Extraction of Digital Terrain Models from Airborne Laser Scanning Data based on Transfer-Learning

by

Weiya Ye

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, 2019

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 rapid urbanization, timely and comprehensive urban thematic and topographic information is highly needed. Digital Terrain Models (DTMs), as one of unique urban topographic information, directly affect subsequent urban applications such as smart cities, urban microclimate studies, emergency and disaster management. Therefore, both the accuracy and resolution of DTMs define the quality of consequent tasks. Current workflows for DTM extraction vary in accuracy and resolution due to the complexity of terrain and off-terrain objects. Traditional filters, which rely on certain assumptions of surface morphology, insufficiently generalize complex terrain. Recent development in semantic labeling of point clouds has shed light on this problem. Under the semantic labeling context, DTM extraction can be viewed as a binary classification task.

This study aims at developing a workflow for automated point-wise DTM extraction from Airborne Laser Scanning (ALS) point clouds using a transfer-learning approach on ResNet. The workflow consists of three parts: feature image generation, transfer learning using ResNet, and accuracy assessment. First, each point is transformed into a feature image based on its elevation differences with neighbouring points. Then, the feature images are classified into ground and non-ground using ResNet models. The ground points are extracted by remapping each feature image to its corresponding points. Lastly, the proposed workflow is compared with two traditional filters, namely the Progressive Morphological Filter (PMF) and the Progress TIN Densification (PTD).

Results show that the proposed workflow establishes an advantageous accuracy of DTM extraction, which yields only 0.522% Type I error, 4.84% Type II error and 2.43% total error. In comparison, Type I, Type II and total error for PMF are 7.82%, 11.6%, and 9.48%, for PTD are 1.55%, 5.37%, and 3.22%, respectively. The root mean squared error of interpolated DTM of 1 m resolution is only 7.3 cm. Moreover, the use of pre-trained weights largely accelerated the training process and enabled the network to reach unprecedented accuracy even on a small amount of training set. Qualitative analysis is further conducted to investigate the reliability and limitations of the proposed workflow.

Acknowledgements

First and foremost, I would like to express my sincerest appreciation to my supervisor, Professor Dr. Jonathan Li, whose passion and professional knowledge guided me through the graduate study.

Second, with respect and admiration, I would like to show my gratefulness to my 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. Wanhong Yang, Professor at the Department of Geography, Environment and Geomatics, University of Guelph, for providing valuable suggestions of my work.

Furthermore, I would like to thank Ying Li, Ming Liu, Mengge Chen, Yue Gu, Zhuo Chen, Lingfei Ma and Gaoxiang Zhou, all the members in Waterloo Mobile Sensing and Geodata Analytics Lab, who shared their valuable knowledge and insight with me during group meetings. I would like to acknowledge the University of Waterloo Geospatial Centre for providing me with the ALS dataset. A special thank goes to my roommates Wanxue Meng and Chaojie Ou and Zhengfang Duanmu, for all the great times we spent together.

Special thanks go to the staff at WatXtract.ai for providing me two-terms of internship and the state-of-the-art computing facility. Thanks to Dr. Yong Hu for his patience and guidance towards my work. I also wish to thank all the staff at the Department of Geography and Environmental Management, especially Alan Anthony and Susie Castela.

Last but most important, I would like to express my deepest gratitude to my parents for their unconditional and perpetual love. The past four years of study at the University of Waterloo would not have been possible without their support and encouragement.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ALDPAT	Airborne LiDAR Data Processing and Analysis Tools
ALS	Airborne laser scanning
BN	Batch normalization
CNN	Convolutional neural network
DEM	Digital elevation model
DSM	Digital surface model
DTM	Digital terrain model
GPS	Global Positioning System
IDW	Inverse distance weighting
ISPRS	International Society for Photogrammetry and Remote Sensing
LiDAR	Light detection and ranging
PCL	Point cloud library
PDAL	Point data abstraction library
PMF	Progressive morphological filter
PTD	Progressive TIN densification
RADAR	Radio detection and ranging
RBF	Radial basis function
ReLU	Rectified linear unit
RMSE	Root mean squared error
SAR	Synthetic aperture radar
SOR	Statistical outlier removal
SRTM	Shuttle Radar Topography Mission
TIN	Triangulated irregular network

Chapter 1
Introduction

1.1 Motivation

High quality digital terrain models (DTMs) are vital to various applications such as urban building reconstruction (Dorninger & Pfeifer, 2008), carbon storage estimation (Chen et al., 2018), off-ground object detection (Jochem et al., 2009), land cover mapping (Matikainen et al., 2017), etc. Current DTMs are mostly generated from using synthetic aperture radar (SAR) interferometry or digital photogrammetry. The most widely used digital elevation model (DEM) products worldwide provided by NASA's Shuttle Radar Topography Mission (SRTM) (Farr & Kobrick, 2000) have a coarse resolution of only 30 m. In Ontario, the DEMs generated by digital photogrammetry under Southwestern Ontario Orthophotography Project (SWOOP) have low vertical accuracy of 50 cm (Ministry of Natural Resources and Forestry, 2016). In addition, photogrammetric techniques can only measure surface elevation, thus may result in poor estimations in dense forested area. Recently, Natural Resources Canada released the High Resolution Digital Elevation Model products which are derived from LiDAR (Natural Resources Canada, 2018), however, the vertical accuracy of the DEM products is no better than 1m. Therefore, it is necessary to develop a workflow for high quality DTM generation from reliable data source.

During the past decades, the production of DTMs from ALS systems has been extensively studied. ALS demonstrated powerful capability of capturing high-resolution 3D point clouds, and thus making creating high-quality DTMs possible even in complex forest or urban areas. High-density point clouds enable the detection and mapping of small variation. Surface features such as buildings, trees, or even pipelines and power lines can be extracted from dense point clouds. Compared with photogrammetric techniques, high-density point clouds can accurately capture slight slope variation of the earth surface, which makes it possible to generate high-quality digital elevation models (DEMs) (Liu, 2008). Another advantage of using the Light Detection and Ranging (LiDAR) systems is its ability to capture multiple returns from one laser pulse. While the information provided by optical imaging sensors is only limited to the target surface, LiDAR can penetrate through partially penetrable objects

(e.g., tree canopies) and acquire the underneath structure. Both first and last pulses are used in different survey applications. While first pulse returns are ideal to measure the outer surface of target object, last returns are suitable for non-penetrable surfaces (e.g., building rooftops, impervious surface) (Jutzi & Stilla, 2005). Most ALS systems are able to capture four to five echos per emitted pulse, which is advantageous in dense canopy covered forest areas (Vosselman & Maas, 2010). LiDAR system is an active remote sensing system. It measures distance by actively emitting a laser pulse to the target and capture the return signal. Compared to other surveying techniques, this characteristic makes it feasible to use LiDAR regardless of weather conditions and independent to sunlight illumination.

Various filtering techniques have been proposed for DTM production. These methods adopted different filtering techniques such as surface adjustment, slope operator, morphological filtering, triangulated irregular networks (TIN) based refinement, etc. Each technique is suitable for certain types of terrain. Sithole and Vosselman (2004) compared eight mainstream filters and tested their performance on eight sites with different terrain characteristics, which were later used as a benchmark for filtering evaluation. Their work also provided a comprehensive assessment of filters that inspired many researchers to refine and combine the existing filters. Despite the large body of research published in this domain, DTM extraction still remains challenging. Meng et al. (2010) listed three types of terrain especially difficult to filters: slope with discontinuity, dense forest canopies, and ground with low vegetation. Algorithms proposed for urban areas often ill performs in forested region and vice versa. In mountainous areas, steep slope and break lines are often misclassified. For areas with mixed terrain types, algorithms using global parameters were found difficult to perform well in all terrain types.

In view of all these difficulties, several promising directions to improve current methods have been put forward. Chen et al. (2017) suggested that different models should be combined to achieve optimal results. Since each model have its distinct advantage and disadvantage in different types of terrains, by combining the merits of each filter, the accuracy of DTM generation can be improved significantly. One example of such combination is the connected

operator proposed by Mongus and Žalik (2014), which is essentially a mixture of morphological filters, multi-scale comparison and segmentation. Another possible advancement in DTM generation is to utilize the information from multiple sources. Since it is difficult to discriminate ground and non-ground points using only elevation, ancillary information such as intensity or features extracted from full waveform LiDAR are often used (Liu, 2008). Due to the frequent availability of ALS point clouds and coincidence aerial or satellite images, Luo et al. (2015) and Singh et al. (2012) proposed to fuse LiDAR data with multispectral remote sensing images to achieve better classification accuracy. Their methods were used in land use classification. Recently, with the advance of multispectral LiDAR, spectral information is inherently available during the point cloud acquisition process. Such ancillary feature empowers classification with LiDAR data to achieve high accuracy even when point clouds are the sole input (Matikainen et al., 2017).

Recent developments in semantic labelling of point clouds have shed light on the DTM extraction problem. Essentially, the DTM extraction problem is a binary classification problem, which classifies point cloud into ground and non-ground points. Deep neural networks developed for semantic labelling of point clouds can be easily modified for DTM extraction. Since deep learning models can learn critical features directly from datasets, the generalization ability of these models is typically stronger than that of traditional filters. Studies applying deep learning models in DTM extraction problem have achieved high accuracy even in mountainous regions (Hu & Yuan, 2016; Rizaldy et al., 2018). The main challenge in applying convolutional neural networks (CNNs) to point cloud classification is the unorganized and irregular data structure of point clouds. Qi et al. (2017) and Yousefhussien et al. (2018) proposed deep learning models that take raw point clouds as input. The networks were designed to be permutation-invariant, which means the order of input points does not affect the classification results. Hu and Yuan (2016) developed a CNN-based model specifically for DTM filtering. Testing on the ISPRS benchmark dataset demonstrated the superiority of this method compared to traditional filtering approaches.

1.2 Objective of the Study

Due to the advantage of ALS data in DTM extraction and the superiority and deep learning models in point cloud labelling, this thesis aims to build a deep learning network for DTM extraction. The specific objectives are described as follows:

Firstly, this study determines the suitability of deep neural networks for extraction of DTMs and examines the power of deep CNNs for DTM generation.

Secondly, this study explores the use of transfer learning and fine-turning for working with limited training data.

Lastly, this study compares the proposed workflow with traditional filtering methods to examine its advantages and limitations.

1.3 Structure of the thesis

This thesis consists of six chapters. Chapter 1 introduces the motivation, objectives and structure of the thesis.

Chapter 2 provides a comprehensive review of the DTM creation process, with focus on different filtering techniques used to differentiate ground versus non-ground points in ALS point clouds. Difficulties and possible solutions are discussed with respect to data structure, model constraints and real-world complexity.

Chapter 3 describes the characteristics of the study area, as well as providing information of the ALS dataset.

Chapter 4 details the proposed methodology, which includes four parts: data pre-processing, feature image creation and model training, raster DTM creation and accuracy assessment.

Chapter 5 presents the results and findings of this study. The performance of proposed workflow is assessed. The DTM extraction results are compared with two widely used filters: progressive morphological filter (PMF) and progressive TIN densification (PTD). The filters are evaluated based on three criteria: point cloud classification accuracy, root mean squared error (RMSE) of interpolated DTM compared to true label, and qualitative analysis.

Chapter 6 summarizes the contributions and limitations of this study, provides recommendations for future research.

Chapter 2
Related Studies

This chapter reviews studies related to digital surface models (DSM) to DTM filtering using ALS data. In Section 2.1, the differences between DTM, DSM and DEM are described. Then, Section 2.2 describes the general workflow of DTM creation, including data preprocessing, filtering and ground points interpolation. Section 2.3 gives a detailed review of existing filters and compares their strengths and weaknesses. Section 2.4 discussed the challenges and possible solutions.

2.1 Differences between DEM, DSM and DTM.

The term DEM, DSM and DTM may appear similar. However, each of these terminologies allude to different models. DEM is defined as a set of earth surface elevation measurements, between which the spatial proximity and spatial relationships can be determined either implicitly or explicitly (Fisher & Tate, 2006). The term DEM can refer to both DTM and DSM. To avoid confusion, the term DEM will not be used throughout this paper.

Raw LiDAR point clouds include both ground and non-ground points. After data geo-referencing, outlier removal and interpolation, the entire point cloud can be transformed into a "DSM", which includes the elevation of both ground and non-ground objects. The creation of "DTM" is much more complex due to the need to remove non-ground points. "DTM" refers to the elevation model that is interpolated only using ground (bare earth) points. Natural or artificial object such as trees and buildings are removed during the creation of DTM.

2.2 General Workflow of Creating DTM

The general workflow of creating DTM is described in Figure 2.1. The creation of DTM involves three steps: data pre-processing, ground points filtering and interpolation. During data pre-processing, no interpretation of the point cloud is made. The raw point cloud contains all returns including both ground and non-ground points. In order to create DTM, non-ground points need to be removed. Since ground points filtering is the most important procedure, it

will be discussed in detail in Section 2.4. Interpolation refers to the process of creating DTM using existing ground points. There are two commonly used data format to store DTM: raster and TIN. The interpolation process will be discussed in Section 2.2.2.

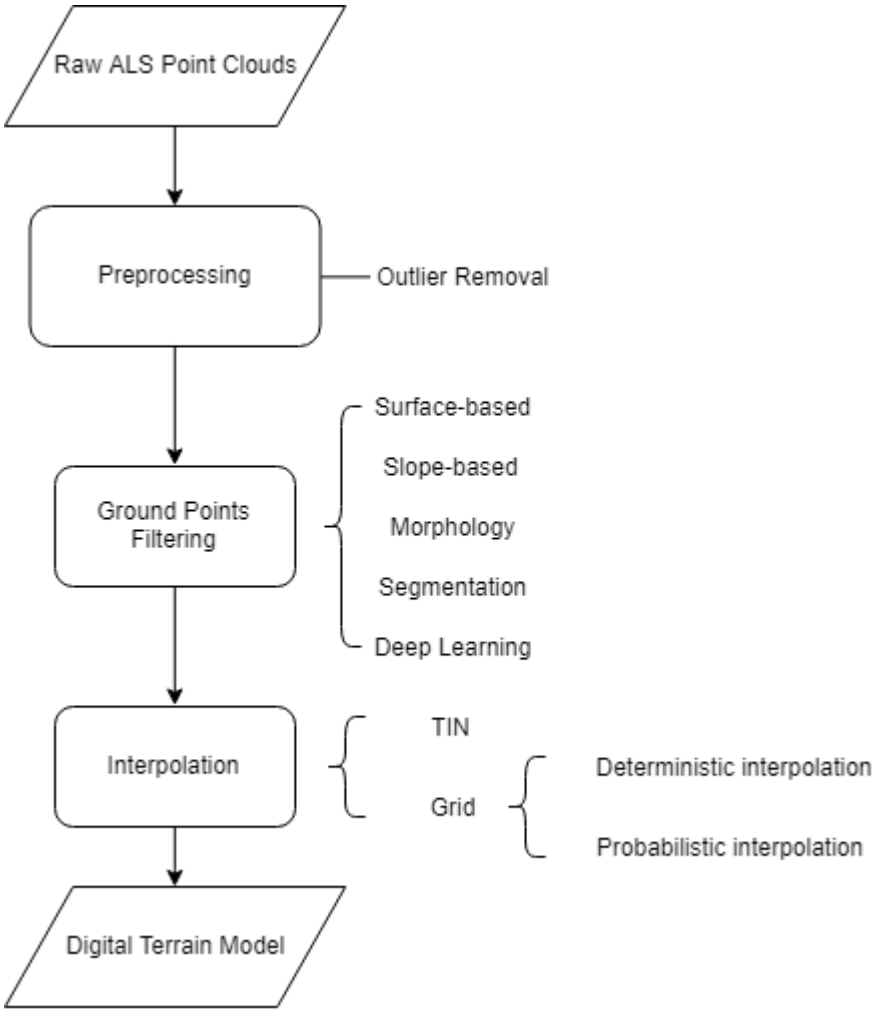


Figure 2.1 General Workflow of Creating DTM

2.2.1 Data Pre-processing

Raw LiDAR point clouds need to be pre-processed before creating DTM. The major task in data pre-processing is outlier removal. In the task of creating DTM, high outliers are less malicious since they can be easily removed by comparing with nearby points. On the other hand, surface-based filters, morphological filters and some statistical filters rely on the

assumption that local minima can be regarded as a ground point. If a low-elevation outlier was misidentified as ground point, it would have negative impact on the performance of filter. Based on elevation, outliers can be classified into local and global outliers. Global outliers are the points that have exceptionally high or low elevation values across the entire data set. Generally, high outliers are caused by echoes from birds, aircrafts, etc. Low outliers are caused by laser range finder malfunctioning or pulses that are reflected multiple times (Sithole & Vosselman, 2004). These outliers only take up a small percentage of the point cloud. Thus, they can be easily filtered by setting a quantile threshold based on the elevation distribution (Silván-Cárdenas & Wang, 2006; Chen et al., 2017).

Local outliers are the points that have exceptionally high or low elevation values compared to neighbouring points. The removal of local outliers is more challenging. In order to remove local outliers, it must be distinguished with its neighbourhood. There are various approaches to identify local outliers, such as mathematical morphology approach, distribution examination, extended local-minima, etc. (Chen et al., 2017).

Silván-Cárdenas and Wang (2006) removed local outlier by setting height difference threshold within each points neighbourhood. The neighbourhood was defined using a Delaunay triangulation network. This method was also adopted by Zhang et al. (2013) and Meng et al. (2009). Sotoodeh (2007) proposed a hierarchical outlier detection algorithm, which also utilizes the use of Delaunay triangulation networks. The algorithms consist of a global phase (rough clustering) and a local phase (fine clustering). During the global phase, statistical information of the data distribution is captured. Then, the local phase makes use of the global statistical information to generate local criteria to further cluster the point cloud. Su et al. (2015) identifies local outliers using both elevation-limiting and angle-limiting method. Elevation-limiting method detects outlier whose height is more than three standard deviation from the mean elevation of all the LiDAR points within its neighbourhood. The searching radius of its neighbourhood is defined by the maximum building size. Angle-limiting method removes points generated by echos from building facades. Similar to the elevation-limiting method presented in Su et al. (2015), Hui et al. (2016) also removed low outliers whose

elevations are outside three standard deviation from local mean. The local area is defined as 3*3 cells in the interpolated grid.

Some filtering algorithms utilize the intensity value of points. For such methods, intensity outliers also need to be removed. Intensity values represent the number of photons hitting the detector. The values are related to the energy reaching the receiver (Vain et al., 2010). Beyond the domain of point cloud pre-processing, some mathematical approaches to remove outliers can also be applied to point clouds. Breunig et al. (2000) proposed a density-based method to identity local outliers. An index called local outlier factor was designed to describe the likelihood of an object being an outlier. Their approach can be used for both elevation and intensity outlier removal.

2.2.2 Ground Points Interpolation

After the non-ground points are removed, the remaining ground points are to be interpolated to create the DTM. There are two common data structure to store a DTM: grid and TIN.

- Triangulated Irregular Networks

Triangulated Irregular Networks (TIN) represent the topography structure by constructing a series of triangles where the vertices are selected from ground points (Gevaert et al., 2018). The triangles are formed in a way that all sampled points are joined in the network as vertices, yet each triangle is empty and does not contain any of the sampled points. This can be achieved through the Delaunay triangulation or distance ordering (ArcGIS, n.d.). The elevation value within a triangle is estimated by linear or cubic polynomial interpolation (Ripley, 2005). TIN has been used in terrain modelling since the 1970s. However, due to the complexity of its data structure and limitation on computational power at that time, grid data was preferred over TIN (Roggero, 2001).

- Grid DTM

Grid is the most common way to store DTM (Fisher & Tate, 2006). A grid DTM is generally created by interpolating ground points. Based on whether spatial autocorrelation is

utilized, interpolation methods can be categorized into two types: non-geostatistical and geostatistical.

Non-geostatistical method is also called deterministic method, which means the value is interpolated directly based on its surrounding measured values. Commonly used deterministic interpolation includes nearest neighbour, inverse distance weighting (IDW), linear or polynomial regression models, trend surface, radial basis function (RBF), etc. Some of the non-geostatistical methods (i.e., IDW) using weighted average technique, which means interpolated value is calculated by

$$\hat{\mathbf{z}}(\mathbf{x_0}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \boldsymbol{\lambda_i} \mathbf{z}(\mathbf{x_i}) \qquad (\mathbf{2.1})$$

where $\hat{z}(x_0)$ is the estimated elevation of point x_0, n is the total number of points, λ_i is the interpolation weight assigned to each point, $z(x_i)$ is the measured elevation of point x_i (Li & Heap, 2008). Thus, the estimated values cannot exceed the minimum and maximum elevation range of measured points. This type of methods works well in high point density areas. However, it is not suitable when the points are sparsely distributed. Protrude geological features such as ridges and hills are likely to be over-smoothed by this type of methods.

Another branch of non-geostatistical methods is called exact interpolation techniques (i.e., RBF, spline interpolation). Values are estimated using a mathematical function that minimizes overall surface curvature. The resulting surface passes exactly through measured points (Liu, 2008). This type of method can yield value outside the range of minimum and maximum elevation in sampled points, which makes it possible to map ridges and valleys that have not been adequately sampled.

Geostatistical method (e.g., Kriging) is also called probabilistic method. Apart from describing spatial pattern and interpolating unsampled locations, it also models the error of the interpolated surface (Li & Heap, 2008). Unlike non-geostatistical methods which only considers the local measurements, Kriging takes into account the spatial variation of the entire

study site. One of the most important concepts in Kriging is semivariance, which describes the spatial autocorrelation. The semivariance of a set of points can be calculated by

$$\hat{\gamma}(h) = \frac{1}{2n}\sum_{i=1}^{n}(z(x_i) - z(x_i + h))^2 \qquad (2.2)$$

where n is the number of points pairs sampled at distance h, z is the measurement to be interpolated (in our case, elevation), $\hat{\gamma}(h)$ is the semivariance. A plot of $\hat{\gamma}(h)$ against h is referred to as experimental variogram, an example is shown in Figure 2.3 (Li & Heap, 2008).

Three most important features of a semivariogram are the nugget (C_0), the range and the sill $(C_0 + C_1)$. The nugget is the residual or variance of data at distances shorter than the minimum sample spacing. The sill is the maximum variability in the data set, which theoretically corresponds to the variance in statistics. The range is the value of h where sill is reached. Points separated by a distance longer than range are spatially independent since the semivariance remain constant beyond range. The difference between sill and range account for the amount of spatial variation (Karl et al., 2010). If the ratio between sill and nugget is close to 1, then most of the variation can be characterized as non-spatial. The window size used in interpolation is set based on range. Commonly used functions to fit the semivariogram include circular, spherical, exponential, Gaussian and linear. After a model is fit to the empirical semivariogram, Kriging calculates weight not only based on distances to surrounding measured points, but also the characteristics of semivariogram. Step-by-step description of calculating Kriging weights can be found in Li and Heap (2008).

A detailed comparison of TIN and grid DTM are listed in Table 2.1.

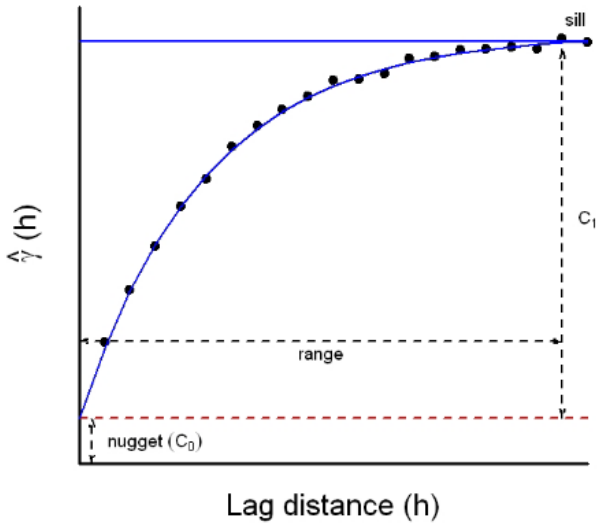


Figure 2.2 Example of Semivariogram (Source: Li & Heap, 2008)

Table 2.1 Comparison of TIN and grid DTM

	Pros	Cons
TIN	• Efficient data storage • Visualization of terrain features (Cvijetinović et al., 2008) • Accommodate irregularly spaced elevation data (Lee, 1991).	• Computationally expensive • Difficult accessibility
Grid DTM	• Preserve elevation detail • Easy accessibility • Image processing functions can be directly applied	• Require massive storage space • Error introduced by interpolation • Represent discontinuous surface

2.3 **Comparison of DTM Filtering Techniques**

Existing DTM filtering methods can be categorized into five types: (1) surface-based, (2) slope-based, (3) morphology-based, (4) segmentation, and (5) deep learning-based.

2.3.1 Surface-based Methods

Surface-based methods aims at approximating terrain surface by iteratively selecting ground points (Zhang et al., 2016). It typically involves two steps: (1) selecting seed points to form an initial sparse surface, (2) iteratively search for candidate ground points that fall within certain threshold to the initial surface (Figure 2.3). A moving window is used to search for ground points near the initial seed points. These methods are very sensible to the size of searching neighbourhood. On the one hand, if the window size is too large, detailed information would be lost, on the other hand, if the window size is too small, large non-ground object cannot be removed (Chen et al., 2017).

Some surface-based methods rely on the generation of TIN. Axelsson (2000) proposed a progressive TIN densification (PTD) model to generate DTM, which was considered a classic method of this genre. The model starts by creating a sparse TIN using by selecting local minima as seed points, then densifies the TIN iteratively. Two thresholds are derived from data: distance to the TIN facet and angel to the vertices. This method is proven to adapt well in discontinued land surface and has been successfully applied in commercial software TerraScan and open source software Airborne LiDAR Data Processing and Analysis Tools (ALDPAT). However, this method struggles in areas with steep terrain. Zhang and Lin (2013) proposed an improved TIN densification approach by embedding smoothness-constrained point cloud segmentation. After initial ground seed points are selected, smoothness-constrained segmentation was implemented to expand the initial selection. Compared to Axelsson (2000), more seed points are used in the initial TIN. Results show that omission error was decreased with this enhancement. Chen et al. (2016) proposed a TIN based approach specifically for residential areas reside steep mountainous region. It enhances the approach of Axelsson (2000) by improving the distribution and reliability of seed points. Since the study focuses on steep mountainous region, a rule-based approach is proposed to strengthen the selection of ridge points as seeds. The RMSE of the proposed filter on two test data sets has been proven to be lower than traditional PTD method.

Another type of surface-based methods simulates the terrain surface by interpolation. Kraus and Pfeifer (1998) proposed a surface-based filtering method using linear prediction. The algorithm constructs the initial surface by averaging the elevation of all points. Then weight was assigned to each point through iteration. Points lie below the surface will have negative residuals and will be assigned higher weights. The iteration continues until the surface is stable or maximum iteration number is reached. This method was first intended for canopy removal in forested areas, after the improvement by (Pfeifer et al., 2001), its usage was extended to terrain modelling in urban areas. Although this method is conceptually simple and easy to implement compared to TIN based approach, it lacks the reliability in discontinued surface. Moreover, it also suffers from large variability in steep regions (Zhang et al., 2016). To address these issues, multi-level interpolation filtering algorithms are developed. Su et al. (2015) proposed a surface-based filter by incorporating a hierarchical moving curve-fitting algorithm. The algorithm uses second-degree polynomial curve to fit the initial surface, then adaptive threshold and a series of decreasing grid sizes are iteratively applied to differentiate ground and non-ground points. Similarly, Hui et al. (2016) developed an algorithm that combines progressive morphological filtering and multi-level Kriging. The use of gradually decreased window sizes enables the filter to preserve abrupt change in slope.

Surface-based methods have achieved satisfactory results in most terrains but struggled to preserve details in steep slope regions. Also, this type of methods has the tendency to misclassify small non-ground objects as ground points (Mongus & Žalik, 2014). Furthermore, these methods rely on multiple iteration to locate candidate ground points, and thus require considerable computational time (Guan et al., 2014).

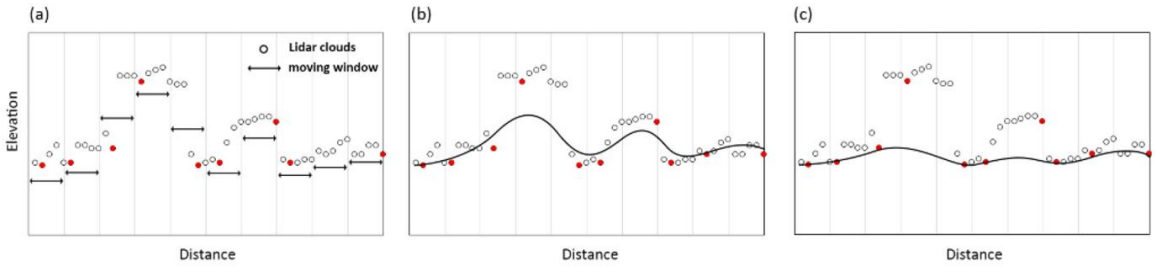


Figure 2.3 Schematic diagram of surface-based filters: (a) lowest point is selected in each window, (b) initial surface is formed by interpolation, (c) points lie below the initial surface are given higher weight, the initial surface is refined accordingly (Source: Chen et al., 2017)

2.3.2 Slope-based Methods

Slope-based filters assume that the slope of terrain is distinctly different from the slope of non-terrain objects (Sithole, 2001; Liu, 2008). Under such assumption, large elevation difference between two points with close horizontal proximity is unlikely caused by steep slope of the terrain, rather, the higher point is likely a non-ground point (Vosselman, 2000). This type of filters aims to create different slope indicators to describe the vertical and horizontal distance between neighbouring points (Chen et al., 2017). Slope between nearby points are calculated and compared to a pre-defined threshold. If the slope between a point and any of its neighbouring point exceeds the threshold, the higher one of these two points will be considered as non-ground point. The lower the threshold, the more points will be identified as ground points (Liu, 2008).

The key of success in such methods is the selection of the threshold. The slope threshold remains constant or as simple function of distance throughout the filtering process. Depending on the terrain type, different threshold should be selected. Prior knowledge of the terrain is essential to setting the optimal threshold (Zhang et al., 2003). However, it is impractical to precisely estimate the terrain characteristics. Also, per-defined slope threshold ill-performs in regions with mixed flat and steep terrain. To solve this problem, Sithole (2001) proposed an adaptive filter whose threshold varies with the slope of the terrain. Results demonstrate that the adaptive filter effectively improves filtering accuracy in steep region. Similarly, Susaki

(2012) incorporated a slope parameter which is updated after each iteration. Through this configuration, information of the local terrain can be included. Roggero (2001) improved the algorithm (Sithole, 2001) by performing local linear regression on interpolated grid. The first stage of the algorithm uses local regression to identify candidate ground points whose heights are compatible with the local slope variance, which only results in an approximation of the DTM. The second stage further classifies the points into ground and non-ground based on a threshold. Shao and Chen (2008) implemented a "climbing and sliding" method that imitates a continuous climbing motion from various local minima to local high grounds. Three parameters are used to control the climbing and sliding movement, namely the general slope, slope increment and maximum slope. After the initial search for ground points, a back-selection step was implemented for densification.

Slope-based filters are simple and computationally efficient. Nonetheless, how to set the optimal slope threshold remains to be the bottleneck in this type of methods.

2.3.3 Morphology-based Methods

Morphological filters are based on the idea of mathematical morphology. Erosion and dilation are two most fundamental operations in morphological filters. Morphological opening includes an erosion followed by dilation, which removes points higher than its neighbourhood. Morphological closing includes a dilation operation followed by erosion, which removes points lower than its neighbourhood (Shao & Chen, 2008). Similar to surface-based filters, morphological filters are sensitive to the operation window size. On the one hand, large window size tends to treat ground points as non-ground points (Zhang et al., 2003), and could result in over-smoothing and loss of information (Shao & Chen, 2008). On the other hand, while small window size is effective in removing small objects such as trees and cars, large buildings in urban environment cannot be removed. Since most mathematical morphology operations are applicable on raster image, morphology-based filters almost always first transform the 3D point clouds into a 2D elevation raster, then apply morphological image-processing techniques on the raster (Chen et al., 2016). Non-ground object generally has higher

elevation value than ground points. When converted to a raster image, object and terrain can be differentiated by their pixel value.

The progressive morphological filter (PMF) proposed by Zhang et al. (2003) is one of the most successful implementations of morphological filters. Same as most morphological filters, PMF first transforms the point cloud into a 2D grid. Then, a series of gradually increased windows are used to filter the point cloud. An initial surface is constructed by performing an opening operation that first erodes and then dilates the points within the window. Buildings larger than current window size are preserved, while trees smaller than window size are removed (Figure 2.4). In each cell, if the elevation difference between the initial surface and the original point cloud surpass a threshold, the point is classified as a non-ground point. The iteration continues until the window size exceeds the predefined maximum window size, which is set as the size of the largest non-ground objects. The major drawback of this method is the presumption of constant slope, which makes the filters ill perform in terrain with mixed relief. To address this issue, Hui et al. (2016) proposed an improved morphological algorithm by introducing a series of downsized windows, within which multi-level Kriging interpolation was performed to calculating the terrain slope gradient. Therefore, instead of remaining constant, the slope can be adjusted within each window. Zakšek and Pfeifer (2006) improved the traditional morphological filter by involving first return point in generating DTM. This method is specifically designed for areas with dense vegetation as it makes the assumption that the forest structure is homogeneous within a grid cell. Under this assumption, the height difference between canopy and ground surface can be regarded as uniform within one cell.

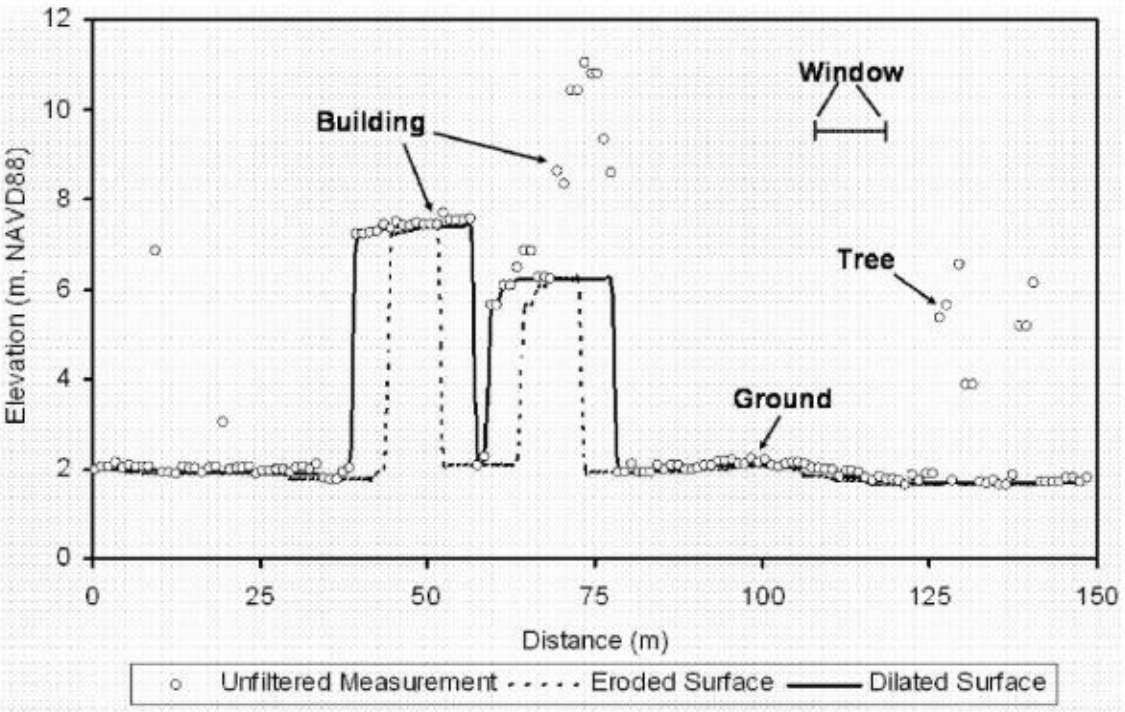


Figure 2.4 Erosion and Dilation (Source: Zhang et al., 2003).

Morphological filters are conceptually easy to implement. However, they usually require the point cloud be transformed into a 2D grid prior to filtering. Although 2D grid is more efficient than unordered point cloud in terms of computational cost (Susaki, 2012), such transformation can cause a significant loss of information (Axelsson, 2000; Vosselman, 2000). Moreover, this type of filters assumes that the terrain is relatively flat within a local region. In mountainous areas where the elevation changes abruptly over break lines, the parameters need to be readjusted to suit each application's need (Chen et al., 2016).

2.3.4 Segmentation-based Methods

Segmentation-based methods are similar to object-based classification in remote sensing image studies. First, raw ALS data is transformed into grid image or voxel, this step is optional but widely adopted in practice due to the difficulty in processing unorganized ALS point cloud. Then, segmentation is performed based on the height or intensity value. After segmentation,

classification is performed according to the geometric characteristics and topographic relationships of segments (Liu, 2008). Segments, instead of points, are treated as the basic processing unit in classification. Due to the fine resolution of LiDAR data, neighbouring points are highly correlated in terms of elevation and intensity, which makes segmentation-based methods applicable (Blaschke & Tomljenovic, 2012).

Segmentation and classification rules are the key to this type of filters. The raw ALS data have relatively few attributes. Most studies utilize only the inter-points geometric relationships as segmentation criteria. Elevation difference, slope and curvature difference are derived from elevation measurements, then, by setting segmentation threshold through region growing (Roggero, 2002) or clustering (Filin, 2002), point cloud can be partitioned into segments. Roggero (2001) also suggested that by incorporating intensity values, first/last return information, the clustering result can be improved. Antonarakis et al. (2008) developed a spatial and spectral segmentation tool using both elevation and intensity value, as well as point distribution frequency to identify heavily vegetated area.

After segmentation, different classification techniques can be applied to discriminate the segments. At this stage, not only the geometric attributes derived by point elevations, but also characteristics of segments such as compactness, roundness, perimeter that can be used in classification. Commonly used classification methods include support vector machine (Zhang et al., 2013), random forests (Niemeyer et al., 2014), CNNs, etc.

Segmentation-based methods are commonly combined with other type of methods. Tovari and Pfeifer (2005) suggested that by combining the segmentation and surface interpolation-based method, the performance of filer can be improved in both urban and forested areas. Lin and Zhang (2014) proposed a method based on segmentation and classic PTD. Experimental results indicate that the combined method can better preserve discontinuity of terrain as well as removing lower part of large object attached to terrain surface.

Segmentation-based methods are suitable for urban areas where steep edges may be found in the region. Unlike surface-based, slope-based and morphology-based filters, segmentation methods do not assume surface continuity. In airborne laser scanning data, homogeneous

surfaces such as roof tops, building facets are grouped into same segment. Since points are group by similarity prior to filtering, sharp edges and break lines can be cleanly delineated. On the contrary, this type of methods tends to struggle in densely forested area since no homogeneous plane can be detected. Moreover, the success of this type of methods heavily rely on the segmentation and classification parameters, thus, tuning the parameters can cause uncertainty to the model performance (Chen et al., 2017).

2.3.5 Deep Learning-based Methods

DSM to DTM filtering is essentially a binary classification problem. The task is to classify raw point cloud into two types of points: ground and non-ground. The aforementioned filters mostly rely on certain assumption of terrain features, which results in misclassification when the environment is complex (Hu & Yuan, 2016). Deep CNNs are able to extract high-level representation features through compositions of low-level features (Girshick et al., 2014). The model does not make assumptions of the terrain, instead, representation terrain features are directly learned from training data. These automatically learned features generally work better than hand-crafted features. Hu and Yuan (2016) proposed a deep learning-based filter that classifies at point level. First, each point and its neighbouring points are transformed into an image. The image is a positive sample if the central point is a ground point and vice versa. Then, the images are treated as input of a deep CNN model. Yang et al. (2018) proposed a similar point-to-image transformation technique which are used for point cloud semantic labeling.

One of the major challenges in deep learning-based filters is the availability of labelled data. Training data is critical in deep learning-based method. The diversity of training data directly affects the model performance. However, large quantity of manual labelled point cloud is difficult to acquire. To solve this problem, Gevaert et al. (2018) first applied morphological filter to select candidate ground and non-ground points, then, only the most confident samples produced by the morphological filter are used to train a fully convolutional network. Tests

show that the automated labelling strategy can yield comparable results to using manual labelled samples as training data.

2.4 **Difficulties and Possible Solutions in Ground Point Filtering**

The DTM filtering problem has been under research for decades. Various filtering algorithms have been proposed to solve this problem. Each has its own advantages and limitations. Most filters are designed to adjust to varied terrain types and are usually able to perform reasonably well in moderate complex landscape (Sithole & Vosselman, 2004). Nevertheless, filtering error cannot be completely eliminated due to data structure, constraints of model, and the complexity of real environment. To date, there is no algorithm that can address this problem satisfactorily. The challenge remains in several aspects:

First, the unordered data structure of point cloud. Irregularly distributed point cloud can be computationally expensive to process; therefore, most studies choose to transform the point cloud into ordered data format first, then perform filtering on the ordered data (Liu, 2008). Commonly used formats include raster (Roggero, 2001; Zhang et al., 2003; Zakšek & Pfeifer, 2006), voxel (Zhao et al., 2009) and isometric strips (Wu et al., 2016). However, these transformations are usually accomplished through interpolation and averaging, thus can result in loss of information. In addition, grid cell whose value is interpolated from both ground and non-ground points can cause difficulty for the filter. Sithole and Vosselman (2005) and Zhang (2007) suggested that filtering should be conducted on raw point cloud instead of interpolated grid. A compromised approach of selecting the point with lowest elevation to represent the cell value has been adopted by Zhang et al. (2003), yet this approach can be more vulnerable to low outliers compared to the interpolated grid.

Second, the constraints of model. Each model is designed based on certain assumption of the terrain surface morphology, thus has its advantage and disadvantage on different types of terrain. Surface-based methods are sensitive to terrain with break lines, steep slope and high variability. They are also computationally expensive. Slope-based methods work well in flat terrain but produce poor results in high relief areas (Hui et al., 2016). The success of

morphological filters relies heavily on the selection of window size (Zhang et al., 2003). Almost all filtering algorithms assume terrain surface to be continuous in all directions (Sithole & Vosselman, 2005). Segmentation methods provides satisfactory result in urban environment but can over segment heavily vegetated area. To date, none of the filter can be successfully applied in large area with complex terrain features. As mentioned in Section 2.3, studies have been exploiting the feasibility of combined models, which is a promising direction in improving the filtering accuracy. The feasibility of deep learning models in DTM extraction is worth investigating. Deep learning models do not make assumption of the terrain, also, representation features learnt directly from the dataset typically work better than hand-crafted data. However, a considerable gap remains in applying deep learning on ALS data for point cloud classification.

Third, the complexity of real-world environment. Normally, artificial objects that have relatively small size, closed outline and slope discontinuity with terrain (e.g., detached house) are easy to be removed. Sparse vegetation that allows LiDAR signal penetration are also easily removed. Hilltops and ridges are higher than local terrain surface and are often misidentified as non-ground objects. Break lines such as ridges and cliffs that cause slope discontinuity in the terrain, are often being often smoothed by the filter (Shao & Chen, 2008). There are certain ground features that make the filtering process difficult. Meng et al. (2010) listed seven terrain features that cause problem for filters: (1) shrubs below 1 m, (2) short walls along walkways, (3) bridges, (4) buildings with various sizes and shapes, (5) cut-off edge, (6) complex mixed covering, and (7) region with both low and high-relief terrain. Break lines are the places where the terrain elevation change abruptly, such as mountain ridges, cliffs and dikes (Sithole & Vosselman, 2004). Shao and Chen (2008) also mentioned that break lines can cause trouble for slope-based method.

For complex scene such as urban and forest, a single sensor may not be sufficient in providing information for classification and feature extraction. LiDAR data is sometimes fused with other data source to improve classification. Optical imagery (Gevaert et al., 2018) and hyperspectral imagery (Ghamisi et al., 2016) are commonly used as ancillary data. Moreover,

the newly emerged multispectral LiDAR (Matikainen et al., 2017) also have the ability to capture spectral characteristic.

2.5 **Chapter Summary**

This chapter first describes the differences between DTM, DSM and DEM. Then, a general workflow of creating DTM is described, with special focus on different ground point filtering techniques. According to the algorithms, filtering techniques can be categorized into five types: surface-based slope-based, morphology-based, segmentation and deep learning. Through a comprehensive viewing of literatures, the advantages and disadvantages of each filter are summarized. The challenges of DTM extraction remains in three aspects: the unordered data structure of point cloud, the constraints of model, and the complexity of real-world environment. Most traditional filters rely on certain assumptions of terrain morphology, thus fail to generalize well in different type of terrains. Deep learning is a promising direction for DTM extraction problem as it does not rely on assumption of the terrain, instead, representative features are learnt directly from dataset.

Chapter 3
DTM Extraction from ALS Point Clouds

This Chapter describes the proposed methodology in detail. Section 3.1 introduces the topography of study area. Section 3.2 describes the datasets used in this study. The subsequent sections detail the workflow. The methodology consists of three parts: preprocessing, feature image creation and model training, and accuracy assessment. Section 3.3 describes the data preprocessing step with emphasis on the low outlier removal. Section 3.4 describes the feature image generation process and explains the transition from a point cloud classification problem to an image classification problem. Section 3.5 introduces the architecture of residual networks and the advantages against traditional CNNs. Section 3.6 describes the details in training configuration. Section 3.7 presents three interpolation techniques used to generate DTM from the extracted ground points. Section 3.8 presents three methods to assess the quality of generated DTM. Finally, Section 3.9 summarizes the chapter.

3.1 Study Area

The study area is the main campus of the University of Waterloo, Ontario, Canada. Total area comprises of approximately 1.15 km^2. The study area portraits a mixture of typical urban layout and forest environment, which consists of large buildings, roads, forest, lawn, parking lots, individual trees, bushes, small lake, and a creek. The majority of the study area is flat, with some elevated area in the northern part of the scene. The complexity of ground objects and the mixture of low and high relief topography make this area suitable for DTM extraction analysis. The location of the study area created by the ALS point cloud is presented in Figure 3.1.Figure 3.2 DSM of the study area

3.2 Datasets

There are three datasets used in this study. The first dataset is the 2014 Regional Municipality of Waterloo Road dataset acquired from the Geospatial Centre, University of Waterloo. Four road segments were employed to delineate the boundary of the study area:

Columbia Street West to southwest, Phillip Street to northeast, University Avenue West to southeast and Westmount Road to southwest (Figure 3.1). The road segments were processed by the ArcGIS Production Editing to create a polygon that covers the study area, which was later used to extract ALS point clouds and orthoimages located within this area.

The second dataset is the City of Waterloo ALS dataset acquired by Leading Edge Geomatics between November 2 and 3, 2014 using a RIEGL Q680i system. The system specifications are shown in Table 3.1. The average flight height was 1,200 m above ground, producing the ALS point clouds with a horizontal accuracy of approximately 31 cm (RMSE) and a vertical accuracy of 6.1 cm (RMSE) (Leading Edge Geomatics, 2015). The dataset was tiled into 1km*1km grids, among which five grids intersects with the study area. The DSM created by the ALS dataset is shown in Figure 3.2.

Table 3.1 Specifications of RIEGL Q680i system

Laser Wavelength	Near infrared
Scan Pattern	Parallel scan lines
Scan Speed	10-200 lines/sec
Scan Angle Range	±30° = 60° total
Laser Pulse Repetition Rate	up to 400,000 Hz
Angle Measurement Resolution	0.001°

The third dataset is an orthophoto collected by the Southwestern Ontario Orthophotography Project in 2015 (Figure 3.4). The data was used to overlay with the ALS point cloud data to visually inspect the validity of labels.

Each point contains the X, Y, and Z coordinates as well as four additional attributes: return number, total number of returns, intensity and classification label. The X, Y, Z coordinates were used to differentiate between ground and non-ground points, as well as to produce the DTM model of the study area. The classification label was used for training the CNNs and quantitatively assessing the accuracy of the proposed workflow. The return number and total

number of returns were discarded in this study. However, for most DTM filtering studies, these two fields are meaningful as only the last return of each laser pulse may indicate echo from ground. In the proposed workflow, in order to preserve as much spatial information as possible during feature image generation (described in Section 3.4), all returns were kept to delineating the complete structure of off-ground objects. The dataset was originally classified into four classes: ground, low vegetation, medium vegetation, high vegetation and building, which were merged into two categories: ground and non-ground to suit the purpose of this study.

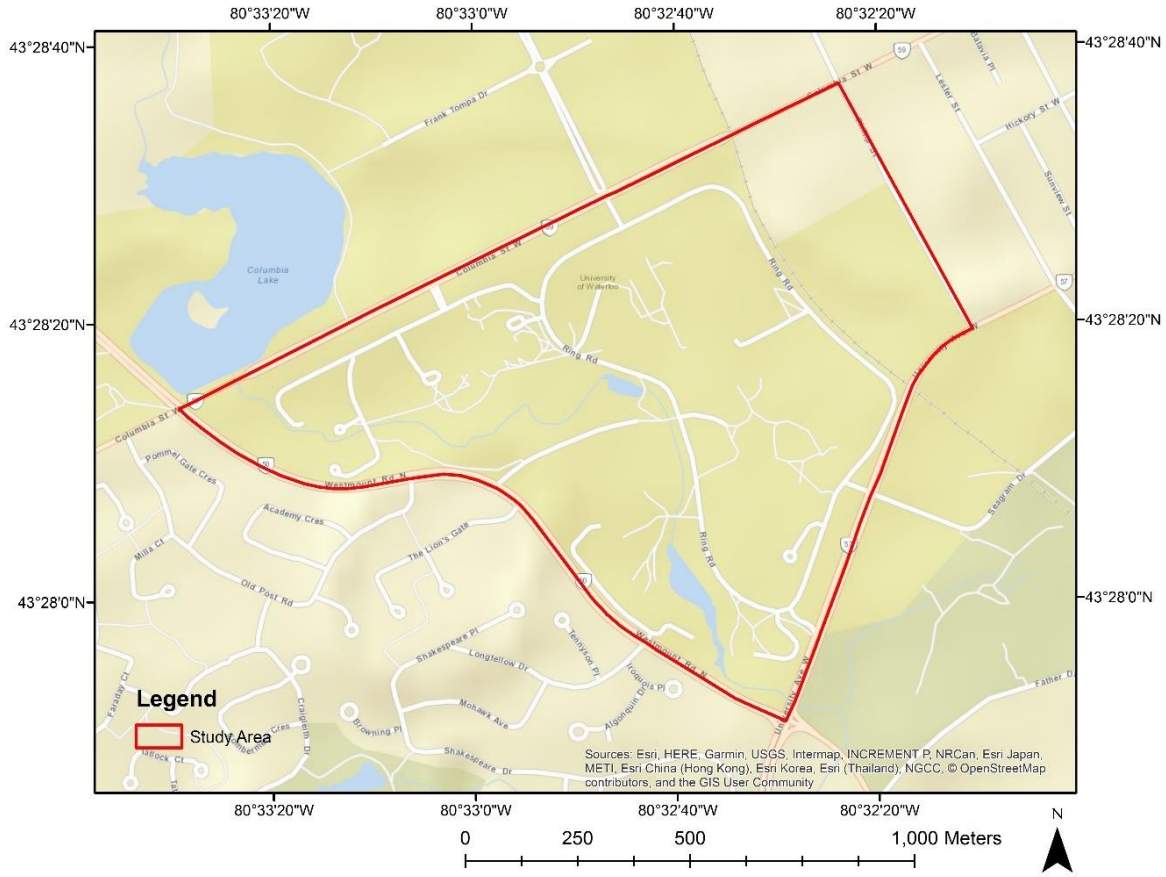


Figure 3.1 Location of the study area delineated by road segments

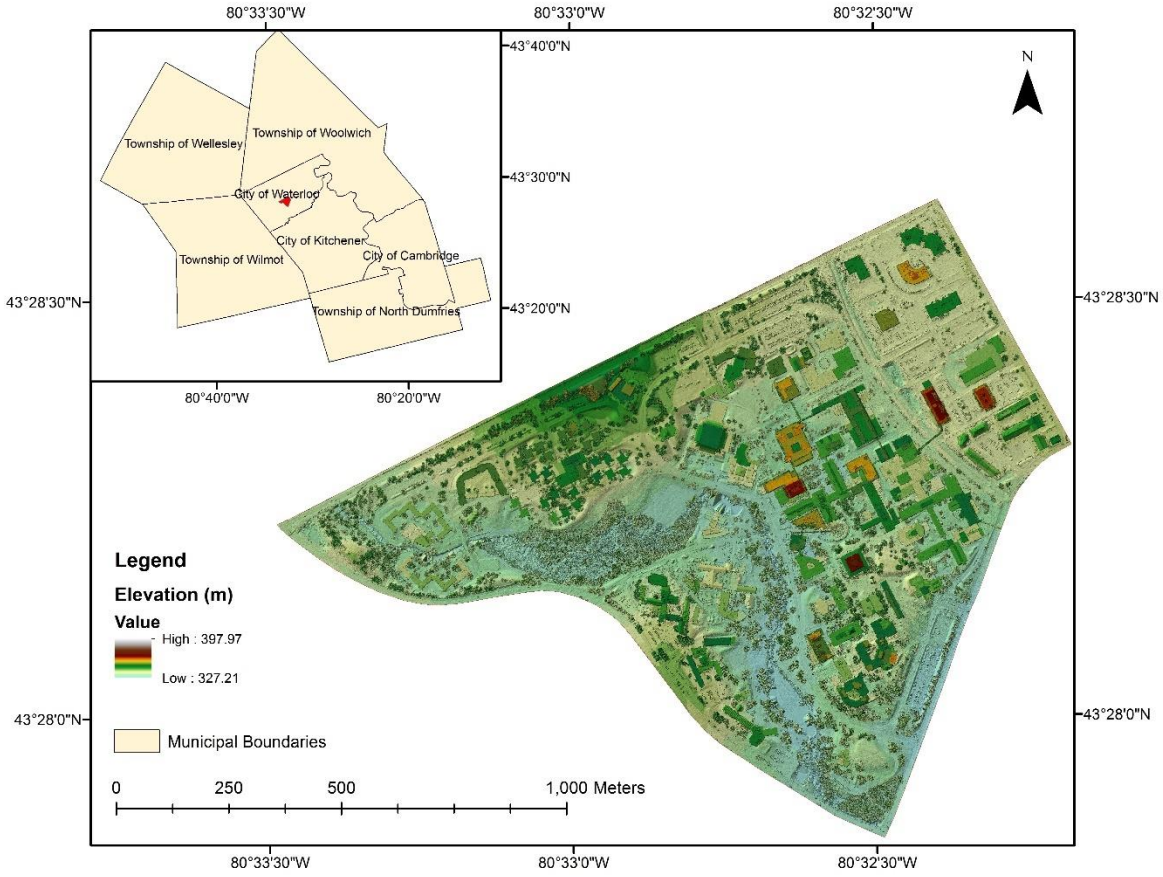
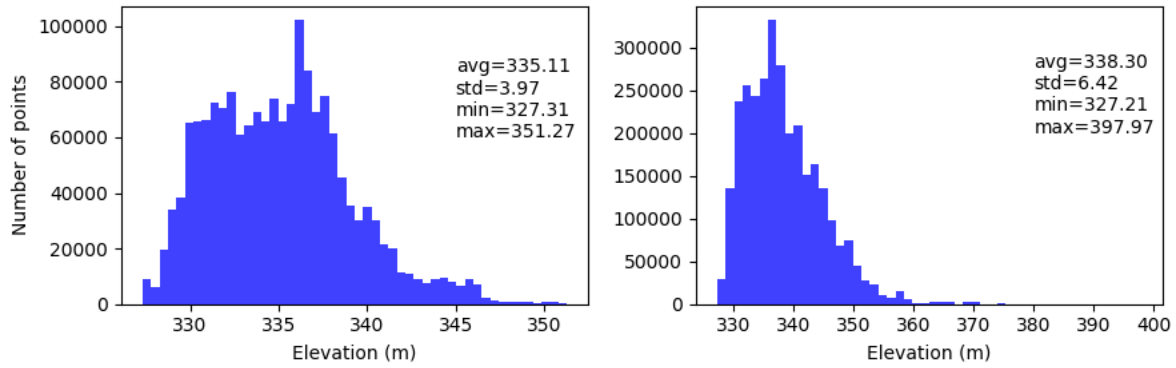


Figure 3.2 DSM of the study area

The elevation distributions of the point clouds are shown in Figure 3.3(b). The average elevation is 338.30 m, with standard deviation of 6.42 m. As can be seen, the data displays a right-skewed distribution, since the high elevation points mostly come from treetops or building rooftops, which only consist of a small amount of the dataset. The elevation distribution of only the ground points is shown in Figure 3.3(a).The average elevation of ground points is 335.11 m, with standard deviation of 3.97 m. The elevation distribution of ground points is also right-skewed due to the terrain variation shown in Figure 3.3.



(a) Ground points (b) Original point cloud

Figure 3.3 Elevation distribution of point clouds

The study area contains certain features that have been identified as difficult to be handled by those filters presented by previous studies (Sithole & Vosselman, 2004; Meng et al., 2010): complex building structure, dense vegetation, bridge and vegetation on slope (Figure 3.4). Complex buildings are particularly troublesome for slope-based and surface-based filters. Due to the flatness of the rooftops, the lower rooftops of the building are likely to be identified as ground. Dense vegetation with tall trees, shrubs and grass present simultaneously are difficult to filter since the structures of off-ground objects are ambiguous. Bridge, ramps and elevated pedestrian walkway are treated as off-ground objects. Different from other objects, these structures are considered special as they span gaps in the bare earth. When encountering sloped area with vegetation or building, filters tend to correctly classify vegetation and buildings at the expense of misclassifying sloped terrain as non-ground (Sithole & Vosselman, 2004).

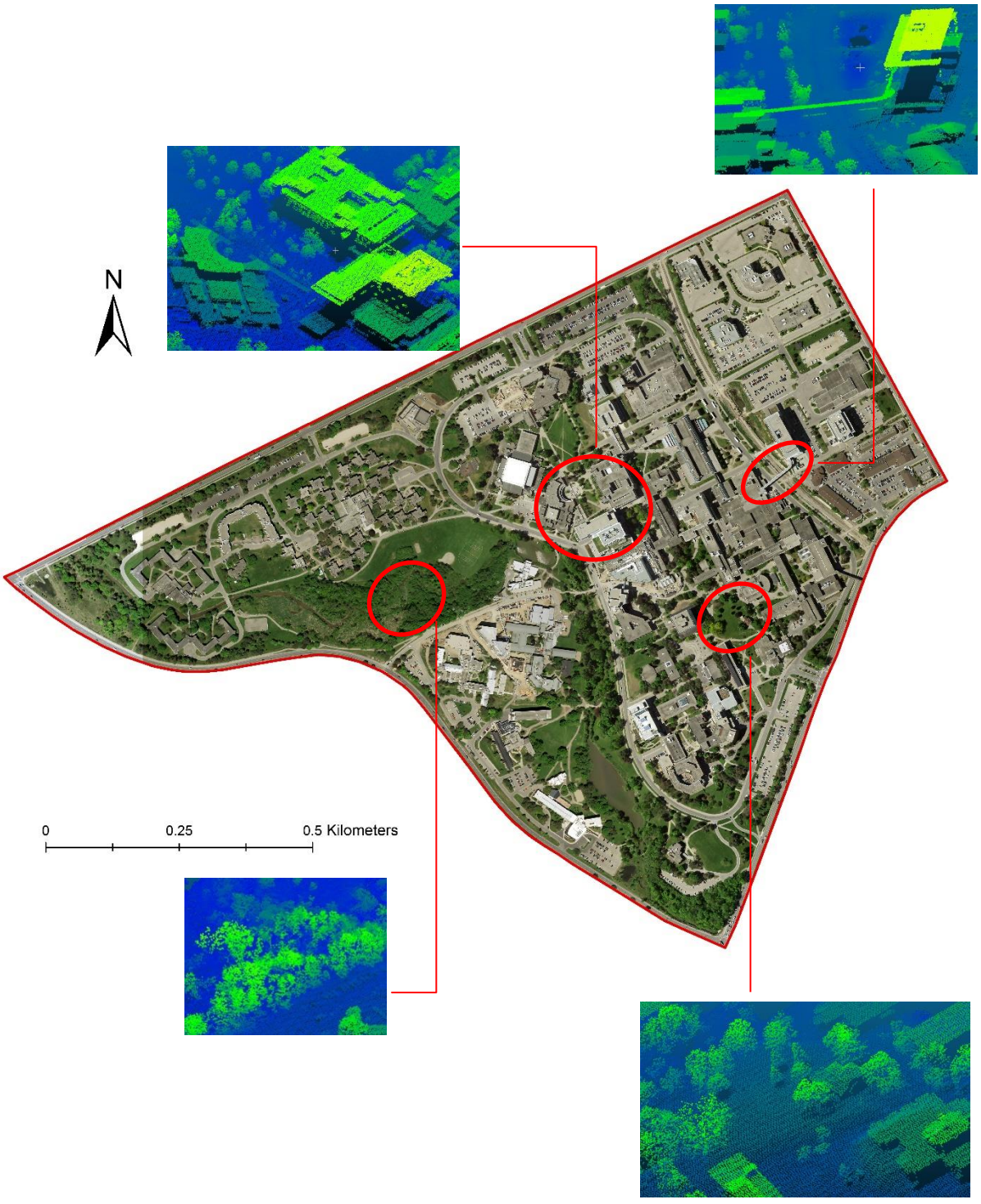


Figure 3.4 Examples of difficult to filter features

3.3 Data Preprocessing

3.3.1 Clip to Study Area

The study area is delineated by four road segments around the main campus of the University of Waterloo (Figure 3.1). To acquire the study area extent, the road polylines were first transformed into a polygon using the feature builder tool in ArcGIS. Then, the ALS dataset was clipped to study area extent using the clip tool in ArcGIS.

3.3.2 Denoising

Prior to DTM extraction, the outliers contained in the raw point cloud need to be removed. Based on elevation, outliers can be categorized into low outliers and high outliers. High outliers are the points that have extremely high elevation compared to its neighbouring points. These points are usually resulted from echoes from birds or other aircrafts. Due to the anomalously high elevation, these points can be easily removed by DTM extraction filters. Low outliers are the points that have extremely low elevation compared to its neighbouring points. These points are typically caused by mechanic errors or multiple reflection. In urban environment, a LiDAR signal may be reflected multiple times by building facades before coming back to the sensor. Since ground and non-ground points are mainly distinguishable by their elevation, low outliers are particularly destructive to the DTM extraction algorithm. Especially for surface-based filters that select local minima as seed ground points and iteratively densify the terrain surface, the presence of low outliers introduces significant error in the first step of the algorithm. Moreover, some filtering algorithms operates under the assumption that the points neighbouring a low point belong to objects (Sithole & Vosselman, 2004), which is a reasonable assumption in most scenarios, however, with the presence of low outliers, the assumption no longer holds, resulting in conspicuous erosion in the filtered surface. Therefore, low outliers need to be removed during preprocessing. The denoising process is completed in CloudCompare using a statistical outlier removal (SOR) filter.

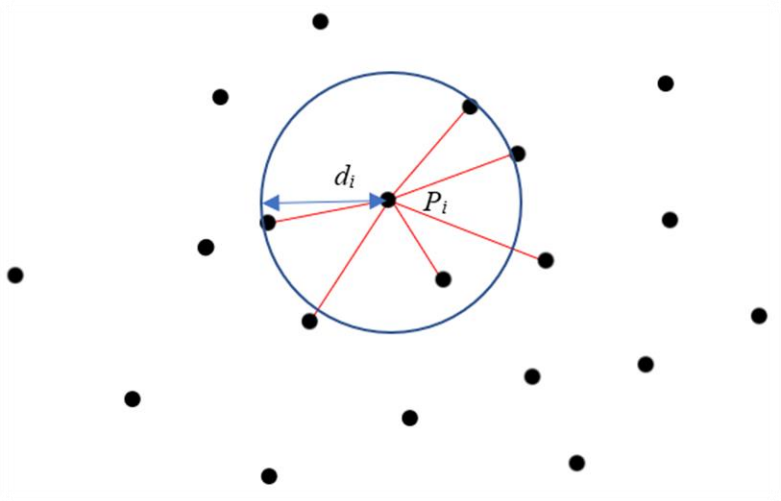


Figure 3.5 Principle of SOR filter

The filter computes the average distance between each point and its six nearest neighbours. Then, assuming the distribution of the calculated distance is normally distributed, any points whose average distance with its six nearest neighbours are greater than the global average distance plus three standard deviations is rejected (Figure 3.5). The original file contains 3,082,000 points, while the number of points after outlier removal is 3,024,044.

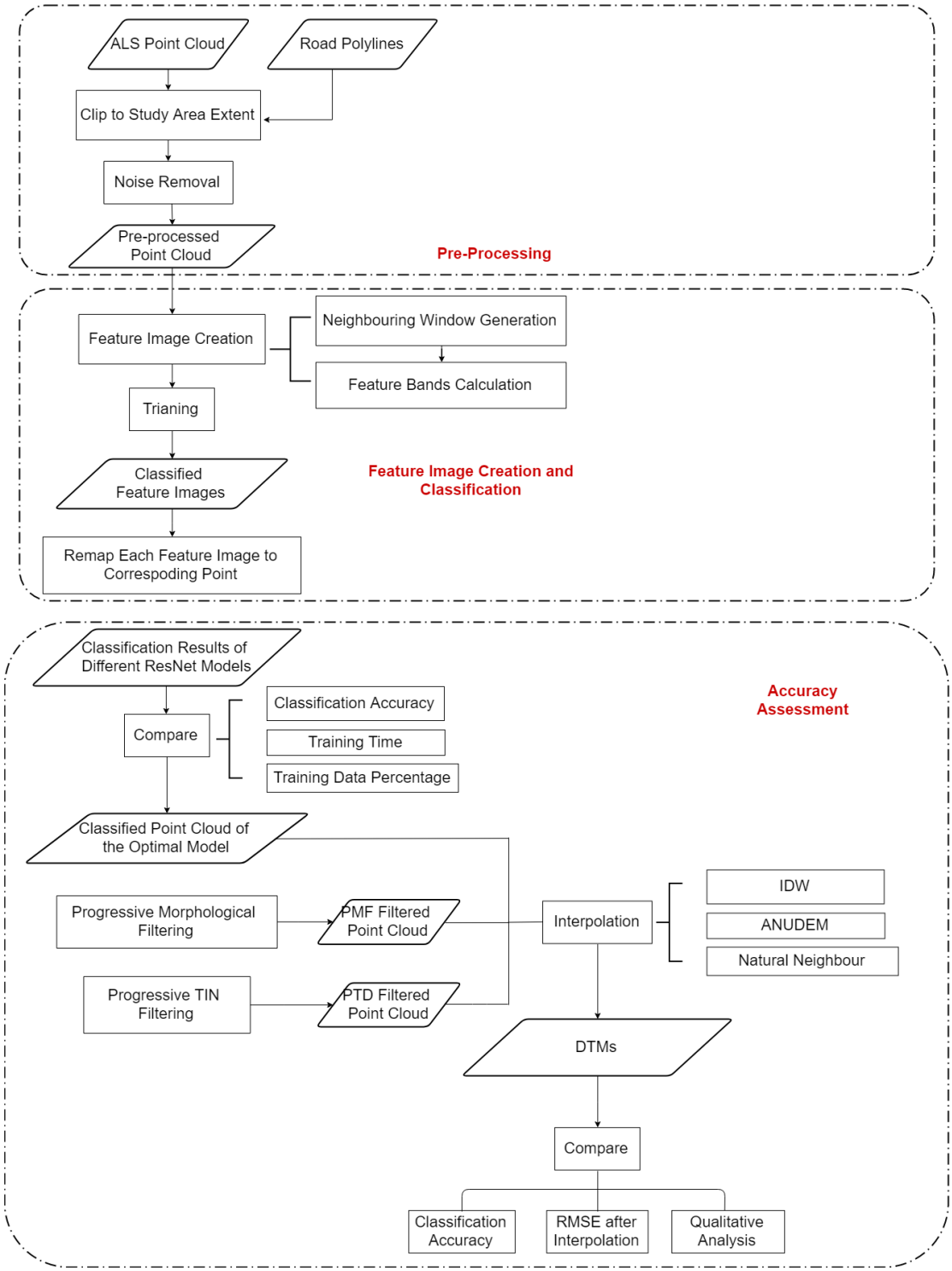


Figure 3.6 Workflow of the methodology

3.4 **Feature Image Creation**

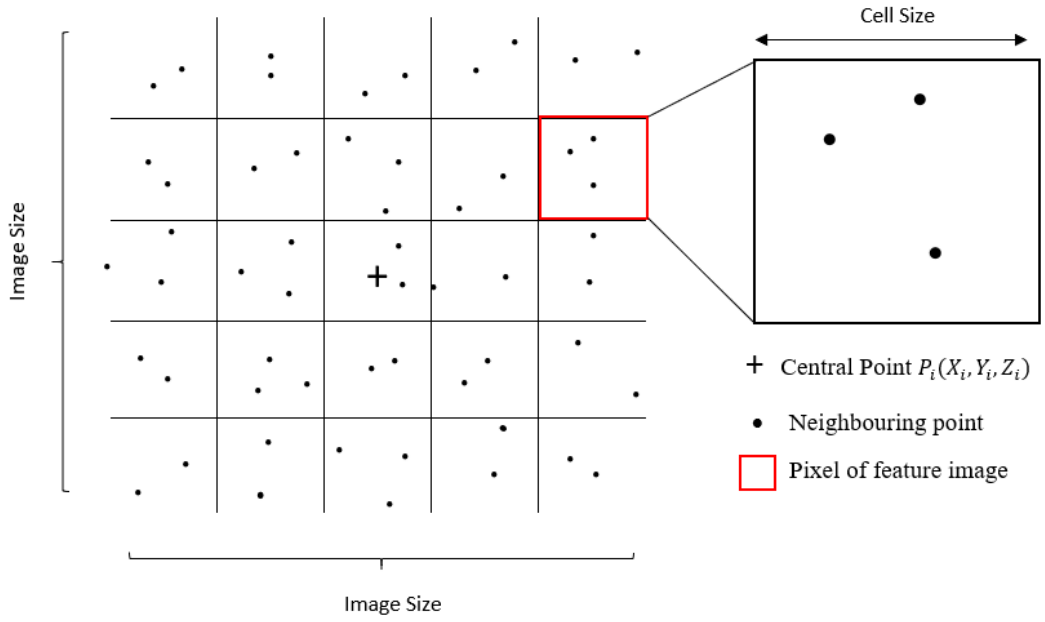


Figure 3.7 Illustration of point-to-image transformation

In order to determine whether a point is ground or non-ground, not only the elevation of the point itself, but also the spatial information of its neighbouring points is needed. For each point P_i, a corresponding image is generated based on the method proposed by Hu and Yuan (2016). The point-to-image transformation was also adopted by Rizaldy et al. (2018) and Politz et al. (2018). The workflow of point to image transformation is described as follows: first, for each point P_i in the point cloud, a corresponding feature image is generated based on its elevation difference with neighbouring point (Figure 3.7). The point P_i is located at the center of this square window and thus will be referred to as the central point. Then, the square window is partitioned into multiple cells based on two parameters: cell size and image size. Image size indicates the number of rows and columns of the image. Cell size indicates the resolution of feature image pixels, which should be set slightly larger than average point spacing. Next, the maximum (Z_{max}), minimum (Z_{min}) and average (Z_{mean}) elevation within each cell is calculated. The last but critical step is to subtract Z_i from Z_{max}, Z_{min} and Z_{mean} to acquire

pixel value for synthetic red, green and blue bands, then, through sigmoid transformation, the elevation differences is mapped into three pixel values between 0 and 255. It is worth noting that although the point to image transformation process may appear similar to rasterization in many ALS related studies, rasterization derives pixel values solely from the points located within each pixel. The point to image transformation, however, considers not only the points' elevations within each pixel, but also the relatively height differences to the central points.

The square window is defined by two parameters: cell size and image size. Since the average point spacing of the point cloud is approximately 1m, to avoid empty cells and make most of the abundant spatial information simultaneously, the cell size is chosen to be 1.5 m, which is slightly larger than the point spacing. The largest building in the dataset has a width of approximately 70 m. To identify such building, a spatial context of approximately double the size is needed. Thus, the image size is chosen to be 128*128 cells, which is equivalent to 196*196 m. The image is denoted as positive sample if P_i is labelled as ground point and vice versa. After the extent of the square window is defined, the value of each cell within the window is mapped as:

$$\boldsymbol{F\_red} = \lfloor \mathbf{256} * \boldsymbol{sigmoid}(\boldsymbol{Z_{max}} - \boldsymbol{Z_i}) \rfloor \tag{3.1}$$

$$\boldsymbol{F\_green} = \lfloor \mathbf{256} * \boldsymbol{sigmoid}(\boldsymbol{Z_{min}} - \boldsymbol{Z_i}) \rfloor \tag{3.2}$$

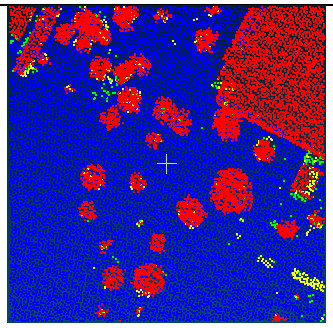
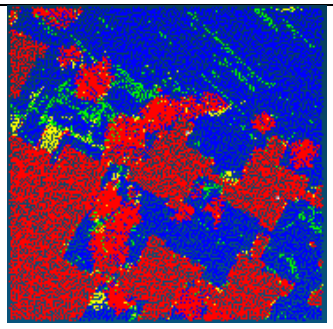
$$\boldsymbol{F\_blue} = \lfloor \mathbf{256} * \boldsymbol{sigmoid}(\boldsymbol{Z_{mean}} - \boldsymbol{Z_i}) \rfloor \tag{3.3}$$

where F_red, F_green and F_blue represent the synthesized pixel values for each band respectively, Z_{max}, Z_{min} and Z_{mean} represent the maximum, minimum and average elevation of points located in each cell. Z_i represents the elevation of P_i. By subtracting the elevation of P_i from the minimum, maximum and average elevation of each cell, the spatial relationship between the central point and its neighbouring points can be thoroughly represented. The sigmoid function is defined in Eq. (3.4), which takes a number x as input and transforms it into a value between 0 and 1. Through this transformation, the elevation differences can be mapped into pixel values between 0 and 255.

$$\boldsymbol{sigmoid}(\boldsymbol{x}) = (\mathbf{1} + \boldsymbol{e}^{-\boldsymbol{x}})^{-\mathbf{1}} \tag{3.4}$$

Examples of point to image transformations are shown in Table 3.2. In the first example, the central point is a ground point, neighbouring trees and buildings all have higher elevation than this point. The trees appear to be bright pink due to their heterogeneous inner structure: the red band is calculated by Eq. (3.1) and thus have high value than the blue and green band. The buildings appear to be white since they are relatively flat, which means the Z_{max}, Z_{min} and Z_{mean} in each cell have similar values, resulting in similar pixel values in three bands. Since the elevation difference between building and ground is large, the pixel values are saturated and appear white. Other ground pixels appear to be grey due to their flat surface and little elevation difference to the central point. In the second example, the central point is a non-ground point from one of the buildings. Other buildings which have similar elevation appear grey, while the building located at the lower left corner has higher elevation and appears to be white. Trees also appear to be pink, but have a darker tone compared to the ground example.

Table 3.2 Point-to-feature-image transformation example

	Point Cloud	Feature Image
Ground		
on-ground		

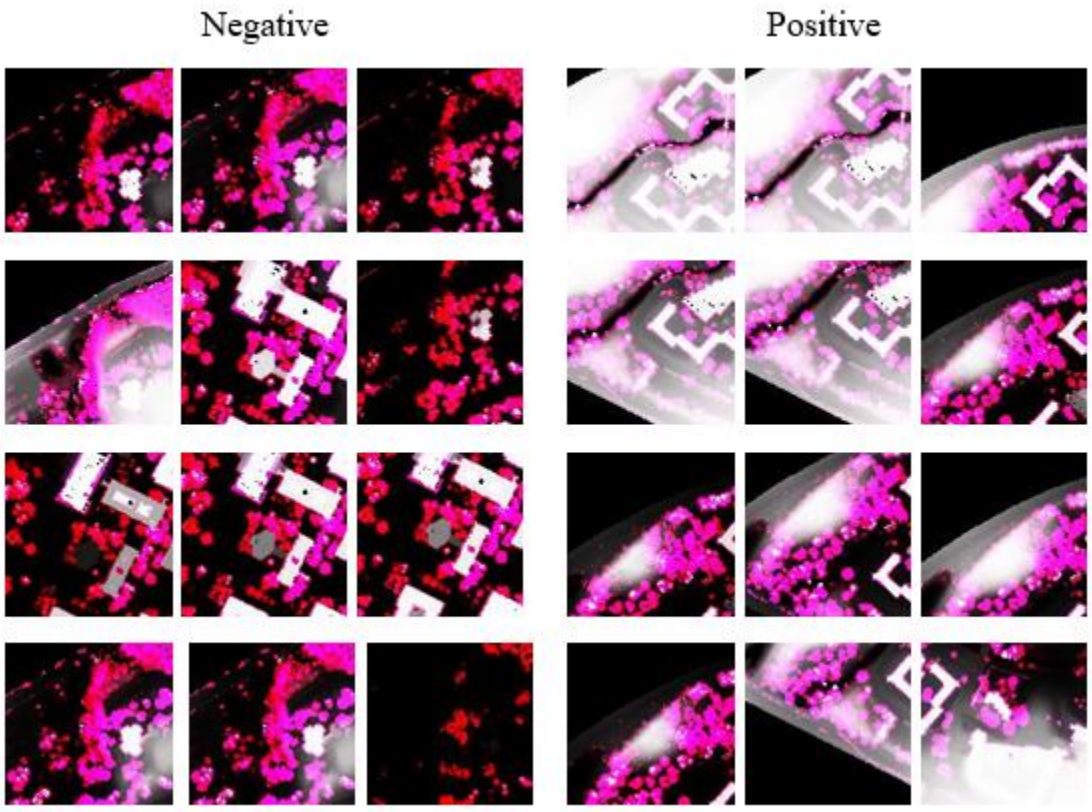


Figure 3.8 Positive and negative feature images

The red band has the highest pixel values among three bands, thus both positive and negative images appear red. Since ground points have lower elevation compared to its neighbouring points, the pixel values calculated by elevation differences will be larger than those of non-ground points. Therefore, ground-point-images generally appear brighter. More examples of ground and non-ground feature images are shown in Figure 3.8 to demonstrate how they can be intuitively distinguished just be visual inspection.

For points located at the edge of the study area, a threshold is applied to minimize the number of empty cells within the feature image: only feature images with less than half empty cells are accepted. This configuration ensures each feature image carries sufficient spatial information for it to be distinguished. The original point cloud contains 3,024,044 points, while

the final generated number of images is 2,991,559. 32,485 points are discarded by this threshold, which comprise of 1.07% of the entire dataset.

3.5 **ResNet**

ResNet was chosen to perform the classification task due to its outstanding performance on the ImageNet dataset. The best performances of popular networks on ImageNet are shown in Table 3.3. The top-N error indicates the percentage of test sample $x_i$whose correct label y_i does not appear in the first N predicted labels. From the table it can be seen that ResNets exceed previous state-of-the-art models' performance. Both top-1 and top-5 error rates achieved better or at least equal results.

Table 3.3 Error rates (%) of single-model results on the ImageNet validation set. (Except + reported on test set) (Source: He et al., 2015)

Method	Top-1 err.	Top-5 err.
VGG (Simonyan and Zisserman, 2015)	-	8.43^{+}
GoogLeNet (Szegedy et al., 2015)	-	7.89
VGG (v5) (Simonyan and Zisserman, 2015)	24.4	7.1
PReLU-net (He et al.., 2015)	21.59	5.71
BN-inception (Ioffe and Szegedy., 2015)	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	19.38	4.49

Compared to plain networks, residual networks can reach much deeper depth while maintain lower complexity (He et al., 2015). The performance of CNNs largely rely on the depth of the network. However, deep networks are difficult to train not only due to the computational complexity, but also because of convergence problems known as

vanishing/exploding gradients (Glorot & Bengio, 2010). Recent advancements in using normalized initialization (He et al., 2015b) and normalized intermediate layers (Ioffe & Szegedy, 2015) largely eased the convergence problem. Nevertheless, even if the network converges, with the increasing depth of network, the training and testing accuracies first saturate and then decrease (He et al., 2015). Experimental results show that such decrease in accuracy is not due to overfitting, since both training and testing accuracies dropped.

The residual networks are designed to ease the training of deep CNNs by adding residual learning blocks to the corresponding "plain" networks (networks that simply stack layers). A typical residual block is shown in Figure 3.9. Assuming the original desired mapping for this block is $H(x)$, the residual network let the stacked nonlinear layers fit a mapping $F(x) = H(x) - x$, by adding an identity mapping x, the residual block still fit the desired $H(x)$. The curved arrow indicates "shortcut connection" that adds the identity mapping to the outputs of the stacked layers. In the case that the dimension of x is different from $F(x)$ due to convolution, a linear projection was performed to transform x into the dimension of $F(x)$. The idea behind such configuration is that optimizing the residual $F(x)$ is easier than optimizing the unreferenced $H(x)$ (He et al., 2015).

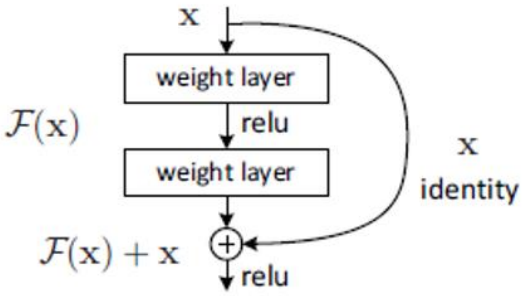


Figure 3.9 Example of a residual block (Source: He et al., 2015)

Apart from the shortcut connection implementation for residual mapping, the functional layers in ResNet is the same as traditional CNNs. In addition to the convolutional layer and pooling layers shown in Table 3.4, batch normalization was applied after each convolution and

before ReLU activation or addition of identity mapping. CNN typically consists of convolutional layers, pooling layers and fully connected layers.

Table 3.4 ResNet Architecture (Source: He et al., 2015)

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

3.5.1 Convolutional Layer

Convolutional layer is the major calculating part in CNN, which use kernels to transform input data into feature maps. The kernel in the convolutional layer connects to a local reception field in the previous layer (either an input image or an intermediate feature map). By sliding over the full extent of the input volume, the output feature map represents the filter response of the input image or feature maps.

A batch normalization (BN) is performed after each convolutional layer and before activation. BN is an effective way to accelerate learning and prevent overfitting. The input data in each mini-batch is normalized using

$$y_i = \gamma \frac{x_i - \mu_B}{\sqrt{{\sigma_B}^2 + \epsilon}} + \beta \tag{3.5}$$

where $B = \{x_1, x_2, \ldots, x_m\}$ denotes current batch, μ_B and ${\sigma_B}^2$ are the mean and variance of the mini-batch, ϵ is a constant added to the mini-batch variance for numerical stability, γ and β are the parameters to be learned (Ioffe & Szegedy, 2015).

A ReLU (Rectified Linear Unit) activation is used after BN to ensure nonlinearity. The ReLU activation computes $F(x) = max(0, x)$, which simply regards all value below zero as zero (Figure 3.10). It has been popular in recent neural networks due to faster convergence speed compared to other activation functions such as sigmoid and TanH.

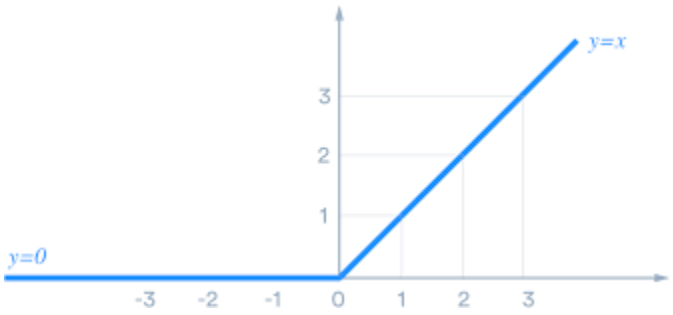


Figure 3.10 ReLU activation function

3.5.2 Pooling Layer

Pooling layer is also referred to as down sampling layer. It usually follows one or several convolutional layers to reduce the dimensions of feature map. The pooling layer reduces the dimension of feature maps, thus reduce the parameters in the network and has certain effect in prevents overfitting. Two pooling layers are implemented in ResNet, a max pool layer and an average pool layer, which are the two most frequently used pooling strategies (Guo et al., 2016). Similar to convolutional layer, two parameters are involved with the pooling layer: kernel size and stride. The max and average pooling layer takes the max and average value in each of its reception field respectively and pass this value to the next layer. The kernel size of max pooling layer in ResNet is three, while the kernel size of average pooling layer depends on the size of input image and number of output classes.

3.5.3 Fully Connected Layer

The neurons in fully connected layer are connected to every neuron in the previous layer. Fully connected layer is the last layer in ResNet, which outputs the probability of each input belongs to a certain class. The dimension of the fully connected layer depends on the number

of classes to be predicted. In our case, the fully connected layer will have two neurons since the task is a binary classification problem.

3.6 Training Configuration

The original ALS dataset contains five classes: ground, low vegetation, medium vegetation, high vegetation and building, and low outliers. Low outliers are removed during noise removal stage. The rest of the points are transformed into feature images based on the workflow described in Section 3.4. Points that are labelled as ground are regarded as positive samples. All other points are regarded as negative samples. The training was performed on a computer with a NVIDIA GeForce GTX 1080 GPU, an Intel CPUi7-9700k 3.6GHz with 8 cores, and 32 GB of RAM.

3.6.1 Transfer-Learning

The training was fine-tuning the ResNet models with pre-trained ImageNet weights on the feature images dataset. ImageNet is a comprehensive image database that contains more than 14 million hand-annotated images in more than 20,000 categories (Deng et al., 2009). Transfer-learning means that the network is initialized by weights trained on another dataset rather than random initialization. Training process can be largely accelerated by using pre-trained model, since the network already learnt some critical features.

In order to use pre-trained weights, the top two layers of the network need to be modified, which are the average pooling layer and the fully connected layer. As sown in Table 3.4, the last convolutional layers of ResNet18 and ResNet50 are of dimension (3*3, 512) and (1*1, 2048), respectively. The first dimension of subsequent fully connected layer should be the same as the number of kernels. Since our problem is a binary classification problem, the fully connected layer is of dimension (512, 2) and (2048, 2) for ResNet18 and ResNet50, respectively. The second last layer, which is the average pooling layer, has a kernel size of 4 and stride=1.

3.6.2 Loss Function

Cross Entropy Loss is used to measure the loss of neural network. The loss is calculated by:

$$J = -\frac{1}{N}\sum_{n=1}^{N}[y_n log(p_n) + (1 - y_n)\log(1 - p_n)] \tag{3.6}$$

since our problem is a binary classification problem, y_n can only be 0 or 1. Eq. (3.6) could be rewritten as Eqs. (3.7) and (3.8) for better understanding.

$$If\ y = 1,\ J = -\frac{1}{N}\sum_{n=1}^{N} y_n log(p_n) \tag{3.7}$$

$$If\ y = 0,\ J = -\frac{1}{N}\sum_{n=1}^{N}(1 - y_n) log(1 - p_n) \tag{3.8}$$

where p_n is the predicted probability of an input belong to its true class, y_n is the label of current input, and N is the batch size. From Eqs. (3.7) and (3.8) it can be seen that for both classes, the losses increase if p_n is low.

3.6.3 Learning Rate

Learning rate is a hyperparameter that controls the speed of gradient descent. If the learning rate is too small, the convergence will be slow or trapped in plateau. Otherwise, if the learning rate is too big, the model may fail to find local minima. In batch gradient descent, the weights are updated by

$$\theta_t = \theta_{t-1} - \alpha \nabla_\theta J_t(\theta_{t-1}) \tag{3.9}$$

where θ_t is the weights to be updated, α is learning rate, $\nabla_\theta J_t(\theta_{t-1})$ is the derivatives of cost function $J(\theta_{t-1})$ at step t with respect to θ. Since it is difficult to specify a predefined learning rate, some optimization techniques are proposed to adjust the learning rate during training. Commonly used optimization algorithms include stochastic gradient descent with momentum, Nesterov accelerated gradient, RMSProp, Adam, etc.

Gradient descent with Adam (Adaptive Moment Estimation) optimizer is used due to its outstanding performance on the MNIST dataset (Kingma & Ba, 2015). As shown in Figure 3.11, Adam not only has faster convergence, but the training cost at convergence is lower than other optimization techniques.

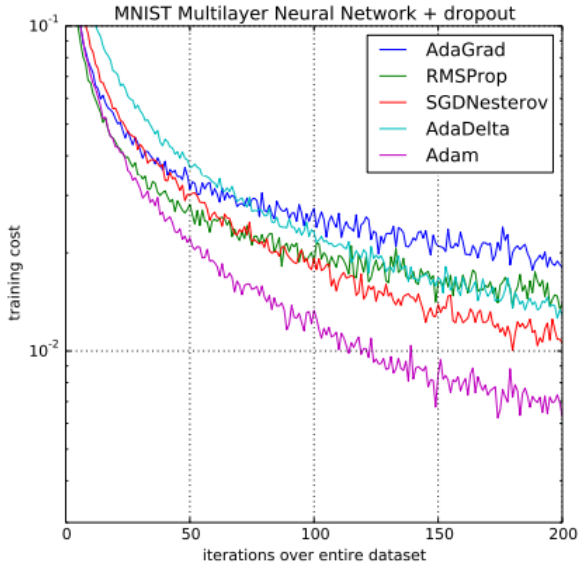


Figure 3.11 Adam performance on the MNIST dataset (Source: Kingma & Ba, 2015)

Adam is an algorithm that computes the adaptive learning rate, but the ultimate purpose is to improve weights updating. Apart from the learning rate, three parameters are used in the Adam algorithm: β_1, β_2, and ε. The weights update rule is presented as follows (Kingma & Ba, 2015):

$$\boldsymbol{g_t = \nabla_\theta f_t(\theta_{t-1})} \tag{3.9}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{3.10}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_1) {g_t}^2 \tag{3.11}$$

$$\widehat{m_t} = \frac{m_t}{1-{\beta_1}^t} \tag{3.12}$$

$$\widehat{v_t} = \frac{v_t}{1-{\beta_2}^t} \tag{3.13}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\widehat{v_t}}+\varepsilon}\widehat{m_t} \tag{3.14}$$

where β_1 is the exponential decay rate for the first moment estimates, default value is 0.9, β_2 is the exponential decay rate for the second moment estimates, default value is 0.999, ε is the very small number to prevent division by zero, default value is $1e^{-8}$, g_t is the derivative of cost function at step t, in the case the cost function is calculated by Cross Entropy Loss

presented in Eq. (3.6), m_t and v_t are the estimates of gradients at the first and the second moments, which as initialized by vectors of zeros.

3.7 Interpolation

Interpolation is the process of transforming the extracted ground points into a continuous surface representing terrain elevation. There are various ways to produce a continuous surface using the input elevation points or contour line. Depending on whether or not spatial correlation is utilized, interpolation can be categorized into deterministic methods and geostatistical methods. Deterministic methods predict values based only on the neighbouring measured values. The rationale behind this is that "everything is related to everything else, but near things are more related than distant things" (Tobler's first law of geography). Commonly used deterministic methods include IDW, spline, natural neighbour, etc. These methods are easy to use and generally do not make assumption of the data distribution. Geostatistical methods account for the distribution and spatial autocorrelation of the data. Spatial correlation is the degree of similarity between nearby objects. When using geostatistical methods, not only the measured values of neighbouring locations, but also the overall spatial autocorrelation will be incorporated. However, some geostatistical methods such as ordinary kriging or simple kriging operate under the assumption that the data is normally distributed. As shown in Figure 3.3, the ground points are not normally distributed. Thus, three deterministic methods are chosen to create DTMs from extracted ground points.

3.7.1 Inverse Distance Weighting

IDW is a deterministic interpolation method that predicts the value at location p using

$$Z_p = \frac{\sum_{i=1}^{N}(\frac{Z_i}{d_i k})}{\sum_{i=1}^{N}(\frac{1}{d_i k})} \tag{3.15}$$

where N is the number of neighbouring points, Z_i is the elevation of i_{th} neighbouring point, d_i is the distance between the i_{th} neighbouring point and location p, and k is the power of distance. The value of Z_p is essentially a weighted average of Z_i (i $= 1, \ldots,$ N). IDW explicitly makes

the assumption that things nearer are more similar than things apart, thus, measured points that are spatially closer to the interpolated location will be assigned higher weights. The method is essentially a weighted average approach: as distance between the measured point and the interpolating location increases, the inverse of the distance decreases and therefore the weight. The power k adjusts the speed of the weights diminish with distances. If $k = 0$, Eq. (3.15) will calculate a simple average of the neighbouring points. As k increases, the weights of distant points will decrease rapidly. IDW is an exact interpolator, which means that the interpolated value will not exceed the minimum or maximum of the elevations used to predict the interpolated value.

3.7.2 ANUDEM

ANUDEM is an interpolation method that creates grid DEM using locally adaptive elevation gridding (Hutchinson et al., 2011). Although this method is designed to work with drainage structure and hydrologically relevant topographic data, it can also produce outstanding quality DEMs with regular elevation point data. The interpolation algorithm is described in Hutchinson (2000). ANUDEM uses a spline fitting method that is computationally efficient and is capable of working with arbitrarily large dataset. A multi-grid method is proposed to generate the DEM starting from coarse grid, then refined the resolution on successive finer grid.

3.7.3 Natural Neighbour

Natural neighbour, also known as “area-stealing” interpolation, is also an exact interpolator. The predicted values do not exceed the minimum and maximum value of input elevations. Similar to IDW, the natural neighbour method does not infer any trend from the input data, instead, it only considers the elevation value of the interpolating location’s direct neighbours, and derives predicted values using weighted average. The key component in natural neighbour interpolation is the Voronoi diagram, which corresponds to the Delaunay triangulation in terms that the Voronoi diagram can be produced by connecting all the circumcircles centres of the Delaunay triangulations (Figure 3.12). First, Voronoi diagram is

created for each of the elevation points. Then, for every location p that needs to be interpolated, a Voronoi polygon is created. Next, points Z_i $(\mathrm{i} = 1, \ldots, \mathrm{N})$ whose Voronoi polygon overlaps with the polygon of location p are defined as p's natural neighbours. Weights are assigned to the natural neighbours based on the overlapping area between the Voronoi polygon of p and the polygons of Z_i $(\mathrm{i} = 1, \ldots, \mathrm{N})$. The predicted value at p will be the weighted average of its natural neighbours.

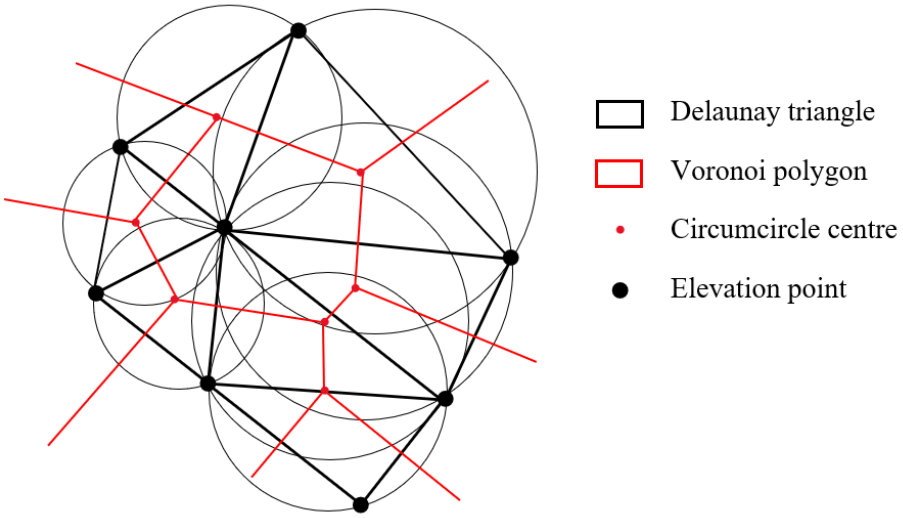


Figure 3.12 Illustration of Voronoi diagram and Delaunay triangulation

3.8 Methods for Accuracy Assessment

The proposed method is compared to two widely implemented filters: namely the Progressive Morphological Filter (PMF) (Zhang et al., 2003) and the Progressive TIN Densification Filter (PTD) (Axelsson, 2000). The filters will be evaluated based on three criteria: point cloud classification accuracy, RMSE (root mean squared error) of interpolated DTM compared to true label, and qualitative analysis.

3.8.1 Point Classification Accuracy

A confusion matrix is generated for accuracy assessment (Table 3.5). Following the accuracy assessment method used in Sithole and Vosselman (2004), three measurements are

used to evaluate the classification accuracy: Type I (false negative), Type II (false positive) and total error. Unlike the user and producer accuracies presented in most remote sensing studies, these three measurements are specifically used for DTM filtering researches. Similar to most classification tasks, a trade-off has to be made between Types I and II errors. Sithole and Vosselman (2004) suggested that DTM extraction techniques should be designed to minimise type I errors, since Type II errors correspond to unremoved object points on terrain surface, which are relatively easier to be fixed by manual post editing, while Type I errors correspond to gaps in terrain.

Table 3.5 Confusion Matrix

	Predicted label					
Reference		Ground	Non-ground	Total	Type I error (%)	e
	Ground	a	b	a + b	Type II error (%)	f
	Non-ground	c	d	c + d	Total error (%)	g

where a is the number of ground points been correctly identified,

b is the number of ground points been identified as non-ground,

c is the number of non-ground points been identified as ground, and

d is the number of non-ground points been correctly identified.

e is the percentage of Type I error, which is calculated by $\frac{b}{a+b} * 100$, describing the amount of ground points been misclassified as non-ground.

f is the percentage of Type II error, which is calculated by $\frac{c}{c+d} * 100$, describing the amount of ground points been misclassified as non-ground.

g is the percentage of total error, which is calculated by $\frac{b+c}{a+b+c+d} * 100$.

3.8.2 RMSE

Misclassifications such as identifying low vegetation as ground, or false removal of ground points, do not necessarily affects the quality of interpolated DTM. However, the effects

of misclassification on DTM quality cannot be directly reflected by point classification accuracy. Thus, the filtered ground points and the true ground points are interpolated to produce raster DTMs. By comparing the RMSE between DTMs, a more thorough assessment can be made. The quality of DTMs is evaluated based on their RMSEs from true ground points, which is calculated by

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(Z_i - Z_{DTM})^2}{N}} \tag{3.16}$$

where Z_i is the elevation of a true ground points $P_i(X_i, Y_i, Z_i)$, Z_{DTM} is the elevation value of DTM pixel at location (X_i, Y_i), N is the number of true ground points.

3.8.3 Qualitative Analysis

Qualitative analysis is presented by inspecting the instances of Types I and II errors. It is an effective way to detect the situations under which misclassifications occur, which helps to propose possible solutions toward such situations. For example, if it is known that vegetation is likely to be misclassified, then introducing spectral information may improve the accuracy. Special terrains such as discontinuity near water surface, buildings with varied sizes are also worth examining.

3.9 Chapter Summary

This chapter describes the proposed workflow, which includes three parts: data preprocessing, feature image generation and training, and accuracy assessment. Data preprocessing includes clip to study area and noise removal. Low outliers labelled by ASPRS classification code are directly removed. An additional statistical outlier filter is applied to remove other noises and high outliers using CloudCompare. The process of transforming points to feature images is described in Section 3.4. The features images are then used as input for ResNet18 and ResNet50. The ResNet architecture and training configuration is described in Section 3.6. Finally, accuracy assessment is conducted on the proposed model. The model

is then compared with two traditional filtering methods (PMF and PTD) with respect to three aspects: point cloud classification accuracy, RMSE, and qualitative analysis.

Chapter 4
Results and Discussion

This chapter presents the experimental results as well as discuss the scientific findings. Section 4.1 compares three ResNet models' performances based on the trade-off between classification accuracy, training time and training data percentage. Section 4.2 presents the classification accuracy of proposed workflow and visually compares the extracted DTM with original DSM. Section 4.3 compares the proposed workflow with two traditional filters. Section 4.4 summarizes this chapter.

4.1 Comparison of different ResNet models

Three ResNet models are used in this study: ResNet18, ResNet34 and ResNet50. The architectures of these models are shown in Table 3.4. As the depth of the network increases, it is capable of representing more complex features. However, the computational time as well as required memory and storage space are also increasing. Also, complex networks may overfit the dataset and fail to generalize well to validation and testing dataset. Thus, to select the optimal network, three ResNet models are compared with four different training data rates. After experimenting on the validation dataset, learning rate of 0.001 and drop out of 0.2 are chosen to train the models.

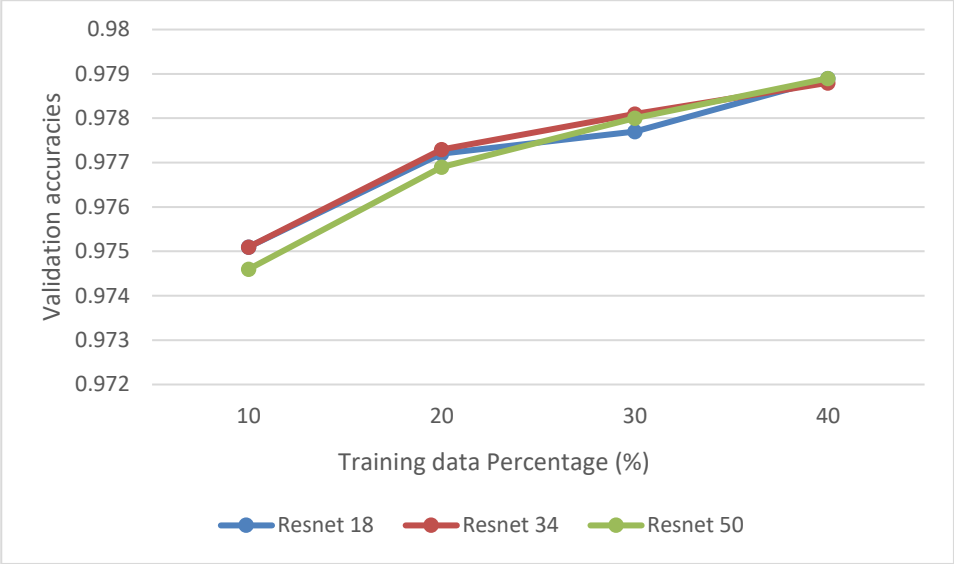


Figure 4.1 Validation accuracies of different models using different training data percentages

As shown in Figure 4.1, the validation accuracies of three models do not differ significantly, which means ResNet18 is sufficient for the classification task. On the other hand, adding more training data can effectively improve the validation accuracy. With 10% of the training data, the average accuracy achieved by three models is 97.49%, while with 40% of the training data, the average accuracy achieved is 97.89%. However, this improvement is relatively trivial since the validation accuracy only improves 0.04%. The amount of time used to train each model is shown in Figure 4.2. It can be seen that as the number of layer increases, the training times also increase. While ResNet 18 and ResNet 50 yield similar results, the amount of time used to train ResNet50 almost tripled. Thus, ResNet 18 is the best model since it provides similar results with deeper model with much shorter time. Training time increases linearly with the amount of input data. For ResNet18, 34 and 50, the time used to train one epoch on 10% of the data is 4m25s, 7m33s and 12m27s, respectively.

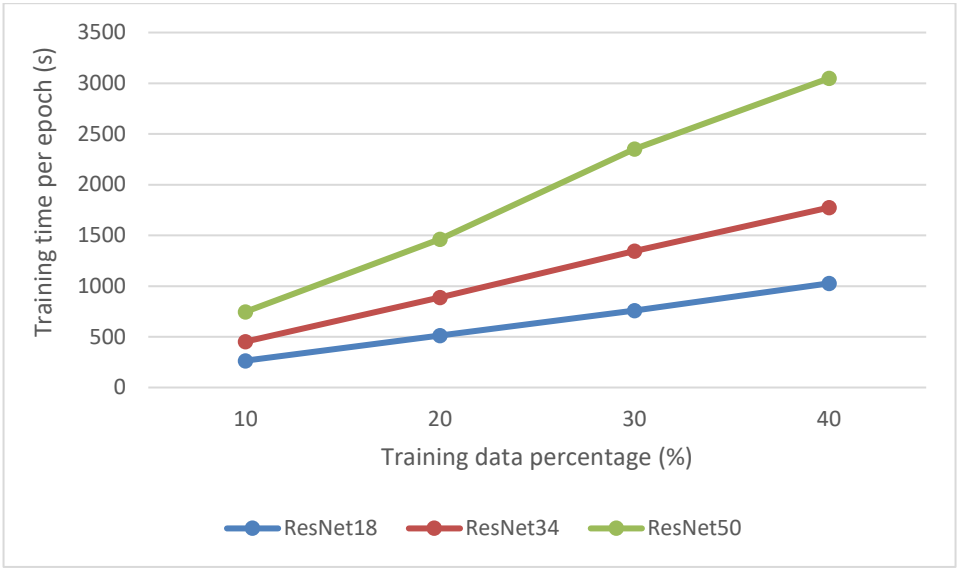


Figure 4.2 Training time of different models

After determining the most suitable network, seven different training percentages are selected to train ResNet18 to determine the best input data rate (Figure 4.3). The training percentage increases from 10% to 70%, while validation percentage is held constant at 10% to make a fair comparison. A trend can be observed that the validation accuracies increase with the volume of training data. However, the amount of increment is trivial: only 0.06% overall improvement in validation accuracy. With every 10% of increase in training data rate, there will be approximately 0.01% increase in validation accuracy.

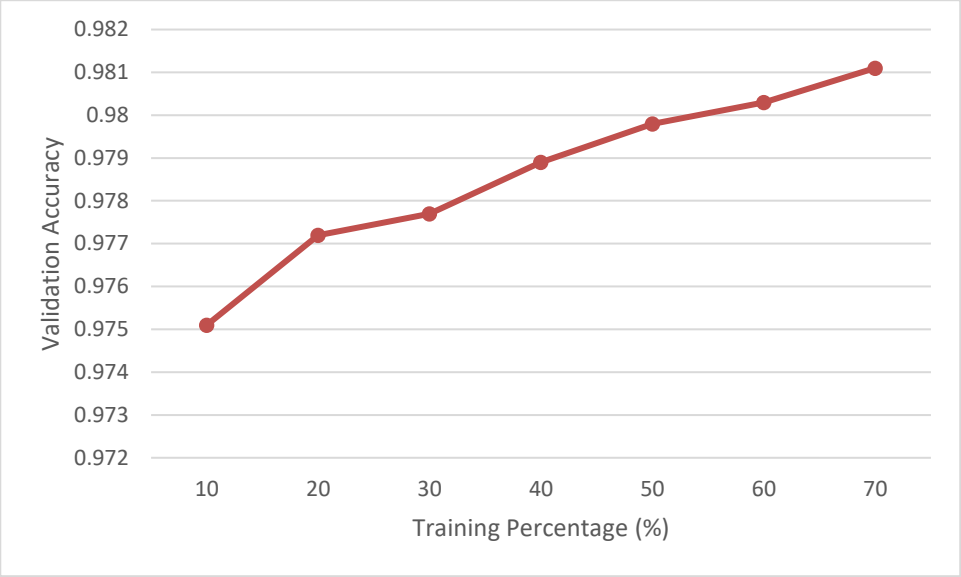


Figure 4.3 Validation accuracies of ResNet18 using different training data percentages

By using pretrained weights on ImageNet, the training process is largely accelerated. The model can achieve satisfactory results within only a few epochs and relatively small training data. The training and validation accuracies of the first 20 epochs are shown in Figure 4.4. The accuracies are tested using different percentages of training data from 10% to 40%, while the validation percentage is held constant at 10%. It can be seen that after the first epoch of training, the validation accuracy exceeds 96.5%, which is already comparable with the PTD results reported in Section 4.3.1. The validation accuracies become stable after only four epochs of training.

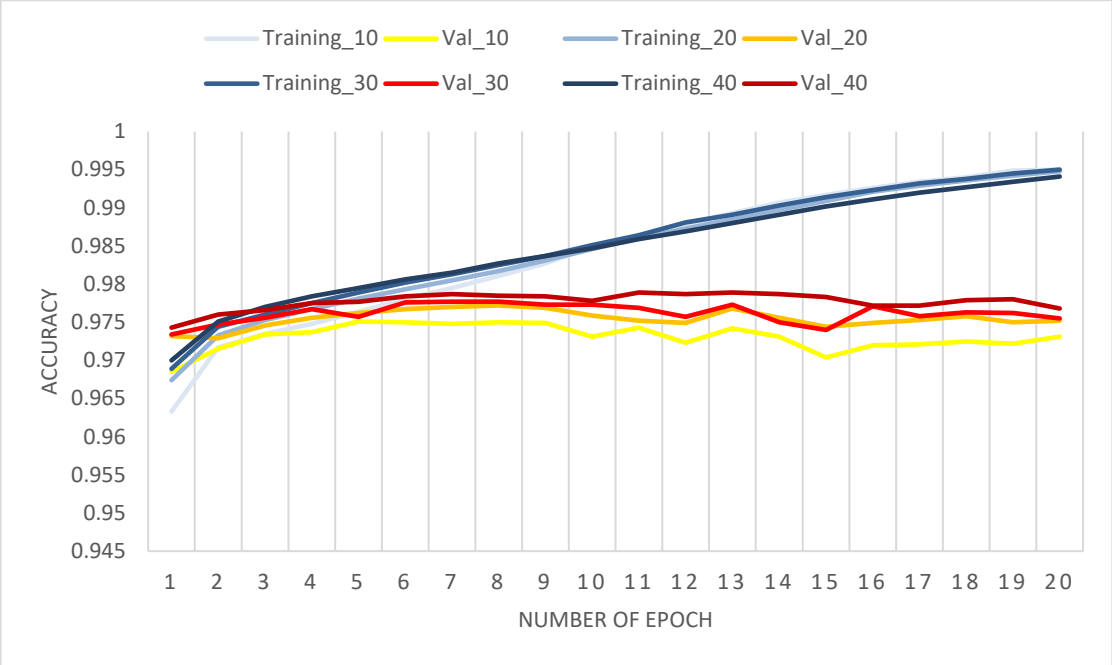


Figure 4.4 Training and validation accuracies of 20 epochs for different training data percentages

A scene near Building Hagey Hall in the UW campus is presented to make a visual comparison of models trained by 10% (Model A) and 70% (Model B) of the data (Figure 4.5). White indicates correctly classified ground points, grey indicates correctly classified non-ground points, green indicates Type I error and red indicates Type II error. It can be seen that Model A has a few Type II errors. Specifically, in the first case, the bare lawn between Buildings Environment-1 and Hagey Hall is misidentified as non-ground. This case is particularly difficult to classify since it is beside a staircase. The abrupt rise in elevation cause it to be higher than neighbouring ground points and thus difficult to discriminate. In the second case, the ramp connecting Buildings Enivronment-2 and PAS is identified as non-ground. This case is controversial as the ramp is partly connected with ground, and partly suspended, making it difficult to draw a clean boundary between ground and non-ground.

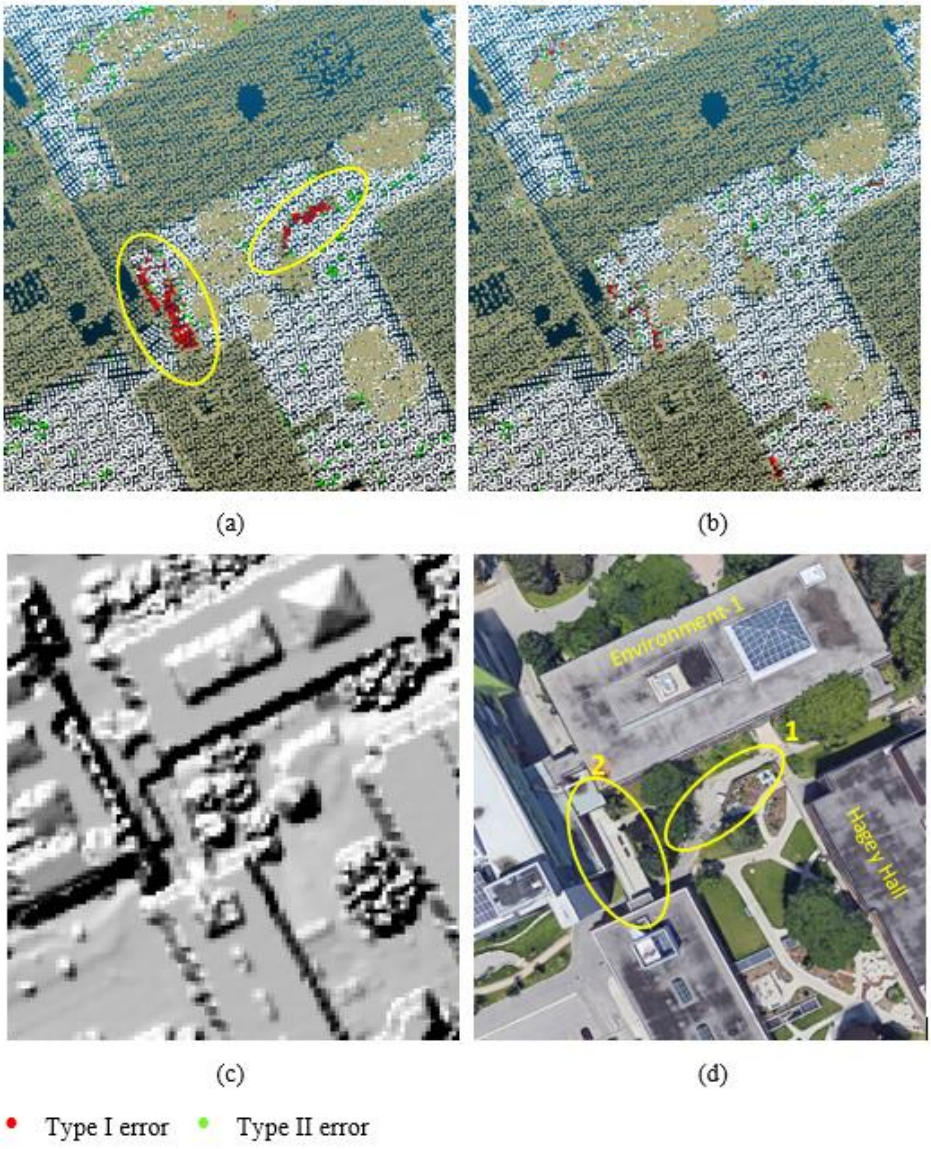


Figure 4.5 Comparison of Model A and Model B. (a) Model A, (b) Model B, (c) Hillshade image, and (d) aerial image from Google Map

4.2 Performance of DTM Extraction

Since both increasing the complexity of the model and increasing the amount of training data do not improve the classification result much, the simplest model with least training data is chosen due to the efficiency in training time and low requirement of labeling. The classification result of ResNet 18 with 10% of training data is presented in Figure 4.6. Red indicates occurrence of Type I errors while green indicates occurrence of Type II errors. Very few Type I error is present in the scene, which means that the terrain points are largely preserved. On the other hand, a few Type II errors can be observed, especially along the railway, where the occurrences of shrub tend to be misclassified.

Figure 4.6 Classification result of proposed workflow with 10% of training data

The classification confusion matrix is shown in Table 4.1. The proposed method can achieve high classification accuracy. The percentages of Type I, Type II and total error are 0.522, 4,84 and 2.43. Also, the system is biased towards making Type II error, which is favourable according to Sithole and Vosselman (2003), filters should strive to minimise Type I errors, since Type II errors are caused by misidentifying off-ground objects as ground. Such errors are typically conspicuous and are relatively easier to remove. On the other hand, Type I errors are the misclassification of ground as non-ground, which result in gaps in terrain and thus difficult to correct.

Table 4.1 Classification confusion matrix of ResNet

	Predicted label					
Reference		Ground	Non-ground	Total	Type I error (%)	0.522
	Ground	1,660,886	8,718	1,669,604	Type II error (%)	4.84
	Non-ground	63,957	1,257,998	1,321,955	Total error (%)	2.43
	Total	1,724,843	1,266,716			

Since the study area covers a large area, two zoomed in scenes are selected to visually present the filtering details. Shaded images of DSM and DTM are created for better visualization. Figure 4.7 shows the filtering results of buildings, roadside trees and cars in parking lots. Figure 4.8 shows the filtering results of dense vegetation. It can be seen that most of the off-ground objects are correctly removed, while the terrain characteristics are preserved.

Figure 4.7 Buildings before (top) and after (bottom) filtering

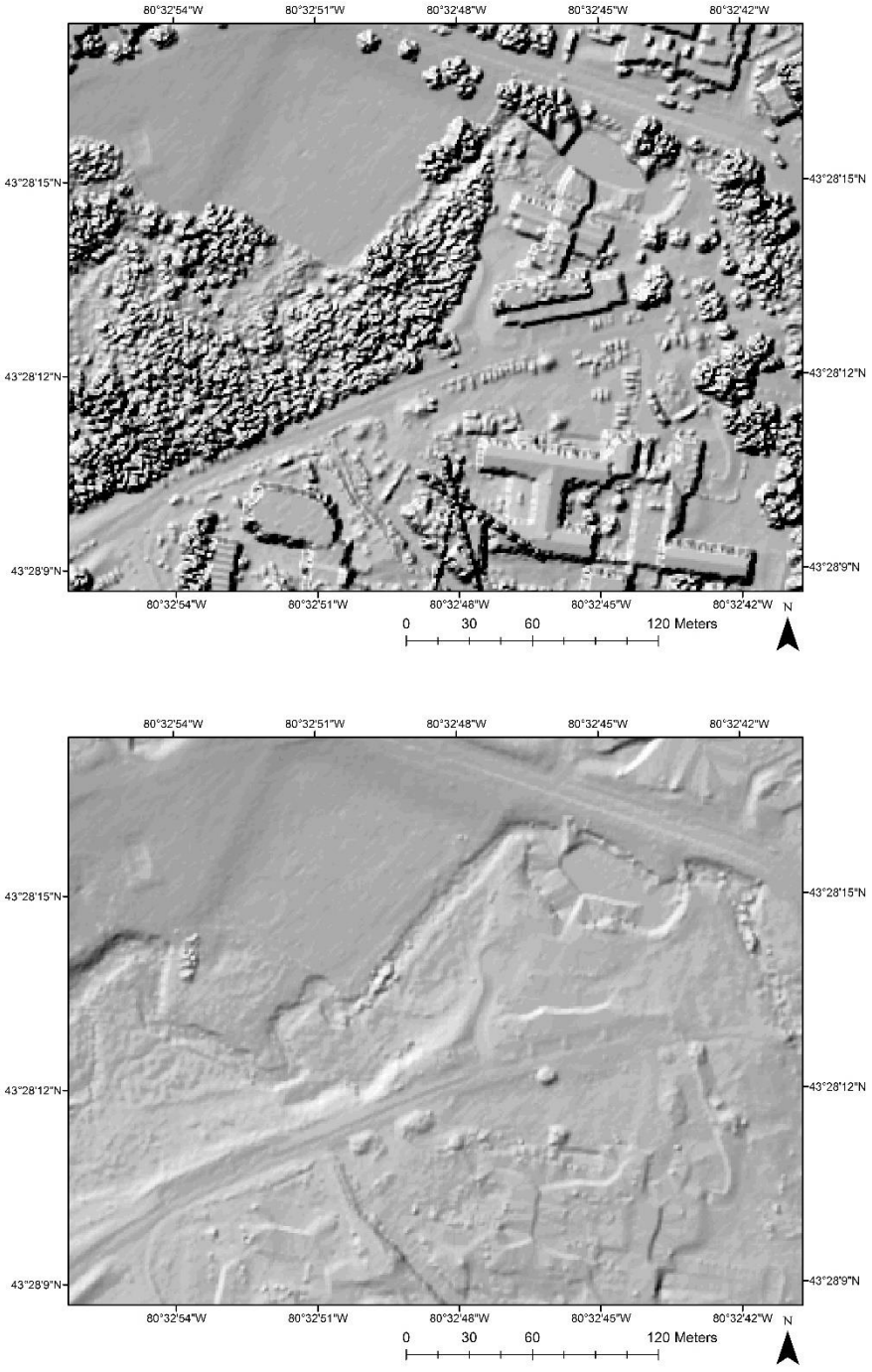


Figure 4.8 Vegetation before (top) and after (bottom) filtering

4.3 **Compare with Traditional Filters**

The proposed method is compared with two traditional filters, namely the Progressive Morphological Filter (PMF) (Zhang et al., 2003) and Progressive TIN Densification filter (PTD) (Axelsson, 2000). PMF is a morphological based filter whose major component is a morphological opening operation, which consists of an erosion followed by dilation. For a point $p(X, Y, Z)$, the erosion (e_p) and dilation (d_p) are defined as:

$$d_p = max_{(X_p,\ Y_p) \in w}(Z_p) \tag{4.1}$$

$$e_p = min_{(X_p,\ Y_p) \in w}(Z_p) \tag{4.2}$$

where points (X_p, Y_p, Z_p) represents the point p's neighbour within the window size *w*.

Erosion removes non-ground objects that have smaller size than current window and shrinks objects that have larger size than current window, while dilation restores the object that have larger size than current window. A threshold is enforced in each iteration to eliminate the false removal of ground points. By gradually increasing filter window size, non-ground objects can be removed while ground points are preserved. The filter can achieve high accuracy such as 3% total error as reported in Zhang et al. (2003). In addition, it is easy to understand and implement. Two widely used open source libraries have implemented this filter, namely the Point Cloud Library (PCL) and the Point Data Abstraction Library (PDAL) .

PTD is a surface-based filter that approximates ground surface by iteratively select candidate ground points. First, a set of initial ground points are selected as seed points. These initial points are highly confident ground points and typically have the lowest elevation within certain neighbourhood. An initial sparse TIN is generated using seed points. Then, candidate ground points are iteratively added to the TIN network based on two thresholds: angles to the existing TIN nodes and distance to the TIN surfaces. PTD is known for its ability to handle surface discontinuity, which is an asset in filtering urban areas. This method has been implemented in the commercial software TerraScan and Lastool.

An overview of ResNet, PTD and PMF classification results, as well as the hill shade image of the interpolated point cloud are shown in Figure 4.9. ResNet and PTD classified scenes appear “clean”, with few errors, while clusters of Type I error can be observed in the PMF scene.

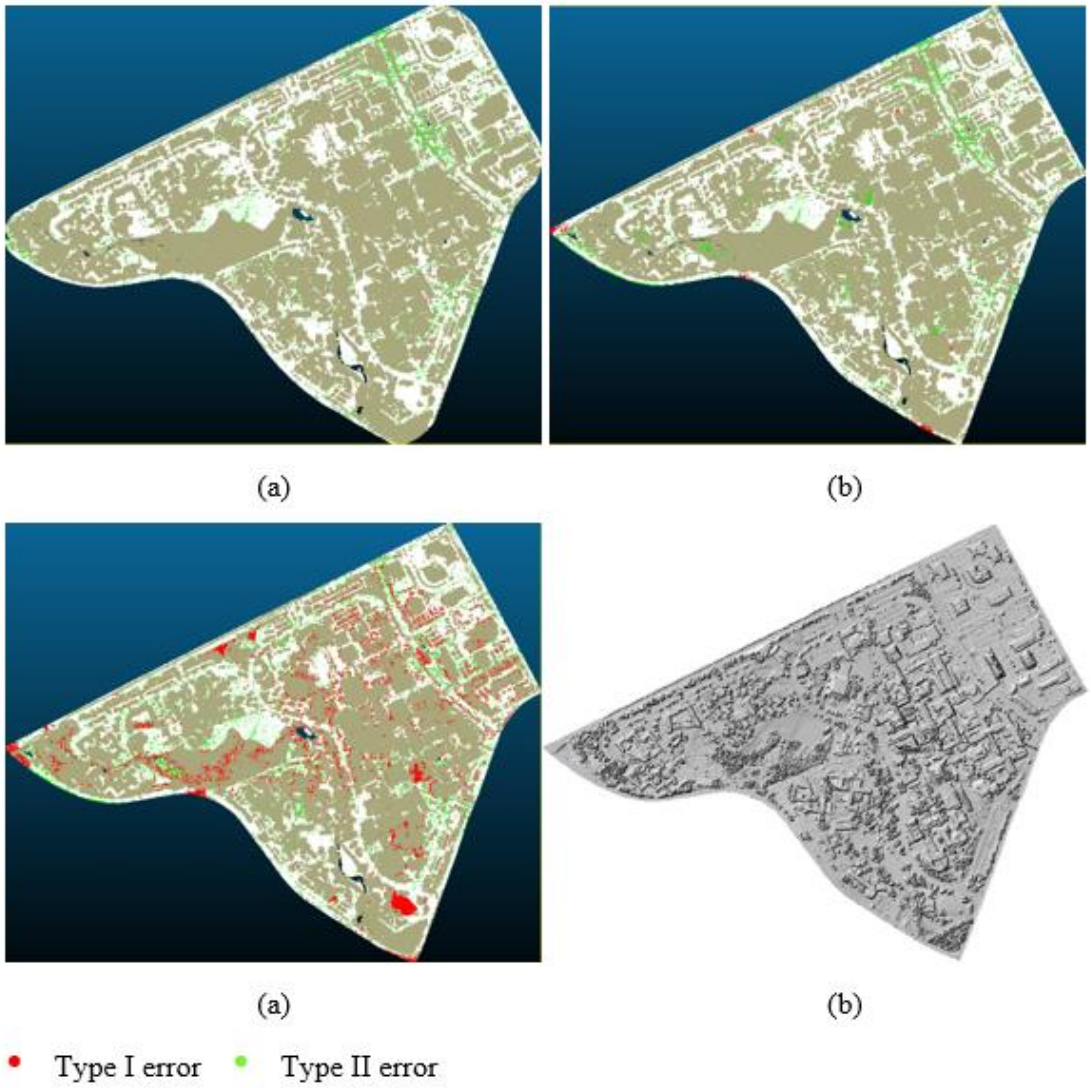


Figure 4.9 Overview of classification results. (a) ResNet, (b) PTD, (c) PMF, and (d) Hillshade image

4.3.1 Point-wise Classification Accuracy

To make a fair comparison, only the classification result of ResNet 18 using 10% of the training data is presented. Although the other models perform slightly better than the selected model, they were given too much training data and the information of true label is considered unfair to other filters. The classification confusion matrix of PMF and PTD are presented in Table 4.2 and Table 4.3.

Table 4.2 Classification confusion matrix of PMF

	Predicted label				
Reference		Ground	Non-ground	Total	Type I error (%)	7.82
	Ground	1,568,844	133,115	1,701,959	Type II error (%)	11.62
	Non-ground	153,575	1,168,510	1,322,085	Total error (%)	9.48
	Total	1,722,419	1,301,625		

Table 4.3 Classification confusion matrix of PTD

	Predicted label				
Reference		Ground	Non-ground	Total	Type I error (%)	1.55
	Ground	1,675,657	26,302	1,701,959	Type II error (%)	5.37
	Non-ground	70,946	1,251,139	1,322,085	Total error (%)	3.22
	Total	1,746,603	1,277,441		

The point-wise classification accuracy of each model is presented in Table 4.4. The type I, type II and total error of ResNet filter are 0.52%, 4.84% and 2.43%, while the errors for PTD are 1.55%, 5.37% and 3.22%, for PMF are 7.82%, 11.62% and 9.48%, respectively. It can be seen that the ResNet model produces the lowest error rates.

Table 4.4 Classification accuracy of ResNet, PTD and PMF

Error rate (%)	ResNet	PTD	PMF
Type I	0.52	1.55	7.82
Type II	4.84	5.37	11.62
Total	2.43	3.22	9.48

4.3.2 RMSE of Interpolated DTM

RMSE is another index that reflects the quality of DTM. After ground points are extracted, interpolation is made to generate raster DTMs. Three interpolation techniques are compared: IDW, ANUDEM and natural neighbour. The RMSEs of interpolated DTMs are shown in Table 4.5. It is not surprising to see that the RMSE results are consistent with point classification accuracies. DTMs generated by ResNet extracted points also have lowest RMSE, which are less than 10 cm. DTMs generated by PTD extracted points have slightly higher RMSE, while DTMs generated by PMF extracted points have RMSEs almost three times higher. Among three interpolation methods, natural neighbour yields the best result for ResNet and PTD extracted points, while ANUDEM yields the best result for PMF extracted points.

Table 4.5 RMSE of interpolated DTMs

RMSE (m)	Resnet	PTD	PMF
IDW	0.0751	0.101	0.313
ANUDEM	0.0816	0.108	0.263
Natural Neighbour	0.0730	0.0944	0.295

4.3.3 Qualitative Assessment

Qualitative assessment is made to examine the filters' performances for different non-ground objects. Certain terrain characteristics are identified as difficult to filter based on visual inspection, such as complex building structure, low vegetation and attached objects. Based on the RMSE analysis, each filter extracted points are interpolated using the most suitable

interpolation technique. Then, hillshade images are generated for each DTM for visual comparison.

4.3.3.1 Vegetation and Buildings in Sloped Areas

Sloped areas with vegetation or building on top are very difficult to filter due to the large variability in slope and terrain discontinuity caused by vegetation or building blockage. This type of terrain is especially troublesome for the PMF filter, which assumes a constant slope parameter for the entire study area. Figure 4.10 shows the filters' performances in a sloped area with trees near the Velocity building. It can be seen that both ResNet and PTD yield only a few errors by misclassifying curbs and grass, while PMF misidentified a large part of terrain as non-ground.

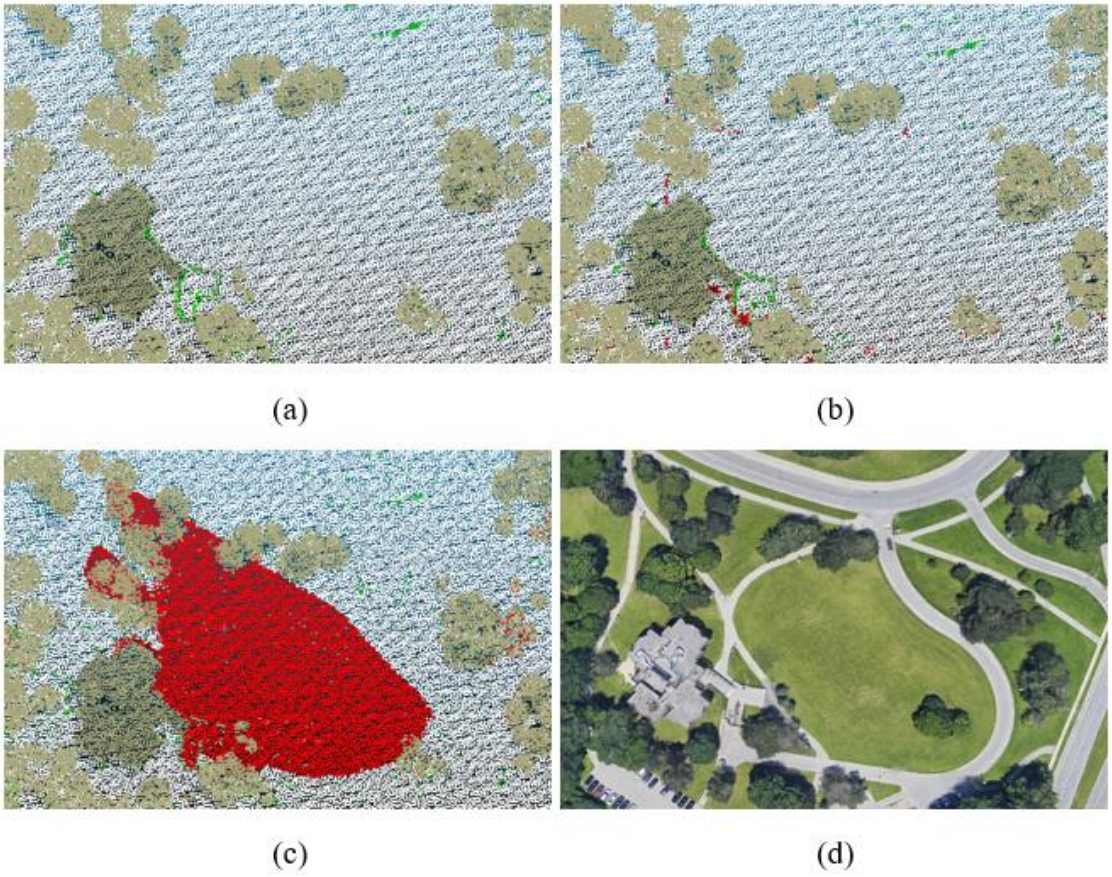


Figure 4.10 Filtered results for sloped area with vegetation: (a) ResNet, (b) PTD, (c) PMF, and (d) aerial image from Google Map

Figure 4.11 shows performances of the filters near Hagey Hall, which is also an elevated area. PMF identified most area near Hagey Hall as non-ground due to its high elevation, while the trees and buildings are correctly classified. The ResNet filter produces a few Type II errors, mainly due to misclassifying vegetation as ground. However, in this scene, PMF has lower Type II errors compare to PTD. Especially in the case of PAS building, where PTD misclassified half of the building as ground. This is a special case where the complexity of building causes trouble for the PTD filter, which will be further discussed in the next subsection.

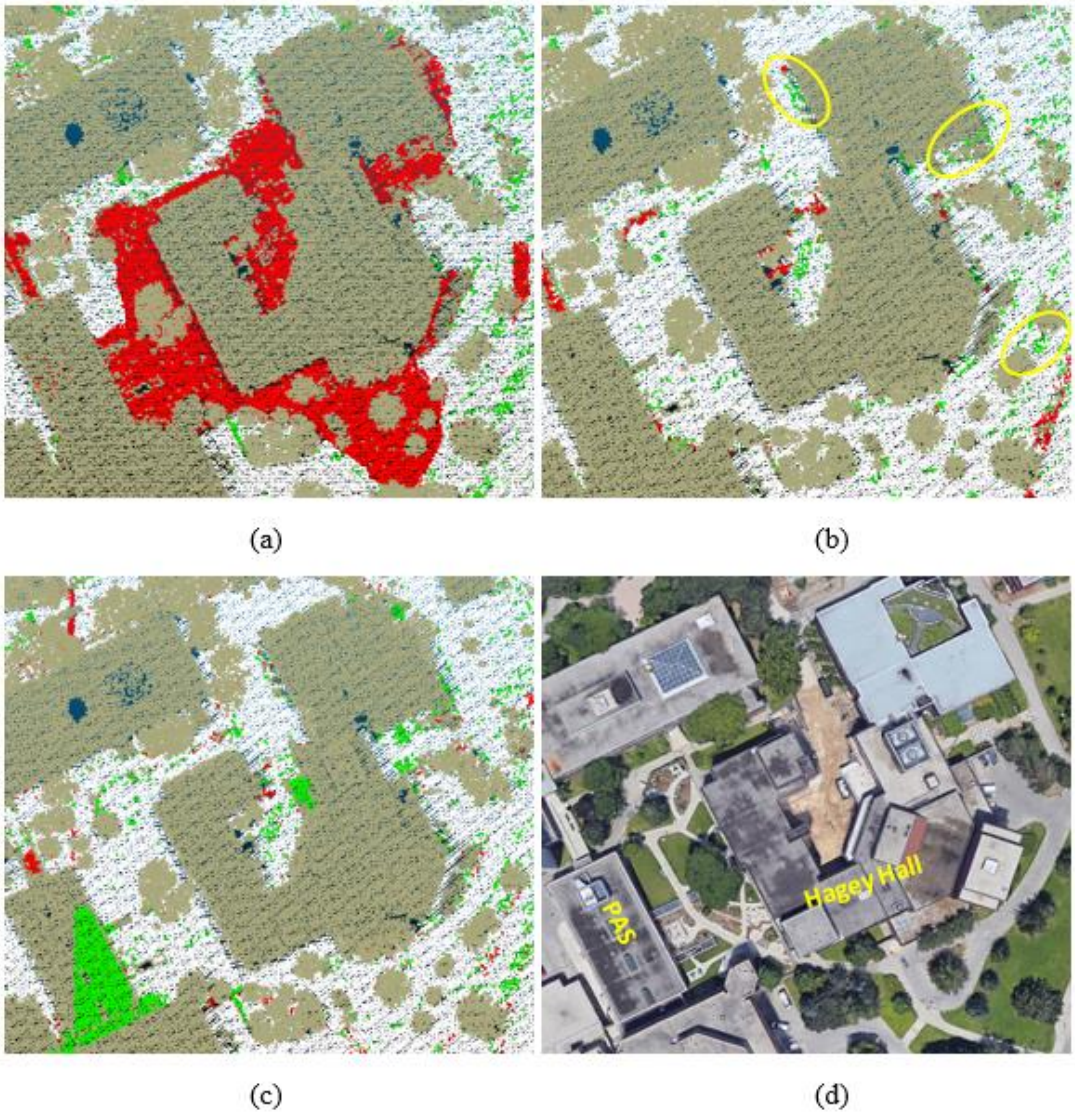


Figure 4.11 Filtered results for sloped area with building. (a) ResNet, (b) PTD, (c) PMF, and (d) aerial image from Google Maps

4.3.3.2 Complex Building Structures

Complex buildings with rooftops at different heights cause trouble for the PTD filter. Two examples of such building structures are shown in Figure 4.12 and Figure 4.13, which is Ron Eydt Village and the Student Life Centre. The building rooftops are at three different

elevations, while the two lower parts of the rooftop are misidentified as ground by the PTD filter. The ResNet filter performs well in this situation since it not only takes into account a point's direct neighbours, but also the elevation difference with all points within 196m distance. The abundant spatial information passed through point to image transformation enables the filter to detect building rooftops even when higher objects present in the scene.

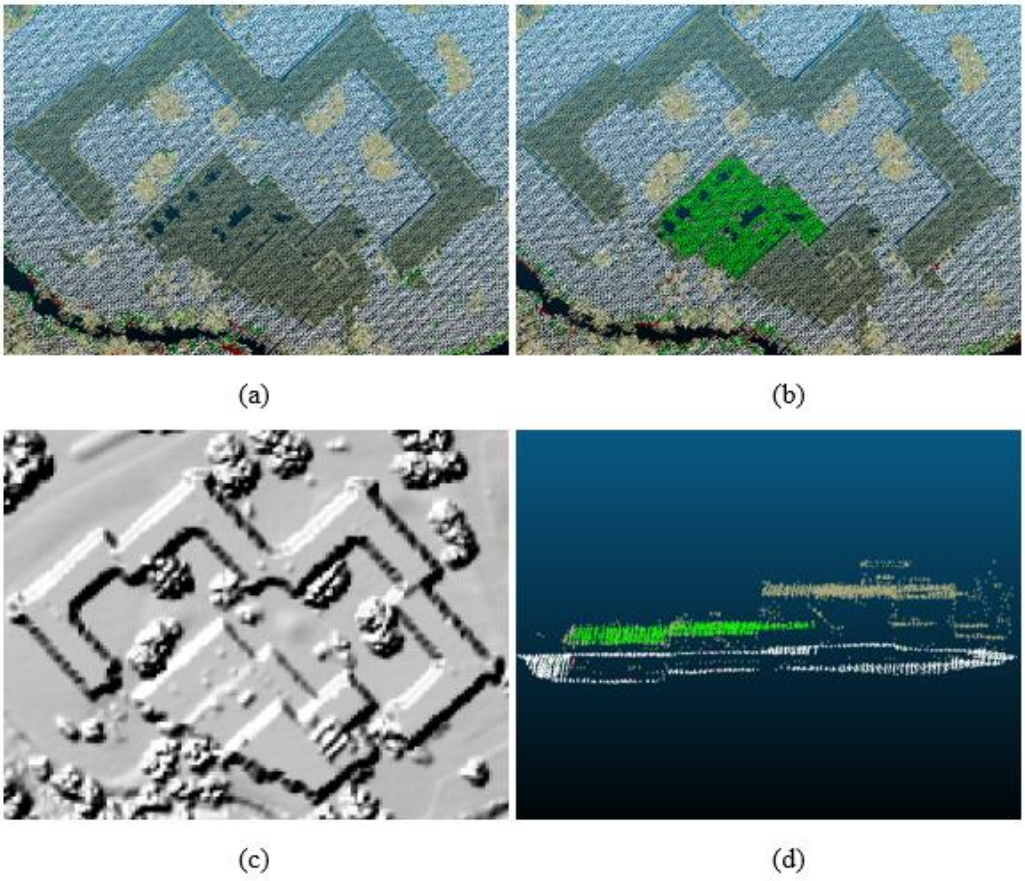


Figure 4.12 Filtered results for complex building (Ron Eydt Village): (a) ResNet, (b) PTD, (c) hillshade image, and (d) side view of (b)

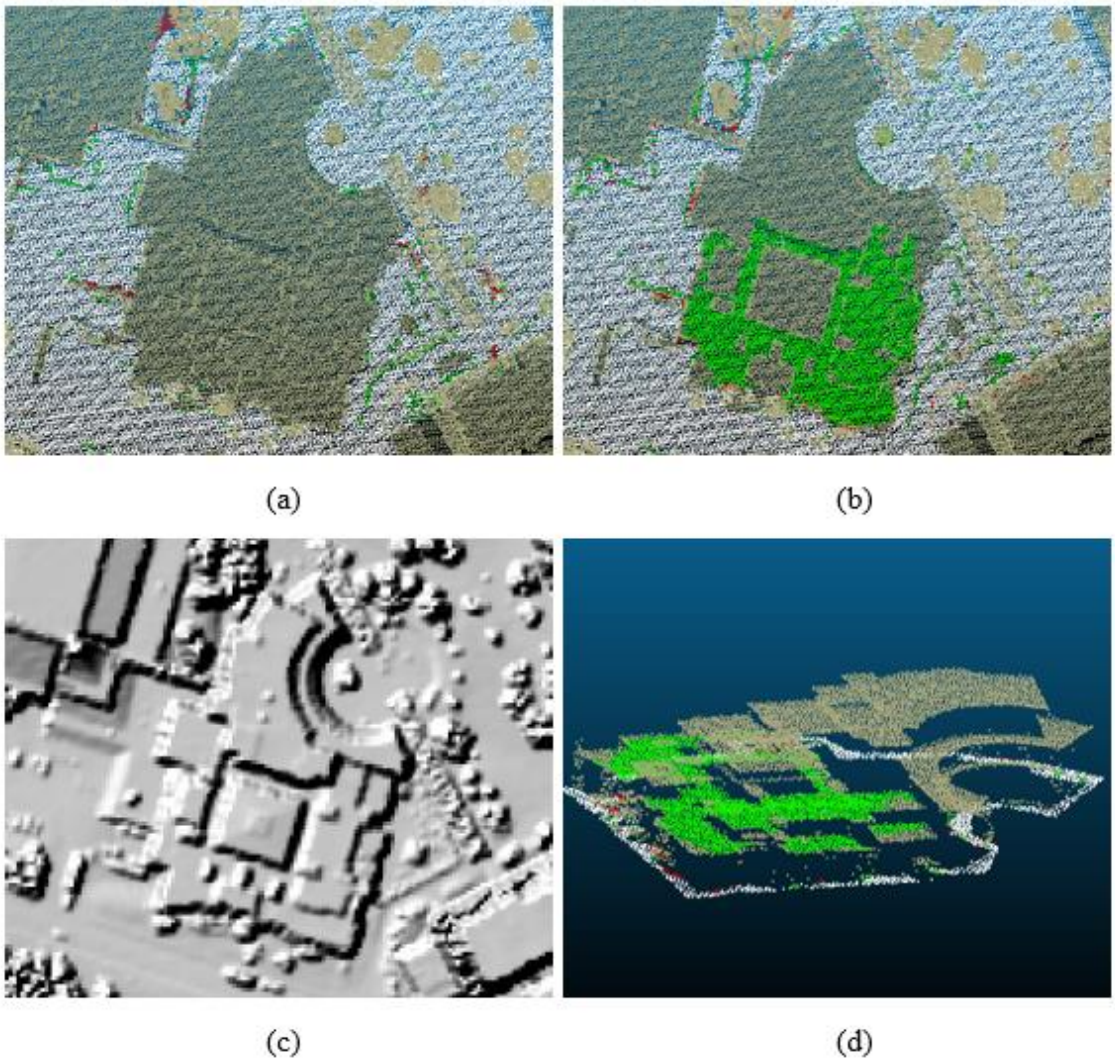


Figure 4.13 Filtered results for complex building (Student Life Centre): (a) ResNet, (b) PTD, (c) hillshade image, and (d) side view of (b)

4.3.3.3 Mixed Buildings and Terrain

Special cases of buildings connected with or built into terrain make it difficult to define the boundary of ground and non-ground. An example of such situation is shown in Figure 4.14. Part of the building is built into the terrain, which makes the building rooftop level with the inner ground. There are three options for classifying this type of structure:

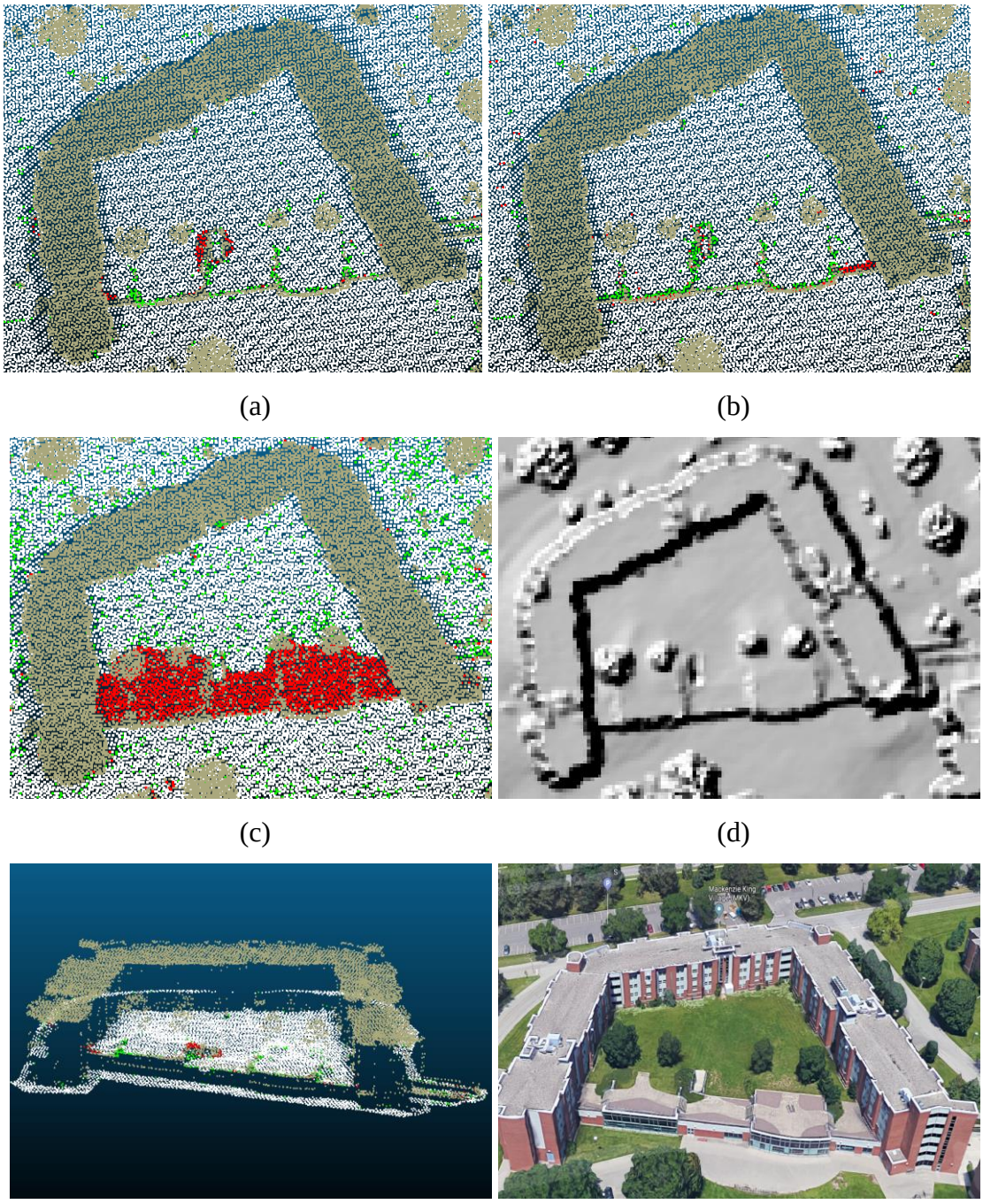


Figure 4.14 Filtered results for connected building rooftop and terrain: (a) ResNet, (b) PTD, (c) PMF, (d) hillshade image, (e) side view of (a), and (f) aerial image from Google Map

- **Option I.** Keep the inner ground and remove all the buildings.
- **Option 2.** Keep the rooftop and the inner ground while removing only the front building façade.
- **Option 3.** Remove the entire building as well as the inner ground. Ground truth label adopts the second option since it preserves most of the spatial information while introduces little error.

Both ResNet and PTD filters abide with the second option: to classify the rooftop and the inner ground as bare earth. However, even though the trees on the ground are removed, the rooftop handrail was misclassified as ground. The PMF complies with the second option, as shown in Figure 4.14 (c), the front building rooftop are all classified as non-ground.

4.3.3.4 Dense Vegetation

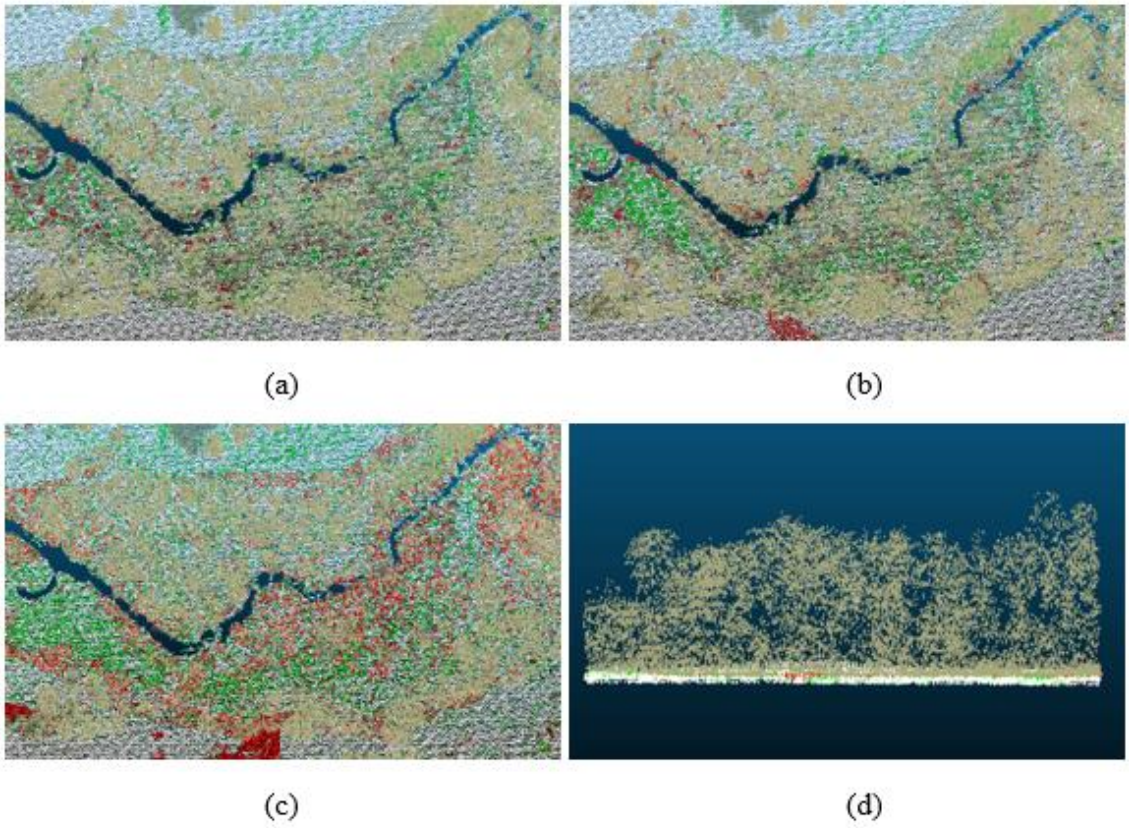


Figure 4.15 Filtered results for dense vegetation: (a) ResNet, (b) PTD, (c) PMF, and (d) front view

All three filters are having trouble identifying terrain in densely vegetated area. As can be seen in Figure 4.15, large quantities of Type I and Type II errors can be observed in all three scenes. However, based on the front view image, high vegetation is correctly classified. The source of the errors come from failing to differentiate between low vegetation and bare ground. It is challenging to differentiate these two classes based only on the height attribute, especially in densely vegetated area where vegetation of various heights is present. A possible solution for this case is to use multispectral LiDAR. With the aid of spectral information, vegetation and ground can be easily distinguished.

4.4 Chapter Summary

This chapter presents the experimental results of proposed workflow. First, the validation accuracies of three ResNet models with four different training percentages are compared. To make a fair comparison, the validation percentage is held constant. With the increase of model depth, there is no obvious improvement in the classification accuracy, while the amount of time used for training increases significantly. Thus, it can be concluded that ResNet18 is the most efficient model for our task. Then, validation accuracies of ResNet18 trained by seven different training data rates from 10% to 70% are compared. Although adding more training data does improve the classification accuracy, the increment is neglectable: with every 10% increase of training data, the validation accuracy improves 0.01%. The training process is largely accelerated by using pre-trained weights on ImageNet. Validation accuracies become stable after 4 epochs.

Second, the proposed workflow is compared with two traditional filters: PMF and PTD. The proposed workflow achieves the lowest Type I, Type II and total error rates, as well as the lowest RMSE of interpolated DTM compared with ground truth. Qualitative analysis shows that the proposed method performs well in specific region with filtering difficulties, such as sloped area with vegetation and building, complex building structure and mixed building and terrain. Despite its success in most of the study area, densely vegetated region remains troublesome.

Chapter 5
Conclusions and Recommendations

This chapter concludes the thesis in Section 5.1 and discusses the limitations and recommendations in Section 5.2.

5.1 Conclusions

In the past decades, the generation of high quality DTM has been advanced by the development in laser scanning technology for fine-resolution point cloud acquisition and the invention of various filtering techniques. However, traditional filters rely on assumptions of terrain morphology such as surface slope, continuity, characteristics of off-ground objects, etc. While these assumptions aid in DTM extraction in designated region, it is difficult for the filter to generalize well to all types of terrains. Moreover, for traditional filters to acquire optimal results, parameters indicating terrain morphology need to be tuned carefully, which is difficult for general users.

This study proposed a workflow for semi-automated generation of DTM using ALS data based on transfer-learning. The proposed workflow was conducted using a subset of the 2014 City of Waterloo LiDAR dataset, which covers the University of Waterloo main campus. Prior to DTM extraction, low outliers were removed due to its destructive impact on the classification algorithm. A statistical outlier removal filter was utilized to remove the noise, low outliers and isolated points in the raw ALS point clouds. Then, to cope with the unordered data structure of point cloud and the CNN requirement of organized input data, point-to-image transformation is conducted. Each ALS point of interest with its neighbouring points were transformed into a feature image based on the elevation differences. The feature images were then served as input for ResNet models. Thus, the DTM extraction task is treated as a binary classification problem. By remapping classified features images into corresponding point cloud, the ground points can be extracted.

The proposed workflow was then compared with two traditional filers (PTD and PMF) in terms of point-wise classification accuracy, RMSE of interpolated DTM, and performance in

special cases. Results show that the proposed extraction workflow is capable of producing high quality DTM with 0.89% Type I error, 3.62% Type II error and 2.1% total error, respectively. Moreover, by using pre-trained weights on ImageNet, the model can achieve high accuracy using only a small percentage of training data. Further analysis of interpolated DTMs reveal that, the RMSE of proposed workflow is 7.3 cm, compared with 9.4 cm produced by PTD and 26 cm produced by PMF. Several special cases that are particularly difficult to filter are presented and discussed. The proposed workflow performed well in most of these cases except for densely vegetated region, where scatters of Types I and II errors can be observed.

In conclusion, the proposed workflow of semi-automated generation of DTMs can extract high quality DTMs accurately and efficiently. The workflow can automatically generate high quality DTM given only small quantity of labeled training sample. The produced DTM has better quality than those produced by traditional filters in terms of classification accuracy, RMSE with ground truth and general performance.

5.2 Limitations and Recommendations

Despite its capability in producing high-quality DTM, the proposed workflow can be improved from the following aspects:

- In order to process large volume of ALS data. The point to image transformation needs to be improved. The generation of feature images was proved to be very time consuming. Also, the storage of feature images requires a lot of space. While each point can be simply represented by three float numbers, the corresponding feature image is a 3-band image with 128*128 pixels.
- Clearer industrial standard of DTM should be made to guide the extraction process. DTM is essentially only an estimation of the terrain surface. Once the terrain is modified by human behaviour, the original morphology can only be restored with approximation, but cannot be modelled free of error. While objects such as buildings and trees should no doubted be removed, controversy arise when it comes to partly attached objects. For example, it is difficult to define the boundary between ground and

non-ground when filtering the building structure shown in section 4.3.3.3. Thus, it is necessary to make a standard for the DTM extraction process in terms of the definition of ground vs. non-ground.

- Based on the findings in Section 4.3.3.4, the proposed workflow struggles in differentiate ground and low vegetation in densely vegetated areas. It is difficult to classify these objects based solely on height attribute. With the recent advancement in multispectral ALS, this problem can be solved by incorporating spectral information.

References

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.). What is a TIN surface? Retrieved from http://desktop.arcgis.com/en/arcmap/latest/manage-data/tin/fundamentals-of-tin-surfaces.htm, last accessed on 18 April 2019.

Axelsson, P. (2000). DEM generation from laser scanner data using adaptive TIN models. *International archives of photogrammetry and remote sensing*, *33*(4), 110-117.

Beraldin, J. A., Blais, F., & Lohr, U. (2010). Laser scanning technology. In Vosselman, G. & Maas, H.-G. (Eds.), *Airborne and Terrestrial Laser Scanning*, 1-42.

Blaschke, T., & Tomljenović, I. (2012). LidarScapes and OBIA. In *Proceedings of the ASPRS. Annual Conference, Sacramento, CA, USA*, 19-23.

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. In *ACM Sigmod Record*, *29*(2), 93–104.

Chen, Q., Wang, H., Zhang, H., Sun, M., & Liu, X. (2016). A point cloud filtering approach to generating DTMs for steep mountainous areas and adjacent residential areas. *Remote sensing*, *8*(1), pp. 71.

Chen, X., Ye, C.,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*, *11*(9), 1-8.

Chen, Z., Gao, B., & Devereux, B. (2017). State-of-the-art: DTM generation using airborne LiDAR data. *Sensors*, *17*(1), pp. 150.

Cvijetinović, Ž., Mihajlović, D., Vojinović, M., Mitrović, M., & Milenković, M. (2008). Procedures and software for high quality TIN based surface. *ISPRS Archives*, *37*, 629–634.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Li, F.-F. (2009). ImageNet: A large-scale hierarchical image database. In *CVPR 2009*, 248-255.

Dorninger, P., & Pfeifer, N. (2008). A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. *Sensors*, *8*(11), 7323–7343.

Farr, T. G., & Kobrick, M. (2000). Shuttle Radar Topography Mission produces a wealth of data. *EOS Transactions American Geophysical Union*, 81(48), 583-585.

Filin, S. (2002). Surface clustering from airborne laser scanning data. *ISPRS Archives*, *34*(3/A), 119–124.

Fisher, P. F., & Tate, N. J. (2006). Causes and consequences of error in digital elevation models. *Progress in Physical Geography*, *30*(4), 467–489.

Gevaert, C. M., Persello, C., Nex, F., & Vosselman, G. (2018). A deep learning approach to DTM extraction from imagery using rule-based training labels. *ISPRS Journal of Photogrammetry and Remote Sensing*, 142, 106–123.

Ghamisi, P., Hofle, B., Zhu, X. X., & Bernhard, H. (2016). Hyperspectral and LiDAR data fusion using extinction profiles and deep convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *10*(6), 1–14.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR 2014*, 580–587.

Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *13th International Conference on Artificial Intelligence and Statistics*, 249-256.

Guan, H., Li, J., Yu, Y., Zhong, L., & Ji, Z. (2014). DEM generation from lidar data in wooded mountain areas by cross-section-plane analysis. *International Journal of Remote Sensing*, *35*(3), 927–948.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *ICCV 2015*, 1026-1034.

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.

Hu, X., & Yuan, Y. (2016). Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote Sensing*, *8*(9), pp. 730.

Hui, Z., Hu, Y., Yevenyo, Y. Z., & Yu, X. (2016). An improved morphological algorithm for filtering airborne LiDAR point cloud based on multi-level Kriging interpolation. *Remote Sensing*, *8*(1), 12–16.

Hutchinson, M. F. (2000). Optimising the degree of data smoothing for locally adaptive finite element bivariate smoothing splines. *ANZIAM Journal 42*, 774-796.

Hutchinson, M. F., Xu, T., & Stein, J. A. (2011). Recent progress in the ANUDEM elevation gridding procedure. *Geomorphometry*, *2011*, 19-22.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML 2015*. https://arxiv.org/pdf/1502.03167.pdf, last accessed on 18 April 2019.

Jochem, A., Höfle, B., Rutzinger, M., & Pfeifer, N. (2009). Automatic roof plane detection and analysis in airborne lidar point clouds for solar potential assessment. *Sensors*, *9*(7), 5241–5262.

Jutzi, B., & Stilla, U. (2005). Waveform processing of laser pulses for reconstruction of surfaces in urban areas. *Measurement Techniques*, *2*(3.1), pp. 2.

Karl, J. W., & Maurer, B. A. (2010). Spatial dependence of predictions from image segmentation: A variogram-based method to determine appropriate scales for producing land-management information. *Ecological Informatics*, *5*(3), 194-202.

Kilian, J., Haala, N., & Englich, M. (1996). Capture and evaluation of airborne laser scanner data.*ISPRSl Archives*, *31*, 383-388.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Leading Edge Geomatics (2015). Kitchener-Waterloo / Cambridge -LiDAR metadata. Retrieved from the University of Waterloo Geospatial Center in Jan 2019.

Lee, J. (1991). Comparison of existing methods for building triangular irregular network, models of terrain from grid digital elevation models. *International Journal of Geographical Information Systems*, *5*(3), 267-285.

Li, J. and Heap, A.D., 2008. A Review of Spatial Interpolation Methods for Environmental Scientists. *Geoscience Australia*, GeoCat # 68229. http://corpdata.s3.amazonaws.com/68229/Rec2008_023.pdf, last accessed on 18 April 2019.

Lin, X., & Zhang, J. (2014). Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments. *Remote Sensing*, *6*(2), 1294-1326.

Liu, X. (2008). Airborne LiDAR for DEM generation: Some critical issues. *Progress in Physical Geography*, *32*(1), 31–49.

Lu, W. L., Murphy, K. P., Little, J. J., Sheffer, A., & Fu, H. (2009). A hybrid conditional random field for estimating the underlying ground surface from airborne lidar data. *IEEE Transactions on Geoscience and Remote Sensing*, *47*(8), 2913-2922.

Matikainen, L., Karila, K., Hyyppä, J., Puttonen, E., Litkey, P., & Ahokas, E. (2017). Feasibility of multispectral airborne laser scanning for land cover classification, road mapping and map updating. *ISPRS Archives*, *42*(3/W3), 119-122.

Meng, X., Currit, N., & Zhao, K. (2010). Ground filtering algorithms for airborne LiDAR data: A review of critical issues. *Remote Sensing*, *2*(3), 833–860.

Meng, X., Wang, L., Silván-Cárdenas, J. L., & Currit, N. (2009). A multi-directional ground filtering algorithm for airborne LiDAR. *ISPRS Journal of Photogrammetry and Remote Sensing*, *64*(1), 117–124.

Ministry of Natural Resources and Forestry (2016). Southwestern Ontario Orthophotography Project (SWOOP) 2015 Digital Elevation Model User Guide, https://uwaterloo.ca/library/geospatial/sites/ca.library.geospatial/files/uploads/files/swoop_2015_dem_-_user_guide.pdf, last accessed on 18 April 2019.

Mongus, D., & Žalik, B. (2014). Computationally efficient method for the generation of a digital terrain model from airborne LiDAR. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(1), 340–351.

Natural Resources Canada (2018). High Resolution Digital Elevation Model (HRDEM) CanElevation Series Product Specifications. Retrieved from: http://ftp.maps.canada.ca/pub/elevation/dem_mne/highresolution_hauteresolution/HRDEM_Product_Specification.pdf, last accessed on 28 April 2019.

Niemeyer, J., Rottensteiner, F., & Soergel, U. (2014). Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, *87*, 152–165.

Pfeifer, N., Stadler, P., & Briese, C. (2001). Derivation of digital terrain models in the SCOP++ environment. In *Proceedings of OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Terrain Models, Stockholm, Sweden*, Vol. 3612.

Politz. F., Kazimi. B., Sester. M. (2018). Classification of laser scanning data using deep learning. *Wissenschaftlich-Technische Jahrestagung der DGPF und PFGK18 Tagung in München.* Vol.27.

Puente, I., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2013). Review of mobile mapping and surveying technologies. *Measurement: Journal of the International Measurement Confederation*, *46*(7), 2127–2145.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 5099-5108.

Ripley, B. D. (2005). *Spatial Statistics*, Vol. 575. Hoboken, New Jersey: John Wiley & Sons.

Rizaldy, A., Persello, C., Gevaert, C. M., & Oude Elberink, S. J. (2018). Fully convolutional networks for ground classification from LiDAR point clouds. *ISPRS Annals*, *4*(2) 231-238.

Roggero, M. (2001). Airborne laser scanning-clustering in raw data. *ISPRS Archives*, *34*(3/W4), 227-232.

Roggero, M. (2002). Object segmentation with region growing and principal component analysis. *ISPRS Archives*, *34*(3/A), 289–294.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Shao, Y. C., & Chen, L. C. (2008). Automated searching of ground points from airborne lidar data using a climbing and sliding method. *Photogrammetric Engineering & Remote Sensing*, *74*(5), 625-635.

Silván-Cárdenas, J. L., & Wang, L. (2006). A multi-resolution approach for filtering LiDAR altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *61*(1), 11–22.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *ICLR 2015*. https://arxiv.org/pdf/1409.1556.pdf, last accessed on 18 April 2019.

Sithole, G. (2001). Filtering of laser altimetry data using a slope adaptive filter. *ISPRS Archives, 34*, 203–210.

Sithole, G., & Vosselman, G. (2004). Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *59*(1-2), 85-101.

Sithole, G., & Vosselman, G. (2005). Filtering of airborne laser scanner data based on segmented point clouds. *ISPRS Archives*, *36*(3), 66-71.

Skaloud, J., & Schwarz, K. P. (2000). Accurate orientation for airborne mapping systems. *Photogrammetric Engineering and Remote Sensing*, *66*(4), 393–401.

Sotoodeh, S. (2007). Hierarchical clustered outlier detection in laser scanner point clouds. *ISPRS Archives*, *36*(3), 383-388.

Su, W., Sun, Z., Zhong, R., Huang, J., Li, M., Zhu, J., … Zhu, D. (2015). A new hierarchical moving curve-fitting algorithm for filtering lidar data for automatic DTM generation. *International Journal of Remote Sensing*, *36*(14), 3616–3635.

Susaki, J. (2012). Adaptive slope filtering of airborne LiDAR data in urban areas for digital terrain model (DTM) generation. *Remote Sensing*, *4*(6), 1804-1819.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *In proceedings of CVPR*, 1-9.

Tóvári, D., & Pfeifer, N. (2005). Segmentation based robust interpolation-a new approach to laser data filtering. *ISPRS Archives*, *36*(3/19), 79-84.

Vain, A., Yu, X., Kaasalainen, S., & Hyyppa, J. (2010). Correcting airborne laser scanning intensity data for automatic gain control effect. *IEEE Geoscience and Remote Sensing Letters*, *7*(3), 511-514.

Vosselman, G. (2000). Slope based filtering of laser altimetry data. *ISPRS Archives*,*33*(B3/2-3), 935-942.

Vosselman, G., & Maas, H. G. (Eds.) (2010). *Airborne and Terrestrial Laser Scanning*. Boca Raton: Taylor & Francis, CRC Press.

Wu, B., Yu, B., Huang, C., Wu, Q., & Wu, J. (2016). Automated extraction of ground surface along urban roads from mobile laser scanning point clouds. *Remote Sensing Letters*, *7*(2), 170–179.

Yang, Z., Tan, B., Pei, H., & Jiang, W. (2018). Segmentation and multi-scale convolutional neural network-based classification of airborne laser scanner data. *Sensors*, 18(10), 3347. doi: 10.3390/s18103347.

Zakšek, K., Pfeifer, N., & IAPŠ, Z. S. (2006). An improved morphological filter for selecting relief points from a LiDAR point cloud in steep areas with dense vegetation. Retrieved from: https://iaps.zrc-sazu.si/sites/default/files/Zaksek_Pfeifer_ImprMF.pdf, last accessed on 18 April 2019.

Zhang, J., & Lin, X. (2013). Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification. *ISPRS Journal of Photogrammetry and Remote Sensing*, *81*, 44–59.

Zhang, J., Lin, X., & Ning, X. (2013). SVM-Based classification of segmented airborne LiDAR point clouds in urban areas. *Remote Sensing*, *5*(8), 3749–3775.

Zhang, K., Chen, S. C., Whitman, D., Shyu, M. L., Yan, J., & Zhang, C. (2003). A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, *41*(4), 872-882.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., & Yan, G. (2016). An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sensing*, *8*(6), 1–22.

Zhao, K., Popescu, S., & Nelson, R. (2009). Lidar remote sensing of forest biomass: A scale-invariant estimation approach using airborne lasers. *Remote Sensing of Environment*, 113(1), 182–196.