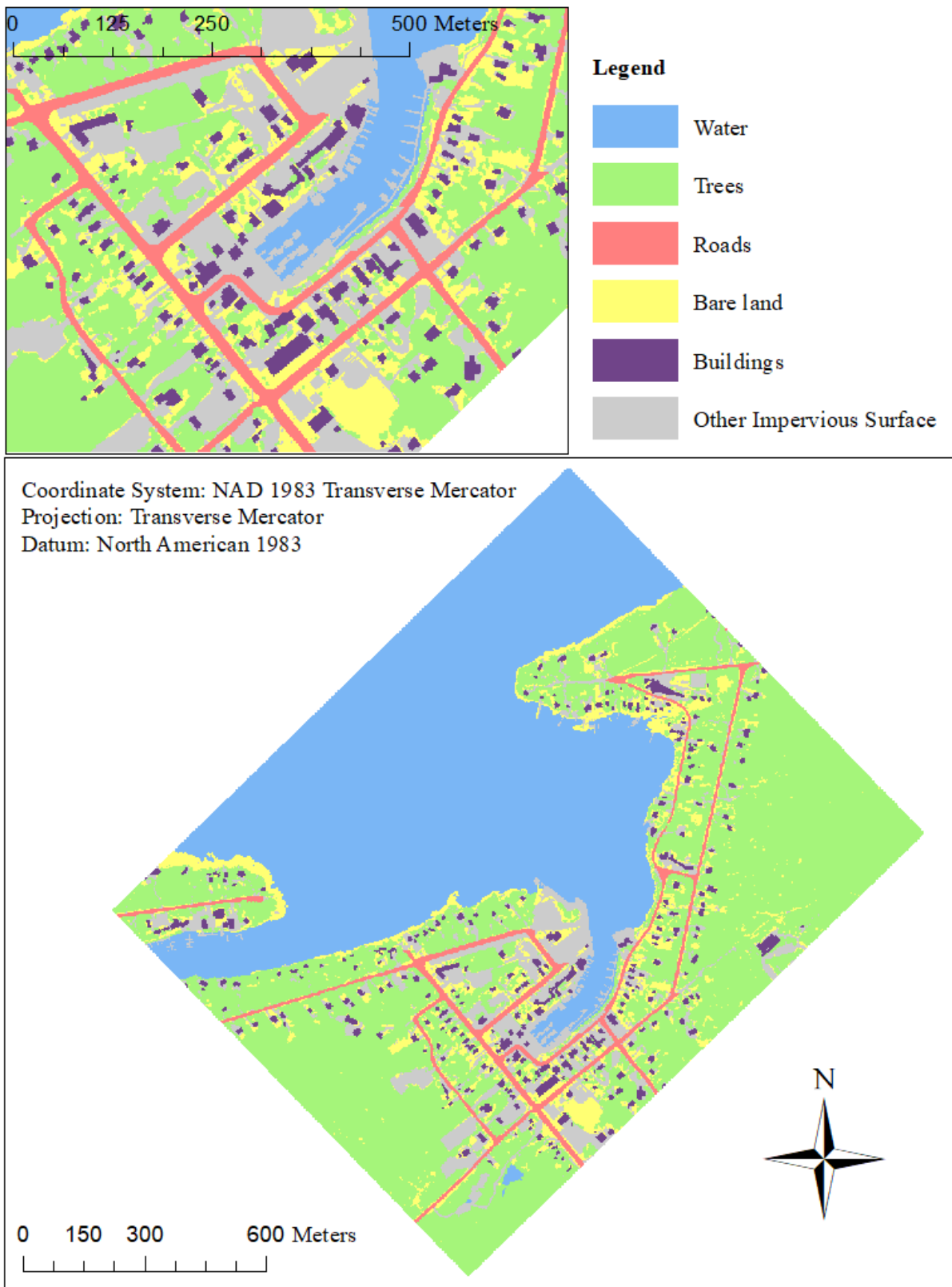
## Results and Discussion

This chapter presents and discusses the results obtained by the stepwise processes proposed in Chapter 3. In Section 4.1, the labelled dataset is displayed and validated. Values of hyper-parameters involved in the establishment and implementation processes of CNNs are discussed and determined in Section 4.2. Furthermore, classification results of the six input data combinations are analyzed and compared to seek the best combination of information extracted from multispectral ALS point clouds for LC classification in Section 4.3. The performance of the three proposed CNNs is also evaluated and compared in detail in Section 4.4. Throughout this chapter, the accuracy of the classification results is assessed via comparing predicted maps with the labelled dataset on a pixel by pixel basis.

### 4.1 Labelled Dataset

#### 4.1.1 Result of Labelling

Each 1m pixel within the study area was labelled as one of the six LC types based on the rules and steps proposed in Section 3.4. The manually labelled image is presented in Figure 4.1. Detailed examples of each class are listed in Table 4.1. As shown in these figures, the boundary of each class has been smoothly and clearly labelled. Additionally, details in the study area such as a few BAL pixels along the ROD have been labelled accurately. Comparing Figure 4.1 and Table 4.1 with Figure 3.1 and Table 3.4, respectively, it can be shown that the labelled dataset can precisely mirror the type, scope, and distribution of LC classes on the surface of the Earth. Table 4.2 shows statistics of the labelled dataset in terms of each class. There is a total of 1,990,682 pixels in the study. As shown in Table 4.2, around 80% of pixels of the study area belong to WAT or TRE while BUD and ROD only occupy less than 3% of pixels, respectively. Moreover, only 7% of pixels are BAL pixels. Excessive imbalance of area of each LC type may negatively influence classification results since the number of pixels of a specific class may be too small to be learned. This was the reason why BUD or BAL was not divided into sub-classes such as Commercial BUD and Residential BUD, or Sand and Grass.



**Figure 4.1** Labeled LC map of the study area

**Table 4.1** Detailed examples of each LC class in labelled dataset

LC Types	Examples	Illustrations
WAT	(4) Open water (5) Harbours (6) Small lake	(1)  (2)  (3) 
TRE	(3) Multiple (4) Single	(1)  (2) 
BUD	(4) Commercial (5) Residential (6) Small shed	(1)  (2)  (3) 
ROD	(3) Straight road (4) Crossroad	(1)  (2) 
BAL	(4) Sand (5) Rocky area (6) Grass	(1)  (2)  (3) 
OIS	(6) Parking lot (7) Concrete open area (8) Pathway (9) Concrete docks (10) Boats	(1)  (2)  (3)  (4)  (5) 

**Table 4.2** Statistics of the labelled dataset

LC Types	Pixels	Percentage
WAT	803,553	40.4
TRE	777,759	39.1
ROD	58,523	2.9
BAL	139,340	7.0
BUD	50,875	2.6
OIS	160,632	8.1
Total	1,990,682	100

### 4.1.2 Validation of Labelling

Based on the method proposed in Section 3.7.1, 100,000 pixels were relabelled to validate the labelled dataset. The time interval between these two labelling tasks was longer than three months. To clearly display the validation result, a confusion matrix was created. In Table 4.3, each row counts the pixels found in a class of the relabelled dataset while each column represents the quantity of pixels in a class of the first-labelled dataset. The Total row indicates the number of pixels that should belong to a given class based on the first-labelled dataset while the Total column states the quantity of pixels in a given class according to the relabelled result. Considering the first-labelled dataset as the reference dataset, the Accuracy row describes the proportion of correctly labelled pixels. Treating the relabelled dataset as the reference dataset, the proportion of accurately labelled pixels is calculated in the Accuracy column.

**Table 4.3** Confusion matrix of first-labelled dataset and relabelled dataset

LC Types		First-labelled dataset							Total	Accuracy (%)
		WAT	TRE	ROD	BAL	BUD	OIS			
Relabelled dataset	WAT	38764	1	0	0	1	0	38766	99.99	
	TRE	0	35286	1	1	2	0	35290	99.99	
	ROD	0	1	3991	0	0	0	3992	99.97	
	BAL	0	0	0	8038	0	3	8041	99.96	
	BUD	1	0	0	1	3926	0	3928	99.95	
	OIS	0	0	0	2	0	9981	9983	99.98	
	Total	38765	35288	3992	8042	3929	9984	100000	N\A	
	Accuracy (%)	99.99	99.99	99.97	99.95	99.92	99.97	N\A	N\A	

As shown in Table 4.3, the accuracy of each class is higher than 99.9% no matter which labelled dataset is used as the reference dataset. Furthermore, OA of statistics listed in Table 4.3 is 99.99%, which means 99.99% pixels have the same labels in the first-labelled dataset and the relabelled dataset. Thus, it can be concluded that the labelled dataset is reliable.

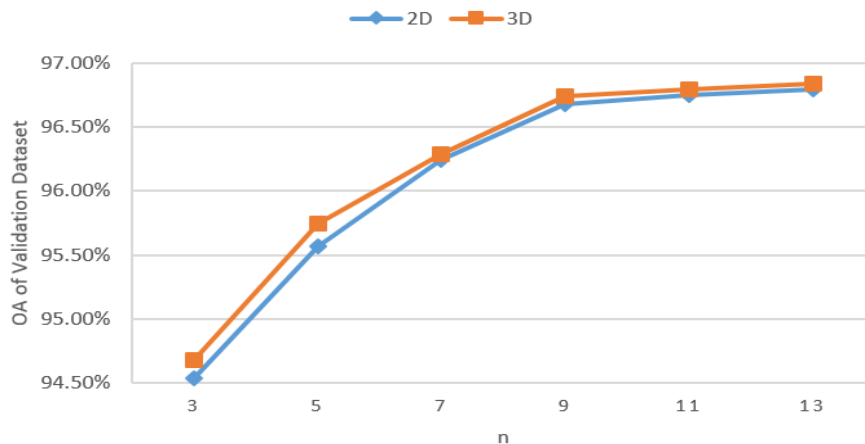
## 4.2 Hyper-parameters

Since values of hyper-parameters cannot be estimated from raw data, it is highly important to determine their values through experiments. To do so, each hyper-parameter was tested separately, keeping all other hyper-parameters constant. To make it convenient to be calculated and stored in binary computers, many values were set to the positive exponential power of two. All of these tests were finished in the training process with validation datasets. According to the accuracy and

efficiency, the most appropriate value of each hyper-parameter will be selected. There are seven significant hyper-parameters involved in the establishment and implementation of each CNN. Because the control variate method was used, an initial value of each hyper-parameter was set. Combination 1, the most classical input data of rasterized multispectral ALS dataset, was used as the input dataset in each test. A total of thirty epochs were run for each test.

#### 4.2.1 Shape of Each Input Unit

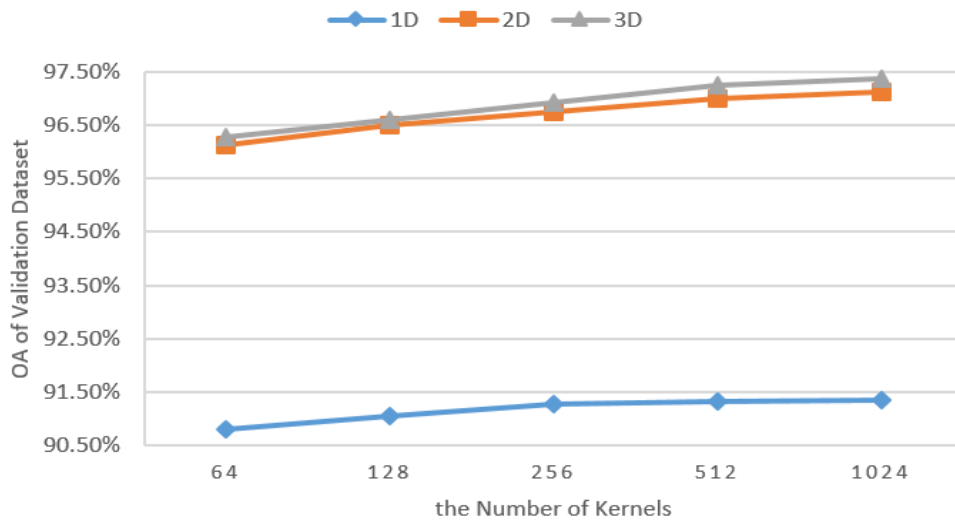
To achieve pixel-wise classification, input datasets were decomposed to pixels during convolution and further computation. In the 1D CNN, the convolution is computed along only the direction of depth for a single pixel; any information of other pixels is needless. Hence, the shape of each input unit for the 1D CNN should be  $(1, 1, m)$ , in which  $m$  refers to the depth of the input. In the 2D CNN and the 3D CNN, as information of adjacent pixels is required, the shape of input unit should be  $(n, n, m)$ , where  $n$  should be an odd number that is greater than one and  $m$  refers to the depth of the input. The predicted LC type of this unit is given to the centre pixel; hence, each valid pixel within the study area should become the centre pixel of an input unit once. The practical meaning of the shape of each input unit is that using information of a pixel and its surrounding  $n \times n - 1$  pixels to predict a LC type for the pixel. To determine the most appropriate values of  $n$  for the 2D and 3D CNNs, respectively, tests with different  $n$  values were done. According to Figure 4.2, the OA of the 2D and 3D CNNs increases significantly when  $n$  is smaller than nine. Moreover, the OA of these networks keeps stable when  $n$  is greater than nine. Furthermore, the number of parameters and run time in the model with eleven as  $n$  value are two times higher than them in the model with nine as  $n$  value. Therefore, considering accuracy and efficiency, the most appropriate shape of each input unit for the 2D and the 3D CNNs is  $(9, 9, m)$ .



**Figure 4.2** Results of different shape of input unit in 2D and 3D CNNs

### 4.2.2 The Number of Kernels

The number of kernels in convolution layers determines the quantity of filters used in these layers and the number of feature images derived from these layers. To decide the most appropriate number of kernels for 1D, 2D and 3D CNNs, respectively, tests with different quantity of kernels were performed. Making it convenient to calculate and store in binary computers, every value of the number of kernels was set to a positive exponential power of two. As shown in Figure 4.3, the OA of the 1D CNN increases sharply before 256 and increases slowly after 256. Moreover, the number of parameters and run time in the model with 256 kernels are two times higher than them in the model with 512 kernels. Thus, 256 is selected as the number of kernels in the 1D CNN. In the 2D and 3D CNNs, based on the same reasons, 512 is selected as the most appropriate number of kernels.

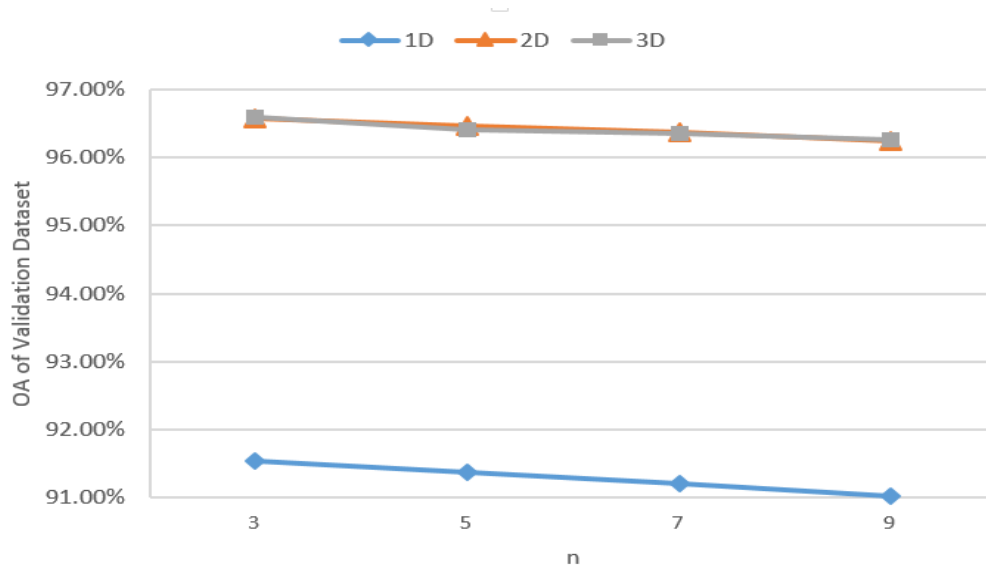


**Figure 4.3** Results of different number of kernels in the 1D, 2D and 3D CNNs

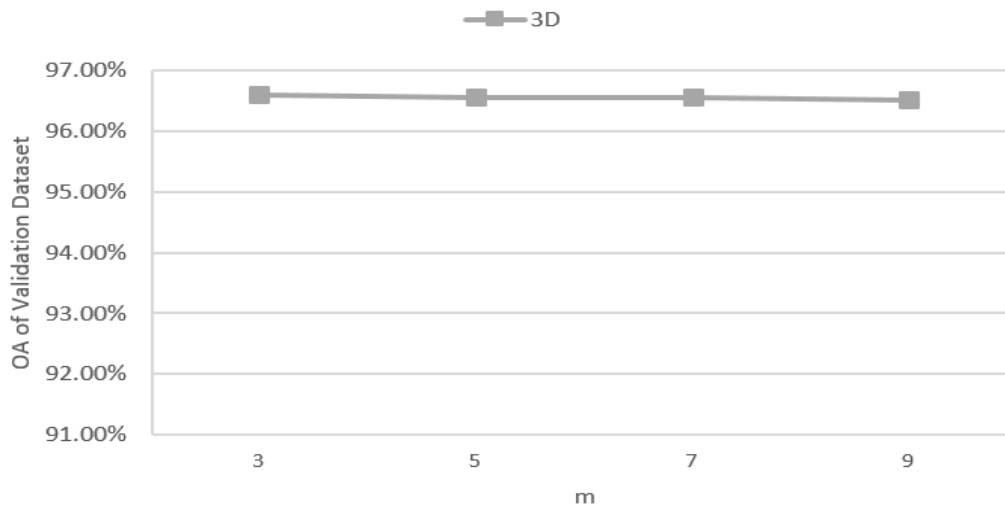
### 4.2.3 Size of Kernels

Kernels in 1D, 2D and 3D convolutional layers should have 1D, 2D and 3D structures, respectively. Based on the convolutional principles in convolutional layers, length, and width of a kernel should be the same; also, length, width, and depth of a kernel should be odd numbers that are greater than one. Hence, the size of kernels in 1D, 2D, and 3D CNNs should be  $(n)$ ,  $(n, n)$  and  $(n, n, m)$ , respectively. The  $n$  and  $m$  were tested separately. As show in Figure 4.4, since the OA slightly decreases while both the number of parameters and run time slightly increase when  $n$  or  $m$  values increase, three should be the most appropriate value of both  $n$  and  $m$ . Thus, the most appropriate kernel size in the 1D, 2D, and 3D CNNs are  $(3)$ ,  $(3, 3)$  and  $(3, 3, 3)$ , respectively.





(a) Tests of n

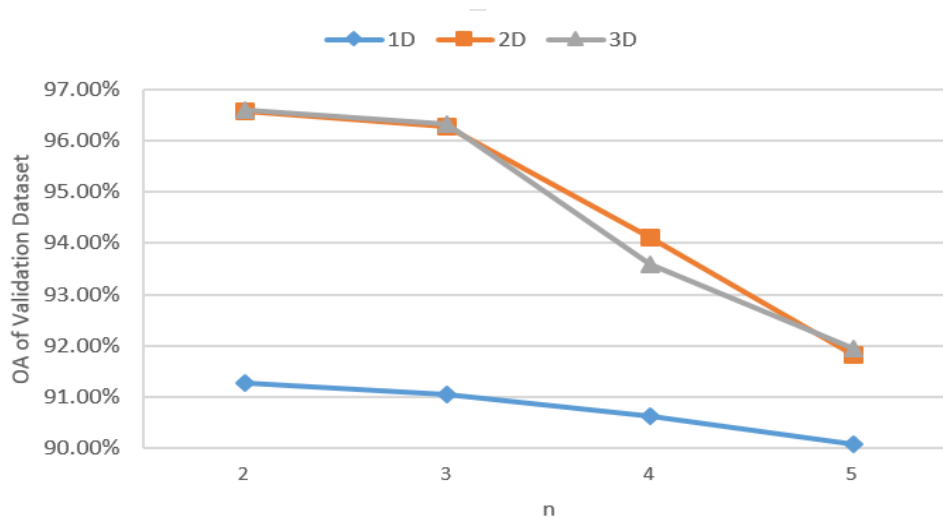


(b) Tests of m

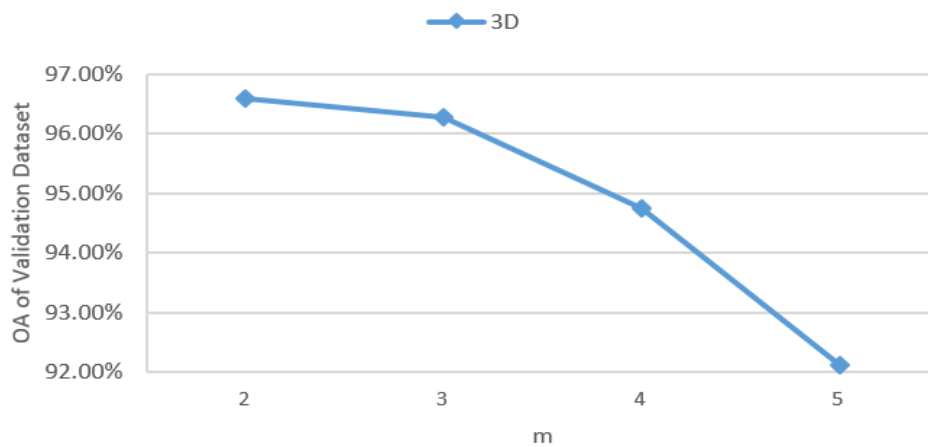
**Figure 4.4** Results of different size of kernels in the 1D, 2D and 3D CNNs

#### 4.2.4 Size of Pooling Windows

Similar to the size of kernels, size of pooling windows in 1D, 2D, and 3D CNNs should also be  $(n)$ ,  $(n, n)$  and  $(n, n, m)$ , respectively. The difference is that  $n$  and  $m$ , in this instance, can be any integer that is greater than one. According to Figure 4.5, two is the most appropriate value for both  $n$  and  $m$ . Therefore, the most appropriate kernel sizes in the 1D, 2D and 3D CNNs are  $(2)$ ,  $(2, 2)$  and  $(2, 2, 2)$ , respectively.



(a) Tests of n

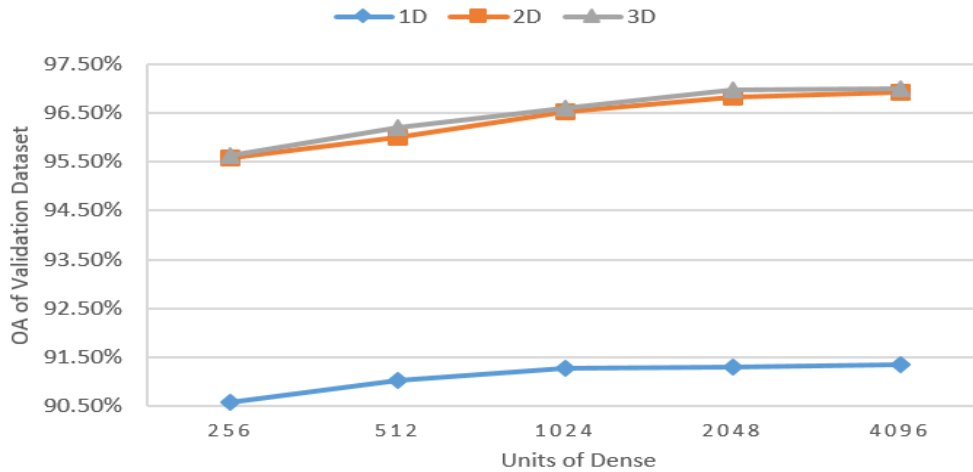


(b) Tests of m

**Figure 4.5** Results of different size of kernels in the 1D, 2D and 3D CNNs

#### 4.2.5 Units of Dense

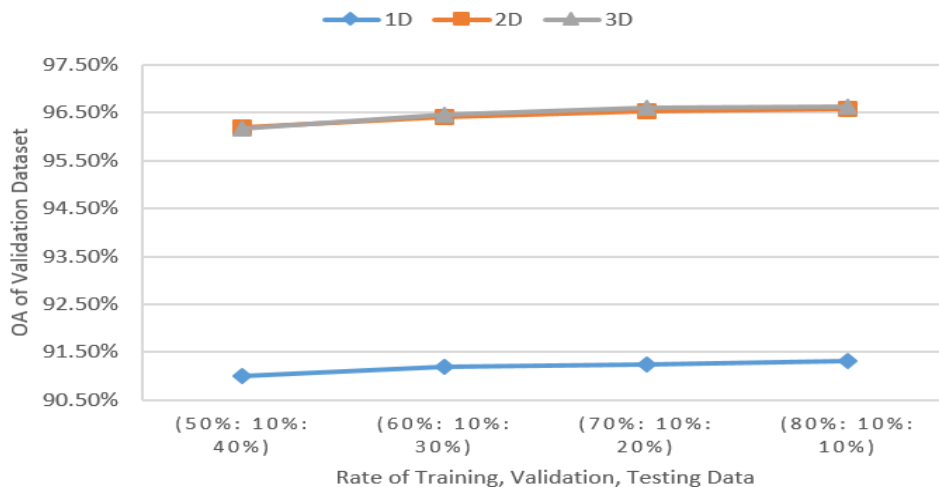
In the dense layers, units of dense determine the number of involved neurons. To choose the most appropriate units of dense for the CNNs, tests with different dense units were done. Making it convenient to calculate and store in binary computers, every value of the units of dense was set to a positive exponential power of two. As shown in Figure 4.6, the OA of the 1D CNN improves gradually when units of density are smaller than 1024, and then maintains smooth when units of density are larger than 1024. Additionally, the number of parameters and run time in the model with 1024 as units of density are more than two times higher than them in the model with 2048 as units of density. Thus, 1024 is selected as the number of kernels in the 1D CNN. In the 2D and 3D CNNs, based on the same reasons, 2048 is chosen.



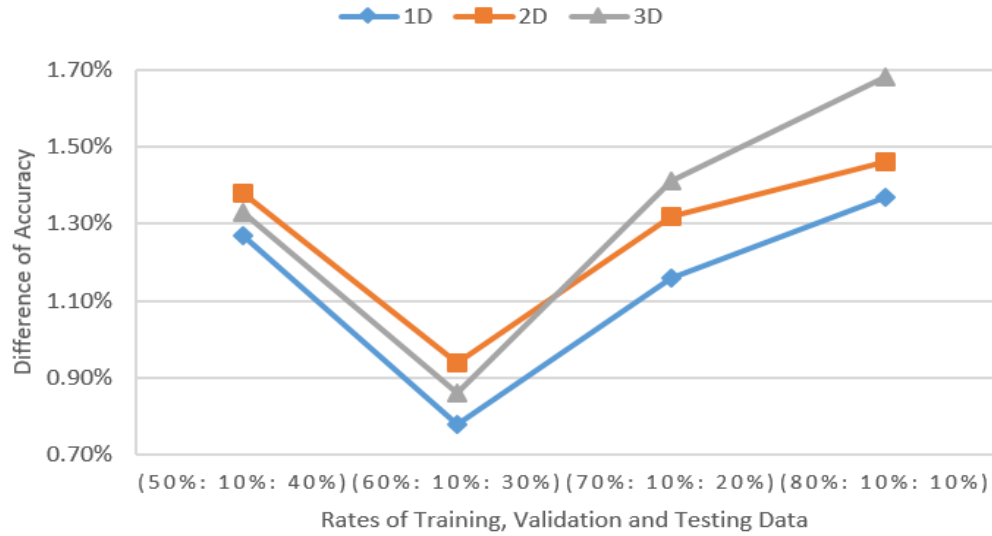
**Figure 4.6** Results of different units of dense in the 1D, 2D and 3D CNNs

#### 4.2.6 Rate of Training, Validation and Testing Data

Usually, to better train a model and avoid overfitting caused by insufficient training data, training data account for more than half of the entire dataset. Validation data often account for only 10% of the dataset. Thus, four rates of training, validation and testing data were tested for each model. Based on Figure 4.7 (a), OA tends to be stable in these models after (60%: 10%: 30%). As a result, (50%: 10%: 40%) is excluded. However, no matter how the OA of validation datasets changes, OA of training datasets increases sharply with the enhancement of the amount of training data. Therefore, from Figure 4.7 (b), it can be shown that differences which are calculated using OAs of training dataset to subtract OAs of validation dataset are lowest when the rate is (60%: 10%: 30%). It indicates that overfitting caused by too many training data is the least when the rate is (60%: 10%: 30%) in each CNN. Consequently, (60%: 10%: 30%) is the best rate of training, validation and testing data for these networks.



(a) OA of validation dataset



(b) Difference of OA

**Figure 4.7** Results of different units of dense in the 1D, 2D and 3D CNNs

#### 4.2.7 Learning Rate

In a deep-learning method, the learning rate controls the size of a learning step for each training iteration. Inappropriate learning rate can result in divergence or slow convergence. To select the most appropriate learning rate for these CNNs, 0.05, 0.01, 0.005, 0.001, 0.0005 and 0.0001 were tested as the learning rate, respectively. Based on the classification results of validation datasets, it can be concluded that 0.005 is the optimal learning rate in each CNN.

#### 4.2.8 Summary of Hyper-parameters

Based on the tests results and discussions above, the optimal values of hyper-parameters are determined as listed in Table 4.4.

**Table 4.4** Summary of hyper-parameters

Hyper-Parameters	1D CNN	2D CNN	3D CNN
Shape of input unit	(1, 1, depth)	(9, 9, depth)	(9, 9, depth)
Number of kernels	256	512	512
Size of kernels	(3)	(3, 3)	(3, 3, 3)
Size of pooling windows	(2)	(2, 2)	(2, 2, 2)
Units of dense	1024	2048	2048
Rate of training, validation, and testing data	(60%: 10%: 30%)	(60%: 10%: 30%)	(60%: 10%: 30%)
Learning rate	0.005	0.005	0.005

### 4.3 Analysis of LC Classification

There are six input data combinations (i.e. Combination 1-6, defined in Section 3.3.4) and three CNNs used in this thesis for LC classification. Therefore, there are totally eighteen trained and validated models. To ensure the significance of the classification accuracy, each model was run 10 times. The averaged OA and kappa coefficient of each model are listed in Table 4.5. It can be seen that the highest overall classification accuracy of 97.2%, with a kappa index of 0.96, can be achieved using the proposed 3D CNN and input data Combination 4. According to Section 2.2, most of the multispectral ALS LC classification accuracy is higher than 90% when using non-deep-learning methods, but only one publication showed accuracies higher than 96%. The kappa indexes of these published methods are around 0.9. Thus, this study achieves an admirable classification result, which is better than most of the published multispectral ALS data classification results.

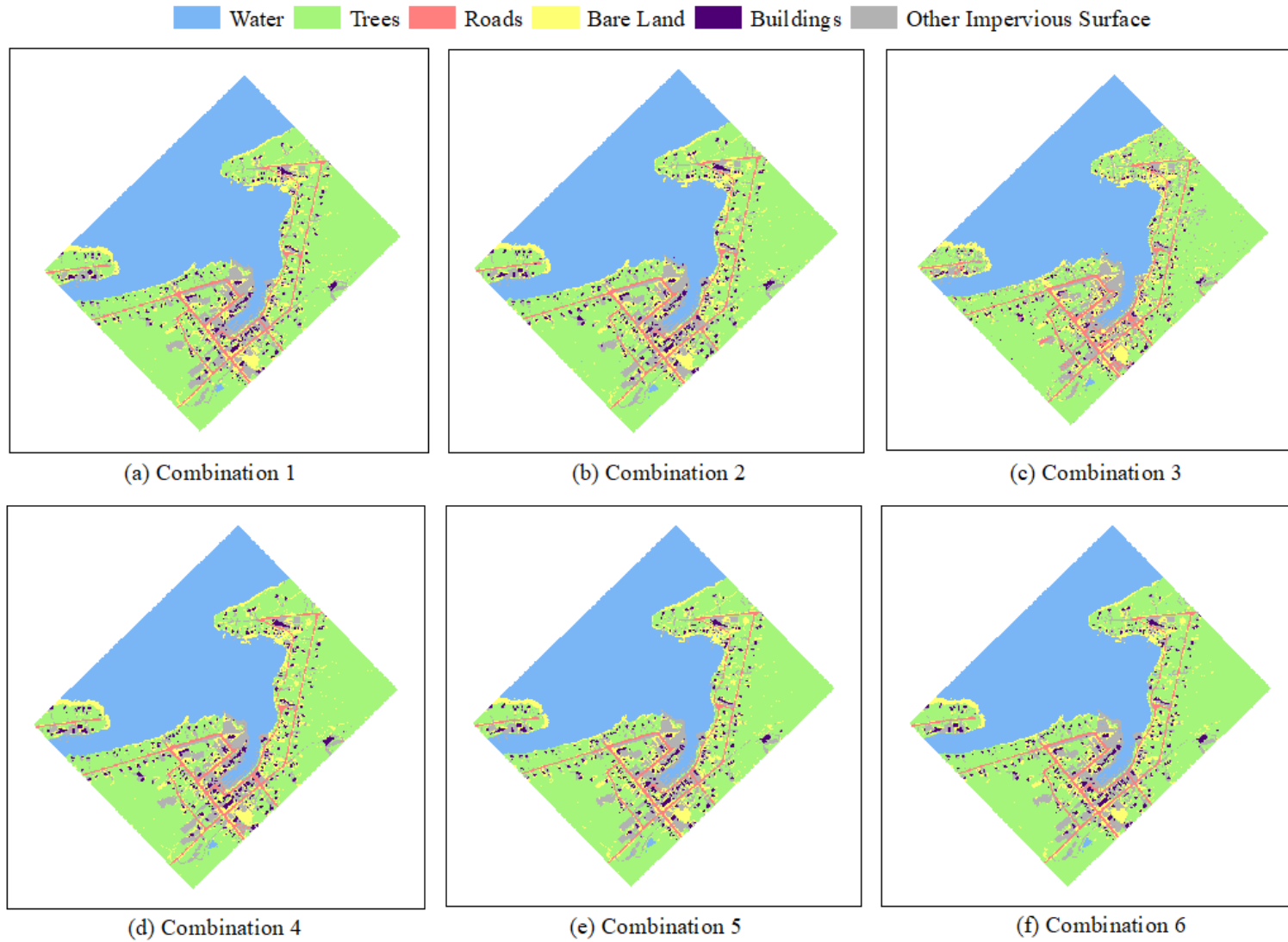
The OA of the three input datasets designed in this study, Combinations 4, 5 and 6, is on average 3.8% point higher than that of the classic input datasets, Combination 1, 2 and 3. Combination 4 achieves the best OA and kappa coefficient no matter which CNN is applied. Combination 3 on average performs worst compared to others, especially when the 2D or 3D CNNs is applied. Combination 2 is the most sensitive to the alteration of CNNs; when using this input data, OA of 2D or 3D CNN is at least 17% higher than OA of 1D CNN. Furthermore, the 3D CNN obtains the highest OA and kappa coefficient, indicating that it has a high success for pixel-wise LC classification and performs significantly better than random. OA and kappa coefficient of the 2D CNN are only slightly lower than them of the 3D CNN, suggesting that the 2D CNN also performs well in pixel-wise LC classification. Conversely, the 1D CNN achieves the lowest classification accuracy with relatively low OA and kappa coefficient. The OA of the 2D and 3D CNNs was on average 8.4% higher than that of the 1D CNN. To better understand the results, these OAs and kappa coefficients would be described and analyzed in detail in the following subsections.

**Table 4.5** OA and kappa coefficient of each model

Input Data Combinations	1D CNN		2D CNN		3D CNN	
	OA (%)	kappa	OA (%)	kappa	OA (%)	kappa
1	90.3	0.85	96.5	0.95	96.8	0.95
2	79.2	0.68	96.5	0.95	96.6	0.95
3	82.5	0.73	90.8	0.87	91.0	0.87
4	91.2	0.87	97.0	0.96	97.2	0.96
5	90.3	0.85	96.6	0.95	96.8	0.95
6	91.0	0.86	97.0	0.96	97.0	0.96

### 4.3.1 Performances of Different Input Data Combinations

To analyze performances of different input data combinations in detail, a confusion matrix with calculated CE, UA, OE and PA is utilized for the best result of each input data combination (see Tables 4.6 – 4.11). These tables clearly describe how well the classification models performed in terms of each LC class and where misclassification occurs frequently. The PA reflects the classification accuracy of each class while the UA reveals the reliability of each class in the classified image. Comparing these tables, variances of classification results of input datasets will be clearly demonstrated. Predicted maps of different input data combinations are also shown in Figure 4.8, which visualizes the classification results of the best trained models. The qualitative comparison in Figure 4.8 between different inputs will be integrated with the quantitative comparison in Tables 4.6 to 4.11. To make logic unambiguous, these images and tables will be described and compared divisionally, integrated with information provided by Table 4.5.



**Figure 4.8** Predicted maps of different input data combinations with the 3D CNN

**Table 4.6** Confusion matrix for Combination 1 with the 3D CNN

LC Classes		Actual							
		WAT	TRE	ROD	BAL	BUD	OIS	Total	UA (%)
Predicted	WAT	240597	49	0	289	21	314	241270	99.7
	TRE	110	229308	119	4336	577	1655	236105	97.1
	ROD	0	91	47005	168	7	583	47854	98.2
	BAL	280	2422	192	35638	322	1849	40703	87.6
	BUD	3	158	8	137	15465	432	16203	95.5
	OIS	301	1103	694	1287	1099	10356	14840	69.8
	Total	241291	233131	48018	41855	17491	15189	596975	N/A
	PA (%)	99.7	98.4	97.9	85.2	88.4	68.2	N/A	N/A

**Table 4.7** Confusion matrix for Combination 2 with the 3D CNN

LC Classes		Actual							
		WAT	TRE	ROD	BAL	BUD	OIS	Total	UA (%)
Predicted	WAT	240613	225	2	324	58	343	241565	99.6
	TRE	122	226729	102	3002	314	1055	231324	98.0
	ROD	0	87	46141	151	8	387	46774	98.7
	BAL	251	3918	149	36248	238	1430	42234	85.8
	BUD	12	377	40	277	15945	843	17494	91.2
	OIS	293	1795	1584	1853	928	11131	17584	63.3
	Total	241291	233131	48018	41855	17491	15189	596975	N/A
	PA (%)	99.7	97.3	96.1	86.6	91.2	73.3	N/A	N/A

**Table 4.8** Confusion matrix for Combination 3 with the 3D CNN

LC Classes		Actual							
		WAT	TRE	ROD	BAL	BUD	OIS	Total	UA (%)
Predicted	WAT	239985	94	5	552	13	626	241275	99.5
	TRE	63	213772	63	2431	428	1698	218455	97.9
	ROD	3	1018	46892	593	1988	4733	55227	84.9
	BAL	642	5816	189	31929	785	7461	46822	68.2
	BUD	41	964	10	150	10609	308	12082	87.8
	OIS	557	11467	859	6200	3668	363	23114	1.6
	Total	241291	233131	48018	41855	17491	15189	596975	N/A
	PA (%)	99.5	91.7	97.7	76.3	60.7	2.4	N/A	N/A



**Table 4.9** Confusion matrix for Combination 4 with the 3D CNN

LC Classes		Actual							
		WAT	TRE	ROD	BAL	BUD	OIS	Total	UA (%)
Predicted	WAT	240780	41	0	236	11	336	241404	99.7
	TRE	67	228887	109	3102	287	1228	233680	98.0
	ROD	0	71	47401	162	6	1587	49227	96.3
	BAL	239	2687	225	36983	175	1727	42036	88.0
	BUD	7	319	23	294	16499	854	17996	91.7
	OIS	198	1126	260	1078	513	9457	12632	74.9
	Total	241291	233131	48018	41855	17491	15189	596975	N/A
	PA (%)	99.8	98.2	98.7	88.4	94.3	62.3	N/A	N/A

**Table 4.10** Confusion matrix for Combination 5 with the 3D CNN

LC Classes		Actual							
		WAT	TRE	ROD	BAL	BUD	OIS	Total	UA (%)
Predicted	WAT	240462	46	0	217	1	229	240955	99.8
	TRE	86	228242	77	3673	284	1474	233836	97.6
	ROD	0	126	46815	144	20	421	47526	98.5
	BAL	382	3101	207	36347	322	2020	42379	85.8
	BUD	2	388	13	229	16003	618	17253	92.8
	OIS	359	1228	906	1245	861	10427	15026	69.4
	Total	241291	233131	48018	41855	17491	15189	596975	N/A
	PA (%)	99.7	97.9	97.5	86.8	91.5	68.7	N/A	N/A

**Table 4.11** Confusion matrix for Combination 6 with the 3D CNN

LC Classes		Actual							
		WAT	TRE	ROD	BAL	BUD	OIS	Total	UA (%)
Predicted	WAT	240884	111	17	413	39	471	241935	99.6
	TRE	27	230415	141	5155	344	1428	237510	97.0
	ROD	0	61	47025	138	12	656	47892	98.2
	BAL	181	1198	193	33932	177	922	36603	92.7
	BUD	0	244	6	245	16177	622	17294	93.5
	OIS	199	1102	636	1972	742	11090	15741	70.5
	Total	241291	233131	48018	41855	17491	15189	596975	N/A
	PA (%)	99.8	98.8	97.9	81.1	92.5	73.0	N/A	N/A

#### **4.3.1.1 Combination 1 versus Combination 2**

Combination 1 is the most classical input data of rasterized multispectral ALS datasets, offering comprehensive spectral information of each channel and general information of the height model. As shown in Table 4.6, WAT, TRE and ROA have the highest PA, which indicates that the classification accuracy of these three classes is the highest when Combination 1 is used. It also states that more than 97% of WAT, TRE and ROA ground truth pixels also appear as WAT, TRE and ROA in the classified image, respectively. Furthermore, the three classes also have the largest UA, which means that the reliability of the three classes in the predicted map is the highest when the input is Combination 1. To be more specific, more than 97% of the WAT, TRE and ROA pixels in the classified map actually represent WAT, TRE and ROA on the ground, respectively. Dissimilar to these three classes, although BUD has relatively low classification accuracy, its reliability is high. This states that even though more than 95% of the pixels identified as BUD in the predicted image are actual BUD pixels, only 88% of the reference BUD pixels have been correctly identified as BUD. Many BUD pixels are mistakenly classified as OIS. Additionally, OIS has the lowest classification accuracy and the lowest reliability. Only about 68% of OIS ground truth pixels are correctly displayed in the categorized image. Also, only less than 70% of OIS pixels in the predicted map actually represent OIS on the ground. A number of OIS pixels are misclassified to TRE and BAL while many TRE, BAL and BUD pixels are incorrectly classified as OIS.

Removing the height information from Combination 1, information in Combination 2 is spectral information that can be extracted from traditional multispectral optical images. According to Table 4.7, WAT, TRE and ROA have the highest classification accuracy and the highest reliability in the classified image when Combination 2 is applied. It means that more than 96% of WAT, TRE and ROA ground truth pixels also appear in corresponding classes in the predicted map; more than 98% of the WAT, TRE and ROA pixels in the categorized image actually represent corresponding classes on the ground. Whereas OIS has the lowest classification accuracy and reliability. It indicates that only about 73% of actual OIS pixels are correctly displayed in the classified image. Also, only 63% of OIS pixels in the predicted image are the true OIS pixels. In detail, plentiful OIS ground truth pixels are mistakenly categorized as TRE, BAL and BUD; a mass of other pixels except WAT pixels are misclassified as OIS in the predicted map. Moreover, the fact, PA of OIS higher than its UA, suggests that pixels misclassified as OIS in classified result

are more than the OIS ground truth pixels which are wrongly classified as other classes.

According to Table 4.5, OA of Combination 1 is on average 3.8% higher than that of Combination 2, which means that typical information provided by multispectral ALS data performs better in CNN-based LC classification than only spectral information which represents information extracted from traditional multispectral optical images. Especially when the 1D CNN is applied, OA of Combination 2 is 11.0% lower than that of Combination 1. It reveals that typical information provided by multispectral ALS dataset is comparatively abundant; the 1D CNN, which has lower classification capability, still can learn many features from Combination 1 and achieve good classification results. Nevertheless, spectral information which represents traditional multispectral optical images is insufficient for CNN-based LC classification. Moreover, comparing Figures 4.8 (a) and (b) with the labelled image, it can be also found that Combination 1 performs better than Combination 2. To be more specific, both of the two inputs perform well for WAT, TRE and ROD as shown in Table 4.6 and Table 4.7. Furthermore, when the input is Combination 2, PA of BUD and OIS is higher; fewer pixels which should belong to these two classes are misclassified to wrong classes. However, compared to Combination 2, Combination 1 generally leads to less misclassification. Specifically, BUD and OIS pixels in the categorized image are more reliable when Combination 1 is used; fewer other ground truth pixels are inaccurately classified as BUD and OIS pixels. Additionally, fewer actual TRE and ROD pixels are misclassified to BAL and OIS. In summary, the multispectral ALS data is superior to traditional multispectral optical imagery in deep-learning LC classification, which is identical to the results obtained from others mentioned in Section 2.1.

#### **4.3.1.2 Combination 3 versus Combination 1**

Deleting the spectral information of Green and SWIR bands from Combination 1, Combination 3 simulates the typical ALS data. As shown in Table 4.8, WAT has the highest classification accuracy and the highest reliability in the classified image when Combination 3 is applied. ROD also has high classification accuracy, but its reliability is relatively low. This reveals that even though more than 97% of the true ROD pixels have been correctly identified as ROD, only less than 85% of the ROD pixels in the predicted map are actual ROD pixels. Specifically, many OIS, BUD and TRE ground truth pixels are erroneously categorized as ROD. Oppositely, TRE has relatively low classification accuracy and high reliability, which suggests that less than 92% of the reference TRE pixels have been correctly identified, although more than 97% of the

TRE pixels in the classified map are actual TRE pixels. Specifically, some TRE ground truth pixels are mistakenly classified as OIS and BAL. Furthermore, the classification accuracy and reliability of BAL and BUD are relatively low. To be more specific actual BAL pixels are incorrectly classified as TRE and OIS; TRE and OIS ground truth pixels are misclassified to BAL. Inversely, BUD ground truth pixels are mistakenly classified as ROD and OIS; TRE ground truth pixels are misclassified as BUD in the predicted image. Moreover, pixels misclassified as BAL are more than the BAL ground truth pixels which are wrongly classified as other classes. Conversely, BUD pixels on the ground misclassified as other classes are more than pixels mistakenly categorized as BUD in the classified map. Additionally, OIS has the extremely low classification accuracy and reliability. Only about 2% of OIS ground truth pixels correctly appear in the prediction result. Also, only less than 2% of OIS pixels in the predicted image actually represent this LC type on the ground. Most of OIS pixels are classified incorrectly.

As shown in Table 4.5, OA of Combination 3 is on average 6.4% lower than that of Combination 1, which shows that the typical ALS data perform worse in CNN-based LC classification than the typical information offered by multispectral ALS data. It also states that the multispectral ALS data provides more sufficient information for CNNs than typical ALS data. Furthermore, comparing Figures 4.8 (a) and (c) with the labelled image, it can also be easily shown that the predicted map of Combination 3 has more misclassified pixels and contains more noise points. To be more specific, the classification accuracy and reliability of almost all classes are higher using Combination 1 as input. Compared to Combination 1, Combination 3 leads to much more misclassification for all classes except WAT. When Combination 3 is implemented, noticeably more TRE, BAL, BUD and OIS ground truth pixels are mistakenly categorized as other classes while more other ground truth pixels are inaccurately classified as ROD, BAL, BUD and OIS in the predicted map. Moreover, Combination 3 makes it harder to distinguish OIS pixels from others, especially from TRE and BAL pixels. To summarize, the multispectral ALS data is superior to typical single-wavelength ALS data in deep-learning LC classification, which is also matching to the results obtained from others mentioned in Section 2.1.

#### **4.3.1.3 Combination 4 versus Combination 1**

Compared to information included in Combination 1, extra spectral information of the first returns is added in the Combination 4. Based on Table 4.9, WAT, TRE and ROD have the highest PA and OA, suggesting that the classification accuracy and reliability of these classes are the

highest. It also shows that more than 98% of the WAT, TRE and ROD ground truth pixels also respectively appear in these three classes in the predicted result; more than 96% of the WAT, TRE and ROD pixels in the classified image actually represent the three classes on the ground, respectively. The classification accuracy and reliability of BAL and BUD are also high. Nonetheless, OIS has the lowest classification accuracy and the lowest reliability. Only about 62% of OIS ground truth pixels are correctly displayed in the predicted map. In addition, only less than 75% of OIS pixels in the categorized image are true OIS pixels on the ground. Specifically, a number of OIS pixels are wrongly classified as TRE, ROD and BAL; many TRE and BAL pixels are erroneously categorized as OIS.

According to Table 4.5, OA of Combination 4 is on average 0.5% higher than that of Combination 1, revealing that the added spectral information of the first returns can improve abundance of input data. With the added spectral information, Combination 4 provides more relevant information for CNNs. Furthermore, comparing Figure 4.8 (a) and (d) with the labelled dataset, it can also be found that Combination 4 performs better than Combination 1. Specifically, both of the two inputs accomplish good classification results for WAT, TRE and ROD as shown in Table 4.6 and Table 4.7. Moreover, when Combination 1 is applied, UA of BUD and PA of OIS are higher, indicating that fewer other pixels are misclassified as BUD in the predicted map and less OIS ground truth pixels are classified to incorrect classes. Nevertheless, when Combination 4 is used, PA of BAL and BUD are obviously higher; UA of OIS is also higher. It indicates that fewer BAL and BUD ground truth pixels are mistakenly classified as other classes; fewer other pixels are misclassified as OIS in the predicted map. In general, Combination 4 results in less misclassification than Combination 1, which suggests that using the classical input data of rasterized multispectral ALS data and the extra spectral information of the first returns together can achieve better results in deep-learning LC classification than using the former solely.

#### **4.3.1.4 Combination 5 versus Combination 1 and 4**

In the Combination 5, more height information is added to Combination 1. It can be observed from Table 4.10 that WAT, TRE and ROD have the highest classification accuracy and the highest reliability in the classified image when the input is Combination 5. It shows that more than 97% of the WAT, TRE and ROD ground truth pixels also appear in corresponding classes in the predicted map; more than 97% of the WAT, TRE and ROD pixels in the classified image actually represent corresponding classes on the ground. Whereas OIS has the lowest classification accuracy

and reliability. Specifically, only about 69% of OIS ground truth pixels are displayed in the right class in the predicted image. Additionally, less than 70% of OIS pixels in the categorized map actually represent this class on the ground. Plentiful OIS pixels are mistakenly classified as TRE and BAL; a mass of other pixels is inaccurately categorized as OIS.

As shown in Table 4.5, an OA of Combination 5 is similar to that of Combination 1 no matter which CNN is implemented, meaning that the added height information does not improve the information in classical input data of the rasterized multispectral ALS dataset. Furthermore, the fact that an OA of Combination 5 is 0.5% lower than that of Combination 4 also demonstrates that the added height information is not as helpful as the added spectral information of the first returns for CNN-based LC classification. Also, details in Figure 4.8 illustrate these two findings. Specifically, PA and UA of all classes except BUD are highly similar between Combination 1 and Combination 5. Some more BUD ground truth pixels are misclassified as TRE and OIS when Combination 1 is applied; while few more actual TRE, BAL and OIS pixels are mistakenly classified as BUD when Combination 5 is used. Furthermore, compared to Combination 4, the classification accuracy of all classes except OIS is lower when the input is Combination 5. Although when Combination 5 is used, less reference OIS pixels are classified to wrong classes, while more other pixels are misclassified as OIS in the predicted image. Thus, OIS is less reliable when Combination 5 is used. Generally, Combination 5 leads to more misclassification than Combination 4; it has similar performance with Combination 1. It indicates that the added height information is less helpful than the added spectral information of the first returns for CNN-based LC classification.

#### **4.3.1.5 Combination 6 versus Combination 1 and 4**

All the nine extracted information layers are involved in the Combination 6. Similar to the result of Combination 5, WAT, TRE and ROD have the highest classification accuracy and reliability when Combination 5 is used (see Table 4.11), while OIS has the lowest classification accuracy and reliability. More than 97% of the WAT, TRE and ROD ground truth pixels are correctly classified to corresponding classes; more than 97% of the WAT, TRE and ROD pixels in the classified image actually represent matching classes on the ground. Only about 73% of actual OIS pixels are properly displayed in the classified map. Also, around 70% of OIS pixels in the predicted image are true OIS pixels on the ground. Specifically, plentiful OIS pixels are inaccurately categorized as TRE and BAL; a mass of other pixels is wrongly classified as OIS.

Dissimilar to the result of Combination 5, reliability of BAL is significantly higher than its classification accuracy, revealing that even though more than 92% of the pixels identified as BAL in the classified map are actual BAL pixels, only 81% of the reference BAL pixels have been correctly identified as BAL. True BAL pixels misclassified as other classes are more than pixels mistakenly classified as BAL.

It can be seen in Table 4.5 that OA of Combination 6 is on average 0.5% higher than them of Combination 1 and on average 0.1% lower than that of Combination 4. This fact reveals that although adding both the extra height information and the additional spectral information can improve CNN-based LC classification of Combination 1, it is not as helpful as adding the additional spectral information solely. Furthermore, adding supplementary height layers to Combination 4 deteriorates the performance of Combination 4. It indicates that the improvement from the result of Combination 1 to the result of Combination 6 is contributed to the added spectral information. Details in Figure 4.8 also illustrate these findings. To be more specific, the classification accuracy of all classes except BAL is higher when Combination 6 is applied, compared to Combination 1. Besides, even though more reference BAL pixels are classified to incorrect classes when use Combination 6 as input, less other ground truth pixels are misclassified as BAL. Thus, BAL is more reliable when Combination 6 is applied. Moreover, reliability of all other classes is similar in terms of Combination 1 and 6. Additionally, the main reason why the classification accuracy of Combination 6 is slightly lower than that of Combination 4 is that more BAL ground truth pixels are mistakenly categorized as TRE and OIS. In short, instead of improving, the added height information is even deteriorating classification performance of Combination 4.

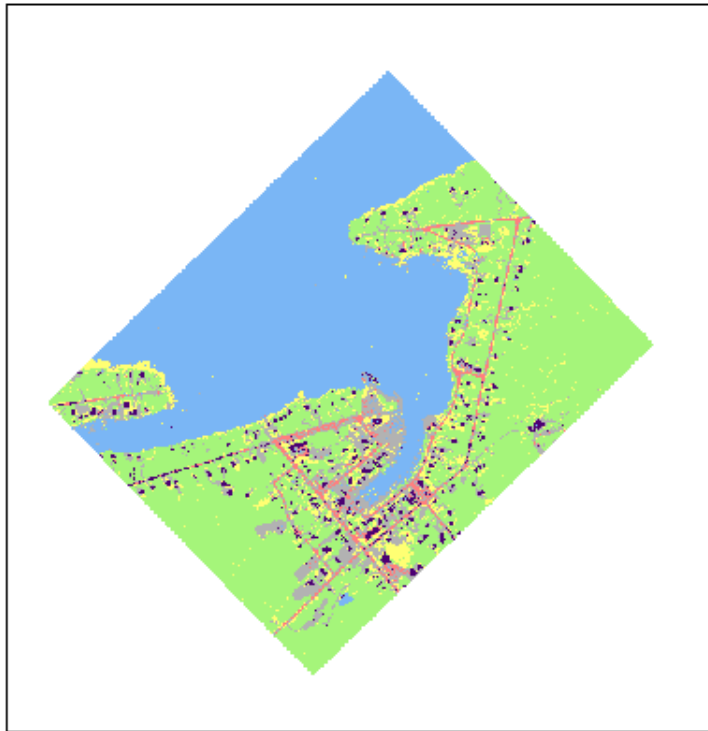
#### **4.3.2 Performances of Different CNNs**

According to Table 4.5, the 3D CNN achieves the highest OA and kappa coefficient, indicating that it has a high ability for pixel-wise LC classification and performs significantly better than random. When Combination 4 and the 3D CNN are applied, 97.2% of pixels in the study area are correctly classified. OA and kappa coefficient of the 2D CNN are only slightly lower than them of the 3D CNN, suggesting that the 2D CNN also performs well in pixel-wise LC classification. Using Combination 4 as the input, OA of the 2D CNN is only about 0.1% lower than OA of the 3D CNN. Conversely, the 1D CNN achieves the lowest OA and kappa coefficient. When the input is Combination 4, the 1D CNN attains its highest OA of 91.2%; however, it is still

approximately 6% lower than the OA of the 2D or 3D CNN.

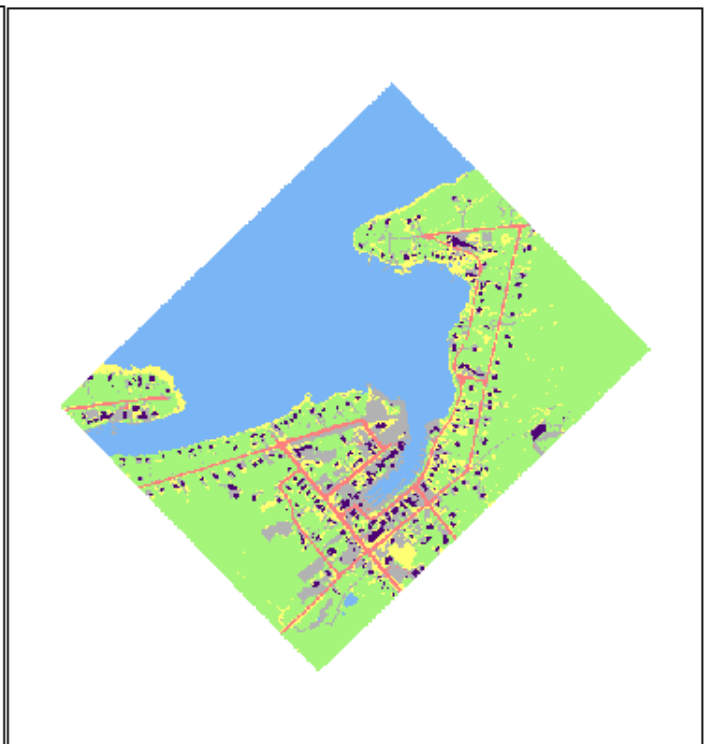
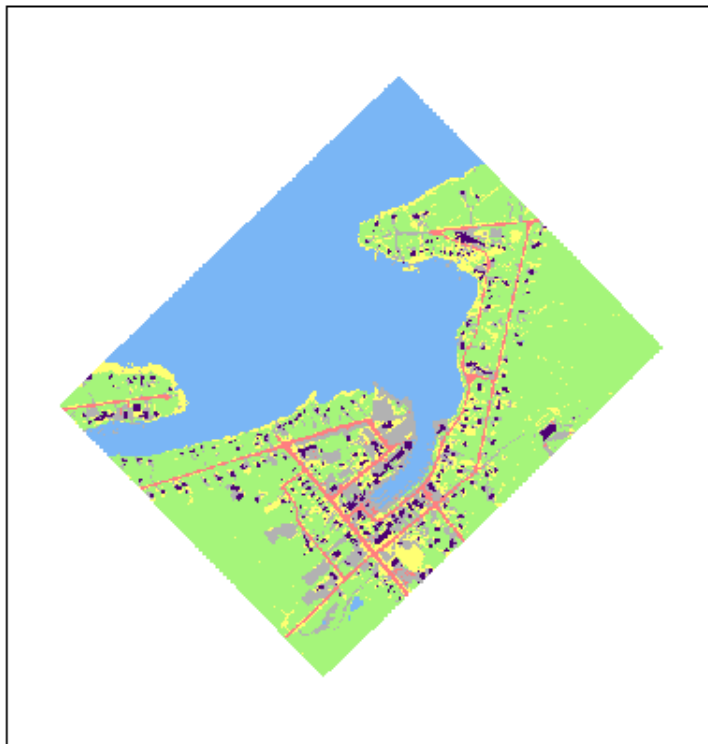
To analyze performances of the 1D, 2D and 3D CNNs in detail in terms of every LC class, confusion matrixes with computed UA and PA of Combination 4 with different CNNs are segmented and regrouped into two kinds of table: UA tables (see Table 4.12, 4.13 and 4.9) and PA tables (see Table 4.12, 4.13 and 4.9). UA tables are designed from the point of view of a map user but not the map maker. A UA table describes the reliability of each CNN in terms of a specific predicted class. Each row of it represents the statistics in the class, with calculated UA, when a CNN is used. PA tables are designed from the point of view of a map maker. A PA table explains the classification accuracy of each CNN in terms of a specific actual class. Each row of producer's table represents the statistics in the class with calculated PA when a CNN is used. These tables clearly state how well the CNNs perform in terms of each LC class and where misclassifications occur frequently. Predicted images of the three CNNs are also shown in Figure 4.9, which visualizes the classification results of the best models. To make logic unambiguous, details of these images will be analyzed separately for each class, integrated with information provided by UA tables and PA tables.





- Water
- Trees
- Roads
- Bare Land
- Buildings
- Other Impervious Surfaces

(a) the 1D CNN



(b) the 2D CNN

(c) the 3D CNN

**Figure 4.9** Predicted maps of different CNNs with Combination 4

#### 4.3.2.1 Performances of CNNs for WAT

Table 4.12 shows that all the three CNNs have high reliability for WAT; more than 98% of the WAT pixels in the three classified images actually represent WAT on the ground. Compared to the 2D and 3D CNNs, more than six times of other pixels are wrongly classified as WAT in the predicted map when the 1D CNN is used. Specifically, the misclassifications mainly occur on the BAL and OIS classes; more than 95% of the false WAT pixels are actual BAL and OIS pixels. Furthermore, it is noteworthy that some ROD ground truth pixels are incorrectly classified to WAT when the 1D CNN is applied while no ROD pixel is wrongly categorized as WAT when the other two CNNs are implemented. Moreover, according to Table 4.13, the classification accuracy of these three CNNs is high in terms of WAT; more than 99% of the WAT ground truth pixels are correctly classified to WAT in the three classified images. Additionally, when the 1D CNN is used, about four times as many as WAT ground truth pixels are misclassified to wrong classes. More than 90% of the misclassified WAT ground truth pixels belong to TRE, BAL and OIS in the classified images. In addition, it is also notable that, applying the 1D CNN, some true WAT pixels are incorrectly classified to ROD; however, the other two CNNs do not lead to this kind of misclassification. Therefore, it seems that the 2D and 3D CNNs can extremely clearly distinguish WAT from ROD, which cannot be done by the 1D CNN. Besides, comparing Figure 4.9 with the labelled map, it can be visibly found that there are many BAL pixels mistakenly appearing in the WAT area and many WAT pixels incorrectly occur in the region of OIS when the 1D CNN is applied. Generally speaking, all the three CNNs have high reliability and classification accuracy for WAT. Also, the 2D and 3D CNNs can clearly differentiate WAT from others especially from ROD, which cannot be done by the 1D CNN. The 1D CNN leads to more misclassification between WAT and all other classes especially BAL and OIS.

**Table 4.12** UA of the predicted WAT for Combination 4 and each CNN

		Actual						Total	UA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Predicted WAT	1D CNN	239357	117	129	1726	160	2107	243596	98.3
	2D CNN	240758	46	0	330	6	327	241467	99.7
	3D CNN	240780	41	0	236	11	336	241404	99.7

**Table 4.13** PA of the actual WAT for Combination 4 and each CNN

		Predicted						Total	PA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Actual WAT	1D CNN	239357	443	14	538	169	770	241291	99.2
	2D CNN	240758	66	0	193	4	270	241291	99.8
	3D CNN	240780	67	0	239	7	198	241291	99.8

**4.3.2.2 Performances of CNNs for TRE**

As shown in Table 4.14, all the three CNNs have high reliability for TRE; more than 93% of the TRE pixels in the three classified images actually represent TRE on the ground. Compared to the 2D and 3D CNNs, more than twice of other pixels are erroneously classified as TRE in the predicted images when the 1D CNN is used. Specifically, the misclassifications mainly occur on the BAL, BUD and OIS classes; more than 93% of the false TRE pixels are actual BAL, BUD and OIS pixels. Besides, according to Table 4.15, the classification accuracy of these three CNNs is also high for TRE; more than 96% of the TRE ground truth pixels are correctly classified to TRE in the classified images. Furthermore, nearly three times of TRE ground truth pixels are misclassified to wrong classes when the 1D CNN is used. More than 91% of the misclassified TRE ground truth pixels belong to BAL and OIS in the classified images. In addition, integrating these two tables, it can be shown that the hardest challenge is to discriminate TRE from BAL, especially for the 1D CNN. The reasons may be that BAL includes grass which has similar spectral reflectivity with TRE, and the height of TRE is varied. Thus, it is hard for CNNs to learn to define TRE itself and to distinguish short trees from the grass. Moreover, comparing Figure 4.9 with the labelled map, it can be visibly found that there are many BAL pixels erroneously appearing in the TRE area and many TRE pixels incorrectly occur in the region of OIS when the 1D CNN is implemented. After all, all the three CNNs have high reliability and classification accuracy for TRE. Nevertheless, it is relatively hard to differentiate TRE from BAL especially for the 1D CNN.

**Table 4.14** UA of the predicted TRE for Combination 4 and each CNN

		Actual						Total	UA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Predicted TRE	1D CNN	443	225606	578	9304	1718	3862	241511	93.4
	2D CNN	66	230223	129	5302	314	1667	237701	96.9
	3D CNN	67	228887	109	3102	287	1228	233680	98.0

**Table 4.15** PA of the actual TRE for Combination 4 and each CNN

		Predicted						Total	PA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Actual TRE	1D CNN	117	225606	60	4166	449	2733	233131	96.8
	2D CNN	46	230223	86	1368	371	1037	233131	98.8
	3D CNN	41	228887	71	2687	319	1126	233131	98.2

#### 4.3.2.3 Performances of CNNs for ROD

According to Table 4.16, these three CNNs have high reliability for ROD; more than 92% of the ROD pixels in the predicted images actually represent ROD on the ground. Compared to the 1D CNN, when the input is the 2D or 3D CNN, only around half of other pixels are mistakenly categorized as ROD in the classified image, especially BUD and OIS ground truth pixels. Additionally, in Table 4.17, the classification accuracy of the 2D and 3D CNNs is high for ROD while it of the 1D CNN is relatively low. Specifically, more than 98% of the ROD ground truth pixels are correctly classified to ROD in the classified images when the 2D and 3D CNNs are used while less than 83% of the ROD ground truth pixels are correctly classified to ROD in the predicted map when the 1D CNN is applied. Furthermore, more than twelve times of ROD ground truth pixels are misclassified to wrong classes when the 1D CNN is used. To be more specific, around 85% of the misclassified ROD ground truth pixels belong to BUD and OIS in the classified images. Moreover, integrating these two tables, it can be shown that the hardest challenge is to discriminate ROD from OIS, especially for the 1D CNN. It may be because both ROD and OIS are impervious surfaces which have the similar spectral reflectivity and altitude. The only difference is that these two classes may have different shape features. Although ROD has fixed shape features, shape features of OIS are varied. Furthermore, there are many cars on the ROD and OIS such as parking lots. Since the locations of cars keep changing all the time, they are ignored in the labelled dataset. However, these cars do exist in the multispectral ALS dataset, which may make the CNNs confused. Thus, it is hard for CNNs to distinguish ROD from OIS especially in the areas with cars. In addition, it is also relatively difficult for the 1D CNN to differentiate ROD from BUD. Nevertheless, the 2D and 3D CNNs can differentiate these two classes well. Thus, this fact states that the 1D CNN has a much lower capability of classification than the other two CNNs. Besides, as discussed in Section 4.3.2.1, the 2D and 3D CNNs can extremely clearly distinguish ROD from WAT, which cannot be done by the 1D CNN. Admittedly, comparing Figure 4.9 with the labelled

map, it can be visibly seen that the boundary of ROD is clearer and smoother when the 2D and 3D CNNs are used. In brief, the 2D and 3D CNNs perform better for classifying ROD while the 1D CNN results in more misclassifications between ROD and WAT, BUD, as well as OIS, respectively. Also, it is relatively hard to differentiate ROD from OIS for the proposed CNNs especially the 1D CNN.

**Table 4.16** UA of the predicted ROD for Combination 4 and each CNN

		Actual						Total	UA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Predicted ROD	1D CNN	14	60	39746	160	498	2662	43140	92.1
	2D CNN	0	86	47346	208	7	1033	48680	97.3
	3D CNN	0	71	47401	162	6	1587	49227	96.3

**Table 4.17** PA of the actual ROD for Combination 4 and each CNN

		Predicted						Total	PA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Actual ROD	1D CNN	129	578	39746	540	1189	5836	48018	82.8
	2D CNN	0	129	47346	187	7	349	48018	98.6
	3D CNN	0	109	47401	225	23	260	48018	98.7

#### 4.3.2.4 Performances of CNNs for BAL

From Table 4.18, it can be seen that the 2D and 3D CNNs have relatively high reliability for BAL while the 1D CNN has low reliability. Specifically, more than 87% of the BAL pixels in the predicted images actually represent BAL on the ground when the 2D and 3D CNNs are used; though, only about 74% of the BAL pixels in the classified maps actually represent BAL on the ground when the 1D CNN is applied. Additionally, compared to the 2D and 3D CNNs, when the 1D CNN is applied, more than twice of other pixels are incorrectly classified as BAL in the predicted map. These misclassifications mainly occur on the TRE and OIS classes; more than 81% of the false BAL pixels are actual TRE and OIS pixels. Furthermore, as shown in Table 4.19, the classification accuracy of the 2D and 3D CNNs is relatively high for BAL while it of the 1D CNN is low. To be more specific, more than 81% of the BAL ground truth pixels are correctly classified to BAL in the classified images when the 2D and 3D CNNs are used; nonetheless, only 61% of the BAL ground truth pixels are correctly classified to BAL in the predicted map when the 1D CNN is applied. Moreover, more than twice of BAL ground truth pixels are misclassified to wrong

classes when the 1D CNN is used. In detail, more than 95% of the misclassified BAL ground truth pixels are WAT, TRE and OIS pixels in the classified images. In addition, it can be found from these two tables that the hardest tasks are distinguishing BAL from TRE and OIS, especially for the 1D CNN. The reasons why it is difficult to separate BAL and TRE have been discussed in Section 4.3.2.2. It is also difficult to discriminate between BAL and OIS. It may be because rock areas which are included in BAL have a similar spectral reflectivity and altitude with OIS. Besides, the shape features of these two classes are various and irregular in the same way. Thus, it is hard for CNNs to distinguish BAL from OIS. Admittedly, comparing Figure 4.9 with the labelled map, it can be visibly found that there are many BAL pixels mistakenly appearing in the TRE and OIS regions. To summarize, the 2D and 3D CNNs perform better for BAL. Also, it is relatively hard to differentiate BAL from TRE and OIS particularly for the 1D CNN.

**Table 4.18** UA of the predicted BAL for Combination 4 and each CNN

		Actual						Total	UA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Predicted BAL	1D CNN	538	4166	540	25603	558	3003	34408	74.4
	2D CNN	193	1368	187	33918	132	1083	36881	92.0
	3D CNN	239	2687	225	36983	175	1727	42036	88.0

**Table 4.19** PA of the actual BAL for Combination 4 and each CNN

		Predicted						Total	PA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Actual BAL	1D CNN	1726	9304	160	25603	296	4766	41855	61.2
	2D CNN	330	5302	208	33918	396	1701	41855	81.0
	3D CNN	236	3102	162	36983	294	1078	41855	88.4

#### 4.3.2.5 Performances of CNNs for BUD

According to Table 4.20, the 2D and 3D CNNs have relatively high reliability for BUD while the 1D CNN has low reliability. Specifically, more than 90% of the BUD pixels in the predicted images actually represent BUD on the ground when the 2D and 3D CNNs are used; though, only about 76% of the BUD pixels in the classified maps actually represent BUD on the ground when the 1D CNN is applied. Furthermore, compared to the 1D CNN, when the input is the 2D or 3D CNN, about one third of other pixels are mistakenly categorized as BUD in the classified image. These misclassifications mainly occur on the ROD and OIS classes; around 75% of the false BUD

pixels are actual ROD and OIS pixels. Additionally, in Table 4.21, the classification accuracy of the 2D and 3D CNNs is about 28% higher for BUD than that of the 1D CNN. To be more specific, more than 94% of the BUD ground truth pixels are correctly classified to BUD in the classified images when the 2D and 3D CNNs are used; nonetheless, only less than 67% of the BUD ground truth pixels are correctly classified to BUD in the predicted map when the 1D CNN is applied. Moreover, about six times of BUD ground truth pixels are misclassified to all other classes when the 1D CNN is used. In detail, about 80% of the misclassified BUD ground truth pixels are TRE and OIS pixels in the classified images. Besides, integrating these two tables, it can be seen that it is hard to discriminate between BUD and OIS, especially for the 1D CNN. It may be because these two classes have similar spectral reflectivity. Also, it is worth noting that when the 1D CNN is applied, some true WAT and ROD pixels are incorrectly classified to BUD, and some BUD ground truth pixels are mistakenly categorized as WAT and ROD. Nonetheless, these misclassifications hardly occur when the 2D or 3D CNN is implemented. It seems that the 2D and 3D CNNs can very clearly distinguish BUD from WAT and ROD, which cannot be done by the 1D CNN. This fact states that the 1D CNN has a much lower capability of classification than the other two CNNs. Admittedly, comparing Figure 4.9 with the labelled map, it can be visibly seen that the boundary of BUD is clearer and smoother when the 2D and 3D CNNs are used. To conclude, the 2D and 3D CNNs perform well in terms of BUD while the 1D CNN results in more misclassifications between BUD and WAT, ROD, as well as OIS, respectively. It is relatively tough to distinguish BUD from OIS especially for the 1D CNN.

**Table 4.20** UA of the predicted BUD for Combination 4 and each CNN

		Actual						Total	UA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Predicted BUD	1D CNN	169	449	1189	296	11686	1546	15335	76.2
	2D CNN	4	371	7	396	16466	891	18135	90.8
	3D CNN	7	319	23	294	16499	854	17996	91.7

**Table 4.21** PA of the actual BUD for Combination 4 and each CNN

		Predicted						Total	PA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Actual BUD	1D CNN	160	1718	498	558	11686	2871	17491	66.8
	2D CNN	6	314	7	132	16466	566	17491	94.1
	3D CNN	11	287	6	175	16499	513	17491	94.3

#### 4.3.2.6 Performances of CNNs for OIS

It can be found from Table 4.16 that all the three CNNs have low reliability for OIS especially the 1D CNN. Specifically, less than 75% of the OIS pixels in the predicted images actually represent OIS on the ground when the 2D and 3D CNNs are used; only less than 11% of the OIS pixels in the classified maps actually represent OIS on the ground when the 1D CNN is applied. When the 1D CNN is implemented, more than 89% of OIS pixels in the predicted result should belong to other classes. When the 2D or 3D CNN is used, about 70% of misclassifications occur in TRE and BAL. In addition, as shown in Table 4.17, the classification accuracy of these three CNNs is low for OIS, especially of the 1D CNN. To be more specific, less than 67% of the OIS ground truth pixels are correctly classified to OIS in the classified images when the 2D and 3D CNNs are used; only less than 14% of the OIS ground truth pixels are correctly classified to OIS in the predicted map when the 1D CNN is applied. Moreover, more than twice of OIS ground truth pixels are misclassified to wrong classes when the 1D CNN is used. When the 2D or 3D CNN is applied, a mass of OIS ground truth pixels are erroneously categorized as all other classes except WAT. Besides, it can be shown from these two tables that it is challenging to differentiate OIS from all other classes except WAT, especially for the 1D CNN, as discussed in the previous sections. Obviously, the three CNNs, especially the 1D CNN, perform worst for OIS among all classes. It is difficult for these CNNs to differentiate OIS from other classes, especially for the 1D CNN.

**Table 4.22** UA of the predicted OIS for Combination 4 and each CNN

		Actual						Total	UA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Predicted OIS	1D CNN	770	2733	5836	4766	2871	2009	18985	10.6
	2D CNN	270	1037	349	1701	566	10188	14111	72.2
	3D CNN	198	1126	260	1078	513	9457	12632	74.9

**Table 4.23** PA of the actual OIS for Combination 4 and each CNN

		Predicted						Total	PA (%)
		WAT	TRE	ROD	BAL	BUD	OIS		
Actual OIS	1D CNN	2107	3862	2662	3003	1546	2009	15189	13.2
	2D CNN	327	1667	1033	1083	891	10188	15189	67.1
	3D CNN	336	1228	1587	1727	854	9457	15189	62.3



### 4.3.3 Efficiency of Different CNNs

As shown in Table 4.24, the number of model parameters and running time increase significantly with the growth of dimensions of CNNs. All types of running time was calculated based on the hardware listed in Section 3.7. The number of model parameters of the 3D CNN is about three times more than that of the 2D CNN and around eight times more than that of the 1D CNN. The average running time per training epoch, total training time and total time of prediction of the 3D CNN are approximately five times longer than those of the 2D CNN and at least ten times more than those of the 1D CNN. As discussed in previous sections, the 2D and 3D CNNs perform much better than the 1D CNN in terms of OA, kappa coefficient, predicted maps, the classification accuracy of each class, or the reliability of each class. However, the overall performance of the 3D CNN is only slightly better than that of the 2D CNN. Hence, it can be concluded that the 2D CNN is the best choice for classification of multispectral ALS data from the view of efficiency.

**Table 4.24** Total model parameters and running time of each CNN with Combination 4

	Total parameters	Average time per epoch (second)	Training time (second)	Prediction time (second)
1D CNN	6,324,230	23	692	274
2D CNN	16,826,374	47	1406	605
3D CNN	50,362,374	241	7233	3169

## 4.4 Comparison of LC Classifications for Multispectral ALS Data

A comparative study of LC classification methods for multispectral ALS data is carried out. Results achieved by Combination 4 and the three CNNs are used for the comparisons. The classification accuracy of proposed CNNs are compared with that of three widely used traditional classification methods (i.e. MLC, SVM, and RF). All involved studies are summarized in Table 4.25.

**Table 4.25** Studies of LC Classification Methods for Multispectral ALS Data

Classification Methods	Author	Classes	OA (%)
CNN	This study	6	97.2
MLC	Bakuła et. al., (2016)	6	90.9
	Fernandez-Diaz et. al., (2016)	5	90.2
	Morsy et. al., (2017a)	4	89.9
SVM	Teo and Wu, (2017)	5	96
RF (Decision Trees)	Zou et al. (2016)	6	95.9
	Matikainen et. al. (2016; 2017a; 2017b)	9	91.6

MLC algorithm, which assigns cells a LC class based on the measure of the highest likelihood, were applied in three studies to map LC classes for multispectral ALS data. Bakuła et. al. (2016) used a raster-based MLC to classify a multispectral ALS point cloud into six classes, achieving an overall accuracy of 91% in the best test. In this attempt, WAT, TRE, and BUD were classified accurately; however, BLD and OIS were misclassified (Bakuła et. al. 2016). Furthermore, Morsy et al. (2017a) also applied a raster-based MLC method to classify multispectral ALS data to four classes and obtained an overall accuracy of 89.9%. Similarly, Fernandez-Diaz et al. (2016) implemented a raster-based MLC to categorize a multispectral ALS dataset into five LC classes with best overall accuracy of 90.2%. To conclude, OA of MLC methods is slightly lower than that of the 1D CNN and at least 6% lower than the 2D and 3D CNNs. The reason might be that the MLC assumes that a training sample is normally distributed, which is often not the case. This incorrect assumption can introduce errors especially when classifying urban landscapes.

SVM applies optimization algorithms to determine the location of ideal boundaries that can most effectively distinguish between classes (Huang et al., 2002). An object-based SVM classification method was tested by Teo and Wu (2017) to categorize multispectral ALS data into five classes, achieving an overall accuracy of 96%. Although OA of the SVM is only slightly lower than that of the 2D and 3D CNNs, SVM classification still has a major limitation since the selection of the kernel function and the setting of proper parameter values are decided subjectively by the user and only few studies have been conducted on the determination of the optimal choice of kernel function and proper settings for corresponding parameter (Petropoulos et al, 2012).

The RF method is a collection of Decision Trees, which are the predictive model that uses a set of binary rules as nodes to acquire a best solution. An object-based decision tree model was

implemented to multispectral ALS data by Zou et al. (2016) to accomplish a 9-class LC classification, reaching an overall accuracy of 91.6%. However, the decision tree algorithm tends to over-fit training data, especially when a tree is particularly deep. Matikainen et al. published several articles (2016; 2017a; 2017b) on the application of an object-based RF LC classification method to multispectral ALS datasets, which achieved an OA of 95.9% for six classes. This method performed better for BUD, TRE and OIS, but lead to low correctness for BAL. The OA of the RF method only slightly lower than that of the 2D and 3D CNNs. Also, the large number of trees in the RF method may make classification process slow, especially when applied to a large dataset such as a dense multispectral ALS point cloud in a large area.

To conclude, the 2D and 3D CNNs proposed in the study can achieve higher LC classification accuracy for multispectral ALS data than the traditional classification methods especially the MLC. Although as classic machine learning algorithms, SVM and RF can provide relatively precise and reliable classification results for multispectral ALS data, both of them require hand-designed features which significantly impact the classification accuracy. This characteristic of classic machine learning algorithms makes them highly user-dependent. This limitation of SVM and RF also can be conquered by all deep-learning networks like the CNNs proposed in the study.

## **4.5 Chapter Summary**

This chapter presented and discussed the results derived from the stepwise processes proposed in Chapter 3. In this chapter, the labelled dataset was shown and validated. The validation result certifies that the labelled dataset is reliable without errors brought by the inertia of thinking. Moreover, values of hyper-parameters involved in the establishment and implementation processes of CNNs were discussed and determined. Using these hyper-parameters, classification results of eighteen models were obtained and further discussed. The highest overall classification accuracy of 97.2% with a kappa index of 0.96 was achieved using the proposed 3D CNN and input data Combination 4. It is a significant classification result, which is better than most of the published multispectral ALS data classification results. As regards different input datasets, the three author-designed input datasets performed better than classic input datasets. Results reveal that the multispectral ALS data is superior to both traditional multispectral optical imagery and typical single-wavelength ALS data in LC classification. Compared to the typical information extracted from rasterized multispectral ALS dataset, the added height information is less helpful than the added spectral information of the first returns for CNN-based LC classification. Instead

of improving, the added height information even deteriorates the classification performance of Combination 4. For proposed models, the 2D and 3D CNNs perform much better than the 1D CNN, no matter from the perspective of OA, kappa coefficient, predicted maps, the classification accuracy of each class, or the reliability of each class. Furthermore, the overall accuracy of the 2D CNN is only at most 0.3% lower than that of the 3D CNN for each input; however, the number of parameters and run time in the 2D CNN only account for about one third and one fifth in the 3D CNN, respectively. Thus, from the view of efficiency, the 2D CNN is the best choice for multispectral ALS classification.

## **Chapter 5**

### **Conclusions and Recommendations**

This chapter summarizes the main findings and contributions of this thesis. The limitations and corresponding recommendations of the proposed methodology are also discussed for further studies.

#### **5.1 Conclusions and Contributions**

This study proposed a workflow for an automated pixel-wise LC classification for multispectral ALS data using deep-learning methods. A total of six input datasets were formed with multi-tiered architecture and three CNNs were proposed to seek an optimal scheme. The results presented in this thesis show that the LC classification accuracy can be improved considerably by using the multispectral ALS data and deep learning. An overall classification accuracy of 97.2%, with a kappa index of 0.96, was achieved using the proposed 3D CNN with Combination 4. It represents a significant classification accuracy since it is on average 4% higher than the accuracy of the published multispectral ALS LC classifications. Generally, this thesis presents the feasibility of combining, for the first time, multispectral ALS data and deep learning to improve the performance of the automatic pixel-wise LC classification. The proposed methodology can map LC classes accurately, efficiently, and automatically, which eliminates errors introduced by data fusion, fills some gap in research, and reduces the challenges of LC supervision induced by rapid and global urbanisation. It is significant as accurate and update LC information becomes increasingly critical for protecting ecosystems, climate change studies and sustainable human-environment development. Moreover, a LC map plays a significant role in policy-making since inaccurate LC maps may lead to inappropriate policies (e.g. Ittersum et. al., 1998). This study pioneers a new direction for the improvement of LC classification.

Furthermore, this study analyzed how different information extracted from the multispectral ALS data impacts the classification accuracy by comparing various input data combinations. The fact, the OA of the designed input datasets was on average 3.8% higher than that of classic input datasets, reveals that either the proposed additional spectral information of the first returns or height information can improve classification accuracy of classic input datasets. The added spectral information is more helpful than the added height information for CNN-based LC

classification. However, when using the additional information together, the added height information deteriorates classification performance when using the additional spectral information solely. Therefore, the optimal rasterized multispectral ALS dataset for LC classification should consist of height information of the first returns and spectral information of the first returns and all returns. These findings may inspire researchers to explore more possibilities of current multispectral ALS datasets and to extract more useful information from them. Furthermore, results of the classic extracted information of multispectral ALS data were compared respectively with results of the simulative spectral information of traditional multispectral optical images and results of the simulative spatial information of typical single-wavelength ALS data. This comparison reveal that the multispectral ALS technique is superior to both traditional multispectral optical imagery and typical single-wavelength ALS data for LC classification. The thesis indicates the potential of multispectral ALS data in LC mapping, which may draw more people' attention to this new technique. Once the multispectral ALS data become widely available, more multispectral ALS data with professional labelling datasets will be published for researchers to investigate, which may further improve LC classification. Consequentially, it may accelerate the development of multispectral ALS techniques.

In this study, three CNNs (i.e. 1D CNN, 2D CNN and 3D CNN) were established, trained, validated and tested to assess how CNNs with different dimensions can affect the classification accuracy of multispectral ALS data. According to the prediction results, the 2D and 3D CNNs performed much better than the 1D CNN regardless of the different perspectives (i.e. overall accuracy, kappa coefficient, predicted maps, the producer's accuracy of each class, and the user's accuracy of each class). It seems that the 2D and 3D CNNs can clearly distinguish between WAT, ROD and BUD, which cannot be done by the 1D CNN. It is particularly difficult for the 1D CNN to differentiate OIS from all other classes. Moreover, the overall accuracy of the 2D CNN is only at most 0.3% lower than the 3D CNN for each input. However, the number of parameters and run time in the 2D CNN only accounts for about one third and one fifth in the 3D CNN, respectively. Thus, from the view of efficiency, the 2D CNN is the best choice for multispectral ALS classification. According to related studies, deep-learning methods are superior to conventional classification methods for LC mapping when using many other kinds of data such as hyperspectral images (e.g. Makantasis et al., 2015; Kussul et al., 2017; Ghamisi et al., 2016). This thesis identifies potential of deep-learning classification methods when applying a new type of data (i.e.

multispectral ALS data). It further proves the power of deep learning in LC classification, which may attract more researchers to consider deep-learning-based LC classification. As a result, a pre-trained deep-learning system of LC classification may be developed. The pre-trained deep-learning system of LC classification can achieve a rapid, accurate and automated classification for any uploaded image without setting any parameter or hyper-parameter. It is similar to the Google released pre-trained deep-learning system of object Detection (Huang, 2017), in which objects can be detected automatically once an image is submitted. The emergence of this system can help many geographers whose research require accurate LC maps and can save their energy and time.

## **5.2 Limitations and Recommendations**

The classification accuracy of the proposed models is limited due to the following reasons. Firstly, labelling work needs abundant experience. Although the validation result identifies that the labelled dataset is reliable without errors brought by the inertia of thinking, it should be closer to reality if it can be completed by a group of experts. Also, since the labelled data should have the same data structure with the input data and it is an impossible job for an individual to complete point-wise labelling, the input datasets cannot be anything but raster images. The point-based classification cannot be chosen in this study. This limitation is unavoidable in this thesis but will be eliminated soon because more professional labelled datasets will be published once more people's attention is attracted to this new technique by this thesis. It is strongly recommended that researchers test if using point-based deep-learning classification methods and multispectral ALS data can improve LC mapping once point-wise labelled datasets are available. Secondly, the multispectral ALS data used in the study was collected by an Optech Titan system which has only three channels. It limits the amount and diversity of information that can be extracted from the raw data. With the development of hyperspectral ALS technology, more useful information can be extracted to improve LC classification accuracy. Moreover, the three established CNNs in this study are foundational CNNs which require less powerful GPU to run. More complicated CNN-based models (e.g. fully connected network) were not implemented because of the limitation of the hardware. To further improve LC classification accuracy, more complicated CNN-based models are recommended to test in the future studies.

## REFERENCES

- Ahokas, E., Hyypä, J., Yu, X., Liang, X., Matikainen, L., Karila, K., Litkey, P., Kukko, A., Jaakkola, A., Kaartinen, H., Holopainen, M., & Vastaranta, M. (2016). Towards automatic single-sensor mapping by multispectral airborne laser scanning. *ISPRS Archives*, XLI-B3, 155-162.
- Antonarakis, A. S., Richards, K. S., & Brasington, J. (2008). Object-based land cover classification using airborne LiDAR. *Remote Sensing of Environment*, 112(6), 2988-2998.
- ArcGIS Desktop Help. (n.d.). LAS Dataset to Raster Function. Retrieved from <http://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/las-dataset-to-raster-function.htm>
- Bakula, K. (2015). Multispectral airborne laser scanning-a new trend in the development of LiDAR technology. *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, 27.
- Bakula, K., Kupidura, P., & Jelowicki, L. (2016). Testing of land cover classification from multispectral airborne laser scanning data. *ISPRS Archives*, XLI-B7, 161-169.
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798-1828.
- Breiman, L., (2001). Random forests. *Machine Learning*, 45(1), 5-32.
- Chen, J. X. (2016). The evolution of computing: AlphaGo. *Computing in Science & Engineering*, 18(4), 4-7.
- Chen, X., Chengming, Y. E., Li, J., & Chapman, M. A. (2018). Quantifying the Carbon Storage in Urban Trees Using Multispectral ALS Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, (99), 1-8.
- Chen, Y., Lin, Z., Zhao, X., Wang, G., & Gu, Y. (2014). Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing*, 7(6), 2094-2107.
- Chen, Y., Zhao, X., & Jia, X. (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), 2381-2392.



- Cihlar, J. (2000). Land cover mapping of large areas from satellites: status and research priorities. *International Journal of Remote Sensing*, 21(6-7), 1093-1114.
- Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence Vol. 22, No. 1*, 1237.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37-46.
- Deng, L. (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3, e2.
- Duan, Y., Liu, F., Jiao, L., Zhao, P., & Zhang, L. (2017). SAR Image segmentation based on convolutional-wavelet neural network and markov random field. *Pattern Recognition*, 64, 255-267.
- Feddema, J. J., Oleson, K. W., Bonan, G. B., Mearns, L. O., Buja, L. E., Meehl, G. A., & Washington, W. M. (2005). The importance of LC change in simulating future climates. *Science*, 310(5754), 1674-1678.
- Fernandez-Diaz, J. C., Carter, W. E., Glennie, C., Shrestha, R. L., Pan, Z., Ekhtari, N., Singhania, A., Hauser, D., & Sartori, M. (2016). Capability assessment and performance metrics for the Titan multispectral mapping LiDAR. *Remote Sensing*, 8(11), 936.
- Ghamisi, P., Chen, Y., & Zhu, X. X. (2016). A self-improving convolution neural network for the classification of hyperspectral data. *IEEE Geoscience and Remote Sensing Letters*, 13(10), 1537-1541.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, 580-587.
- Glennie, C. L., Carter, W. E., Shrestha, R. L., & Dietrich, W. E. (2013). Geodetic imaging with airborne LiDAR: the Earth's surface revealed. *Reports on Progress in Physics*, 76(8), 086801.
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning (Vol. 1)*. Cambridge: MIT press. 11-69

- Guan H, Li J, Cao S, & Yu Y. (2016). Use of mobile LiDAR in road information inventory: a review, Taylor & Francis: SCI, International Journal of Image and Data Fusion, 7(3): 219-242. doi: 10.1080/19479832.2016.1188860
- Guida-Johnson, B., & Zuleta, G. A. (2013). Land-use land-cover change and ecosystem loss in the Espinal ecoregion, Argentina. Agriculture, Ecosystems & Environment, 181, 31-40.
- Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. Neurocomputing, 187, 27-48.
- Hebel, M., & Stilla U., (2012). Simultaneous calibration of ALS systems and alignment of multiview LiDAR scans of urban areas. IEEE Transactions on Geoscience and Remote Sensing 50(6), 2364-2379. doi: 10.1109/TGRS.2011.2171974.
- Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1(282-317), 2.
- Hou, B., Kou, H., & Jiao, L. (2016). Classification of polarimetric SAR images using multilayer autoencoders and superpixels. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9(7), 3072-3081.
- Hu, W., Huang, Y., Wei, L., Zhang, F., & Li, H. (2015). Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors, 2015, 12.
- Huang, C., Davis, L.S., & Townshend, J.R.G. (2002). An assessment of support vector machine for land cover classification. International Journal of Remote Sensing, 23 (4), 725-749
- Huang, J. (2017). Supercharge your computer vision models with the TensorFlow Object Detection API. Retrieved from <https://ai.googleblog.com/2017/06/supercharge-your-computer-vision-models.html>
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
- Ittersum, M. K., Rabbinge, R., & Latesteijn, H. C. (1998). Exploratory land use studies and their role in strategic policy making. Agricultural Systems, 58(3), 309-330.
- Kashani, A. G., Olsen, M. J., Parrish, C. E., & Wilson, N. (2015). A review of LiDAR radiometric processing: from ad hoc intensity correction to rigorous radiometric calibration. Sensors,

15(11), 28099-28128.

- Kim, Y., & Kim, Y. (2014). Improved classification accuracy based on the output-level fusion of high-resolution satellite images and airborne LiDAR data in urban area. *IEEE Geoscience and Remote Sensing Letters*, 11(3), 636-640.
- Kraus, K., & Pfeifer, N. (1998). Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(4), 193-203.
- Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5), 778-782.
- Li, Y., Zhang, H., & Shen, Q. (2017). Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sensing*, 9(1), 67.
- Liou, C. Y., Cheng, W. C., Liou, J. W., & Liou, D. R. (2014). Autoencoder for words. *Neurocomputing*, 139, 84-96.
- Lodha, S. K., Kreps, E. J., Helmbold, D. P., & Fitzpatrick, D. N. (2006). Aerial LiDAR data classification using support vector machines (SVM). *3DPVT*, 567-574.
- Lunetta, R.S., Ediriwickrema, J., Johnson, D.M., Lyon, J.G., & McKerrow, A. (2002). Impacts of vegetation dynamics on the identification of land-cover change in a biologically complex community in North Carolina, USA. *Remote Sens. Environ.* 82, 258-270.
- Luque, I. F., Aguilar, F. J., Álvarez, M. F., & Aguilar, M. Á. (2013). Non-parametric object-based approaches to carry out ISA classification from archival aerial orthoimages. *IEEE Journal of selected topics in applied earth observations and remote sensing*, 6(4), 2058-2071.
- Lv, Q., Dou, Y., Niu, X., Xu, J., Xu, J., & Xia, F. (2015). Urban land use and land cover classification using remotely sensed SAR data through deep belief networks. *Journal of Sensors*, 2015, 10.
- Makantasis, K., Karantzalos, K., Doulamis, A., & Doulamis, N. (2015). Deep supervised learning for hyperspectral data classification through convolutional neural networks. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 4959-4962.
- Matikainen, L., Hyypä, J., & Litkey, P. (2016). Multispectral airborne laser scanning for automated map updating. *ISPRS Archives*, XLI-B3, 323-330.

- Matikainen, L., Karila, K., Hyypä, J., Litkey, P., Puttonen, E., & Ahokas, E. (2017a). Object-based analysis of multispectral airborne laser scanner data for land cover classification and map updating. *ISPRS Journal of Photogrammetry and Remote Sensing*, 128, 298-313.
- Matikainen, L., Karila, K., Hyypä, J., Puttonen, E., Litkey, P., & Ahokas, E. (2017b). Feasibility of multispectral airborne laser scanning for land cover classification, road mapping and map updating. *ISPRS Archives*, XLII-3-W3, 119-122.
- MIT Technology Review. (2013). 10 Breakthrough Technologies 2013. Retrieved from: <https://www.technologyreview.com/lists/technologies/2013/>.
- Morsy, S., Shaker, A., & El-Rabbany, A. (2017a). Multispectral LiDAR data for land cover classification of urban areas. *Sensors*, 17(5), 958.
- Morsy, S., Shaker, A., & El-Rabbany, A. (2017b). Clustering of multispectral airborne laser scanning data using Gaussian decomposition. *ISPRS Archives*, XLII-2-W7, 269-276.
- National Research Council. (2005). Radiative forcing of climate change: expanding the concept and addressing uncertainties. The National Academies Press, Washington, DC, USA. 2-5
- Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1. *Vision Research*, 37(23), 3311-3325.
- Pal, M., and Mather, P.M. (2005). Support vector machines for classification in remote sensing. *International Journal of Remote Sensing*. 26 (5), 1007-1011
- Papadomanolaki, M., Vakalopoulou, M., Zagoruyko, S., & Karantza, K. (2016). Benchmarking deep learning frameworks for the classification of very high resolution satellite multispectral data. *ISPRS Annals*, 3, 83.
- Petropoulos, G.P., Kalaitzidis, C., and Vadrevu, K.P. (2012). Support vector machine and object-based classification for obtaining land-use/cover cartography from hyperion hyperspectral imagery. *Journal of Computer and Geoscience*. 41, 99-107
- Pontius Jr, R. G., & Millones, M. (2011). Death to Kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment. *International Journal of Remote Sensing*, 32(15), 4407-4429.
- Puente, I., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2013). Review of mobile

- mapping and surveying technologies. *Measurement*, 46(7), 2127-2145. doi: 10.1016/j.measurement.2013.03.006
- Pugh, C. (2014). *Sustainability the Environment and Urbanisation*. Routledge. 38-41
- Qin, F., Guo, J., & Sun, W. (2017). Object-oriented ensemble classification for polarimetric SAR imagery using restricted Boltzmann machines. *Remote Sensing Letters*, 8(3), 204-213.
- Romero, A., Gatta, C., & Camps-Valls, G. (2016). Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3), 1349-1362.
- Santara, A., Mani, K., Hatwar, P., Singh, A., Garg, A., Padia, K., & Mitra, P. (2017). BASS Net: band-adaptive spectral-spatial feature learning neural network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9), 5293-5301.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85-117.
- Scuba & H2O Adventure Magazine. (2017). the freshwater capital of the world: a Tobermory adventure. Retrieved from <http://www.divenewsnetwork.com/single-post/2017/03/14/The-Freshwater-Capital-of-the-World-A-Tobermory-Adventure>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.
- Steffen, W., Sanderson, R. A., Tyson, P. D., Jäger, J., Matson, P. A., Moore III, B., Oldfield, F., Richardson, K., Schellnhuber, H.J., Turner, B.L., & Wasson, R.J. (2006). *Global change and the earth system: a planet under pressure*. Springer Science & Business Media. 72-154
- Tao, C., Pan, H., Li, Y., & Zou, Z. (2015). Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geoscience and Remote Sensing Letters*, 12(12), 2438-2442.
- Teledyne Optech Titan, (2015). *Multispectral LiDAR system: high precision environmental mapping*, Retrieved from <http://www.teledyneoptech.com/wp-content/uploads/Titan-Specsheet-150515-WEB.pdf>.
- Teo, T. A., & Wu, H. M. (2017). Analysis of land cover classification using multi-wavelength

- Lidar system. *Applied Sciences*, 7(7), 663.
- The United Nations. (2015). 2014 revision of world urbanization prospects. New York. Retrieved from <https://esa.un.org/unpd/wup/>
- Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In 2015 IEEE International Computer Vision Conference (ICCV), 4489-4497.
- Vosselman G. & Maas H. G. (2010). *Airborne and Terrestrial Laser Scanning*, Whittles Publishing. 29-104
- Weng, Q. (2012). Remote sensing of impervious surfaces in the urban areas: requirements, methods, and trends. *Remote Sensing of Environment*, 117, 34-49.
- Wichmann V, Bremer M, Lindenberger J, Rutzinger M, Georges C, & Petrini-Monteferri F, (2015). Evaluating the potential of multispectral airborne LIDAR for topographic mapping and land cover classification. *ISPRS Annals*, II-3, 113-119. doi:10.5194/isprsannals-II-3-W5-113-2015.
- Wilkinson, G. G. (2005). Results and implications of a study of fifteen years of satellite image classification experiments. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3), 433-440.
- Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.
- Yan, W. Y., Shaker, A., & El-Ashmawy, N. (2015). Urban land cover classification using airborne LiDAR data: A review. *Remote Sensing of Environment*, 158, 295-310.
- Yan, W. Y., Shaker, A., Habib, A., & Kersting, A. P. (2012). Improving classification accuracy of airborne LiDAR intensity data by geometric calibration and radiometric correction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67, 35-44.
- Zeiler, M. D. (2013). *Hierarchical convolutional deep learning in computer vision*, PhD Thesis, Department of Computer Science, New York University.
- Zhang, J., Shan, S., Kan, M., & Chen, X. (2014). Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *European Conference on Computer Vision*. Springer, Cham. 1-16.

- Zhang, L., Ma, W., & Zhang, D. (2016). Stacked sparse autoencoder in PolSAR data classification using local spatial information. *IEEE Geoscience and Remote Sensing Letters*, 13(9), 1359-1363.
- Zhang, X., Chen, G., Wang, W., Wang, Q., & Dai, F. (2017a). Object-based land-cover supervised classification for very-high-resolution UAV images using stacked de-noising autoencoders. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(7), 3373-3385.
- Zhang, Z., Wang, H., Xu, F., & Jin, Y. Q. (2017b). Complex-valued convolutional neural network and its application in polarimetric SAR image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12), 7177-7188.
- Zhao, Z., Jiao, L., Zhao, J., Gu, J., & Zhao, J. (2017). Discriminant deep belief network for high-resolution SAR image classification. *Pattern Recognition*, 61, 686-701.
- Zhong, Z., Li, J., Luo, Z., & Chapman, M. (2018). Spectral-spatial residual network for hyperspectral image classification: a 3-d deep learning framework. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2), 847-858.
- Zhou, Y., Wang, H., Xu, F., & Jin, Y. Q. (2016). Polarimetric SAR image classification using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 13(12), 1935-1939.
- Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4), 8-36.
- Zou, X., Zhao, G., Li, J., Yang, Y., & Fang, Y. (2016). 3D land cover classification based on multispectral LiDAR point clouds. *ISPRS Archives, XLI-B1*, 741-747.